Fazi klasterovanje

Uvod

U ovom radu je predstavljena upotreba fazi klasterovanja za kategorizaciju dokumenata.

Na početku raspolažemo skupom dokumenata, koji su iz različitih oblasti (kategorija), npr možemo imati dokumente iz ekonomije, politike, medicine, računarskih nauka itd.

Algoritam će date dokumente klasterovati tj svrstati u različite kategorije.

Opis rešenja

Dokument možemo predstaviti nizom uređenih parova. Uređeni par se sastoji od reči i broja koji predstavlja broj pojavljivanja date reči u dokumentu.

Bez gubitka na opštosti pretpostavimo da su svi dokumenti predstavljeni na ovaj način. Ako ipak nisu predstavljeni na ovaj način, to lako možemo učiniti.

Pretprocesiranje

Formirajmo skup A koji se sastoji od svih reči iz dokumenata koje klasterujemo.

Naglasimo da je u našem skupu A bitan poredak reči, i pretpostavimo da su reči u njemu leksikografski uređene.

Nadalje, svakom dokumentu pridružujemo jedan vektor. Dimenzija tog vektora je kardinalnost skupa A. I-ta komponenta vektora predstavlja broj pojavljivanja I-te reči iz skupa u tom dokumentu.

Radi kasnijeg efikasnijeg klasterovanja, izvršićemo transformaciju vektora. Svaku komponentu vektora podelimo sa ukupnom sumom svih komponenti u vektoru.

Na taj način dobijamo relativnu učestalost reči u dokumentu.

Na osnovu navedenog zaključujemo da dobijamo matricu (označimo je sa **X**) čije su vrste dokumenti, a kolone reči iz skupa A.

Algoritam klasterovanja

Predstavljeni algoritam je modifikacija algoritma K-sredina (K means). Umesto standardnog predstavljanja pripadnosti klasteru (pripada/ne pripada), koristićemo fazi logiku. Ovakav algoritam se naziva C-means algoritam.

Ideja je da imamo stepen pripadnosti, što znači da dokument može pripadati klasteru za stepenu pripadnosti iz intervala [0,1].

Primetimo da za određeni dokument suma svih stepena pripadnosti po klasterima mora biti 1.

U implementaciji smo se opredelili za random inicijalizaciju centroida klastera. Postoje različiti pristupi za inicijalizaciju centroida (neki su dosta složeni), ovde smo se opredelili za ovaj, zbog dosta dobrog ponašanja i jednostavne implementacije.

Za implementaciju ovog algoritma, potrebno je definisati rastojanje izmedju dva dokumenta. Rastojanje koje ćemo koristiti je kosinusno rastojanje. Upotreba kosinusnog rastojanja za dokumenta je veoma česta praksa.

Pseudokod algoritma

1. Inicijalizuj fazi matricu. Matricu ćemo označiti sa **U.** Ovu matricu inicijalizujemo nasumično.
2. U k-tom koraku, određujemo $C^{(k)}$ iz $U^{(k)}$ po formuli

$$C_{ik} = \begin{cases} 0 \\ \frac{\sum_{j=1}^{N} (\mu_{ij})^m x_{jk}}{\sum_{j=1}^{N} (\mu_{ij})^m} \end{cases}$$

3. Računamo $U^{(k+1)}$ iz $U^{(k)}$ pomoću formula

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{d_{ij}^2}{d_{kj}^2} \right)^{\frac{1}{m-1}}} \qquad d_{ij}^2 = \sum_{k=1}^{n} v_{ik} (x_{jk} - C_{ik})^2$$

Zbog jednostavnosti, a bez gubitka na kvalitetu rada pretpostavicemo da je V_{ik} uvek 1, za svako **i** i **k**.

4. Korake ponavljamo odredjen fiksiran broj iteracija, u implementaciji taj broj je 200.

C je matrica koordinata klastera.

Detaljnije objasnjenje oznaka možete videti u samom radu.

Kao ulaz algoritma zadajemo i broj klastera, to zahteva C-means algoritam kao što je to slučaj i kod K-means algoritma.
Takodje ulaz u algoritam je i parametar m, u implementaciji smo uzeli da je m = 2.

Bez ulaženja u detalje, napomenimo da postoje načini da se odredi najbolji broj klastera, naprimer korišćenjem silueta koeficijenata.

Okruženje za pokretanje

Algoritam je testiran na operativnom sistemu Ubuntu 20.04. Korišćen je Python 3.8.5

Pokretanje algoritma i rezultati

Kako je naš algoritam stohastički, stoga znamo da se za iste podatke može dobiti različit izlaz. Stohastičnost dolazi od nasumične inicijalizacije matrice pripadnosti.

Stoga, mi ćemo pokazati izlaz jednog pokretanja.

Naši dokumenti (ulazni podaci), dolaze iz 3 različite kategorije, ima ih 15, iz svake kategorije dolazi po 5 dokumenata.

Izlazni rezultat je matrica pripadnosti. Vrste su dokumenti, a kolone su klasteri. Element A_{ij} matrice predstavlja stepen pripadnosti dokumenta **i** klasteru **j.**

Dobijen rezultat je:

[[**0.62357613** 0.3494202 0.02700367]
 [0.13991092 **0.58153614** 0.27855295]
 [0.05231272 **0.68513415** 0.26255313]
 [0.04230943 **0.89535786** 0.06233272]
 [**0.54676446** 0.24068098 0.21255456]
 [**0.75997458** 0.04784774 0.19217769]
 [**0.57384213** 0.10412857 0.32202929]
 [0.112692   **0.83765767** 0.04965033]
 [0.03245756 0.05706147 **0.91048098**]
 [0.15922394 **0.71732977** 0.12344629]
 [0.07678158 0.03921469 **0.88400373**]
 [0.10950181 0.18845397 **0.70204422**]
 [0.11188957 **0.7625027** 0.12560773]
 [0.21033571 **0.64718698** 0.1424773]]

Možemo se uveriti da je suma vrednosti u svakoj vrsti tačno 1.

U svakoj vrsti podebljana je najveca vrednost, odnosno najveci stepen pripadnosti.

Raspodela dokumenata po klasterima je 4,7,3, ako odlučimo da najveći stepen pripadnosti posmatramo kao da taj dokument pripada tom klasteru.

Vreme izvšavanja algoritma na navedenoj platformi za ovu implementacijue je oko 0.7 sekundi (200 iteracija).

Pravci daljeg razvoja

Potencijalno, radi unapredjenja klasterovanja možemo promeniti funkciju rastojanja, umesto kosinusnog možemo uzeti neko drugo.

Takođe možemo probati sa nekom drugom vrednošću parametra m.

Možemo koristiti i neku drugu strategiju za inicijalizaciju vrednosti matrice U.

Literatura

Implementacija je rađena na osnovu smernica iz rada:

https://www.researchgate.net/publication/262002491_Fuzzy_clustering_and_categorization_of_text_documents