UM NOVO ALGORITMO DE AGRUPAMENTO BASEADO EM COLÔNIA DE FORMIGAS

Giscard Fernandes Faria
Pós-Graduação em Computação Aplicada, Instituto Nacional de Pesquisas Espaciais (INPE),
Av. dos Astronautas, 1758, Jd. da Granja, São José dos Campos, SP, 12227-010
giscardff@gmail.com

Stephan Stephany
Laboratório de Computação e Matemática Aplicada, Instituto Nacional de Pesquisas Espaciais (INPE),
Av. dos Astronautas, 1758, Jd. da Granja, São José dos Campos, SP, 12227-010
stephan@lac.inpe.br

José Carlos Becceneri
Laboratório de Computação e Matemática Aplicada, Instituto Nacional de Pesquisas Espaciais (INPE),
Av. dos Astronautas, 1758, Jd. da Granja, São José dos Campos, SP, 12227-010
becce@lac.inpe.br

Resumo

Algoritmos de agrupamento baseados em Colônia de Formigas são inspirados na organização apresentada por esse tipo de colônia para gerenciar seu peculiar habitat. Nesta classe de algoritmos, objetos são espalhados aleatoriamente numa grade bidimensional, e cada formiga coleta e deposita estocasticamente um dado objeto, com base na similaridade de objetos vizinhos, determinada por uma métrica escolhida, normalizada por um parâmetro de similaridade. O algoritmo AntKSiMM, proposto em 2008, incorporou novos mecanismos, inclusive o uso de uma função núcleo para ponderar a similaridade. Entretanto, seu desempenho de agrupamento depende muito da escolha do valor desse parâmetro de similaridade, que é escolhido por tentativa-e-erro. O presente artigo apresenta uma nova versão do algoritmo denominada AntKSiMM+, que incorpora uma estratégia acoplada para inicialização e ajuste adaptativo desse parâmetro. São apresentados resultados numéricos que comparam o desempenho de agrupamento do algoritmo proposto ao daquele do algoritmo AntKSiMM e de outros algoritmos de agrupamento.

Palavras-Chaves: agrupamento; Colônia de Formigas; classificação de dados;

Abstract

Clustering algorithms based on Ant Colony are inspired by the organization of such colonies for the management of their peculiar habitat. In this class of algorithms, objects are randomly spread in a two-dimensional grid, and every ant stochasticaly collects and deposits a given object, based on its similarity to neighbor objects, determined by a chosen metric that is normalized by a similarity parameter. The AntKSiMM algorithm, proposed in 2008, included embodied new mechanisms, including the use of a kernel function in order to weight the similarity. However, its clustering performance depends heavily on the choice of the value of the similarity parameter, chose by trial and error. The current work presents a new version of this algorithm, named AntKSiMM+, that imbeds a coupled strategy for the initialization and adptive adjustment of such parameter. Numerical results are presented, comparing the clustering performance of the proposed algorithm to that of AntKSiMM and other clustering algorithms.

Keywords: clustering; Ant Colony; data classification;

1. INTRODUÇÃO

No começo dos anos 90, algoritmos baseados em Colônia de Formigas foram aplicados a problemas de otimização de rotas, tais como o Problema do Caixeiro Viajante. Hoje, tais algoritmos são aplicados a uma gama enorme de problemas, com ênfase em problemas de otimização combinatória. A Colônia de Formigas, como qualquer outra metaheurística, depende do ajuste conveniente de parâmetros intrínsecos para obter bom desempenho. O presente artigo apresenta o *AntKSiMM+*, uma nova versão do algoritmo *AntKSiMM* (Peterson et al., 2008). Este último é um algoritmo recente de agrupamento de dados baseado em Colônia de Formiga, que introduziu uma função núcleo *K (kernel function*) e um modelo de similaridade de memória *SiMM* (S*imilarity Memory Model*). Neste algoritmo, tais como nos anteriores, o desempenho do agrupamento depende muito do ajuste do parâmetro que pondera a similaridade entre objetos, sendo aqui proposta uma nova estratégia acoplada para a inicialização e ajuste adaptativo deste parâmetro.

Agrupamento de dados trata a heterogeneidade de um conjunto de dados (Lattin, 2011). Agrupar dados é procurar padrões não descobertos em um conjunto de dados, formando conjuntos chamados de grupos (*clusters*), com base em atributos específicos desses dados. Idealmente, objetos com alta similaridade entre si devem ficar no mesmo grupo, enquanto que objetos com baixa similaridade entre si, em grupos diferentes. Como citado em Everitt (1993), agrupamento de dados é utilizado em diversas áreas, tais como psiquiatria, medicina, serviço social, pesquisa de mercado, educação, etc. Quase todos os problemas de agrupamentos de grande porte, exigem uma abordagem heurística, dado que o número possível de agrupamentos cresce de modo exponencial, em função do número de objetos a serem classificados, e que o custo computacional também cresce com o número de atributos.

No decorrer dos anos, muitos algoritmos de agrupamento foram propostos. Algoritmos tradicionais, como o *k-means*, muitas vezes não apresentam resultados satisfatórios, porque a qualidade do agrupamento depende da escolha inicial adequada do número de grupos (*k*), o qual não é conhecido *a priori*. Outros algoritmos como o *g-means*, o χ*-means* e os algoritmos de agrupamento *Bayesianos*, tal como o *AutoClass*, identificam automaticamente o número de grupos, mas são mais custosos computacionalmente.

A avaliação de um algoritmo de agrupamento pode ser medida considerando a qualidade do agrupamento realizado, a qual é dada pela similaridade entre objetos de um mesmo grupo, e pelo seu tempo de processamento. Tipicamente, são utilizadas bases de dados cujos resultados de agrupamento foram publicados na literatura da área. Um bom algoritmo de agrupamento minimiza a variância intra-agrupamento enquanto maximiza a variância inter-agrupamento.

Um primeiro algoritmo foi proposto em Deneubourg et al. (1990) para dados compostos por atributos categóricos (literais). A similaridade do objeto a ser coletado ou depositado em relação à vizinhança era inferida a partir da memória individual da lista de objetos encontrados por cada formiga em seu trajeto aleatório. Lumer e Faieta (1994) criaram um novo algoritmo baseado em formigas introduzindo a função de densidade de similaridade para a vizinhança de cada formiga, sendo essa similaridade medida entre o objeto a ser coletado/depositado e os objetos da vizinhança. A seguir, foi proposto o algoritmo ATTA (Handl et al., 2004), que utilizava uma nova função de densidade de similaridade, empregava um memória de curto prazo dos objetos coletados por cada formiga e um raio de percepção variável. Diversos outros algoritmos foram propostos tais, como Chen et al 2004 e Dai et al 2009. Uma resenha pode ser encontrada em Jafar e Sivakumar 2010.

Tipicamente, um algoritmo de agrupamento inspirado em Colônia de Formigas agrupa objetos que foram espalhados aleatoriamente em uma grade bidimensional. Cada objeto ocupa uma célula desta grade e pode ser realocado para outra célula que esteja livre. As formigas

utilizam uma função estocástica para coletar e depositar os objetos. A probabilidade de coletar um objeto aumenta na proporção direta da dissimilaridade do objeto em relação a outros da sua vizinhança. A probabilidade de depositar um objeto aumenta na proporção de quão similar é o objeto que a formiga está carregando considerando a vizinhança por onde a formiga passa. Após muitas iterações de coletas e depósitos, objetos similares deverão estar agrupados.

O novo algoritmo *AntKSiMM+* foi aplicado para conjuntos de dados já publicados na literatura da área, tendo obtido um desempenho superior, especificamente em relação ao algoritmo precedente, o *AntKSiMM*. Basicamente, as formigas coletam e depositam objetos, retirando-os de regiões onde sejam dissimilares em relação aos objetos vizinhos e depositando-os em regiões onde haja objetos mais similares. O coeficiente de normalização da similaridade pondera a similaridade entre objetos, sendo que um valor muito alto facilita a formação de grupos (*clusters*) e um valor muito baixo dificulta a agregação de novos objetos a grupos existentes. Assim, inicialmente, um valor alto deste parâmetro seria ideal, porém deveria ser decrementado ao longo das iterações de forma a refinar os grupos existentes. A nova estratégia proposta consiste no uso de um esquema para inicialização deste parâmetro acoplado a um esquema para seu ajuste adaptativo ao longo das iterações. Além de sua inicialização, o esquema permite a definição de limites máximo e mínimo convenientes para este parâmetro com base numa métrica de similaridade escolhida (no caso, a distância euclidiana). Estes limites expressam, respectivamente, a maior e a menor distância interclasse e intraclasse. Essas distâncias são calculadas num espaço M-dimensional, sendo M é o número de atributos de cada objeto. Ao longo das iterações, o parâmetro pode ser reajustado dentro desses limites, com base na taxa de depósitos, que é forma de medir a dinâmica do agrupamento.

Este artigo está organizado da seguinte forma. A seção seguinte apresenta o algoritmo *AntKSiMM+*, enquanto que a seção 3 ilustra a estratégia acoplada para inicialização e ajuste adaptativo do parâmetro de similaridade em questão. A seção 4 apresenta os resultados de agrupamento obtidos para bases de dados já referenciadas na literatura para o *AntKSiMM+*, comparados aos resultados do *AntKSiMM* e outros algoritmos, seguida dos comentários finais, apresentados na seção 5.

2. O ALGORITMO ORIGINAL *ANTKSIMM*

O algoritmo *AntKSiMM* é um algoritmo recente de agrupamento baseado em formigas (Peterson; Kubler, 2008). Este algoritmo foi derivado do algoritmo ATTA, apresentando melhor desempenho de agrupamento que os demais anteriores. As duas melhorias principais inseridas em relação ao ATTA foram o uso da ***Função Kernel*** (K) e de um ***Modelo de Similaridade por Memória*** (SiMM).

- ***Função Kernel:*** Permite a ponderação da métrica de similaridade considerada, de forma a acentuar a similaridade (ou então, dissimilaridade) entre objetos em função da distância euclidiana entre estes na grade. Observando a Figura 1, nota-se que quanto maior a distância euclidiana entre objetos (eixo *X*) menor é o valor da função (eixo *Y*) utilizado para ponderar a similaridade entre os objetos, diminuindo-se assim a influência da similaridade no cálculo da densidade dentro da área de percepção da formiga.

- ***Modelo de Similaridade por Memória:*** Neste modelo é utilizada uma memória de curto prazo na qual são mantidos os últimos objetos transportados e as células onde

foram depositados, indicando para cada formiga a célula do objeto mais similar em relação ao que está transportando.
A formiga então se desloca para esta célula buscando uma célula livre nas vizinhanças para o depósito, condicionado à função de densidade. A memória é então atualizada com o novo objeto. Este modelo de memória também incorpora características semelhantes às aplicadas nos algoritmos baseados em *Simulated Annealing* (Kirkpatrick at al., 1983), nos quais a temperatura é utilizada para decidir se o uso da memória deve ou não ser efetuado. Esta temperatura é específica de cada objeto e vai sendo reduzida gradativamente conforme uma dada taxa de resfriamento.

Uma terceira melhoria implementada no *AntKSiMM* , em relação ao ATTA, mas de menor impacto, foi a adição de um probabilidade que define se a movimentação das formigas será aleatória ou condicionada ao gradiente da função de densidade de similaridade. A Figura 2 abaixo possui *"instantâneos"* da execução do algoritmo *AntKSiMM* para uma base de dados sintética, na qual objetos similares possuem a mesma cor. O algoritmo se inicia espalhando aleatoriamente todos os objetos da base de dados nas células da grade com dimensão *NxN*. A cada iteração, as formigas coletam e depositam os objetos com base na função de densidade de similaridade. O algoritmo finaliza ao atingir um número limite de iterações. Quando comparado a outros algoritmos de agrupamento baseado em Colônia de Formigas, o *AntKSiMM* apresentou convergência duas vezes mais rápida na formação dos grupos, mostrando potencial para problemas de agrupamento mais complexos (maior número de objetos e de atributos).

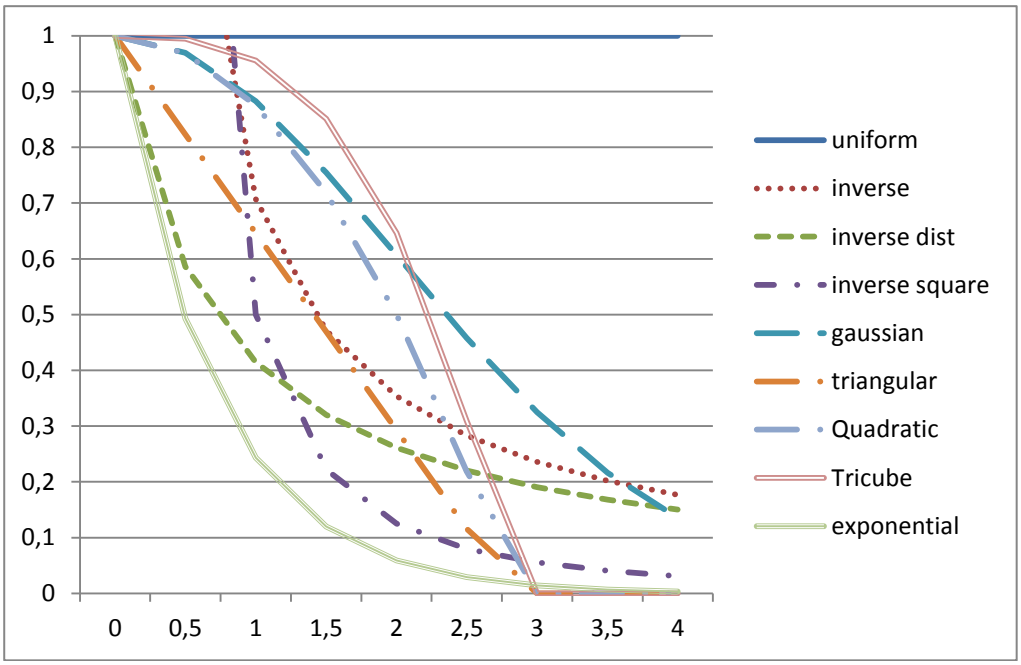

Figura 1 – Valor da função kernel em função da distância entre objetos. (Fonte: Peterson, Kluber, 2008).

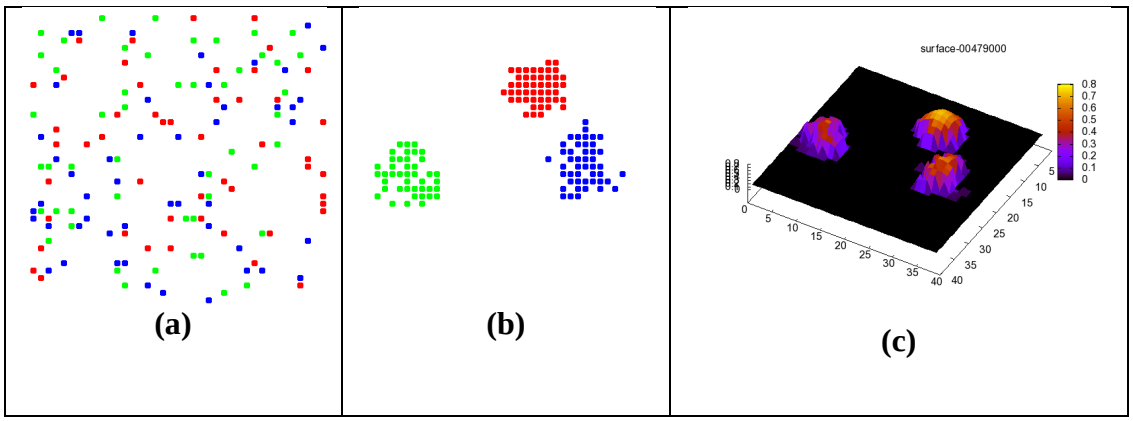

Figura 2 - Imagens referentes ao agrupamento de objetos de cores diferentes: (a) distribuição inicial, (b) distribuição final e (c) gráfico da densidade do agrupamento.

3. O ALGORITMO PROPOSTO *ANTKSIMM+*

Um ponto que foi deixado em aberto no algoritmo *AntKSiMM* e em outros algoritmos de agrupamento baseados em Colônia de Formigas agrupamento por formigas é a determinação do coeficiente de normalização da similaridade adequado para cada base de dados. O *AntKSiMM* determinou o valor mais apropriado empiricamente. O presente trabalho propõe uma nova estratégia acoplada de inicialização (3.1) e ajuste adaptativo deste parâmetro (3.2). Assim, o *AntKSiMM+* consiste numa versão aprimorada do *AntKSiMM* que incorpora as estratégias descritas a seguir.

3.1. ESQUEMA DE INICIALIZAÇÃO E ESCOLHA DO INTERVALO CONVENIENTE DO COEFICIENTE *α*

Assume-se um intervalo de valores possíveis para o parâmetro α definido pela maior distância interclasse (limite superior) e pela menor distância intraclasse (limite inferior), calculadas como a seguir, denotando-se por N o número de classes. O parâmetro α é então inicializado com o limite superior. Para o cálculo destas distâncias são definidas as seguintes matrizes:

- ***Matrizes de Distâncias Mínimas (M^{min})***: Matriz quadrada de dimensão N em que cada elemento M_{ij}^{min} é dado pela menor dentre todas as distâncias entre objetos da classe *i* e da classe *j*.
- ***Matrizes de Distâncias Máximas (M^{max})***: Matriz quadrada de dimensão N em que cada elemento M_{ij}^{max} é dado pela maior dentre todas as distâncias entre objetos da classe *i* e da classe *j*.

As matrizes acima permitem inferir os limites mínimo (α_{MIN}) e máximo (α_{MAX}), para definir um intervalo conveniente para o coeficiente de normalização da similaridade (α), em função da separabilidade entre classes, conforme se segue:

- α_{MIN} é igual à menor dentre as distâncias intraclasse (dadas pelos elementos da diagonal da matriz ***M^{min}***);
- α_{MAX} é igual à menor dentre as distâncias interclasse (dadas pelos elementos fora da diagonal da matriz ***M^{max}***);

Um segundo esquema aprimorado permite escolher um intervalo ainda mais estreito, com o uso das seguintes matrizes:

- ***Matrizes de Distâncias Médias (M^{med}):*** Matriz quadrada de dimensão N em que cada elemento M_{ij}^{med} é dado pela distância média entre objetos da classe *i* e da classe *j*.
- ***Matrizes de Desvio-Padrão das Distâncias (M^{dp}):*** Matriz quadrada de dimensão N em que cada elemento M_{ij}^{dp}, é dado pelo desvio-padrão das distâncias entre objetos da classe *i* e da classe *j*.

Essas outras matrizes permitem também inferir os limite mínimo (α_{MIN}) e máximo (α_{MAX}), para definir um intervalo conveniente para o coeficiente de normalização da similaridade (α), também em função da separabilidade entre classes, conforme se segue:

- α_{MIN} é igual à menor dentre as "metades inferiores dos intervalos de confiança" intraclasse, definidos por $(M_{ii}^{med} - M_{ii}^{dp})$ para a classe *i*;
- α_{MAX} é igual à maior dentre as "metades superiores dos intervalos de confiança" interclasse, definidos por $(M_{ij}^{med} + M_{ij}^{dp})$ entre as classes *i* e *j* ($i \neq j$);

3.2. ESQUEMA DE AJUSTE ADAPTATIVO DO COEFICIENTE α

O ajuste adaptativo do coeficiente de normalização da similaridade (α) é inspirado em Handl et al. (2005), que sugeria o ajuste com base na taxa de depósitos, conforme explicado abaixo, mas inicializando α aleatoriamente, sem definir um valor inicial ou um intervalo convenientes, como aqui propostos. Nesse esquema adaptativo, α é incrementado ou decrementado de uma quantidade ε ao longo das iterações em função do quociente (r_d) entre os depósitos não realizados e o número de iterações (soma-se ε quando $r_d \leq 0{,}1$ ou substrai-se ε quando $r_d > 0{,}1$)

No presente trabalho, inicializa-se r_d com um valor conveniente, muito alto, e inicializa-se α com seu limite superior α_{MAX}, limitando-se sua variação ao intervalo referido $[\alpha_{MIN}, \alpha_{MAX}]$. Isto permite aumentar (ou diminuir) o valor deste coeficiente, quando a taxa é baixa (ou alta), de forma a maximizar (ou minimizar) a probabilidade de novos depósitos. A implementação proposta adota um número fixo de iterações, sendo a primeira metade dessas iterações empregada para se obter um α ótimo, conforme o esquema adaptativo acima explicado, enquanto a segunda metade utiliza este valor ótimo para refinar o agrupamento.

4. RESULTADOS

O algoritmo de agrupamento proposto *AntKSiMM+* foi testado para bases de dados conhecidas, já empregadas em trabalhos publicados referentes a outros algoritmos de agrupamento. Em particular, no caso do conjunto de dado Wine, o próprio *AntKSiMM* (Peterson et al., 2008) é comparado com alguns outros algoritmos de agrupamentos: K-*means*, EM (*expectation-maximization*), *density based clustering*, e *farthest first traversal*. No caso do *AntKSiMM+* foram realizadas 10 execuções, cada uma com uma semente diferente para a geração de números aleatórios. A métrica adotada para avaliar a qualidade do agrupamento foi a função F-Measure, comumente empregada [Peterson et al 2008], sendo expressa por uma média e um desvio-padrão (idealmente, 1.0 e 0.0). As Tabelas 1 e 2 apresentam os resultados para os aos conjuntos de dados *Iris* e *Wine*, respectivamente. Os resultados da primeira tabela, mostram o bom desempenho do *AntKSiMM+* em relação ao *AntKSiMM* e aos demais algoritmos para o conjunto de dados *Iris*. Entretanto, pode-se observar na segunda tabela, referente ao conjunto de dados *Wine*, que outros algoritmos tiveram desempenho melhor;

contudo o AnKSiMM+ mostrou-se um algoritmo competitivo. Isso poderia ser atribuído ao fato do *AntKSiMM+* ainda estar sendo estudado.

	Iris					
	K-Means	**EM**	**Farthest First**	**Density Based**	**AntKSiMM**	**AntKSiMM+**
Média	0,89	0,90	0,86	0,90	0,77	0,92
D.P.	0,00	0,00	0,00	0,00	0,05	0,03

Tabela 1 - Média e Desvio-Padrão (D.P.) do *F-Measure* para 10 execuções dos diversos algoritmos de agrupamento para os conjuntos de dados *Iris.*

	Wine				
	K-Means	**EM**	**Farthest First**	**Desnity Based**	**AntKSiMM+**
Média	0,94	0,97	0,64	0,95	0,91
D.P.	0,00	0,00	0,00	0,00	0,06

Tabela 2 - Média e Desvio-Padrão (D.P.) do *F-Measure* para 10 execuções dos diversos algoritmos de agrupamento para os conjuntos de dados *Wine.*

A seguir, efetuaram-se testes de agrupamento com uma base de dados considerada mais difícil, referente à distinção entre pessoas saudáveis e aquelas com Mal de Parkinson's por meio da disfonia, ou seja, conjunto de alterações da voz. Este problema foi abordado em Little et al. (2009), que propôs uma métrica baseada em entropia aplicada ao algoritmo de classificação supervisionada SVM (*Support Vector Machine*). Houve também trabalhos anteriores relativos a esta base de dados, composta por um conjunto de 195 registros ou objetos, cada um com 22 medidas biométricas de alterações da voz. Destes objetos, 147 eram de pessoas com essa doença, enquanto as demais 48 eram saudáveis. Little et al. (2009) obteve seu melhor resultado para o conjunto reduzido dos 4 atributos mais signficativos, mas também obteve um resultado razoável com um único atributo. Neste trabalho, os testes referentes foram também para esses conjuntos reduzidos de atributos, e analogamente aos testes para os conjuntos de dados *Iris* e *Wine*, foram realizadas 10 execuções e a qualidade dos grupos foi avaliada pela função *F-measure*. As Tabelas 3 e 4 apresentam os resultados obtidos pelos algoritmos *AntKSiMM+, AntKSiMM, K-means, EM, density based clustering e farthest first traversal*, além do classificador SVM (Little et al., 2009) referentes às médias e os desvios-padrão do *F-measure* para a base de dados de disfonia, com 1 e com 4 atributos, respectivamente. Estes resultados permitem constatar que os desempenhos de agrupamento do *AntKSiMM+* e do *AntKSiMM* foram melhores que os demais algoritmos, exceto pelo SVM que teve uma versão específica, desenvolvida para este problema.

	Disfonia Little (1 atributo)					
	K-Means	**EM**	**Farthest First**	**Desnity Based**	**AntKSiMM+**	**SVM**
Média	0,61	0,64	0,65	0,62	0,79	0,81
D.P.	0,00	0,00	0,00	0,00	0,03	0,10

Tabela 3 - Média e Desvio-Padrão (D.P.) do *F-Measure* para 10 execuções dos diversos algoritmos de agrupamento (e o algoritmo de classificação SVM) para o conjuntos de dados de disfonia, com um único atributo.

	Disfonia Little (4 atributos)					
	K-Means	**EM**	**Farthest First**	**Desnity Based**	**AntKSiMM+**	**SVM**
Média	0,62	0,60	0,65	0,58	0,77	0,91
D.P.	0,00	0,00	0,00	0,00	0,04	0,04

Tabela 4 - Média e Desvio-Padrão (D.P.) do *F-Measure* para 10 execuções dos diversos algoritmos de agrupamento (e o algoritmo de classificação SVM) para o conjuntos de dados de disfonia, com 4 atributos.

A Tabela 5 ilustra o intervalo de variação adotado para o coeficiente de normalização da similaridade (α) no algoritmo *AntKSiMM+* para cada conjunto de dados, denotado pelos seus valores mínimos ($\alpha_{MIN)}$ e máximos ($\alpha_{MAX)}$, além do melhor valor (α_{BEST}). Esse intervalo foi determinado pelos limites máximo e mínimo encontrados pelo esquema proposto, baseado na distância mínima intraclasse e na distância máxima interclasse.

	α_{MIN}	α_{MAX}	α_{BEST}
Iris	0,83	1,29	0,40
Wine	0,97	1,35	0,9
Disfonia (1 atributo)	0,05	0,90	0,14
Disfonia (4 atributos)	0,59	0,80	0,35

Tabela 5 - Intervalo de variação adotado e valor ótimo para o coeficiente de normalização da similaridade (α) no algoritmo *AntKSiMM+*

Finalmente, os melhores resultados das execuções referidas nas tabelas anteriores são apresentados nas figuras seguintes: Figura 3 (*Iris*), Figura 4 (*Wine*), Figura 5 (disfonia com 1 atributo) e Figura 6 (disfonia com 4 atributos), novamente em função da média do *F-Measure*. Unicamente para o caso do algoritmo SVM, não havia disponibilidade do melhor resultado, sendo apresentadas nas últimas duas figuras o resultado médio obtido por esse algoritmo. Dessa forma, o algoritmo SVM deve ter apresentado alguns resultados melhores e outros piores.

Neste trabalho, o algoritmo *AntKSiMM+* foi implementado na linguagem Java, sendo efetuadas 600.000 iterações em cada execução, demandando um tempo médio de 3,0 minutos com um processador Intel de 2,6 GHz. Destas iterações, a primeira metade permitiu a obtenção de um valor adequado para α, enquanto a segunda metade permitiu refinamento do agrupamento, mantendo este valor com baixa variação. Empregou-se sempre a função kernel Gaussiana e a distância Euclidiana como métrica de similaridade. O algoritmo proposto avalia a qualidade do agrupamento periodicamente por meio da média do *F-Measure*. Nos testes

realizados esta avaliação é feita a cada 100 iterações, para não incorrer em um custo computacional alto. Quanto aos demais algoritmos, foram tomadas as versões implementadas no ambiente Weka (http://www.cs.waikato.ac.nz/ml/weka/), exceto pelo SVM. Neste caso, simplesmente tomaram-se os valores apresentados em Little et al. (2009).

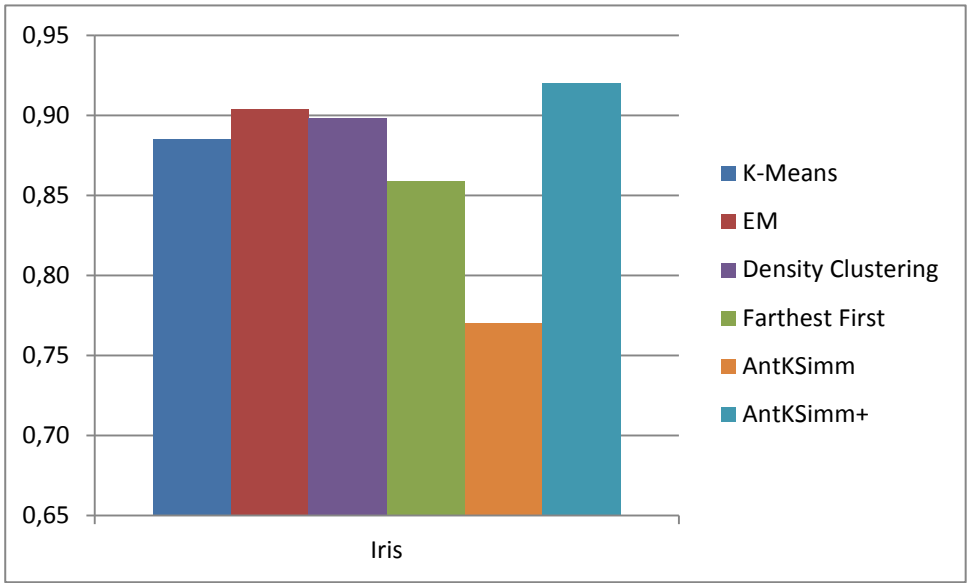

Figura 3 - Melhores valores obtidos com os diversos algoritmos de agrupamentos para o conjunto de dados *Iris*.

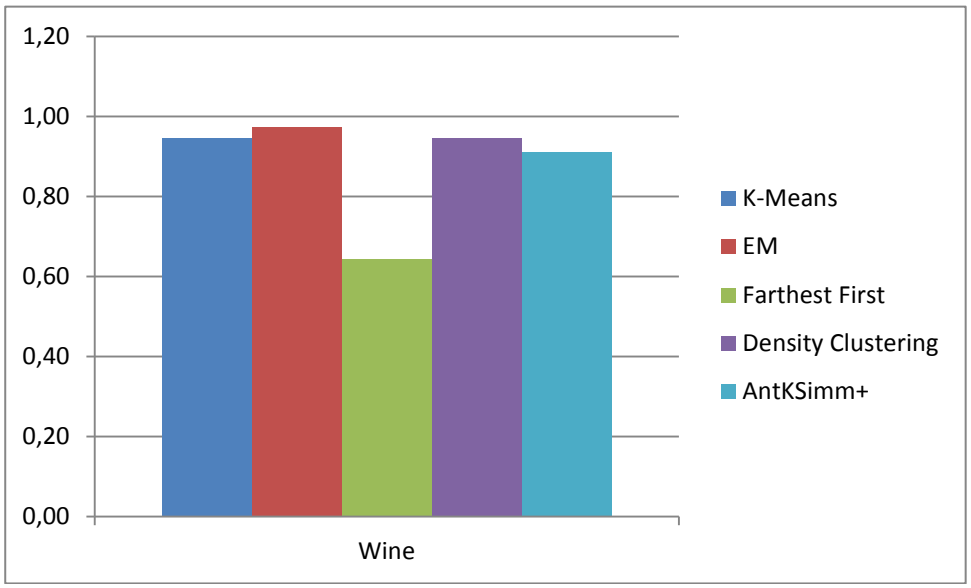

Figura 4 - Melhores valores obtidos com os diversos algoritmos de agrupamentos para o conjunto de dados *Wine*.

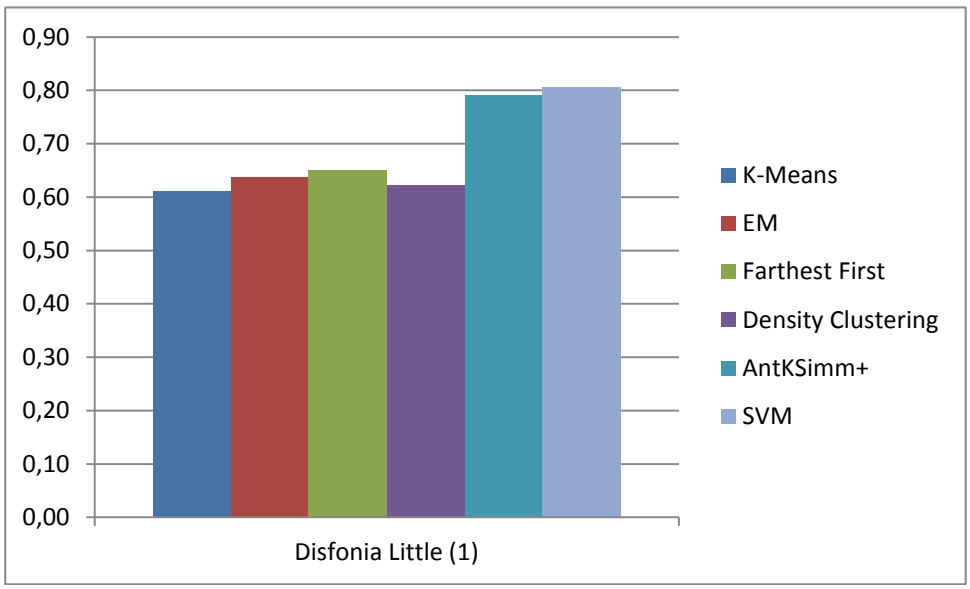

Figura 5 - Melhores valores obtidos com os diversos algoritmos de agrupamentos para o conjunto de dados de disfonia com 1 atributo.

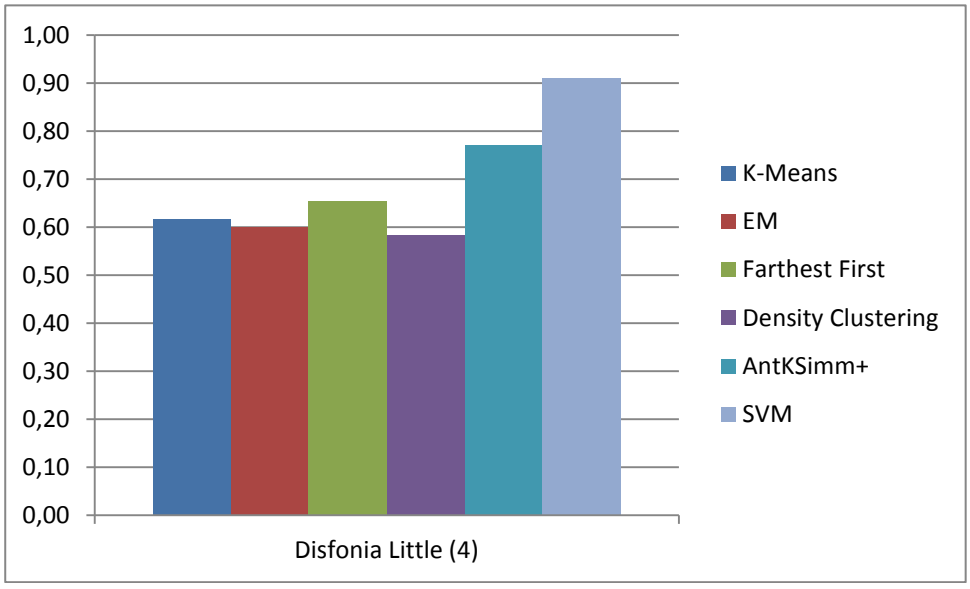

Figura 6 - Melhores valores obtidos com os diversos algoritmos de agrupamentos para o conjunto de dados de disfonia com 4 atributos.

5. COMENTÁRIOS FINAIS

O presente trabalho apresentou o algoritmo AntKSiMM+, uma nova versão do algoritmo de agrupamento baseado em Colônia de Formigas AntKSiMM. O algoritmo proposto implementa uma nova estratégia acoplada de inicialização e ajuste adaptativo do parâmetro que pondera a similaridade entre objetos, o coeficiente de normalização da similaridade (α), o qual tinha um valor fixo atribuído empiricamente no algoritmo original AntKSiMM. No algoritmo novo adotou-se um número de iterações fixo, sendo que a metade destas iterações visava à obtenção de um valor próximo ao ótimo deste coeficiente, enquanto que a outra metade das iterações fixava este valor de forma a refinar o agrupamento. Assim sendo, pode-se dizer que a segunda metade destas iterações equivaleria à execução do algoritmo original AntKSiMM. Entretanto, neste algoritmo, a escolha empírica deste

coeficiente implicaria num custo computacional indeterminado, uma vez que se baseia na tentativa e erro. Os resultados demonstram a importância da escolha adequada desse parâmetro no desempenho do agrupamento e o potencial do emprego de algoritmos baseados em formigas em agrupamento de dados. Os casos de teste são referentes a bases de dados rotuladas (utilizadas propositalmente para facilitar a mensuração de qualidade do cluster), embora o AntKSiMM+ possa ser aplicado à bases de dados não-rotuladas mantendo as mesmas vantagens.

Um aperfeiçoamento viável do AntKSiMM+ seria implementar estratégias de avaliação do desempenho de agrupamento que sejam mais eficientes em termos de tempo de processamento. Outro trabalho futuro será aperfeiçoar o esquema adaptativo utilizado para o α, de forma a evitar iterações desnecessárias, como é o caso de se adotar um número fixo de iterações. E, analogamente, ao se determinar o valor ótimo de α, adotar um critério de parada conveniente. Outro trabalho futuro será a avaliação do desempenho computacional do AntKSiMM+ e sua comparação com algoritmos tradicionais de agrupamentos para conjuntos de dados com dimensionalidade maior, para os quais o AntKSiMM teria melhor desempenho computacional (Handl et al, 2004).

6. REFERÊNCIAS BIBLIOGRÁFICAS

[1] **Deneubourg, J.L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C. e Chretien, L.** In: *Proceedings of the First International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, 356-365, MIT Press, Cambridge, MA, 1991.

[2] **Everitt, B. S., Landau, S., Leese, M.** *Cluster Analysis*, Wiley-Blackweel, 2011.

[3] **Handl, J., Knowles, J. e Dorigo, M.** (2005), Ant Based Clustering and Topographic Mapping, Artificial Life, 12(1), 35-62.

[4] **Kirkpatrick, S., Gellat C. D. e Vecchi M. P.** (1983), Optimization by Simulated Annealing. Science, NewSeries, 220 (4598), 671–680.

[5] **Lattin, J., Carrol, D. e Green, P. E.** *Análise de Dados Multivariados*, Cengage Learning, São Paulo, 2011.

[6] **Little, M. A., McSharry, P. E., Hunter, E. J., Spielman, J. e Ramig, L. O.** (2009), Suitability of Dysphonia Measurements for Telemonitoring of Parkinson's Disease, *IEEE Transactions on Biomedical Engineering*, 56(4), 1015-1022.

[7] **Lumer, E. e Faieta, B.** Diversity and adaption in population of clustering ants. In: *Proceedings of the Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, 501-508, MIT Press, Cambridge, MA, 1994.

[8] ***Peterson, G. L., C.B. Mayer e Kubler, T. L.*** *(2008), Ant clustering with locally weighted ant perception and diversified memory,* Swarm Intelligence, *2(1): 43-68.*

[9] **Chen, L.; Xu , Xiao-Hua; Chen, Yi-Xin**. An adaptive ant colony clustering algorithm. *Proceedings of 2004 International Conference on Machine Learning and Cybernetics 2004*, vol.3, no., pp. 1387- 1392 vol.3, 26-29 Aug. 2004.

[10] **Dai , W.; Liu, S.; e Liang, S.** An Improved Ant Colony Optimization Cluster Algorithm Based on Swarm Intelligence JOURNAL OF SOFTWARE, VOL. 4, NO. 4, JUNE 2009.

[11] **Jafar, O. A. M. e Sivakumar R.**. Ant-based Clustering Algorithms: A Brief Survey. International Journal of Computer Theory and Engineering, Vol. 2, No. 5, October, 2010 pages 1793-8201.