Análise de metaheurísticas em problemas de classificação

Gabriel dos Santos Sereno[1]

Abstract

Este artigo tem como objetivo analisar a aplicação de algoritmos metaheurísticos em bases acadêmicas para construir um comparativo com a taxa de acurácia e demais dados. Nisso, foi construído três algoritmos, são eles: Hill Climbing, Simulated Annealing e Genetic, além de utilizar o algoritmo do Heterogeneous Pooling com os classificadores GaussianNB, DecisionTreeClassifier e o KNeighborsClassifier, utilizando o ScikitLearn. Dessa forma, as base de dados também foram utilizadas com o pacote ScikitLearn e são elas: Wine, Breast cancer e Digits.

Keywords: `C`lassificadores, Base de dados, Algoritmos, *ScikitLearn*, *Metaheurísticas*

1. Introdução

No contexto empresarial e científico, os problemas envolvendo inteligência artificial para a resolução de problemas se tornou extremamente complexo. Com isso, é importante a utilização de técnicas de metaheurística para potencializar a decisão e a assertividade dos algoritmos de inteligência artificial.

Nesse contexto, o artigo tem como objetivo comparar os algoritmos de metaheurística em conjunto com classificadores para determinar qual metaheurística potencializa melhor a busca por uma solução ótima nas bases de dados. Nisso, foi utilizados comparativos de acurácia, testes estatísticos, gráficos *Box plot* para

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

ilustrar de forma clara as diferenças das técnicas. As metaheurísticas testados foram: *Hill Climbing*, *Simulated Annealing*, *Genetic* em conjunto com o *Heterogeneous Pooling* com os classificadores *GaussianNB*, *DecisionTreeClassifier* e o *KNeighborsClassifier* advindos do pacote *ScikitLearn*.

Todas as bases foram padronizadas com o *StandardScaler*, além de passarem pelo *Cross-validation* e o *Grid Search* para a busca de hiperparâmetros. Após a obtenção de dados, o comparativo foi feito com testes estatísticos como *T-Student* e o *Wilcoxon*, além de tabelas com as informações da mediana, desvio padrão, acurácia e os suportes superiores e inferiores de cada classificador testado.

2. Metaheurísticas

O foco do artigo é voltado inteiramente na análise do poder das metaheurísticas, bem como na assertividade e técnica em comparação com técnicas simples. Além disso, para engrandecer o conhecimento transmitido por este artigo, foi implementado os algoritmos de *Hill Climbing*, *Simulated Annealing* e o *Genetic* em conjunto com o *Heterogeneous Pooling*.

2.1. Hill Climbing

O algoritmo *Hill Climbing* tem como objetivo procurar melhores valores a partir de pequenos movimentos, verificando se o passo atual é melhor que o passo anterior. Nisso, o *Hill Climbing* consegue encontrar bons resultados em pouco tempo e com pouco processamento. Entretanto, tende-se a não encontrar locais que podem ser um dos melhores resultados.

Nesse artigo, foi implementado o *Hill Climbing* determinístico, que pesquisa em todos os estados possíveis a melhor combinação de classificadores.

O único hiperparâmetro fornecido para a classe, foi o valor de tempo máximo de execução (*max_time*), para que a aplicação não fique por muito tempo em execução.

Os métodos utilizados no *Hill Climbing* seguem a ideia original do algoritmo. Um deles é a geração de estados, que é fornecido um estado pré-criado contendo

um *array* contendo zero e um ao longo do seu tamanho. Ao fim desse método, é utilizado a avaliação de todos os estados para selecionar o melhor e verificar se houve melhora com o passo anterior, caso não houve melhora na acurácia, o algoritmo retorna o melhor valor guardado.

Pela característica do *Hill Climbing*, o algoritmo geralmente retorna os primeiros classificadores como os melhores, pois tem grandes chances dos testes intermediários resultarem em valores menores que o ótimo já localizado, tornado extremamente difícil para chegar ao final.

2.2. Simulated Annealing

O algoritmo *Simulated Annealing* tem como objetivo procurar melhores valores similarmente a técnica empregada no resfriamento de materiais através da área metalúrgica. Essa técnica utiliza o conceito de temperatura, sendo que nas temperaturas maiores o algoritmo tende a procurar os melhores valores em todos os estados disponíveis. Ao chegar em temperaturas mais baixas, o algoritmo tende a procurar melhores combinações dos classificadores no melhor ponto encontrado. Dessa forma, diferentemente do *Hill Climbing*, o algoritmo perde a tendencia de ficar preso em pontos de ótimos locais e ganha mais poder para encontrar o ótimo global.

O método de criação de estados tem o objetivo de criar todos os vizinhos do estado exemplo e com isso é possível analisar boa parte da base dados e suas variações.

Para buscar em todas as possibilidades de combinações e não somente aos que tendem a ser melhores, foi implementado o método de probabilidade, fazendo com que em determinada temperatura possa ser possível procurar os melhores resultados, evitando estagnar em um resultado que é apenas um ótimo local.

Dessa forma, o *Simulated Annealing* tende a utilizar mais tempo que o *Hill Climbing*, pois busca mais combinações, tornando-se mais preciso.

2.3. Genetic

O algoritmo *Genetic* visa procurar os melhores valores através da busca genética, similar ao sistema biológico humano, no qual utiliza mutações, troca

de genes e outras características importantes desse algoritmo. A principal ideia do algoritmo *Genetic* é utilizar as melhores combinações para construir novas combinações com pequena diferenciação, com o objetivo de aprimorar a cada geração o resultado obtido.

Para isso, o método de criação de dados faz uma população para iniciar os testes. Nisso, é verificado qual de todas as combinações são as melhores para iniciar o elitismo e a mutação para gerar novas combinações de classificadores e novamente fazer mais uma etapa da criação da população. No artigo, para encerrar os ciclos a procura do melhor classificador, foi pelo método de parada após 120 segundos.

Como o algoritmo *Genetic* pode gerar combinações que já foram testadas e também com alto número de combinações por ciclo, tendendo a gastar várias horas até determinar o melhor classificador encontrado.

3. Comparativo entre os resultados

Para realizar o estudo das metaheurísticas, foi utilizado a técnica de validação cruzada estratificada (*Cross-validation*) de 10 *folds*, com 3 repetições internas.

Além disso, foi necessário configurar os hiperparâmetros, utilizado em cada algoritmo que está sendo representado na tabela a seguir:

Metaheurística	**Hiperparâmetros**
Hill Climbing	max_time = 120
Simulated Annealing	t = 200; alfa = 0.1; iter_max = 10; max_time = 120
Genetic	pop_size = 20; max_iter = 100; cross_ratio = 0.9; mut_ratio = 0.1; max_time = 120; elite_pct = 20
Heterogeneous Pooling	n_Samples = [1, 3, 5, 7]

Table 1: Distribuição dos hiperparâmetros das metaheurísticas

3.1. Digits

A base *Digits* são uma das maiores bases em comparação com as demais, contendo mais de 1700 linhas com 64 colunas, além de ter 10 classes que torna o trabalho do classificador um pouco mais árduo. Os resultados dos classificadores utilizados estão na tabela a seguir:

Técnica	**Acurácia**	**Desvio padrão**	**Lim. inf.**	**Lim. sup.**
Hill Climbing	0.9705	0.0113	0.9664	0.9745
Simulated Annealing	0.9701	0.0117	0.9659	0.9743
Genetic	0.9762	0.0105	0.9724	0.9800
Het. Pooling	0,9560	0,0130	0,9513	0,9607
RandomForest	0,9756	0,0103	0,9719	0,9793

Table 2: Resultados das técnicas na base *Digits*

Analisando o gráfico, percebe-se que todas as técnicas tiveram o mesmo desempenho, entretanto é importante lembrar que o algoritmo de *Hill Climbing* e o *Heterogeneous Pooling* foram os maís rápidos para serem executados, gastando menos de 5 minutos.

Entretanto, o *Genetic* obteve o melhor resultado, mas muito aproximado dos demais, em torno de 97% de acurácia. Esse resultado foi possível por causa das técnicas do próprio algoritmo genético que procura com maior amplitude um melhor resultado e em decorrência disso, é mais custoso.

Essa similaridade é observável no gráfico de *Boxplot* a seguir:

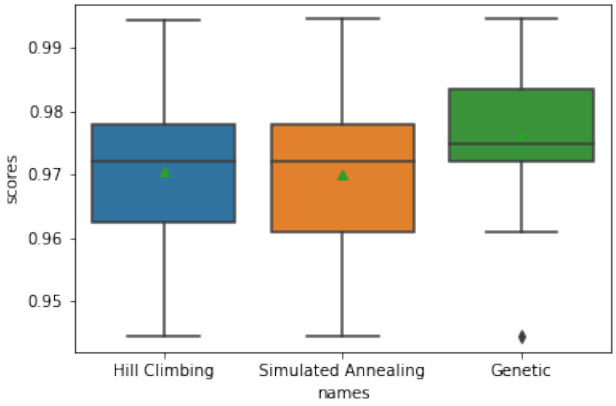

Figure 1: Gráfico *Boxplot* a partir dos resultados das técnicas na base *Digits*.

No gráfico fica ainda mais claro a similaridade entre os algoritmos, inclusive entre *Hill Climbing* e *Simulated Annealing* que basicamente tem os mesmos valores. Dessa forma, é necessário utilizar os testes de *Wilcoxon* e o *T-Student* para verificar se os métodos são realmente similares. A Tabela 3 mostra os resultados dos testes:

Hill Climbing	0.7725	0.0144
0.8829	**Simulated Annealing**	0.0022
0.0101	0.0028	**Genetic**

Table 3: Gráfico dos testes na tabela pareada da base *Digits*.

Na Tabela 3, a representação dos classificadores que tiveram a hipótese nula rejeitada são os classificadores que tiveram o resultado abaixo de 5%, em outras palavras, onde há uma diferença significativa e assim, os demais classificadores aceitam a hipótese.

Os testes entre os algorítimos de *Hill Climbing* com *Genetic* e o *Simulated Annealing* com *Genetic* mostram uma diferença significativa, tendo valores abaixo de 1% e os demais mostraram grandes similaridades, com resultados acima de 75%.

Portanto, a utilização de algoritmos de metaheurística mostra-se importante em bases grandes para potencializar a busca da melhor combinação dos dados. Entretanto, deve-se utilizar técnicas mais rápidas para efetuar essa busca, pois todos os algoritmos testados, tiveram resultados muito similares, mas tendo o algoritmo de *Hill Climbing* com menos de 20 minutos tendo um dos melhores resultados. Além disso, uma das técnicas utilizadas no artigo anterior e que teve o mesmo resultados das metaheurísticas foi o *Random Forest*, utilizando menos de 3 minutos para mostrar resultados excelentes. Com isso, mostra que as metaheurísticas devem ser testadas ao serem utilizadas em bases que tem bons resultados com classificadores comuns, pois tendem a terem o mesmos resultados ou com pouca melhoria.

3.2. Wine

A base *Wine* é menor comparado ao *Digits*, contendo apenas 3 classes e cerca de 180 linhas com 13 colunas. Dessa forma, os classificadores tendem a ter resultados mais precisos pelo menor número de classe, mas menos generalizado pela quantidade de linhas. Os resultados dos classificadores utilizados estão na tabela a seguir:

Técnica	**Acurácia**	**Desvio padrão**	**Lim. inf.**	**Lim. sup.**
Hill Climbing	0.9812	0.0366	0.9681	0.9943
Simulated Annealing	0.9700	0.0452	0.9538	0.9862
Genetic	0.9716	0.0484	0.9543	0.989
Het. Pooling	0,9660	0,0541	0,9466	0,9853
RandomForest	0,9831	0,0385	0,9693	0,9969

Table 4: Resultados das técnicas na base *Wine*

Como na base *Wine*, as metaheurísticas e o *Heterogeneous Pooling* tiveram ótimos resultados em torno de 97%. Entretanto, o algoritmo de *Hill Climbing* mostrou o melhor resultado entre eles, surpreendendo no limite superior de quase 100% com poucos minutos de execução.

Dessa forma, como o algoritmo de *Hill Climbing* implementado no artigo é a versão determinística e a base *Wine* é bem menor em comparação as outras bases, então o tempo de execução é bem menor e torna-se mais fácil testar todas as combinações possíveis de classificadores, encontrando a melhor combinação e tendo o melhor resultado.

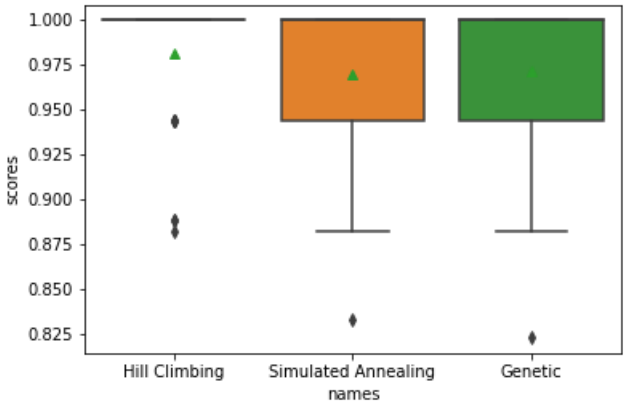

Figure 2: Gráfico *Boxplot* a partir dos resultados das técnicas na base *Wine*.

O gráfico de *Boxplot* mostra a diferenciação entre o algoritmo de *Hill Climbing* entre os demais, mostrando o poder do algoritmo em pouco tempo. Entretanto, houve alguns *outliers* que não obtiveram resultados melhores que os demais classificadores.

Hill Climbing	0.1685	0.0845
0.0616	**Simulated Annealing**	0.8836
0.0953	0.7995	**Genetic**

Table 5: Gráfico dos testes na tabela pareada da base *Wine*.

De acordo com a Tabela 5, nenhum teste apresentou grandes diferenças que resultaram em porcentagens menores que 5%, mostrando que as metaheurísticas estão encontrando as melhores combinações entre os classificadores.

Novamente, o gráfico *Boxplot* e as tabelas apresentadas nesse tópico mostram

que não é recomendado utilizar algoritmos que utilizam muito tempo para encontrar um resultado, pois o algoritmo *Hill Climbing* mostrou um ótimo desempenho, chegando quase a 100% de acerto com pouco tempo. Além disso, deve-se considerar a necessidade da utilização das metaheurísticas em conjunto a classificadores, pois os resultados teve pouco ganho em comparação aos classificadores que não utilizaram a técnica.

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 classificadores tendem a ter um excelente desempenho, pois contem um bom número de linhas e colunas e o trabalho mais fácil de classificar apenas duas classes. Dessa forma, os classificadores podem ter resultados semelhantes. Os resultados dos classificadores utilizados estão na tabela a seguir:

Técnica	**Acurácia**	**Desvio padrão**	**Lim inf.**	**Lim. sup.**
Hill Climbing	0.9525	0.0272	0.9428	0.9623
Simulated Annealing	0.9537	0.0237	0.9452	0,9762
Genetic	0,9583	0,0233	0,9500	0.9622
Het. Pooling	0,9548	0,0238	0,9463	0,9634
AdaBoost	0,9677	0,0235	0,9593	0,9762

Table 6: Resultados das técnicas na base *Breast cancer*

Novamente, todas as técnicas tiveram resultados similares, girando em torno de 96% de acurácia e com os limites inferiores e superiores de 94% e 96% respectivamente.

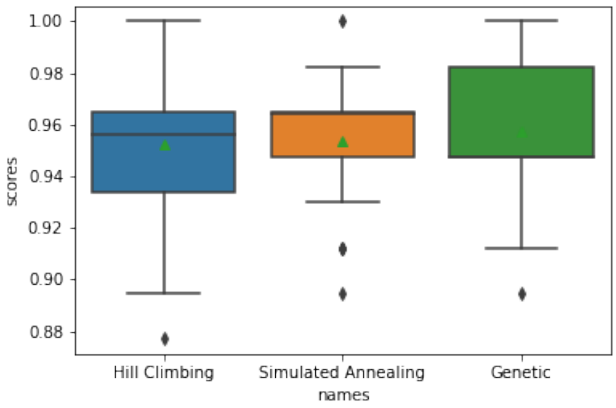

Figure 3: Gráfico *Boxplot* a partir dos resultados das técnicas na base *Breast cancer*.

Os resultados mostrados no gráfico de *Boxplot* mostram a igualdade do desempenho entre os classificadores e também o poder de classificação utilizando metaheurística.

Hill Climbing	0.9859	0.3662
0.7732	**Simulated Annealing**	0.3409
0.2445	0.3601	**Genetic**

Table 7: Gráfico dos testes na tabela pareada da base *Breast cancer*.

De acordo com a Tabela 7, os testes de *Wilcoxon* e do *T-student* mostram que todos as metaheurísticas utilizadas são parecidos e que não tem diferença significativa, tendo resultados de até 98% nos testes.

Dessa forma, não é recomendado a utilização de metaheurísticas com classificador em bases mais fáceis de classificar como a *Breast cancer*, pois tiveram os mesmos resultados e o *ensemble AdaBoost* teve o melhor desempenho em comparação as demais técnicas. Portanto, o uso de metaheurísticas podem atrasar a classificação desnecessariamente.

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 ver a diferença das metaheurísticas com os classificadores, principalmente no uso do tempo e computacional. É possível concluir que as metaheurísticas são mais eficazes em classificadores que não obteve uma taxa de acurácia aceitável, aprimorando a busca pelo melhor resultado.

Ademais, as metaheurísticas se demonstraram poderosas em aprimorar os resultados dos classificadores, 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, pois para produzir os dados desse artigo, foi necessário acima de 8 horas para cada base.

Com os testes de *Wilcoxon* e *T-test* foi possível analisar a diferença entre as metaheurísticas, logo determinando qual técnica teve o melhor desempenho em conjunto com as tabelas e os gráficos de apoio apresentados. Geralmente, as melhores metaheurísticas tiveram uma diferença significativa aos demais classificadores, determinando a sua superioridade.

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 e também de bases em que classificadores não tiveram boa efetividade para verificar a eficácia na implementação dessas técnicas. Uma outra futura contribuição, é a utilização de metaheurísticas em maquinas lineares e *Deep Learning* para a buscas de pesos.

References

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