Spektralno grupiranje

Špela Ognjanović in Žiga Trojer

24. november 2018

1 Navodilo

Implement spectral clustering algorithms that use several types of Laplace matrices. Consider unnormalized spectral clustering, normalized spectral clustering according to Shi and Malik (2000), and normalized spectral clustering according to Ng, Jordan, and Weiss (2002). Generate various data sets or find some examples of real world data and use different types of similarity graphs such as complete graph, the ϵ -neighborhood graph and k-nearest neighbor graph. Compare the results. Use different methods for determining the optimal number of clusters.

2 Kratek opis

Imamo bazo podatkov, naš cilj je te podatke razvrstiti v skupine oz. grupe s podobnimi lastnostmi. Podatke lahko prikažemo z grafi, grafe pa z matrikami. Eden takih primerov je **matrika sosednosti**, njena razsežnost je n × n. Element v *j*-tem stolpcu *i*-te vrstice pove število povezav, ki povezujejo točki *i* in *j*.

Predstavili bomo, kako lahko na različne načine pogrupiramo podatke, podane z množico točk x_1, x_2, ... , x_n z utežmi $w_{i,j} \geq 0$, za $\forall i, j = 1, ...n$ ali razdaljami $d_{i,j} \geq 0$, za $\forall i, j = 1, ...n$. Če o podatkih nimamo veliko informacij, je najlažje, če jih predstavimo z grafom *G=(V, E)*. Vsako vozlišče v_i predstavlja en podatek x_i. Potrebujem matriko, ki določa, kako blizu skupaj sta vozlišči v_i in v_j. To matriko imenujemo podobna matrika in jo označimo s *S*. Dva vozlišča povežemo, če zadoščata danemu pogoju in to ponazorimo v podobnem grafu. Želimo, da imajo vozlišča v istih skupinah čim manjšo uteženost, kar pomeni da imajo podatki podobne lastnosti, sicer velja obratno. Spodaj so opisani trije primeri podobnih grafov, s katerimi bomo operirali.

Graf ϵ -ske okolice
Pri ϵ -ski okolici grafa povežemo paroma vse točke, katerih razdalja je manjša od ϵ. Največkrat je obravnavan kot graf brez uteži, saj nam utež predstavlja predpisana razdalja ϵ.

Graf k najbližjih sosedov
Cilj je, da povežemo vozlišče v_i z njegovimi k najbližjimi sosedi. Dobimo, da je v_i med k najbližjimi sosedi od v_j ter v_j med k najbližjimi sosedi od v_i.

Poln graf
Je graf, kjer so vsa vozlišča med seboj povezana.

Vrnimo se na podobno matriko *S*. Njene elemente izračunamo s pomočjo neke primerne funkcije. Primer take funkcije je Gaussian Karnelova funkcija, ki nam izračuna oddaljenost dveh vozlišč v_i in v_j. Formula za izračun je:

$$s(v_i, v_j) = exp(-||v_i - v_j||^2/(2\delta^2)),$$

pri čemer parameter δ opiše razdaljo med vozlišči, podobno kot ϵ pri grafu epsilonske okolice (vzamemo $\delta = 1$). Vrednosti $s_{i,j}$, izračunane po tej meri, so

na intervalu [0, 1]. Povejo nam, da manjša kot je vrednost, bolj sta točki oddaljeni. Če je vrednost 0, pomeni, da sta vozlišči daleč narazen. Matrika S je simertrična, saj za toliko kot je vozlišče v_i oddaljena od vozlišča v_j, je tudi vozlišče v_j oddaljena od vozlišča v_i. Očitno je, da so diagonalni elementi enaki 1.

Naslednji korak je izračun **Laplaceove matrike**. Privzamemo, da imamo utežen, neusmerjen graf G. Uteži so prikazane v matriki W, $w_{i,j} \geq 0$, za $\forall i, j = 1, ...n$. Definirajmo še diagonalno degree matriko D:

$$d_i = \sum_{j=1}^{n} w_{i,j}.$$

Glede na normalizirane in nenormalizirane ločimo dva primera Laplaceovih matrik:

1. Nenormalizirana Laplaceova matrika, ki se izračuna po formuli $L = D - W$, kjer je W matrika uteži in D diagonalna matrika. Njene lastnosti so:
   i) Je simetrična, pozitivno semi-definitna.
   ii) Njena najmanjša lastna vrednost je 0, njen pripadajoč lastni vektor je konstanten vektor $\mathbb{1}$.
   iii) Ima n nenegativnih, realni lastnih vrednosti $0 \leq \lambda_1 \leq ... \leq \lambda_n$.
   iv) Za vsak $f \in \mathbb{R}^n$ je
   $$f'Lf = \frac{1}{2} \sum_{i,j=1}^{n} w_{i,j}(f_i - f_j)^2.$$

2. Normalizirana Laplaceova matrika, sta v bistvu dve matriki, ki sta med seboj povezani, prva je simetrična in se izračuna $L_{sym} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, druga pa se izračuna po formuli $L_{rw} = I - D^{-1}W$. Njune lastnosti so:
   i) Sta pozitivno semi-definitni in imata n nenegativnih, realnih lastnih vrednosti.
   ii) 0 je lastna vrednost matrike L_{rw}, s pripradajočim konstantnim vektorjem $\mathbb{1}$. 0 je tudi lastna vrednost matrike L_{sym}, s pripadajočim lastnim vektorjem $D^{\frac{1}{2}}\mathbb{1}$.
   iii) λ je lastna vrednost matrike L_{rw} z lastnim vektorjem u, natanko tedaj, če u reši sistem $Lu = \lambda Du$.
   iv) λ je lastna vrednost matrike L_{rw} z lastnim vektorjem u, natanko tedaj, če je λ lastna vrednost matrike L_{sym} z lastnim vektorjem $w = D^{\frac{1}{2}}u$.
   v) Za vsak $f \in \mathbb{R}^n$ je
   $$f'L_{sym}f = \frac{1}{2} \sum_{i,j=1}^{n} w_{i,j}(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}})^2.$$

S pomočjo Laplaceove matrike ugotovimo, kako dobro smo podatke pogrupirali.

3 Načrt za reševanje

Prednost algoritma spectral clustering je, da sestoji na preprosti linearni algebri (npr. izračun lastnih vektorjev, lastnih vrednosti, karakterističnega polinoma, ...). Imamo n točk x_1, x_2, ... , x_n, ki lahko ponazarjajo poljubne objekte. Merimo paroma podobne $s_{i,j} = s(x_i, x_j)$, glede na neko simetrično, nenegativno funkcijo (mi uporabimo Gaussian Karnelovo funkcijo). Tako označimo pripadajočo podobno matriko $S = (s_{i,j})_{i,j=1,...,n}$. Ločimo:

a) Nenormaliziran algoritem.

b) Normaliziran algoritem glede na Shi in Malik.

c) Normaliziran algoritem glede na Ng, Jordan in Weiss.

Pri vseh se reševanja lotimo na enak način. Vhodni podatek je podobna matrika $S \in \mathbb{R}^{n \times n}$, s k označimo število grup, ki jih želimo skonstruirati. Nato skonstruiramo enega od podobnih grafov, opisanega v prejšnjem razdelku. Z W označimo uteženo matriko sosednosti. Po formulah izračunamo pripadajočo normalizirano ali nenormalizirano Laplaceovo matriko L. Nato poračunamo prvih k najmanjših pripadajočih lastnih vektorjev matrike L. V matriko U $\in \mathbb{R}^{n \times k}$ zložimo k lastnih vektorjev v stolpce, po padajočem vrestnem redu glede na pripradajočo lastno vrednost. Vpeljemo zanko for $i = 1, ..., n$. Nastali vektor y_i predstavlja i-to vrstico matrike U. S pomočjo tega lahko skonstruiramo graf in vidimo, kako dobro smo pogrupirali. Če so točke na grafu zelo narazen pomeni, da smo zelo dobro pogrupirali in obratno.

Množico točk $(y_i)_{i=1,...,n} \in \mathbb{R}^k$ s pomočjo k-means algoritma razvrstimo v grupe $C_1, ..., C_k$. Dobimo $A_1, ..., A_k$, kjer $A_i = \{j | y_i \in C_i\}$. Grupe lahko tudi grafično prikažemo.

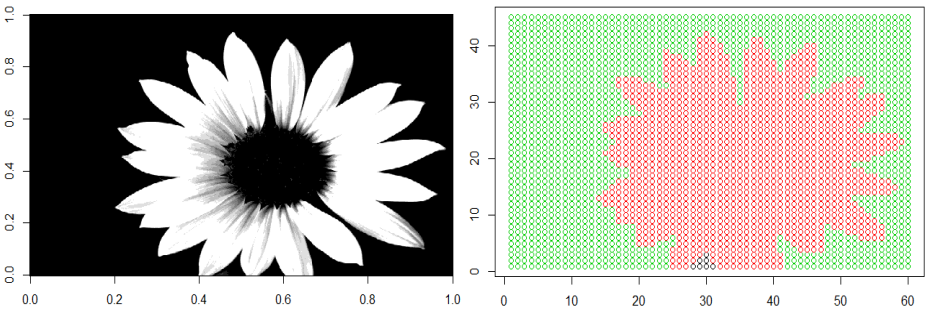
Glavna stvar, ki smo jo naredili je, da smo podatke x_i prevedli na $y_i \in \mathbb{R}^k$ in jih tako lažje pogrupirali. S pomočjo lastnih vrednosti Laplaceove matrike lahko izračunamo tudi optimalen k, oziroma optimalno število grup, ki bi jih naj algoritem skonstruiral glede na dano množico podatkov. To naredimo tako, da lastne vrednosti po naraščajočem vrstnem redu upodobimo s točkastim grafom. Optimalno število grup lahko kar odčitamo z grafa, tako, da preštejemo število točk do preskoka.

4 Primeri in uporaba

Uporaba algoritma je zelo razširjena v podatkovni analizi, tudi na vseh naravoslovnih področjih, saj vedno kadar se lotimo nekega problema, kjer so baze podatkov, poskušamo dobiti prvi vtis tako, da pogledamo skupne lastnosti. S pomočjo tega algoritma dobimo podatke zelo dobro posortirane in posledično boljši vtis.

Oglejmo si kako algoritem deluje na najinih primerih.

Za lažjo predstavo sva se odločila narediti še en poseben primer, zgled uporabe algoritma na slikah. Sprogramirala sva ga v programu R, tako, da sva najprej vzela sliko z interneta, jo spravila v matriko in nato na njej izvedla algoritem. Rezultati so vidni spodaj na dveh slikah.



(a) Prvotna slika (b) Rezultat