Algoritmo de clusterização através de inteligência de enxame

Gabriela Moreira
Departamento de Ciência da Computação
Universidade do Estado de Santa Catarina
Joinville, Brasil
gabrielamoreira05@gmail.com

Luiz Gustavo Eburneo
Departamento de Ciência da Computação
Universidade do Estado de Santa Catarina
Joinville, Brasil
botucacontato@gmail.com

***Abstract*—Algoritmos de clusterização inspirados em colônia de formigas são utilizados como métodos para realizar agrupamento de dados. Este relatório apresenta como foi feita a descrição, caracterização, implementação e resultados dos algoritmos implementados em cima de três bases de dados distintas.**

***Index Terms*—clusterização, algoritmo bio-inspirado, formigas, inteligência de enxame, inteligencia artificial**

I. Introdução

Algoritmos de clusterização são parte da área de aprendizado de máquina não supervisionado, onde não se conhece os grupos específicos nos quais os dados devem ser agrupados. Métodos inspirados em fenômenos biológicos como a colônia de formigas são utilizados como abordagens probabilísticas, e são conhecidos como inteligencia de enxame.

Para isso, é implementado um sistema multi-agente com técnicas meta-heurísticas de otimização para agrupar dados com características multi-dimensionais.

O agrupamento ou clusterização de dados é visto com importância crescente conforme a expansão das bases de dados devido ao alto fluxo de coleta de informação. Sua aplicação na classificação de dados pode ser feita nos mais diversos campos, desde redes sociais até diagnósticos médicos.

Em decorrência do relatório, será explicado a metodologia do desenvolvimento que conta com a descrição do problema, seguido com os modelos que foram testados e suas características como variáveis, fórmulas, critério de parada e o *dataset*. Em seguida, é feita uma análise dos resultados obtidos e por fim uma conclusão.

II. Metodologia do Desenvolvimento

A. Descrição do problema

O experimento a ser reproduzido é a limpeza de um formigueiro. Algumas espécies juntam corpos e parte de corpos de formigas mortas em regiões específicas do formigueiro, para organizar e melhorar o deslocamento das formigas. Pequenos amontoados vão crescendo atraindo uma maior quantidade de corpos naquela região do espaço.

A seguir, será mostrado a descrição do problema para o agrupamento:

- Ambiente: Formigas vivas e mortas e espaço de atuação.
- Sensores: Posição das formigas vivas e mortas e perceber o local em que ela se encontra.
- Atuadores: Mover formigas mortas (pegar, soltar) e se mover.
- Medida de Desempenho: Amontoar a fim de liberar mais caminho e o tempo para finalizar a tarefa.
- Propriedades do ambiente:
  - Parcialmente observável: Pois é delimitação de área.
  - Estocástico: Pois não há como saber a direção que a formiga irá fazer.
  - Sequencial: Qualquer ação feita, terá um efeito na minha medida de desempenho.
  - Dinâmico: O que uma formiga fizer irá alterar a decisão de outra.
  - Contínuo: Entropia no final.
  - Discreto: Pois há a posição da formiga, a posição final, o caminho usado para movimentar a formiga morta, a quantidade de itens mortos, a quantidade de ”placas” e a quantidade de células livres.
  - Multiagente. Pois há vários agentes que atuam para o objetivo final.

Para atingir o comportamento desejado e dar ”vida” ao agrupamento, foi necessário definir certas estratégias que foram utilizadas em nossas simulações. Para o movimento, as formigas caminham aleatoriamente. Para a decisão de pegar um item, a formiga viva precisa enxergar o item no seu raio de visão, ela não pode estar segurando outro item, se o ”item” escolhido não estiver amontoado, então há uma chance alta de pegá-lo, caso contrário, a chance será baixa.

Para largar um item, a formiga viva tem que estar segundo algo, estar perto de um amontoado e a célula que será largado deve estar vazia.

Assim, para alcançar a clusterização, foi desenvolvida uma simulação de um ambiente com a combinação de formigas vivas e mortas, onde as formigas vivas agrupam as mortas em montes (chamados de placas).

B. Itens uniformes

No primeiro estágio, foi desenvolvido um modelo considerando todos os itens agrupados em uma única placa. Assim, não havia necessidade de cálculo de dissimilaridade, considerando apenas a quantidade de itens vistos. Para esse

modelo, foram necessários apenas os seguintes parâmetros:

Nome	Descrição
N	tamanho do grid (NxN)
n_{dead}	número de itens (formigas mortas)
n_{ants}	número de agentes (formigas vivas)
life	tolerância de turnos ociosos
vision_range	alcance de visão dos agentes

As fórmulas utilizadas para a probabilidade de um agente pegar e soltar um item foram:

$$P_{pick} = 1 - n_{local}/field\_size$$

$$P_{drop} = n_{local}/field\_size$$

Onde n_{local} é a quantidade de itens vistos e $field\_size$ é a quantidade de campos dentro do alcance de visão definido.

C. Itens com classes

Para esse estágio, o modelo foi evoluído para agrupar itens de classes diferentes em placas diferentes. Assim, cada item é descrito com características multi-dimensionais e os agentes consideram a dissimilaridade dessas características para fazer o agrupamento. Essa dissimilaridade é calculada através da distância euclidiana, que para duas dimensões é dada por:

$$D_{Euclidiana} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Assim, as probabilidades de pegar e soltar um item são definidas por:

$$P_{pick} = min(\frac{1}{f(i)^2}, 1)$$

$$P_{drop} = max(f(i)^4, 1)$$

onde

$$f(i) = \frac{1}{\sigma^2} \sum_{vizinho} max(1 - \frac{d(item, vizinho)}{\alpha}, 0)$$

com vizinho sendo um item dentro do alcance de visão do agente.

Assim, os parâmetros para esse estágio são:

Nome	Descrição
N	tamanho do grid (NxN)
n_{ants}	número de agentes (formigas vivas)
life	tolerância de turnos ociosos
vision_range	alcance de visão dos agentes
α	porcentagem de itens similares
σ^2	penalização para o somatório

D. Critério de parada

O critério de parada é analisado individualmente para cada agente. A cada iteração, um atributo do agente é:

- Incrementado, caso ele não pegue ou solte um item,
- Zerado, caso ele pegue ou solte um item.

Se esse atributo ultrapassar o valor estabelecido para o parâmetro *life* e o agente não estiver carregando um item, ele é destruído. Assim, o método encerra quando todos os agentes tiverem sido destruídos

III. Descrição dos Experimentos e Resultados Obtidos

A. Primeiro experimento

O primeiro conjunto de dados para testar o método é extremamente comportado e possui quatro classes diferentes. A distribuição dos dados pode ser visualizada na Figura 1.

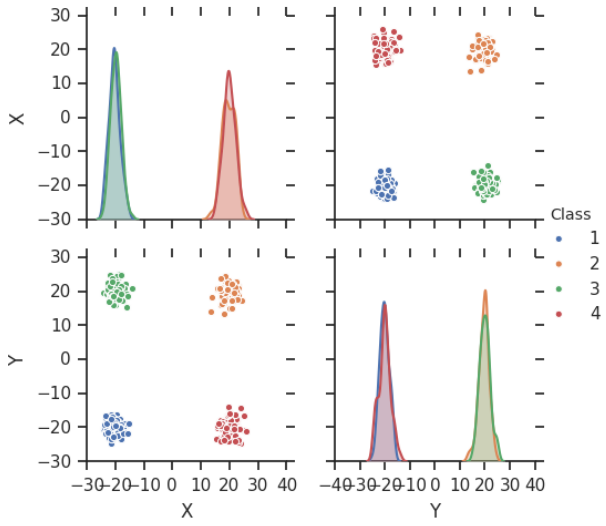

Fig. 1. Conjunto de Dados 1

Os valores dos parâmetros para a execução do método foram escolhidos manualmente:

Parâmetro	Valor
N	30
n_{ants}	70
life	100
vision_range	1
α	9
σ^2	4

B. Segundo experimento

O segundo conjunto de dados é ainda bem comportado, mas agora com 15 classes diferentes. A primeira percepção é que o número alto de classes eleva bastante a dificuldade do problema, fazendo com que a velocidade de convergência aumente significativamente. O comportamento da distribuição desses dados é observado na Figura 2.

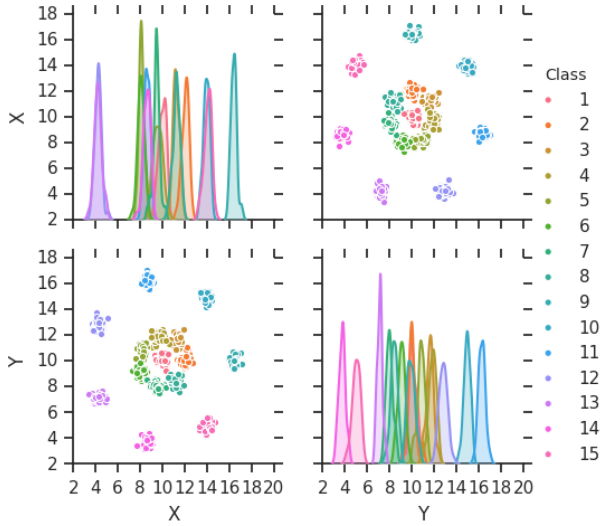

Fig. 2. Conjunto de dados 2

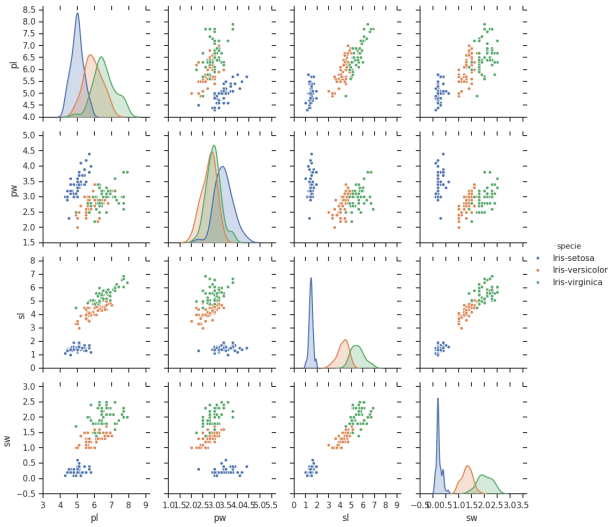

Fig. 3. Conjunto de dados 3

Para aplicar o método de clusterização para esses dados, foram usados os seguintes valores para os parâmetros:

Parâmetro	Valor
N	40
n_{ants}	100
life	100
vision_range	1
α	1.4
σ^2	4

C. Terceiro experimento

O terceiro conjunto é uma amostra utilizada em muitos estudos de estatística, classificação e clusterização: o conjunto Iris. Os dados são parcialmente bem comportados, com em um nível mais realístico. Existe um grupo bem distinto, e dois grupos com características bem similares. Assim, há dificuldade em encontrar parâmetros que realmente separem esses grupos. Esse conjunto também possui uma complexidade maior por ter quatro dimensões. A relação entre os dados para as dimensões pode ser observada na Figura 3.

Os parâmetros que apresentaram os melhores resultados foram:

Parâmetro	Valor
N	30
n_{ants}	20
life	100
vision_range	1
α	1.5
σ^2	3

IV. Análise dos resultados obtidos

A. Primeiro experimento

O comportamento nesse experimento mais simples é de agrupamento rápido dos itens em placas médias, e então da junção em placas maiores. Um problema encontrado foi o alto número de interação mesmo após atingir um estado satisfatório. A disposição final dos itens é conforme a Figura 4.

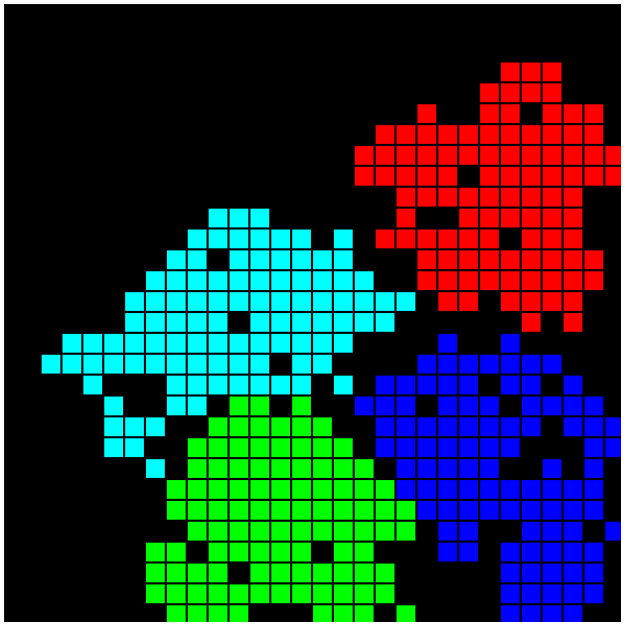

Fig. 4. Resultado experimento 1

B. Segundo experimento

Considerando grupos bem comportados mas com características mais próximas, foi necessária uma grande diminuição no parâmetro α. Com isso, as 15 placas passam a ser formadas, primeiramente de forma dispersa e então se agrupando. O número de iterações e, consequentemente, o tempo de execução é consideravelmente maior para esse experimento, considerando o aumento da quantidade de dados. Devido a

esse mesmo fator, o tamanho do grid e o número de agentes também tiveram de ser aumentados. As placas construídas são conforme a Figura 5.

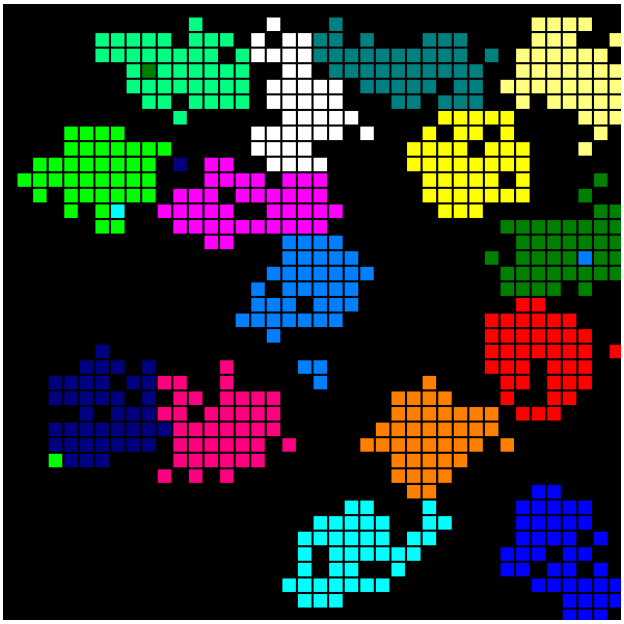

Fig. 5. Resultado experimento 2

C. Terceiro experimento

Esse experimento se assemelha mais com a condição de dados reais, onde o agrupamento é mais difícil. Com mais dimensões e menos distinção entre as classes, as placas ficam mais heterogêneas em relação aos dados completamente comportados. O resultado obtido encontra-se na Figura 6.

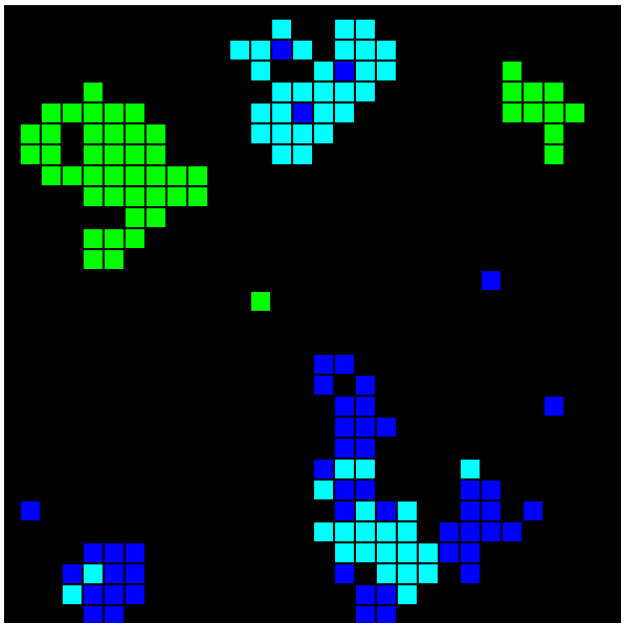

Fig. 6. Resultado experimento 3

V. Conclusão

O algoritmo de clusterização para o primeiro e segundo experimento funcionou bem em uma velocidade aceitável com base nos valores dos parâmetros utilizados. Com uma pequena mudança na base de dados, o tempo para uma boa convergência pode variar bastante e deve-se encontrar outros valores para os parâmetros, visto que eles são específicos para cada base e atribuídos manualmente.

Para o terceiro experimento, os resultados não são tão satisfatórios, e seria necessário um refinamento no método para que haja uma aplicação mais concreta dos resultados. Entre as melhorias possíveis, estaria uma abordagem de movimento mais estratégica do que a aleatória, dinamização dos parâmetros (em específico, dos parâmetros α e vision_range) e um grid esférico, sem delimitadores.

Os resultados demonstram que uma método simples simulando agentes com operações básicas e independentes é suficiente para clusterizar grupos bem comportados.

References

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] Handl, J, Knowles, J Dorigo, M 2003, Ant-based clustering: a comparative study of its relative performance with respect to k-means, average link and 1d-som. in Design and Application of Hybrid Intelligent Systems. Citeseer, pp. 204-213.

[3] RUSSEL, Stuart, NORVIG Peter. Inteligência Artificial. 2004