5. O Método DBSCAN

DBSCAN, abreviação do termo 'Density Based Spatial Clustering of Application with Noise' (Clusterização Espacial Baseada em Densidade de Aplicações com Ruído) é um método de clusterização não paramétrico baseado em densidade, proposto por ESTER et al (1996), que é significativamente efetivo para identificar clusters de formato arbitrário e de diferentes tamanhos, identificar e separar os ruídos dos dados e detectar clusters "naturais" e seus arranjos dentro do espaço de dados, sem qualquer informação preliminar sobre os grupos. O método requer somente um parâmetro de entrada, mas dá suporte para determinar um apropriado valor para ele.

ESTER *et al.* (1996) escrevem que a noção de clusters e o algoritmo DBSCAN se aplicam para espaços *Euclidianos* de duas e três dimensões, como para qualquer espaço característico de alta dimensão. O método DBSCAN é aplicável a qualquer base de dados contendo dados de um espaço métrico (isto é, bases de dados com uma função de distância para pares de objetos) (ESTER *et al.*, 1998). Os autores salientam ainda que a abordagem trabalha com qualquer função de distância, de maneira que uma função apropriada pode ser escolhida para alguma dada aplicação. Neste trabalho usamos a distância euclidiana, definida em (27), Seção 4.3.

A ideia chave do método DBSCAN é que, para cada ponto de um cluster, a vizinhança para um dado raio contém, no mínimo, certo número de pontos, ou seja, a densidade na vizinhança tem que exceder um limiar. Para entender o método é necessário conhecer algumas definições específicas listadas a seguir.

Definição 1: (ε-vizinhança de um ponto) A vizinhança de um objeto p com raio ε é chamada de ε-vizinhança de p é dada por: $N\varepsilon(p) = \{ q \text{ em } D | \text{ dist}(p,q) < \varepsilon \}$. Na Figura 5.1 abaixo os círculos representam respectivamente a ε-vizinhança do ponto q e ε -vizinhança do ponto p.

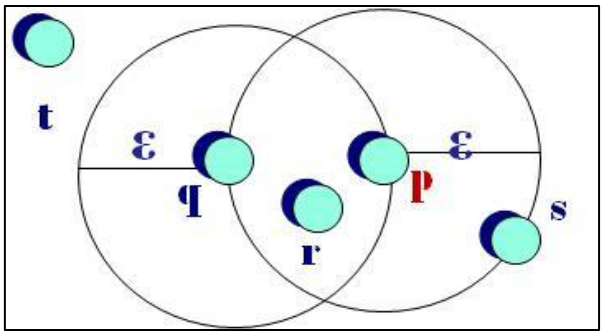

Figura 5.1. ε-vizinhança de *q e* ε-vizinhança de p.

Uma abordagem ingênua poderia exigir para cada ponto em um cluster que haja pelo menos um número mínimo (*MinPts*) de pontos na *ε* -vizinhança daquele ponto. No entanto, esta abordagem falha porque há dois tipos de pontos em um cluster, pontos dentro do cluster (pontos centrais) e pontos na fronteira do cluster (pontos de borda).

Definição 2 (Ponto Central) : Se a ε-vizinhança de um objeto p contém ao menos um número mínimo, *MinPts*, de objetos, então o objeto p é chamado de ponto central . Por exemplo, na Figura 5.1, se adotarmos *MinPts* = 4, p é um ponto central e os demais não são pontos centrais.

Definição 3 (pontos de borda): Se a ε–vizinhança de um objeto p contém menos que *MinPts* mas contém algum ponto central, então o objeto p é chamado de ponto de borda. Na Figura 5.1, q, r e s são pontos de borda.

Em geral, a ε-vizinhança de um ponto de borda contém significativamente menos pontos do que a ε-vizinhança de um ponto central. Portanto, deve-se definir o número mínimo de pontos, *MinPts*, para um valor relativamente baixo, de modo a incluir todos os pontos pertencentes a um mesmo cluster. Este valor, no entanto, não pode ser característica para o respectivo conjunto - particularmente na presença de ruído. Portanto, exige-se que, para cada ponto p em um cluster C exista um ponto q em C, de modo que p está dentro da ε-vizinhança de q e Nε (q) contém pelo menos *MinPts* pontos. Esta definição é elaborada como segue:

Definição 4 (Alcance Direto por Densidade): Um objeto *p* é alcançável por densidade diretamente do objeto *q*, com respeito à ε e a *MinPts*, se *p* está na ε-vizinhança de *q*, e *q* é um ponto central.

Alcance direto por densidade é simétrico para pares de pontos centrais. Contudo, em geral, o alcance direto por densidade não é simétrico se um ponto central e um ponto de borda estão envolvidos. Na Figura 5.2. a seguir, por exemplo, p é alcançável por densidade diretamente de q; mas q não é alcançável por densidade diretamente de p, porque q não é ponto central.

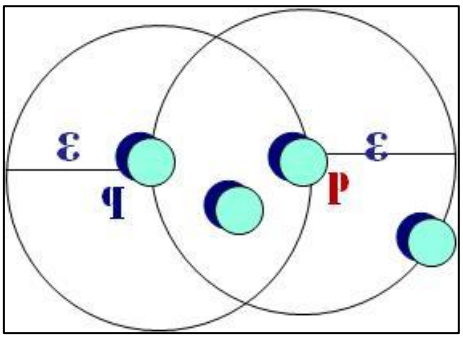

Figura 5.2: Alcance direto por densidade no método DBSCAN.

Definição 5 (Alcance por Densidade): Um objeto *p* é alcançável por densidade do objeto *q* com respeito à *ε e MinPts* em um conjunto *D*, se existe uma cadeia de objetos $\{p_1,..., p_n\}$, tais que $p_1 = q$ e $p_n = p$ *e* p_{i+1} é alcançável por densidade diretamente de p_i com respeito a *ε* e *MinPts*, para $1 \leq i \leq n$, p_i em *D*.

Na Figura 5.3, *q* é alcançável por densidade de *p e p* é alcançável por densidade de *q*. Há, portanto, um fechamento transitivo do alcance por densidade. Alcance por densidade é uma extensão canônica do alcance direto por densidade. Essa relação é transitiva, mas não é simétrica.

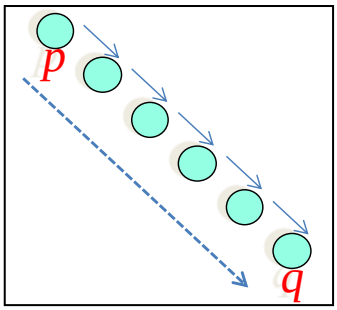

Figura 5.3: Alcance por densidade no método DBSCAN.

Embora não simétrica em geral, é óbvio que o alcance por densidade é simétrico para os pontos centrais. Dois pontos de fronteira de um mesmo cluster C não são, possivelmente, alcançáveis por densidade uns dos outros, porque a condição ponto central pode não valer para ambos. No entanto, deve haver um ponto central em C a partir do qual os dois pontos de fronteira de C são alcançáveis por densidade. Por isso, é introduzida a definição de densidade de conectividade por densidade que cobre esta relação de pontos de borda.

Definição 6 (Conexão por densidade): Um objeto *p* é conectado por densidade ao objeto *q* com respeito à ε e *MinPts* em um conjunto de objetos *D*, se existe um objeto *o* em *D* tal que ambos *p* e *q* são alcançáveis por densidade do objeto o com respeito à ε *e MinPts*. Na Figura p e q são conectados por densidade através de o.

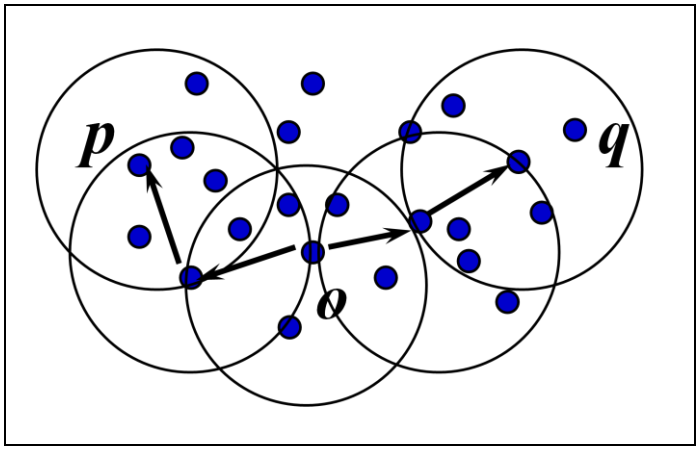

Figura 5.4: Conexão por densidade no método DBSCAN.

Conexão por densidade é uma relação simétrica. Para pontos alcançáveis por densidade a relação de conectividade por densidade é também reflexiva.

Definição 7 (Cluster DBSCAN): Seja D uma base de dados de pontos. Um cluster C com respeito à ε e *MinPts* é um subconjunto não vazio de D satisfazendo as seguintes condições:

1) ∀ p, q: se p ∈ C e q é alcançável por densidade a partir de p com respeito à ε e *MinPts*, então q ∈ C (Maximalidade) .
2) ∀ p, q ∈ C: p é conectado por densidade a q com respeito à ε e *MinPts* (Conectividade).

Intuitivamente, um cluster DBSCAN é o conjunto de pontos conectados por densidade que é maximal com respeito ao alcance por densidade.

Definição 8 (Ruído): Sejam $C_{1, \ldots}$, C_k os clusters da base de dados D com respeito aos parâmetros ε e *MinPts*, i = 1,. . ., K. Então, define-se o ruído como o conjunto de pontos na base de dados D que não pertença a qualquer grupo Ci, ou seja, o ruído = {p ∈ D | ∀ i: p ∉ Ci}. Um objeto que não é ponto central nem ponto de borda, é ruído. Na Figura 5.1, ***t é ruído.***

Assim definido, um cluster C com respeito à ε e *MinPts* contém pelo menos *MinPts* pontos por causa das seguintes razões. Uma vez que C contém pelo menos um ponto p, p deve ser conectado por densidade a si mesmo, através de algum ponto *o* (que pode ser igual a p). Assim pelo menos *o* deve satisfazer a condição de ponto central e, consequentemente, a ε–vizinhança de *o* contém, pelo menos, *MinPts* pontos.

Os seguintes Lemas são importantes para validar a definição do algoritmo DBSCAN. Intuitivamente, estes lemas afirmam o seguinte: dados os parâmetros ε e *MinPts*, pode-se descobrir um cluster em uma abordagem de duas etapas. Primeiro, escolher um ponto arbitrário do banco de dados satisfazendo a condição ponto central, como uma semente. Depois, recupera-se todos os pontos que são alcançados por densidade, a partir do ponto inicial, obtendo o cluster que contém e ponto inicial.

Lema 1: Seja p um ponto em D e | Nε (p)) | ≥ *MinPts*. Então o conjunto

O ={o | o ∈ D, o e é alcançável por densidade a partir de p, com respeito à ε e *MinPts*} é um cluster com respeito à ε e *MinPts*.

Não é óbvio que um cluster C com respeito à ε e *MinPts* é unicamente determinado por qualquer de seus pontos centrais. No entanto, cada ponto C é alcançável por densidade a partir de qualquer um dos pontos centrais de C e, portanto, um cluster C contém exatamente os pontos que são alcançáveis por densidade a partir de um arbitrário ponto central de C.

Lema 2: Seja C um cluster com respeito à ε e *MinPts* e seja p qualquer ponto em C com | Nε (p)) | ≥ *MinPts* . Então C é igual ao conjunto O = {o | o é alcançável por densidade a partir de p com respeito à ε e *MinPts*.

De acordo com as definições 7 e 8, o DBSCAN é designado para descobrir clusters e ruído em uma base de dados espacial. Isso qualifica o método em classificar diretamente os ruídos, que por outros métodos de clusterização, como o k-means, hierárquico, CLARANS, etc., seriam colocados obrigatoriamente, e equivocadamente, em algum cluster, não necessariamente formado só por ruídos.

O procedimento para encontrar um cluster é baseado no fato de que um cluster é inequivocamente determinado por qualquer de seus centros (ESTER *et al.*, 1998). Se um índice espacial é usado, como uma *R-tree*, o método DBSCAN alcança melhor desempenho, obtendo a complexidade computacional de O(*n* log *n*), onde *n* é o número de objetos da base de dados. Caso contrário, ela é de O(n^2) (SHEIKHOLESLAMI *et al.*, 1998).

Idealmente tem-se que saber os parâmetros *ε* e *MinPts* adequados pois pode-se recuperar todos os pontos que são alcançáveis por densidade a partir de um dado ponto usando os parâmetros corretos; o algoritmo DBSCAN é assim muito sensível aos parâmetros definidos pelo usuário (HAN e KAMBER, 2001). Em ESTER et al. (1996) os autores propõem um método heurístico eficaz para estimar os parâmetros iniciais *ε* e *MinPts* do mais fino, ou seja, o menos denso cluster de uma base de dados. Por isso, o DBSCAN usa valores globais para ε e *MinPts*, isto é, os mesmos valores para

determinar todos os clusters. Os parâmetros de ε e *MinPts* do "mais fino" e menos denso cluster são, por isso, bons candidatos para os valores globais especificando a menor densidade que não é considerada ser ruído.

Para determinar valores para estes parâmetros de entrada, Ester et al. (1996) recomendam que para um dado k (recomendado ser igual a 4) seja definida uma função Distk que mapeia cada ponto p do conjunto de dados em um numero real dado por distancia de p ao seu k-ésimo vizinho mais próximo. Então a 'Distk(p)- vizinhança' de p contém exatamente k+1 pontos para quase todos os pontos p. A 'Distk(p)- vizinhança' de p contém mais que k+1 pontos somente se todos os pontos tem exatamente a mesma distancia Distk(p) de p, o que é bastante improvável. Além do mais, mudar k para um ponto em um clutster não resulta em grandes mudanças de Distk(p). Isto só aconteceria se o k-ésimo vizinho mais próximo de p, para k=1,2,3,... fossem localizados em uma linha reta, numa base de dados bidimensional, que é, em geral não comum para pontos em um cluster.

Quando classificados os pontos em ordem decrescente dos seus valores de Distk {Distk}, deve-se traçar o Gráfico de {1,...n} vs {Distk}; este gráfico é chamado Gráfico da DistK ordenada . O gráfico desta função dá algumas sugestões com respeito à distribuição de densidade na base de dados. Se escolhermos um ponto arbitrário p, escolhemos o parâmetro ε para DistK(p) e escolhemos o parâmetro *MinPts* para k. Todos os pontos com um valor igual ou menor do que Distk serão pontos centrais. Se pudéssemos encontrar um ponto de limiar com o valor Distk máximo para o "mais fino" cluster de D teríamos os desejado valores dos parâmetros. O ponto de limiar é o primeiro ponto do primeiro "vale" do gráfico da Distk ordenada (ver Figura 6). Todos os pontos com um valor mais alto Distk (à esquerda do limiar) são considerados ruídos, e todos os outros pontos (à direita do limiar) são atribuídos a algum cluster.

Assim, tal heurística recomenda procurar pelo 'primeiro vale' da curva resultante, ou seja, o ponto em que há uma mudança nítida na tendência do gráfico, passando por um ponto de inclinação aproximadamente nula. O valor de Distk neste ponto é o valor estimado para ε; o valor de k é o valor estimado para *MinPts*. ESTER et al. (1996) propõem usar sempre k=4 e a partir daí obter os valores de Distk para k=4 para qualquer base de dados e encontrar a estimativa de ε. Os autores garantem por experimentos que

gráficos da Distk para k>4 não diferem significativamente do gráfico da Dist4 e necessitam consideravelmente de maior esforço computacional. Embora tal procedimento seja eficiente, ele é muito subjetivo ao depender da obtenção da coordenada deste 'vale'. Daí o estudo da distribuição de Distk supondo um percentil da distk para separar ruído dos dados também pode ser usado, principalmente nos casos onde a visualização do "vale da curva' ou a obtenção de suas coordenadas não é possível. Por exemplo, avançando na idéia de explorar a distribuição de probabilidade da medida Distk, XU et al. (1998) desenvolveram o método de clusterização DBCLASD que compara a distribuição esperada com a distribuição observada de distk usando a 'filosofia' do teste Qui-quadrado.

5.1 Algoritmo DBSCAN

O método DBSCAN encontra clusters verificando a vizinhança ε de cada ponto na base de dados, começando por um objeto arbitrário p. Se p é um ponto central, um novo cluster com *p* como um centro é criado (Definição 2 e Lema 2). Se p é um ponto de fronteira, nenhum ponto é alcançável por densidade a partir de p e DBSCAN visita o próximo ponto na base. O método DBSCAN, então, iterativamente coleta objetos alcançáveis por densidade diretamente de pontos centrais, que pode envolver a união de alguns clusters alcançáveis por densidade. O processo termina quando nenhum novo ponto pode ser adicionado a qualquer cluster. Para o algoritmo DBSCAN assim definido, quaisquer dois pontos centrais com distância menor ou igual a ε são colocados no mesmo cluster. Qualquer ponto de borda que está perto de um ponto central é colocado no mesmo cluster do ponto central. Pontos que não são diretamente atingíveis por algum ponto central são classificados como ruído.

Resumindo, o agrupamento de objetos a partir de qualquer cluster de C é um processo de duas etapas. Na primeira, um objeto central arbitrário X do cluster 1 (X_{C1}) é identificado. Em seguida, todos os objetos alcançáveis por densidade a partir de X_{C1} são buscados. Na segunda etapa, cada cadeia de objetos partindo de X_{C1} é detectada de forma recursiva.

Duas sub-rotinas (DBSCAN e ExpandCluster) são apresentadas como pseudo-códigos para o algoritmo DBSCAN nas Figuras 5.5 e 5.6. Todos os objetos centrais nas cadeias são armazenados na chamada lista ‘Seeds’. A lista ‘Seeds’ é atualizada durante um loop 'for' sempre que um novo objeto central é descoberto (dentro de uma ε - vizinhança que está a ser avaliada) e quando um elemento de uma cadeia é classificado.

```
Algoritmo DBSCAN (DataBase, Eps, MinPts)
// Todos os pontos em DataBase estão Unclassified

 N:= Tamanho da Base de Dados;
 ClusterId := 1;

 For i from 1 to N do
      If x_i in ClusterId= Unclassified;
        ExpandCluster (x_i, ClusterId, Eps, MinPts);
         If ExpandCluster == Expansion Successful
            ClusterId := ClusterId+1;
         END If
     END If
 END For
END Algoritmo DBSCAN
```

Figura 5.5: Pseudo Código do Algoritmo DBSCAN- Subrotina Principal

```
Algoritmo ExpandCluster (x_i,ClusterId, Eps, MinPts)

  Seeds:= N_Eps (x_i)

  If |Seeds|< MinPts            //Density(x_i)
    Mark x_i as Ruído;          // x_i está Classified
    Return Expansion without success;
    Else       // x_i is Ponto Central e está Classified

    // Passo 1: x_i é identificado como um ponto central
inicial para o Cluster ClusterId
    // Passo 2: Identifica todos objetos alcançáveis por
densidade a partir de x_i

     Assign all objects in seeds list to ClusterId;
     Delete x_i from Seeds list;

      For all x_j in Seeds List
         N_Eps (x_j)
         If | N_Eps (x_j) |≥ MinPts  // x_j é um ponto central
            Mark x_j as Ponto Central; // x_i está Classified

    //mais expansão de cluster somente para o ponto central

           For all x_k in N_Eps (x_j)
                If  x_k is UNCLASSIFIED or is Ruído
                  If x_k is UNCLASSIFIED
                     Add x_k to seeds list;
                  End If;
                  Assign x_k to ClusterId; // Reachable Density
                End If;
            End For;
        End If;
       End For;

    Return Expansion Successfull;

  End If;
  End Algoritmo ExpandCluster;
```

Figura 5.6: Pseudo Código do Algoritmo DBSCAN- Subrotina de Expansão dos Clusters.

5.2 DBSCAN Revisado

Para um conjunto de dados com clusters bem definidos, com fronteiras distantes, o método DBSCAN definido por ESTER et al. (1996) trabalha bem. Mas no caso de existir uma cadeia densa de objetos conectando dois clusters, ou seja, quando pontos de borda de dois clusters estão relativamente muito perto um do outro, GUHA *et al.* (1998) lembram que o DBSCAN original sofre do problema de falta de robustez que também

importuna os métodos hierárquicos de clusterização que utilizam todos os objetos: pode acabar por juntar os dois clusters ou atribuir pontos de bordas a cluster errados e crescer os clusters de forma errada perto da borda. Além disso, nestes casos, a clusterização final dependerá da ordem em que os objetos foram processados na fase de extensão do algoritmo.

O que acontece neste caso é que haverá pelo menos um ponto de borda compartilhando cadeias de pontos centrais alcançáveis por densidade originadas de dois clusters diferentes, ou seja, nestes casos os pontos de borda podem ser alcançados no algoritmo por diferentes caminhos, por diferentes pontos centrais na ε -vizinhança deste ponto de borda. Em outras palavras, um ponto de borda pode ser alcançado por densidade por cadeias de pontos centrais diferentes, potencialmente originárias de clusters diferentes. Não podendo ser assignado para os dois clusters, este ponto de borda será alcançado pela primeira cadeia visitada pelo algoritmo; e o ponto de borda será assignado ao cluster descoberto primeiro na expansão do algoritmo. Uma vez que o primeiro objeto central de cada cluster é qualquer objeto que cumpre a propriedade de objeto central, a ordem de descoberta dos clusters vai interferir totalmente no resultado final da clusterização, mostrando a fragilidade do método. Além disso, a fim de identificar clusters corretamente, objetos dos dados na área de contato poderão ser reconhecidos como pontos de borda. Daí como uma regra, quanto mais áreas de contato no espaço de dados, mais objetos de borda seriam detectados e uma vizinhança mais ampla poderia ser construída equivocadamente. Resumindo, o DBSCAN na presença de clusters densos e adjacentes pode produzir uma clusterização sensível à ordem de busca do algoritmo e equivocada: juntando dois clusters e/ou atribuindo pontos de bordas a cluster errados e/ou crescendo os clusters de forma errada perto da borda.

Em TRAN et al. (2013) os autores revisaram o conceito do método DBSCAN e ajustaram o algoritmo para alcançar uma performance mais robusta contra este problema. O resultado estende a aplicabilidade do algoritmo para muitos tipos de dados e supera o problema de amostras de fronteira pertencentes a grupos adjacentes. Nesta tese, o DBSCAN será utilizado para clusterizar dados muito densos, as d séries de tamanho N; para tal, esta robustez se faz necessária e por isso essa nova versão do DVSCAN Revisado é a versão utilizada para os resultados desta tese.

A cadeia de densidade alcançável de objetos {p_1, p_2, ..., p_n}, pode estar ou na forma [$p_{central_1}$p, $p_{central_2}$, ..., $p_{central_{n-1}}$, $p_{central_n}$] com todos os objetos centrais ou [$p_{central_1}$p, $p_{central_2}$, ..., $p_{central_{n-1}}$, p_{borda}] com todos os objetos centrais, exceto o último objeto sendo um objeto de borda. O ponto de fronteira, portanto, não contribui para o mecanismo de expansão das cadeias de objetos alcançáveis por densidade, o passo essencial de DBSCAN. Por esta razão, o melhoramento proposto por TRAN et al. (2013) visa desconectar os últimos objetos de borda da cadeia de pontos alcançáveis por densidade. Isto pode ser conseguido através da exploração de um novo conceito, os chamados "objetos centrais alcançáveis por densidade".

Definição 9 (Objetos Centrais Alcançáveis por Densidade): Objetos Centrais Alcançáveis por Densidade são cadeias de objetos {p_1, p_2, ..., p_n},, onde p_i é objeto central para todo $i \leq n$.

A partir daí, a etapa de expansão de agrupamento do algoritmo é revisto; a fim de usar somente as cadeias de objetos centrais alcançáveis por densidade (ou seja, somente as cadeias que contem apenas objetos centrais) ao invés das tradicionais cadeias de objetos alcançáveis por densidade. Uma vez que eles contêm um número semelhante de objetos centrais, serão identificados os mesmos objetos centrais para cada cluster. No entanto, os objetos de borda (originalmente atribuído durante o passo de expansão do cluster) permanecem temporariamente não classificados até que todos os objetos centrais de todos os clusters são identificados. Somente após a detecção de todos os grupos, durante o último passo da DBSCAN Revisado, cada objeto de borda é atribuído à sua melhor cadeia de objetos centrais alcançáveis por densidade. Uma alternativa lógica é a cadeia alcançável por densidade mais próxima (por exemplo, com o objeto central mais próximo ao objeto de borda considerado) e o objeto borda é atribuído a um cluster ao qual a cadeia de objetos centrais alcançáveis por densidade pertence. As rotinas do pseudocódigo do DBSCAN Revisado são apresentadas nas Figuras 5.7 e 5.8.

TRAN et al. (2013) salientam que para conjunto de dados com clusters bem definidos, clusters distantes um dos outros o DBSCAN Revisado coincide com o

DBSCAN porque neste caso não existirão objetos de borda compartilhando a cadeia de pontos centrais conectados por densidade dos clusters "adjacentes' pois não haverá clusters adjacentes. Mas em situações onde os clusters estão muito próximos e os dados são muito densos, o DBSCAN Revisado supera o algoritmo original e fornece uma melhor atribuição de um objeto de borda para o cluster esperado (ao ter informações de todos os grupos adjacentes) e resolvendo com sucesso a questão da objetos de borda e sua atribuição. Além disso, os autores garantem que os clusters obtidos pelo DBSCAN Revisado também independem da ordem em que os clusters são descobertos.

```
Algoritmo DBSCAN (DataBase, Eps, MinPts)
   // Todos os pontos em DataBase estão Unclassified

    N:= Tamanho da Base de Dados;
    ClusterId := 1;

    For i from 1 to N do
         If x_i in ClusterId= Unclassified;
           ExpandCoreCluster (x_i, ClusterId, Eps, MinPts);
           If ExpandCoreCluster == Expansion Successful
              ClusterId := ClusterId+1;
           END If
        END If
    END For

   // Etapa 3
   For x_border in Border List
      N_Eps (x_Border);
      Assign x_Border to ClusterId do ponto central mais próximo em N_Eps(x_Border)
   End For

END Algoritmo DBSCAN
```

Figura 5.7: Pseudo Código do Algoritmo DBSCAN- Revisado / Subrotina Principal.

```
Algoritmo ExpandCoreCluster (x_i,ClusterId, Eps, MinPts)

  Seeds:= N_Eps (x_i)

  If |Seeds|< MinPts            //Density(x_i)
    Mark x_i as Ruído;          // x_i está Classified
    Return Expansion without success;
    Else       // x_i is Ponto Central e está Classified

    // Passo 1: x_i é identificado como um ponto central
inicial para o Cluster ClusterId
    // Passo 2: Identifica todos objetos alcançáveis por
densidade a partir de x_i

     For all x_j in Seeds List
         N_Eps (x_j)
         If | N_Eps (x_j) |≥ MinPts     //x_j é um ponto central

     //mais expansão de cluster somente para o ponto central

     Assign x_j to ClusterId        //somente pontos centrais
     Add all UNCLASSIFIED objects N_Eps (x_j) to Seeds List;
     Label  UNCLASSIFIED  and NOISE in N_Eps (x_j) as Border;

          End If;
     End For;

     Return Expansion Successfull;

  End If;
  End Algoritmo ExpandCoreCluster;
```

Figura 5.8: Pseudo Código do Algoritmo DBSCAN- Revisado / Subrotina de Expansão dos Clusters.

5.3 Avaliações de Performance e Aplicações do DBSCAN

Em ESTER et al. (1996) os autores compararam a eficiência do DBSCAN e do CLARANS aplicados a dados sintéticos com ruídos e com clusters de diferentes tamanhos, formatos arbitrários e não convexos. O DBSCAN encontrou corretamente todos os clusters identificando seus formatos distintos e detectou todos os pontos de ruído, superando a clusterização feita pelo CLARANS que não foi capaz de separar os ruídos e dividiu de forma equivocada vários clusters. As eficiências dos métodos foram comparadas quando estes foram aplicados a uma base de dados reais do banco de

referências SEQUIOA 2000. Os resultados dos experimentos mostraram que o tempo de rotina do DBSCAN foi levemente maior que linear no número de objetos. Porém, o tempo de rotina do CLARANS foi quase quadrático no número de pontos. Os resultados mostraram que o DBSCAN supera o CLARANS por um fator entre 250 a 1900 vezes, que cresce com o crescimento do tamanho da base de dados.

Em MUNTAZ & DURAISWAMY (2010) os autores comparam a eficiência do DBSCAN com os seguintes métodos de clusterização: SOM (Mapas Auto-Organizáveis de Kohonen, baseado em redes neurais artificiais) e o famoso método k-means, para identificarem de clusters de diferentes tamanhos; formatos arbitrários, não convexos e espiralados. A Figura 10 a seguir, publicada pelos autores, mostra o desempenho superior do DBSCABN em identificar clusters naturais, de formatos arbitrários corretamente, diferente dos método SOM e k-means.

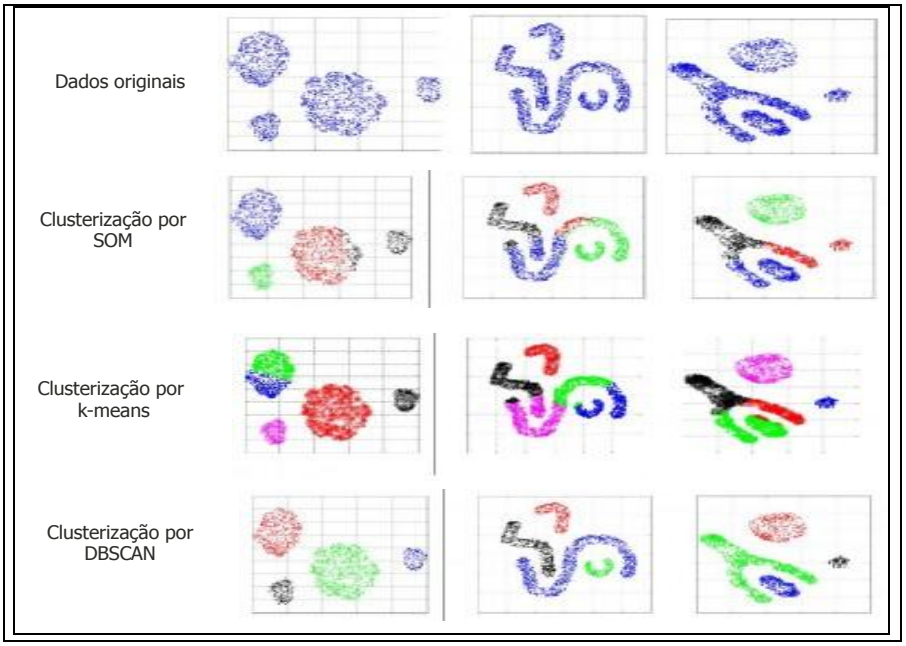

Figura 5.9: Desempenho de Diferentes Métodos de Clusterização para Dados Espaciais
Fonte: MUNTAZ & DURAISWAMY (2010).

Uma importante comparação entre DBSCAN e variações do DBSCAN é feita por ALI et al. (2010). Neste trabalho os autores pesquisaram algumas técnicas importantes nas quais o original DBSCAN é modificado. Muitas versões modificadas do DBSCAN trazem resultados melhores na complexidade de tempo ou no tratamento de dados com densidades variadas devido à utilização do sistema de indexação. Na maioria dos experimentos o DBSCAN Linear baseado no LSH proposto por ZHANG et al. (2007) e

o FAST-DBSCAN de LIU (2006) se mostraram como os mais eficientes para grandes bancos de dados. ALI et al. (2010) lembram que embora alguns métodos melhorem a complexidade do tempo para dados complexos, quando o DBSCAN não demanda muito tempo para execução, usá-lo é mais vantajoso pela simplicidade. ESTER *et al.* (1996) colocam que o método DBSCAN é eficiente mesmo para base de dados espaciais, e que ele é um dos mais eficientes algoritmos em bases de dados grandes (ESTER *et al.*, 1998).

Em PARIMALA et al. (2011) os autores comparam os métodos *DBSCAN, VDBSCAN, DVBSCAN, ST-DBSCAN e DBCLASD* e verificam que o DBSCLAD supera o DBSCAN na clusterização de dados sintéticos mas não identifica o ruído eficientemente. Em NAGPAL & MANN (2011), o DBCLASD, DENCLUE e DBSCAN foram usados para clusterização de um conjunto de dados da Flor Iris com quatro atributos largura do pedúnculo, comprimento do talo, largura da pétala e comprimento de pétala. Para cada método, a complexidade, a forma dos Clusters obtidos, a facilidade em lidar com os parâmetros de entrada, a manipulação de ruído do método, a qualidade do Cluster e o tempo de execução foram observados. Os autores concluíram que o tempo de execução do algoritmo é menor para o DENCLUE enquanto para o DBCLASD o tempo de execução é o mais elevada (5 vezes maior que o tempo do DENCLUE). Em termos de qualidade de cluster os autores elegeram o DBCLASD como o melhor método de clusterização. No tratamento de ruídos o DBSCAN fica em primeiro lugar em eficiência e o DBCLASD em segundo lugar. No geral, os autores concluem que o DBCLASD e DENCLUE são superiores ao DBSCAN. Mas em casos onde o tempo não é um problema na execução do DBSCAN e para dados om ruído, recomenda-se usar o DBSCAN.

Alguns exemplos de aplicações do DBSCAN incluem aplicações em diversos campos. Em Engenharia Civil, DBSCAN foi usado para agrupar redes de infraestrutura civil espaciais em OLIVEIRA et al. (2011); em Química DASZYKOWSKI et al. (2001) usaram DBSCAN para reconhecer padrões naturais nos dados; na área de Espectrocopia ZHOU et al. (2006) clusterizam por DBSCAN espectros de massa de partículas simples e também em ZHAO et al. (2008) os autores clusterizam espectros de massa do tempo de voo aerosol por DBSCAN; em Biologia, LIU et al. (2004) fizeram agrupamento de colônia de formigas por DBSCAN; em Ciências Sociais, GHOSH et al. (2008) usaram

DBSCAN para clusterizar dados de feromônios químicos. Na área médica de diagnósticos PLANT et al. (2010) utilizaram DBSCAN para detecção automática de padrões de atrofias cerebrais para predição da doença de Alzheimer; CELEBI et al. (2005) utilizaram DBSCAN para clusterização de imagens biomédicas; e METE et al. (2011) utilizaram DBSCAN para detectar padrões de lesões dermatológicas. DBSCAN também pode ser aplicado no campo de detecção remota para realizar segmentação de imagens em três dimensões (imagens hiperespectrais), como foi feito por GONG et al. (2008).

TRAN et al. (2012) aplicaram a metodologia DSBCAN na segmentação de imagem 3-D de raios-X de microtomografia. O algoritmo levou em consideração o sistema de coordenadas dos dados da imagem para melhorar o desempenho computacional e corrigir o problema de instabilidade na detecção de objetos nos limites do DBSCAN. Outra aplicação relevante do DBSCAN foi feita por EDLA & JANA (2012), onde os autores usaram o DBSCAN na clusterização de genes.

A aplicação mais recente de DBSCAN foi feita por MATEO et al. (2013) onde os autores comparam diversos métodos de aprendizagem de máquina para previsão de temperatura ambiente. Neste trabalho os autores avaliaram o DBSCAN, o k-means, o Fuzzy C-means, o Cumulative Hierarchical Tree e o k-medoids. O DBSCAN destacou-se como o melhor método de clusterização dentre os investigados. Também em 2013 o trabalho de CASSISI et al. (2013) faz uso do DBSCAN, propondo um algoritmo que remove ouliers para melhorar a performance do DBSCAN, as análises são feitas com dados sintéticos de diferentes densidades.

Em TRAN et al (2013) os autores comparam o DBSCAN Revisado com o DBSCAN para clusterizar dados simulados com dois clusters muito densos e adjacentes. Nesta situação, o DBSCAN se torna sensível à ordem em que os pontos da base são visitados e fornece diferentes clusters em diferentes execuções do algoritmo. Ao contrário, o DBSCAN Revisado se mostra robusto e eficiente nesta situação determinando uma fronteira única para os clusters.