OPTICS: Ordering Points to Identify the Clustering Structure

Cássio Martini Martins Pereira
cpereira@icmc.usp.br

Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo

Índice

Motivação

- Várias pesquisas em *clustering* focando em **eficiência**, ao invés de **eficácia** (qualidade, utilidade, ...).
- Algumas razões para problemas de eficácia:
  1. quase todos os algoritmos requerem **parâmetros** – difícil determiná-los;
  2. algoritmos são muito **sensíveis** aos parâmetros;
  3. dados reais geralmente apresentam distribuições **assimétricas**, que não podem ser modeladas por um único parâmetro global.

Parâmetros

- Para muitos *data sets* reais não é possível caracterizar sua estrutura de agrupamento utilizando um **único parâmetro global** de densidade.

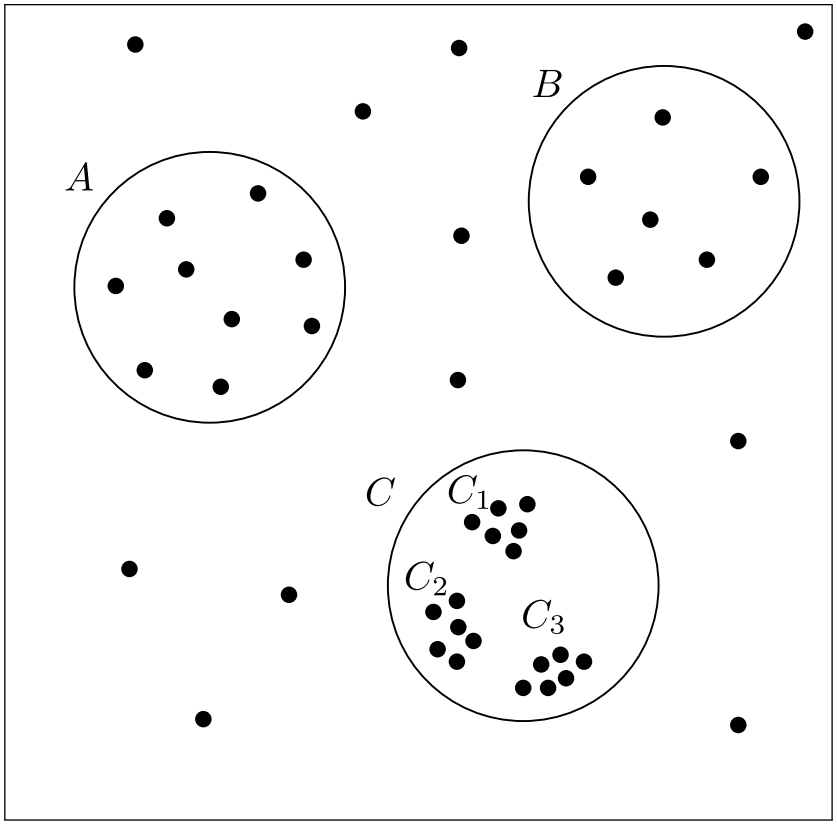


Figura: Diferentes densidades (baseada na original de [Ankerst et al, 1999]).

Proposta

- [Ankerst et al, 1999] introduzem um algoritmo para **análise de agrupamento**: OPTICS.
- **Não** é *explicitamente* um algoritmo de agrupamento.
- A saída é uma **ordenação da base de dados** que permite extrair agrupamentos baseados em **densidade** para infinitas configurações de parâmetros a um custo computacional reduzido.

Agrupamento baseado em densidade

Ideia principal – Origem no DBSCAN

Para cada objeto core de um grupo, sua vizinhança em um dado raio ϵ tem que conter no mínimo *MinPts* objetos.

Vizinhança-ϵ [Ester et al, 1996]

$N_\epsilon(p) = \{q \in D | dist(p, q) \leq \epsilon\}$

Note que o próprio ponto conta como um elemento da sua vizinhança.

Diretamente alcançável por densidade

Definição [Ankerst et al, 1999]

O objeto $p \in D$ é *diretamente alcançável por densidade* a partir de $q \in D$ se:

1. $p \in N_\epsilon(q)$
2. $Card(N_\epsilon(q)) \geq MinPts$ (*core object*)

Note que isso implica que só será possível ser diretamente alcançável por densidade se o ponto de partida for um **core object**.

Alcançável por densidade

Definição [Ankerst et al, 1999]

O objeto $p \in D$ é *alcançável por densidade* a partir de $q \in D$ se existe uma cadeia de objetos $p_1, \ldots, p_n$, $p_1 = q$ e $p_n = p$ se p_{i+1} é **diretamente** alcançável por densidade a partir de p_i.

Conectado por densidade

Definição [Ankerst et al, 1999]

O objeto $p \in D$ é *conectado por densidade* a $q \in D$ se existe um objeto $o \in D$ tal que ambos p e q são **alcançáveis por densidade** a partir de o.

Visualizando as definições

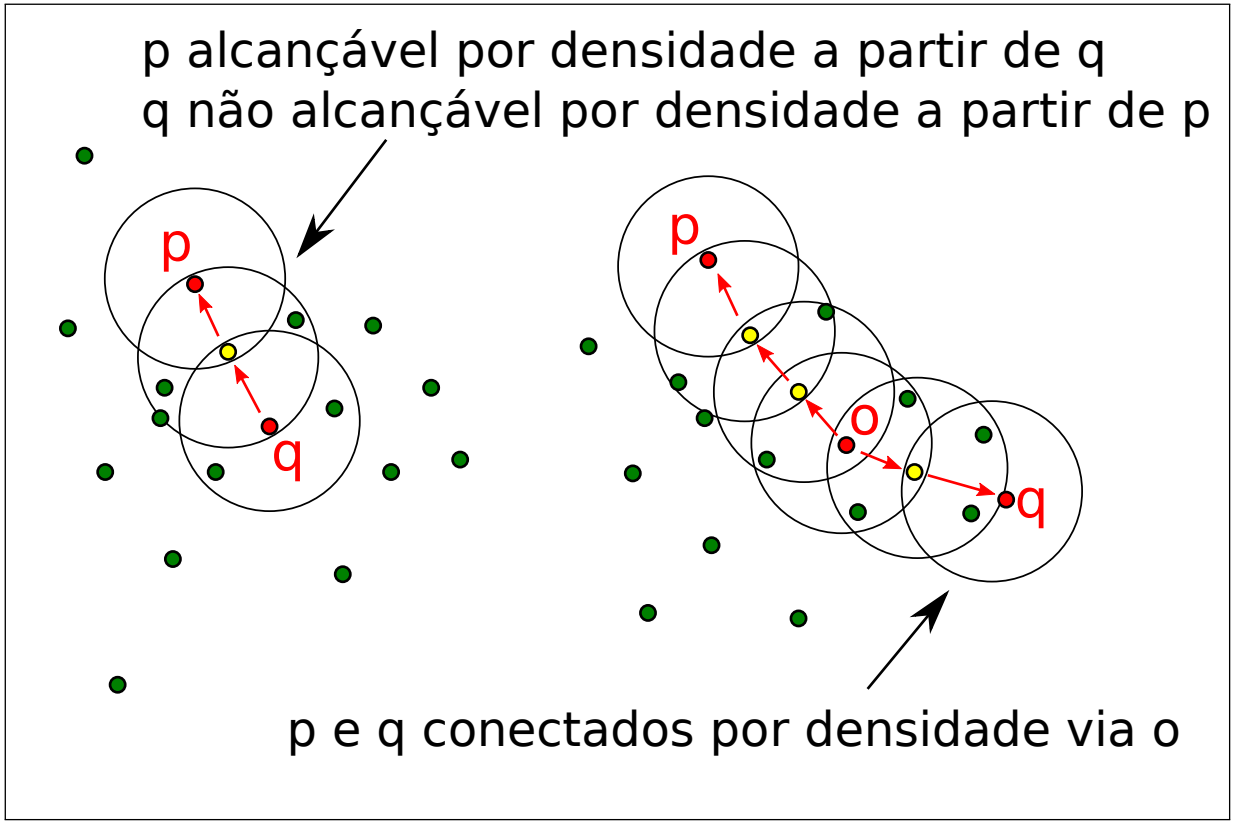


Figura: Medidas de conectividade (baseada na original de [Ankerst et al, 1999])

Considere MinPts = 3.

Cluster

Definição [Ankerst et al, 1999]

Um cluster C é um subconjunto não vazio de D que satisfaz:

1. **Maximalidade**: $\forall p, q \in D$: se $p \in C$ e q é *alcançável por densidade* a partir de p, então $q \in C$.
2. **Conectividade**: $\forall p, q \in C$: p é *conectado por densidade* a q.

Todo objeto que não está contido em nenhum grupo é considerado ruído.

Cluster

- Note que um cluster não contém somente *core objects*.
- Os outros objetos são denominados de **border objects**. Ele são diretamente alcançáveis a partir de pelo menos um *core object* do cluster.

Cluster

- **Fato**: Um cluster é equivalente ao conjunto de todos os objetos em D que são alcançáveis por densidade a partir de um *core object* **arbitrário** do grupo [Ester et al, 1996].

DBSCAN [Ester et al, 1996]

1. Obtenha o próximo ponto p não processado da base.
2. Se $|N_\epsilon(p)| \geq MinPts$ crie um cluster C com todos os objetos em $N_\epsilon(p)$.
3. Verifique a ϵ-vizinhança de todo ponto $q \in C$ ainda não processado.
4. Se $|N_\epsilon(q)| \geq MinPts$ adicione todo ponto em $N_\epsilon(q)$ a C.
5. Volte ao passo 3 caso existam pontos ainda não processados no grupo, caso contrário volte ao passo 1.

Clusters encontrados pelo DBSCAN

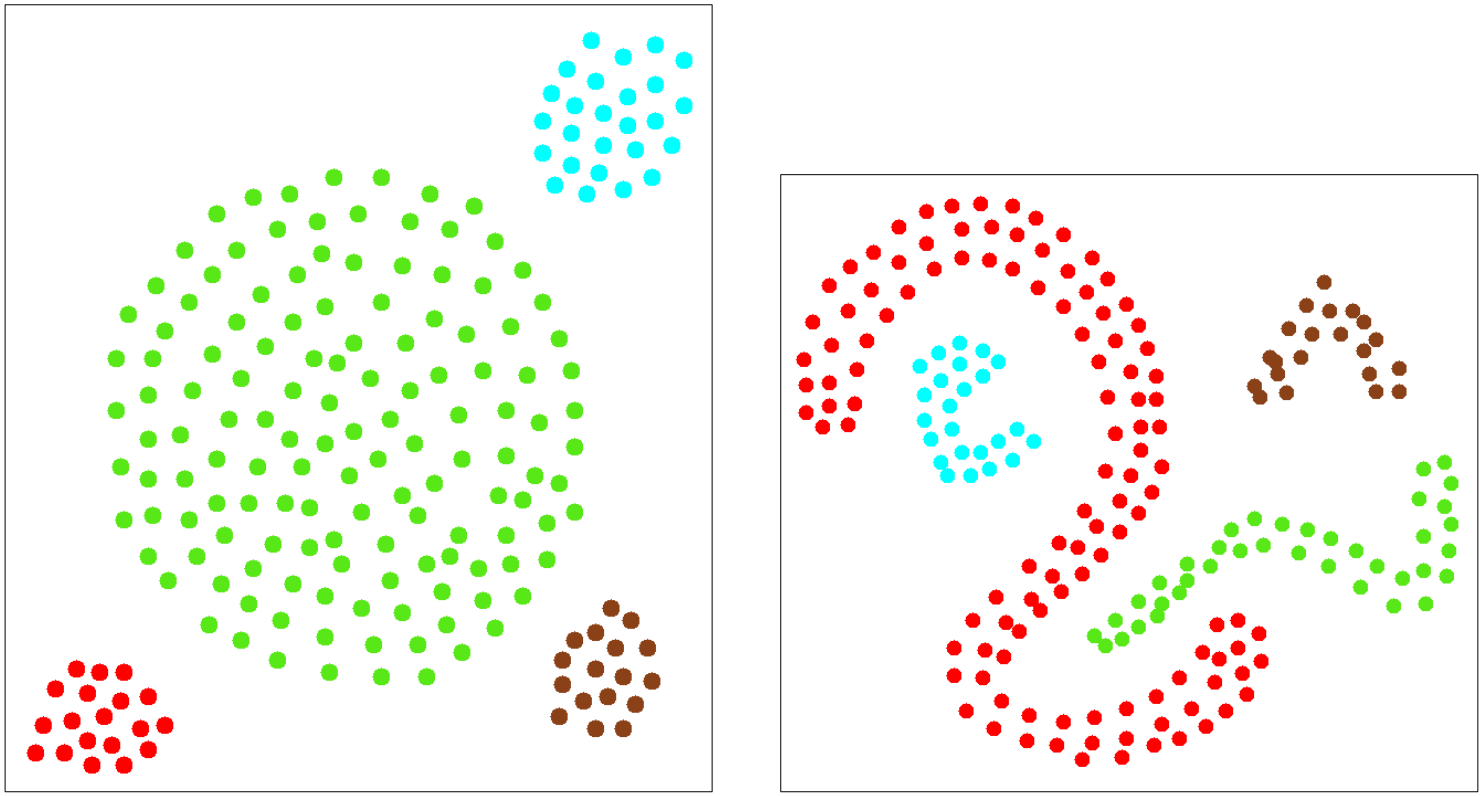


Figura: Clusters obtidos [Ester et al, 1996]

Ordenação de agrupamento baseado em densidade

- Fato: Para um valor **fixo** de *MinPts*, agrupamentos de alta densidade (menor ϵ) estão completamente contidos em grupos de baixa densidade (maior ϵ) conectados [Ankerst et al, 1999].

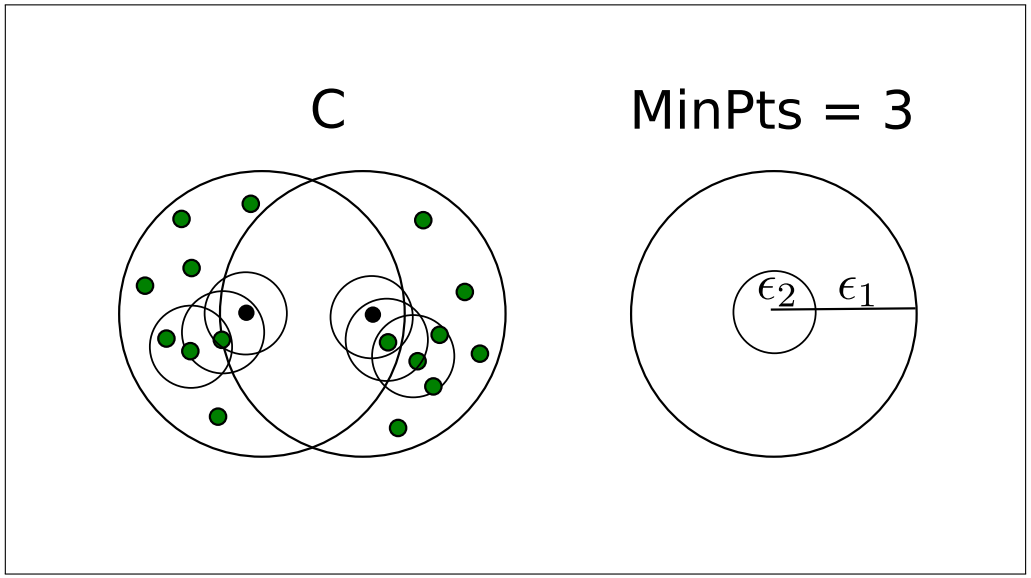


Figura: Grupos aninhados (baseada na original de [Ankerst et al, 1999].)

$$\uparrow d = \frac{\text{nro. de objetos}}{\downarrow raio}$$

Ideia

- Estender o DBSCAN para que vários valores de distância ϵ sejam processados simultaneamente, ou seja, construir agrupamentos com diferentes densidades ao mesmo tempo.
- Para produzir um resultado consistente, os objetos são selecionados de forma que são alcançáveis por densidade com o **menor valor** de ϵ, para garantir que grupos de **alta densidade** sejam terminados primeiro.

Ideia

- Algoritmo funciona para um número infinito de distâncias ϵ_i menores ou iguais à *distância geradora* ϵ, ou seja, $0 \leq \epsilon_i \leq \epsilon$.
- Não mapeia pontos para clusters.
- Armazena a **ordem** em que os objetos são processados mais duas informações (para cada objeto):
  1. *Core distance*
  2. *Reachability distance*

Core distance

Definição [Ankerst et al, 1999]

Seja $p \in D$ e $MinPts\_distance(p)$ a distância de p a seu vizinho $MinPts$. Então a **core distance** é definida como:

$$core\_distance_{\epsilon, MinPts}(p) = \begin{cases} \text{não definida} & \text{, se } Card(N_\epsilon(p)) < MinPts \\ MinPts\_distance(p) & \text{, caso contrário} \end{cases}$$

Traduzindo: É a menor distância ϵ' entre p e um objeto na sua ϵ-vizinhança tal que p seria um *core object* com respeito a ϵ'.

Reachability distance

Definição [Ankerst et al, 1999]

Sejam $p \in D$ e $o \in D$. A *reachability distance* de p a o é definida como:

$$reachability\_distance_{\epsilon, MinPts}(p, o) = \begin{cases} \text{não definida, se } |N_\epsilon(o)| < MinPts \\ max(core\_distance(o), distance(o, p)), \text{ senão} \end{cases}$$

Traduzindo: É a menor distância de p a o tal que p é diretamente alcançável por densidade a partir de o, se o for um *core object*.

Visualizando as definições

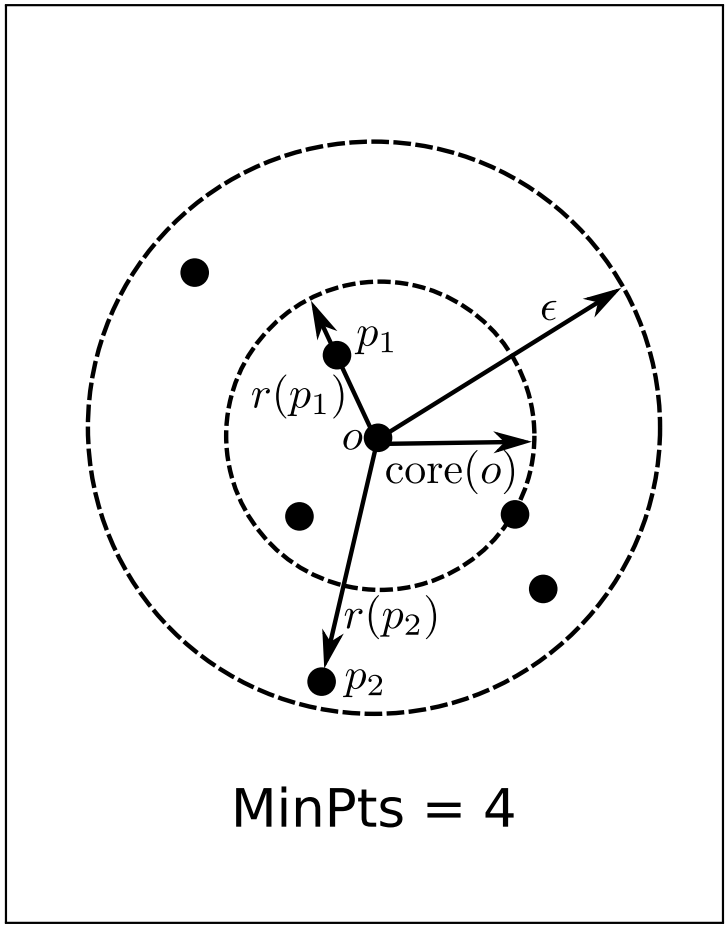


Figura: Core(o), r(p_1, o), r(p_2, o) (baseada na original de [Ankerst et al, 1999]).

Notação

Considere:

- $\mathbf{X}$ a matriz $n \times d$ de dados (n elementos, d dimensões);
- $\mathbf{x_i}$ o i-ésimo elemento (linha) da matriz de dados;
- ϵ o raio desejado;
- MinPts o número mínimo de pontos na vizinhança-ϵ para definir um core object;
- orderSeeds uma fila de prioridades, em que o elemento de maior prioridade é o que tem a menor reachability distance;
- insert() é uma função que adiciona um elemento à fila orderSeeds;
- decrease() é uma função que atualiza a posição de um elemento na fila orderSeeds;
- UNDF representa uma distância infinita.

Algoritmo – OPTICS

OPTICS(**X**, ϵ, MinPts)
for $i = 1..n$ **do**
    if $\mathbf{x_i}.processed \neq True$ **then**
        expandClusterOrder(**X**, $\mathbf{x_i}$, ϵ, MinPts);
    end
end

Algoritmo – OPTICS

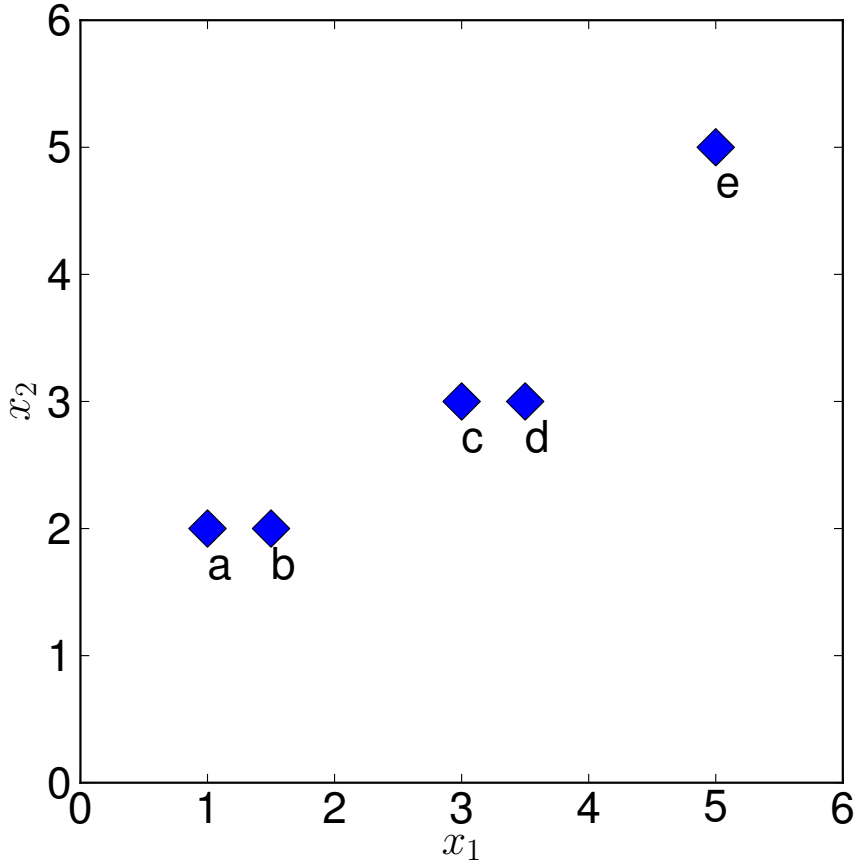
```
ExpandClusterOrder(X, x_i, ε, MinPts)
N = neighbors(x_i, ε);
x_i.processed = True;
x_i.reachdist = UNDF;
x_i.setCoreDist(N, ε, MinPts);
write(x_i);
if x_i.coredist ≠ UNDF then
    orderSeeds.update(N, x_i);
    while !orderSeeds.empty() do
        current = orderSeeds.next();
        N = neighbors(current, ε);
        current.processed = True;
        current.setCoreDist(N, ε, MinPts);
        write(current);
        if current.coredist ≠ UNDF then
            orderSeeds.update(N, current);
        end
    end
end
```

```
orderSeeds.update(N, o)
cdist = o.coredist;
for p ∈ N do
    if !p.processed then
        newrdist = max(cdist, dist(o, p));
        if p.reachdist = UNDF then
            p.reachdist = newrdist;
            insert(p, newrdist);
        end
        else
            if newrdist < p.reachdist then
                p.reachdist = newrdist;
                decrease(p,newrdist);
            end
        end
    end
end
```

Notação

- Por simplicidade, nos gráficos apresentados a seguir, x_1 representa o primeiro atributo do i-ésimo objeto da base, enquanto x_2 representa seu segundo atributo.

Exemplo de execução – OPTICS



$MinPts = 2, \epsilon = 2.0$

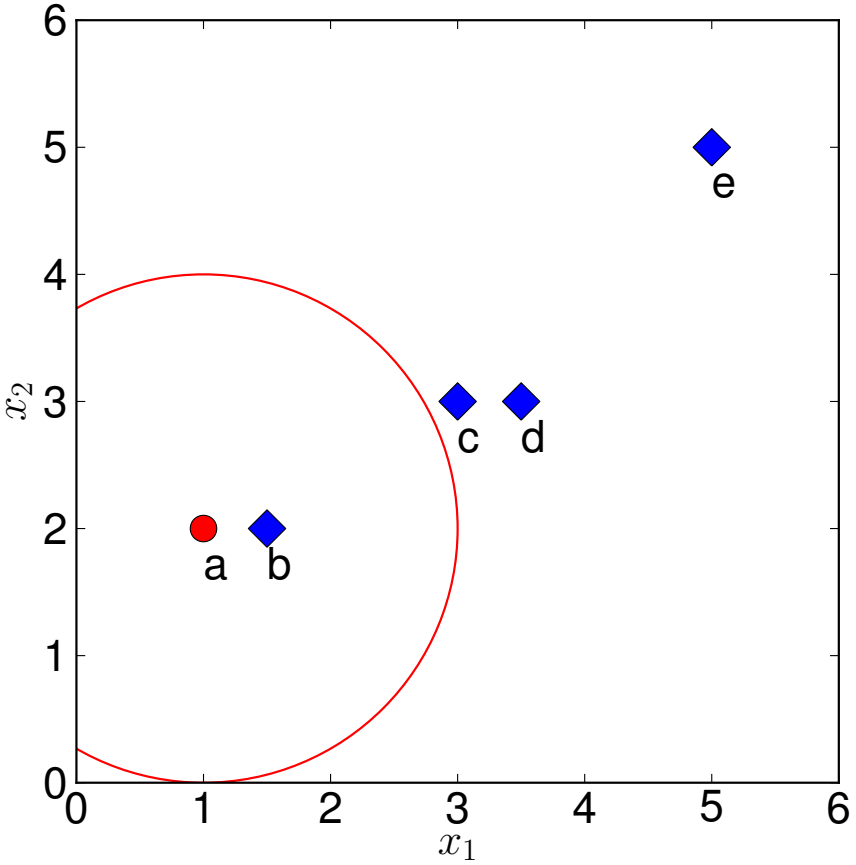
Tabela: Base

Índice	x_1	x_2
a	1	2
b	1.5	2
c	3	3
d	3.5	3
e	5	5

Tabela: Distâncias

d(a,b)	0.50	d(b,d)	2.24
d(a,c)	2.24	d(b,e)	4.61
d(a,d)	2.69	d(c,d)	0.50
d(a,e)	5.00	d(c,e)	2.83
d(b,c)	1.80	d(d,e)	2.50

Exemplo de execução – OPTICS



$MinPts = 2, \epsilon = 2.0$

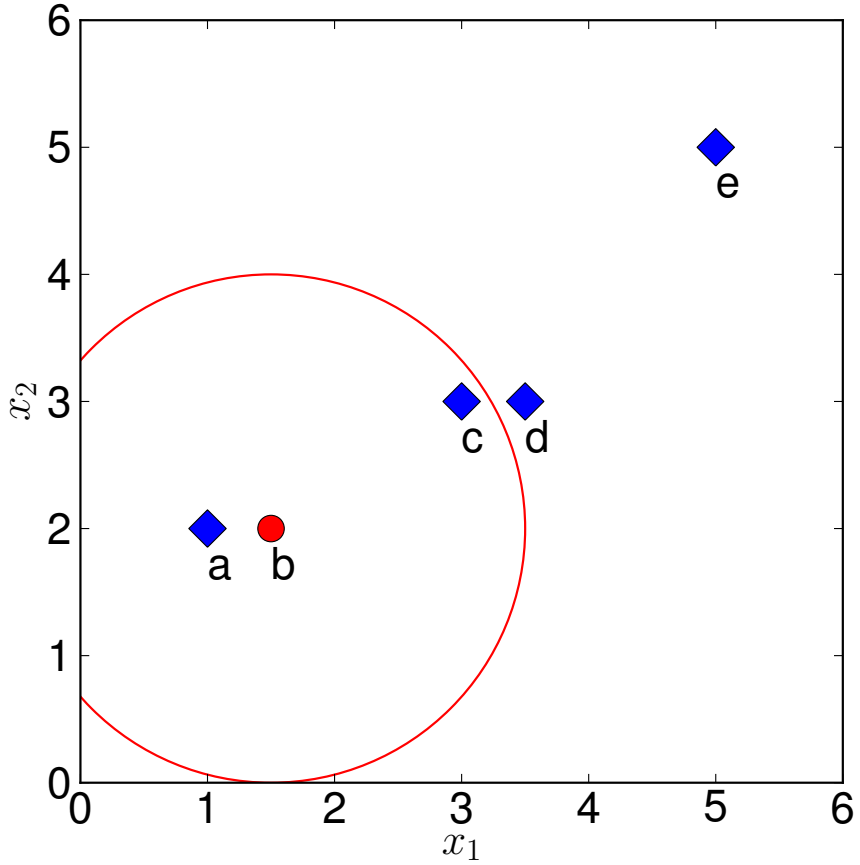
Tabela: Base

Índice	x_1	x_2
a	1	2
b	1.5	2
c	3	3
d	3.5	3
e	5	5

Tabela: Distâncias

d(a,b)	0.50	d(b,d)	2.24
d(a,c)	2.24	d(b,e)	4.61
d(a,d)	2.69	d(c,d)	0.50
d(a,e)	5.00	d(c,e)	2.83
d(b,c)	1.80	d(d,e)	2.50

Exemplo de execução – OPTICS



$MinPts = 2, \epsilon = 2.0$

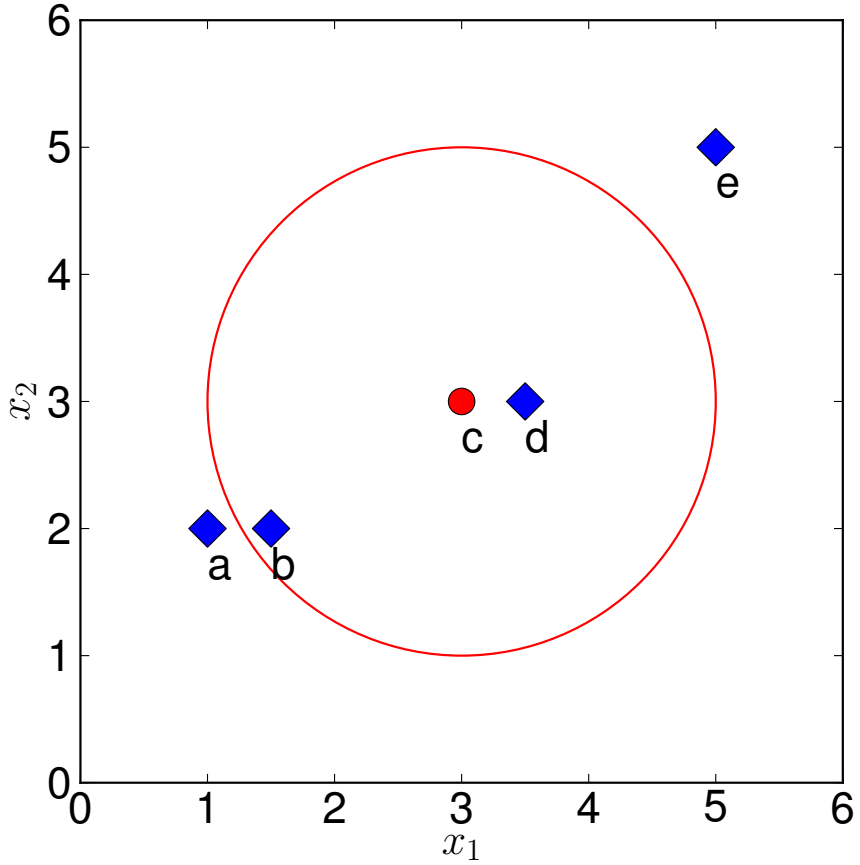
Tabela: Base

Índice	x_1	x_2
a	1	2
b	1.5	2
c	3	3
d	3.5	3
e	5	5

Tabela: Distâncias

d(a,b)	0.50	d(b,d)	2.24
d(a,c)	2.24	d(b,e)	4.61
d(a,d)	2.69	d(c,d)	0.50
d(a,e)	5.00	d(c,e)	2.83
d(b,c)	1.80	d(d,e)	2.50

Exemplo de execução – OPTICS



$MinPts = 2, \epsilon = 2.0$

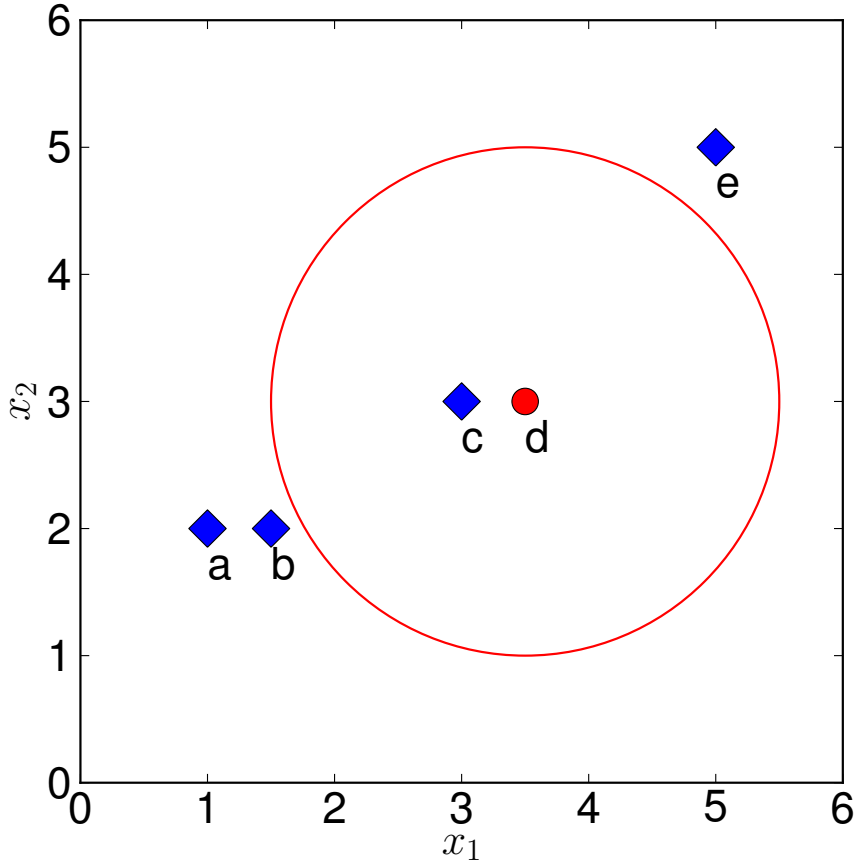
Tabela: Base

Índice	x_1	x_2
a	1	2
b	1.5	2
c	3	3
d	3.5	3
e	5	5

Tabela: Distâncias

d(a,b)	0.50	d(b,d)	2.24
d(a,c)	2.24	d(b,e)	4.61
d(a,d)	2.69	d(c,d)	0.50
d(a,e)	5.00	d(c,e)	2.83
d(b,c)	1.80	d(d,e)	2.50

Exemplo de execução – OPTICS



$MinPts = 2, \epsilon = 2.0$

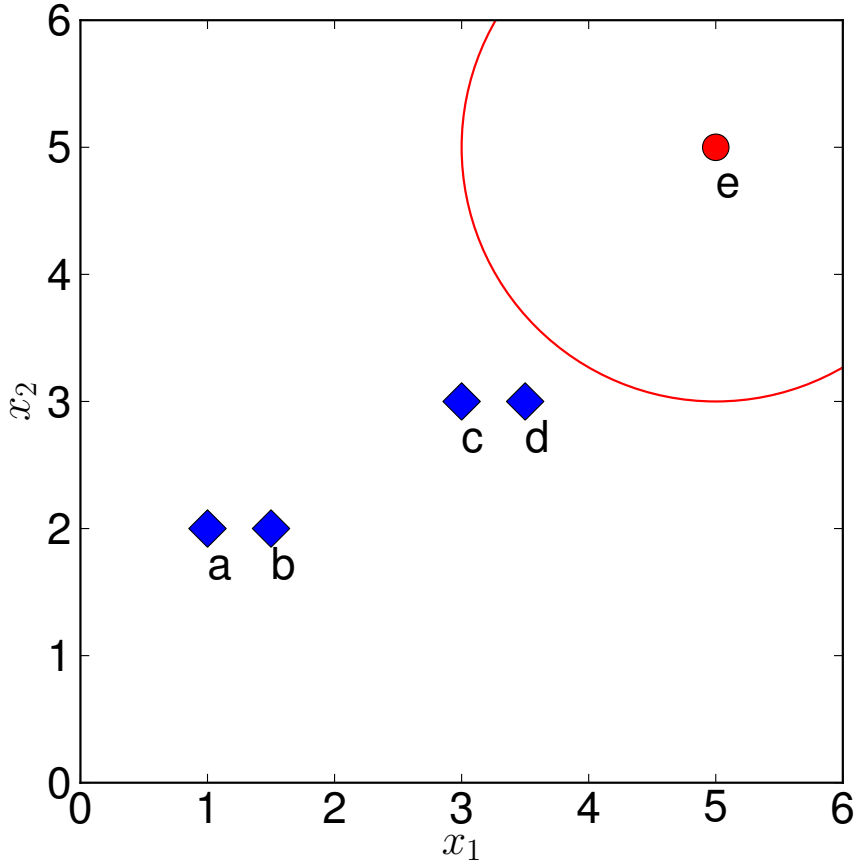
Tabela: Base

Índice	x_1	x_2
a	1	2
b	1.5	2
c	3	3
d	3.5	3
e	5	5

Tabela: Distâncias

d(a,b)	0.50	d(b,d)	2.24
d(a,c)	2.24	d(b,e)	4.61
d(a,d)	2.69	d(c,d)	0.50
d(a,e)	5.00	d(c,e)	2.83
d(b,c)	1.80	d(d,e)	2.50

Exemplo de execução – OPTICS



$MinPts = 2, \epsilon = 2.0$

Tabela: Base

Índice	x_1	x_2
a	1	2
b	1.5	2
c	3	3
d	3.5	3
e	5	5

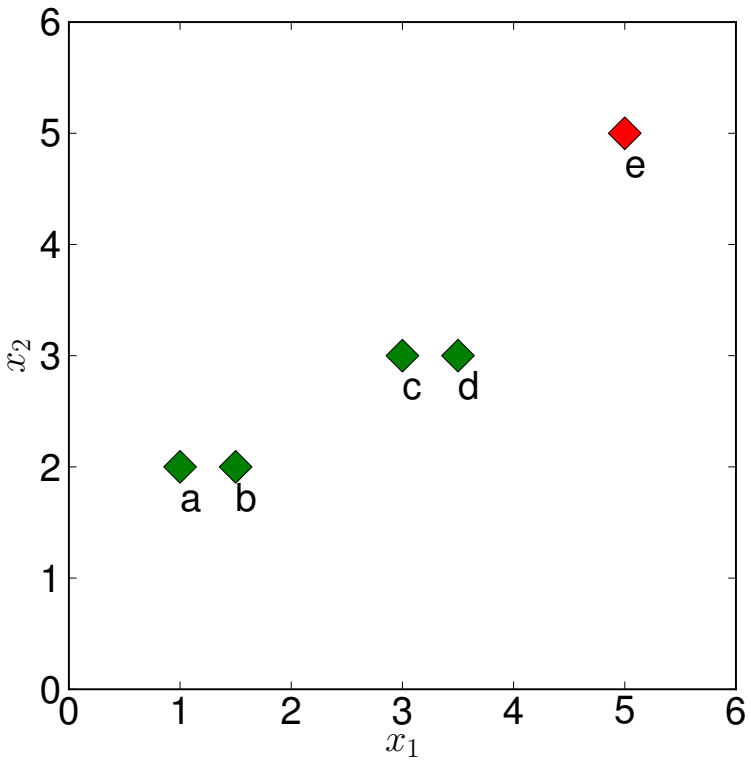
Tabela: Distâncias

d(a,b)	0.50	d(b,d)	2.24
d(a,c)	2.24	d(b,e)	4.61
d(a,d)	2.69	d(c,d)	0.50
d(a,e)	5.00	d(c,e)	2.83
d(b,c)	1.80	d(d,e)	2.50

Algoritmo – ExtractDBSCAN-Clustering

ExtractDBSCAN-Clustering(OrderedObjects, ϵ', MinPts)
ClusterID = NOISE;
for $i = 1..n$ **do**
    if $\mathbf{x_i}.reachdist > \epsilon'$ **then**
        if $\mathbf{x_i}.coredist \leq \epsilon'$ **then**
            ClusterID = next(ClusterID);
            $\mathbf{x_i}.membership$ = ClusterID;
        end
        else
            $\mathbf{x_i}.membership$ = NOISE;
        end
    end
    else
        $\mathbf{x_i}.membership$ = ClusterID;
    end
end

Exemplo de extração de agrupamento



$MinPts = 2, \epsilon = 2.0$

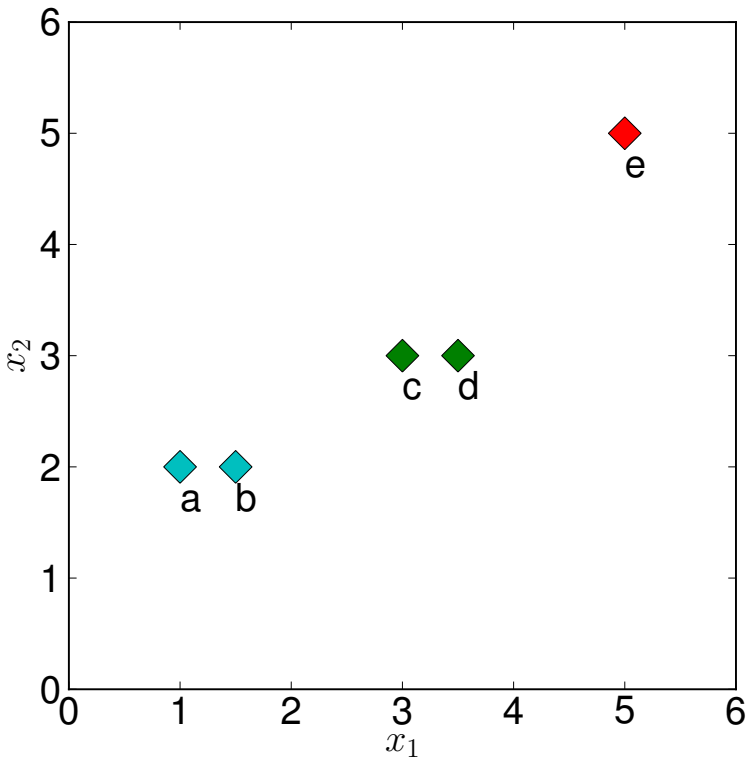
Tabela: Saída OPTICS

Índice	Coredist	Reachdist
a	0.5	UNDF
b	0.5	0.5
c	0.5	1.8
d	0.5	0.5
e	UNDF	UNDF

Tabela: Saída ExtractDBSCAN

Índice	Membership
a	0
b	0
c	0
d	0
e	-1

Exemplo de extração de agrupamento



$MinPts = 2, \epsilon = 0.5$

Tabela: Saída OPTICS

Índice	Coredist	Reachdist
a	0.5	UNDF
b	0.5	0.5
c	0.5	1.8
d	0.5	0.5
e	UNDF	UNDF

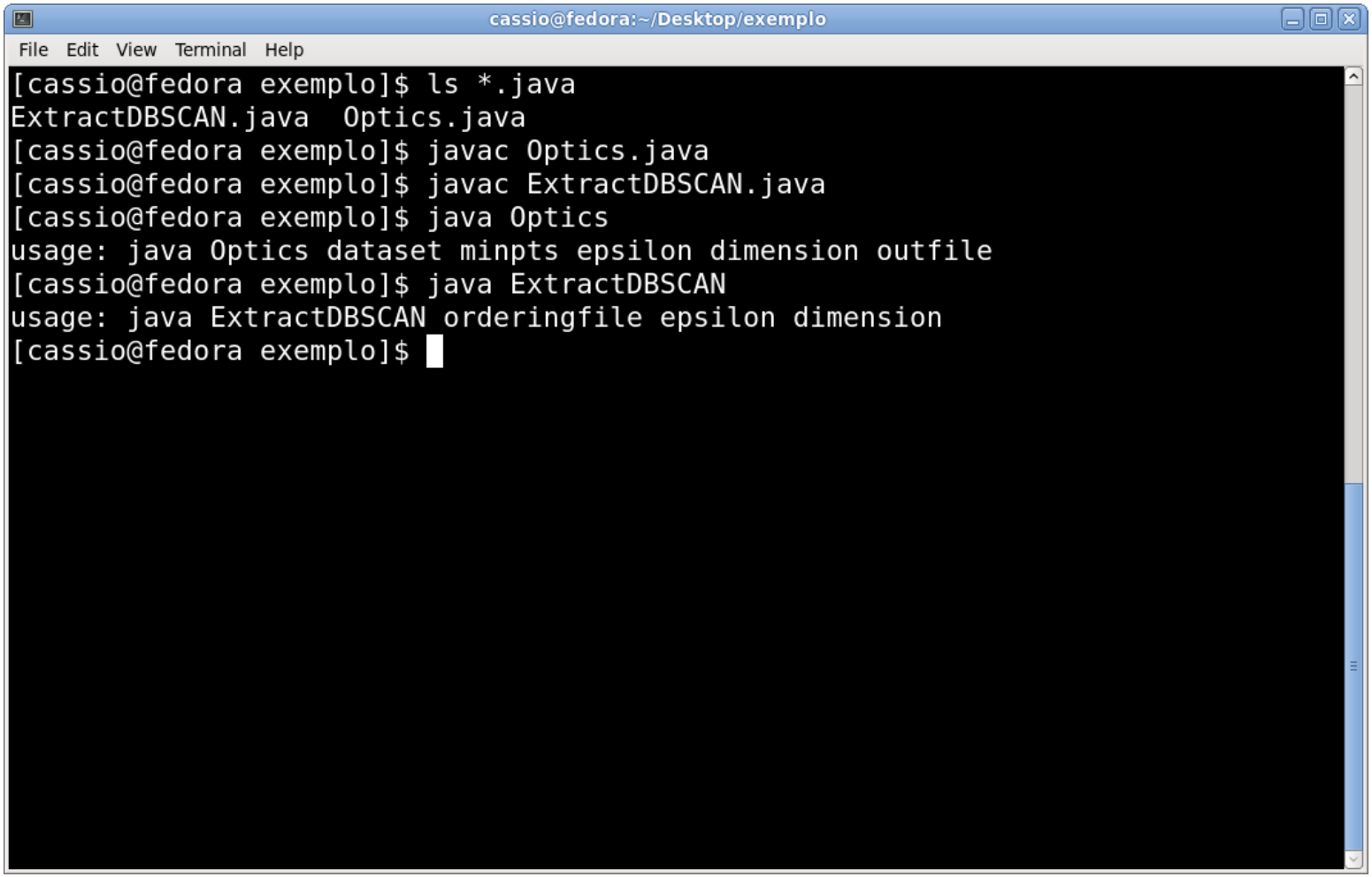
Tabela: Saída ExtractDBSCAN

Índice	Membership
a	0
b	0
c	1
d	1
e	-1

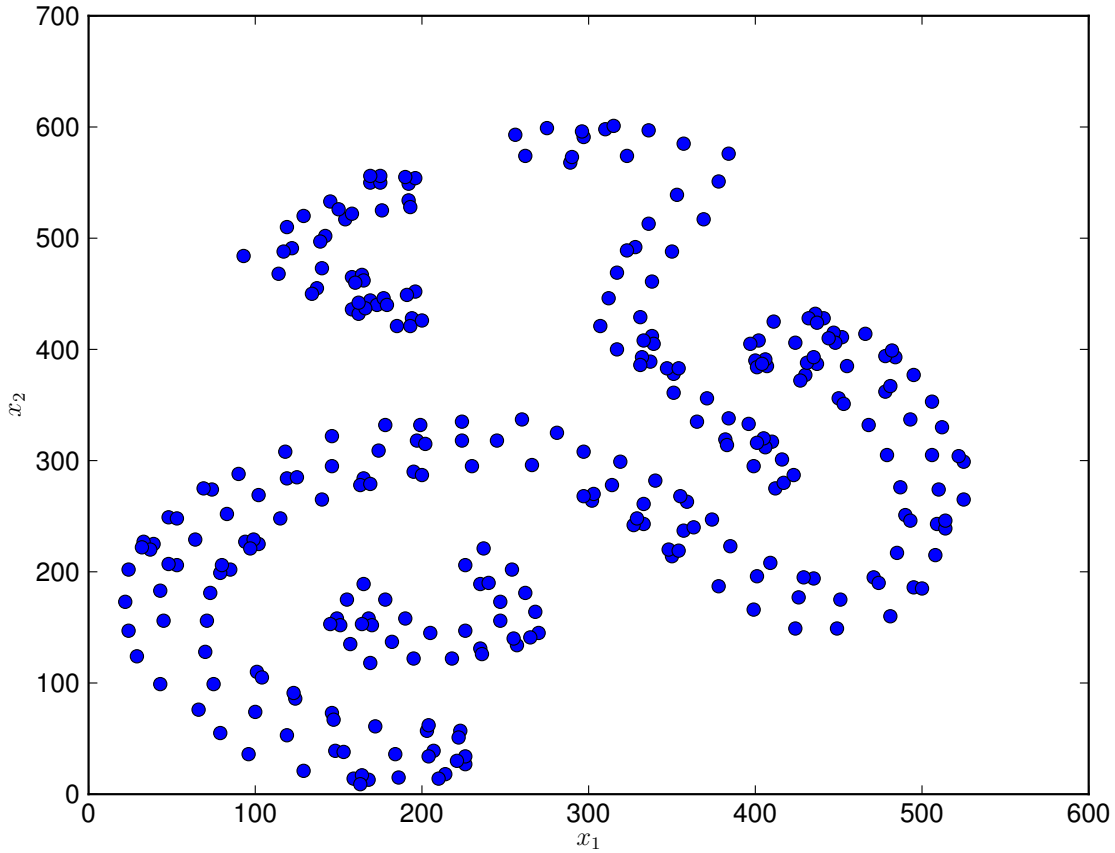
Implementação

- Implementação do algoritmo em Java:
  - Optics
  - ExtractDBSCAN-Clustering
  - Disponível em: www.icmc.usp.br/~cpereira
- Weka: possui implementação do OPTICS, porém não tem o Extract-DBSCAN. É possível usar o OPTICS para analisar os dados e encontrar valores mais adequados de parâmetros, a fim de executar a implementação disponível do DBSCAN.
- **Atenção:** Weka re-escala atributos numéricos para que fiquem entre 0 e 1. Só é possível desabilitar essa normalização alterando o código fonte!

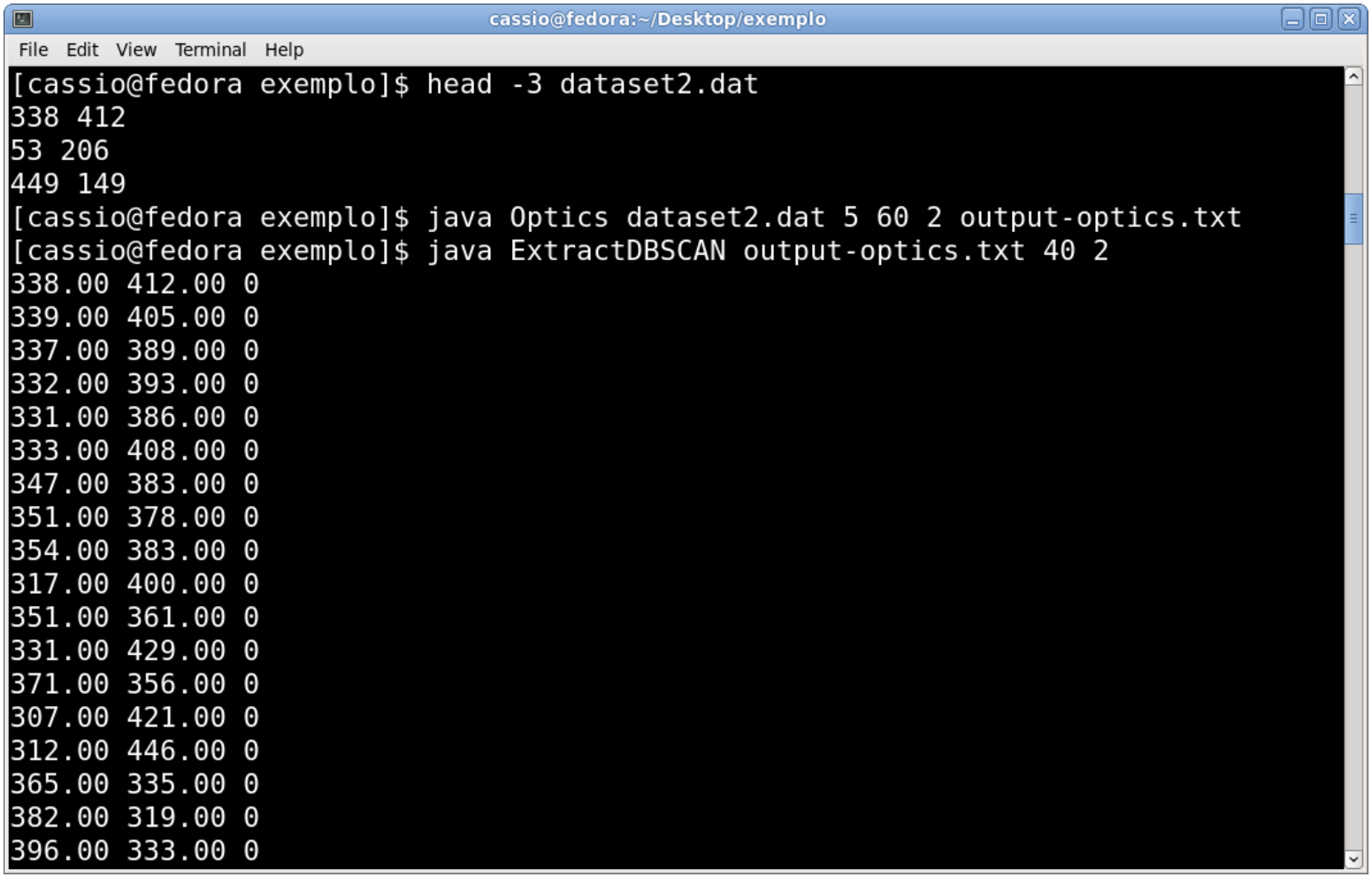
Compilação


cassio@fedora:~/Desktop/exemplo
File Edit View Terminal Help
[cassio@fedora exemplo]$ ls *.java
ExtractDBSCAN.java Optics.java
[cassio@fedora exemplo]$ javac Optics.java
[cassio@fedora exemplo]$ javac ExtractDBSCAN.java
[cassio@fedora exemplo]$ java Optics
usage: java Optics dataset minpts epsilon dimension outfile
[cassio@fedora exemplo]$ java ExtractDBSCAN
usage: java ExtractDBSCAN orderingfile epsilon dimension
[cassio@fedora exemplo]$

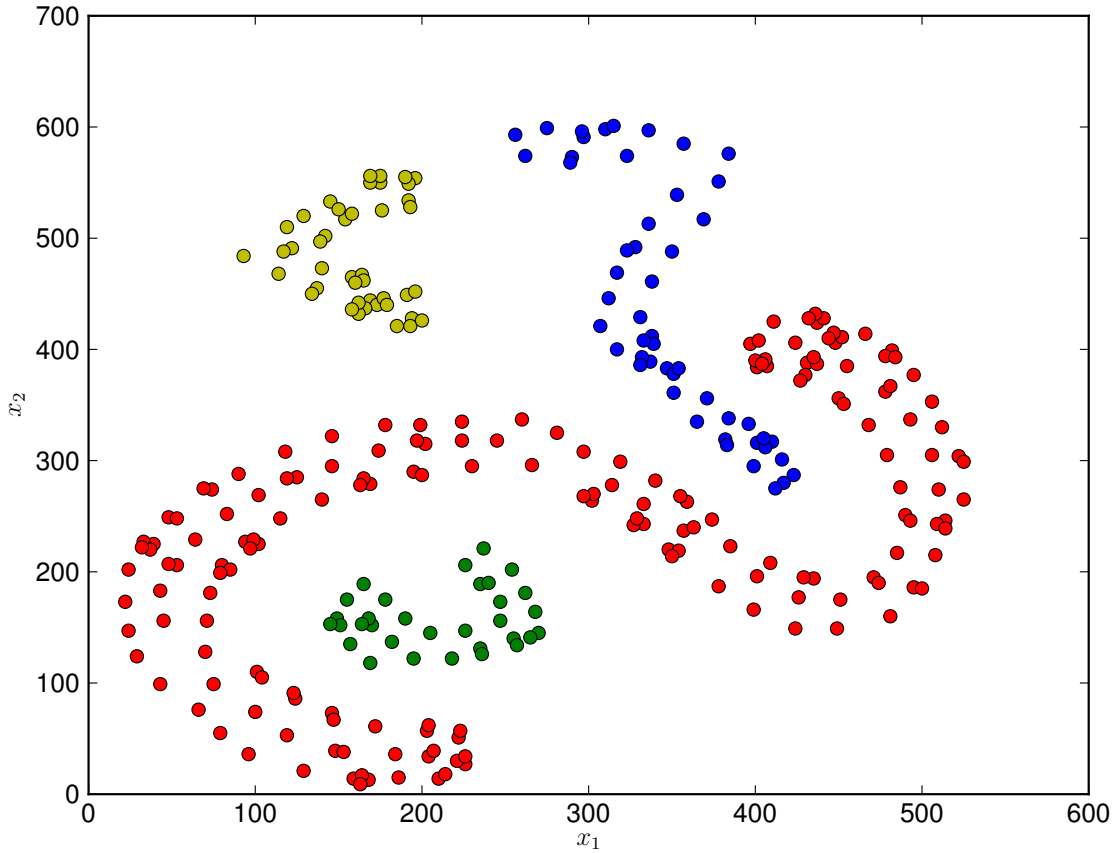
Data set



Execução


cassio@fedora:~/Desktop/exemplo
File Edit View Terminal Help
[cassio@fedora exemplo]$ head -3 dataset2.dat
338 412
53 206
449 149
[cassio@fedora exemplo]$ java Optics dataset2.dat 5 60 2 output-optics.txt
[cassio@fedora exemplo]$ java ExtractDBSCAN output-optics.txt 40 2
338.00 412.00 0
339.00 405.00 0
337.00 389.00 0
332.00 393.00 0
331.00 386.00 0
333.00 408.00 0
347.00 383.00 0
351.00 378.00 0
354.00 383.00 0
317.00 400.00 0
351.00 361.00 0
331.00 429.00 0
371.00 356.00 0
307.00 421.00 0
312.00 446.00 0
365.00 335.00 0
382.00 319.00 0
396.00 333.00 0

Clusters encontrados



Clustering & Visualização

[Jain e Dubes, 1988]

"Cluster analysis is a tool for exploring data and must be supplemented by techniques for visualizing data. The most direct visualization is a two-dimensional plot showing the objects to be clustered as points. Multivariate data cannot always be faithfully reproduced in two dimensions, but when valid such a representation is helpful in verifying the results of a clustering algorithm."

Técnicas de Visualização

- Uma das técnicas propostas por [Ankerst et al, 1999] é plotar os pontos processados versus a reachability distance de cada um deles, montando um reachability plot.
- Nesse gráfico, clusters são identificados como "vales", nos quais há uma queda no valor da reachability distance dos pontos.

- O primeiro ponto de um cluster é o último ponto antes do primeiro com uma reachability distance baixa. Ex: ver cluster 3-16 ilustrado a seguir. Isso ocorre porque para que um determinado ponto tenha uma reachability distance baixa ele precisa necessariamente estar próximo ao ponto imediatamente anterior, conforme a ordenação feita pelo algoritmo. Assim, apesar do ponto 3 ter uma reachability distance alta, ele está próximo ao ponto 4, o que é denotado pela reachability distance baixa do ponto 4.

Reachability Plot

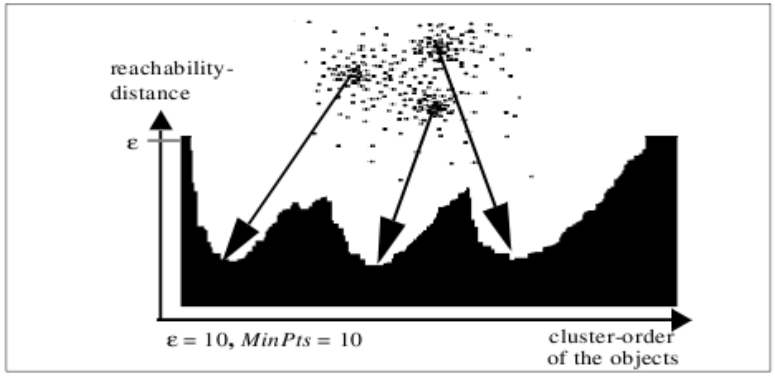


Figura: [Ankerst et al, 1999]

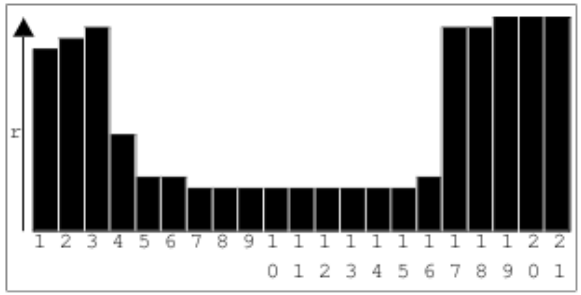


Figura: Cluster 3-16 [Ankerst et al, 1999]

Sensibilidade à mudança de parâmetros

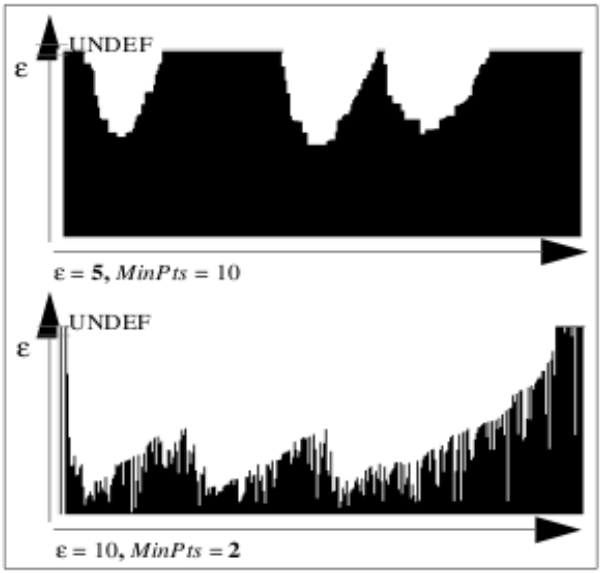


Figura: [Ankerst et al, 1999]

Mesmo com alteração de parâmetros ainda é possível distinguir uma estrutura de agrupamento nos dados.

Técnicas de Visualização

- Outra técnica de visualização chama-se Attribute Plot, na qual se plota embaixo do reachability plot um outro gráfico, que consiste de uma discretização dos valores dos atributos em uma escala de 0 a 255. Isso permite colorir em tons de cinza os valores dos atributos dos pontos.
- Os pontos que pertencem a um mesmo cluster tendem a ter, em geral, tons similares para um ou mais atributos, os quais destoam dos tons para pontos pertencentes a outros clusters. Ex: ver Attribute plot a seguir para dados em 9 dimensões, no qual cada cluster se distingue de outro majoritariamente pelo valor de um único atributo.

Reachability & Attribute plots

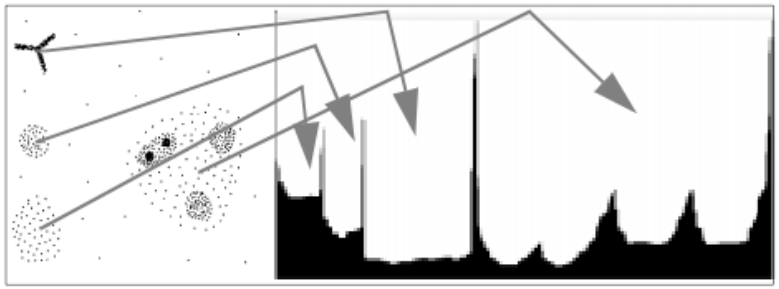


Figura: Clusters de diferentes tamanhos e formas [Ankerst et al, 1999]

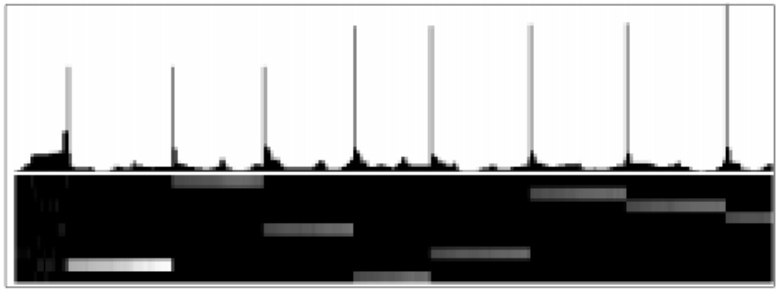


Figura: Attribute plot para data set 9-dimensional [Ankerst et al, 1999]

Conclusões

Vantagens:

- A principal vantagem do OPTICS é que ele não se limita a um único parâmetro global ϵ, porém isso requer um valor fixo de MinPts;
- A estrutura ordenada equivale a todas as execuções do DBSCAN para $\epsilon' \leq \epsilon$ (com MinPts fixo);

Conclusões

Vantagens:

- Isso permite extrair, com baixo custo computacional – $O(n)$ – grupos com diferentes densidades. Porém, **esses grupos estarão em partições diferentes** (extraídas executando Extract-DBSCAN com diferentes valores de ϵ);
- A diferença disso para executar várias vezes o DBSCAN original com parâmetros diferentes é o **custo computacional**. No caso do OPTICS, após rodar o algoritmo de ordenação (que tem o mesmo custo do DBSCAN – $O(n \log n)$ para implementações eficientes), é possível extrair qualquer agrupamento para distâncias $\epsilon' \leq \epsilon$ com custo $O(n)$.

Conclusões

Vantagens:

- Agrupamentos extraídos a partir dele são resistentes a ruído e são capazes de encontrar grupos com formatos arbitrários;
- As técnicas de visualização propostas com base na saída do OPTICS auxiliam a conhecer melhor a distribuição dos dados de um data set;
- Elas são independentes da dimensão dos dados.

Conclusões

Desvantagens:

- A complexidade de tempo do algoritmo é $O(n^2)$ para cálculos de distância feitos com a implementação óbvia (cálculo de distância de todos para todos).
- Isso pode ser aliviado utilizando estruturas de dados para seleção eficiente de vizinhos: $O(n \log n)$, e.g:
  - R*-tree [Beckmann et al, 1990];
  - X-tree [Berchthold et al, 1996];
  - M-tree [Ciaccia et al, 1997].
- Para bases com milhares de atributos o algoritmo tem limitações, devido ao seu custo computacional e à maldição da dimensionalidade, a qual faz com que os pontos se tornem esparsos e se perca regiões densas que caracterizam grupos.

Referências I

OPTICS: Ordering points to identify the clustering structure.
Ankerst, M. and Breunig, M.M. and Kriegel, H.P. and Sander, J.
SIGMOD Rec. V. 28, 2 (Jun. 1999), p. 49-60

A Density based algorithm for discovering clusters in large spatial databases with noise.
Ester, M. and Kriegel, H.P. and Sander, J. and Xu, X.
Proc. 2nd Int Conf on Knowledge Discovery and Data Mining, Portland, 1996, p. 226-231

Algorithms for clustering data.
Anil K. Jain e R. C. Dubes.
Prentice Hall, New Jersey 1988. ISBN: 013022278X

The R*-tree: An Efficient and Robust Access Method for Points and Rectangles.
Beckmann N., Kriegel H.-P., Schneider R., Seeger B.
Proc. ACM SIGMOD Int. Conf. on Management of Data, Atlantic City, NJ, ACM Press, New York, 1990, pp. 322-331.

Referências II

The X-Tree: An Index Structure for High-Dimensional Data.
Berchthold S., Keim D., Kriegel H.-P.
22nd Conf. on Very Large Data Bases, Bombay, India, 1996, pp. 28-39.

M-tree: An Efficient Access Method for Similarity Search in Metric Spaces.
Ciaccia P., Patella M., Zezula P.
Proc. 23rd Int. Conf. on Very Large Data Bases, Athens, Greece, 1997, pp. 426-435.