Uvod

Klasterovanje je grupisanje objekata na takav način da su objekti koji pripadaju istoj grupi (klasteru) sličniji jedni drugima nego onim objektima iz drugih grupa. Klasterovanje ima široke primene i koristi se u mašinskom učenju, prepoznavanju obrazaca, analizi slika, računarskoj grafici, kompresiji podataka, bioinformatici itd. U ovom radu biće izložene neke metode klasterovanja, uključujući Lojdov algoritam. Glavni deo rada odnosi se na primenu algoritma maksimizacije očekivanja (eng. *Expectation Maximization algorithm*) na Lojdov algoritam klasterovanja čiji će način rada biti detaljno opisan u nastavku. Sam algoritam maksimizacije očekivanja je vrlo popularna tehnika mašinskog učenja i ima primene u različitim oblastima. Mi ćemo ovde prikazati neke konkretne primene u bioinformatici koje se odnose na klasterovanje genskih ekspresija.

Klasterovanje k-sredina

Dat je skup *Data* sa *n* tačaka i skup *Centers* sa *k* centara. Kod klasterovanja k-sredina, potrebno je odrediti *k* centara tako da se minimizuje deformisanost kvadratne greške. Deformisanost (distorzija) kvadratne greške se definiše kao rastojanje svake tačke od njenog najbližeg centra:

$$\text{DISTORTION}(Data, Centers) = \frac{1}{n} \sum_{\text{all points } DataPoint \text{ in } Data} d(DataPoint, Centers)^2$$

Jedan od bitnih pojmova koji ovde treba da definišemo, a koji će nam biti bitan u nastavku, je centar gravitacije skupa tačaka. Centar gravitacije je tačka čija je i-ta koordinata prosečna vrednost i-tih koordinata svih tačaka skupa. Određivanje centra gravitacije skupa tačaka predstavlja specijalan slučaj rešavanja problema klasterovanja k-sredina za *k =1*.

Lojdov algoritam

Lojdov algoritam (eng. *Lloyd's algorithm*) predstavlja najčešće korišćenu heuristiku klasterovanja k-sredina. Najpre se proizvoljno bira *k* tačaka iz skupa podataka za centre, a zatim se iterativno ponavljaju dva koraka:

Korak od centara ka novim klasterima: kada su centri odabrani, pridružiti svaku tačku klasteru koji je određen najbližim centrom.
Korak od klastera ka novim centrima: kada su tačke dodeljene, izabrati centar gravitacije kao novi centar klastera.

Lojdov algoritam konvergira kada ne dolazi do promene centara između dve iteracije. Kako bi se algoritam završio, deformisanost kvadratne greške se mora smanjivati u svakom koraku. U koraku od centara ka novim klasterima, svakoj tački je pridružen novi centar, pa ta tačka mora biti bliža novom nego prethodno izabranom centru.

U koraku od klastera ka novim centrima, ako je centar klastera ažuriran kao centar gravitacije, tada je novi centar jedina tačka koja minimizuje deformisanost kvadratne greške za tačke koje tom klasteru pripadaju.

Jedno od ograničenja Lojdovog algoritma je jednoznačno pridruživanje svake tačke tačno jednom klasteru. U slučaju da se tačka nalazi na približno jednakom rastojanju od dva centra, ideja je da se ona nalazi u dva klastera. Navedeno svojstvo predstavlja motivaciju za tzv. *meko klasterovanje*.

Motivacija za algoritam maksimizacije očekivanja

Pristrasni novčić se baca *n* puta, potrebno je odrediti verovatnoću θ da će pojedinačno bacanje dovesti do događaja da je dobijena glava. Najbolja procena parametra θ predstavlja odnos broja ishoda kada je dobijena glava i ukupnog broja bacanja novčića. Na osnovu date sekvence od *n* bacanja koja sadrži *i* dobijenih glava, verovatnoća da će pristrasni novčić sa pristrašnošću θ generisati sekvencu je $f(\theta) = \theta^{i}(1- \theta)^{n-i}$.

Definišimo složeniji problem, na raspolaganju su nam dva pristrasna novčića *A* i *B*, sa pristrasnostima dobijanja glave θ_A i θ_B. Nakon posmatranja niza sekvenci, cilj je proceniti koji novčići su korišćeni. Nepoznate veličine jednim imenom nazivamo parametrima.

										Data
H	T	T	T	H	T	T	H	T	H	0.4
H	H	H	H	T	H	H	H	H	H	0.9
H	T	H	H	H	H	H	T	H	H	0.8
H	T	T	T	T	T	H	H	T	T	0.3
T	H	H	H	T	H	H	H	T	H	0.7

Ako je poznato da je u prvoj i četvrtoj sekvenci bacanja korišćen novčić *A*, pristrasnost θ_A računamo na sledeći način:

$$\theta_A = \frac{Data_1 + Data_4}{2} = \frac{0.4 + 0.3}{2} = 0.35.$$

Slično, dobijamo pristrasnost θ_B za drugi novčić. Izbor novčića po sekvencama predstavićemo kao binarni vektor *HiddenVector = (1,0,0,1,0)*, gde je na k-toj poziciji 1 ukoliko je za datu sekvencu korišćen novčić *A*, a 0 ukoliko je korišćen novčić *B*. Ova notacija nam omogućava da jednačine parametara napišemo u terminima podataka i skrivenog vektora.

$$\theta_A = \frac{\sum_i HiddenVector_i \cdot Data_i}{\sum_i HiddenVector_i} = \frac{1 \cdot 0.4 + 0 \cdot 0.9 + 0 \cdot 0.8 + 1 \cdot 0.3 + 0 \cdot 0.7}{1 + 0 + 0 + 1 + 0} = 0.35$$

$$\theta_B = \frac{\sum_i (1 - HiddenVector_i) \cdot Data_i}{\sum_i (1 - HiddenVector_i)} = \frac{0 \cdot 0.4 + 1 \cdot 0.9 + 1 \cdot 0.8 + 0 \cdot 0.3 + 1 \cdot 0.7}{0 + 1 + 1 + 0 + 1} = 0.80$$

Definišemo i vektor jedinica iste dužine kao i *HiddenVector (*nadalje u tekstu kao *skriveni vektor)*. U opštem slučaju parametre računamo na sledeći način:

$$\theta_A = \frac{HiddenVector \cdot Data}{HiddenVector \cdot \vec{1}}$$

$$\theta_B = \frac{(\vec{1} - HiddenVector) \cdot Data}{(\vec{1} - HiddenVector) \cdot \vec{1}}$$

Ako znamo parametre, možemo da odredimo koji je novčić korišćen za generisanje nepoznate sekvence. Neka su vrednosti parametra *Parameters* = (θ_A, θ_B) = *(0.6, 0,82)*. Ako je za generisanje sekvence korišćen novčić *A*, verovatnoća generisanja nove sekvence je:

$$\theta_A^7(1-\theta_A)^3 = 0.6^7 \cdot 0.4^3 \approx 0.00179 .$$

Ako je novčić *B* korišćen, verovatnoća generisanja nove sekvence je:

$$\theta_B^7(1-\theta_B)^3 = 0.82^7 \cdot 0.18^3 \approx 0.00145 .$$

Kako je 0.00179 > 0.00145 postavljamo petu koordinatu skrivenog vektora na 1.
Ako je poznat skriveni vektor, određivanje parametara se vrši na prethodno opisani način. Na sličan način se na osnovu parametara može rekonstruisati skriveni vektor. U realnim problemima, najčešće je potrebno odrediti i skriveni vektor i parametre.

Uslovna verovatnoća generisanja jednog podatka (pisma ili glave), na osnovu skrivenog vektora i parametara je:

$$\Pr(Data_i|HiddenVector, Parameters) = \begin{cases} \Pr(Data_i|\theta_A) & \text{if } HiddenVector_i = 1 \\ \Pr(Data_i|\theta_B) & \text{if } HiddenVector_i = 0 \end{cases}$$

Nadalje definišemo uslovnu verovatnoću generisanja sekvence na osnovu skrivenog vektora i parametara:

$$\Pr(Data|HiddenVector, Parameters) = \prod_{i=1}^{n} \Pr(Data_i|HiddenVector, Parameters)$$

Na osnovu podataka, problem koji rešavamo je određivanje skrivenog vektora i parametara maksimizovanjem *Pr(Data|HiddenVector, Parameters)*.

Bacanje novčića i Lojdov algoritam

Dat je skup od *n* tačaka, $Data = (Data_1, ..., Data_n)$, pri čemu je svaka tačka dimenzije *m*. Tačka pripada nekom od *k* klastera i tu pripadnost definišemo n-dimenzionalnim vektorom: $HiddenVector = (HiddenVector_1, ..., HiddenVector_n)$, pri čemu svaka koordinata vektora $HiddenVector_i$ uzima celobrojnu vrednost od 1 do k. Predstavljamo centre kao *k* tačaka m-dimenzionalnog prostora, $Parameters = (\theta_1, \theta_2, ..., \theta_k)$.

Kod klasterovanja k-sredina, slično kao u slučaju bacanja novčića skriveni vektor i parametri su nepoznati. Lojdov algoritam započinje slučajnim izborom parametara. Dva osnova koraka se mogu zapisati na sledeći način:

Korak **od centara ka novim klasterima**: *(Data, ?, Parameters) -> HiddenVector*.
Korak **od klastera ka novim centrima**: *(Data, HiddenVector, ?) -> Parameters*.

Jedina razlika između algoritma bacanja novčića i Lojdovog algoritma je u načinu izvršavanja koraka od centara ka novim klasterima. Kod bacanja novčića, računamo vrednost $HiddenVector_i$ poređenjem verovatnoća $Pr(Data_i \mid \theta_A)$ i $Pr(Data_i \mid \theta_B)$. Kod klasterovanja, tačku pridružujemo klasteru određenu najbližim centrom.

Algoritam maksimizacije očekivanja u slučaju bacanja novčića

U algoritmu se iterativno ponavljaju dva koraka:

1) **korak očekivanja** (E-step)
2) **korak maksimizacije** (M-step)

Korak očekivanja:

Na osnovu parametara (θ_A, θ_B) možemo jednoznačno odrediti vrednosti skrivenog vektora poređenjem verovatnoća $Pr(Data_i \mid \theta_A)$ i $Pr(Data_i \mid \theta_B)$. Ukoliko su ove verovatnoće približno jednake, nismo sigurni koji novčić je korišćen. Iz tog

razloga definišemo vektor odgovornosti da je novčić generisao datu sekvencu. Ove odgovornosti se sumiraju na 1.

Kod klasterovanja k-sredina ukoliko se tačka nalazi na podjednakom rastojanju od dva centra, tada oni imaju istu odgovornost za pripadanje tačke tim klasterima. Isto kao u slučaju novčića, ove odgovornosti se sumiraju na 1.

Korak maksimizacije:

Pri jednoznačnom određivanju pripadnosti, parametre računamo iz podataka i skrivenog vektora na ranije opisan način. U slučaju dva novčića, odgovornosti možemo predstaviti matricom. Pojavljivanje 1 na i-toj poziciji u prvom redu označava da je novčić *A* generisao i-tu sekvencu i slično važi za novčić *B*.

$$HiddenMatrix \begin{matrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{matrix}$$

Dakle, prvi red skrivene matrice u oznaci $HiddenMatrix_A$ je sam skriveni vektor. Drugi red skrivene matrice u oznaci $HiddenMatrix_B$ predstavlja vrednost 1 – *HiddenVector*. Parametre sada izražavamo u terminima skrivene matrice:

$$\theta_A = \frac{HiddenMatrix_A \cdot Data}{HiddenMatrix_A \cdot \overrightarrow{1}}$$

$$\theta_B = \frac{HiddenMatrix_B \cdot Data}{HiddenMatrix_B \cdot \overrightarrow{1}}$$

Primena algoritma maksimizacije očekivanja na meko klasterovanje k-sredina

Algoritam započinje slučajnim izborom centara i ponavlja sledeća dva koraka:

1) Korak od centara ka novim klasterima (**E - step**): kada su centri odabrani, pridružiti tački odgovornosti za pripadanje svakom klasteru, pri čemu veća odgovornost označava jaču pripadnost klasteru.
2) Korak od klastera ka novim centrima (**M - step)**: kada su tačke dodeljene klasterima, vrši se ažuriranje centara.

Korak od centara ka novim klasterima:

Na osnovu *k* centara *Centers* = $(x_1, \ldots, x_k)$ i *n* tačaka podataka *Data* = $(Data_1, \ldots, Data_n)$, konstruišemo *k x n* matricu pri čemu vrednost $HiddenMatrix_{i,j}$ predstavlja odgovornost da tačka *j* pripada centru *i*. Ova odgovornost se može izračunati u skladu sa Njutnovim zakonom gravitacije:

$$HiddenMatrix_{i,j} = \frac{1/d(Data_j, x_i)^2}{\sum_{\text{all centers } x_i} 1/d(Data_j, x_i)^2}$$

U praksi se particiona funkcija pokazala kao bolje rešenje (u implementaciji ćemo koristiti ovu funkciju):

$$HiddenMatrix_{i,j} = \frac{e^{-\beta \cdot d(Data_j, x_i)}}{\sum_{\text{all centers } x_i} e^{-\beta \cdot d(Data_j, x_i)}}$$

gde β predstavlja parametar mekoće.

Korak od klastera ka novim centrima:

Ukoliko sa $HiddenMatrix_i$ označimo i-ti red skrivene matrice, tada vršimo ažuriranje centra x_i . Za centar x_i njegovu j-tu koordinatu u oznaci $x_{i,j}$ računamo kao:

$$x_{i,j} = \frac{HiddenMatrix_i \cdot Data^j}{HiddenMatrix_i \cdot \overrightarrow{1}}$$

Ažurirani centar x_i se naziva težinski centar gravitacije.

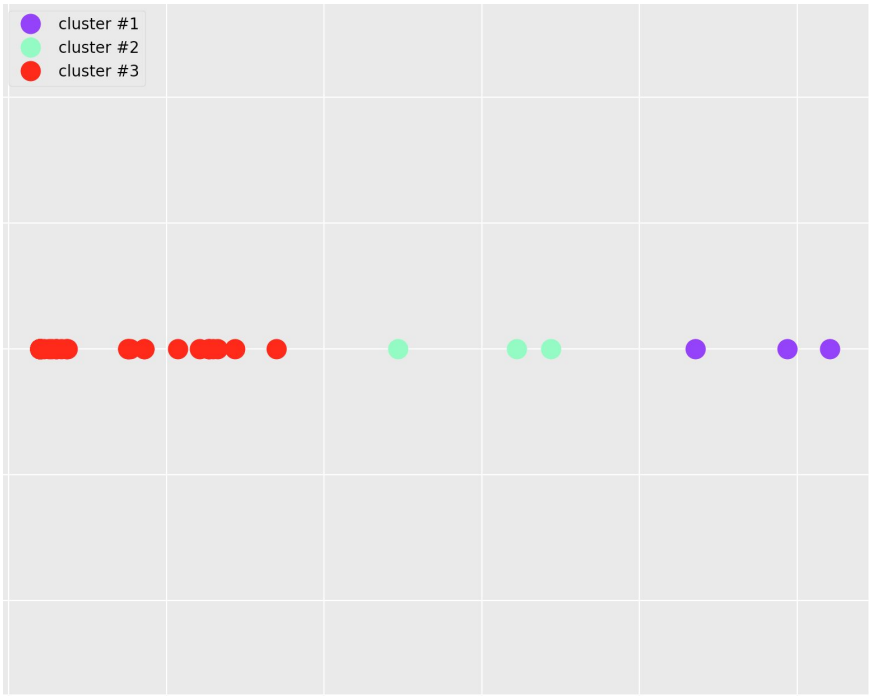
Primeri primene u bioinformatici

1) **Klasterovanja genskih ekspresija u cilju detektovanja kancera debog creva**

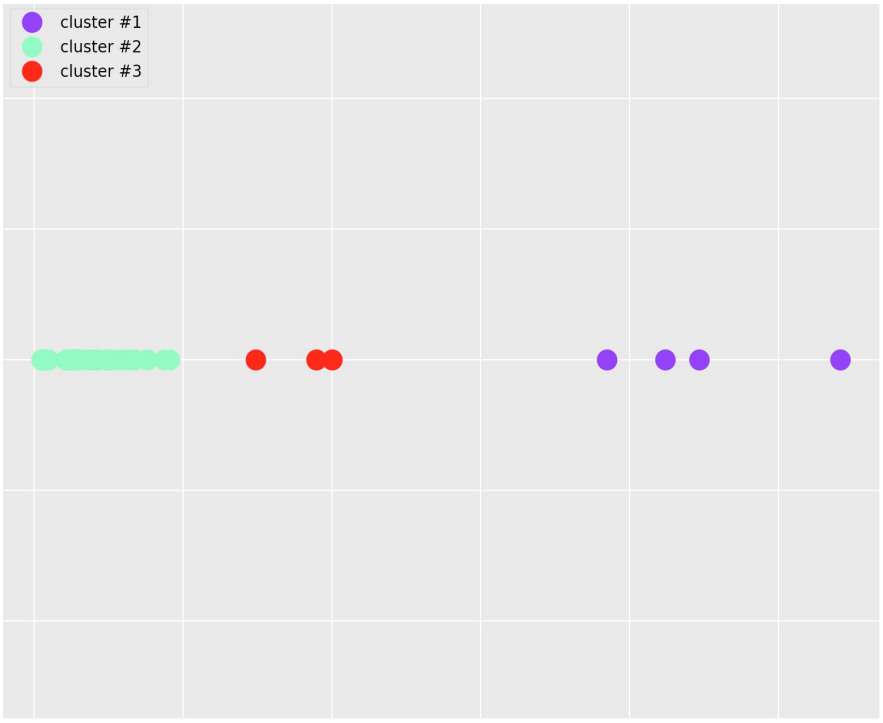
Poznate su genske ekspresije zdravih osoba i osoba obolelih od kancera debelog creva. Cilj je izvršiti klasterovanje primenom algoritma maksimizacije očekivanja i za nepoznatu instancu odrediti kom klasteru pripada. Ceo skup podataka se sastoji od 2000 gena i njihovih ekspresija u 40 uzoraka kancera debelog creva, kao i 20 uzoraka zdravih tkiva. Kako bismo ubrzali proces računanja, umesto posmatranja ekspresije jednog gena u 60 uzoraka, kao relevantan podatak se uzima prosečan intenzitet ekspresije u tumornom tkivu i normalnom tkivu, čija je vrednost računski određena na osnovu svih intenziteta ekspresija. Intenzitet ekspresije predstavlja određivanje prosečnih vrednosti gena uključujući metode za filtriranje i centriranje podataka. Način predstavljanja genskih ekspresija prikazan je u tabeli:

Accession no.	Description	Intensity in tumor	Intensity in normal
X54489	**Human gene for MGSA**	105.1	10.0
U22055	**Human 100 kDA coactivator mRNA, complete cds** [a]	72.9	10.0
D14657	Human mRNA for *KIAA0101* gene, complete cds	64.8	10.0
M61832	Human *S*-adenosylhomocysteine hydrolase (AHCY) mRNA, complete cds	123.1	20.7

U našem primeru dva puta smo izvršili klasterovanje: klasterovanje intenziteta genskih ekspresija u tumorom tkivu, kao i klasterovanje i normalnih genskih ekspresija. Broj klastera u oba slučaja je 3. U nastavku je data vizualizacija rezultata:



Klasteri ekspresija u tumornom tkivu



Klasteri ekspresija u normalnom tkivu

Na osnovu prikaza zaključujemo da tačke koje pripadaju prvom klasteru imaju sličnu raspodelu. Dakle, geni koji pripadaju ovom klasteru nisu relevantni za dalje tumačenje rezultata.
Sa druge strane, tačke koji pripadaju drugom i trećem klasteru imaju različite raspodele, pa se geni koji odgovaraju ovim tačkama uzimaju u obzir pri daljoj analizi i detektovanju bolesti kao potencijalni biomarkeri, odnosno identifikatori patološkog procesa u organizmu. Na ovaj način odbacili smo značajan broj gena i smanjili prostor pretrage. Nadalje, pacijentu za kog se sumnja da ima tumor se mogu uraditi analize samo malog skupa gena (u ovom slučaju detektovanog na ovaj način) i tako sa velikom verovatnoćom doneti zaključak o stanju pacijenta.

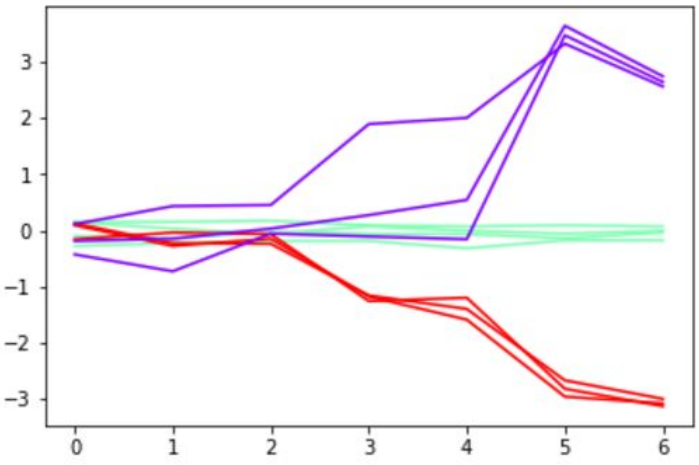
2) Klasterovanje gena kvasca

Vrste kvasca *S. cerevisiae* vrši fermentaciju alkohola pretvarajući glukozu iz voća u etanol. Kada kvasac iskoristi svu glukozu iz voća, a kako bi preživeo, počinje da se hrani etanolom koji je sam proizveo. Na taj način, alkohol koji se proizvodi gubi na kvalitetu. Postavlja se pitanje na kojim genima kvasca dolazi do promene kako bi pomenuti proces bio izvodljiv. Rađena je studija u kojoj je posmatrano 6400 gena kvasca u 7 vremenskih trenutaka i cilj studije je bio klasterovati ove gene. Mi smo naš algoritam klasterovanja pustili nad manjim broj gena (zbog jednostavnosti izračunavanja) i podaci koje smo koristili su dati u tabeli:

	GSM887	GSM888	GSM889	GSM890	GSM891	GSM892	GSM893
0	0.14	0.03	-0.06	0.07	-0.01	-0.06	-0.01
1	0.12	-0.23	-0.24	-1.16	-1.40	-2.67	-3.00
2	-0.10	-0.14	-0.03	-0.06	-0.07	-0.14	-0.04
3	-0.43	-0.73	-0.06	-0.11	-0.16	3.47	2.64
4	0.11	0.43	0.45	1.89	2.00	3.32	2.56
5	0.09	-0.28	-0.15	-1.18	-1.59	-2.96	-3.08
6	-0.16	-0.04	-0.07	-1.26	-1.20	-2.82	-3.13
7	-0.28	-0.23	-0.19	-0.19	-0.32	-0.18	-0.18
8	-0.19	-0.15	0.03	0.27	0.54	3.64	2.74
9	0.15	0.15	0.17	0.09	0.07	0.09	0.07

Gene smo grupisali u tri klastera, tako da po klasterima ekspresije gena rastu, opadaju ili su konstantne u različitim vremenskim trenucima. Nivoi ekspresije gena koji utiču na pomenuti proces treba da se razlikuju u datim vremenskim

trenucima i samo takvi geni su nam od značaja, pa iz tog razloga klaster sa konstantim ekspresijama možemo odbaciti jer se na grafiku vidi da on ni na koji način ne utiče na ovaj proces. Na ovaj način smo takođe smanjili prostor pretrage gena od interesa.



Prikaz klastera i ekspresija gena

Reference:

[Podaci - tumor/normal geni]
[Podaci - geni kvasca]
[Bioinformaticki algoritmi - poglavlje 8]