КЛАСТЕРИРАЊЕ СО DBSCAN

ДЕТАЛЕН ПРЕГЛЕД

Благоја Евкоски

АБСТРАКТ

Целта на овој труд е детален преглед на алгоритмот за кластерирање DBSCAN. Ќе бидат презентирани основните концепти на алгоритмот; ќе биде анализиран самиот алгоритам, неговата комплексност и начини за подобрување на истата, предности и недостатоци, а неговите преформанси ќе бидат споредени со алгоритмот за кластерирање CLARANS.

1. ВОВЕД

Density-based spatial clustering of applications with noise (DBSCAN) е алгоритам за кластерирање податоци првпат презентиран во 1996 година. Креиран е од потребата за:

- Минимално знаење за доменот при одредување на влезните параметри
- Одредување кластери со произволна форма
- Ефикасност над големи податочни множества

DBSCAN е density-based алгоритам (алгоритам базиран на густина). Оваа група на алгоритми ги дефинираат кластерите како површини со поголема густина од остатокот од податочното множество. Објектите кои се наоѓаат во празните делови на податочното множество обично се дефинираат како шум или гранични објекти.

DBSCAN може да идентификува кластери во големи податочни множества според локалната густина на елементите во базата, користејќи еден влезен параметар. Додатно, корисникот добива сугестија за тоа која вредност на параметарот најмногу би одговарала, што ја намалува потребата за знаење за доменот. И покрај тоа, алгоритмот работи многу брзо и е високо скалабилен; неговата комплексност расте скоро линеарно како што расте големината на базата. Најбитно, DBSCAN може да одреди кластери со произволна форма (пример, слика 1).

СЛИКА 1: РАСПРЕДЕЛБАТА НА ОБЈЕКТИТЕ ВО ТРИ РАЗЛИЧНИ БАЗИ НА ПОДАТОЦИ, ЗЕМЕНИ ОД БЕНЧМАРК БАЗАТА SEQUOIA 2000

2. ОСНОВНИ КОНЦЕПТИ

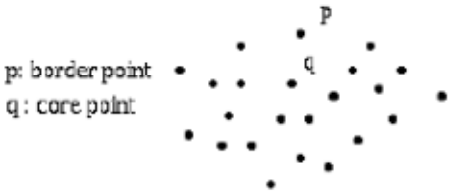
Процесот на пресметување на алгоритмот DBSCAN се основа на шест дефиниции, од кои се изведуваат две леми.

Дефиниција 1: (Eps-соседство на точка)

За дадена точка да припаѓа на еден кластер, мора да има барем една точка која лежи поблиску до неа од растојанието Eps.

$$N_{Eps}(p) = \{q \in D \mid dist(p,q) < Eps\}$$

Постојат два типа на точки кои припаѓаат на еден кластер; гранични точки и средишни точки, како што може да се види на слика 2.



СЛИКА 2: ГРАНИЧНИ И СРЕДИШНИ ТОЧКИ

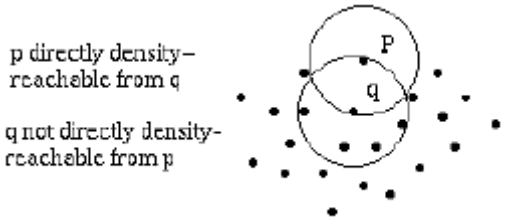
Дефиниција 2: (Директно густински достигливи)

Eps-соседството на гранична точка тежнее кон тоа да има значајно помалку точки од Eps-соседството на средишна точка. За гранична точка p да припаѓа на одреден кластер, таа точка мора да припаѓа на Eps-соседството на средишна точка q.

1) p е $N_{Eps}(q)$

За една точка да биде средишна точка, мора да има некој одреден минимален број на точки во нејзиното Eps-соседство.

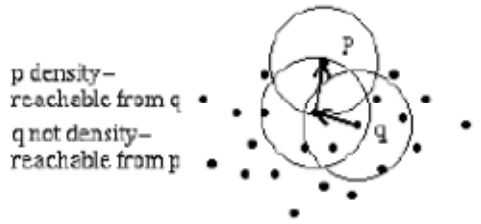
2) $|N_{Eps}(q)| \geq MinPts$(core point condition)



СЛИКА 3: ПРИМЕР ЗА ГРАНИЧНИ И СРЕДИШНИ ТОЧКИ

Дефиниција 3: (Густински достигливи)

Точка p е густински достиглива од точка q при параметри Eps и MinPts ако има ланец од точки $p_1...,p_n$, каде p_1=q, а p_n=p така што секоја точка p_{i+1} е директно густински достиглива од p_i.

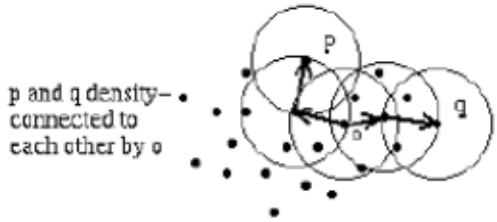


СЛИКА 4: P Е ГУСТИНСКИ ДОСТИГЛИВА ОД Q, НО НЕ И ОБРАТНО

Постојат случаи кога две гранични точки кои припаѓаат на истиот кластер не делат иста средишна точка. Во тој случај мора да постои средишна точка o од која и двете се густински достигливи.

Дефиниција 4: (Густински поврзани)

Точка p е густински поврзана со точка q при параметри Eps и MinPts ако постои точка o од која и двете, p и q се густински достигливи.



СЛИКА 5: ГУСТИНСКА ПОВРЗАНОСТ

Дефиниција 5: (Кластер)

Ако точка p е дел од кластер C и точка q е густински достиглива од p при параметри Eps и MinPts, тогаш q е исто така дел од кластерот C.

1) ∀p,q ако p∈C и q е густински достиглива од p при параметри Eps и MinPts, тогаш q∈C

Две точки припаѓаат на истиот кластер C е исто со – p е густински поврзана со q при параметри Eps и MinPts.

2) ∀p,q∈C: p е густински поврзана со q при параметри Eps и MinPts

Дефиниција 6: (Шум)

Шум е множество точки во базата на податоци кои не припаѓаат на ниту еден кластер.

Лема 1:

Кластер може да се формира од која било средишна точка. Притоа, кластерот секогаш ќе ја има истата форма.

Лема 2:

Нека p е средишна точка во кластер C со дадено минимално растојание (Eps) и минимален број на точки во тоа растојание (MinPts). Ако множеството O е множество од точки кои се густински поврзани со p при истите параметри Eps и MinPts, тогаш кластерот C е еднаков на множеството O.

3. АЛГОРИТАМ

DBSCAN побарува два параметри: E (Eps) и минималниот број на точки за формирање на регион (MinPts).

Алгоритмот започнува со произволна почетна точка која не била посетена. Се одредува Eps-соседството на таа точка и ако соседството содржи доволно точки, се започнува кластер. Во спротивно, точката се означува како шум. Битно е да се напомене дека доколку истата точка подоцна биде пронајдена како дел од доволно големо Eps-соседство на некоја друга точка, може да биде додадена на кластер.

Ако некоја точка е одредена како средишна точка на кластер, нејзиното Eps-соседство е исто така дел од тој кластер. Па така, сите точки означени како дел од Eps-соседството се додаваат, како што се додаваат и точките од нивното Eps-соседство доколку се одреди дека и тие се средишни точки. Овој процес продолжува сè додека не се формира комплетно густински поврзан кластер. Потоа, се посетува и процесира нова непосетена точка, која води до откривање на нов кластер или пак шум.

Во продолжение е даден максимално описен псевдокод на алгоритмот, кој детално го опишува процесот на пресметување на кластерите.

```
DBSCAN(D, Eps, MinPts)
    за секоја непосетена точка P во м-ство D
        означи ја P како посетена
        пресметај NeighborPts за P
        ако големината на NeighborPts е < MinPts
            означи го P како шум
        ако големината на NeighborPts е ≥ MinPts
            направи нов кластер C
            повикај го expandCluster

expandCluster(P, NeighborPts, C, Eps, MinPts)
    додади ја P во кластерот C
    за секоја точка P' во NeighborPts
        ако P' не е посетена
            означи ја P' како посетена
            пресметај NeighborPts за P'
            ако големината на NeighborPts' е ≥ MinPts
                додади го NeighborPts' на NeighborPts
        ако P' не е дел од кластер
            додади ја P' на кластерот C
```

4. КОМПЛЕКСНОСТ

DBSCAN ја посетува секоја точка од базата на податоци најмалку еднаш (ја посетува повеќе пати како кандидат за различни кластери). Практично гледано, временската комплексност е главно одредена од бројот на пресметувања на соседствата на точките во податочното множество. DBSCAN извршува една иста пресметка за секоја точка, и поради тоа, постојат методи кои можат значително да ја намалат комплексноста на алгоритмот.

Еден таков метод се R* дрва, структури кои се користат за индексирање на информации во просторни бази на податоци, со цел да ги оптимизираат просторните прашалници. Просторните бази на податоци се специјален тип на бази кои се оптимизирани за чување и влечење на податоци кои репрезентираат објекти дефинирани во геометриски простор.

Со помош на овој метод, соседството на една точка може да биде одредено со комплексност од O(log n), што ќе резултира со комплексност на алгоритмот од O(nlogn). Без користењето на ваква структура, комплексноста би била $O(n^2)$.

Често, матрицата на оддалеченост со големина $(n^2-n)/2$ се материјализира за да се избегнат повторни пресметувања на оддалеченостите. Овој метод сепак, побарува $O(n^2)$ меморија, додека имплементацијата без матрица побарува само O(n) меморија.

5. ПРОЦЕНКА НА ПАРАМЕТРИТЕ

За секоја задача во податочно рударење постои проблемот на одредување на параметрите. Секој параметар влијае на алгоритмот на различни начини. За DBSCAN, потребни се параметрите Eps и MinPts, кои мора да бидат дефинирани од корисникот.

Одредување на MinPts

MinPts може да се изведе од бројот на димензии во податочното множество, така што MinPts ≥ D+1. Доколку го сведеме проблемот на две димензии, вредноста MinPts=1 не би имала смисла, бидејќи во тој случај секоја точка ќе биде кластер. Со MinPts=2, резултатот ќе биде ист како резултатот од хиерархиско кластерирање во кое секоја точка во кластерите ќе биде поврзана со само една друга точка. Поголеми вредности на MinPts подобро ќе се справат со шум и ќе дефинираат поцврсти кластери. Исто така, вредноста на MinPts треба да скалира заедно со големината на податочното множество.

Одредување на Eps

Вредноста на Eps може да биде одредена со користење на k-distance график, кој би го претставил растојанието до k-тиот најблизок сосед за секоја точка, каде k би бил еднаков на MinPts. Добра вредност за Eps е вредноста каде овој график покажува силен екстрем. Доколку се одреди премногу мало Eps, голем дел од податоците нема да биде кластериран; додека ако се одреди премногу големо, кластерите ќе се спојат и поголемиот дел од објектите ќе припаѓаат на ист кластер.

6. ПРЕДНОСТИ И НЕДОСТАТОЦИ

Предности:

1. За разлика од k-means, DBSCAN не бара да се дефинира бројот на кластери a priori
2. DBSCAN може да открие кластери со произволна форма; дури и може да открие кластер комплетно опколен од друг кластер
3. Во DBSCAN е вграден концептот за шум, и е робустен во поглед на outlier-и
4. DBSCAN побарува само два параметри и е главно отпорен на различните подредувања на точките во базата на податоци
5. DBSCAN е дизајниран за користење со бази на податоци кои можат да ги забрзаат просторните прашалници со помош на R* дрво

Недостатоци:

1. DBSCAN не е целосно детерминистички – гранични точки кои се достигливи од повеќе од еден кластер може да бидат дел од кој било од тие кластери; DBSCAN* е варијација на алгоритмот кој ги третира граничните точки како шум, со што се постигнуваат целосно детерминистички резултати
2. Квалитетот на кластерирањето зависи од мерката за растојание помеѓу точките; најчеста метрика за растојание е Евклидово растојание
3. DBSCAN не е добар во кластерирањето на податочни множества со големи разлики во густините бидејќи комбинацијата од параметри не може да се одреди соодветно за сите кластери одеднаш

7. СПОРЕДБА

За да се добие претстава за ефикасноста на DBSCAN, неговите перформанси ќе бидат споредени со перформансите на друг алгоритам за кластерирање – CLARANS.

CLARANS (Clustering Large Applications based on RANdomized Search) е подобрување на алгоритмот k-medoid, кој за разлика од неговиот претходник работи многу поефикасно со поголеми бази (до 1000 објекти). Кога базата на податоци расте над оваа бројка, CLARANS започнува да заостанува во споредба со DBSCAN.

Сликите 6 и 7 ја покажуваат разликата во квалитетот на кластерирањето помеѓу DBSCAN и CLARANS (се користи истата база на податоци од слика 1).

СЛИКА 6: РЕЗУЛТАТОТ ОД КЛАСТЕРИРАЊЕТО СО CLARANS

СЛИКА 7: РЕЗУЛТАТОТ ОД КЛАСТЕРИРАЊЕТО СО DBSCAN

Освен супериорноста на DBSCAN во квалитетот на кластерирањето, тој е супериорен и во поглед на времето на извршување на двата алгоритми. Додека DBSCAN има скоро линеарно зголемување на времето на извршување релативно на бројот на објекти во базата, времето на извршување на CLARANS расте експоненцијално, што прави значително да заостанува за DBSCAN.

ЗАКЛУЧОК

Во овој труд детално беше презентиран алгоритмот за кластерирање DBSCAN и неговите предности и недостатоци.

DBSCAN побарува само еден влезен параметар и дава поддршка на корисникот за одредување на соодветна вредност за истиот. Исто така, резултатите од приложената споредба демонстрираат дека DBSCAN е значително поефикасен и побрз во одредувањето на кластери со произволна форма од алгоритмот CLARANS, кој всушност е подобрување на k-medoids.

Најголем недостаток на алгоритмот е неефикасноста во кластерирање податочни множества со големи разлики во густината.

Сепак, прицнипите кои стојат позади DBSCAN се исклучително важни, и се основа на многу други алгоритми. Меѓу нив се GDBSCAN, кој е генерализирана верзија на DBSCAN и OPTICS, кој иако е алгоритам за хиерархиско кластерирање, црпи од идеите на презентираниот алгоритам.

КОРИСТЕНА ЛИТЕРАТУРА

1. A Density-Based Algorithm for Discovering Clusters In Large Spatial Databases with Noise (Ester / Kriegel / Sander / Xu) - http://dns2.icar.cnr.it/manco/Teaching/2005/datamining/articoli/KDD-96.final.frame.pdf
2. DBSCAN – A Density-Based Spatial Clustering of Application with Noise (Backlund / Hedblom / Neijman) - http://staffwww.itn.liu.se/~aidvi/courses/06/dm/Seminars2011/DBSCAN(4).pdf
3. DBSCAN & Its Implementation on Atlas (Zhou / Luo / Zaniolo) - http://wis.cs.ucla.edu/wis/atlas/doc/dbscan.ppt
4. DBSCAN (Wikipedia) - http://en.wikipedia.org/wiki/DBSCAN
5. Spatial index (Wikipedia) - http://en.wikipedia.org/wiki/Spatial_index#Spatial_index
6. R* tree (Wikipedia) - http://en.wikipedia.org/wiki/R*_tree

Скопје – Април, 2014