Mašinsko učenje

Modeliranje tema

Nemanja Maček

Sadržaj

Uvodne napomene

U čemu je razlika između klasterovanja i modeliranja tema?

- Na prethodnom predavanju grupisali smo tekstualne dokumente pomoću klasterovanja.
- Svaki dokument nakon klasterovanja pripada tačno jednom klasteru.
- Međutim, predpostavite da imamo knjigu koja se bavi mašinskim učenjem i programskim jezikom Python.
- Postavljamo pitanje: u koji klaster treba svrstati knjigu – u klaster u kome su grupisane knjige o mašinskom učenju ili u klaster u kome su grupisane knjige o Python-u?
- U klasičnoj knjižari knjiga će se nalaziti na jednoj polici u izlogu (dakle, u pitanju je jedna tema).
- U Internet prodavnici ovu knjigu možemo da obeležimo i kao knjigu koja se bavi mašinskim učenjem i kao knjigu koja se bavi Python-om, što znači da bi je trebalo navesti u dva odjeljka u on-line knjižari (dve teme).
- To ne znači da će knjiga biti navedena u svim odeljcima, npr. u odeljku u kome se nalaze knjige o kuvanju, okultizmu i slično.

Uvodne napomene

Šta ćemo raditi na ovom času?

- Na ovom času ćemo obraditi metode koje ne grupišu dokumene u potpuno odvojene grupe, već dopuštaju da se svaki dokument odnosi na nekoliko tema, pri čemu se teme automatski identifikuju iz kolekcije tekstualnih dokumenata.
- Podoblast mašinskog učenja koja se bavi ovim problemom naziva se modeliranje tema.
- Dokumenti mogu biti knjige ili kraći tekstovi poput blogova, vesti ili e-pošte.
- Neke teme su za dati dokument „centralne“, dok su neke više „periferne“. Na primer, u prethodnim beleškama se često pominje vizuelizacija – međutim, ova tema je više periferna jer je fokus izlaganja bio na mašinskom učenju.

LDA

Naravite razliku.

- LDA je skraćenica od engleskih naziva sledećih metoda:
  - *latent Dirichlet allocation*, i
  - *linear discriminant analysis*.
- Metode se odnose na potpuno različite stvari – modeliranje tema i klasifikaciju, respektivno.

LDA

Jednostavno objašnjenje.

- Pretpostavićemo da imamo sledeće rečenice:
  - 1. „I ate a banana and spinach smoothie for breakfast.“
  - 2. „I like to eat broccoli and bananas.“
  - 3. „Chinchillas and kittens are cute.“
  - 4. „My sister adopted a kitten yesterday.“
  - 5. „Look at this cute hamster munching on a piece of broccoli.“
- LDA je način automatskog otkrivanja tema koje sadrže ove rečenice.

LDA

Jednostavno objašnjenje.

- Za prethodne rečenice i dve teme, LDA može proizvesti sledeće:
  - rečenice 1 i 2 – 100% tema A,
  - rečenice 3 i 4 – 100% tema B,
  - rečenica 5 – 60% tema A, 40% tema B.
  - tema A – 30% broccoli, 15% bananas, 10% breakfast, 10% munching, itd., tako da možemo da zaključimo da se tema A odnosi na hranu,
  - tema B – 20% chinchillas, 20% kittens, 20% cute, 15% hamster, itd., tako da možemo da zaključimo da se tema B odnosi na životinje.
- Pitanje je, kako ovo zapravo funkcioniše.

LDA

Jednostavno objašnjenje.

- LDA „posmatra“ dokument kao mešavinu tema na osnovu kojih su generisane reči sa različitim verovatnoćama.
- LDA je zasnovana na pretpostavci da je svaki dokument u kolekciji napisan na sledeći način – dok pišete dokument, Vi:
  - birate broj reči *N* koje će dokument imati;
  - birate mešavinu tema za dokument (shodno Dirihleovoj distribuciji na konačnom skupu od *K* elemenata) – na primer 1/3 se odnosi na hranu, 2/3 na životinje;
  - generišete svaku reč u dokumentu tako što
    - birate temu sa datom verovatnoćom (npr. hrana sa verovatnoćom 1/3), a zatim
    - koristite temu da generišete reč (npr. „broccoli“ sa verovatnoćom 30%, „bananas“ sa verovatnoćom 15%, itd.)
- Pretpostavljajući ovaj model generisanja dokumenata, LDA „kretanjem unazad“ pronalazi skup tema koje su najverovatnije generisale kolekciju dokumenata.

LDA

Jednostavno objašnjenje.

- Primer generisanja dokumenta *d*.
  - Dokument *d* će imati 5 reči.
  - Pola dokumenta će se odnositi na hranu, a pola na životinje.
  - Prvu reč biramo iz teme koja se odnosi na hranu – „broccoli“.
  - Drugu reč biramo iz teme koja se odnosi na životinje – „panda“.
  - Treću reč biramo iz teme koja se odnosi na životinje – „adorable“.
  - Četvrtu reč biramo iz teme koja se odnosi na hranu – „cherries“.
  - Petu reč biramo iz teme koja se odnosi na hranu – „eating“.
- Dokument generisan na osnovu LDA modela je „broccoli panda adorable cherries eating“.
- Kao što možete da primetite, LDA je tzv. *bag-of-words* model.

LDA

Jednostavno objašnjenje – obučavanje metodom *collapsed Gibbs sampling*.

- Pretpostavimo da imamo kolekciju dokumenata za koje želimo da otkrijemo *K* tema.
- Prolazimo kroz svaki dokument i na slučajni način dodeljujemo svaku reč jednoj od *K* tema.
- Slučajna dodela rezultuje „ne baš najsmislenijom“ dodelom tema dokumentima i distribucijom reči u odnosu na teme, pa se primenjuje sledeći algoritam (v. str. 11) kako bi se nakon određenog broja iteracija došlo do smislene dodele tema dokumentima.

LDA

Jednostavno objašnjenje – obučavanje metodom *collapsed Gibbs sampling*.

- Za svaki dokument *d*, svaku reč *w* u dokumentu *d*, i svaku temu *t* izračunati:
  - p(*t*|*d*) – verovatnoća pojavljivanja reči u dokumentu *d* koje su dodeljene temi *t*, i
  - p(*w*|*d*) – verovatnoća dodela svih dokumenata koji sadrže reč *w* temi *t*.
  - Dodeli reč *w* novoj temi, pri čemu se tema *t* bira sa verovatnoćom p(*t*|*d*) × p(*w*|*d*).
- Na osnovu generativnog modela zaključujemo da je ovo verovatnoća da je tema *t* generisala reč *w* tako da je smisleno da zamenimo trenutnu temu koja je generisala reč.
- Drugim rečima, u ovom koraku pretpostavljamo da su veze između tema i reči korektne za sve reči osim za onu koju trenutno obrađujemo, tako da trenutnu reč prilagođavamo.
- Nakon većeg broja iteracija, dolazi se do stabilnog stanja sa vrlo dobrim dodelama.

Izgradnja tematskog modela

Šta nam je potrebno?

- Potreban nam je paket gensim koji daje podršku za LDA. Paket možete da instalirate komandom:

```
pip install gensim
```

- Osim paketa gensim, potreban nam je i skup podataka. Korisićemo kolekciju Associated Press* vesti, koja se često koristi u istraživanju koje se odnosi na modeliranje tema.
- Skup se nakon preuzimanja uvozi na sledeći način:

```
>>> from gensim import corpora, models
>>> corpus = corpora.BleiCorpus('./data/ap/ap.dat', './data/ap/vocab.txt')
```

* Skup podataka se može preuzeti sa adrese: http://www.cs.columbia.edu/~blei/lda-c/ap.tgz

Izgradnja tematskog modela

Izgradnja modela.

- Promenljiva corpus sadrži sve tekstualne dokumente u formatu pogodnom za dalju obradu.
- Tematski model kreiramo koristeći ovaj objekat kao ulaz:

```
model = models.ldamodel.LdaModel(corpus, num_topics=100, id2word=corpus.id2word)
```

- Rezultujući model se može ispitivati na različite načine, npr:

```
>>> doc = corpus.docbyoffset(0)
>>> topics = model[doc]
>>> print(topics)
[(3, 0.023607255776894751),
(13, 0.11679936618551275),
....
(92, 0.10781541687001292)]
```

Izgradnja tematskog modela

Analiza modela.

- Rezultat prethodnog koda se u manjoj meri razlikuje sa svakim pokretanjem na istom skupu podataka, zato što se u fazi obuke modela koriste slučajni brojevi.
  - Ako teme koristimo za poređenje dokumenata, onda je algoritam dovoljno robusan a izmene male.
  - S druge strane, redosled različitih tema će biti sasvim drugačiji.
- Format rezultata je popis parova: (topic_index, topic_weight).
- U prethodnom primeru nema težine za teme 0, 1 i 2, odnosno težina tih tema je 0. Iako to nije 100% tačno, jer sve teme imaju ne-nula verovatnoću u LDA modelu, neke od njih imaju dovoljno malu verovatnoću da se zaokruživanjem na nulu postiže dobra aproksimacija.
- Tematski model je tzv. *sparse* model – iako postoji veliki broj mogućih tema, za svaki dokument se koristi samo nekoliko.

Izgradnja tematskog modela

Vizuelna analiza.

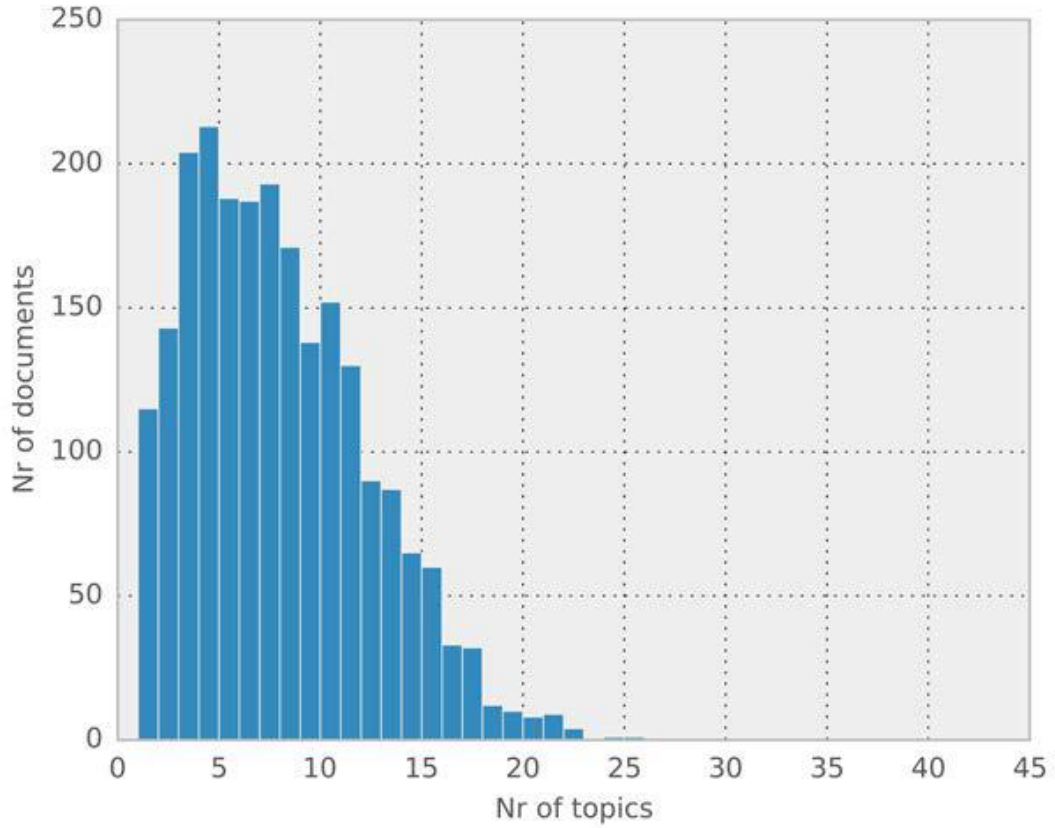
- Generisaćemo histogram sa koga se može videti koliko je tema dodeljeno dokumentima.

```
>>> num_topics_used = [len(model[doc]) for doc in corpus]
>>> plt.hist(num_topics_used)
```

- Na osnovu grafika na str. 16 se vidi da je najveći broj dokumenata povezan sa oko 5 tema, a da gotovo i ne postoje dokumenti koji su povezani sa više od 20 tema.

Izgradnja tematskog modela

Vizuelna analiza.



Izgradnja tematskog modela

Korekcija histograma.

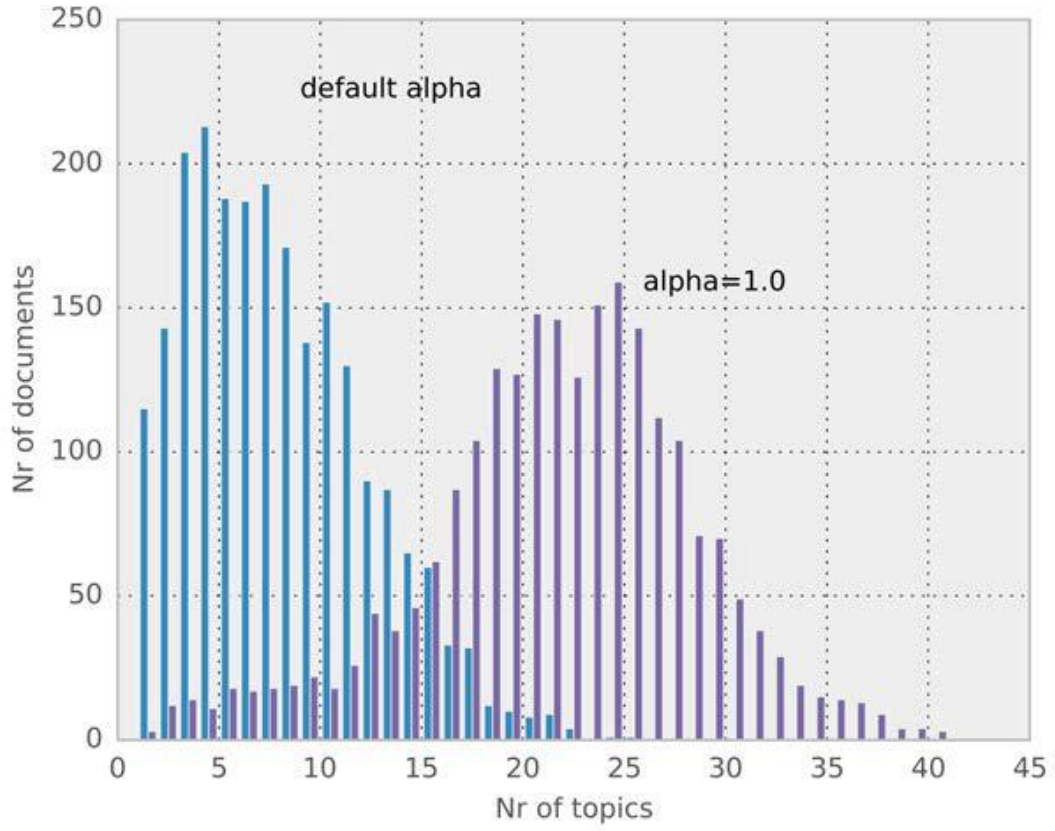
- Postavljamo sledeće pitanje: zašto histogram izgleda ovako, odnosno zašto povezujemo dokumente sa malim brojem tema?
- Jedan od glavnih uzroka je tzv. alfa parametar. Bez zalaženja u „matematiku“, veće vrednosti alfa parametra rezultovaće većim brojem tema koje su relevantne za dokument.
- Vrednost parametra mora da bude pozitivna, ali se obično postavlja na vrednost manju od jedan.
- Podrazumevano, gensim će postaviti vrednost parametra na 1/num_topics, ali se ona može „ručno“ proslediti LdaModel konstruktoru na sledeći način:

```
>>> model = models.ldamodel.LdaModel(corpus, num_topics=100, id2word=corpus.id2word, alpha=1)
```

- Na novom histogramu na str. 18 možemo videti da su mnogi dokumenti povezani sa 20 do 25 različitih tema. Ako umanjite vrednost, dogodiće se suprotno.

Izgradnja tematskog modela

Korekcija histograma.



Izgradnja tematskog modela

Šta su to teme?

- Tehnički, to su multinomijalne raspodele, što znači da dodeljuju verovatnoću svakoj reči u rečniku.
- Reči sa velikom verovatnoćom su više povezane sa tom temom nego reči sa manjom verovatnoćom.
- Izdvojićemo reči sa najvećim verovarnoćama za nekoliko tema.

Izgradnja tematskog modela

Šta su to teme?

Tema br.	Tema
1	dress military soviet president new state capt carlucci states leader stance government
2	koch zambia lusaka oneparty orange kochs party i government mayor new political
3	human turkey rights abuses royal thompson threats new state wrote garden president
4	bill employees experiments levin taxation federal measure legislation senate president whistleblowers sponsor
5	ohio july drought jesus disaster percent hartford mississippi crops northern valley virginia
6	united percent billion year president world years states people i bush news
7	b hughes affidavit states united ounces squarefoot care delaying charged unrealistic bush
8	yeutter dukakis bush convention farm subsidies uruguay percent secretary general i told
9	kashmir government people srinagar india dumps city two jammukashmir group moslem Pakistan
10	workers vietnamese irish wage immigrants percent bargaining last island police hutton I

Izgradnja tematskog modela

Šta su to teme?

- Iako na prvi pogled deluje čudno, kada pročitamo spisak reči, jasno možemo videti da teme nisu samo popis slučajnih reči, već da su u pitanju logične grupe.
- Takođe vidimo da se ove teme odnose na starije vesti, npr. koje datiraju iz vremena je Sovjetski Savez i dalje postojao, a Gorbačov bio generalni sekretar.
- Takođe možemo predstaviti teme kao tzv. „oblake reči“ (engl. *word cloud*), u kojima su verovatnije reči ispisane kao veće.
- Kao primer dat je oblak reči koji se odnosi na bliski istok i politiku.
- Kompletan kod za iscrtavanje oblaka reči dat je u dodatku knjige, datoteka `/ch04/wordcloud.py`, delovi su dati u nastavku beleške.

Izgradnja tematskog modela

Oblak reči.

Izgradnja tematskog modela

Generisanje slike oblaka reči.

- Neophodno je instalirati pytagcloud.

```
def create_cloud(oname, words, maxsize=120, fontname='Lobster'):
  # oname - izlazna datoteka, words - lista reči (value, str)
  # maxsize - veličina najveće reči (int, opciono), fontname - ime fonta (str, opciono)
  from pytagcloud import create_tag_image, make_tags
  # gensim vraća težinu između 0 and 1 za svaku reč, a pytagcloud očekuje ceo broj.
  # Množimo sve velikim brojem i zaokružujemo što je dobra aproksimacija za ovu vizuelizaciju.
  words = [(w,int(v*10000)) for w,v in words]
  tags = make_tags(words, maxsize=maxsize)
  create_tag_image(tags, oname, size=(1800, 1200), fontname=fontname)
```

Izgradnja tematskog modela

Da li je potrebno dodatno predprocesiranje?

- Iz oblaka reči se vidi da bi neke reči trebale biti uklonjene (na primer, reč „I“).
- Stop reči ne sadrže toliko informacija i ne treba im pridavati isti značaj kao rečima koje se ne pojavljuju često u različitim kontekstima.
- Dakle, prilikom modeliranja tema potrebno je filtrirati stop reči, jer se u suprotnom može formirati tema koja se sastoji isključivo od stop reči.
- Osim toga, potrebno je izvršiti normalizovanje reči, odnosno svesti reči na stem oblik (v. belešku 03), npr. normalizacija različitih oblika glagola.

Poređenje dokumenata pomoću tema

Poređenje dokumenata u „prostoru tema“.

- Nakon modeliranja tema, moguće je za svaki document proceniti koliki deo dokumenta „potiče“ iz jedne teme, koliki deo iz druge, itd.
- To znači da možemo porediti dokumente u „prostoru tema“ (engl. *topic space*).
- Jednostavnije rečeno, umesto poređenja reči, mi kažemo da su dva dokumenta slična ako se odnose na iste teme.
- Ovo je vrlo upotrebljivo u slučajevima u kojima dva dokumenta koja sadrže nekoliko zajedničkih reči upućuju na istu temu koristeći različite konstrukcije (na primer, jedan dokument sadrži „predsednik Sjedinjenih Država“, dok drugi sadrži „Donald Trump“).
- Drugim rečima, više ne poredimo dokumente na osnovu vektora reči, već na osnovu vektora tema (engl. *topic vector*).

Poređenje dokumenata pomoću tema

Poređenje dokumenata u „prostoru tema“.

- Dokumente projektujemo u prostor tema – to znači da želimo da imamo vektor tema koje sumiraju dokument.
- Kada su teme izračunate za svaki dokument, možemo izvršiti operacije na vektoru tema i zaboraviti na originalne reči.
- Ako su teme bitne, one će biti potencijalno informativnije od sirovih reči.
- Pored toga, ovo može dovesti do smanjenja ukupnog izračunavanja prilikom poređenja vektora, pošto je mnogo brže upoređivanje 100 vektora koji sadrži težine tema, nego vektora koji će sadržati hiljade termina.

Poređenje dokumenata pomoću tema

Poređenje dokumenata.

```
>>> from gensim import matutils
>>> topics = matutils.corpus2dense(model[corpus], num_terms=model.num_topics)
>>> from scipy.spatial import distance
>>> pairwise = distance.squareform(distance.pdist(topics))
>>> largest = pairwise.max()
>>> for ti in range(len(topics)): pairwise[ti,ti] = largest+1
>>> def closest_to(doc_id): return pairwise[doc_id].argmin()
```

- Promenljiva `topics` je matrica tema.
- Međusobna rastojanja računamo koristeći SciPy `pdist` funkciju.
- Jednim pozivom funkcije računamo sve vrednosti: `sum ((topics[ti] - topics[tj])**2)`.
- Dijagonalne elemente matrice postavljamo na vrednost veću od bilo koje vrednosti u matrici. Da li možete da predpostavite zašto je to neophodno?
- Funkcija `closest_to` funkcioniše slično metodi najbližih suseda.

Poređenje dokumenata pomoću tema

Primer.

- Upitni dokument (redni broj 1):

From: geb@cs.pitt.edu (Gordon Banks)
Subject: Re: request for information on "essential tremor" and Indrol?
In article <1q1tbnINNnfn@life.ai.mit.edu> sundar@ai.mit.edu writes:
Essential tremor is a progressive hereditary tremor that gets worse when the patient tries to use the effected member. All limbs, vocal cords, and head can be involved. Inderal is a beta-blocker and is usually effective in diminishing the tremor. Alcohol and mysoline are also effective, but alcohol is too toxic to use as a treatment.

Gordon Banks N3JXP, geb@cadre.dsl.pitt.edu | "Skepticism is the chastity of the intellect, and it is shameful to surrender it too soon."

Poređenje dokumenata pomoću tema

Primer.

- Ukoliko zatražimo najsličniji dokument prethodnom, `closest_to(1)` će vratiti post istog autora u kome su takođe razmatrani medikamenti.

From: geb@cs.pitt.edu (Gordon Banks)
Subject: Re: High Prolactin
In article <93088.112203JER4@psuvm.psu.edu> JER4@psuvm.psu.edu (John E. Rodway) writes:
> Any comments on the use of the drug Parlodel for high prolactin in the blood?
>
It can suppress secretion of prolactin. Is useful in cases of galactorrhea. Some adenomas of the pituitary secret too much.

Gordon Banks N3JXP, geb@cadre.dsl.pitt.edu | "Skepticism is the chastity of the intellect, and it is shameful to surrender it too soon."

Modeliranje Vikipedije

Postavka problema.

- Prvobitne implementacije LDA su spore (originalni algoritam je objavljen u naučnom radu 2003. godine), što ograničava njihovu upotrebu na male kolekcije dokumenata.
- Međutim, savremeni algoritmi dobro rade sa veoma velikom kolekcijom podataka (metod koji ćemo koristiti nadalje za obradu Vikipedije je nastao 2010. godine).
- Prateći dokumentaciju paketa gensim, napravićemo tematski model za jedan deo Vikipedije na engleskom jeziku.
- Uzmite u obzir da je ovo izvodljivo na vašem laptopu, ali da je vremenski zahtevno.

Modeliranje Vikipedije

Rešenje problema.

- Najpre preuzimamo Wikipedia dump datoteku (uzmite u obzir da se radi o datoteci veličine nekoliko GB): https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2.
- Zatim je indeksiramo koristeći gensim alat, pri čemu se poziva iz komandne linije a ne iz Python shell-a:

```
python -m gensim.scripts.make_wiki enwiki-latest-pages-articles.xml.bz2 wiki_en_output
```

- Nakon nekog vremena, indeks će biti generisan u istom direktorijumu.

Rešenje problema.

- Nakon toga uvozimo gensim paket i paket za evidentiranje događaja:

```
>>> import logging, gensim
```

- Događaje evidentiramo koristeći standardni Python logging modul (koji genism koristi da prikazuje statusne poruke).
- Ovaj korak nije obavezan, ali je u slučajevima da obrada podataka dugo traje poželjno da dobijemo poneko obaveštenje o tome šta se dešava.

```
>>> logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)
```

Modeliranje Vikipedije

Rešenje problema.

- Predprocesirane podatke učitavamo na sledeći način:

```
>>> id2word = gensim.corpora.Dictionary.load_from_text('wiki_en_output_wordids.txt')
>>> mm = gensim.corpora.MmCorpus('wiki_en_output_tfidf.mm')
```

- LDA model kreiramo kao i ranije.
- Uzmite u obzir da i ova operacija traje dugo. Na konzoli ćete videti koji je deo posla algoritam odradio, na osnovu čega možete da zaključite koliko ćete još čekati.

```
>>> model = gensim.models.ldamodel.LdaModel(corpus=mm, id2word=id2word, num_topics=100,
            update_every=1, chunksize=10000, passes=1)
```

Modeliranje Vikipedije

Kako možemo da sačuvamo model?

- Nakon generisanja modela, isti možete snimiti metodom `save()` kako ne bi ponavljali proces.

```
>>> model.save('wiki_lda.pkl')
```

- Model se može u novoj sesiji učitati na sledeći način:

```
>>> model = gensim.models.ldamodel.LdaModel.load('wiki_lda.pkl')
```

Modeliranje Vikipedije

Kako izgleda model?

- Iako smo imali značajno veći broj dokumenata nego u prethodnom slučaju (oko 4 miliona 2015. godine), model je još uvek tipa *sparse*.

```
>>> lens = (topics > 0).sum(axis=0)
>>> print(np.mean(lens))
6.41
>>> print(np.mean(lens <= 10))
0.941
```

- Ovo znači da je prosečni dokument povezan sa 6,4 tema i da je 94% dokumenata povezano sa 10 ili manje tema.

Modeliranje Vikipedije

Koja je tema najprisutnija na Vikipediji?

- Postavićemo sledeće pitanje: o čemu se najviše priča na Vikipediji?
- Najpre ćemo izračunati ukupnu težinu za svaku temu (sabiranjem težina svih dokumenata) a zatim izdvojiti reči relevantne za tu temu.
- To možemo da uradimo sledećim kodom:

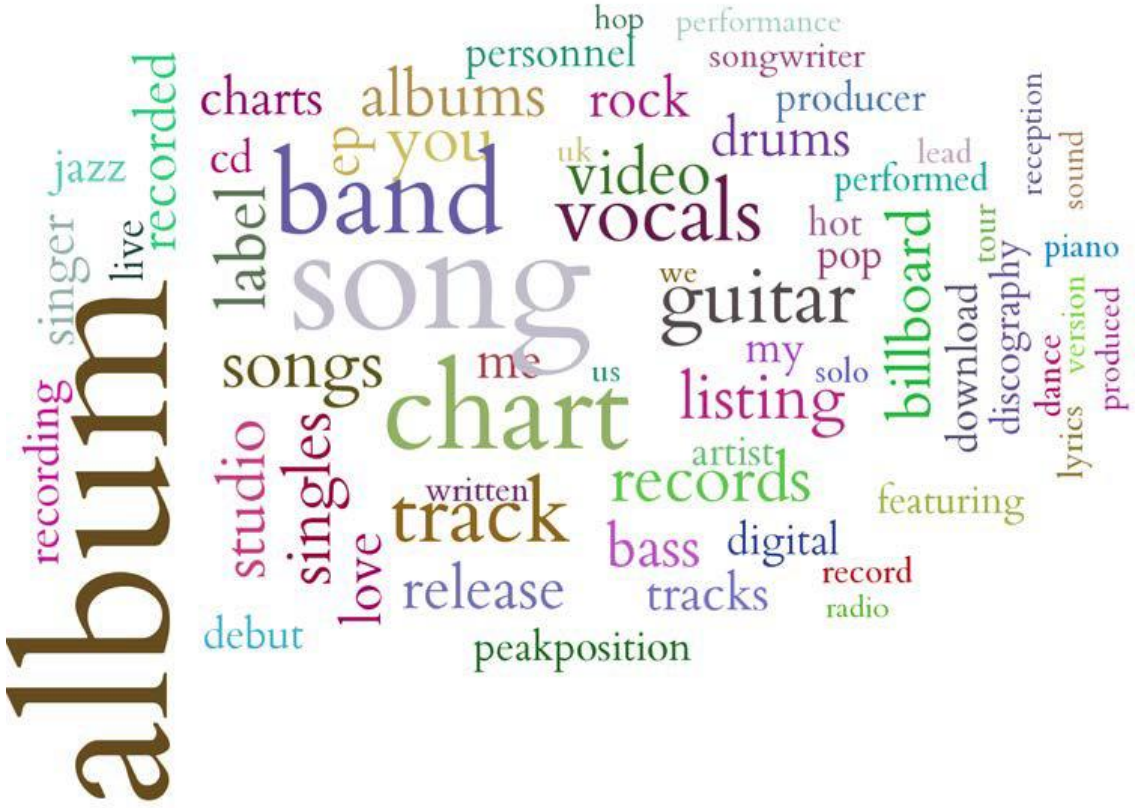
```
>>> weights = topics.sum(axis=0)
>>> words = model.show_topic(weights.argmax(), 64)
```

- Nakon vizuelizacije (v. str. 36) vidimo da je muzika najviše zastupljena tema – oko 18% stranica Vikipedije se odnose na tu temu, a oko 5,5% svih reči u Vikipediji su dodeljeni toj temi.

* Oblak reči za Vikipediju je iscrtan na osnovu podataka iz 2015. godine.
S obzirom da se Vikipedija stalno menja, najverovatnije ćete dobiti različit oblak reči.

Modeliranje Vikipedije

Koja je tema najprisutnija na Vikipediji?



Modeliranje Vikipedije

Koja je tema najmanje prisutna na Vikipediji?

- Ukoliko `argmax()` zamenimo sa `argmin()` dobićemo oblak reči za najmanje prisutnu temu.

```
>>> words = model.show_topic(weights.argmin(), 64)
```

- Temu o kojoj se najmanje priča je teže interpretirati.
- Neke od najznačajnijih reči ukazuju na aerodrome u istočnim zemljama.
- Oko 1,6% dokumenata se odnosi na tu temu, a manje od 0,1% svih reči u Vikipediji su dodeljeni toj temi.

Koja je tema najmanje prisutna na Vikipediji?

Završne napomene

Da li broj tema značajnije utiče na konačan ishod?

- Do sada smo koristili fiksni broj tema za naše analize – 100.
- To je bio čisto proizvoljan broj – mogli smo da koristimo i 20 ili 200 tema.
- Za mnoge namene ovaj broj nije isuviše važan.
- Ako teme koristimo samo kao srednji korak (kao pri traženju sličnih postova), konačno ponašanje sistema retko je značajnije osetljivo na tačan broj tema korišćenih u modelu.
- To znači da sve dok koristite dovoljno tema, bez obzira da li koristite 100 ili 200, preporuke koje proizilaze iz procesa neće biti mnogo drugačije.
- 100 tema je često dovoljno dobar izbor, dok je 20 premalo za opštu kolekciju tekstualnih dokumenata.
- Isto važi i za podešavanje vrednosti alfa parametra – tokom eksperimentisanja sa njim može se promeniti tema, ali su konačni rezultati robusni protiv ove promene.

Završne napomene

Automatsko određivanje broja tema.

- Postoji nekoliko metoda koji će automatski odrediti broj tema zavisno od skupa podataka.
- Jedan od modela se hijerarhijski Dirichlet-ov proces.
- Za razliku od LDA, gde se broj tema fiksira, u ovom slučaju se teme generišu zajedno sa podacima – jedna po jedna.
- U opštem slučaju to znači sledeće: što više dokumenata, to više tema.

Literatura

- Beleške pripremljene prema knjizi – Luis Pedro Coelho, Willi Richert (2015): „Building Machine Learning Systems with Python, Second Edition“. Packt Publishing.

Hvala na pažnji

Pitanja su dobrodošla.