Agrupamento de Dados Baseado em Dinâmica de Troca de Energia

Roberto Alves Gueleri[1], **Zhao Liang**[1]

[1]Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
São Carlos – SP – Brasil

{gueleri,zhao}@icmc.usp.br

Abstract. *We present a data clustering method based on dynamics of energy exchange. Initially, each object is assigned an energy state. So the process consists of the progressive exchange of energy among objects. When two of them reach close enough energy states, they are grouped together. The distance between each pair of objects defines its degree of interaction. Thus, it tends to gradually cluster the regions of higher density of objects. The natural and dynamic property of the proposed method shows interesting results. Simulations with artificial and real data show its potentialities and limitations.*

Resumo. *Apresentamos um método de agrupamento de dados baseado em dinâmica de troca de energia. No início, a cada objeto é atribuído um estado de energia. O processo de agrupamento consiste então na gradativa troca de energia entre os objetos. Quando dois deles atingem estados de energia suficientemente semelhantes, eles são agrupados. A distância que separa cada par de objetos define seu grau de interação. Assim, esse método tende a agrupar gradualmente as regiões com maior densidade de objetos. O caráter natural e dinâmico desse método mostra resultados interessantes. Simulações com dados artificiais e reais procuram mostrar as potencialidades e limitações desse método.*

1. Introdução

A tarefa do agrupamento de dados (*clustering*) pode ser entendida como: dado um conjunto não-classificado de objetos, encontrar a melhor partição desse conjunto, de modo que objetos de um mesmo grupo sejam mais semelhantes que objetos de grupos distintos [Mitchell 1997, Jain et al. 1999, Alpaydin 2004, Xu and Wunsch 2005, Zhao et al. 2005, Bishop 2006, Hastie et al. 2009]. Os métodos de agrupamento de dados costumam ser classificados em *hierárquicos* e *particionais* [Jain et al. 1999, Xu and Wunsch 2005]. Um método hierárquico agrupa os objetos segundo uma sequência de partições, desde aquela que atribui um *cluster* distinto para cada objeto até aquela que confina todos os objetos num único *cluster*. Já um método particional divide diretamente o conjunto de objetos num número pré-especificado de *clusters*. Os métodos de agrupamento hierárquico são, por sua vez, classificados em *aglomerativos* e *divisivos* [Jain et al. 1999, Xu and Wunsch 2005]. A essencial

diferença entre eles é a ordem da sequência de partições. Enquanto os métodos aglomerativos iniciam com uma partição que atribui um *cluster* distinto para cada objeto, os métodos divisivos seguem o caminho contrário, iniciando com uma partição que confina todos os objetos num único *cluster*.

Tipicamente, o agrupamento de dados é um problema combinatoriamente difícil [Jain et al. 1999, Xu and Wunsch 2005, Abraham et al. 2007], sendo comumente conhecido como um problema de otimização. A inspiração na natureza, além da sua intrínseca elegância, tem sido uma prática valiosa para lidar com problemas computacionais complexos, combinatoriamente difíceis [Castro 2007]. Blatt et al. (1997) desenvolveram um método de agrupamento de dados baseado nas propriedades físicas de um ferromagneto heterogêneo. Um *spin* é atribuído a cada objeto, de modo que cada um desses *spins* será influenciado pelos *spins* dos objetos vizinhos. Ao final, a correlação entre os *spins* dirá quais objetos pertencem a quais *clusters*. Abraham et al. (2007), em seu trabalho intitulado *Swarm Intelligence Algorithms for Data Clustering*, mostram aplicações bem-sucedidas de técnicas bioinspiradas em problemas de agrupamento de dados. Mais especificamente, o trabalho trata da otimização por colônia de formigas e da otimização por nuvem de partículas. Ambas as abordagens fundamentam-se na inteligência que emerge da interação de muitos indivíduos relativamente simples. Oliveira (2008) também aplica com sucesso a técnica de otimização por nuvem de partículas a problemas de agrupamento de dados. Zhao et al. (2005) propuseram um método de agrupamento dinâmico e auto-organizável baseado na atração de objetos similares. A evolução do processo de agrupamento consiste em gradativamente concentrar os grupos de objetos em pontos, que são considerados os centros de cada *cluster*.

O presente trabalho propõe um método de agrupamento de dados baseado em dinâmica de troca de energia. Trata-se de um método hierárquico e aglomerativo. No início, a cada objeto é atribuído um estado de energia. O processo de agrupamento consiste então na gradativa troca de energia entre os objetos. Quando dois deles atingem estados de energia suficientemente semelhantes, eles são agrupados. Não pretende-se, contudo, modelar fielmente as leis físicas que regem os processos naturais de troca de energia. Pelo contrário, essas "leis" são justamente parâmetros a serem modificados. Assim, este trabalho objetiva investigar o comportamento dinâmico do método proposto, observando, para diferentes parâmetros e diferentes conjuntos de dados, como os *clusters* são formados durante o processo de agrupamento. A Seção 2 descreve esse método. A Seção 3 mostra algumas simulações realizadas.

2. O método de agrupamento de dados baseado em dinâmica de troca de energia

Duas são as entidades envolvidas no método aqui descrito: *objetos* e *clusters*. Os objetos são os elementos a serem agrupados, enquanto os *clusters* são justamente os grupos de objetos. Cada *cluster* é caracterizado por um *estado de energia*, significando que cada um de seus objetos compartilham esse estado de energia. O processo de agrupamento é então caracterizado pela progressiva troca de energia entre os *clusters*, agrupando-os conforme atingem estados de energia suficientemente semelhantes.

Seja *natt* a quantidade de atributos que caracterizam cada objeto. Assim, cada objeto $\mathbf{o}_i \in \mathbb{R}^{natt}$ é definido como

$$\mathbf{o}_i = \Big(o_{i,1}, o_{i,2}, o_{i,3}, \ldots, o_{i,natt}\Big) \tag{1}$$

A evolução do processo de agrupamento dá-se através de sucessivas iterações, denotadas pela variável $t \in \mathbb{N}$, tal que $t = 0$ refere-se ao estado inicial do sistema. A cada iteração, os *clusters* (representando seus objetos) trocam certa quantidade de energia entre si. Quando dois ou mais atingem estados de energia suficientemente semelhantes, eles combinam-se, originando um *cluster* maior. A essa combinação dá-se o nome de *fusão*. Assim, cada iteração t é dividida em dois sub-estados: *pré-fusão* e *pós-fusão*.

Seja *nobj* a quantidade de objetos envolvidos no problema. Denota-se por $\mathbf{c}'_i(t)$ e $\mathbf{c}_i(t) \in \mathbb{R}^{nobj}$ os *clusters* na pré-fusão e na pós-fusão de t, respectivamente.

$$\mathbf{c}'_i(t) = \Big(e'_{i,1}(t), e'_{i,2}(t), e'_{i,3}(t), \ldots, e'_{i,nobj}(t)\Big) \tag{2}$$

$$\mathbf{c}_i(t) = \Big(e_{i,1}(t), e_{i,2}(t), e_{i,3}(t), \ldots, e_{i,nobj}(t)\Big) \tag{3}$$

Aqui, as componentes $e'_{i,j}(t)$ e $e_{i,j}(t)$ representam o estado de energia do *cluster*. Denotam-se também por $C'(t)$ e $C(t)$ os conjuntos de todos os *clusters* existentes na pré-fusão e na pós-fusão de t, respectivamente.

Todo *cluster* está associado a seus objetos. Assim, definem-se *cluster*$'$, *cluster* e *obj* como sendo as relações entre os *clusters* e seus respectivos objetos. Tais relações são definidas reciprocamente como

$$cluster'(\mathbf{o}_i, t) = \mathbf{c}'_j(t) \iff \mathbf{o}_i \in obj(\mathbf{c}'_j(t)) \tag{4}$$

$$cluster(\mathbf{o}_i, t) = \mathbf{c}_j(t) \iff \mathbf{o}_i \in obj(\mathbf{c}_j(t)) \tag{5}$$

Ou seja, *cluster*$'$ e *cluster* resultam no *cluster* ao qual pertence um certo objeto. Já *obj* representa a relação inversa, resultando nos objetos que compõem um certo *cluster*.

Inicialmente, na pré-fusão de $t = 0$, atribui-se um *cluster* distinto a cada objeto. Têm-se pois *nobj clusters*. A cada um desses *clusters* é então atribuído um estado inicial de energia:

$$e'_{i,j}(0) = \begin{cases} 0 & \text{para } i \neq j \\ 1 & \text{para } i = j \end{cases} \tag{6}$$

A disposição simétrica assim concebida garante que nenhum dos objetos seja eventualmente privilegiado. Disso decorre a necessidade de tantas componentes de energia quantos são os objetos envolvidos no problema.

Descreve-se agora o processo de fusão. Essencialmente, há dois tipos de fusão: um para $t = 0$ e outro para $t > 0$. Após a disposição inicial dos *clusters* em $t = 0$, é conveniente já agrupar objetos situados muito perto um do outro. Denomina-se essa proximidade por *omin*, tal que objetos cuja distância euclidiana seja menor ou igual

a *omin* são agrupados. Essa é a fusão para $t = 0$. Já para $t > 0$, procura-se, a cada iteração, por *clusters* cujos estados de energia sejam suficientemente semelhantes. Denomina-se essa semelhança por *emin*, de modo que *clusters* cujos estados de energia tenham distância euclidiana menor ou igual a *emin* são agrupados.

A fim de realizar a fusão, é necessário criar uma partição do conjunto $C'(t)$, tal que os *clusters* contidos numa mesma parte sejam todos agrupados entre si. A partição $P(t)$ procurada é definida como

$$P(t) = \Big\{ M_m(t) \subseteq C'(t) \Big\}, \quad m = 1 \ldots \text{tamanho da partição} \tag{7}$$

onde $M_m(t)$ representa cada parte da partição. Todo *cluster* estará em uma e somente uma parte. Havendo dois *clusters* tal que a distância entre seus respectivos objetos seja menor ou igual a *omin* ($t = 0$), ou a distância entre seus estados de energia seja menor ou igual a *emin* ($t > 0$), esses *clusters* estarão na mesma parte e serão agrupados.

Define-se $merge(M_m(t))$ como sendo uma operação capaz de tomar uma parte $M_m(t)$ de alguma partição e transformá-la num novo *cluster*, resultado da fusão dos *clusters* de $M_m(t)$:

$$merge(M_m(t)) = \begin{cases} \dfrac{\displaystyle\sum_{\mathbf{c}'_i(t) \in M_m(t)} \mathbf{c}'_i(t)}{|M_m(t)|} & \text{para } t = 0 \\ \dfrac{\displaystyle\sum_{\mathbf{c}'_i(t) \in M_m(t)} \Big[\mathbf{c}'_i(t) \cdot |obj(\mathbf{c}'_i(t))| \Big]}{\displaystyle\sum_{\mathbf{c}'_i(t) \in M_m(t)} |obj(\mathbf{c}'_i(t))|} & \text{para } t > 0 \end{cases} \tag{8}$$

onde $|M_m(t)|$ e $|obj(\mathbf{c}'_i(t))|$ referem-se, respectivamente, ao tamanho da parte $M_m(t)$ e ao tamanho do conjunto $obj(\mathbf{c}'_i(t))$ de objetos do *cluster* $\mathbf{c}'_i(t)$. Nota-se que, para $t = 0$, o *cluster* resultante possui estado de energia igual à média dos estados de energia dos *clusters* que entraram em fusão. Para $t > 0$, o estado de energia resultante é igual à média ponderada pelo tamanho dos *clusters*, ou seja, quanto mais objetos um *cluster* possui, mais influente ele é. Na verdade, a primeira expressão, para $t = 0$, é um caso especial da segunda, visto que em $t = 0$ todo *cluster* tem tamanho igual a 1. Então, o novo conjunto $C(t)$ resultante da fusão dos *clusters* em $C'(t)$, agora organizados na partição $P(t)$, é

$$C(t) = \Big\{ merge(M_m(t)) \Big\}, \quad m = 1 \ldots \text{tamanho da partição} \tag{9}$$

Passa-se agora ao processo de troca de energia. Define-se $neighborhood(\mathbf{o}_i, t)$ como sendo o conjunto dos *nneighbors* $\mathbf{o}_j$ mais próximos de $\mathbf{o}_i$, que não façam parte do mesmo *cluster* de $\mathbf{o}_i$ na pós-fusão da iteração t. O parâmetro *nneighbors* simplesmente limita a quantidade de vizinhos considerados na troca de energia. $neighborhood(\mathbf{o}_i, t)$ pode resultar num conjunto menor que *nneighbors*, caso a quantidade de objetos não seja mais suficiente. Observar que não faz sentido a interação

entre objetos contidos num mesmo *cluster*, uma vez que já compartilham o estado de energia.

Então, define-se uma função chamada *resultant* que informa a variação do estado de energia que cada *cluster* "deseja" ter entre os instantes t e $t+1$. Mais precisamente, trata-se de um par de funções, uma para os *clusters*:

$$resultant(\mathbf{c}_i(t)) = \Big(resultant(e_{i,1}(t)), \ldots, resultant(e_{i,nobj}(t))\Big) \tag{10}$$

e outra para as componentes dos *clusters*:

$$resultant(e_{i,k}(t)) = \sum_{\substack{\mathbf{o}_m \in obj(\mathbf{c}_i(t)) \\ \mathbf{o}_n \in neighborhood(\mathbf{o}_m, t)}} \left[\frac{e_{j,k}(t) - e_{i,k}(t)}{|obj(\mathbf{c}_i(t))|} \cdot \right.$$
$$\left. \cdot\ exchange(\mathbf{c}_i(t), \mathbf{c}_j(t), k)\ \cdot\ falloff(\|\mathbf{o}_m - \mathbf{o}_n\|) \right] \tag{11}$$

onde $\mathbf{c}_j(t) = cluster(\mathbf{o}_n, t)$, $e_{j,k}(t)$ refere-se às componentes de $\mathbf{c}_j(t)$ e $\|\mathbf{o}_m - \mathbf{o}_n\|$ é a distância entre $\mathbf{o}_m$ e $\mathbf{o}_n$. Observam-se, na definição de *resultant*, duas outras funções: *exchange* e *falloff* (*fall off*).

$$exchange : \mathbb{R}^{nobj} \times \mathbb{R}^{nobj} \times \mathbb{N} \longmapsto \mathbb{R} \tag{12}$$
$$falloff : \mathbb{R} \longmapsto \mathbb{R} \tag{13}$$

Essas outras duas funções fornecem fatores para o cálculo de *resultant* e constituem parâmetros do presente método de agrupamento. Informalmente, *exchange* diz qual a taxa de troca de energia entre dois *clusters* em função da diferença entre seus estados de energia. Já *falloff* diz o quanto diminui a influência entre dois objetos em função da distância que os separa.

Foi dito que *resultant* refere-se à variação do estado de energia "desejada" por cada *cluster*. No entanto, quando os estados de energia de dois (ou mais) *clusters* caminham um em direção ao outro, convergindo ao equilíbrio, deve-se evitar que quando cheguem razoavelmente perto, uma próxima iteração lance-os com demasiada violência a lados opostos, tornando-os mais distantes do que estavam. A Figura 1(a) ilustra essa situação indesejada. Introduz-se então a função $normalization(t)$, cujo objetivo é gerar um fator capaz de normalizar a variação de energia de todos os *clusters*, de modo que a maior das variações seja igual à constante *emin*:

$$normalization(t) = \frac{emin}{max\Big(\Big\{\|resultant(\mathbf{c}_i(t))\|\Big\}\Big)}, \quad i = 1 \ldots |C(t)| \tag{14}$$

onde $\|resultant(\mathbf{c}_i(t))\|$ e $|C(t)|$ denotam, respectivamente, a magnitude do vetor $resultant(\mathbf{c}_i(t))$ e o tamanho do conjunto $C(t)$. A função max resulta, consequentemente, na maior das variações do estado de energia. A Figura 1(b) ilustra a variação dos estados de energia sendo limitada por *emin*. Essa figura ilustra o caso onde a variação é a máxima possível, ou seja, igual a *emin*. Vale lembrar que quando a

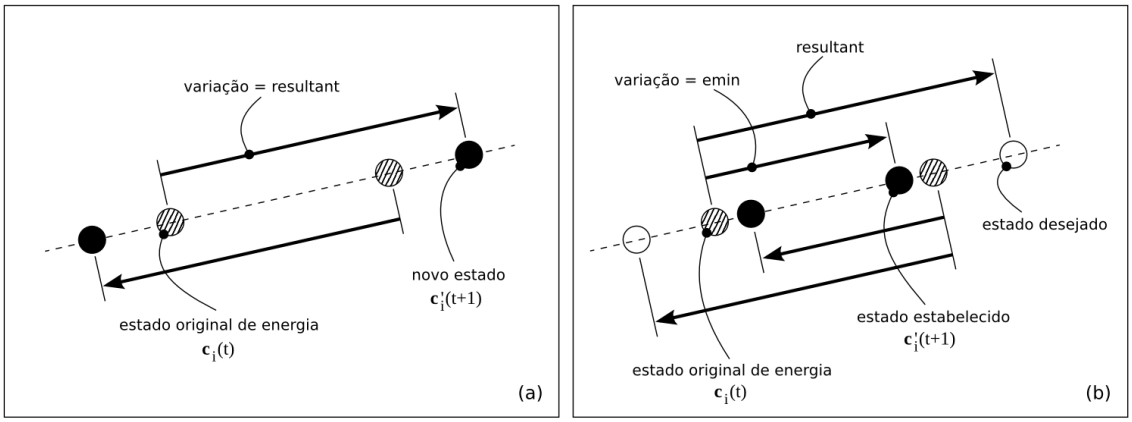

Figura 1. Dois estados de energia num movimento simétrico de aproximação. Em (a), a situação indesejada, onde a distância entre eles após a variação se torna maior que a distância antes da variação. Em (b), a variação sendo limitada por $emin$**.**

Tabela 1. Os parâmetros do método baseado em dinâmica de troca de energia.

Parâmetro	Descrição
$omin$	Objetos cuja distância seja menor ou igual a $omin$ são agrupados já em $t = 0$.
$emin$	*Clusters* cuja diferença entre seus estados de energia seja menor ou igual a $emin$ são agrupados. Valores grandes de $emin$ aceleram a convergência, mas reduzem a precisão do processo de agrupamento.
$nneighbors$	Número de vizinhos considerados. Cada objeto, a cada iteração, só troca energia com os $nneighbors$ vizinhos mais próximos que não pertencem ao mesmo *cluster*.
$exchange(\mathbf{c}_i(t), \mathbf{c}_j(t), k)$	Define a taxa de troca de energia entre dois *clusters* em função da diferença entre seus estados de energia.
$falloff(\|\mathbf{o}_m - \mathbf{o}_n\|)$	Define o quanto diminui a influência entre dois objetos em função da distância que os separa.

distância entre os estados de energia de dois *clusters* torna-se menor ou igual a $emin$, esses *clusters* são definitivamente agrupados. Importante notar também que, caso o maior dos valores para *resultant* seja menor que a constante $emin$, *normalization* aumenta todas as variações de energia, evitando que a convergência do processo de agrupamento seja muito lenta. Assim, valores grandes de $emin$ aumentam a velocidade de convergência, mas reduzem a precisão do processo de agrupamento.

Finalmente, a variação efetiva do estado de energia de cada *cluster*:

$$e'_{i,j}(t+1) = e_{i,j}(t) + resultant(e_{i,j}(t)) \cdot normalization(t) \tag{15}$$

Importante notar que todo o processo de agrupamento consiste na intercalação de dois sub-processos: fusão dos *clusters* e troca de energia entre os *clusters*. A fusão leva de um conjunto $C'(t)$ de *clusters* a um novo conjunto $C(t)$, eventualmente menor que $C'(t)$. A troca de energia leva o estado de energia de cada *cluster*

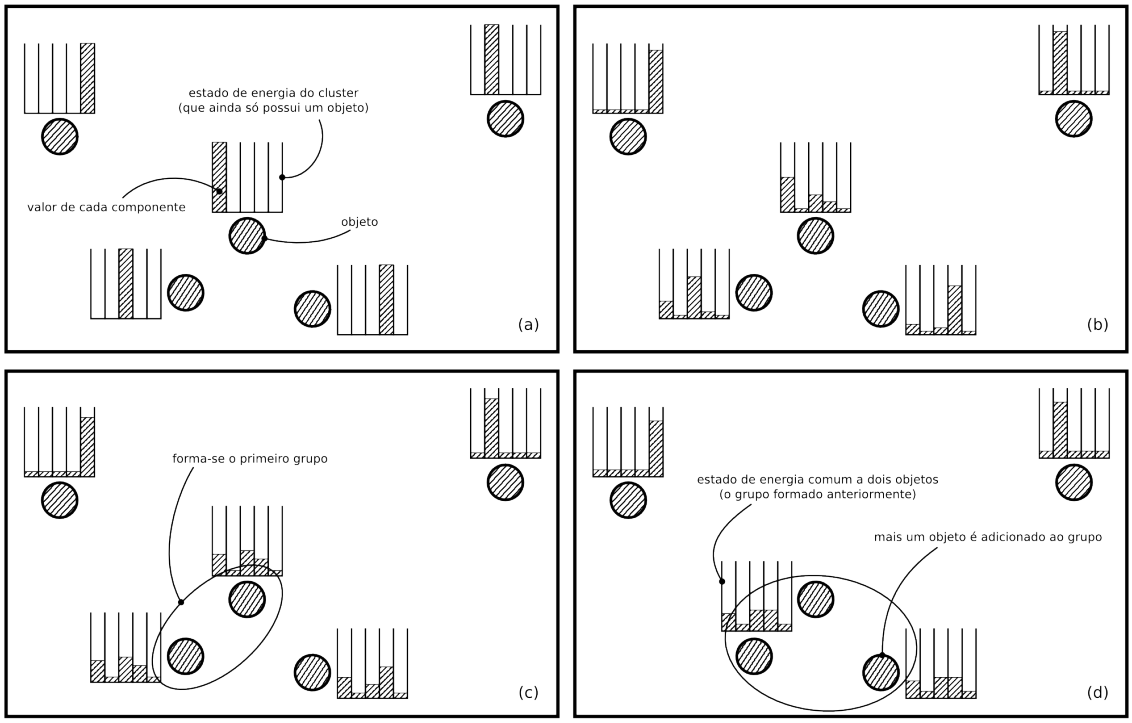

Figura 2. Ilustração do processo de agrupamento. (a): o estado inicial dos objetos. (b): algum estado após a evolução do processo. (c): dois dos objetos atingem estados de energia suficientemente semelhantes e são agrupados. (d): um terceiro objeto é adicionado ao grupo.

$\mathbf{c}_i(t)$ a um novo valor em $\mathbf{c}'_i(t+1)$. A Tabela 1 sumariza os parâmetros do método descrito.

A fim de ilustrar o processo de agrupamento, a Figura 2 mostra um conjunto contendo cinco objetos. A cada objeto é atribuído um estado inicial de energia, conforme a Equação (6). Então esses objetos começam a interagir entre si e os grupos vão se formando. Quanto maior a distância separando dois objetos, menor o grau de interação entre eles. Assim, os objetos mais próximos são os primeiros a atingirem estados de energia semelhantes, portanto são os primeiros a serem agrupados.

3. Simulações realizadas

Simulações foram realizadas com o método de agrupamento de dados baseado em dinâmica de troca de energia proposto neste trabalho. As Seções 3.1 e 3.2 trazem, respectivamente, os resultados de simulações sobre conjuntos de dados artificiais e reais.

Todas as simulações empregaram os seguintes parâmetros: $omin = 0,1$, $emin = 0,02$, $nneighbors = 20$ e $exchange(\mathbf{c}_i(t), \mathbf{c}_j(t), k) = 1$. Esse valor para $exchange$, quando aplicado na Equação (11), faz a intensidade da troca de energia entre dois objetos ser diretamente proporcional à diferença entre seus valores de energia. O parâmetro variado foi a função *falloff*, que para todos os conjuntos testados, assumiu: $falloff = 1/\|\mathbf{o}_m - \mathbf{o}_n\|$ e $falloff = 1/\|\mathbf{o}_m - \mathbf{o}_n\|^5$, lembrando que $\|\mathbf{o}_m - \mathbf{o}_n\|$ representa a distância entre os objetos.

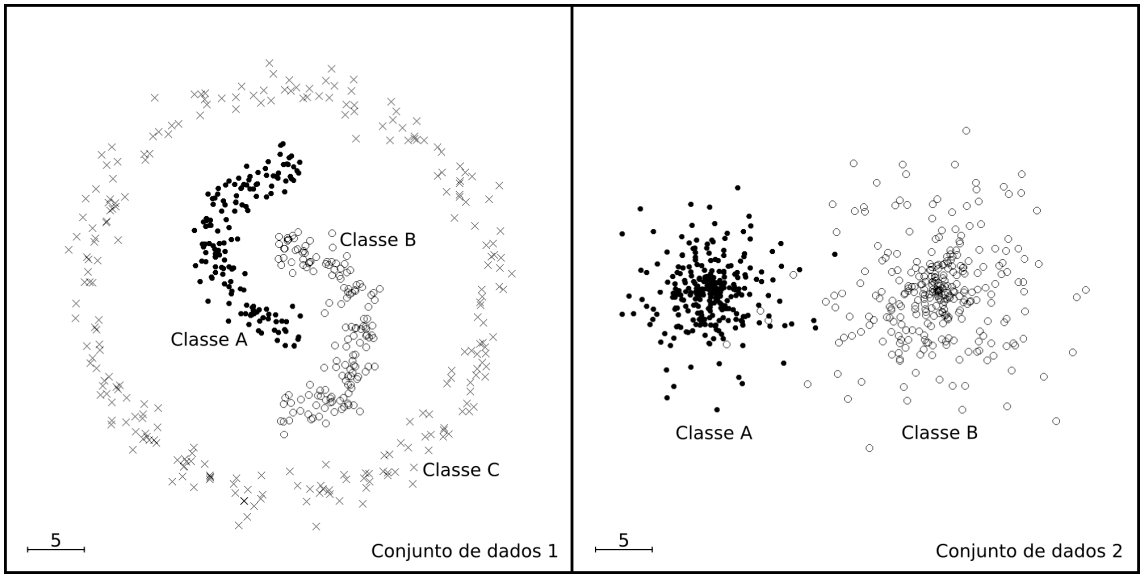

Figura 3. Dois conjuntos artificiais de dados utilizados nas simulações.

Os conjuntos de dados utilizados neste trabalho são formados por classes de objetos. Essas classes existem *a priori*. Contudo, o método de agrupamento não as "conhece", evidentemente. Seu objetivo é justamente "descobri-las". Assim, a fim de mensurar o processo de agrupamento, rastreou-se o crescimento de um *cluster* representativo de cada classe do conjunto de dados. Seja uma classe A. Então procurou-se pelo *cluster* que, ao longo de todo o processo de agrupamento, detivesse a maior quantidade de objetos da classe A. Mediu-se então a relação entre a quantidade de objetos da classe A presentes no *cluster* e o tamanho do próprio *cluster*.

3.1. Simulações com dados artificiais

Geraram-se dois conjuntos artificiais de dados, como mostra a Figura 3. Cada conjunto é identificado simplesmente como *Conjunto de dados 1* e *Conjunto de dados 2*.

O Conjunto de dados 1 contém três classes. As Classes A e B são formadas, cada uma, por 150 objetos distribuídos aleatoriamente por uma semicircunferência de raio 7,5. A Classe C, ao redor das outras duas, é formada por 250 objetos distribuídos aleatoriamente por uma circunferência de raio 18. Em todas as classes, o valor angular de cada objeto teve probabilidade uniforme pela circunferência (ou semicircunferência) e o valor radial seguiu uma distribuição de probabilidade normal (gaussiana) com valor esperado igual ao raio e desvio padrão igual a 1.

O Conjunto de dados 2 contém duas classes. Cada uma delas é formada por 300 objetos distribuídos aleatoriamente ao redor de um ponto, seu centro. Seguiu-se uma distribuição de probabilidade normal bivariada, com desvio padrão igual a 4 para a Classe A e 6 para a Classe B, em ambas as dimensões. Os centros de cada classe distam 20 unidades um do outro.

A Figura 4 mostra os resultados sobre o Conjunto de dados 1. Dentre os dois experimentos, observa-se desempenho consideravelmente melhor com *falloff* =

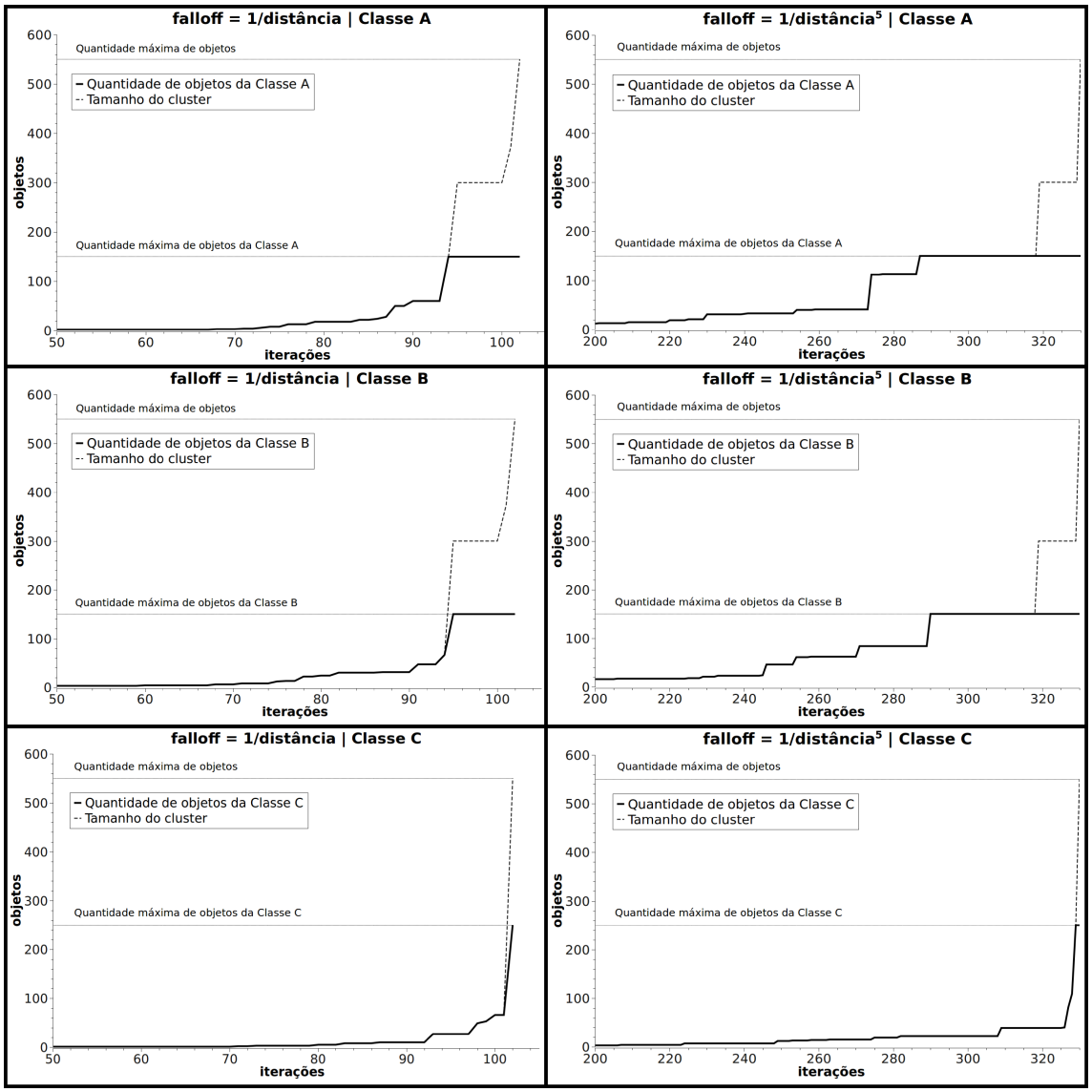

Figura 4. Resultados das simulações sobre o Conjunto de dados 1.

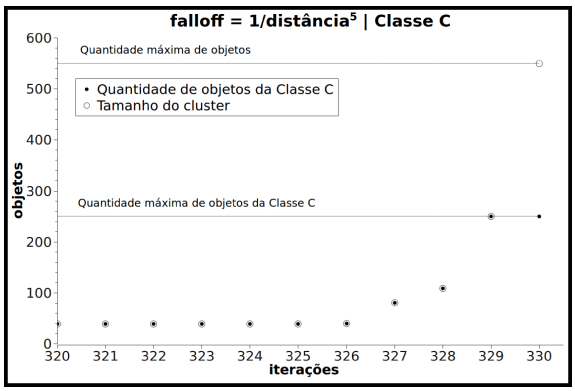

Figura 5. Evolução do cluster representativo da Classe C, nas últimas iterações da simulação sobre o Conjunto de dados 1, empregando $falloff = 1/\|\mathbf{o}_m - \mathbf{o}_n\|^5$.

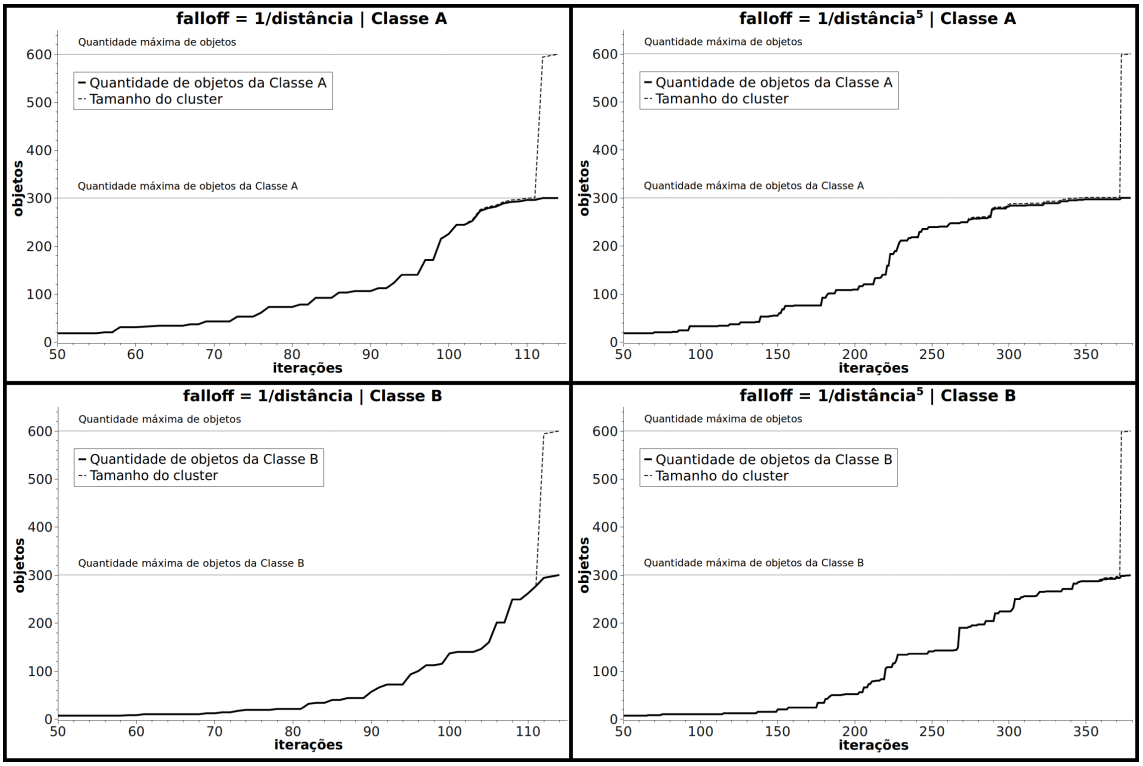

Figura 6. Resultados das simulações sobre o Conjunto de dados 2.

$1/\|\mathbf{o}_m - \mathbf{o}_n\|^5$. Nele, as Classes A e B são perfeitamente agrupadas, e permanecem assim por várias iterações, antes que comecem a se misturar. A Classe C atinge também seu estado ideal, permanecendo assim durante uma iteração somente, a iteração $t = 329$, como detalha a Figura 5.

A Figura 6 mostra os resultados sobre o Conjunto de dados 2. Observa-se desempenho ligeiramente melhor com $falloff = 1/\|\mathbf{o}_m - \mathbf{o}_n\|^5$. As duas classes mantêm-se praticamente puras por bastante tempo, mas quando chegam perto de tornarem-se completas, misturam-se bruscamente.

3.2. Simulações com dados reais

Mostram-se aqui os resultados de simulações sobre um conjunto de dados denominado *Iris Data Set*, obtido do *UCI Machine Learning Repository* [UCI]. Esse conjunto é comumente empregado para avaliação de métodos de classificação ou agrupamento de dados. Ele é composto por três classes, cada uma contendo 50 objetos. Cada classe representa uma espécie da planta *Iris*: *Iris setosa*, *Iris versicolor* e *Iris virginica*. Cada objeto é formado por quatro atributos: tamanho e largura da sépala e da pétala, em centímetros.

A Figura 7 mostra os resultados obtidos. Observa-se bom desempenho para a classe *Iris setosa*, principalmente com $falloff = 1/\|\mathbf{o}_m - \mathbf{o}_n\|^5$. Ela chega perto de ser completamente agrupada antes que comece a se misturar com as duas outras. A partir daí, a mistura é brusca. Para as outras duas classes, no entanto, houve desempenho insatisfatório. De fato, as classes *Iris versicolor* e *Iris virginica* estão "encostadas" uma na outra, enquanto a classe *Iris setosa* mantém-se distante.

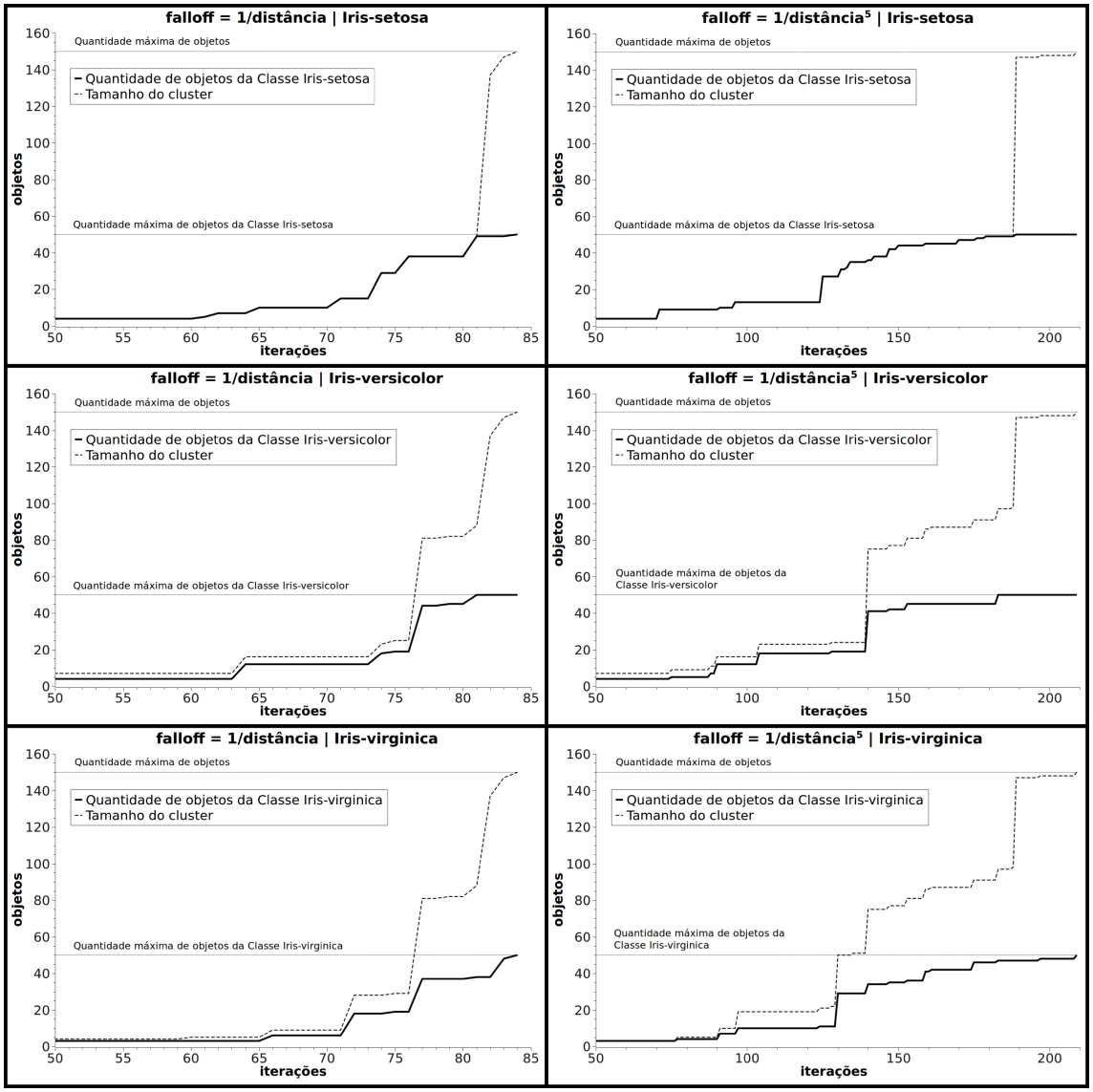

Figura 7. Resultados das simulações sobre o conjunto Iris.

4. Conclusões

O método proposto e estudado neste trabalho exibe características interessantes. Conceitualmente, é um método simples, baseado na gradativa troca de energia entre os elementos constituintes do sistema, os objetos. Contudo, essa simplicidade mostra-se capaz de agrupar objetos com distribuição relativamente complexa, como mostraram as simulações sobre o Conjunto de dados 1. Os resultados sugerem que esse método é capaz de separar e agrupar classes de formas arbitrárias, desde que a distância entre objetos adjacentes de uma mesma classe seja, em geral, menor que a distância entre objetos de classes distintas.

Alguns conjuntos de dados não exibem, entretanto, uma distribuição contendo classes tão bem-separadas. Neles ocorrem pareamento ou sobreposição de classes. Os resultados das simulações sobre o Conjunto de dados 2 e sobre o conjunto Iris exibem a dificuldade em se agrupar objetos com uma distribuição desse

tipo.

O método de agrupamento aqui apresentado é considerado como sendo um processo de aprendizado não-supervisionado, ou seja, toma um conjunto contendo somente objetos não-classificados (não-rotulados) [Mitchell 1997, Jain et al. 1999, Alpaydin 2004, Xu and Wunsch 2005, Bishop 2006, Hastie et al. 2009]. Pretende-se estender esse método de modo a contemplar conjuntos de objetos parcialmente classificados, ou seja, conjuntos contendo uma mistura de objetos classificados e não-classificados. Este último processo é denominado aprendizado semissupervisionado [Zhu 2005]. Espera-se um considerável aumento de desempenho através desse novo método semissupervisionado.

Referências

Uci machine learning repository. Disponível em <`http://archive.ics.uci.edu/ml/index.html`>. Acessado em maio de 2011.

Abraham, A., Das, S., and Roy, S. (2007). *Soft Computing for Knowledge Discovery and Data Mining*, chapter Swarm Intelligence Algorithms for Data Clustering. Springer Verlag.

Alpaydin, E. (2004). *Introduction to Machine Learning*. The MIT Press.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer.

Blatt, M., Wilseman, S., and Domany, E. (1997). Data clustering using a model granular magnet. *Neural Computation*, 9(8):1805–1842.

Castro, L. N. (2007). Fundamentals of natural computing: an overview. *Physics of Life Reviews*, 4(1):1–36.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, 2. edition.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys*, 31(3).

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Series in Computer Science. McGraw-Hill.

Oliveira, T. B. S. (2008). Clusterização de dados utilizando técnicas de redes complexas e computação bioinspirada. Master's thesis, Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo – São Carlos.

Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678.

Zhao, L., Damiance Jr., A. P. G., and Carvalho, A. C. P. L. F. (2005). *Advances in Natural Computation*, volume 3610 of *Lecture Notes in Computer Science*, chapter A Self-organized Network for Data Clustering. Springer.

Zhu, X. (2005). Semi-supervised learning literature survey. Computer Sciences, University of Wisconsin-Madison, n. 1530.