Learning and Nonlinear Models (L&NLM) – Journal of the Brazilian Computational Intelligence Society, Vol. 10, Iss. 4, pp. 242-262, 2012

PROPOSTA DE UM MÉTODO DE CLASSIFICAÇÃO BASEADO EM DENSIDADE PARA A DETERMINAÇÃO DO NÚMERO IDEAL DE GRUPOS EM PROBLEMAS DE CLUSTERIZAÇÃO

Gustavo Silva Semaan[1] , Marcelo Dib Cruz[2], José André de Moura Brito[3] , Luiz Satoru Ochi[1]

[1] Instituto de Computação - Universidade Federal Fluminense (IC-UFF)
[2] Departamento de Matemática - Universidade Federal Rural do Rio de Janeiro (DEMAT-UFRRJ)
[3] Escola Nacional de Ciências Estatísticas (ENCE-IBGE)

{gsemaan@ic.uff.br, dib@ufrrj.br, jose.m.brito@ibge.gov.br, satoru@ic.uff.br}

Resumo. A área de *Cluster Analysis* agrega diversos métodos de classificação não supervisionada que podem ser aplicados com o objetivo de identificar grupos dentro de um conjunto de dados, supondo fixado o número de grupos e uma função objetivo, ou identificar o número ideal de grupos mediante avaliação de algum índice ou coeficiente. Em particular, o presente trabalho traz a proposta de um novo método de classificação denominado MRDBSCAN, que foi concebido a partir de uma calibração dos valores de parâmetros que são utilizados no conhecido método DBSCAN, que trabalha com o conceito de densidade. A qualidade das soluções obtidas é indicada pelo coeficiente silhueta, que combina coesão e separação. Os resultados apresentados neste estudo indicam que o método proposto é de fácil implementação e é competitivo em relação à qualidade das soluções quando comparado com os algoritmos mais sofisticados da literatura.

Palavras Chave: Problema de Agrupamento Automático, Densidade, Silhueta, Algoritmo DBSCAN.

1 - INTRODUÇÃO

A resolução do problema de agrupamento de dados consiste na classificação não supervisionada de objetos em grupos (*clusters*), não sendo necessário um conhecimento prévio sobre as suas classes ou categorias [Jain and Dubes, 1988]. Seu objetivo é obter grupos que apresentem padrões (características) semelhantes e que possam refletir a forma como os dados são estruturados. Para isso, deve-se maximizar a similaridade (homogeneidade) entre os objetos de um mesmo grupo e minimizar a similaridade entre objetos de grupos distintos [Han and Kamber, 2006] [Larose, 2005] [Goldschimidt and Passos, 2005].

Formalmente, este problema pode ser definido da seguinte maneira: dado um conjunto formado por *n* objetos $X = \{x_1, x_2, ..., x_i, ..., x_n\}$, com cada objeto x_i possuindo *p* atributos (dimensões ou características), ou seja, $x_i = \{x_i^1, x_i^2, ..., x_i^p\}$, deve-se construir ***k*** grupos C_j $(j = 1, ..., k)$ a partir de X, onde os objetos de cada grupo sejam homogêneos segundo alguma medida de similaridade. Além disso, devem ser respeitadas as restrições concernentes a cada problema particular abordado [Han and Kamber, 2006] [Ester et al., 1995] [Baum, 1986] [Hruschka and Ebecken, 2001] [Dias and Ochi, 2003]. No presente trabalho será abordado o problema clássico de agrupamento definido pelas restrições:

$$\bigcup_{i=1}^{k} C_i = X \tag{1}$$

$$C_i \bigcap C_j = \varnothing \text{ i,j=1, .., k e i} \neq \text{j} \tag{2}$$

$$C_i \neq \varnothing \text{ i=1, .., k} \tag{3}$$

Estas restrições determinam, respectivamente, que: O conjunto ***X*** corresponde à união dos objetos dos grupos, cada objeto pertence a exatamente um grupo e todos os grupos possuem ao menos um objeto.

Para este problema, o número de soluções possíveis, ou seja, o total de maneiras em que os *n* objetos podem ser agrupados, considerando um número fixo de *k* grupos, é dado pelo número de *Stirling* (*NS*) de segundo tipo [Jr, 1968], e podem ser obtidas pela Equação 4 [Liu, 1968]. Para problemas de agrupamento em que o valor de *k* é desconhecido (agrupamento automático), o número de soluções possíveis aumenta ainda mais. Este número é dado pela Equação 5, que corresponde ao somatório da Equação 4 para o número de grupos variando no intervalo $[1, k_{max}]$, sendo k_{max} o número máximo de grupos. Para que se tenha uma ideia da ordem de grandeza deste número, no caso de *n*=10 objetos a serem alocados em *k*=3 grupos, o número de soluções a serem consideradas é de 9.330. Mas considerando apenas dobro de objetos, ou seja, *n*=20 e *k*=3, o número de soluções possíveis (Equação 4) sobe para 580.606.446. No problema de agrupamento automático estes valores crescem exponencialmente com o aumento da quantidade de objetos (*n*). Esta característica torna proibitiva a obtenção

da solução ótima mediante a aplicação de um procedimento de enumeração exaustiva. Esta questão é comentada em vários trabalhos da literatura, como por exemplo, no trabalho de Naldi (2011).

$$NS(n,k) = \frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}j^n \tag{4}$$

$$NS(n) = \sum_{j=1}^{k_{\max}} NS(n,j) \tag{5}$$

Conforme Kumar et. al.(2009), as últimas décadas foram marcadas pelo desenvolvimento de diversos algoritmos de agrupamento. Por sua vez, estes algoritmos encontram aplicação em diversos domínios, como por exemplo: inteligência artificial, reconhecimento de padrões, marketing, economia, ecologia, estatística, pesquisas médicas, ciências políticas, etc. Não obstante, nenhum desses algoritmos é apropriado para todos os tipos de dados, formatos de grupos e aplicações. Esta última observação sugere que há espaço para o estudo e o desenvolvimento de novos algoritmos de agrupamento que sejam mais eficientes ou mais apropriados, levando em conta as características específicas de conjuntos de dados. Em muitos casos, inclusive, a análise de "*o que é uma boa solução*" é subjetiva.

O presente artigo está estruturado em cinco seções, incluindo a introdução. A seção dois apresenta uma revisão da literatura em relação a algoritmos para a obtenção da quantidade ideal de grupos. É apresentado também o índice relativo silhueta, utilizado para avaliar as soluções obtidas com a aplicação do método proposto. Na seção três são apresentados o clássico algoritmo da literatura DBSCAN e a técnica para a seleção automática de parâmetros para esse algoritmo. Já a seção quatro apresenta as instâncias utilizadas, os resultados obtidos nos experimentos computacionais e os comparativos com algoritmos da literatura. Por fim, a seção cinco relata conclusões obtidas nas pesquisas e apresenta propostas de pesquisas e de trabalhos futuros.

2 - REVISÃO DA LITERATURA

Segundo [Kumar et. al., 2009], talvez um dos problemas de seleção de parâmetros mais conhecido seja o de determinar o número ideal de grupos em um problema de agrupamento. Neste sentido, diversas técnicas não supervisionadas de avaliação de soluções podem ser utilizadas.

Uma dessas técnicas consiste analisar o valor da Soma dos Erros Quadráticos (SEQ, Equação 6) das soluções obtidas em função do número de grupos. O objetivo é encontrar a quantidade natural de grupos, procurando por uma quantidade de grupos em que exista uma inflexão no valor do SEQ. Essa abordagem pode falhar em algumas situações, quais sejam: quando existem grupos entrelaçados, superpostos ou até mesmo aninhados. Na Equação 6, $dist(c_i, x)$ indica a distância (Euclidiana: Equação 7) entre o objeto *x* e o centróide a ele mais próximo (c_i).

$$SEQ = \sum_{i=1}^{k}\sum_{x \in C_i} dist(c_i, x)^2 \tag{6}$$

Essa primeira abordagem pode ser realizada, por exemplo, com a aplicação de um algoritmo clássico de agrupamento denominado *k*-Means (Han and Kamber, 2006), considerando que *k* é um inteiro que assume todos os valores no intervalo [2,n]. Dessa forma, aplica-se o algoritmo de agrupamento para cada valor de *k* e, em seguida, calcula-se o valor de *SEQ* para cada uma nas *(n - 1)* soluções obtidas. A partir destes valores, torna-se possível construir um gráfico *SEQ* versus o número de grupos, conforme apresenta a Figura 1.

É importante destacar que o algoritmo *k-Means* é sensível à seleção de protótipos (objetos ou centróides) iniciais. Ou seja, uma seleção aleatória desses protótipos para a formação dos grupos iniciais do algoritmo geralmente resulta em agrupamentos pobres, de baixa qualidade (Kumar et. al., 2009) no que concerne à estrutura ou ao valor da similaridade. Dessa forma, recomenda-se que para cada numero *k* de grupos no intervalo estabelecido (nessa análise [2,n]), esse algoritmo seja executado considerando diferentes protótipos iniciais. Em seguida, são consideradas para a análise somente a melhor solução (menor SEQ) para cada número *k* de grupos.

Outra abordagem concernente à determinação do número ideal de grupos consiste na avaliação da função silhueta proposta por Rousseeuw (1987) e utilizada em diversos trabalhos, dentre os quais: Naldi (2011), Cruz (2010), Wang et. al. (2007), Soares (2004) e Tseng and Yang(2001). Mais especificamente, aplica-se um algoritmo de agrupamento para alguns valores de *k* no intervalo [2,n] , escolhendo-se como o *k* ideal aquele associado ao maior valor da função-silhueta (Figura 1). Uma vez que esse trabalho utilizou a função silhueta, há uma seção específica para a descrição detalhada da mesma.

Ainda em relação à análise da função índice silhueta, também é possível executar tal abordagem considerando múltiplas execuções do algoritmo *k-Means*. Novamente, aconselha-se a executar esse algoritmo com diferentes inicializações de protótipos para cada número de grupos e, ao final, deve-se utilizar a função silhueta para avaliar as soluções obtidas para cada *k*

Com base no algoritmo *k-Means*, foi proposto por [Pelleg and Moore, 2000] o algoritmo *X-Means* para a resolução do problema de agrupamento automático. Este algoritmo recebe como parâmetros a instância a ser processada e um intervalo com a quantidade de grupos [k_{min}, k_{max}] e utiliza o índice BIC (*Bayesian Information Criterion*) para identificar e retornar qual o melhor número de grupos. Em [Zalik, 2008] é apresentado um algoritmo que também adapta o *k-Means* para resolver um problema de agrupamento automático.

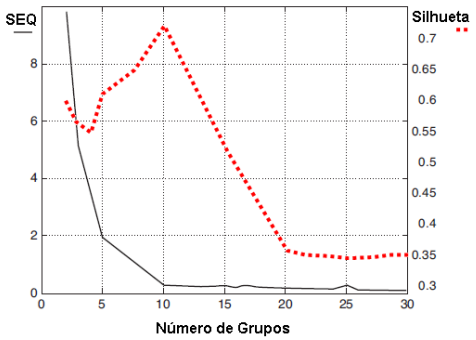

Figura 1: *SEQ* versus Número de Grupos e Silhueta versus Número de Grupos (adaptação de [Kumar et. al., 2009])

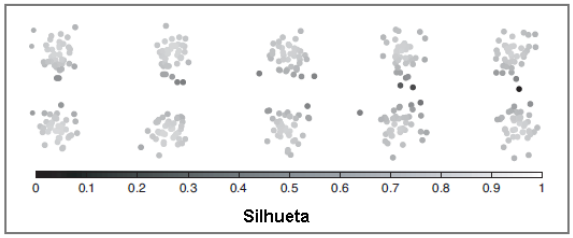

Figura 2: Instância associada ao gráfico da Figura 1 ([Kumar et. al., 2009]).

O *Bisecting k-Means*, proposto por Steinbach et. al. (2000), corresponde a uma versão hierárquica do *k-Means*, em que a cada iteração, um grupo é selecionado e dividido em dois novos grupos. Dessa forma, novamente são obtidas soluções para todos os valores de *k* pertencentes a um intervalo de *k* pré-estabelecido. O critério de seleção do grupo a ser dividido pode ser, por exemplo, o grupo com maior diâmetro (distância entre dois objetos em um mesmo grupo) ou o grupo com o menor valor de função silhueta.

Ainda no contexto do problema agrupamento automático, vários trabalhos na literatura propõem algoritmos baseados em metaheurísticas que têm por objetivo encontrar um número ideal de grupos e a sua solução correspondente. Dentre estes, destacam-se os seguintes trabalhos: [Soares, 2004] [Cruz, 2010] [Cole, 1998] [Cowgill, 1999] [Bandyopadhyay and Maulik, 2001] [Bandyopadhyay and Maulik, 2002b] [Hruschka and Ebecken, 2003] [Hruschka et. al., 2004a][Hruschka et. al., 2004b] [Hruschka et. al., 2006] [Ma et. al., 2006] [Alves et. al. 2006] [Tseng and Yang, 2001] [Naldi and Carvalho, 2007] [Pan and Cheng, 2007]

Existem, também, as heurísticas que utilizam alguns procedimentos de busca local baseados no algoritmo *k-Means*. Em um primeiro momento, essas heurísticas utilizam algoritmos para construção de grupos, denominados grupos parciais (temporários, componentes conexos) com o objetivo de unir os objetos mais homogêneos. Em seguida, são aplicados algoritmos de busca local e de perturbação sobre esses grupos produzindo soluções de boa qualidade [Cruz, 2010] [Tseng and Yang, 2001] [Hruschka et. al., 2004b] [Alves et. al. 2006] [Hruschka et. al., 2006] [Naldi and Carvalho, 2007].

Em [Tseng and Yang, 2001] foi apresentado um algoritmo genético denominado CLUSTERING, que também utiliza a função silhueta para determinar o número ideal de grupos. Para isso, esse algoritmo constrói um grafo, identifica os seus componentes conexos e atua no agrupamento desses componentes com o objetivo de maximizar a função silhueta.

O trabalho Soares [Soares, 2004] apresenta alguns algoritmos baseados nas metaheurísticas *Simulated Annealing* e Algoritmos Evolutivos para a resolução do problema de agrupamento automático. Este trabalho também propõe algoritmos para construção de soluções, perturbações e refinamentos (buscas locais), incluindo um procedimento de reconexão por

caminhos (*path relinking*) que atua na busca de soluções de melhor qualidade. Em seus experimentos foram realizadas algumas comparações com o algoritmo CLUSTERING [Tseng and Yang, 2001].

O algoritmo CLUES (*CLUstEring based on local Shirinking*) [Wang et. al., 2007] também aborda o problema do agrupamento automático, possibilitando a aplicação da função silhueta ou do índice CH (índice de *Calinski-Harabasz*) para a determinação do número ideal de grupos. Trata-se de um algoritmo iterativo que, com a utilização de um procedimento de encolhimento (*Shirinking procedure*) baseado nos *k*-vizinhos mais próximos, realiza a união dos objetos mais homogêneos segundo os seus atributos.

Após a aplicação do procedimento de encolhimento, o CLUES constrói soluções, avaliando-as mediante o valor da função de silhueta ou do Índice CH. Ainda em Wang et. al. (2007) é relatado que os resultados obtidos com a utilização da função de silhueta e do índice CH foram comparados. A partir dessa comparação, observou-se que mediante a aplicação do Índice Silhueta foram produzidas soluções de melhor qualidade no que concerne ao número de grupos definidos e à formação de soluções denominadas *perfeitas* em tal trabalho. Esse algoritmo foi desenvolvido em *R* e o seu código fonte está disponível em um pacote do software estatístico R.

O trabalho de Cruz [Cruz, 2010] traz uma proposta de algoritmos heurísticos mais sofisticados no que concerne aos procedimentos de construção, de busca local e de perturbação. Mais especificamente, estes algoritmos foram baseados nas metaheurísticas Algoritmos Genéticos, Busca Local Iterada (*Iterated Local Search*) e GRASP (*Greedy Randomized Adaptive Search Procedure*). O diferencial desses algoritmos está na incorporação de procedimentos para a construção de grupos parciais, conceitos de Memória Adaptativa e Buscas Locais que utilizam o algoritmo *k-means* para a união de grupos parciais.

Ainda no trabalho de Cruz (2010) foram propostos também métodos híbridos. Estes métodos utilizam algumas das soluções produzidas pelos algoritmos heurísticos, ou seja, soluções associadas com alguns valores de *k* e que sejam consideradas promissoras no que concerne ao número ideal de números, porém não necessariamente a melhor solução para tal número. Considerando estes valores específicos de *k*, são aplicadas duas formulações de programação inteira, quais sejam: para o problema de agrupamento com diâmetro mínimo e dos *k-Medoids* [Cruz, 2010]. Nos experimentos apresentados neste trabalho foram realizadas comparações com o algoritmo da literatura CLUES [Wang et. al., 2007].

O presente trabalho propõe um método de classificação baseado em densidade que tem por objetivo a identificação do número ideal de grupos. Ou seja, identificar de forma não supervisionada padrões semelhantes e que possam refletir na forma como os dados são estruturados. O método proposto consiste na aplicação de um algoritmo de agrupamento clássico baseado em densidade DBSCAN (***D**ensity **B**ased **S**patial **C**lustering of **A**pplications with **N**oise*) [Ester et. al., 1996]. Esse algoritmo necessita de dois parâmetros, sejam eles: a distância e a densidade (quantidade de objetos no raio de alcance de um objeto, incluindo o próprio objeto). Para a calibração desses parâmetros foi utilizada a técnica proposta na literatura denominada *DistK* e descrita na seção 3.2.

É importante ressaltar que o DBSCAN foi adaptado para que todos os objetos que compõem uma instância sejam considerados. Essa modificação decorre do fato de o algoritmo DBSCAN tradicional classificar os objetos em *Interiores*, *Limítrofes* e *Ruídos* e, os objetos classificados como *Ruídos* serem ignorados pelo algoritmo em sua versão original.

Como o algoritmo DBSCAN é determinístico, foram obtidos diferentes valores para cada um de seus parâmetros com o objetivo de encontrar soluções diversificadas no que diz respeito ao número de grupos e à distribuição dos objetos nesses grupos. Conforme Naldi [Naldi, 2011], os índices de validação relativos têm sido utilizados e investigados extensivamente, tendo estes apresentado resultados satisfatórios em diversos cenários.

Os índices relativos, como próprio nome sugere, têm como finalidade avaliar a qualidade relativa das soluções produzidas por diferentes métodos de agrupamento. Estes índices não têm a propriedade de monotonicidade, ou seja, não são afetados pelo aumento ou pela redução do número de grupos da solução. Dessa forma, podem ser utilizados na avaliação de diversas soluções, provenientes de diversos algoritmos.

No presente trabalho, assim como nos algoritmos da literatura considerados nos experimentos, as soluções obtidas são avaliadas pelo índice de silhueta, que é um índice relativo. Ou seja, buscar-se-á a resolução de um problema de otimização cuja função deve ser maximizada.

2.1 - A Silhueta

O Índice Silhueta foi proposta por Rousseeuw [Rousseeuw, 1987]. Esta medida determina a qualidade das soluções com base na proximidade entre os objetos de determinado grupo e na distância desses objetos ao grupo mais próximo. O índice silhueta é calculado para cada objeto, sendo possível identificar se o objeto está alocado ao grupo mais adequado. Esse índice combina as ideias de coesão e de separação. Os quatro passos a seguir explicam, brevemente, como calculá-lo:

1. Neste trabalho d_{ij} (Equação 7) corresponde à distância euclidiana entre os objetos *i* e *j,* e *p* é a quantidade de atributos dos objetos. Para cada objeto x_i calcula-se a sua distância média $a(x_i)$ (Equação 8) em relação a todos os demais objetos do mesmo grupo. Na Equação 8, $|C_w|$ representa a quantidade de objetos do grupo C_w , ao qual o objeto x_i pertence.

$$d_{ij} = \sqrt{\sum_{x=1}^{p} (i_x - j_x)^2} \tag{7}$$

$$a(x_i) = \frac{1}{|C_w| - 1} \sum d_{ij} \quad \forall x_j \neq x_i, \quad x_j \in C_w \tag{8}$$

2. A Equação 9 apresenta a distância entre o objeto x_i e os objetos do grupo C_t, em que $|C_t|$ é a quantidade de objetos do grupo C_t . Para cada objeto x_i calcula-se a sua distância média em relação a todos os objetos dos demais grupos ($b(x_i)$) (Equação 10).

$$d(x_i, C_t) = \frac{1}{|C_t|} \sum d_{ij} \quad \forall x_j \in C_t \tag{9}$$

$$b(x_i) = \min d(x_i, C_t) \quad C_t \neq C_w \quad C_t \in C \tag{10}$$

3. O coeficiente silhueta do objeto x_i ($s(x_i)$) pode ser obtido pela Equação 11.

$$s(x_i) = \frac{b(x_i) - a(x_i)}{\max\{b(x_i), a(x_i)\}} \tag{11}$$

4. O cálculo da silhueta de uma solução S é a média das silhuetas de cada objeto, conforme apresenta a Equação 12, em que *n* é a quantidade de objetos da solução. Essa função deve ser maximizada.

$$Silhueta(S) = \frac{1}{n} \sum_{i=1}^{n} s(x_i) \tag{12}$$

Os valores positivos de silhueta indicam que o objeto está bem localizado em seu grupo, enquanto valores negativos indicam que o objeto está mais próximo de outro(s) grupo(s). A Figura 2 apresenta um exemplo gráfico de uma solução constituída por dez grupos e objetos com duas dimensões. As cores dos objetos indicam as suas silhuetas e, quanto mais escuro o tom de cinza, menor o valor da silhueta (próximo de zero). Observa-se que nesse exemplo nenhum objeto possui a silhueta negativa.

Conforme Naldi [Naldi, 2011], este índice é mais apropriado para agrupamentos volumétricos, com grupos gerados de acordo com distribuições Gaussianas multidimensionais hiperesféricas ou moderadamente alongadas, porém ele não obteve bons resultados para grupos com formatos arbitrários [Rousseeuw, 1987].

Em [Hruschka et. al., 2004a] é proposta uma versão simplificada do índice de silhueta. Nesta versão são efetuadas modificações nos cálculos de $a(x_i)$ e $b(x_i)$ com o objetivo de reduzir a complexidade do algoritmo de $O(n^2)$ para $O(n)$. Segundo

os autores desse trabalho, mesmo com a redução da complexidade, esse novo índice mantém a qualidade próxima ao da silhueta tradicional, o que é confirmado por [Vendramin et. al., 2009] [Vendramin et. al., 2010].

3 – O MÉTODO PROPOSTO

Com o objetivo de identificar o número ideal de grupos em cada instância, propõe-se no presente trabalho um método que consiste em uma variante do algoritmo DBSCAN [Ester et. al., 1996], denominado MRDBSCAN (do inglês *Multiple Runs of DBSCAN*). Mais especificamente, a partir dos parâmetros iniciais do DBSCAN são considerados diferentes valores de entradas, determinados a partir de uma técnica denominada *Distk*, técnica essa baseada nas distâncias dos *k-vizinhos* mais próximos a cada objeto. As soluções obtidas são avaliadas com a utilização do índice relativo de silhueta, que deve ser maximizado. Consequentemente, as soluções com os maiores valores para esse índice são consideradas de melhor qualidade, sendo os números de grupos (valores de k) associados a essas soluções apresentados como os ideais. De forma a facilitar o entendimento dessa nova variante, apresenta-se a seguir (subseção 3.1) uma descrição concisa do algoritmo DBSCAN.

3.1 - Algoritmo DBSCAN

Os algoritmos de agrupamento baseados em densidade têm como objetivo a determinação de grupos (regiões) de alta densidade de objetos separados por regiões de baixa densidade. Nesse contexto, o algoritmo DBSCAN [Ester et. al., 1996] é um dos mais conhecidos da literatura e possui uma complexidade computacional $O(n^2)$. Trata-se de um algoritmo simples, eficiente, e que contempla conceitos importantes, que servem de base para qualquer abordagem baseada em densidade.

O DBSCAN utiliza-se de um conceito de densidade tradicional baseada em centro, ou seja, a densidade de um objeto x_i é a quantidade de objetos em um determinado raio de alcance de x_i, incluindo o próprio objeto. Este algoritmo possui como parâmetros de entrada o raio (*raioDBSCAN*) e a quantidade mínima de objetos em um determinado raio (*qtdeObjetos*). Assim, a densidade de um objeto depende do raio especificado. Deve-se, então, calibrar o parâmetro *raioDBSCAN* para que o seu valor não seja tão alto de forma que todos os objetos tenham densidade *n* (solução com apenas um grupo), e nem tão baixo em que todos os objetos terão densidade 1 (solução com *n* grupos denominados *singletons*). A abordagem da densidade baseada em centro realiza a classificação dos objetos em:

- **Interiores ou Centrais**: objetos que pertencem ao interior de um grupo baseado em densidade. Deve possuir uma quantidade de objetos em seu raio *raioDBSCAN* igual ou superior ao parâmetro *qtdeObjetos - 1*.
- **Limítrofes**: não é um objeto central, mas é alcançável por ao menos um objeto central, ou seja, está dentro do raio de vizinhança de algum objeto central.
- **Ruídos**: demais objetos que não são Centrais e nem estão na vizinhança de um objeto central.

Para a aplicação do algoritmo DBSCAN são considerados os seguintes passos:

1. Classificar os objetos como Objetos *Centrais*, *Limítrofes* ou *Ruídos*.
2. Eliminar os objetos que sejam classificados como *Ruídos*.
3. Adicionar arestas entre todos os Objetos Centrais que estejam dentro do *raioDBSCAN*.
4. Tornar cada grupo de Objetos de centro um grupo separado.
5. Atribuir cada Objeto limítrofe a um dos grupos dos seus objetos centrais associados.

Como base nestas informações, a Figura 3 ilustra a classificação dos objetos em *Ruído*, *Limítrofe* ou *Interior*. Essa mesma figura apresenta também uma solução obtida com a execução do DBSCAN, em que é possível observar que objetos identificados como dos tipos *Interior* ou *Borda* formam grupos enquanto objetos do tipo *Ruído* permanecem isolados e não fazem parte de nenhum grupo.

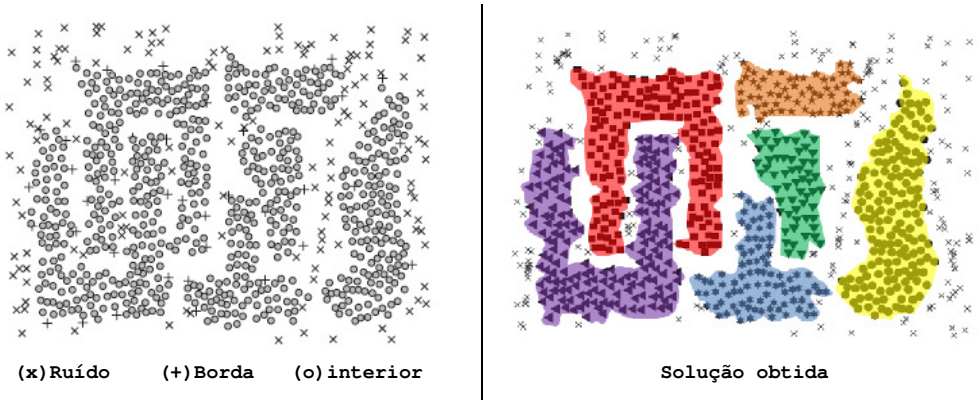

Figura 3: classificação de 3000 objetos de duas dimensões pelo DBSCAN [Kumar et. al., 2009]

Tendo em vista que o DBSCAN é um algoritmo baseado em densidade, o mesmo é imune a ruídos, uma vez que esses objetos são identificados e ignorados (não pertencem a grupos). Além disso, o algoritmo pode trabalhar com grupos de tamanhos (número de objetos) e formas arbitrárias. Dessa forma, ele é capaz de identificar grupos que não poderiam ser encontrados mediante a aplicação de outros algoritmos, como por exemplo, o *k*-means. Conforme foi comentado, o *K-means* tende a produzir grupos com formato hiperesférico, de tamanhos semelhantes e bem separados. Entretanto, ao aplicar-se o DBSCAN em instâncias que possuem densidades muito variadas, pode implicar na classificação dos objetos pertencentes a áreas de baixa densidade como ruídos. Este fato tende reduzir a qualidade dos resultados obtidos no que diz respeito à quantidade de grupos e ao índice silhueta.

3.2-Seleção de Parâmetros para o Algoritmo DBSCAN

Em [Kumar et. al., 2009] é apresentada uma abordagem para calibrar o raio apropriado, intitulada *Distk*. Esta abordagem consiste em, para um valor inteiro k^* fornecido como o parâmetro de entrada, analisar o comportamento das distâncias entre cada objeto e o seu vizinho de índice k^* mais próximo, ou seja, o seu k-ésimo vizinho mais próximo. O objetivo desse procedimento é identificar os valores de distâncias que resultariam em soluções de qualidade, obtidas mediante a execução do algoritmo DBSCAN.

Um valor baixo para a distância entre um objeto x_i e o seu vizinho de índice k^* mais próximo indica que esses objetos pertencem a um mesmo grupo, enquanto valores relativamente altos indicam que os objetos não pertencem ao mesmo grupo ou ainda, indica a ocorrência de objetos classificados como ruídos. A abordagem consiste, basicamente, nos passos a seguir:

1. Para cada objeto x_i, obter o seu vizinho mais próximo x_j de índice k^* e a distância d_{ij}.
2. Adicionar as distâncias entre esses objetos em um vetor de distâncias V_{dist}.
3. Ordenar V_{dist} de forma crescente e construir um gráfico Dist*K* com os valores de V_{dist}.
4. Identificar as grandes variações nos valores das distâncias de V_{dist}. Espera-se que uma mudança abrupta (inflexão) nesses valores corresponda a um valor apropriado para o parâmetro *raioDBSCAN*.

Para obter o *k-ésimo-vizinho* mais próximo de cada objeto x_i, as distâncias entre x_i e os demais objetos são adicionadas em um vetor, que deve ser ordenado (custo computacional $O(n \log n)$ em que n é a quantidade de objetos). Uma vez que a ordenação deve ser realizada para cada objeto, o custo computacional total para a obtenção dos parâmetros que devem ser informados ao algoritmo DBSCAN é $O(n^2 \log n)$, custo esse superior, inclusive, ao do algoritmo DBSCAN que é $O(n^2)$.

O algoritmo DBSCAN original utilizou $k^* = 4$. Segundo [Kumar et. al. 2009], esse é um valor razoável para a maioria dos conjuntos de dados bidimensionais. Porém, ainda é necessário identificar um valor interessante para o parâmetro *raioDBSCAN*. Com o objetivo de obter diferentes soluções para a calibragem do DBSCAN, conforme a abordagem *DistK*,

foram utilizados os valores de k* pertencentes ao conjunto B = {3,4,5,10,15,20,50}. Além disso, para a determinação do parâmetro *raioDBSCAN* foram utilizadas quatro regras, obtidas empiricamente através de experimentos preliminares, quais sejam:

1. **Mediana:** Considerar o valor da mediana obtida a partir de V_{dist}.
2. **Maior:** Considerar o maior valor de V_{dist}.
3. **Pico10:** Dividir o vetor V_{dist} em 10 partes com a mesma quantidade de distâncias e verificar a maior diferença entre V_{dist}[i+1] e V_{dist}[i], para i = {1,2, ..., 10}. A distância considerada será (V_{dist}[i+1]+V_{dist}[i]) / 2.
4. **Pico20:** Dividir o vetor V_{dist} em 20 partes com a mesma quantidade de distâncias e verificar a maior diferença entre V_{dist}[i+1] e V_{dist}[i], para i = {1,2, ..., 20}. A distância considerada será (V_{dist}[i+1]+V_{dist}[i]) / 2.

As quantidades de partes utilizadas pelas regras Pico10 e Pico20 foram determinadas com base em experimentos empíricos. Tendo em vista que as quantidades de partes são constantes (10 e 20 partes), a identificação de grandes variações de valores de V_{dist} ocorre em O(1) (constante).

Uma vez que foram utilizados sete valores distintos de k* e, para cada valor foram obtidas quatro distâncias de raio (*raioDBSCAN*), os experimentos realizados consideraram 28 configurações diferentes.

A Figura 4 apresenta um gráfico *DistK* para a instância 200DATA considerando k*=3. Com base neste gráfico, a linha que intercepta o eixo Y em um valor próximo de 3,0 corresponde à regra 2 (maior distância obtida). A linha que intercepta o eixo Y próximo de 0,5 corresponde à regra 1 (a mediana das distâncias obtidas), enquanto as linhas dos valores 2,1 e 1,9 representam, respectivamente, as distâncias obtidas mediante a aplicação das regras 3 e 4.

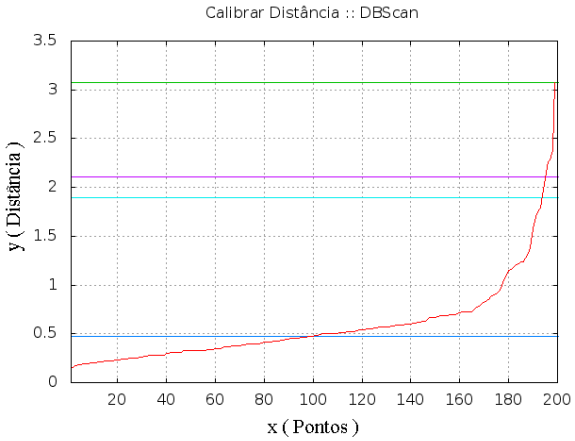

Figura 4: instância 200DATA *DistK* para K*=3.

4-Experimentos Computacionais

A presente seção traz um conjunto de resultados computacionais obtidos a partir da aplicação de alguns dos algoritmos citados na seção 2 e do novo método que utiliza o algoritmo DBSCAN (MRDBSCAN). Observa-se que os algoritmos da literatura foram implementados utilizando diferentes linguagens de programação, compiladores e foram executados em diferentes máquinas e sistemas operacionais. Além disso, alguns códigos fonte da literatura não estavam disponíveis até o momento da preparação desse trabalho. Em face destas observações, a comparação entre os algoritmos da literatura e o MRDBSCAN, no que concerne à sua performance, ficou restrita à qualidade das soluções com base na função silhueta e nas quantidades de grupos identificadas. Ou seja, os tempos de processamento estão disponibilizados apenas para o MRDBSCAN.

A implementação do MRDBSCAN foi feita em Linguagem C++, utilizando o paradigma de orientação a objetos, em um ambiente de desenvolvimento Eclipse for C/C++ Developers. Todos os experimentos computacionais foram realizados em um computador dotado de um processador i7 de 3.0 GHz e com 8GB de RAM e o sistema operacional Ubuntu 9.10, kernel 2.6.18.

É importante destacar que não foi explorada a capacidade de multiprocessamento do equipamento utilizado e não foi utilizado nenhum conhecimento prévio sobre as instâncias ou resultados obtidos por outros trabalhos da literatura. Os algoritmos da literatura para os quais resultados foram apresentados e comparados foram os seguintes:

- **CLUES (***CLUstEring based on local Shirinking***)** [Wang et. al., 2007]: implementado no software estatístico *R* [Matloff 2011] e disponível no pacote *clues*.
- **CLUSTERING** [Tseng and Yang, 2001]: implementação de um Algoritmo Genético em C++ realizada por [Soares, 2004].
- **SAPCA** (*Simulated Annealing*) e **AEC-RC** (Algoritmo Evolutivo com Reconexão por Caminhos): proposto e implementado em C++ por [Soares, 2004].
- **AECBL1** (Algoritmo Evolutivo com Busca Local), **GBLITRC1** (GRASP com Reconexão de Caminhos) e **IBLITRC1** (Busca Local Iterada com Reconexão de Caminhos) de [Cruz, 2010]: os melhores resultados obtidos para cada instância considerando os três algoritmos. Desenvolvimento feito em linguagem C++.

Em relação ao intervalo relacionado com o número de grupos, é uma prática comum em abordagens sistemáticas utilizar [2, k_{max}], sendo $k_{max} = n^{1/2}$ ([Pal and Bezdek, 1995][Pakhira et. al., 2005][Campello et. al., 2009]. Em particular, no MRDBSCAN esse intervalo foi considerado para indicar se o número de grupos torna válida ou não a solução, uma vez que o algoritmo não possui o parâmetro do número de grupos.

Para a realização dos experimentos foram utilizadas 83 instâncias da literatura que estão distribuídas em três conjuntos de (DS - *Datasets*). Estas instâncias possuem um número de objetos variando entre 30 e 2000, o número de dimensões (atributos) entre 2 e 60 e diferentes características relacionadas, por exemplo, com a coesão, à separação e às densidades dos grupos.

O primeiro conjunto (DS1), apresentado pela Tabela 1, possui 10 instâncias bem conhecidas da literatura com a quantidade de objetos entre 75 e 1484 e dimensões (quantidade de atributos) entre 2 e 60 [Fisher, 1936][Ruspini, 1970][Maronna and Jacovkis, 1974][Wang et. al., 2007][Hastie et. al., 2001][Naldi, 2011].

Tabela 1: Conjunto de Instâncias DS1

Instância	Nº Objetos	Dimensão
200DATA	200	2
chart	600	60
gauss9	900	2
iris	150	4
maronna	200	2
ruspini	75	2
spherical_4d3c	400	3
vowel2	528	2
wine	178	13
yeast	1484	7

O segundo conjunto (DS2), apresentado na Tabela 2, possui 51 instâncias que foram construídas por [Cruz, 2010] utilizando a ferramenta *Dots* desenvolvida por [Soares and Ochi, 2004]. Estas instâncias possuem uma quantidade de objetos entre 100 e 2000, sendo todas com duas dimensões e o número de grupos entre 2 e 27.

Nesse conjunto os nomes das instâncias foram definidos conforme a quantidade de objetos, de grupos, e se os grupos são bem definidos, coesos e separados (denominados "*comportados*" em Cruz (2010)).

A Figura 5 apresenta, respectivamente, as instâncias 200p4c e 300p4c1, em que 200p4c indica uma instância "*comportada*" com 200 objetos e 4 grupos e a instância 300p4c1 indica uma instância "*não comportada*" com 300 objetos e 4 grupos.

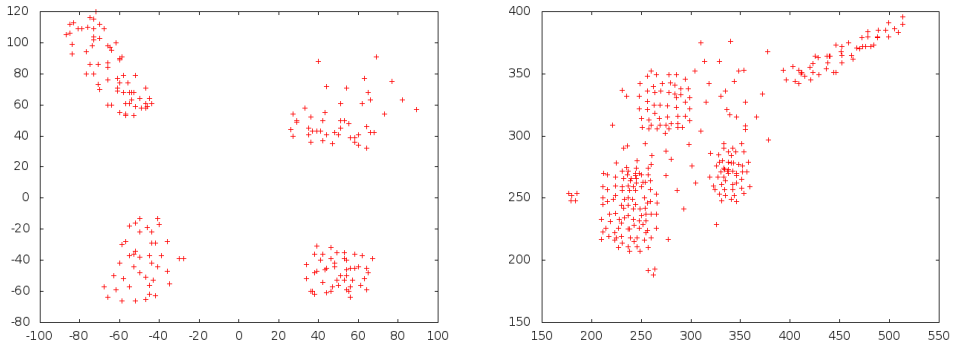

Figura 5: ilustrações das instâncias 200p4c e 300p4c1

Tabela 2: Conjunto de Instâncias DS2

Instância	Nº Objetos	Instância	Nº Objetos	Instância	Nº Objetos
100p10c	100	300p3c	300	800p23c	806
100p2c1	100	300p3c1	300	900p12c	900
100p3c	100	300p4c1	300	900p5c	900
100p3c1	100	300p6c1	300	1000p14c	1000
100p7c	100	400p3c	400	1000p5c1	1000
100p8c1	106	400p4c1	400	1000p6c	1000
100p5c1	110	400p17c1	408	1000p27c1	1005
100p7c1	112	500p19c1	500	1100p6c1	1100
200p2c1	200	500p3c	500	1300p17c	1300
200p3c1	200	500p4c1	500	1500p6c1	1500
200p4c	200	500p6c1	500	1800p22c	1800
200p4c1	200	600p15c	600	1900p24c	1901
200p7c1	210	600p3c1	600	2000p11c	2000
200p8c1	212	700p4c	700	2000p26c	2000
200p12c1	222	700p15c1	703	2000p9c1	2000
300p13c1	235	800p10c1	800		
300p10c1	300	800p18c1	800		
300p2c1	300	800p4c1	800		

O terceiro conjunto (DS3), apresentado pela Tabela 3, possui 11 instâncias que foram construídas e utilizadas por [Soares and Ochi, 2004][Soares, 2004]. Tais instâncias possuem quantidade de objetos entre 30 e 2000, sendo todas com duas dimensões.

Tabela 3: Conjunto de Instâncias DS3

Instância	Nº Objetos	Instância	Nº Objetos
30p	30	300p4c	300
outliers_ags	80	350p5c	350
97p	97	numbers	437
3dens	128	450p4c	450
Outliers	150	moreshapes	489
157p	157	500p3c	500
convdensity	175	numbers2	540
181p	181	600p3c	600
convexo	199	900p5c	900
2face	200	1000p6c	1000
Face	296	2000p11c	2000

No primeiro experimento são apresentados os resultados obtidos com a execução do DBSCAN nos três conjuntos de dados. Neste experimento são apresentados as quantidades de grupos das melhores soluções, o valor do índice silhueta e algumas estatísticas em relação aos tempos de execução.

Em um primeiro momento**,** o MRDBSCAN utiliza a técnica de calibração de parâmetros *DistK*. Nesse sentido, para a obtenção dos valores de *raioDBSCAN* são utilizadas as regras propostas no presente trabalho, quais sejam: *mediana*, *maior*, *Pico10* e *Pico20*. Uma vez que são considerados sete valores de *k** e quatro regras, são obtidas 28 configurações, sendo cada uma destas configurações correspondente a um valor para o parâmetro *raioDBSCAN* e um valor para o parâmetro *qtdeObjetos*. De posse dessas configurações, o algoritmo DBSCAN adaptado deve ser executado. Por fim, as soluções obtidas devem ser avaliadas por meio da aplicação do Índice Silhueta. A quantidade de grupos da solução que resulta no maior valor do índice silhueta é indicada como a ideal.

As Tabelas 4, 5, 6 e 7 apresentam os melhores resultados em relação às 28 configurações do DBSCAN obtidos para, respectivamente, as instâncias do DS1, DS2 parte 1, DS2 parte 2 e DS3. Nessas tabelas, a coluna *k* indica o número de grupos correspondente ao maior valor da silhueta, a coluna *FX* corresponde ao maior valor de silhueta e a coluna *Tempo* possui colunas com o menor, o maior, a média dos tempos de execução dos algoritmos que encontraram o maior valor da silhueta (em segundos) e o desvio padrão (DESVP) dos tempos de execução.

É possível observar que o valor da silhueta foi positivo e maior ou igual a 0,5 para todas as instâncias do DS1, o que indica, por sua vez, que os grupos têm uma boa estrutura [Rousseeuw, 1987]. Além disso, apenas para as duas maiores instâncias em quantidade de objetos o tempo de execução foi superior a 1 segundo, sejam elas: gauss9 (900 objetos) e a yeast (1484 objetos).

Em relação aos resultados apresentados pelas Tabelas 5 e 6, referentes às instâncias do grupo DS2, observa-se que o valor da silhueta foi negativo apenas para 2 das 51 instâncias e ambas são consideradas instâncias "não *comportadas"*. Os tempos de processamento variaram entre 1 e 11 segundos, considerando as instâncias com um número de objetos entre 900 e 2000 objetos.

Tabela 4: Melhores Resultados Produzidos pelo MRDBSCAN Considerando o Conjunto DS1

			Tempo (segundos)			
Instância	**k**	**FX**	**Menor**	**Maior**	**Médio**	**Desvp**
DS1-200DATA	3	0,823	0,021	0,022	0,022	0,000
DS1-chart	2	1,000	0,360	0,361	0,360	0,000
DS1-gauss9	2	0,151	1,032	1,032	1,032	0,000
DS1-iris	2	0,687	0,013	0,019	0,016	0,003
DS1-maronna	2	0,562	0,021	0,022	0,022	0,000
DS1-ruspini	4	0,738	0,006	0,010	0,008	0,002
DS1-spherical_4d3c	4	0,689	0,106	0,107	0,106	0,000
DS1-vowel2	2	0,417	0,229	0,229	0,229	0,000
DS1-wine	2	0,545	0,019	0,025	0,022	0,003
DS1-yeast	3	0,550	4,600	4,634	4,617	0,024

Em relação aos resultados apresentados pela Tabela 6, referentes às instâncias do grupo DS3, observa-se que o valor da silhueta foi positivo para todas as instâncias. Os tempos de processamento foram da ordem de 1,5 segundos para a instância com 1000 objetos (1000p6c) e da ordem de 11 segundos para a instância maior, com 2000 objetos (2000p11c).

As Tabelas 4, 5, 6 e 7 apresentam os melhores resultados obtidos independente dos parâmetros submetidos ao MRDBSCAN. Com o objetivo de identificar os melhores parâmetros, ou seja, a melhor calibração realizada**,** a Tabela 8 (apresentada mais à frente) traz os resultados concernentes à aplicação das quatro regras, considerando cada um dos 7 valores de *k** para o *Distk*.

Tabela 5: Melhores Resultados Produzidos pelo MRDBSCAN Considerando o Conjunto DS2

			Tempo (segundos)			
Instância	**k**	**FX**	**Menor**	**Maior**	**Médio**	**Desvp**
DS2-1000p14c	15	0,808	1,445	1,468	1,457	0,011
DS2-1000p27c1	3	-0,293	1,448	1,454	1,451	0,004
DS2-1000p5c1	2	0,164	1,479	1,500	1,489	0,014
DS2-1000p6c	6	0,736	1,562	1,566	1,564	0,001
DS2-100p10c	8	0,692	0,009	0,009	0,009	0,000
DS2-100p2c1	2	0,743	0,009	0,009	0,009	0,000
DS2-100p3c	3	0,786	0,008	0,009	0,008	0,000
DS2-100p3c1	5	0,104	0,008	0,008	0,008	0,000
DS2-100p5c1	2	0,423	0,010	0,015	0,012	0,002

DS2-100p7c	7	0,834	0,008	0,009	0,008	0,000
DS2-100p7c1	2	-0,013	0,009	0,015	0,012	0,002
DS2-100p8c1	9	0,402	0,009	0,009	0,009	0,000
DS2-1100p6c1	5	0,369	1,945	1,968	1,956	0,016
DS2-1300p17c	18	0,806	3,044	3,075	3,060	0,014
DS2-1500p6c1	18	0,123	4,804	4,804	4,804	0,000
DS2-1800p22c	23	0,791	8,081	8,129	8,106	0,023
DS2-1900p24c	25	0,788	9,372	9,376	9,374	0,002
DS2-2000p11c	11	0,713	11,123	11,131	11,126	0,002
DS2-2000p26c	27	0,789	10,980	11,036	11,008	0,026
DS2-2000p9c1	2	0,164	10,990	11,045	11,018	0,027
DS2-200p12c1	3	0,403	0,029	0,035	0,032	0,004
DS2-200p2c1	6	0,625	0,026	0,026	0,026	0,000
DS2-200p3c1	2	0,648	0,023	0,024	0,023	0,000

Tabela 6: Melhores Resultados Produzidos pelo MRDBSCAN Considerando o Conjunto DS2

			Tempo (segundos)			
Instância	**k**	**FX**	**Menor**	**Maior**	**Médio**	**Desvp**
DS2-200p4c	4	0,773	0,022	0,030	0,026	0,003
DS2-200p4c1	3	0,623	0,024	0,030	0,027	0,004
DS2-200p7c1	3	0,392	0,026	0,032	0,029	0,004
DS2-200p8c1	13	0,423	0,026	0,026	0,026	0,000
DS2-300p10c1	3	0,512	0,055	0,055	0,055	0,000
DS2-300p13c1	3	0,404	0,032	0,038	0,035	0,004
DS2-300p2c1	4	0,621	0,071	0,071	0,071	0,000
DS2-300p3c	3	0,766	0,055	0,063	0,059	0,004
DS2-300p3c1	2	0,640	0,056	0,064	0,060	0,004
DS2-300p4c1	3	0,269	0,055	0,063	0,059	0,004
DS2-300p6c1	2	0,549	0,055	0,055	0,055	0,000
DS2-400p17c1	14	0,183	0,120	0,120	0,120	0,000
DS2-400p3c	3	0,799	0,114	0,124	0,119	0,005
DS2-400p4c1	2	0,379	0,117	0,117	0,117	0,000
DS2-500p19c1	20	0,136	0,211	0,211	0,211	0,000
DS2-500p3c	3	0,825	0,210	0,212	0,211	0,001
DS2-500p4c1	2	0,305	0,209	0,221	0,215	0,006
DS2-500p6c1	12	0,495	0,212	0,212	0,212	0,000
DS2-600p15c	15	0,781	0,335	0,336	0,336	0,000
DS2-600p3c1	2	0,687	0,354	0,354	0,354	0,000
DS2-700p15c1	2	0,123	0,532	0,532	0,532	0,000
DS2-700p4c	4	0,797	0,524	0,540	0,532	0,007
DS2-800p10c1	2	0,079	0,765	0,783	0,773	0,009
DS2-800p18c1	24	0,266	0,757	0,774	0,765	0,012
DS2-800p23c	23	0,787	0,791	0,792	0,792	0,000
DS2-800p4c1	2	0,509	0,780	0,797	0,788	0,012
DS2-900p12c	12	0,841	1,061	1,088	1,072	0,011
DS2-900p5c	5	0,716	1,092	1,094	1,093	0,001

A Tabela 8 traz uma síntese dos resultados obtidos considerando os *Gaps* (Equação 13) em relação aos melhores resultados obtidos para as regras Maior, Mediana, Pico10 e Pico20, respectivamente. Essa tabela apresenta, respectivamente, os *gaps* médio, mediano, o desvio padrão, o maior e o menor *gaps* em relação aos melhores valores obtidos por conjunto de instâncias.

Ainda na Tabela 8, em relação à regra Maior, os maiores *gaps* para as instâncias de DS1 e DS3 foram de apenas 0,1%. Observa-se, porém, que para o conjunto DS2 o maior Gap foi de 12,2%. Com base na coluna Média, os conjuntos DS1 e DS3 apresentaram gaps médios de 0%, e o conjunto DS2 um *gap* de apenas 1,1%.

A mesma tabela apresenta uma síntese de resultados com a aplicação da regra Mediana. Nesse caso, a média e os maiores gaps não foram satisfatórios, embora em cada conjunto de instâncias, ao menos para uma instância a melhor solução obtida foi alcançada. A média dos gaps foi de 7,7% e o maior *gap* foi de 56,3%.

Tabela 7: Melhores Resultados Produzidos pelo MRDBSCAN Considerando o Conjunto DS3

			Tempo (segundos)			
Instância	**k**	**FX**	**Menor**	**Maior**	**Médio**	**Desvp**
DS3-1000p6c	6	0,736	1,435	1,437	1,436	0,001
DS3-157p	4	0,666	0,016	0,016	0,016	0,000
DS3-181p	6	0,737	0,020	0,020	0,020	0,000
DS3-2000p11c	11	0,713	11,011	11,039	11,016	0,006
DS3-2face	2	0,667	0,023	0,024	0,023	0,000
DS3-300p4c	4	0,750	0,056	0,056	0,056	0,000
DS3-30p	2	0,382	0,004	0,004	0,004	0,000
DS3-350p5c	5	0,759	0,082	0,093	0,087	0,004
DS3-3dens	2	0,762	0,011	0,012	0,011	0,000
DS3-450p4c	4	0,766	0,154	0,159	0,155	0,001
DS3-500p3c	3	0,825	0,210	0,210	0,210	0,000
DS3-600p3c	3	0,751	0,349	0,371	0,357	0,007
DS3-900p5c	5	0,716	1,100	1,102	1,101	0,001
DS3-97p	3	0,711	0,008	0,012	0,010	0,003
DS3-convdensity	3	0,854	0,019	0,025	0,022	0,003
DS3-convexo	6	0,669	0,023	0,023	0,023	0,000
DS3-face	2	0,079	0,067	0,067	0,067	0,000
DS3-moreshapes	7	0,728	0,196	0,196	0,196	0,000
DS3-numbers	9	0,560	0,143	0,143	0,143	0,000
DS3-numbers2	10	0,600	0,251	0,268	0,258	0,006
DS3-outliers	2	0,787	0,014	0,019	0,017	0,004
DS3-outliers_ags	7	0,754	0,007	0,007	0,007	0,000

Tabela 8: Síntese dos Melhores Resultados Obtidos (*Gaps*) Mediante Aplicação das Quatro Regras

		Médio	**Mediano**	**DESVP**	**Maior**	**Menor**
Maior	**DS1**	0,00%	0,00%	0,00%	0,00%	0,00%
	DS2	1,10%	0,00%	2,73%	12,19%	0,00%
	DS3	0,01%	0,00%	0,02%	0,10%	0,00%
	TODAS	**0,58%**	**0,00%**	**2,03%**	**12,19%**	**0,00%**
Mediana	**DS1**	13,08%	5,79%	21,64%	56,30%	0,00%
	DS2	7,28%	4,44%	8,05%	20,26%	0,00%
	DS3	3,38%	0,00%	8,27%	20,26%	0,00%
	TODAS	**7,68%**	**1,39%**	**12,08%**	**56,30%**	**0,00%**
Pico10	**DS1**	0,74%	0,00%	1,57%	4,04%	0,00%
	DS2	6,17%	0,00%	13,81%	61,78%	0,00%
	DS3	0,18%	0,00%	0,40%	1,37%	0,00%
	TODAS	**3,92%**	**0,00%**	**11,15%**	**61,78%**	**0,00%**
Pico20	**DS1**	0,00%	0,00%	0,00%	0,00%	0,00%
	DS2	3,44%	0,00%	9,22%	40,04%	0,00%
	DS3	0,43%	0,00%	1,10%	4,57%	0,00%
	TODAS	**2,23%**	**0,00%**	**7,39%**	**40,04%**	**0,00%**

$$gap = \frac{(Silhueta_{best} + 1) - (Silhueta + 1)}{Silhueta_{best} + 1} \quad (13)$$

A Tabela 8 apresenta também uma síntese de resultados com a aplicação das regras Pico10 e Pico20. Embora novamente em cada conjunto de instâncias, ao menos para uma instância a melhor solução obtida foi alcançada, a média e os maiores gaps não foram satisfatórios. Para a regra Pico10 a média foi de 3,9% e o maior gap foi da ordem de 61,8% enquanto para a regra Pico20 a média foi de 2,2% e o maior gap foi da ordem de 40,0%.

Conforme os resultados apresentados na Tabela 8, a Regra Maior apresentou-se superior às demais regras. Porém, os resultados apresentados nessa tabela não discriminam quais valores de *k** foram utilizados para a obtenção das distâncias (parâmetro raio) e, consequentemente, à aplicação das quatro regras.

A Tabela 9 apresenta uma síntese dos resultados obtidos com a aplicação da análise *DistK* para todos os valores de *k** utilizados nos experimentos desse trabalho, sejam eles *k** = {3,4,5,10,15,20,50}. Nessa tabela a coluna Menor, que apresenta o menor gap em relação aos melhores resultados obtidos para cada instância, indica que para todos os valores de *k** utilizados foi possível alcançar o valor de silhueta da melhor solução obtida nos experimentos desse trabalho. A coluna mediana foi diferente de 0% somente para *k** = 20 no conjunto de dados DS3, em que o gap foi de apenas 0,1%.

Tabela 9: Resultados do Distk3 em Relação ao Melhor Resultado Obtido

K*	DS	MEDIA	MEDIANA	DESVP	MAIOR	MENOR
3	DS1	6,02%	0,00%	11,15%	34,95%	0,00%
	DS2	7,19%	0,00%	12,08%	40,04%	0,00%
	DS3	1,87%	0,00%	5,66%	25,87%	0,00%
	TODAS	**5,62%**	**0,00%**	**10,76%**	**40,04%**	**0,00%**
4	DS1	5,93%	0,00%	10,92%	34,20%	0,00%
	DS2	9,69%	0,00%	15,10%	61,66%	0,00%
	DS3	1,27%	0,00%	2,28%	7,07%	0,00%
	TODAS	**6,87%**	**0,00%**	**12,78%**	**61,66%**	**0,00%**
5	DS1	3,04%	0,00%	4,90%	10,37%	0,00%
	DS2	10,39%	0,00%	17,68%	61,78%	0,00%
	DS3	0,85%	0,00%	1,71%	7,07%	0,00%
	TODAS	**6,71%**	**0,00%**	**14,30%**	**61,78%**	**0,00%**
10	DS1	1,62%	0,00%	3,46%	10,36%	0,00%
	DS2	6,65%	0,00%	13,13%	51,92%	0,00%
	DS3	1,54%	0,00%	2,99%	9,85%	0,00%
	TODAS	**4,39%**	**0,00%**	**10,28%**	**51,92%**	**0,00%**
15	DS1	2,01%	0,00%	3,42%	10,12%	0,00%
	DS2	5,95%	0,00%	9,53%	36,84%	0,00%
	DS3	1,40%	0,00%	2,52%	8,88%	0,00%
	TODAS	**3,87%**	**0,00%**	**7,44%**	**36,84%**	**0,00%**
20	DS1	7,52%	0,00%	17,20%	49,64%	0,00%
	DS2	7,36%	0,00%	10,84%	37,74%	0,00%
	DS3	3,51%	0,10%	6,98%	22,23%	0,00%
	TODAS	**6,27%**	**0,00%**	**10,82%**	**49,64%**	**0,00%**
50	DS1	8,19%	0,00%	18,45%	56,30%	0,00%
	DS2	6,84%	0,00%	14,38%	55,57%	0,00%
	DS3	1,29%	0,00%	3,10%	11,02%	0,00%
	TODAS	**5,30%**	**0,00%**	**13,05%**	**56,30%**	**0,00%**

Ainda com base na Tabela 9, observa-se que os menores gaps médios foram observados para os valores *k**=15 e *k**=10, com respectivamente 3,87% e 4,39%. Além disso, com base na coluna Maior, que possui o maior gap em relação ao melhor resultado obtido nesse experimento, o menor valor foi obtido nos experimentos considerando *k** = 15, que também possui o menor desvio padrão. Como foi apresentado anteriormente, neste trabalho as soluções foram classificadas em válidas e inválidas conforme a quantidade de grupos. Uma solução válida possui a quantidade de grupos no intervalo [2, $n^{1/2}$].

A Figura 6 apresenta o gráfico de barras com os percentuais de soluções válidas considerando os conjuntos de instâncias DS1, DS2 e DS3, bem como as quatro regras. Com base nessa figura, podemos destacar as regras Pico10 e Pico20 com percentuais de soluções válidas próximas ou iguais a 100% em todos os conjuntos de instâncias.

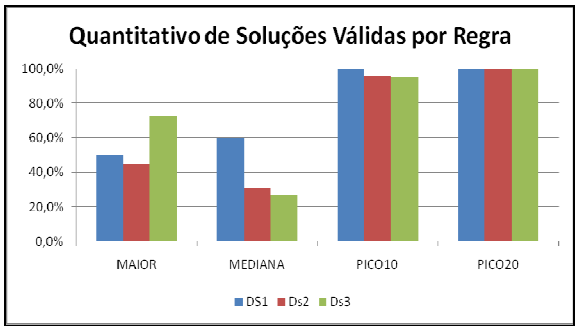

Figura 6: quantitativo de soluções válidas por regra

A Figura 7 apresenta o gráfico de barras com os percentuais de soluções válidas considerando os conjuntos de instâncias DS1, DS2 e DS3 bem como os valores de K^* para o experimento *DistK*. Nesse gráfico destacam-se os resultados obtidos para $K^* = \{3,4,5\}$, em que os percentuais de soluções válidas foram de 100% para os conjuntos de instâncias DS1 e DS3 e superior a 88% para o DS2.

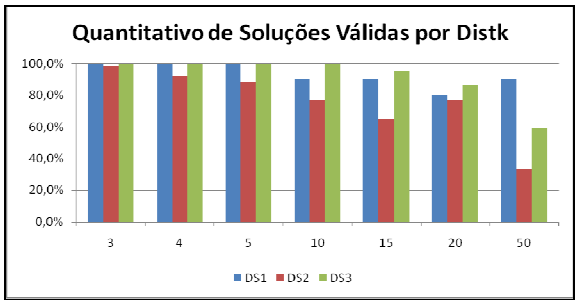

Figura 7: quantitativo de soluções válidas por valor de k em DistK

A Figura 8 apresenta os quantitativos de soluções válidas do conjunto de instâncias DS2, separando as instâncias consideradas "*comportadas*" das "*não comportadas*" (classificação utilizada no trabalho de [Cruz, 2010]). As soluções das instâncias "*comportadas*" foram superiores em quantitativos de soluções válidas tanto na média quanto considerando a Mediana. A Figura 9 apresenta a média e a mediana dos valores das melhores soluções obtidas considerando, também, a divisão entre as instâncias "*comportadas*" e "*não comportadas*". Nesse gráfico observa-se novamente a superioridade dos resultados relacionados às instâncias "*comportadas*". Enquanto a média e a mediana das soluções das instâncias não "*comportadas*" são inferiores a 0,4, os resultados das instâncias "*comportadas*" são superiores a 0,73.

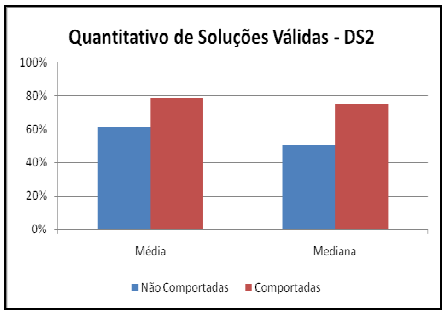

Figura 8: quantitativo de soluções válidas do DS2

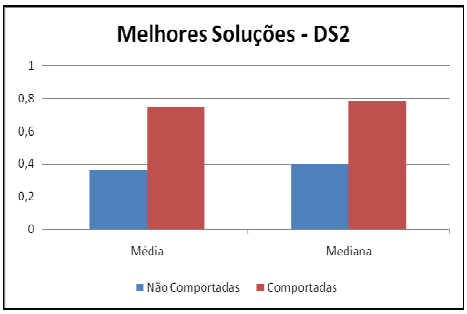

Figura 9: média e mediana das soluções obtidas do DS2

No segundo experimento apresentado no presente trabalho, além da apresentação e análises dos resultados obtidos pelo Método Proposto, foram efetuadas comparações com os algoritmos da literatura que consideram a mesma função de avaliação (Índice Silhueta).

Com base nos resultados apresentados por [Soares, 2004], para a comparação foram considerados os melhores resultados obtidos pelos algoritmos SAPCA e AEC-RC. O MRDBSCAN obteve resultados equivalentes ou superiores em 13 das 16 instâncias. Além disso, em relação ao número de grupos das três instâncias em que o método obteve resultados inferiores, o resultado para a instância *Iris* indica a mesma quantidade de grupos e nos resultados das instâncias *Face* e *Moreshapes* a diferença no número de grupos foi de apenas uma unidade.

A Tabela 10 apresenta resultados comparativos obtidos entre o MRDBSCAN e os algoritmos propostos por [Tseng and Yang, 2001] e por [Soares, 2004] em um subconjunto com 16 instâncias consideradas neste trabalho. Devido à heterogeneidade dos ambientes e das tecnologias em que os experimentos foram realizados, foram apresentados apenas os valores do índice silhueta e o número de grupos das melhores soluções obtidas para cada instância.

Tabela 10: Comparação com Resultados da Literatura

	TZENG E YANG	**SOARES**				**MRDBSCAN**	
INSTÂNCIA	**CLUSTERING**	**SAPCA**	**AEC-RC**	**BEST**	**k**	**Silhueta**	**k**
200Data	0,541	0,823	0,823	0,823	3	**0,823**	3
Iris	0,601	0,686	0,686	0,686	3	**0,687**	2
Ruspini	0,550	0,737	0,737	0,737	4	**0,738**	4
1000p6c	0,367	0,735	0,727	0,735	6	**0,736**	6
157p	0,657	0,667	0,667	**0,667**	**4**	0,666	**4**
2000p11c	0,287	0,658	0,611	0,658	11	**0,713**	11
2face	0,513	0,666	0,666	0,666	2	**0,667**	2
350p5c	0,568	0,758	0,758	0,758	5	**0,759**	5
3dens	0,742	0,762	0,762	0,762	2	**0,762**	2
97p	0,706	0,710	0,710	0,710	-	**0,711**	3
Convdensity	0,818	0,854	0,854	0,854	3	**0,854**	3
Convexo	0,618	0,667	0,667	0,667	3	**0,669**	6
Face	0,402	0,511	0,511	**0,511**	**3**	0,079	**2**
Moreshapes	0,436	0,731	0,725	**0,731**	**6**	0,728	**7**
Numbers	0,417	0,546	0,542	0,546	10	**0,560**	9
Numbers2	0,513	0,527	0,565	0,565	10	**0,600**	10

Em relação aos resultados do algoritmo CLUSTERING, proposto por [Tseng and Yang, 2001], o MRDBSCAN apresentou resultados superiores em 15 das 16 instâncias. Além disso, na instância Face, única em que as soluções possuíram silhuetas inferiores, o número de grupos do algoritmo CLUSTERING diferiu do número de grupos do MRDBSCAN em apenas uma unidade.

A Tabela 11 sumariza os melhores resultados produzidos pelos algoritmos propostos por [Cruz, 2010] e por [Wang, X. et al., 2007] para um subconjunto com 49 instâncias consideradas neste trabalho. Novamente, em decorrência da heterogeneidade dos ambientes e das tecnologias utilizadas nos experimentos realizados, foram apresentados apenas os valores do índice silhueta e o número de grupos das melhores soluções obtidas para cada instância. **Tabela 10:** Comparação com Resultados da Literatura.

A partir dos resultados reportados na Tabela 11, foi avaliada a diferença entre o número de grupos associado à melhor solução (métodos da literatura) produzida para as instâncias do conjunto DS2, em relação às soluções obtidas com o método MRDBSCAN. Com objetivo de tornar esta análise correta e justa, foram consideradas, separadamente, as instâncias "*comportadas"* (total de 17) e as "*não comportadas"* (total de 28). As Figuras 10 e 11 mostram estes resultados.

Tabela 11: Comparação com Resultados da Literatura

	Best		MRDBSCAN	
Nome	**FX**	**K**	**FX**	**K**
Ruspini	0,7370	4	**0,7377**	4
Iris	0,6862	3	**0,6867**	2
Maronna	0,5745	4	0,5622	2
200data	0,8231	3	**0,8232**	3
DS2-1000p14c	0,8306	14	0,8085	15
DS2-1000p27c1	0,5631	14	-0,2934	3
DS2-1000p5c1	0,6391	5	0,1640	2
DS2-1000p6c	0,7356	6	**0,7357**	6
DS2-100p10c	0,8336	10	0,6917	8
DS2-100p2c1	0,7427	2	**0,7427**	2
DS2-100p3c	0,7858	3	**0,7858**	3
DS2-100p3c1	0,5966	3	0,1044	5
DS2-100p5c1	0,7034	6	0,4235	2
DS2-100p7c	0,8338	7	**0,8339**	7
DS2-100p7c1	0,5511	7	-0,0127	2
DS2-1100p6c1	0,6847	6	0,3690	5
DS2-1300p17c	0,8229	17	0,8059	18
DS2-1500p6c1	0,6597	6	0,1233	18
DS2-1800p22c	0,8036	22	0,7913	23
DS2-2000p11c	0,7129	11	**0,7130**	11
DS2-2000p9c1	0,6230	9	0,1640	2
DS2-200p12c1	0,5753	13	0,4033	3
DS2-200p2c1	0,7642	2	0,6246	6
DS2-200p3c1	0,6797	3	0,6484	2
DS2-200p4c	0,7725	4	**0,7725**	4

	Best		MRDBSCAN	
Nome	**FX**	**K**	**FX**	**K**
DS2-200p4c1	0,7544	4	0,6227	3
DS2-200p7c1	0,5759	8	0,3922	3
DS2-300p13c1	0,5944	9	0,4039	3
DS2-300p2c1	0,7764	2	0,6208	4
DS2-300p3c	0,7663	3	**0,7664**	3
DS2-300p3c1	0,6768	3	0,6397	2
DS2-300p4c1	0,5924	4	0,2690	3
DS2-300p6c1	0,6636	8	0,5485	2
DS2-400p17c1	0,5524	15	0,1832	14
DS2-400p3c	0,7985	3	**0,7986**	3
DS2-400p4c1	0,6204	4	0,3790	2
DS2-500p3c	0,8249	3	**0,8249**	3
DS2-500p4c1	0,6597	3	0,3054	2
DS2-500p6c1	0,6684	6	0,4945	12
DS2-600p15c	0,7812	15	**0,7812**	15
DS2-600p3c1	0,7209	3	0,6868	2
DS2-700p15c1	0,6804	15	0,1227	2
DS2-700p4c	0,7969	4	**0,7970**	4
DS2-800p10c1	0,5071	8	0,0792	2
DS2-800p18c1	0,6941	19	0,2655	24
DS2-800p23c	0,7873	23	**0,7874**	23
DS2-800p4c1	0,7143	4	0,5088	2
DS2-900p12c	0,8408	12	**0,8409**	12
DS2-900p5c	0,7160	5	**0,7160**	5

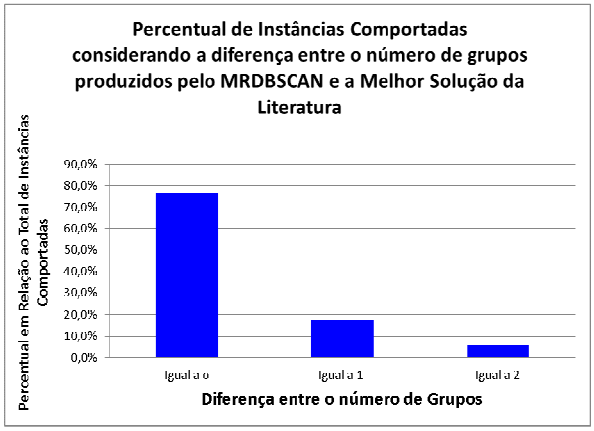

Figura 10: Instâncias Comportadas

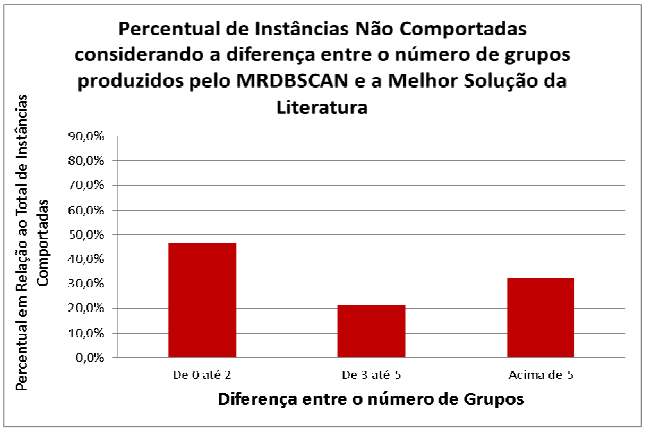

Figura 11: Instâncias Não Comportadas

Com base nos resultados apresentados na figura dez, observa-se, que na maioria dos casos (77% das instâncias comportadas), o MRDBSCAN produziu o número de grupos igual ao número de grupos associado à melhor solução da literatura. Além disso, em menos de 10% das instâncias esta diferença foi de duas unidades.

No que concerne às instâncias não comportadas, os resultados foram apenas razoáveis. Mais especificamente, em cerca da metade dos casos (47% das instâncias) o MRDSCAN produziu um número de grupos com até duas unidades de

diferença em relação à melhor solução da literatura. Além disso, em cerca de 20% das instâncias diferença entre o número de grupos variou de três até cinco. E finalmente, para 33% das instâncias, esta diferença foi superior a cinco.

Com base nos experimentos realizados, uma explicação plausível para os resultados obtidos para as instâncias não comportadas seria a ausência de procedimentos de busca local para o refinamento de soluções no método proposto. Uma alternativa apresentada na literatura por [Cruz, 2010], [Wang, X. et al., 2007], [Tseng and Yang, 2001] e [Soares, 2004] é a formação de grupos iniciais. Estes grupos seriam formados mediante a aplicação conjunta do MRDBSCAN e de algoritmos heurísticos com procedimentos de busca local. E neste caso, o objetivo seria produzir soluções finais de melhor qualidade no que concerne à quantidade de grupos e à maximização do índice silhueta.

5-CONCLUSÕES E TRABALHOS FUTUROS

Com o objetivo de identificar o número ideal de grupos em cada instância, o método proposto neste trabalho consiste na aplicação do algoritmo DBSCAN [Ester et. al., 1996] considerando diferentes parâmetros. Estes parâmetros foram obtidos utilizando uma técnica denominada *DistK*, baseada nas distâncias dos *k*-vizinhos mais próximos de cada objeto. A qualidade das soluções obtidas (agrupamentos) é indicada pelo coeficiente silhueta. E, quanto mais próximo de um estiver o valor desse coeficiente, mais interessante é a quantidade de grupos da solução.

A aplicação das quatro regras propostas neste trabalho, considerando um conjunto de valores de k^* para a análise do *DistK*, resultou na construção de soluções de boa qualidade. Com base nas Tabelas 8 e 9, por exemplo, observa-se que para qualquer valor de k^* e para qualquer regra, ao menos um experimento a melhor solução foi obtida.

A utilização da regra Maior obteve os melhores resultados em relação à qualidade de soluções, conforme apresenta a Tabela 8. O mesmo ocorreu para o *Distk*, k^*=15 (Tabela 9). Já em relação aos quantitativos de soluções consideradas válidas, as regras Pico10 e Pico15 e os valores de k^*={3, 4, 5} obtiveram maiores quantitativos.

Nos comparativos com os algoritmos da literatura, o MRDBSCAN foi superior ao CLUSTERING [Tseng and Yang, 2001] em 15 das 16 instâncias considerando o valor da silhueta e, na instância em que os resultados foram inferiores, a quantidade de grupos foi diferente em apenas uma unidade.

Os comparativos com os melhores algoritmos heurísticos propostos por [Soares, 2004], SAPCA e ARC-RC, o MRDBSCAN foi equivalente ou superior em 13 das 16 instâncias. Além disso, nas três instâncias em que o valor do índice silhueta foi inferior, os números de grupos foram equivalentes diferentes em apenas uma unidade das melhores soluções apresentadas na literatura.

Com base nos comparativos entre o MRDBSCAN e os algoritmos CLUES [Wang et. al., 2007] e os melhores algoritmos propostos em [Cruz, 2010] (AECBL1, GBLITRC1 e IBLITRC1), a análise realizada discriminou o conjunto de instâncias bem "comportadas" e "não comportadas", conforme denominação utilizadas por [Cruz, 2010], responsável pela criação de tais instâncias.

Em relação às instâncias "*comportadas*", a média da diferença entre a quantidade de grupos das melhores soluções da literatura e das soluções obtidas pelo MRDBSCAN foi de apenas 0,1 e, em 100% dos experimentos a diferença entre a quantidade de grupos foi de até 2 unidades. Para as instâncias "*não comportadas*" a média foi de 2,0 e em 60,71% dos experimentos a diferença foi de até 2 unidades.

A dificuldade do método em obter a quantidade de grupos em instâncias consideradas "*não comportadas"* decorre, principalmente, da ausência de uma busca local para refinar a solução, realizando migrações, união ou divisão de grupos e também da característica do índice silhueta. Além disso, conforme foi mencionado, esse índice é mais apropriado para agrupamentos volumétricos, com grupos gerados de acordo com distribuições Gaussianas multidimensionais hiperesféricas ou moderadamente alongadas.

Como propostas de trabalhos futuros temos:

- Utilizar a versão simplificada da silhueta proposta em [Hruschka et. al., 2004a] que reduz o custo computacional de $O(n^2)$ para O(n) e que mantém a qualidade próxima ao da silhueta tradicional [Vendramin et. al., 2009] [Vendramin et. al., 2010].
- Desenvolver heurísticas baseadas em metaheurísticas considerando o método proposto neste trabalho como uma heurística para a construção de soluções iniciais. Dessa forma, os procedimentos de busca local e as perturbações podem percorrer um novo espaço de busca para formação de novas soluções, que não seriam obtidas apenas com a utilização apenas do DBSCAN Tradicional.

AGRADECIMENTOS

Os autores agradecem ao apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ).

REFERÊNCIAS

[Alves et. al. 2006] Alves, V., R. Campello, & E. Hruschka (2006). *Towards a fast evolutionary algorithm for clustering. In IEEE Congress on Evolutionary Computation*, 2006, Vancouver, Canada, pp. 1776–1783.

[Bandyopadhyay and Maulik, 2001] Bandyopadhyay, S. & U. Maulik (2001). *Nonparametric genetic clustering: Comparison of validity indices. IEEE Transactions on Systems, Man and Cybernetics*, Part C : Applications and Reviews. 31 (1), 120–125.

[Bandyopadhyay and Maulik, 2002b] Bandyopadhyay, S. & U. Maulik (2002b). *Genetic clustering for automatic evolution of clusters and application to image classification. Pattern Recognition* 35, 1197–1208.

[Baum, 1986] Baum, E.B. *Iterated descent: A better algorithm for local search in combinatorial optimization problems. Technical report Caltech*, Pasadena, CA. Manuscript, 1986.

[Calinski and Harabasz, 1974] Calinski, R. B. & J. Harabasz (1974). *A dendrite method for cluster analysis. Communications in Statistics 3*.

[Campello et. al., 2009] Campello, R. J. G. B., E. R. Hruschka, & V. S. Alves (2009). *On the efficiency of evolutionary fuzzy clustering. Journal of Heuristics* 15 (1), 43–75.

[Cole, 1998] Cole, R. M. (1998). *Clustering with genetic algorithms*. MSc Dissertation, Department of Computer Science, University of Western Australia.

[Cowgill, 1999] Cowgill, M. C., R. J. Harvey, & L. T. Watson (1999). *A genetic algorithm approach to cluster analysis. Computational Mathematics and its Applications* 37, 99–108.

[Cruz, 2010] Cruz, M. D. O Problema de Clusterização Automática. Tese de Doutorado, UFRJ, Rio de Janeiro, 2010.

[Dias and Ochi, 2003] Dias, C.R.; & Ochi, L.S.. *Efficient Evolutionary Algorithms for the Clustering Problems in Directed Graphs*. Proc. of the IEEE Congress on Evolutionary Computation (IEEE-CEC), 983-988. Canberra, Austrália, 2003.

[Ester et al., 1995] Ester, M., Kriegel, H.-P., and Xu, X., *A Database Interface for Clustering in Large Spatial Databases*, In: Proceedings of the 1st International Conference on Knowledge Discovery in Databases and Data Mining (KDD-95), pp. 94- 99, Montreal, Canada, August, 1995.

[Ester et. al., 1996] Ester, M., H.-P. Kriegel, J. Sander, & X. Xu (1996). *A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining* (KDD'96), pp. 226–231.

[Fisher, 1936] Fisher, R. (1936). *The use of multiple measurements in taxonomic problems*. Annual Eugenics 7,pp. 179-188.

[Goldschimidt and Passos, 2005] Goldschimidt R.; Passos, E. Data Mining: um guia prático. Editora Campus, Rio de Janeiro: Elsevier, 2005.

[Han and Kamber, 2006] Han, J., e Kamber, M., *Cluster Analysis*. In: Morgan Kaufmann. Publishers (eds.), *Data Mining: Concepts and Techniques*, 2 ed., chapter 8, New York, USA, Academic Press, 2006.

[Handl and Knowles, 2007] Handl, J. & J. Knowles (2007). *An evolutionary approach to multiobjective clustering.IEEE Trans. on Evolutionary Computation* 34, 56–76.

[Hastie et. al., 2001] Hastie, t.;Tibshirani, R.; Friedman, J. (2001). *The Elements of Statistical Learning. Data Mining, Inference, and prediction*. Springer.

[Hruschka and Ebecken, 2001] Hruschka, E. R., Ebecken, N. F. F. *A Genetic algorithm for cluster analysis. IEEE Transactions on Evolutionary Computation* , 2001.

[Hruschka and Ebecken, 2003] Hruschka, E. R. & Ebecken, N. F. F. (2003). *A genetic algorithm for cluster analysis*. Intelligent Data Analysis 7 (1), 15–25.

[Hruschka et. al., 2004a] Hruschka, E. R., R. J. G. B. Campello, & L. N. de Castro (2004a). *Evolutionary algorithms for clustering gene-expression data*. In Proc. IEEE Int. *Conf. on Data Mining*, Brighton/England, pp. 403–406.

[Hruschka et. al., 2004b] Hruschka, E. R., R. J. G. B. Campello, & L. N. de Castro (2004b). *Improving the efficiency of a clustering genetic algorithm. In Advances in Artificial Intelligence* - IBERAMIA 2004: 9th Ibero-American Conference on AI, Puebla, Mexico, November 22-25. Proceedings, Volume 3315, pp. 861–870. Springer-Verlag GmbH, Lecture Notes in Computer Science.

[Hruschka et. al., 2006] Hruschka, E. R., R. J. G. B. Campello, & L. N. de Castro (2006). *Evolving clusters in gene-expression data. Information Sciences* 176 (13), 1898–1927.

[Jain and Dubes, 1988] Jain, A. & R. Dubes (1988). *Algorithms for Clustering Data*. Prentice Hall.

[Jr, 1968] Jr, H. S. (1968). *Cardinality of finite topologies*. Journal of Combinatorial Theory 5 (1), 82–86.

[Kumar et. al., 2009] Kumar, V. ; Steinbach, M. ; Tan, P. N. Introdução ao Data Mining - Mineração De Dados. Ciência Moderna, 2009.

[Larose, 2005] Larose, D. T. *Discovering Knowledge in Data, An Introduction to Data Mining*. John Wiley & Sons, 2005.

[Liu, 1968] Liu, G. (1968). Introduction to Combinatorial Mathematics. McGraw Hill.

[Ma et. al., 2006] Ma, P. C. H., K. C. C. Chan, X. Yao, & D. K. Y. Chiu (2006). *An evolutionary clustering algorithm for gene expression microarray data analysis*. IEEE Trans. Evolutionary Computations 10 (3), 296–314.

[Maronna and Jacovkis, 1974] Maronna, R.; Jacovkis, P. M. (1974). *Multivariate clustering procedures with variable metrics*. Biometrics30, pp. 499-505.

[Matloff 2011] Matloff, N. *The Art of R Programming: A Tour of Statistical Software Design*. No Starch. Press, 2011.

[Naldi and Carvalho, 2007] Naldi, M. C. & A. C. P. L. F. Carvalho (2007). *Clustering using genetic algorithm combining validation criteria*. In Proceedings of the 15th European Symposium on Artificial Neural Networks, ESANN 2007, Volume 1, pp. 139–144. Evere.

[Naldi, 2011] Naldi, C. N. *Técnicas de Combinação para Agrupamento Centralizado e Distribuído de Dados*. Tese de Doutorado, USP - São Carlos, 2011.

[Pakhira et. al., 2005] Pakhira, M., S. Bandyopadhyay, & U. Maulik (2005). *A study of some fuzzy cluster validity indices, genetic clustering and application to pixel classification*. Fuzzy Sets Systems 155 (2), 191–214.

[Pal and Bezdek, 1995] Pal, N. & J. Bezdek (1995). *On cluster validity for the fuzzy c-means model. IEEE Transactions of Fuzzy Systems* 3 (3), 370–379.

[Pan and Cheng, 2007] Pan, S. & K. Cheng (2007). *Evolution-based tabu search approach to automatic clustering. IEEE Transactions on Systems*, Man, and Cybernetics, Part C - Applications and Reviews 37 (5), 827–838.

[Pelleg and Moore, 2000] Pelleg, D. & A. Moore (2000). *X-means: extending k-means with efficient estimation of the number of clusters*. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 727–734.

[Rousseeuw, 1987] Rousseeuw, P. J. (1987). *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis*. Journal of Computational and Applied Mathematics 20, 53–65.

[Ruspini, 1970] Ruspini, E. H. (1970). *Numerical methods for fuzzy clustering*. Information Science.pp. pp. 319-350.

[Soares and Ochi, 2004] Soares, S. S. R. F., Ochi, L. S. *Um Algoritmo Evolutivo com Reconexão de Caminhos para o Problema de Clusterização Automática*. in *XII Latin Ibero American Congress on Operations Research*, 2004, Havana. Proc. of the XII CLAIO (em CD-ROM). ALIO, 2004. v.1, p. 7 -13.

[Soares, 2004] Soares, A. S. R. F. Metaheurísticas para o Problema de Clusterização Automática, Dissertação de Mestrado, UFF - Niterói, 2004.

[Steinbach et. al., 2000] Steinbach, M., G. Karypis, & V. Kumar (2000). *A comparison of document clustering techniques. Technical Report* 34, University of Minnesota.

[Tseng and Yang, 2001] Tseng, L. & . Yang, S.B. (2001). *A genetic approach to the automatic clustering problem*. Pattern Recognition 34, 415–424.

[Vendramin et. al., 2009] Vendramin, L., R. J. G. B. Campello, & E. R. Hruschka (2009). *On the comparison of relative clustering validity criteria.* In SIAM International Conference on Data Mining, Sparks/USA, pp. 733–744.

[Vendramin et. al., 2010] Vendramin, L., R. J. G. B. Campello, & E. R. Hruschka (2010). *Relative clustering validity criteria: A comparative overview*. Statistical Analysis and Data Mining 3 (4), 209–235.

[Wang et. al., 2007] Wang, X., Qiu, W., Zamar, R. H. (2007). *CLUES: A non-parametric clustering method based on local shrinking*. Computational Statistics & Data Analysis 52, pp. 286-298.

[Zalik, 2008] An Efficient *K'-Means Clustering Algorithm*, Pattern Recognition Letters 29, 2008.