Análise de metaheurísticas com K-means

Gabriel dos Santos Sereno[1]

Abstract

Com o aumento exponencial da Inteligência Artificial e principalmente do uso de estratégias de clusterização, este artigo visa analisar a aplicação de algoritmos metaheurísticos para a otimização do algoritmo K-Means. Para isso, foi utilizado duas metaheurísticas: Differential Evolution e Evolution Strategies, ambos em conjunto com o K-Means em bases de estudos, como: Wine, Breast Cancer e Iris para efetuar o benchmark entre as soluções. De fato, os resultados foram satisfatórios e potencializados com os algoritmos metaheurísticos, que no que lhe concerne, procuram um resultado ótimo sem a necessidade de fornecer pontos dos centroides iniciais de forma empírica do usuário, automatizando a escolha dos melhores pontos iniciais.

Keywords: `K-Means`, Differential Evolution, Evolution Strategies, *Clusterização*, *Metaheurísticas*

1. Introdução

Atualmente os algoritmos de Inteligência Artificial estão crescendo exponencialmente em popularidade no meio acadêmico e também no meio empresarial. Esse crescimento é devido a grande eficácia na área de classificação e clusterização. Por isso, esse relatório tem o foco de mostrar um comparativo entre algoritmos de clusterização com otimização de metaheurísticas, são eles: Differential Evolution e Evolution Strategies.

[1] Aluno do curso de mestrado com enfase em Inteligência Artificial na Universidade Federal do Espírito Santo

As bases escolhidas para testar o algoritmo K-Means em conjunto com a otimização, foram as bases: Wine, Breast Cancer e Iris.

Para as comparações, os dados foram retirados a partir de 10 repetições para cada algoritmo. Dessa forma, os dados retirados foram: média, acurácia máxima, acurácia mínima e desvio padrão, além dos melhores centroides encontrados.

2. Metaheurísticas

O foco do relatório é voltado inteiramente na análise do poder das metaheurísticas em conjunto com o K-Means, bem como na assertividade.

2.1. Differential Evolution

O algoritmo *Differential Evolution* visa procurar os melhores valores a partir do *Crossover* e da Mutação, verificando se o passo atual é melhor que o passo anterior. Nisso, o algoritmo consegue encontrar bons resultados em pouco tempo e com pouco processamento.

Os métodos utilizados no *Differential Evolution* seguem a ideia original do algoritmo. Um deles é o *Crossover*, para gerar novos estados filhos a partir de dois conjuntos pais. Essa estratégia é gerado a partir de dois principais fatores, gera um valor randômico e verifica se é menor que a taxa de *Crossover* para recombinação e assim é gerado o novo estado.

Já a mutação é feita em determinadas partes do estado para ter uma diferenciação do estado original. Essa técnica é utilizada para verificar se o novo valor é melhor que o anterior, para explorar o ponto atual em que esse estado está.

Algorithm 1 Differential Evolution

```
1: Gera os limites conforme a base
2: Gera a população inicial
3: Avalia toda a população
4: for Nº de Iterações do
5:     for Tamanho da População do
6:         Seleciona três candidatos
7:         Faz a mutação entre os candidatos selecionados
8:         Faz o Crossover
9:         Avalia os resultados
10:        if Resultado Crossover é maior que Resultado de um individuo? then
11:            Atualiza o individuo pelo objeto do Crossover
12:        end if
13:    end for
14:    Guarda o melhor resultado da população
15:    if Melhor individuo da população é maior Melhor individuo encontrado?
   then
16:        Atualiza o melhor individuo pelo melhor da população
17:    end if
18: end for
19: return Retorna o melhor individuo encontrado
20:
```

2.2. Evolution Strategies

O algoritmo *Evolution Strategies* visa procurar melhores valores a partir de novos estados criados a partir da função provido. Nesse caso, esse novo estado é provido a partir do K-Means e assim é multiplicado com o estado selecionado, convergindo para o melhor ponto encontrado.

Algorithm 2 Evolution Strategies

```
Gera os limites conforme a base
Gera a população inicial
for Nº de Iterações do
    Avalia toda a população
    Seleciona os melhores
    for Tamanho da População do
        Verifica se o resultado atual é maior que o melhor individuo
        if Resultado atual é maior que Resultado do melhor individuo? then
            Atualiza o melhor individuo com o atual
        end if
        Cria indivíduos a partir do pai com indivíduos criados aleatoriamente
    end for
end for
return Retorna o melhor individuo encontrado
```

3. Comparativo entre os resultados

Para realizar o comparativo, cada algoritmo foi repetido por 10 vezes para gerar resultados sólidos. Com isso, foram gerados os seguintes resultados:

3.1. Iris

A base *Iris* é uma das mais famosas bases em comparação com as demais, contendo 150 linhas com 4 colunas, além de ter 3 classes que torna o trabalho

do classificador fácil. Os resultados dos classificadores utilizados estão na tabela a seguir:

Indicadores	**Resultados**
Média	90,26%
Desvio Padrão	0,0095
Mínimo	89,33%
Máximo	92%

Table 1: Resultados do algoritmo Differential Evolution na base Iris

O algoritmo *Differential Evolution* apresentou excelentes e consistentes resultados devido à característica de *Crossover* e também de Mutação, fazendo com que os novos estados criados a partir dos estados pais, sejam explorados em um determinado mínimo local ou global rapidamente.

Indicadores	**Resultados**
Média	73%
Desvio Padrão	0,1102
Mínimo	64,66%
Máximo	90,66%

Table 2: Resultados do algoritmo Evolution Strategies na base Iris

Já no algoritmo *Evolution Strategies* teve uma grande probabilidade de não gerar o melhor estado para gerar os centroides do *K-Means*. Isso ocorre porque os passos não chegaram aos locais, por utilizar um número de iterações baixos, já que o aprendizado é efetuado a cada iteração.

3.2. Wine

A base *Wine* contem apenas 3 classes e cerca de 180 linhas com 13 colunas. A seguir, as tabelas com os resultados encontrados:

Indicadores	Resultados
Média	70,28%
Desvio Padrão	0,0016
Mínimo	70,22%
Máximo	70,78%

Table 3: Resultados do algoritmo Differential Evolution na base Wine

O algoritmo *Evolution Strategies* obteve resultados similares ao *Evolution Strategies*. É possível analisar esse fato na tabela a seguir:

Indicadores	Resultados
Média	54,55%
Desvio Padrão	0,1224
Mínimo	39,88%
Máximo	70,22%

Table 4: Resultados do algoritmo Evolution Strategies na base Wine

Ambos os resultados apresentaram baixa acurácia, pois o conjunto de dados contem algumas características que não permite separar os centroides por classes. Dessa forma, o recomendado é utilizar o algoritmo PCA para encontrar as melhores características e diminuir a dimensionalidade, que através dessa seleção, remove características que podem atrapalhar a classificação e assim aumentar a taxa de acurácia. Além disso, os algoritmos apresentariam melhor desempenho, pois o *Evolution Strategies* dependendo da base escolhida, pode levar vários minutos até achar uma solução.

3.3. Breast Cancer

A base *Breast Cancer* contêm 2 classes apenas, significando se o paciente possui ou não o câncer. Além disso, é uma base mediana, contendo 569 linhas com 30 colunas. Com essas características, os algoritmos tendem a ter um excelente desempenho, pois contem um bom número de linhas e colunas. O

trabalho torna-se mais fácil ao classificar apenas duas classes. Esses resultados encontrados estão na tabela a seguir:

Indicadores	**Resultados**
Média	91,91%
Desvio Padrão	0,0013
Mínimo	91,56%
Máximo	92,09%

Table 5: Resultados do algoritmo Differential Evolution na base Breast Cancer

O algoritmo *Evolution Strategies* obteve resultados similares ao *Evolution Strategies*. É possível analisar esse fato na tabela a seguir:

Indicadores	**Resultados**
Média	88,43%
Desvio Padrão	0,0856
Mínimo	62,74%
Máximo	91,56%

Table 6: Resultados do algoritmo Evolution Strategies na base Breast Cancer

Novamente os resultados foram similares em ambas as técnicas, entretanto o algoritmo de *Evolution Strategies* leva muito tempo de processamento, devido ao alto número de iterações necessárias para finalizar o algoritmo.

4. Conclusões

Trabalhar com comparações é essencial para achar o melhor resultado e o classificador que encaixa melhor a um determinado problema. Com os testes feitos desse artigo, foi possível observar a diferença das metaheurísticas com o K-Means, principalmente no uso do tempo e computacional. É possível concluir que as metaheurísticas são mais eficazes em conjunto com o K-Means no cenário em que você necessita de descobrir melhores centroides sem o esforço manual

ou repetitivo, fazendo com que seja automático o descobrimento dos melhores centroides.

Ademais, as metaheurísticas se demonstraram poderosas em aprimorar os resultados do K-Means, além de ser facilmente implementados e altamente customizáveis. Diante disso, deve-se analisar o uso computacional se é viável ou não em um projeto que visa classificar rapidamente bases de dados.

Esse trabalho contribuiu com o aprofundamento das técnicas de metaheurística e de testes, possibilitando o aprendizado mais palpável, na prática. As descobertas ao implementar empiricamente a teoria foram fundamentais para consolidar e tornar base para situações reais. Por isso, para contribuições futuras, sugiro a utilização de outras metaheurísticas.

References

- Slides, artigos e notas transmitidas em aula das disciplinas tutoradas pelo professor.