Semestrální práce z KIV/PRO

Využití Voroného diagramu pro inicializaci K-means shlukování

Jméno Příjmení (Osobní číslo)

11. prosince 2014

Obsah

1 Úvod

Celá tato práce vychází ze článku [1] *Initialization for K-means clustering using Voronoi diagram* od autorů Damodar Reddy a Prasanta K. Jana. Veškeré údaje a obrázky v tomto dokumentu jsou vlastní tvorby a z původního článku jsou zde reprezentovány pouze myšlenky. V celém dokumentu je snaha vše psát co nejsrozumitelněji, aby nebyl problém pro studenta 2. ročníku snadno pochopit celou problematiku.

K-means je jeden z nejznámějších a nejpoužívanějších algoritmů ve shlukování. Tento algoritmus je velice jednoduchý a poměrně efektivní, ale má jisté nevýhody. Hlavním problémem této metody je to, že není možné zaručit globální optimum. Je to díky tomu, že se počáteční středy shluků vybírají náhodně. V případě, že jsou data dostatečně rozdílná, výsledky jsou konzistentní. Ovšem pokud shluky nejsou jednoznačné, je pravděpodobné, že se při opakovaném spuštění algoritmu, budou výstupy nad stejnými daty lišit.

Autoři původního článku [1] proto přicházejí s inovativní myšlenkou - Využívají k inicializaci Voroného diagramu, který je zkonstruován ze vstupních dat. Počáteční středy shluků jsou efektivně vybrány z bodů, které leží na největších Voroného kruzích. Výsledkem je algoritmus, který automaticky vytvoří vhodné počáteční středy shluků pro metodu K-means. Autoři článku [1] tuto metodu testovali jak na reálných, tak na uměle vytvořených datech, a z výsledků těchto testů zjistili, že použití tohoto algoritmu umožňuje přesnější shlukování než při využití tradičního K-means nebo vylepšeného K-means.

2 Vysvětlení pojmů

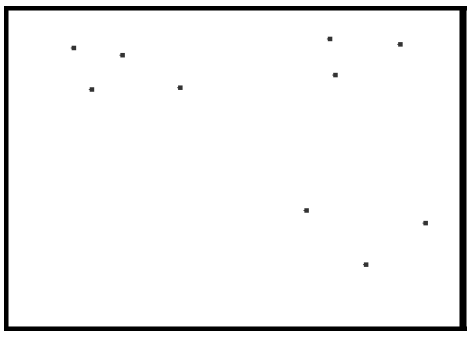
2.1 K-means shlukování

Tento algoritmus [2] vytvoří z množiny n bodů, kde body mohou být libovolné dimenze (viz Obrázek 1.), k shluků. K-means jednotlivé body přiřadí do shluků tak, aby se v každém shluku nacházely body, které si jsou v prostoru nejblíže (viz Obrázek 2.).

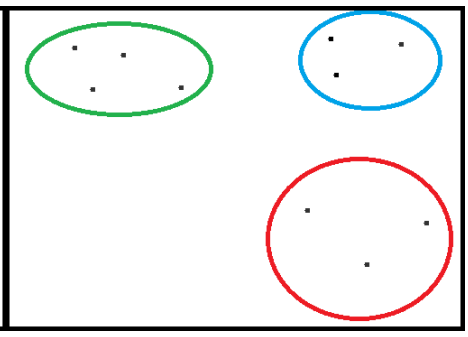
Algoritmus se skládá z těchto kroků:

1. Nejprve se náhodně vybere k bodů ze vstupní množiny n. Tyto body budou sloužit jako počáteční středy shluků.
2. Nyní se vezmou všechny body množiny n a porovná se, ke kterému ze středů mají jednotlivé body nejblíže. Tímto vznikne k skupin bodů. Není důležité, kolik bodů se v jednotlivém shluku nachází. Může se tedy stát, že se v některém shluku bude nacházet pouze středový bod.
3. Jakmile jsou všechny body přiřazeny, vypočtou se nové středy shluků. To se provede tak, že se vypočte středová hodnota všech bodů dané skupiny (tedy provede se aritmetický průměr jednotlivých složek bodů).
4. Pokud se středy již nezměnily, shlukování je dokončeno a jednotlivé skupiny jsou naše žádané shluky. V opačném případě se vracíme k 2. kroku.

Po pochopení algoritmu je patrné, že počáteční určení středů má velký význam na finální rozložení shluků.



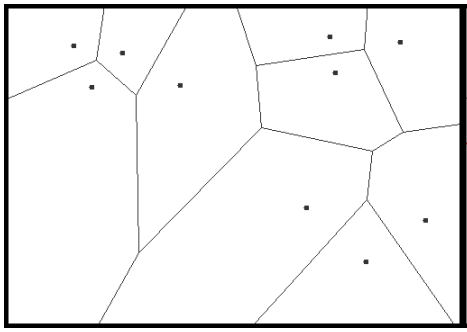
Obrázek 1: Počáteční situace



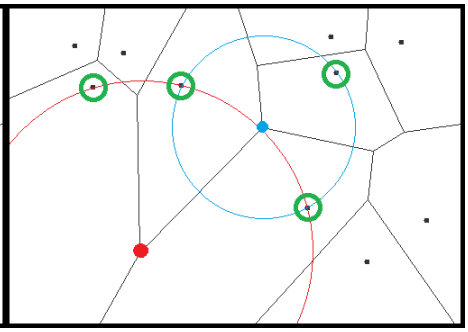
Obrázek 2: K-means

2.2 Voroného diagram

Je dána množina S o n bodech v m dimenzionálním euklidovském prostoru (viz Obrázek 3.). Voroného diagram[3] funguje tak, že každému z těchto bodů je přidělen prostor. Hranice těchto prostorů jsou přesně specifikovány a to tak, že každému bodu je přidělena ta oblast, která žádnému jinému bodu není blíže. Místa, která jsou stejně vzdálená od více bodů, pak tvoří hranice těchto oblastí. Průsečíky těchto hranic jsou nazývány vrcholy, kterých se v celém diagramu nachází maximálně *2n - 5*. V případě, že ve vrcholu utvoříme střed kružnice a zvolíme její poloměr jako vzdálenost k nejbližším bodům, všechny tyto body, tedy všechny body v hraničních oblastech, budou ležet na obvodu této kružnice. Těmto kružnicím se říká Voroného kružnice a jedná se o největší prázdné kružnice. Při hledání těchto největších prázdných kružnic musíme také vzít v potaz průsečíky hran diagramu s konvexní obálkou. I tyto body mohou být kandidáty a je třeba je prověřit. To znamená, že je zde snaha udělat z daného bodu co největší poloměr tak, aby se žádný bod nenacházel uvnitř (viz Obrázek 4.).



Obrázek 3: Voroného diagram



Obrázek 4: Voroného kruhy

2.3 Míra chybovosti

Kvalita celého algoritmu byla poměřována pomocí tzv. míry chybovosti. Tato hodnota byla vyjádřena v procentech a jedná se o poměr počtu chybně klasifikovaných prvků oproti celkovému počtu prvků. Z toho samozřejmě vyplývá, že čím menší je míra chybovosti pro danou situaci, tím lépe jí algoritmus zvládl.

*ER = Počet chybně klasifikovaných prvků / Celkový počet prvků * 100*

3 Navržená metoda

3.1 Algoritmus

Inovační metoda[4] navržena autory článku[1] se skládá ze dvou fází. V první fázi se spočtou pozice počátečních středů jednotlivých shluků. Ve druhé fázi se tyto středy použijí jako vstup do algoritmu K-means.
Celá metoda tedy funguje tímto způsobem. Nejprve se ze vstupní množiny bodů sestaví Voroného diagram. Poté vyhledáme všechny vrcholy, které zformováním diagramu vznikly. Tyto vrcholy seřadíme podle poloměru Voroného kružnice, kterou bychom tu mohli sestrojit, od největšího po nejmenší a v tomto pořadí je uložíme do pole *sortedVertices[]*. Nyní si vytvoříme další pole, které pojmenujeme *test[]*. Postupně do něj budeme ukládat body ležící po obvodu kružnice, tedy všechny naše kandidáty. Do dalšího pole, které pojmenujeme *centers[]*, budeme ukládat výsledné středy. Postupně budeme nad polem *sortedVertices[]* provádět tyto akce:

1. Všechny body, které leží na obvodu aktuální kružnice, uložíme do *test[]*. Jestliže bude vzdálenost mezi libovolnými dvěma body v tomto poli menší, než je poloměr kružnice, jeden z těchto bodů odebereme (není důležité který). Takto ověříme celé pole *test[]*.

2. Nyní provedeme stejné ověření, pouze s tím rozdílem, že porovnáme vzdálenosti mezi body v *test[]* a body v *centers[]* (v první iteraci je pole *centers[]* prázdné a tak tento krok přeskočíme). Pokud bude vzdálenost mezi těmito body menší než poloměr aktuální kružnice, bod v poli *test[]* smažeme.

3. Zbývající body v poli test[] přesune do *centers[]* a pole *test[]* vyprázdníme. Pokud se po přesunu bodů bude v *centers[]* nacházet k bodů, tedy tolik, kolik jsme si zvolili, že má být středů, algoritmus ukončíme. Pokud po přesunutí celého pole bude v *centers[]* méně než k bodů, pokračujeme další iterací, tedy další kružnicí.

Jakmile se v *centers[]* nachází k prvků, můžeme toto pole předat algoritmu K-means. Ten se již spustí běžným způsobem a tyto body použije jako počáteční středy shluků.

3.2 Pseudokód

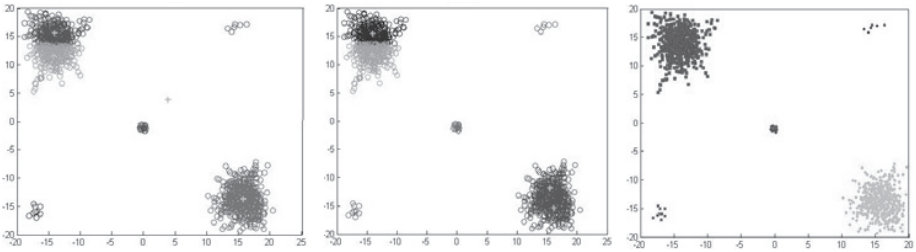
vstup: Množina S o n bodech a počet požadovaných shluků k
výstup: k shluků vytvořených z množiny dat S

Metody použité v pseudokódu:
voronoi(S): vytvoří Voroného diagram z množiny bodů S a vrátí pole vrcholů verteces[]
r(v): vrátí poloměr největší prázdné kružnice se středem v daném vrcholu
sort(vertices): sestupně seřadí pole vrcholů na základě poloměru r(v)
circumpoints(v): nalezne všechny body na obvodu kružnice se středem ve vrcholu v
dist(p, q): spočte euklidovskou vzdálenost mezi body p a q
kMeans(S, centers[]): vrátí shluky vytvořené z množiny dat S, počátečními středy shluků budou prvky v centers[], počet shluků závisí na velikosti centers[]

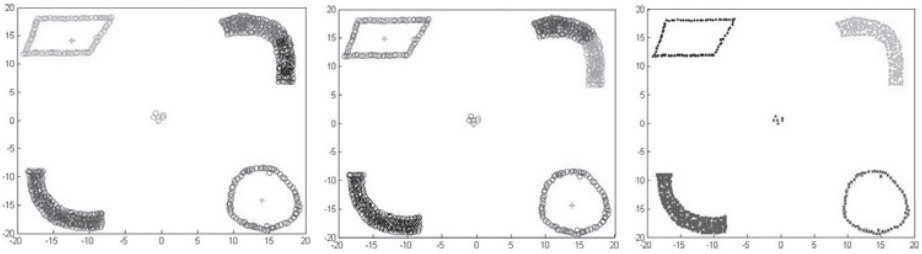
```
01 vertices[] = voronoi(s)
02 sortedVertices = sort(vertices)
04 pro každé v ze sortedVertices[]{
05    test[] = circumpoints(v)
06    pro každý bod z test[], i=0 {
07       pro každý bod z test[], j=0 {
08          if (i!=j && dist(test[i], test[j]) < r(v))
09             odeber test[i] z test[]                ##šlo by i test[j]
10       }
11    }
12    pro každý bod z test[], i=0 {
13       pro každý bod z centers[], j=0 {
14          if (dist(test[i], centers[j]) < r(v))
15             odeber test[i] z test[]
16       }
17    }
18    zkopíruj obsah test[] do centers[], obsah test[] vymaž
19    if(sizeof(centers[]) == k)
20       break
21    else if(sizeof(centers[]) > k)
22       ořízni centers[] na velikost k
23       break
24 }
25 kMeans(S, centers[])
```

3.3 Výsledky a experimenty

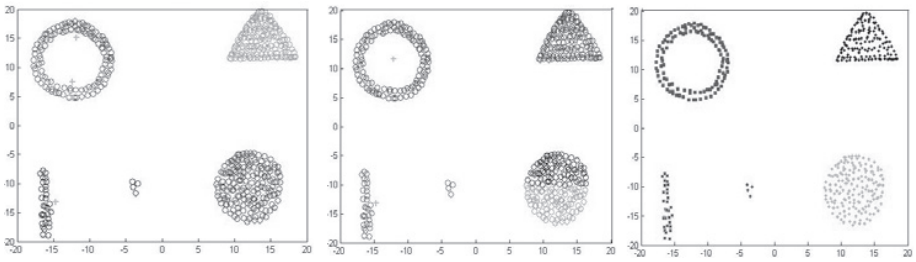
Aby bylo možné zhodnotit výsledky, autoři spustili navržený algoritmus nad třemi různými umělými i reálnými daty z databáze UCI[5]. Tyto výsledky[6] byly porovnány s tradičním i vylepšeným K-means. Z následujících obrázků je patrné, že si tyto algoritmy vedly různě, nicméně autory navržený algoritmus nad nimi vyčnívá. V prvním případě byly algoritmy aplikovány na množinu dat s pěti clustery, kde tři z nich jsou velice malé, aby simulovaly výskyt dat, která nikam nezapadají. (viz Obrázek 5. a-c). Algoritmy dosáhly podobných výsledků i u další množiny zvané Ring-curve-parallelogram (viz Obrázek 6. a-c). Poslední testy byly provedeny na množině dat se čtyřmi clustery (viz Obrázek 7. a-c). Porovnání jednotlivých výsledků je znázorněno v tabulce (viz. Obrázek 8.). ER, které je použité v tabulce, znamená míru chybovosti. Tento pojem byl vysvětlen v kapitole 2.3.



Obrázek 5: Výsledky dat s pěti clustery a) K-means; b) Vylepšený K-means; c) Navržený algoritmus



Obrázek 6: Výsledky dat Ring-curve-parallelogram a) K-means; b) Vylepšený K-means; c) Navržený algoritmus



Obrázek 7: Výsledky dat se čtyřmi clustery a) K-means; b) Vylepšený K-means; c) Navržený algoritmus

Typ množiny	Množství vzorků	Špatně zařazené vzory			ER		
		K-means	K-means+	Návrh	K-means	K-means+	Návrh
Iris	150	22	20	17	14.67	13.33	11.33
S. Heart	187	14	13	14	7.49	6.95	7.49
Wine	178	23	19	17	12.92	10.67	9.56
Ecoli	336	65	54	45	19.34	16.7	13.39
St. Heart	270	44	41	30	16.23	15.19	11.11
P. I. Diabetes	768	193	153	109	25.13	19.92	14.19
Soybean	47	11	8	8	23.40	17.2	17.2
B. Tissue	106	23	19	16	21.69	17.92	15.9

Obrázek 8: Tabulka výsledků

3.4 Zhodnocení

Díky této metodě[4] nám nyní bude K-means vracet přesnější výsledky. Další výhodou je také to, že výstupy nad danými daty budou stejné, což značně usadní testování. Z pokusů provedených autory plyne, že oproti klasickému K-means je zde přibližně 41% zlepšení v míře chybovosti a oproti vylepšenému K-means je chybovost nižší přibližně o 18%. To jsou velmi zajímavá čísla, obzvláště vezmeme-li v potaz to, že se časová náročnost celého algoritmu příliš nezmění. Vorného diagram je vytvořen za $O(n \log n)$ času a stejné množství času je také potřeba pro seřazení pole vrcholů. Samotný K-means má složitost $O(knt)$, kde k je počet clusterů, n je množství dat a t je počet iterací. Složitost K-means tedy dominuje a složitost celého algoritmu je také $O(knt)$

4 Závěr

Myslím si, že tento algoritmus[4] má velice slibnou budoucnost, pokud se dostane do podvědomí veřejnosti. Zlepšení přesnosti oproti standardnímu a vylepšenému K-means je značné, ale zároveň se příliš nemění časové náklady. Samozřejmě je zde o něco více kódování, ale pokud pracujete na aplikace, kde je rozdělení do shluků kritické, je to dle mého názoru dobrá časová investice.
Autoři se plánují do budoucna zaměřit na další vývoj tohoto algoritmu[7]. Jejich cílem je, aby byla byla součástí návrhu metoda schopná zjistit, kolik shluků se v množině dat nachází a nebylo tak nutné tuto hodnotu ručně zadávat. To by opět otevřelo úplně nové možnosti pro K-means, protože se často při shlukování ocitáme v situaci, kdy tuto hodnotu neznáme. Takto plně zautomatizovaný K-means by bylo velice pohodlné používat a pokud by se časová složitost udržela na rozumné hodnotě, měli bychom co do činění s dalším krokem kupředu v oblasti shlukování.

Reference

[1] Damodar Reddy, Prasanta K. Jana: *Initialization for K-means Clustering using Voronoi Diagram.* Procedia Technology, Elsevier, 2012

[2] Damodar Reddy, Prasanta K. Jana: *Initialization for K-means Clustering using Voronoi Diagram.* Procedia Technology, Elsevier, 2012;2.1:396

[3] Damodar Reddy, Prasanta K. Jana: *Initialization for K-means Clustering using Voronoi Diagram.* Procedia Technology, Elsevier, 2012;2.2:396-397

[4] Damodar Reddy, Prasanta K. Jana: *Initialization for K-means Clustering using Voronoi Diagram.* Procedia Technology, Elsevier, 2012;3:397

[5] UCI (University of California, Irvine) Machine Learning Repository *http://archive.ics.uci.edu/ml/datasets.html*

[6] Damodar Reddy, Prasanta K. Jana: *Initialization for K-means Clustering using Voronoi Diagram.* Procedia Technology, Elsevier, 2012;4:399-400

[7] Damodar Reddy, Prasanta K. Jana: *Initialization for K-means Clustering using Voronoi Diagram.* Procedia Technology, Elsevier, 2012;5:400