Problema Ant-based Clustering: Algoritmo de solução com Dados reais

André Nunes[1], **Rafael Stubs Parpinelli**[2]

[1]Estudante de Ciência da Computação – Universidade Do estádo de Santa Catarina (UDESC)

[2]Professor de Ciência da Computação – Universidade Do estádo de Santa Catarina (UDESC)

dekonunesss@gmail.com, rafael.parpinelli@udesc.br

Resumo. *Com o crescimento da computação os algoritmos utilizados no passado começaram a ficar defasados e, por consequência, não serem mais viáveis para o uso no dias de hoje. Atualmente no campo da Inteligência Artificial buscam-se soluções baseadas na natureza. A simulação apresentada é feita em um algoritmo de colonização de formigas, onde com movimentos aleatórios e tomadas de decisões baseadas em estátísticas os itens espalhados em uma matriz começam a criar diferentes aglomerados, dependendo de sua similaridade.*

1. Introdução

Com os avanços da computação a quantidade de dados aumentou significativamente, em decorrência disso houve o surgimento de novas pesquisas, uma delas é a de Data Mining, que consiste em retirar conhecimento de uma grande quantidade de dados [Jafar and Sivakumar 2010], para tal fim existem inúmero tipos de algoritmos. A pesquisa de tipo Data Mining não é usada apenas na área da computação, como também na área da sociologia, psicologia, comércio, biologia, entre outros [Handl et al. 2003]. Será apresentado no presente artigo um modelo de Data Mining baseado na maneira que as formigas organizam seus formigueiros. O artigo visa a elaboração de um algoritmo de agrupamento baseado em ant-based clustering utilizando dados reais, de modo que fiquem agrupados dependendo da sua similaridade.

O artigo é composto da problemática de ant-based clustering com dados reais na seção 2, o algoritmo criado e o detalhamento do mesmo na seção 3, os testes e análises sendo discutidos na seção 4 e a conclusão do artigo na última seção.

2. Problema Ant-based Clustering

As formigas da classe operária fazem a limpeza das formigas mortas na colônia agrupando-as e criando cemitérios. O movimento das formigas (agentes) é aleatório e sua decisão de pegar ou largar um item também é realizada de forma aleatória, porém com a influência de seu raio de visão e dos itens ao seu redor. A probabilidade do agente deixar o objeto (dados) acrescida a cada item semelhante ao seu no seu redor, assim como a probabilidade de pegar um item é acrescida quando um item é diferente dos seus arredores ou ainda nenhum item em seu campo de visão [Jafar and Sivakumar 2010].

O principal objetivo do algoritmo é agrupar dados parecidos em uma grade usando como base a formação de grupos do algoritmo de Ant-Based Clustering.

2.1. PEAS

PEAS (Performance, Environment, Actuators, Sensors) é uma ótima estratégia para facilitar o entendimento do problema e detalhar melhor as partes para que se consiga criar um algoritmo menos propenso a falhas.

2.1.1. Características do ambiente

- Agentes: Formigas (agentes que podem se mover);
- Medida de desempenho: velocidade das formigas, quantidade de objetos carregados e máximo de itens agrupados;
- Ambiente: Recipiente virtual para o deslocamento dos agentes;
- Sensores: Antenas, patas, vizinhança;
- Atuadores: Braço das formigas, movimento, tomada de decisão.

2.1.2. Propriedades do ambiente

- Ambiente de tarefa: Matriz;
- Observável parcialmente: A formiga (agente) tem apenas a visão do seu raio de visão e não da matriz toda;
- Estocástico: O movimento das formigas (agentes) é totalmente aleatório;
- Sequencial: O item colocado em um determinado local influencia no resultado de outra formiga;
- Dinâmico: As formigas mudam o ambiente pegando ou deixando um item;
- Discreto: A quantidade dos dados, formigas, itens e matriz não se altera;
- Multiagente: Existe mais de um agente no ambiente.

3. Algoritmo Implementado

O algoritmo para a resolução do problema Ant-based Clustering com dados reais pode ser implementado de diferentes maneiras, podendo ser simples ou com várias inteligências. O algoritmo criado no presente trabalho contém algumas inteligências, como a de que cada formiga pode ter seu próprio raio de visão, bem como no final das interações todas as formigas continuam até localizar um local para deixar os itens, por exemplo.

Toda a lógica se baseia em uma matriz, nela ficam representados os itens e as formigas que são modificados durante o processo e usados para os algoritmos de decisão.

Para acelerar o processo, no início do programa é feito o povoamento dos itens e das formigas e aguarda-se alguns segundos para fazer a análise da matriz inicial. Após isso, é desativa a interface gráfica e mostra-se o resultado final, com os itens agrupados em aglomerados de similaridades.

A tecnologia usada foi a lingaguem C++ e a biblioteca Simple and Fast Multimedia Library (SFML-2.3.1) para a interface gráfica.

3.1. Implementação

Como descrito por [Handl et al. 2003] a matriz é povoada randomicamente, tanto as formigas quanto os itens nela espalhados. Os itens são lidos de uma base de dados, podendo

variar a quantidade de dimensão. Com o objetivo de facilitar a visualização, as células são diferenciadas por cores distintas, de modo que toda célula de cor branca significa que não há nada nela, as células de cor **azul escuro**, azul claro, magenta e **verde escuro** representam um item, as células de cor **vermelha** representam que há uma formiga sem item; as células de cor **preto** representa uma formiga com item; as células de cor **verde claro** representa que há uma formiga sem possuir item e que não há nenhum item abaixo dela; quando as células forem amarelas representa que há uma formiga carregando um item com outro item abaixo dela. O movimento das formigas pode ser para qualquer lado, sendo possível para cima ou para baixo, para a direita ou para a esquerda, bem como para suas diagonais. O movimento é aleatório, tendo como única restrição de movimento o caso onde o local escolhido já tenha outra formiga.

Usando dados reais o algoritmo tem mudanças em suas fórmulas, agora existindo uma fórmula para pegar um item, e outra para deixar um item. Essas decisões continuam sendo tomadas dentro de uma análise estátística onde a formiga analisa dentro do seu raio de visão todas as células e guarda em sua variável o valor de células que foi possível analisar e a quantidade de itens que estão nessas células analisadas bem como a similaridade entre cada item analisado. Após, é aplicada a fórmula $P_p(i) = (k_p/k_p + f(i))^2$ onde o P_p representa a probabilidade de pegar o item, k_p é uma constante e $f(i)$ é oriundo de uma fórmula explicado abaixo [Handl et al. 2003]. Para a formiga que deixa um item foi usado a fórmula $P_d(i) = (f(i)/k_d + f(i))^2$ citada no artigo [Handl 2003], seguindo a mesma lógica de k_d sendo uma constante e $f(i)$ da mesma fórmula explicada abaixo. Para o experimento foi usado $k_p = 0.1$ e $k_d = 0.3$ seguindo a lógica de [Handl 2003]. Após a aplicação da fórmula é gerado um número aleatório de 1 a 100, caso a formiga esteja carregando um item e esteja sobre uma célula em branco ela compara se o número gerado aleatoriamente é menor que o valor de probabilidade calculado na fórmula, caso verdadeiro, o item é colocado na célula e a formiga muda seu estádo de carregando um item para vazia, o oposto ocorre quando a formiga está sem um item e está sobre uma célula com item. Todos os cálculos são realizados novamente para saber se o local que o item se encontra é um local ideal (próximo de outros itens), caso não seja um local propício ela pega o item e continua sua busca para localizar um local que tenha itens com uma similaridade maior.

O cálculo do $f(i)$ é composto por duas fórmulas, sendo está o cálculo de distância e a sua fórmula. O cálculo da distância euclidiana depende do tipo de itens a serem analisados, como para esse experimento foram usado itens com dominio continuo o cálculo foi feito usando a fórmula da distância euclidiana:

$$D(x_i, x_j) = \sqrt{\sum_{k=1}^{n} \left(x_{i,k} - x_{j,k}\right)^2}$$

Figura 1. Fórmula da distância Euclidiana

Após conseguir o resultado da fórmula, é aplicada a fórmula abaixo. O $f(i)$ varia de [0,1], o valor de s^2 é a quantidade de celulas que a formiga pode observar $(2 * Raio + 1)^2$, o valor de α também é uma constante, onde quanto menor esse valor, mais rigoroso fica o cálculo da distância, sendo fixada em $\alpha = 0.5$ para o experimento com os itens

fornecidos pelo professor e $\alpha = 0.1$ para o experimento da iris. Segue a figura abaixo a fórmula de [Lumer 1994].

$$f(i) = \begin{cases} \frac{1}{s^2} \sum_j (1 - d(i,j)/\alpha) & \text{if } f(i) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Figura 2. Fórmula de Lumer

Para se chegar ao resultado final todas as formigas que não estejam carregando um item são retiradas para que não atrapalhem as que ainda não colocaram seus itens. O sistema só finaliza após todas as formigas terem colocado os itens nos locais encontrados por elas.

3.2. Software

Os parâmetros para o sistema são:

- Quantidade de linhas;
- Quantidade de colunas;
- Quantidade de formigas;
- Quantidade de interações;
- Quantidade máxima do raio.

O raio das formigas é heterogêneo, variando de um até o número escolhido no parâmetro, porém para os testes do algoritmo do presente artigo houve uma adaptação para que os agentes fossem homogêneos e com isso fosse possível uma melhor comparação nos resultados.

A quantidade de dados é dada pela quantidade de linhas do arquivo, sendo cada linha um item com suas respectivas dimensões e seu valor de referência.

4. Simulações e Resultados

Nas simulações foi usado a mesma base de dados, mudando apenas o valor do raio em um caso e a quantidade de formigas em outros para mostrar o que acontece. Os testes foram realizados usando uma matriz 40x40, 250 itens e 500000 interações. A comparação entre os testes foi feita de maneira visual entre as imagens inicias e finais dos testes.

4.1. Dados

Foram usados 400 itens para o experimento 1 e 150 itens para o experimento 2, no experimento 1 os itens continham 2 dimensões para o experimento 2 tinha 4 dimensões. São dados conhecidos, o experimento 1 faz partes dos dados chamados iris onde organiza a iris para escolher uma lente de contato, sendo 50 Iris-setosa, 50 Iris-versicolor e 50 Iris-virginica, os outros 40 dados do experimento 1 foram cedidos pelo professor.

4.2. Experimento 1 - Raio 1 e 30 formigas

O experimento de Raio 1 com 30 formigas teve melhor resultado que os demais, criando pequenos aglomerados de cada tipo de itens. Fica claro pela imagem a união das bordas no aglomerado de cor magenta e do azul claro, onde uma ponta do aglomerado está do outro lado da grid.

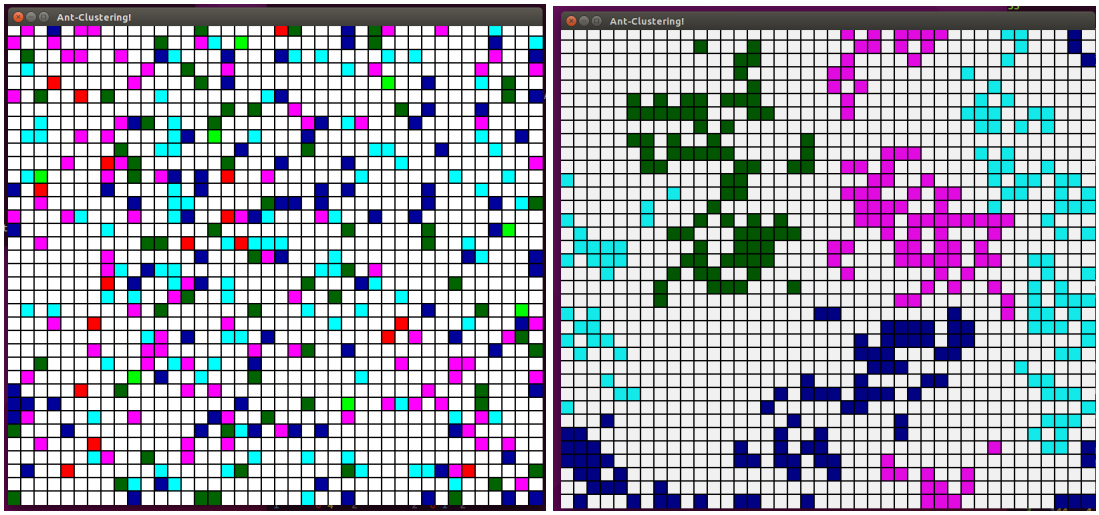

Figura 3. Raio 1 - inicial e final

4.3. Experimento 1 - Raio 2 e 30 formigas

O experimento de Raio 2 obteve um agrupamento mediano, mas ainda assim com um resultado satisfatório. Originou aglomerados com as bordas mais espaçadas e dispersas que o experimento com Raio 1, isto se deve ao fato de se analisar uma quantidade maior de dados, não muito diferente do trabalho 1.

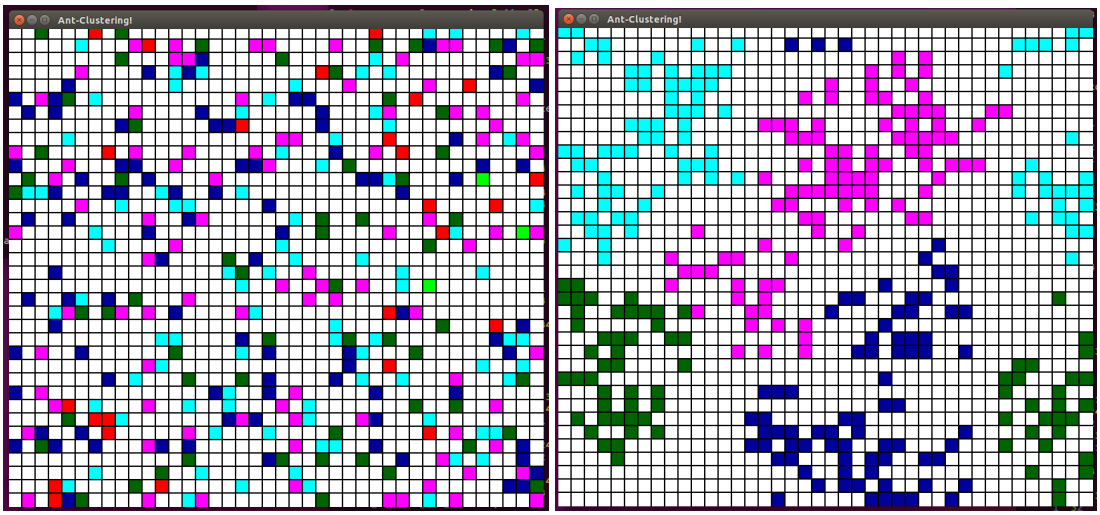

Figura 4. Raio 3 e 30 formigas

4.4. Experimento 2 - Iris - Raio 1 e 30 formigas

Para esse experimento houveram algumas modificações, como o mudança do tamanho da grid para 30x30, mudança do alfa para $\alpha = 0.1$ e quantidade de interações foi para 5000000, essas mudanças foram feitas devido ao sistema ter complexidade maior que o experimento 1, com os mesmo parâmentros do experimento 1 ele não teve um bom resultado. Os itens de cor azul claro e azul escuro tem uma diferença muito pequena de semilaridade, devido a isso acontece de 1 deles se unir a outra cor ou criar aglomerados próximos um do outro.

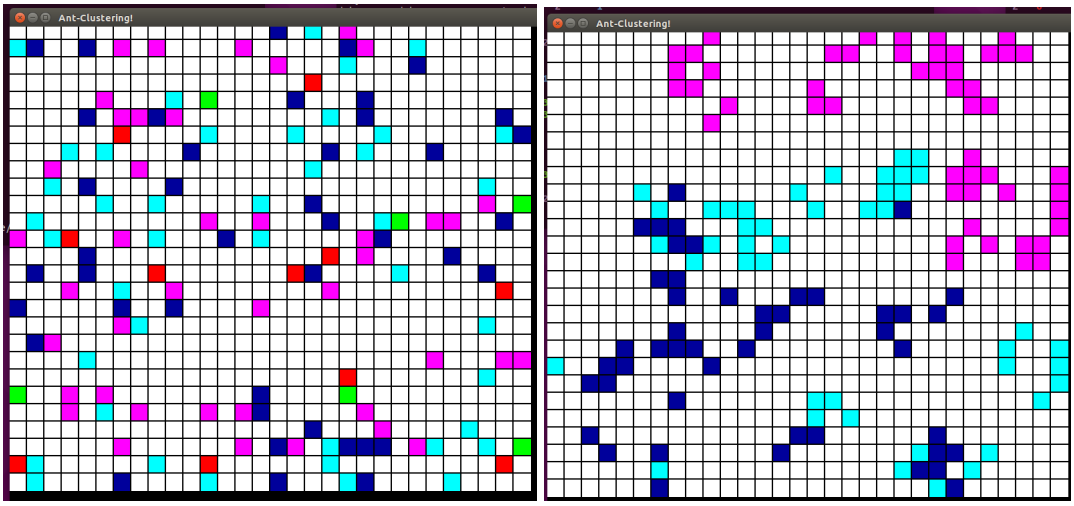

Figura 5. Raio 1 e 90 formigas

5. Conclusão

O algoritmo demonstra que essa técnica pode ser aplicada para agrupamentos dos dados pela sua similaridade com um resultado satisfatório, podendo ser aplicado na área de mineração de dados para facilitar as buscas de informações, um exemplo desses dados são os dados da iris que foi usado para ser agrupado pelo algoritmo de Ant-Based Clustering.

Nas simulações foi mostrado que mesmo com uma quantidade pequena de formigas é possível se conseguir um bom resultado. Mostrou-se também que dois ou mais itens podem ter uma pequena similaridade, com uma pequena chance de se agruparem ou agrupando-se próximos de outro grupo.

Referências

Handl, J. (2003). Ant-based methods for tasks of clustering and topographic mapping:extensions, analysis and comparison with alternative methods. *Master's thesis, University of Erlangen-Nuremberg, Germany*.

Handl, J., Knowles, J., and Dorigo, M. (2003). Ant-based clustering: a comparative study of its relative performance with respect to k-means, average link and id-som. In *Proceedings of the Third International Conference on Hybrid Intelligent Systems, IOS Press*.

Jafar, O. M. and Sivakumar, R. (2010). Ant-based clustering algorithms: A brief survey. *International Journal of Computer Theory and Engineering*, 2(5):1793–8201.

Lumer, F. (1994). Diversity and adaptation in populations of clustering ants. *In Proceedings of the Third International Conference on Simulation of Adaptive Behaviour: From Animals to Animats*.