ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH

KHOA ĐÀO TẠO CHẤT LƯỢNG CAO

NGÀNH CÔNG NGHỆ THÔNG TIN

HCMUTE

BÁO CÁO ĐỒ ÁN 2

MACHINE LEARNING

TÌM HIỂU VỀ THUẬT TOÁN OPTICS CLUSTERING

GVHD: ***Thầy Trần Nhật Quang***

Nhóm thực hiện: **16**

ĐOÀN TIẾN PHÁT **16110177**

HUỲNH TRẦN PHƯỚC HƯNG **16110**

Tp.HCM, ngày 23 tháng 5 năm 2019

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Giáo viên hướng dẫn

(ký và ghi họ tên)

……....

MỤC LỤC

MỤC LỤC ẢNH

LỜI CẢM ƠN

Để hoàn thành đồ án 2. Nhóm em xin tỏ lòng biết ơn sâu sắc đến Trần Nhật Quang đã tận tình hướng dẫn nhóm em trong suốt quá trình vừa qua.

Nhóm em xin chân thành cảm ơn Thầy đã tận tình truyền đạt kiến thức, tạo mọi điều kiện tốt nhất cho em học tập. Với vốn kiến thức được tiếp thu trong quá trình học không chỉ là nền tảng cho quá trình học tập sau này mà còn là hành trang quý báu để em bước vào đời một cách vững chắc và tự tin.

Cuối cùng em kính chúc Thầy dồi dào sức khỏe và thành công trong sự nghiệp cao quý, nhiều thành công tốt đẹp trong công việc.

Thành phố Hồ Chí Minh, ngày 23 tháng 5 năm 2019

LỜI NÓI ĐẦU

Trong 3 – 4 năm trở lại đây, AI – Artificial Intelligent (Trí tuệ nhân tạo), cụ thể hơn là ML – Machine learning (Học máy hoặc máy học) đang phát triển mạnh mẽ và được sự quan tâm đặc biệt từ cộng đồng học thuật cũng như từ xã hội. Là trái tim của cuộc Cách mạng công nghiệp lần thứ 4, trí tuệ nhân tạo đang len lõi vào mọi lĩnh vực trong cuộc sống hằng ngày của chúng ta mà có thể chúng ta không nhận ra. Ví dụ như: hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý quảng cáo từ Facebook, khả năng tự động tag khuôn mặt trong ảnh của Facebook, tính năng tự động phân loại ảnh theo Album của Google photo và nổi bật hơn cả không thể không kể đến xe tự lái của Google, Tesla; máy chơi cờ vây AlphaGo của Google DeepMind, . . .

Hiểu được tầm quan trọng của Machine learning hiện nay, chúng em quyết định tìm hiểu về thuật toán OPTICS CLUSTERING một thuật toán phân cụm dữ liệu dự trên mật độ. Trong quá trình nghiên cứu và hoàn thành vẫn còn rất nhiều thiếu sót chúng em mong nhận được sự góp ý của thầy để bài báo cáo này được hoàn chỉnh hơn.

TÌM HIỂU VỀ ĐẠO VĂN

Ở Việt Nam, so với nước ngoài như Mỹ, Anh, Nhật. Vấn đề đạo văn đang trở nên ngày càng vấn đề "nóng". Một số người vẫn thờ ơ, mặc nhiên chia sẽ những thành quả, tác phẩm của mình với mục đích tham khảo. Có thể "mượn" thậm chí là "xin luôn" một cách miễn phí. Điều này đã góp phần lớn hình thành nên thói quen đạo văn ở hầu hết các tầng lớp trong xã hội.

Việc đạo văn không phải đơn thần là đánh cắp ý tưởng. Mà nó còn ảnh hưởng đến tư duy sáng tạo phát triển của cả một xã hội. Nếu một bài toán có sẵn lời giải sẽ thì ta không cần phải suy nghĩ nữa...Như thế ảnh hưởng rất nhiều về phát triển tư duy đặc biệt là tầng lớp đang phát triển: học sinh, sinh viên.

Như vậy chúng ta phải làm thế nào để tránh việc đạo văn?

Để tránh việc đạo văn bạn nên tuân thủ các nguyên tắc và quy định sau:

- ✓ Khi mượn từ ngữ của người khác, dùng dấu ngoặc kép và ghi rõ đầy đủ thông tin nguồn (tên tác giả, ngày tháng, số trang)
- ✓ Nguồn từ internet cũng phải được nêu rõ.
- ✓ Khi mượn ý tưởng của người khác, phải nêu rõ nguồn gốc ý tưởng đó.
- ✓ Không diễn giải đoạn văn của người khác và truyền đạt như nó là của chính bạn. - Khi ghi nguồn nên tuân thủ theo các kiểu ghi nguồn tham khảo phổ biến:
- ✓ Dẫn nguồn tham khảo kiểu APA (American Psychological Association)
- ✓ Dẫn nguồn tham khảo kiểu Chicago
- ✓ Dẫn nguồn tham khảo kiểu MLA References (Modern Language Association)

Khi nào nên và không nên trích dẫn tài liệu?

- Nhà nghiên cứu nên trích dẫn khi:

- ✓ Bảo vệ quan điểm, luận cứ khoa học (trích dẫn để củng cố cho luận điểm của mình)
- ✓ Nêu ví dụ, dẫn kết quả, số liệu, hình ảnh đã được công bố, kiểm chứng hay thừa nhận
- ✓ Tóm tắt ý kiến, giả thuyết, kết luận của người khác

- Nhà nghiên cứu không nên trích dẫn:

- ✓ Những chi tiết nhỏ nhặt
- ✓ Nguyên văn các đoạn dài
- ✓ Những ý có thể tự diễn đạt
- ✓ Những kinh nghiệm, ghi nhận, ý kiến của bản thân (trừ khi dẫn từ các tài liệu đã công bố)
- ✓ Những kiến thức đã trở nên phổ thông

Việc nhận ra kiến thức nào là phổ thông, kiến thức nào là mới, đòi hỏi người nghiên cứu phải có kiến thức tích lũy khá tốt. Một kiến thức đôi khi là mới, cần trích dẫn đối với người mới vào nghề nhưng lại là kiến thức phổ thông với người có chuyên môn. Tuy nhiên, việc xem một kiến thức nào đó đã trở thành phổ thông và kiến thức nào còn mới, nhất là đối với các ngành khoa học xã hội là một ranh giới mập mờ. Vậy nên, các bạn sinh viên nên trích dẫn nguồn nếu đã có tham khảo và đưa vào bài viết những ý tưởng của người khác.

"Chúng tôi xin cam đoan đồ án này do chính chúng tôi thực hiện. Chúng tôi không sao chép, không sử dụng bất kì tài liệu, mã nguồn của người khác mà không ghi rõ nguồn góc. Chúng tôi xin chịu hoàn toàn trách nhiệm nếu vi phạm"

Ký tên

Đoàn Tiến Phát

Huỳnh Trần Phước Hưng

NỘI DUNG

CHƯƠNG 1 GIỚI THIỆU VỀ THƯ VIỆN ĐÃ DÙNG

1. Thư viện Numpy

1.1 Khái niệm

- NumPy là một từ viết tắt của "Numeric Python" hoặc "Numerical Python". Nó là một mô-đun mở rộng mã nguồn mở cho Python, cung cấp các chức năng biên dịch nhanh cho các thao tác toán học và số. Hơn nữa, NumPy làm phong phú ngôn ngữ lập trình Python với các cấu trúc dữ liệu mạnh mẽ để tính toán hiệu quả các mảng và ma trận đa chiều. Việc thực hiện thậm chí là nhằm vào ma trận và mảng khổng lồ. Bên cạnh đó các mô-đun cung cấp một thư viện lớn các chức năng toán học cấp cao để hoạt động trên các ma trận và mảng.
- Numpy là một package chủ yếu cho việc tính toán khoa học trên Python. Vì Numpy hỗ trợ mạnh mẽ việc tính toán với matrix, vector và các các hàm đại số tuyến tính cơ bản nên nó được sử dụng nhiều trong việc implement các thuật toán Machine Learning.
- Numpy bao gồm:
    - một đối tượng mảng N-chiều mạnh mẽ
    - các hàm phức tạp/tinh vi (sophisticated) nhưng tổng quát/đa dụng (broadcasting/universal)
    - các công cụ để tích hợp code C/C++ hay Fortran
    - có khả năng điện sốngẫu nhiên (random number), biến đổi Fourier và các phép đại số tuyến tính hữu ích
- Bên cạnh các công dụng khoa học rõ ràng, NumPy còn dùng như một container đã chiều để chứa các dữ liệu tổng quát. Kiểu dữ liệu tuỳ biến (arbitrary data-types) có thể được định nghĩa. Điều này cho phép NumPy tích hợp một cách liên tục và nhanh chóng một loạt các CSDL khác nhau.

1.2 Ưu điểm của việc sử dụng Numpy với Python:

- Tính toán theo mảng
- Triển khai hiệu quả các mảng đa chiều
- Được thiết kế để tính toán khoa học

Một số thuộc tính của ndarray

- Numpy array được gọi là ndarray, n-dimensional array, để phân biệt với array cơ bản của python, chỉ có 01 chiều.
- ndarray. ndim
  - số chiều của array
- ndarray. shape
  - chiều của array, hay đúng hơn là một tuple mô tả chiều của array.
  - ví dụ 01 matrix có *n* rows và *m* columns, shape sẽ là (n, m).
  - độ dài của shape do đó chính là rank (ndim).
- ndarray. size
  - tổng phần tử trong array.
- ndarray. dtype
  - trả về 01 object miêu tả kiểu dữ liệu của các phần tử trong array.
  - dtype gồm các kiểu tiêu chuẩn của Python công thêm các kiểu riêng của NumPy như: numpy.int32, numpy.int16, and numpy. float64…
  - mình nghĩ là nên xài kiểu của NumPy, kiểu gì thì SciPy cũng sẽ tối ưu tính toán hơn khi dùng các kiểu dữ liệu riêng do chính họ tạo ra. Vì họ work nhiều trên chúng mà.
- ndarray. itemsize
  - kích thước tính bằng byte của mỗi phần tử trong mảng
  - ví dụ, một array có dtype là float64 sẽ có itemsize 8 (=64/8)
- ndarray.data
  - bộ đệm chứa các phần tử của mảng.

- thường thì ta sẽ không dùng thuộc tính này, vì việc truy xuất mảng thông qua index sẽ tiện lợi hơn

2. Thư viện Scikit-Learn

2.1. Lịch sử ra đời

- Scikit-learn ban đầu được phát triển bởi David Cournapeau như một dự án mã mùa hè của Google vào năm 2007.
- Sau đó Matthieu Brucher tham gia dự án và bắt đầu sử dụng nó như là một phần của công việc luận án của mình. Vào năm 2010, INRIA đã tham gia và phiên bản công khai đầu tiên (v0.1 beta) đã được xuất bản vào cuối tháng 1 năm 2010.
- Dự án hiện có hơn 30 người đóng góp tích cực và đã được tài trợ từ INRIA , Google, Tinyclues và Python Software Foundation .

2.2. Khái niệm

- Scikit-learn cung cấp một loạt các thuật toán học tập có giám sát và không giám sát thông qua một giao diện nhất quán trong Python.
- Nó được cấp phép theo giấy phép BSD đơn giản hóa cho phép và được phân phối theo nhiều bản phân phối Linux, khuyến khích sử dụng học thuật và thương mại.

2.3. Bao gồm

Thư viện Scikit Learn bao gồm các thư viện sau:

- NumPy: Gói mảng n chiều cơ sở
- SciPy: Thư viện cơ bản cho máy tính khoa học
- Matplotlib: Âm mưu 2D / 3D toàn diện
- IPython: Bảng điều khiển tương tác nâng cao
- Sympy: Toán học tượng trưng
- Pandas: Cấu trúc dữ liệu và phân tích

2.4. Các mô hình được cung cấp bởi Scikit-Learn

- Clustering: để nhóm các dữ liệu không được gắn nhãn như KMeans.
- Cross Validation: để ước tính hiệu suất của các mô hình được giám sát trên dữ liệu chưa xem.
- Datasets: cho bộ dữ liệu thử nghiệm và để tạo bộ dữ liệu với các thuộc tính cụ thể để điều tra hành vi mô hình.
- Dimensionality Reduction: để giảm số lượng thuộc tính trong dữ liệu để tóm tắt, trực quan hóa và lựa chọn tính năng, chẳng hạn như phân tích thành phần chính.
- Ensemble methods: để kết hợp các dự đoán của nhiều mô hình được giám sát.
- Feature extraction: để xác định các thuộc tính trong dữ liệu hình ảnh và văn bản.
- Feature selection: để xác định các thuộc tính có ý nghĩa từ đó tạo các mô hình được giám sát.
- Parameter Tuning: để tận dụng tối đa các mô hình được giám sát.
- Manifold Learning: Để tóm tắt và mô tả dữ liệu đa chiều phức tạp.
- Supervised Models: một mảng rộng lớn không giới hạn ở mô hình tuyến tính tổng quát, phân tích phân biệt, vịnh ngây thơ, phương pháp lười biếng, mạng lưới thần kinh, máy vectơ hỗ trợ và cây quyết định.

CHƯƠNG 2 TÌM HIỂU VỀ UNSUPERVISED LEARNING

Unsuppervised Learning (Học không có giám sát) là một thuật toán chúng ta không biết được *outcome* hay *nhãn* mà chỉ có dữ liệu đầu vào. Thuật toán Unsupervised Learning sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân nhóm (clustering) hoặc giảm số chiều của dữ liệu (dimension reduction) để thuận tiện trong việc lưu trữ và tính toán.

Hay nói một cách khác, Unsupervised Learning là khi chúng ta chỉ có dữ liệu vào X mà không biết *nhãn* Y tương ứng. Những thuật toán loại này được gọi là Unsupervised Learning vì không giống như Supervised learning, chúng ta không biết câu trả lời chính xác cho mỗi dữ liệu đầu vào. Giống như khi ta học, không có thầy cô giáo nào chỉ cho ta biết đó là chữ A hay chữ B mà tự chúng ta tìm cách nào đó phân ra được đó là A và B.

Các bài toán Unsupervised learning được tiếp tục chia nhỏ thành hai loại: Clustering (phân nhóm), Association.

1. Giới thiệu chung về Clustering (phân nhóm)

Một bài toán phân nhóm toàn bộ dữ liệu X thành các nhóm nhỏ dựa trên sự liên quan giữa các dữ liệu trong mỗi nhóm. Ví dụ: phân nhóm khách hàng dựa trên hành vi mua hàng. Điều này cũng giống như việc ta đưa cho một đứa trẻ rất nhiều mảnh ghép với các hình thù và màu sắc khác nhau, ví dụ tam giác, vuông, tròn với màu xanh và đỏ, sau đó yêu cầu trẻ phân chúng thành từng nhóm. Mặc dù không cho trẻ biết mảnh nào tương ứng với hình nào hoặc màu nào, nhiều khả năng chúng vẫn có thể phân loại các mảnh ghép theo màu hoặc hình dạng.

Và các bài toán Clustering tiếp tục chia nhỏ ra thành các loại sau:

+ K-mean: thuật đoán điển hình cho bài toán Clustering
+ Hierarchical
+ DBSCAN
+ OPTICS

2. Giới thiệu chung về Association

Là bài toán khi chúng ta muốn khám phá ra một quy luật dựa trên nhiều dữ liệu cho trước. Ví dụ: những khách hàng nam mua quần áo thường có xu hướng mua thêm đồng hồ hoặc thắt lưng; những khán giả xem phim Spider Man thường có xu hướng xem thêm phim Bat Man, dựa vào đó tạo ra một hệ thống gợi ý khách hàng (Recommendation System), thúc đẩy nhu cầu mua sắm.

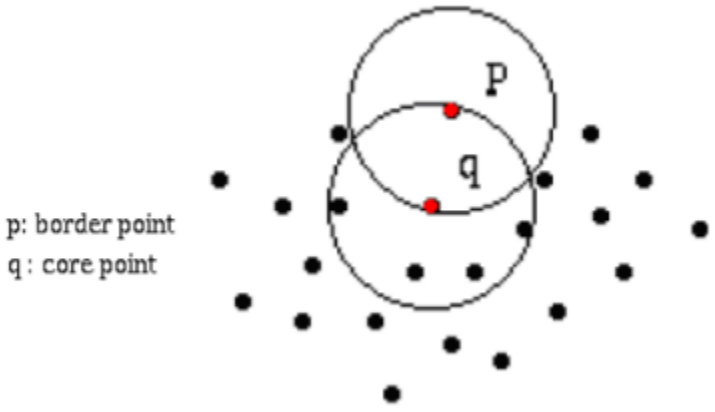
CHƯƠNG 3: TÌM HIỂU VỀ THUẬT TOÁN DBSCAN (CLUSTERNING)

1. Giới thiệu thuật toán

Thuật toán DBSCAN ((Density Based Spatial Clustering of Applications with Noise) do Martin Ester và các tác giả khác đề xuất là thuật toán gom tụm dựa trên mật độ, hiệu quả với cơ sở dữ liệu lớn, có khả năng xử lý nhiễu.

Ý tưởng chính của thuật toán là vùng lân cận mỗi đối tượng trong một cụm có số đối tượng lớn hơn ngưỡng tối thiểu. Hình dạng vùng lân cận phụ thuộc vào hàm khoảng cách giữa các đối tượng (nếu sử dụng khoảng cách Manhattan trong không gian 2 chiều thì vùng lân cận có hình chữ nhật, nếu sử dụng khoảng cách Eucler trong không gian 2 chiều thì vùng lân cận có hình tròn).

Các đối tượng trong mỗi cụm được phân làm 3 loại: đối tượng bên trong cụm (core point: đối tượng lõi), đối tượng nằm trên đường biên của cụm (border point: đối tượng biên) và đối tượng nhiễu (noise)

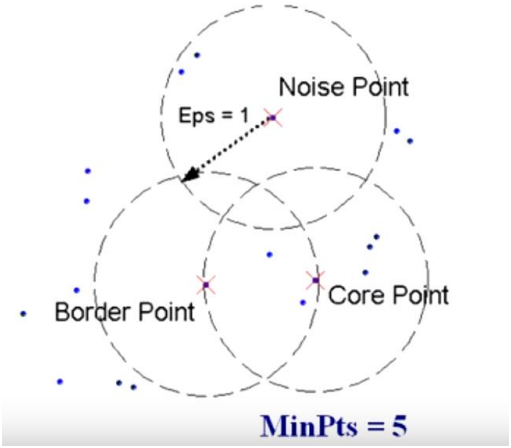


Hình 1 Ảnh DBSCAN

2. Một số định nghĩa của thuật toán

2.1 Khái niệm cơ bản

- Eps: Là bán kính lớn nhất của vùng lân cận
- MinPts: là số đối tượng nhỏ nhất trong vùng lân cận bán kính Eps
- Mật độ là số đối tượng nằm trong vùng lân cận bán kính Eps



Hình 2. Core-Point và Border-Point

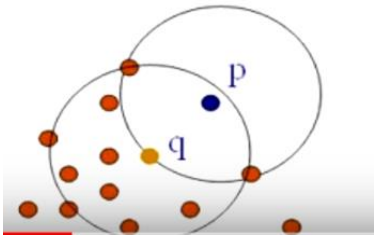
- Đối tượng được gọi là đối tượng nòng cốt (core point) khi số đối tượng lân cận trong bán kính Eps lớn hơn số đối tượng MinEts.
    - ⇨ Như vậy hình trên đối tượng nòng cốt (core point) vì số đối tượng lân cận trong khoảng Eps lớn hơn MinEts.
- Đối tượng được gọi là đối tượng biên (Border Point) khi số đối tượng nằm trong vùng lân cận ít hơn số đối tượng MinEts nhưng vẫn nòng trong đối tượng nong cốt.
    - ⇨ Như vậy trên hình đối tượng biên (border point) khi có số tượng nằm trong vùng lân cận nhưng có số đối tượng ít hơn MinEts trong khoảng Eps
- Đối tượng được gọi là đối tượng nhiễu (Noise Point) không thuộc dạng 2 đối tượng trên. [2]

2.2 Định nghĩa 1 (Mật độ đạt được trực tiếp/ directly density_reachable)

Một đôi tượng p là đối tượng có mật độ trực tiếp từ một đối tượng q theo Eps và MinPts nếu thõa mãn điều kiện sau:

+ p ∈ $N_{Eps}(q)$

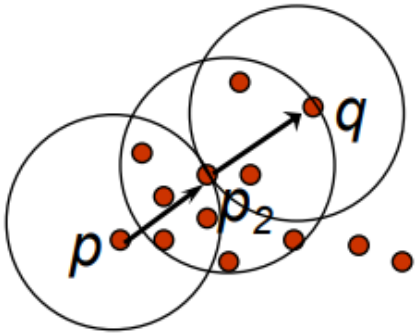
+ Và q phải là một đối tượng nòng cốt



Hình 3

2.3 Định nghĩa 2 (Mật độ có thể đạt được /density_reachable)

Một đối tượng p là một đối tượng có thể đạt được từ một đối tượng q theo MinPts và Eps nếu tồn tại một dây chuyền đối tượng p1, p2, ….pn với p1=q và pn=p sao cho p_{i+1} có mật độ có thể đạt được trực tiếp từ pi.



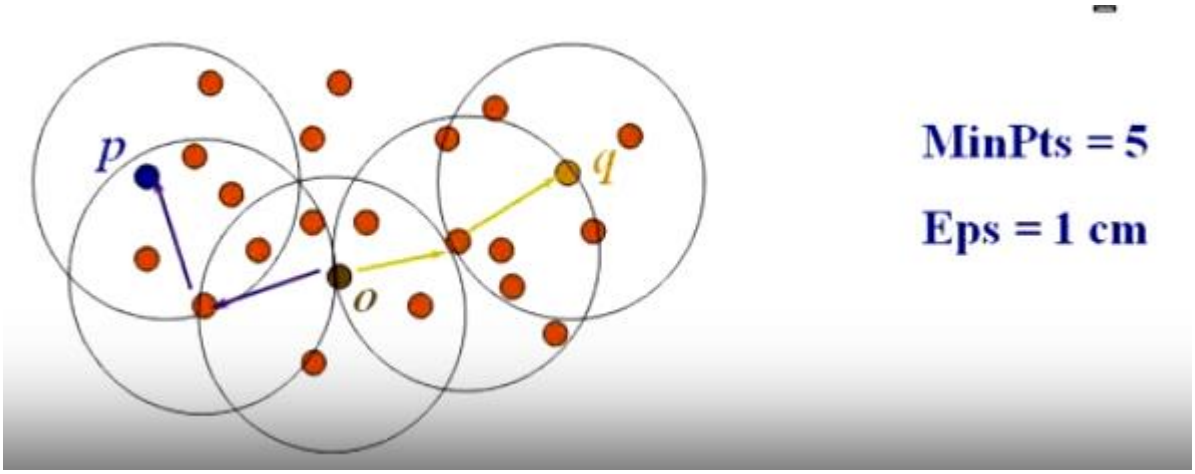
Hình 4 Định nghĩ 2 DBSCAN

- Ta thấy p_2 thuộc vào đối tượng nòng cốt p và p2 cũng là một tượng nồng cốt nhưng q thuộc vào đối tượng trực tiếp của p_2.

  ⇨ Như vậy ta có thể nói q có mật độ có thể đạt được từ p

2.4 Định nghĩ 3 (Mật độ liên thông/density_connecting)

Một đối tượng q được gọi là đối mật độ liên thông từ một đối tượng q theo MinEts và Eps nếu tồn tại một đối tượng o sao cho cả q và p đều là đối tượng có thể đạt được từ o theo Eps và MinPts.



Hình 5. Định nghĩa 3 DBSCAN

- Ta thấy p và q không thể có mật độ trực tiếp và cũng không có mật độ có thể đạt được.
- Nhưng p và q đều có mật độ có thể đạt được từ o nên có thể nói p và q có mật độ liên thông qua o.

3. Thuật toán DBSCAN

B1: Tìm một đối tượng ngẫu nhiên p

B2: Tìm tất cả các đối tượng có mật độ có thể đạt được từ p theo Eps và MinEts

B3: Nếu p là đối tượng nòng cốt thì nó hình thành 1 nhóm

+ Nếu p là một đối tượng biên (không có đối tượng nào có mật độ đạt được từ p) thì DBSCAN sẽ xem xét đối tượng tiếp theo trong data.

B4: Tiếp tục cho đến khi tất cả đối tượng đều được xử lý.

4. Nhận xét về thuật toán DBSCAN

4.1 Ưu điểm

- Làm việc tốt từ dữ liệu nhiễu
- Có thể giải quyết dữ liệu hình dáng và kích thước bất kì

4.2 Nhược điểm

- Độ phức tạp thuật toán cao
- Dữ liệu mật độ dày đặt lại trở nên ko chính xác và rõ ràng

- Phụ thuộc hoàn toàn và giá trị Eps và MinPts

CHƯƠNG 4: TÌM HIỂU THUẬT TOÁN OPTICS THÔNG QUA THUẬT TOÁN DBSCAN

1. Giới thiệu về thuật toán

Thuật toán OPTICS *(Ordering Points To Identify the Clustering Structure)* do Ankerst, Breunig, Kriegel và Sander đề xuất năm 1999, là thuật toán mở rộng cho thuật toán DBSCAN, bằng cách giảm bớt các tham số đầu vào.

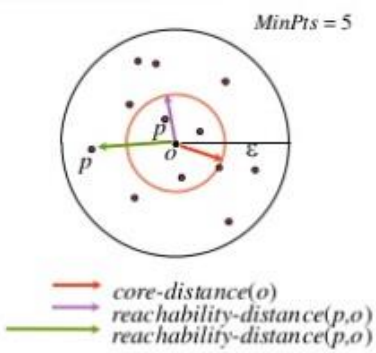
Thuật toán thực hiện tính toán và sắp xếp các đối tượng theo thứ tự tăng dần nhằm tự động phân cụm và phân tích cụm tương tác hơn là đưa ra phân cụm một tập dữ liệu rõ ràng. Thứ tự này diễn tả cấu trúc dữ liệu phân cụm dựa trên mật độ chứa thông tin tương đương với phân cụm dựa trên mật độ với một dãy các tham số đầu vào.

Thuật toán OPTICS xem xét bán kính tối thiểu nhằm xác định các láng giềng phù hợp với thuật toán. Thuật toán DBSCAN và OPTICS tương tự với nhau về cấu trúc và có cùng độ phức tạp: *O(nLogn)* (n là kích thước của tập dữ liệu). [3]

2. Một số định nghĩa của thuật toán

Do thuật toán DBSCAN và OPTICS tương tự với nhau về cấu trúc nên chúng em đã trình bày sơ lược ở phía trên và thêm một số định nghĩa cho thuật toán OPTICS

2.1 Định nghĩa 1 (Core_distane)



$$\text{core-dist}_{\varepsilon,MinPts}(o) = \begin{cases} \infty & \text{if} \quad |\text{rangeQuery}(o,\varepsilon)| < MinPts \\ MinPts - \text{dist}(o) & \text{otherwise} \end{cases}$$

Hình 6. Định nghĩa 1 OPTICS

- Là khoảng cách từ đối tượng gần nhất thõa mãn điều kiện số đối tượng vùng lân cận của o bằng với MinPts.

2.2 **Định nghĩa 2 (*Reachability Distance*)**

$$\text{reachability-dist}_{\varepsilon,MinPts}(o,p) = \begin{cases} \text{UNDEFINED} & \text{if } |N_\varepsilon(p)| < MinPts \\ \max(\text{core-dist}_{\varepsilon,MinPts}(p), \text{dist}(p,o)) & \text{otherwise} \end{cases}$$

Hình 7. Định nghĩa 2 OPTICS

- Khoảng cách tiếp cận o, p:
  + Bằng khoảng cách core-dist (o) khi p thuộc trong khoảng cách từ o đến core-dist (o)
  + Và bằng khoảng cách của từ nó đến o khi nó ko nằm trong tập core-dist (o) của O.

3. Thuật toán OPTICS

- Xử lý theo thứ tự của tất cả các đối tượng trong data
  + Lưu trữ lại core-distance và các reachability-distance của từng đố tượng này vào trong một list OrderSeeds.
- List OrderSeeds:
  + Được sắp xếp theo reachability-distance từ các đối tượng đến đối tượng nòng cốt (core-point)
  +Nghĩa là khoảng cách của một đối tượng nào đó gần với đối tượng nồng cốt nhất.

B1: Tìm một đối tượng ngẫu nhiên p

B2: Lấy ε đi xác định core-distance và reachability-distance có xác định hay không? (hay nói các khác kiểm tra xem p này có phải core-point hay không)

B3: Nếu p là một đối tượng nòng cốt(core-object)

+ Sau đó cho từng đối tượng q trong vùng lân cận của p

+ OPTICS cập nhật reachability-distance của nó từ p

+ Và chèn q vào list OrderSeed nếu q chưa được xử lý

Nếu p không phải là một đối tượng nồng code (core-object)

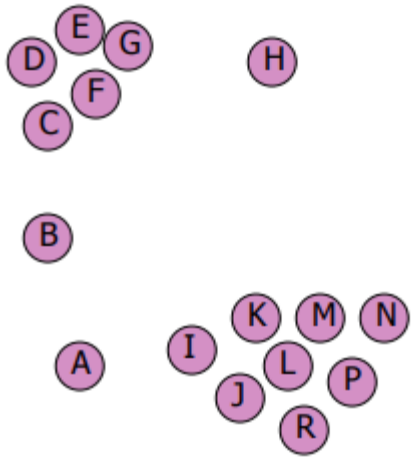
+ p chỉ cần chuyển sang đối tượng tiếp theo trong danh sách OrderSeed

+ Nếu p là một đối tượng biên (danh sách OrderSeed trống) thì p sẽ xem xét đối tượng tiếp theo trong data.

B4: Lặp lại cho đến khi các đối tượng điều được xử lý và danh sách OrderSeed nó trống.

4. Bài Toán Ví Dụ Cho Thuật Toán Optics

Giả sử ta có bộ dữ liệu gồm có 16 điểm như sau: [1]



$\varepsilon = 44, \quad MinPts = 3$

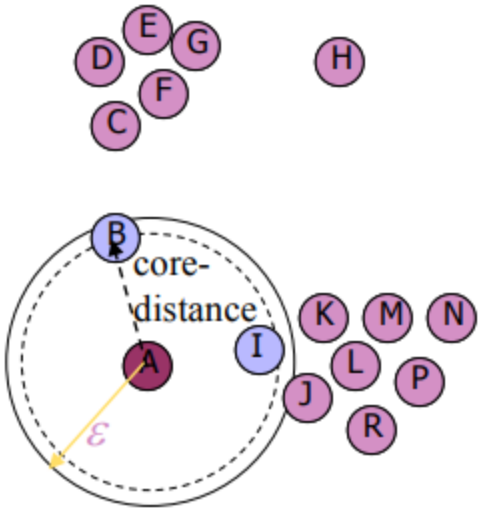
Hình 8. Ví dụ thuật toán OPTICS

- Giải sử chọn ngẫu nhiên một đối tượng.

+ Chọn A đối tượng được chọn ngẫu nhiên:

+ A là đối tượng core-object

+ Và Core-distance của B, A và Core-distance I, A bằng nhau.

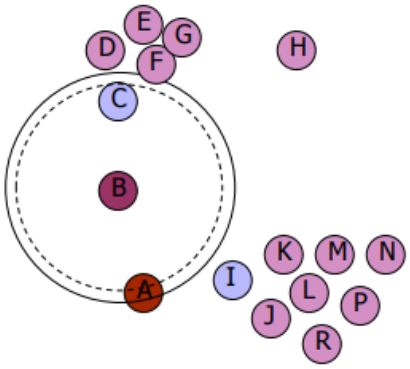


Hình 9. Ví dụ thuật toán OPTICS

+ Lưu core-distance vào OrderSeed:

OrderSeed: B(A,40), C(A,40)

+ Ta chọn B là đối tượng tiếp theo



+ Chọn B:

+ B là đối tượng core-object.

+ Core-distance của B, A và Core-distance I, A bằng nhau.

+ Lưu core-distance vào OrderSeed:

OrderSeed: I(A,40), B(A,40)

⇨ C và I thuộc hai nhóm khác nhau.

Hình 10. Ví dụ thuật toán OPTICS

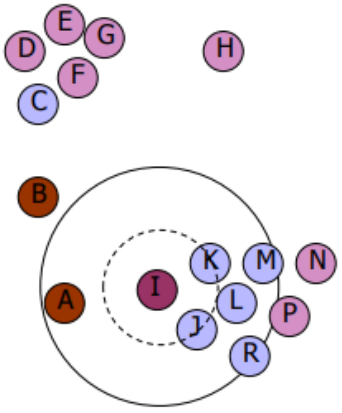
+ Chọn I:

+ I là một đối tượng core-object.

+ Tính core-distance từ các đối tượng trong vùng lân cận đến I.

+ Lưu core-distance vào OrderSeed:

OrderSeed: (J, 22) (K, 22) (L, 31) (C, 40) (M, 40) (R, 43).



Hình 11. Ví dụ thuật toán OPTICS

+ Chọn J:

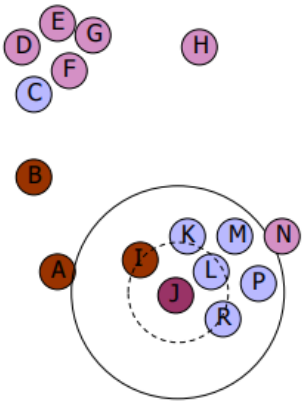
+ J là một đối tượng core-object.

+ Tính core-distance từ các đối tượng trong vùng lân cận đến J.

+ Lưu core-distance vào OrderSeed:

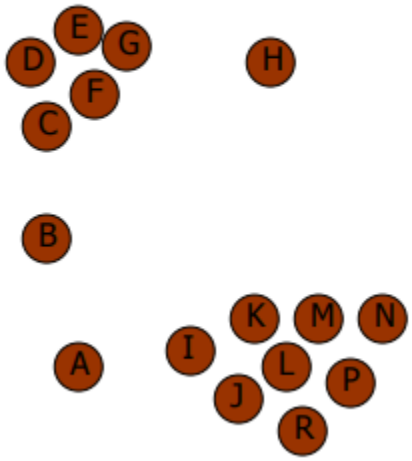
OrderSeed: (L, 19) (K, 20) (R, 21) (M, 30) (P, 31) (C, 40)

⋮



Hình 12. Ví dụ thuật toán OPTICS

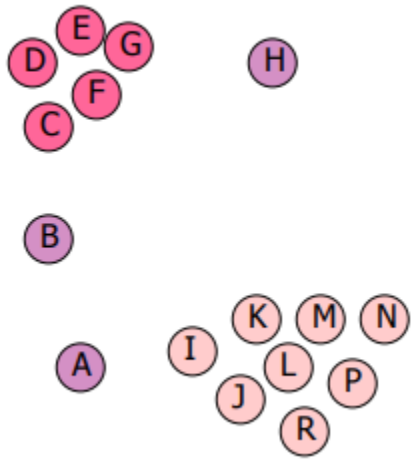
+ Sau khi duyệt hết tất cả các đối tượng và list OrderSeed rỗng thì thuật toán dừng lại



Hình 13. Ví dụ thuật toán OPTICS

- Ta thu được kết quả sau:

+Gồm có 2 nhóm và 3 điểm nhiễu



Hình 15. Ví dụ thuật toán OPTICS

5. Áp Dụng Thuật Toán Optics Trên Python Dùng Thư Viện Sklearn.

+ Các thư viện sử dụng:

```
from sklearn.cluster import OPTICS, cluster_optics_dbscan
import matplotlib.gridspec as gridspec
import matplotlib.pyplot as plt
import numpy as np
```

Hình 16. Thư viện Python

+ Khởi tạo dữ liệu cho thuật toán:

```
np.random.seed(0) # Random với giá trị khởi tạo ở phía trước
n_points_per_cluster = 250

C1 = [-5, -2] + .8 * np.random.randn(n_points_per_cluster, 2)
C2 = [4, -1] + .1 * np.random.randn(n_points_per_cluster, 2)
C3 = [1, -2] + .2 * np.random.randn(n_points_per_cluster, 2)
C4 = [-2, 3] + .3 * np.random.randn(n_points_per_cluster, 2)
C5 = [3, -2] + 1.6 * np.random.randn(n_points_per_cluster, 2)
C6 = [5, 6] + 2 * np.random.randn(n_points_per_cluster, 2)
X = np.vstack((C1, C2, C3, C4, C5, C6))  #Theo thứ tự theo chiều dọc
```

Hình 17. Tạo data cho thuật toán

+ Định nghĩa hàm và run.

```
clust = OPTICS(min_samples=50, xi=.05, min_cluster_size=.05) #Truyền tham số có hàm
# OPTICS với MinPts:50 , e=0,05


# Run the fit
clust.fit(X)    # Run OPTICS

labels_050 = cluster_optics_dbscan(reachability=clust.reachability_,
                                   core_distances=clust.core_distances_,
                                   ordering=clust.ordering_, eps=0.5)#RUN DBSCAN VỚI EPS=0,5
labels_200 = cluster_optics_dbscan(reachability=clust.reachability_,
                                   core_distances=clust.core_distances_,
                                   ordering=clust.ordering_, eps=2)#RUN DBSCAN VỚI EPS=2
```

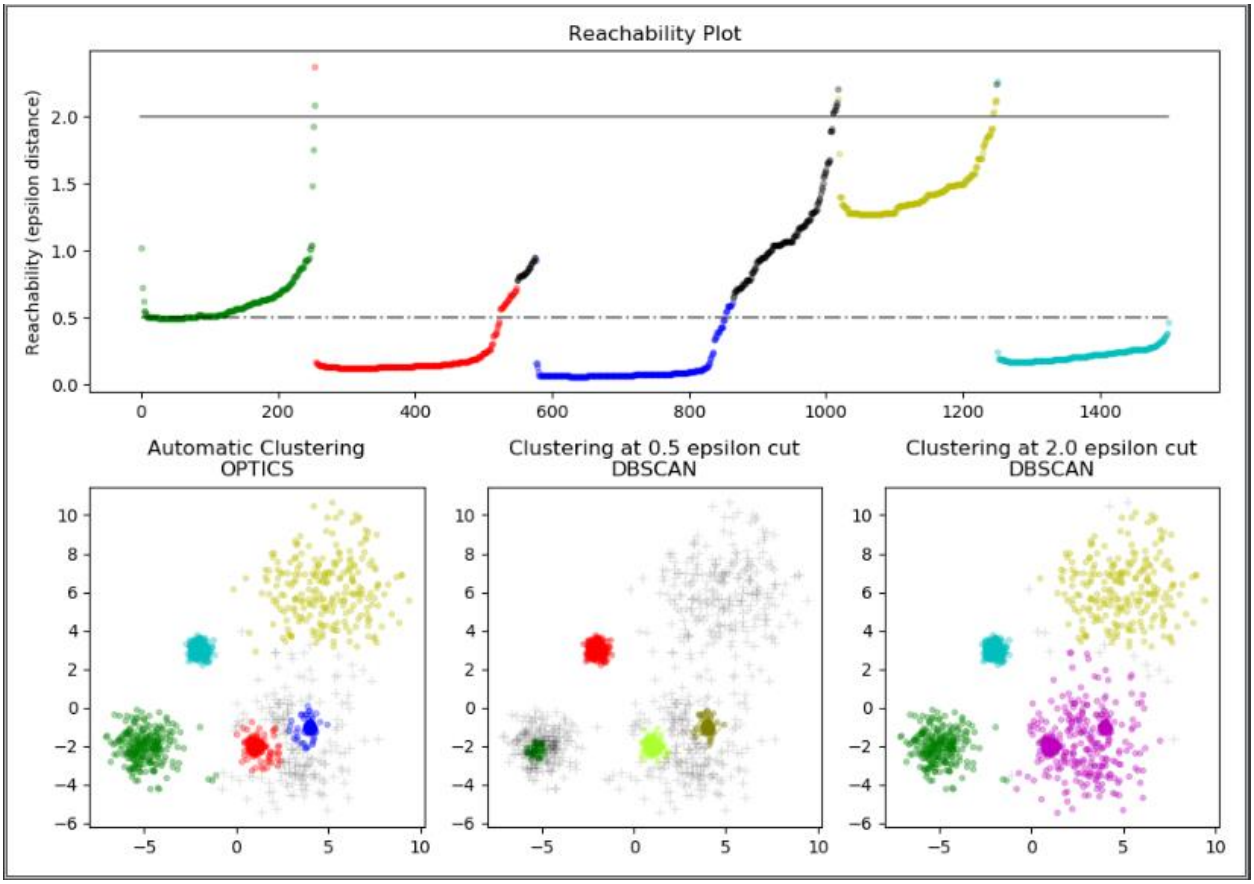
Hình 18. Code hàm và Traning thuật toán

+ Vẽ trên matplotlib

```
# Reachability plot  # Đồ thị biểu diễn cho phân nhóm của thuật toán sklearn
colors = ['g.', 'r.', 'b.', 'y.', 'c.']
for klass, color in zip(range(0, 5), colors):
    Xk = space[labels == klass]
    Rk = reachability[labels == klass]
    ax1.plot(Xk, Rk, color, alpha=0.3)
ax1.plot(space[labels == -1], reachability[labels == -1], 'k.', alpha=0.3)
ax1.plot(space, np.full_like(space, 2., dtype=float), 'k-', alpha=0.5)
ax1.plot(space, np.full_like(space, 0.5, dtype=float), 'k-.', alpha=0.5)
ax1.set_ylabel('Reachability (epsilon distance)')
ax1.set_title('Reachability Plot')


# OPTICS
colors = ['g.', 'r.', 'b.', 'y.', 'c.']
for klass, color in zip(range(0, 5), colors):
    Xk = X[clust.labels_ == klass]
    ax2.plot(Xk[:, 0], Xk[:, 1], color, alpha=0.3)
ax2.plot(X[clust.labels_ == -1, 0], X[clust.labels_ == -1, 1], 'k+', alpha=0.1)
ax2.set_title('Automatic Clustering\nOPTICS')

# DBSCAN at 0.5
colors = ['g', 'greenyellow', 'olive', 'r', 'b', 'c']
for klass, color in zip(range(0, 6), colors):
    Xk = X[labels_050 == klass]
    ax3.plot(Xk[:, 0], Xk[:, 1], color, alpha=0.3, marker='.')
ax3.plot(X[labels_050 == -1, 0], X[labels_050 == -1, 1], 'k+', alpha=0.1)
ax3.set_title('Clustering at 0.5 epsilon cut\nDBSCAN')

# DBSCAN at 2.
colors = ['g.', 'm.', 'y.', 'c.']
for klass, color in zip(range(0, 4), colors):
    Xk = X[labels_200 == klass]
    ax4.plot(Xk[:, 0], Xk[:, 1], color, alpha=0.3)
ax4.plot(X[labels_200 == -1, 0], X[labels_200 == -1, 1], 'k+', alpha=0.1)
ax4.set_title('Clustering at 2.0 epsilon cut\nDBSCAN')

plt.tight_layout()
plt.show()
```

Hình 19. Vẽ trên Matplotlib

+ Kết quả thu được:



Hình 20. Kết quả thu được

⇨ Nhận xét:

+ Thuật toán OPTICS hoạt động tốt đối với dữ liệu có mật độ dày đặc

+ Cùng với một dữ liệu thuật toán OPTICS đưa ra số phân nhóm dữ liệu rõ ràng hơn thuật toán DBCAN. [5]

KẾT LUẬN

Sau tìm hiểu thuật toán OPTICS CLUSTERING chúng em biết thêm được một thuật toán phân nhóm theo mật độ dữ liệu một thuật toán thuộc nhóm Unsuppervised (Học không dám sát). Tạo điều kiện sau này có thể mở rộng và áp dụng vào các dự án thực tế.

TÀI LIỆU KHAM KHẢO

1. Nguồn Internet

[1]. file:///C:/Users/Phat_Doan/Downloads/optics-orderingpointstoidentifytheclusteringstructure-140806123727-phpapp02.pdf.

[2]. https://www.youtube.com/watch?v=5E097ZLE9Sg

[3]. https://scikit-learn.org/dev/auto_examples/cluster/plot_optics.html

2. Sách

[4].

Géron, A. (March 2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow.* America: First Edition.