Identifikacija šablona ponašanja studenata korišćenjem podataka o pokretima očiju

Andrija Cvejić
Računarstvo i automatika
Fakultet tehničkih nauka, Univerzitet u Novom Sadu
Novi Sad, Srbija
andrijacvejic@uns.ac.rs

***Abstrakt* — Predmet ovog rada je praćenje pogleda studenata tokom provere znanja. Takođe, da se pribavi informacije o pogledu studenata za svako individualno pitanje na testu. Ove informacije su bitne ljudima koji sastavljaju testove, one nam pružaju uvid u proces razmišljanja studenata tokom rada na testu. Pored toga, glavni cilj projekta je da grupiše studente po načinu gledanja tokom rešavanja testa. Za postizanje smislenog grupisanja vršimo sledeće korake. Prvo, filtracija podataka dobijenih iz aparata za praćenje očiju, gde uklanjamo šum u podacima. Drugo, pretvaraju se podaci u sekvencu pogleda nad regione, time pretvaramo informacije koordinata u regione i vreme sprovedeno u njemu; to se vrši pomoću mapiranja koordinata nad regione od interesa, gde su regioni obeleženi od strane osobe koja je kreirala test. Treće, takvi podaci se klasteruju pomoću više nenadgledanih algoritma za klasterovanje, kao što su: *hirarhisko* i *DBscan (Density-based spatial clustering of applications with noise)*. Konačno vršimo tumačenje rezultata dobijenih klastera, time potvrđujemo da li su napravljeni smisleni klasteri.**

Ključne reči—klusterovanje; eye tracker; dbscan clustering; hierarchical clustering;sgt;

I. Motivacija

Tokom provere znanja mi gledamo tačnost odgovora na pitanja. Međutim, to nam ne daje potpunu sliku toka razmišljanja studenata tokom odlučivanja za odgovor. Zbog ovog, uključujemo analizu pogled studenata, time možemo da uradimo sledeće stvari. Prvo, možemo da poboljšamo kvalitet pitanja, gde uklanjamo lose formirana ili dvosmislena pitanja; na primer to se može primetiti dugim zadržavanje pogleda studenata na pitanje. Drugo, možemo da dobijemo fiziološke karakteristike studenata. Treće, možemo da dobijemo šablon ponašanja studenata kod rešavanja određenih pitanja, time možemo da vidimo koji pristup je uspešniji za određeni problema. Četvrti, da profesori razumeju uzrok problema kod studenata, time mogu da koriguju pristup materiji ili dodatno objasne neke oblasti; tada značajno poboljšavamo edukaciju budućih generacija.

II. Istraživačka pitanja

Cilj rada je da svrsta studente u šablone ponašanja i da budemo u stanju da novog studenta svrstamo u već postojaće šablone, da bi to postigli cilj moramo odgovoriti na sledeća pitanja. Kako ćemo vreme da uračunamo u klasterizaciji iz *eye tracker*-a, za ovaj problem smo prihvatili zaključak kod rada [2], gde su primetili preveliki uticaj vremena na podatake, tada se gubi smisao sadržaja i postaje pseudo nasumično klasterovanje; da bi umanjili uticaj vremena primenjujemo metode transformacije vremena u atribute. Kako da iz sekvence podataka povežemo sa semantičkim regionima, da bi dodelili semantiku pažnje studenata. Kako da proverimo klastere da li su dobro formirani? Za validaciju podataka će se koristiti vizualni pristup koji je predložen u radu [1].

III. Metodologije

Skup pitanja je napravljen sa semantičkim obeleženim regionima. Podaci iz *eye tracker*-a se prikupe na skup pitanja prethodno napravljen, zatim testirano nad studentima. Tada se vrši filtracija podataka, zatim se dobijena sekvenca pogleda mapira na regione od interesa. Nakon toga, vrši se sabijanje regiona, u slučaju da se gledalo dva puta zaredom u isti region; prilikom sabijanja regiona vremena se saberu. Nakon sabijanja regiona, vršimo diskretizaciju vreme za svaki region. Rezultat diskretizacije zamenjuje atribut vremena, on može da ima vrednosti malo, srednju ili veliku pažnju posvećenu nad regionom. Zatim se vrši klasterizacija nad skupom regiona. Konačno se vrši validacija podataka.

IV. Rešenje/Diskusije

Podaci o pogledima čoveka će biti prikupljeni pomoću aparata *eye tracker*-a marke “Gaze Point” i softvera *Gazepoint Analysis* koji je razvijen od strane firme *Gaze Point*. Klasterizacija se vrši pomoću vise algoritama zbog poređenja rezultata, gde su korišćeni algoritmi hirarhiskog i *DBScan* klasterovanja. Konačno se vrši provera dobijenih rezultata klastera uz pomoć više vizualnih reprezentacija, konkretno se koristi “t-distributed stochastic neighbor embedding” (t-SNE)

978-1-5386-1066-4/18/$31.00 ©2018 IEEE

[5] i *scatter plot*. Data realizacija rešenja je izvršena u programskom jeziku *Python*.

A. Prikupljanje podataka pomoću gaze point analysis

Podaci su prikupljeni sa opremom *GP3 Eye Tracker,* gde je učestvovalo 23 studenata i odgovaralo se na 33 pitanja. Uviđeno je više problema koji su nastali u toku realizacije ovog dela, oni su nastali zbog loše preciznosti opreme prilikom merenja. Prvi problem, primećeno je da za neke studente nije adekvatno pratio pokret očiju. Odnosno eye tracker je imao *off-set* za svakog studenata drugačiji, primer toga je registrovanje 5cm udesno pogleda u odnosu na zapravo gde se gleda. Primenjeno je rešenje kod sigurno pogrešnih ekstremnih slučajeva, za svakog studenta ručno ispravljeno u softveru za njegov uočen *off-set u slučaju da postoji*. Drugi problem, predstavlja lošu implementaciju softvera *Gazepoint analysis* zbog koga dodavanje znanja, odnosno razdvajanja podataka po pitanjima se moralo da vrši dva puta. Reč je o lošem čuvanju csv podataka, prilikom odabira čuvanja podataka vremenski period se može izaberati, međutim promena vremenskog perioda za eksportovanje podataka ne menja koji podaci se sačuvaju (Uvek se za ceo interval testa sačuvaju podaci). U Tab. 1 pogledati značenje parametra dobijenih od softvera.

TABELA I. ZNAČENJE PODATAKA DOBIJENIH OD SOFTVERA GAZEPOINT ANALYSIS (SAMO MALI DEO PODATAKA)

Param	**Tip**	**Opis**
TIME	*float*	Proteklo vreme u sekundama od prethodne sistemske inicijalizacije.
TIME TICK	*long long*	Brojač otkucaja
FPOGV	*int*	Oznaka validnosti tačke fiksacije pogleda
FPOGID	*int*	Oznaka tačke fiksacije
FPOGX	*float*	Tačke fiksacije X kord.
FPOGY	*float*	Tačke fiksacije Y kord.
FPOGS	*float*	Početak fiksacije (sek.)
FPOGD	*float*	Dužina fiksacije (vreme od početka fiksacije (sek.))
LPOGX	*float*	Leva tačka pogleda X kord.
LPOGY	*float*	Leva tačka pogleda Y kord.
LPOGV	*int*	Oznaka validnosti leve tačke pogleda
RPOGX	*float*	Desna tačka pogleda X kord
RPOGY	*float*	Desna tačka pogleda Y kord.
RPOGV	*int*	Oznaka validnosti desne tačke pogleda

B. Deljenje podataka

Ovaj korak se odnosi na dodavanje dodatne semantičke vrednosti redovima (deljenje torki na pitanja), odnosno na osnovu vremenske odrednice da se dodeli kojem pitanju ono pripada. Ovaj postupak je rađen ručno tako što se gledao video snimak praćenja pogleda i zapisivanjem vremena. Prvobitno je pokušano u softveru *gazepoint analysis*, međutim njegova funkcionalnost *export* za određeno vreme je uočeno da ne radi, zatim je rađeno u *Python*-u.

C. Filtracija podataka

Filtracija podataka podrazumeva više izvršenih koraka:

1)Prvi korak ovog postupka predstavlja identifikacija korisnih informacijama u podacima dobijenih od strane *gazepoint analysis*. Od svih dobijenih izabrani su TIME, FPOGD, FPGOX, FPOGY i FPGOV za dalju obradu.

2)Drugi korak se odnosi na transformaciju podataka funkcijom f(fpogx,fpogy) -> region od interesa. Regioni od interesa u ovom radu se odnose na konkretne regione kao što su ostali, pitanje, odgovor pod 1, odgovor pod 2, odgovor pod 3, odgovor pod 4 i dugme za potvrdu izabranog rešenja sa nastavljanjem na sledeće pitanje. Dati regioni su obeleženi (kordiante X i Y) od strane ljudi koji su sastavljali test.

3)Treći korak predstavlja uklanjanje suvišnih podataka (redukcija), tako što sabijamo u podacima redove (n-torke) koje uzastopno pripadaju istom regionu, gde se vreme sabira. Sabijanja se može predstaviti primerom jedne sekvence, gde red podataka sadrži oznaku regiona i vreme provedeno u njemu, rezultat za datu sekvencu “S2A2A2A3S1” kroz funcije sabijanja g(x) je g(S2A2A2A3S1) -> S2A7S1.

4)Četvrti korak se odnosi na uklanjanje invalidnih podataka gde se vrši na dva načina. Prvi je pomoću parametra FPOGV, uklanjaju se torke u slučaju niskog vremena nakon trećeg koraka. Uklanjanje pomoću FPGOV je zbog rečene netačnosti u dokumentaciji ostalih podataka u slučaju *false* vrednosti. Drugi predstavlja uklanjanje u odnosu na jako nisku vrednost vremena, to se odnosi na uklanjanje šuma u podacima.

Na kraju filtracije podaci koji nam ostanu su:

- Region od interesa (*int* val ∈{-1,0,1,2,3,4,5})
- Koje je pitanje (*int* val ∈{1..33})
- Trajanje pogleda(*float*)

D. Uticaj vremena na klasterovanje

Zbog prevelikog uticaja vremena na klasterizovanje podataka, odnosno čini klasterizaciju podataka približno pseudo-nasumičnom izborom podataka[2]. Iz tog razloga vršimo transformaciju vremena u drugi oblik, odnosno diskretizujemo vreme u kategorije malo, srednje i puno pogled proveden u regionu od interesa.

E. Problem varabilnih sekvenci prilikom klasterovanja

Pre korišćenja *DBscan* ili hirarhiskog algoritma za klasterovanje naišli smo na problem sa sekvencom.

Odnosno cilj projekta je da se uoči šablon ponašanja kod studenata prilikom odgovaranja na pitanje, iz tog razloga se razdvajaju podaci na pitanja. Međutim kod ljudi nije jednak vremenski interval odgovaranja na testu, iz tog razloga sekvence za klasterovanje nisu jednake. Da bi rešili ovaj problem primenjen je algoritam *SGT* (*Sequence Graph Transform*)[4], dati algoritam primenjuje *sekvence embedding* vektora regiona i trajanje u visoko dimensioni vektor brojeva (~400 dimensioni). Gde se u nekoj meri čuvaju odnosi kratkih i dugih veza originalnog vektora.

F. Problem sa visoko dimensionim vektorima

Zbog validacionih razloga dimensionalnost se morala redukovati kod vektora dobijenih od strane *SGT* algoritma. Za predlog rešenja iskorišćen je procedura *PCA* (*Principal component analysis*)[6]. Takođe iskorišćen je mašinski pristup za redukciju dimensionalnosti za prikaz zvanim *t-SNE.*

V. Rezultati

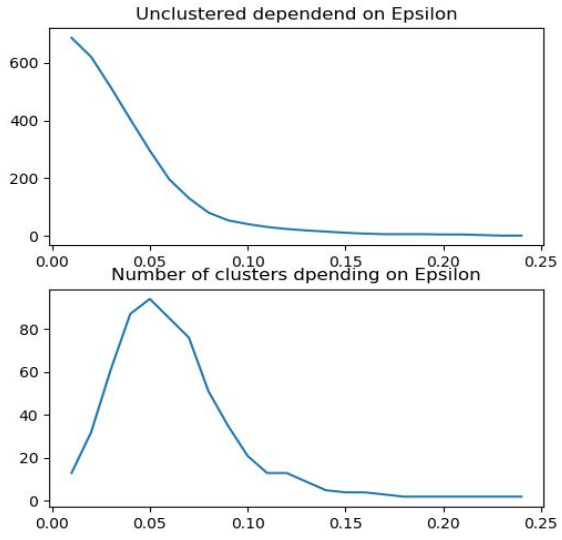
Nakon svih koraka u rešenju poredimo rezultate hirahiskog i *Dbscan*-a, sa više različitih tehinika prikaza (*t-SNE* i *Scatter plot*).

A. Uočavanje koliko treba izabrati klastera

Za uspešnu analizu odgovarajućeg broja klastera kod različitih algoritama, korišćene su različite vizualne tehnike:

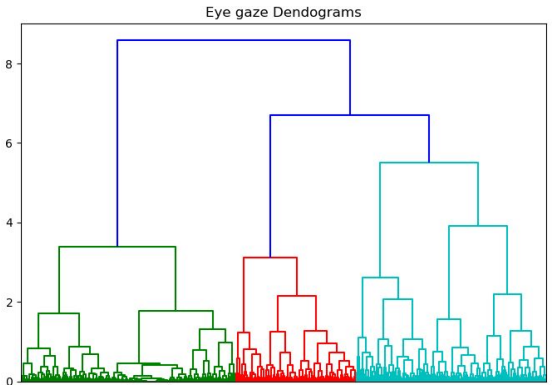
1) Broj klustera kod *DBscan* zavisi od *epsilon*-a, njegov odnos se prikazuje na Fig. 1 sa brojem klastera i šumom. Ostali parametri za *DBscan* jesu metrika za rastojanje i minimalan broj torki u klasteru, gde su oni *Euklidsko rastojanje* i dva elementa respektivno. Korišćena je *sklearn* implementacija *DBscan* biblioteke u *Python*-u. Za dalju klasterizaciju epsilon **je 0.1** vrednost koja se koristi.

Figura 1. Zavisnost Epsilon-a u odnosu na broj klastera i broj podataka koji nisu u klasteru kod DBscan-a



2) Broj klastera kod hirarhiskog klasterovanja se može utvrditi pomoću dendrograma, za ovaj problem dendrograma se vidi na slici 2.. Na osnovu njegove izgleda i interpretacije se izabere željeni broj klustera. Korišćena je *sklearn* implementacija *DBscan* biblioteke u *Python*-u. Za uspešan vizuelni prikaz dendrograma je korišćen *PCA* procedura za redukciju dimensionalnosti. Interpretracijom dendrograma kod Fig. 2 je utvrđeno odgovarajući broj klastera deset.

Figura 2. Dendrogram pogleda po pitanjima svih studenata



B. Broj jedinki u klasterima nakon odabranih broja klastera

TABELA II. Značenje podataka dobijenih od softvera Gazepoint Analysis (Samo mali deo podataka)

Broj klastera	**Broj jedinki u *DBscan* klasteru**	**Broj jedinki u hirarhiskom klasteru**
1.	**644**	72
2.	*8*	111
3.	*2*	129
4.	*3*	55
5.	*2*	64
6.	*2*	166
7.	*3*	27
8.	*3*	38
9.	*3*	38
10.	*10*	50
11.	*2*	//
12.	*2*	//
13.	*7*	//
14.	*2*	//
15.	*2*	//
16.	4	//
17.	2	//
18.	2	//
19.	3	//
20.	3	//
-1.	**39**	//

Primećen je trend kod Tab. 2 za *DBscan*, da sve torke sekvenca pitanja teže da se pridruže prvom klasteru, dok

ostali uglavnom sadrže samo do 10 elemenata. U slučaju da se izabere manji epsilon kod *DBscan*-a, odnosno da imamo više klastera, isto ponašanje je primećeno samo se pojavi veći broj malih klastera i neklasterovanih elemenata (srp. šum, eng. *noise*). Međutim kod hirarhiskog vidimo bolju distribuciju sličnih elemenata u klasaterima.

C. Vizualni prikaz klastera

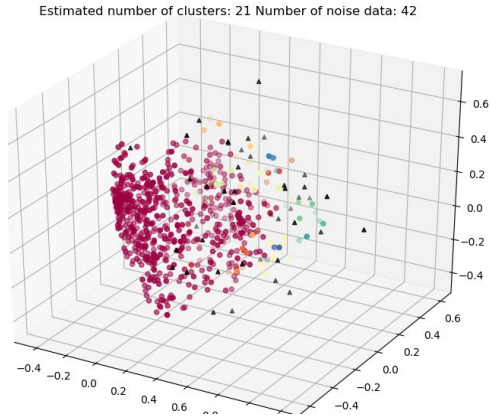
Elementi vizualizacije se odnosi za:

1) *DBscan* klasterovanje
2) Hirarhisko klasteraovanja

Vizualizacija se vrši pomoću *t-SNE* i *scatter plot* sa *PCA*. Za prikaz su isti parametri korišćeni za *DBscan* i hirarhisko klasterovanje kod *t-SNE* i *scatter plot*.

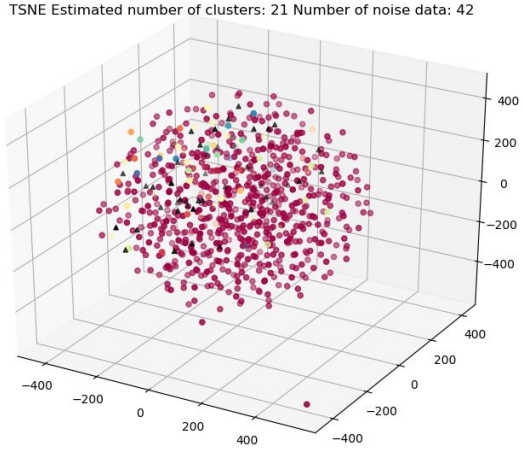
1a) Za *DBscan* na Fig. 3. se može primetiti da većina je pripalo prvom klasteru (crvena boja) i svi se nalaze u levom delu prostora. Dok mali klasteri se nalaze desnom delu prostora.

FIGURA 3. PRIKAZ SCATTER PLOT-A ZA DBSCAN-A



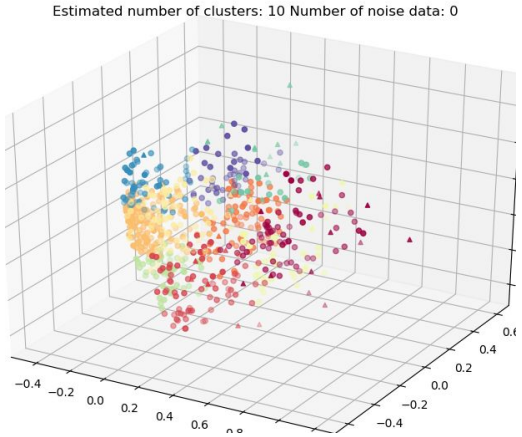
1b) Za prikaz *DBscan* na Fig. 4, za algoritam t-SNE korišćeni su parametri *perplexity* sa 50, *n_components* sa 3 i *verbos* sa 1. Pokušalo se iscrtavanje na 2d i 3d sa ovom tehinkom, međutim nije se moglo jasno razdvojiti prvi klaster od drugih. Nije značajno znanje dobijeno od ove tehnike (moguć uzrok zbog loše podeljenih klastera).

FIGURA 4. PRIKAZ T-SNE ZA DBSCAN-A



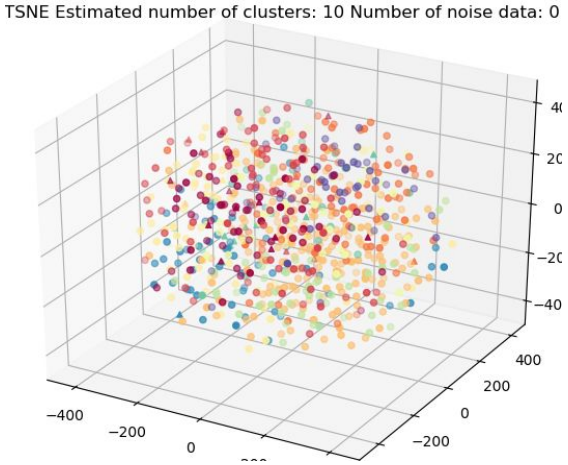
2a) Za prikaz hirarhiskog klasterovanja na Fig. 5 pomoću scatter plot-a sa *PCA*. Vidimo znatno bolju podelu klastera u odnosu na prostor, time dobijamo više korsnih informacija, zapravo se vidi podela na par šablona ponašanja studenata.

FIGURA 5. PRIKAZ SCATTER PLOT-A ZA HIRARHISKO CLASTEROVANJE



2b) Na Fig. 6. se prikazuje rezultat hirarhiskog klasterovanja pomoću t-SNE metode. Međutim takođe isto se desilo, nisu se uspele izvući korisne informacije, iako su probati preporučeni parametri *perplexity* od 5 do 50.

FIGURA 6. PRIKAZ T-SNE ZA HIRARHISKO CLASTEROVANJE



VI. Zaključak

Izvršena je ručna analiza podataka u odnosu na dodeljen klaster kojem pripada sekvenca, gde poredimo uočene stvari za hirahisko i *DBscan*-a klasterovanje. Prilikom pregleda *DBscan*-a nije se moglo ustanoviti po kom kriterijumu elementi pripadaju najvećem klasteru, dok ostali klasteri sa 3-10 elemenata su sačinjeni od istim ponavlajućim sekvencama. Dubljom analizom hirarhiskog klasterovanja jeste dalo više informacija, gde neke osobine koje čine klaster karakteristične su:

1)Pogledi koji su takvi da poćinju na pitanju zatim kratko na jednom odgovoru.(Uglavnom podrazumeva kratko gledanje)

2)Pogledi ka više odgovora, onda pitanje tako da se vraća više puta na različita pitanja. (uglavnom nezavisno od vremena)

3)Pogledi ka samo odgovorima (uglavnom jedan ili dva odgovora duže).

Primeti se uticaj vremena na klasterovanje, tako da nije unistilo podatke. Prethodno uočena ponašanja su primećene kod više sekvenci. Međutim kod svih ovih primera klastera, postoje sekvence koje ne pripadaju ovom ponašanju a pripadaju njemu, takođe ostale sekvence je teško bilo razdvojiti u jednu grupu. Za bolje utvrđivanje šablona kod studenata neophodno je prikupiti veči broj podataka. Takođe, onda će biti bolja i analiza samih tih podataka.

Reference

[1] Göbel, F., & Martin, H. (2018). Unsupervised Clustering of Eye Tracking Data. ETH Zurich. https://doi.org/10.3929/ETHZ-B-000290476

[2] Keogh, E., & Lin, J. (2005). Clustering of time-series subsequences is meaningless: implications for previous and future research. Knowledge and Information Systems, 8(2), 154–177. https://doi.org/10.1007/s10115-004-0172-7

[3] Guralnik, V., & Karypis, G. (n.d.). A scalable algorithm for clustering sequential data. Proceedings 2001 IEEE International Conference on Data Mining. Presented at the 2001 IEEE International Conference on Data Mining. https://doi.org/10.1109/icdm.2001.989516

[4] Ranjan, Chitta & Ebrahimi, Samaneh & Paynabar, Kamran. (2016). Sequence Graph Transform (SGT): A Feature Extraction Function for Sequence Data Mining https://arxiv.org/abs/1608.03533

[5] van der Maaten, L.J.P.; Hinton, G.E. (Nov 2008). "Visualizing Data Using t-SNE" (PDF). Journal of Machine Learning Research. 9: 2579–2605.

[6] Pearson, K. (1901). "On Lines and Planes of Closest Fit to Systems of Points in Space". Philosophical Magazine. 2 (11): 559–572. doi:10.1080/14786440109462720.