

Comparação experimental do uso de metaheurísticas na construção de combinados de classificadores

Pedro Anselmo Santana De Angeli*

Abstract

Este artigo tem o objetivo de realizar uma comparação experimental entre um conjunto pré-definido de técnicas de aprendizado para classificação automática, baseadas na ideia do uso de metaheurísticas para construção de combinados de classificadores. As metaheurísticas escolhidas são: *Hill Climbing*, *Simulated Annealing* e *Genetic Algorithm*. As bases de dados a serem utilizadas são *digits*, *wine* e *breast cancer*.

Keywords: Classificadores, Ensembles, Problemas de Classificação, Metaheurísticas

1. Introdução

Um problema de classificação pode ser definido pela busca de um modelo capaz de receber como entrada um caso a ser avaliado e indicar qual classe ele pertence. São problemas de aprendizagem supervisionada, no qual é um utilizado um conjunto de dados onde as classes são conhecidas para construir o modelo. Esse modelo, uma vez construído, poderá ser aplicado a novos dados para prever suas respectivas classes. Um *ensemble*, ou combinado de classificadores, consiste em um conjunto de classificadores treinados individualmente cujas decisões são combinadas de alguma forma. Uma metaheurística é um método para resolver de forma genérica problemas de otimização. São geralmente aplicadas a problemas para os quais não se conhece algoritmo eficiente.

*Aluno do curso de Engenharia de Computação da Universidade Federal do Espírito Santo.

Este artigo tem o objetivo de realizar uma comparação experimental entre três metaheurísticas na construção de combinados de classificadores: *Hill Climbing*, *Simulated Annealing* e *Genetic Algorithm*. Todas as três buscas foram re-
15 alizadas com o objetivo de encontrar a melhor combinação de classificadores do *ensemble Heterogeneous Pooling*¹ Todos os combinados de classificadores foram treinados nas bases de dados *digits*, *wine* e *breast cancer*, que são disponibilizadas pela biblioteca *scikit-learn*. A métrica utilizada para comparação foi a acurácia dos resultados obtidos. Para cada base, o procedimento experimental
20 de treinamento foi realizado através de 3 rodadas de validação cruzada estratificada de ciclos aninhados de validação e teste, com o ciclo interno de validação contendo 4 folds e o externo de teste com 10 folds. Antes de cada treinamento, os dados são padronizados utilizando a normalização com *z-score*.

2. Descrição dos métodos implementados

25 Foi implementado o *Heterogeneous Pooling classifier* e as metaheurísticas *Hill Climbing*, *Simulated Annealing* e *Genetic Algorithm*. Nas três buscas, o estado do problema é definido como o combinado de classificadores atual. O tempo limite de execução de cada busca é 2 minutos. Além disso, foi criada uma classe *Heterogeneous Pooling* para definir todos os parâmetros necessários
30 para o treinamento, com o objetivo de modularizar e facilitar a compreensão do código.

2.1. *Heterogeneous Pooling*

Construtor: O construtor de cada classe define todos os parâmetros necessários para o treinamento. O modelo é instanciado, assim como o padronizador.
35 Também são definidos os hiperparâmetros que serão testados, o *grid search*, a estratégia de validação cruzada e a metaheurística a ser utilizada. No caso de não especificação da metaheurística, o *Heterogeneous Pooling classifier* é

¹Combinado de classificadores composto por *Nearest Neighbors*, *Gaussian Naive Bayes* e *Decision tree*

treinado em sua forma original. Caso contrário, uma busca é feita no combinado de classificadores.

40 ***fit()***: Método utilizado para treinamento. É responsável pela chamada da função do *scikit-learn* que fará o treinamento passando os parâmetros definidos no construtor.

results(): Método que retorna a média, o desvio padrão e o intervalo de confiança dos resultados obtidos no treinamento.

45 ***getScores()***: Retorna as acurácias obtidas em cada fold de teste.

2.2. *Heterogeneous Pooling classifier*

O *Heterogeneous Pooling classifier* é um combinado de classificadores composto por *Nearest Neighbors*, *Gaussian Naive Bayes* e *Decision tree*. Os três classificadores foram obtidos pela biblioteca *scikit-learn*. O *ensemble* recebe um
50 atributo chamado *n_samples*, que representa a quantidade individual de cada um dos três classificadores. Se *n_samples* = 2, por exemplo, o combinado será composto por 6 classificadores: 2 *Nearest Neighbors*, 2 *Gaussian Naive Bayes* e 2 *Decision tree*. Os métodos *fit()* e *predict()* descritos abaixo detalham o funcionamento do combinado.

55 ***fit()***: Primeiramente, são obtidas as frequências de cada classe na base de treino. Em seguida, cada um dos classificadores é treinado com uma base de treino de mesmo tamanho da original montada coletando aleatoriamente exemplos da base original com reposição. Se for especificada a metaheurística, uma busca é feita no *ensemble* original e um novo combinado é gerado.

60 ***predict()***: Cada exemplo é classificado por todos os classificadores. O resultado é selecionado utilizando votação majoritária. Se houver empate na votação, a classe mais frequente na base de treino é escolhida.

2.3. *Hill Climbing*

O *Hill Climbing* foi implementado em sua forma determinística. Inicial-
65 mente, o estado ótimo é um combinado vazio. Então, seus estados vizinhos

são gerados, ou seja, é adicionado um classificador no estado atual. Os classificadores adicionados são os treinados inicialmente no *Heterogeneous Pooling classifier*. Se nenhum vizinho melhorar a acurácia do estado ótimo, a busca acaba e o melhor *ensemble* é retornado. Caso contrário, o vizinho que obteve
70 melhor acurácia vira o estado atual e a busca continua.

2.4. Simulated Annealing

O *Simulated Annealing* foi implementado de forma não determinística. O estado inicial é gerado aleatoriamente a partir do combinado original do *Heterogeneous Pooling classifier*. A temperatura inicial é igual a 200, o fator de redução
75 é 0.1 e o número de iterações é igual a 10. A temperatura é reduzida de forma geométrica, ou seja: $nova_temperatura = temperatura_atual \cdot fator_de_reducao$. A partir do estado atual, são gerados os seus vizinhos, ou seja, é removido ou adicionado um classificador ao estado atual. Os classificadores adicionados são os treinados inicialmente no *Heterogeneous Pooling classifier*. Então, um vizinho é escolhido aleatoriamente. O estado atual vira esse vizinho com probabilidade igual a 100% se a acurácia do vizinho for maior que o estado atual ou $e^{-(\Delta A)/t} \cdot 100\%$ caso contrário, onde ΔA é a diferença de acurácia do estado atual e o vizinho escolhido e t é a temperatura atual. Esse procedimento de mudança de estado é executado 10 vezes. A busca se encerra quando a tem-
80 peratura fica menor do que 1 e o *ensemble* com maior acurácia encontrado é retornado.

2.5. Genetic Algorithm

O *Genetic Algorithm* foi executado de forma não determinística. As razões de *Crossover* e mutação, a porcentagem de elitismo, o tamanho da população
90 e número de gerações são respectivamente: 0.9, 0.1, 20%, 10 e 20. Uma população inicial é gerada. Cada indivíduo dessa população é gerado aleatoriamente a partir do combinado original do *Heterogeneous Pooling classifier*. Então, pra cada geração o algoritmo executa o procedimento a seguir. Cada indivíduo é avaliado, ou seja, é calculada sua acurácia. Então, a partir da porcentagem de

95 elitismo, os indivíduos mais bem avaliados são selecionados para fazerem parte da nova população. Então, os indivíduos da população atual passam por um processo de seleção, cruzamento e mutação. A nova população é composta pelos indivíduos selecionados pelo elitismo e pelos processos citados. O processo de seleção é realizado de maneira aleatória. Quanto maior a acurácia do indivíduo, maior a probabilidade dele ser selecionado. O *Crossover* também é
100 a realizado aleatoriamente. São escolhidos dois pais e o cruzamento gera dois filhos que contém classificadores selecionados aleatoriamente de cada um dos pais. A mutação consiste na remoção de um classificador do indivíduo. O melhor indivíduo encontrado ao longo das gerações é retornado.

105 3. Descrição dos experimentos realizados

Antes de cada treinamento, os dados são padronizados utilizando a normalização com *z-score*. O procedimento experimental consiste em validação e teste através de 3 rodadas de validação cruzada estratificada de ciclos aninhados, com o ciclo interno de validação contendo 4 folds e o externo de teste com 10
110 folds. A etapa de validação é feita pelo *grid search* considerando os seguintes hiperparâmetros:

Classificador	Hiperparâmetro
HeterogeneousPooling	n_samples = [3, 5, 7]

Table 1: Hiperparâmetros do ensemble.

3.1. Digits

A base de dados *digits* possui 1797 amostras, 64 atributos e 10 classes. Cada classe possui aproximadamente 180 amostras. Os resultados obtidos após a
115 execução do procedimento experimental são exibidos na Tabela 2. O gráfico boxplot obtido para o experimento é exibido na Figura 1.

Método	Média	Desvio Padrão	Limite Inferior	Limite Superior
Sem metaheurística	0.967	0.013	0.92	0.971
Hill Climbing	0.968	0.012	0.963	0.972
Simulated Annealing	0.965	0.019	0.958	0.971
Genetic	0.960	0.015	0.954	0.965
RandomForest	0.976	0.009	0.973	0.979

Table 2: Resultados obtidos na base digits.

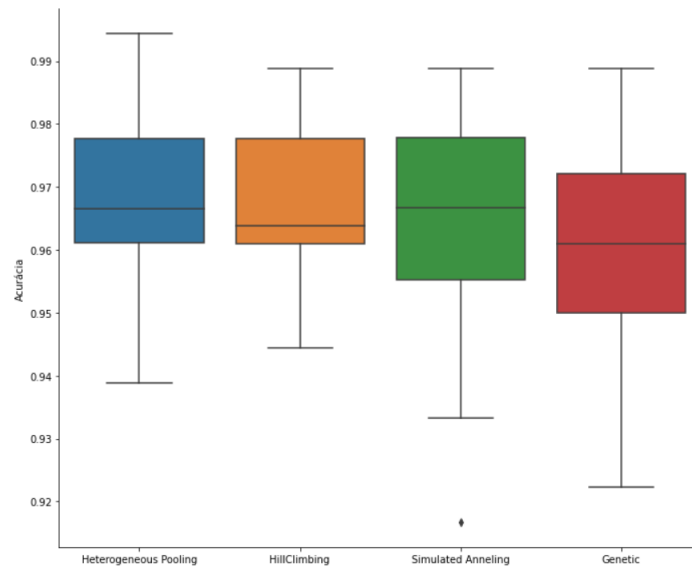


Figure 1: Boxplot obtido para a base digits.

Da tabela e do boxplot, nota-se que os métodos apresentaram comportamento muito semelhantes em termos de acurácia, tendo o *Hill Climbing* apresentado melhor acurácia e menor desvio padrão. Também foi acrescentado o *ensemble Random Forest*, pois foi o que obteve melhor resultado nos experimentos realizados no primeiro trabalho na base *digits*. Porém, essa avaliação não é suficiente para avaliar se um classificador é melhor do que o outro. Para isso, foi utilizado testes estatísticos. A tabela 3 mostra na matriz triangular superior

os resultados do teste t pareado e na matriz triangular inferior os resultados do
 125 teste não paramétrico de wilcoxon.

Hill Climbing	0.334	0.007
0.353	Simulated Annealing	0.155
0.012	0.174	Genetic

Table 3: Testes de hipóteses na base digits.

As células em negrito representam os testes em que a hipótese nula foi re-
 jeitada. Isso quer dizer que há diferença significativa entre os classificadores.
 Os demais aceitam a hipótese nula, logo não podemos concluir qual apresenta
 um melhor desempenho.

130 3.2. Wine

A base de dados *wine* possui 178 amostras, 13 atributos e 3 classes. Cada
 classe possui 59, 71 e 48 amostras. Os resultados obtidos após a execução do
 procedimento experimental são exibidos na Tabela 4.

Método	Média	Desvio Padrão	Limite Inferior	Limite Superior
Sem metaheurística	0.980	0.040	0.965	0.994
Hill Climbing	0.978	0.037	0.964	0.991
Simulated Annealing	0.961	0.060	0.939	0.982
Genetic	0.974	0.053	0.955	0.993
RandomForest	0.975	0.041	0.961	0.990

Table 4: Resultados obtidos na base wine.

O gráfico boxplot obtido para o experimento é exibido na Figura 2.

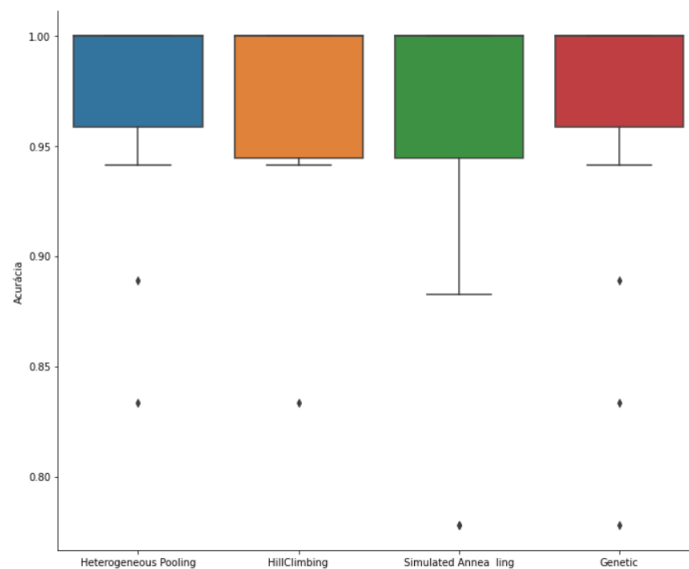


Figure 2: Boxplot obtido para a base wine.

Assim como na base *digits*, temos um desempenho muito parecido entre os experimentos. Houve uma pequena melhora em termos de acurácia em relação ao *Random Forest*, que foi o classificador que obteve melhor resultado nos experimentos realizados no primeiro trabalho na base *wine*. A tabela 5 mostra na matriz triangular superior os resultados do teste t pareado e na matriz triangular inferior os resultados do teste não paramétrico de wilcoxon.

Hill Climbing	0.025	0.586
0.018	Simulated Annealing	0.067
0.854	0.080	Genetic

Table 5: Testes de hipóteses na base wine.

3.3. Breast Cancer

A base de dados *breast cancer* possui 569 amostras, 30 atributos e 2 classes. Cada classe possui 212 e 357 amostras. Os resultados obtidos após a execução do procedimento experimental são exibidos na Tabela 6.

Método	Média	Desvio Padrão	Limite Inferior	Limite Superior
Sem metaheurística	0.953	0.021	0.945	0.961
Hill Climbing	0.941	0.027	0.932	0.951
Simulated Annealing	0.953	0.024	0.944	0.961
Genetic	0.955	0.022	0.947	0.963
AdaBoost	0.968	0.024	0.959	0.976

Table 6: Resultados obitdos na base breast cancer.

145 O gráfico boxplot obtido para o experimento é exibido na Figura 3.

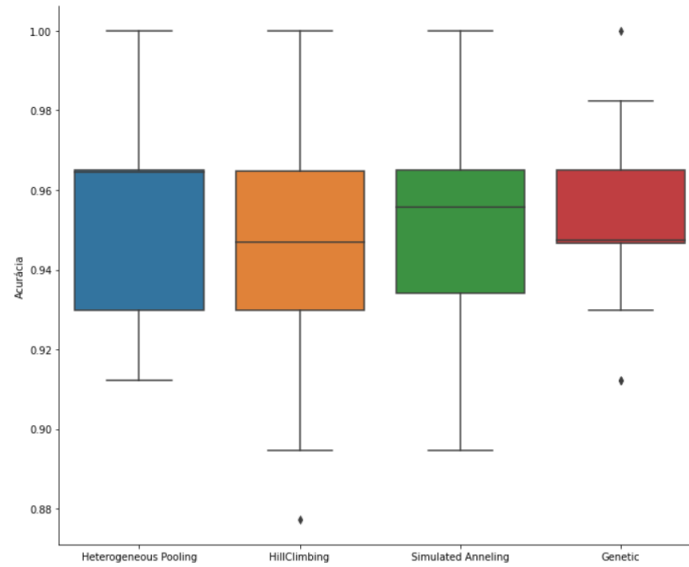


Figure 3: Boxplot obtido para a base breast cancer.

Os resultados pioraram em relação *Adaboost*, que foi o classificador que obteve melhor resultado nos experimentos realizados no primeiro trabalho na

base *breast cancer*. A tabela 7 mostra na matriz triangular superior os resultados do teste t pareado e na matriz triangular inferior os resultados do teste não paramétrico de wilcoxon.

Hill Climbing	0.014	0.015
0.010	Simulated Annealing	0.504
0.018	0.985	Genetic

Table 7: Testes de hipóteses na base breast cancer.

4. Conclusões

4.1. Análise geral dos resultados

De maneira geral, os modelos se comportaram de forma semelhante em todas as bases de dados. Com exceção da base *breast cancer*, a maioria dos testes aceitaram a hipótese nula, logo não podemos concluir qual apresenta melhor desempenho. As metaheurísticas não melhoraram significativamente o desempenho dos classificadores pois o espaço de busca utilizado e o tempo de cada busca foi muito pequeno.

4.2. Contribuições do Trabalho

Este trabalho contribuiu para um maior entendimento de técnicas de aprendizagem de máquina aplicadas a problemas de classificação e no uso de metaheurísticas, assim como no entendimento de suas comparações utilizando medidas, gráficos e testes estatísticos.

4.3. Melhorias e trabalhos futuros

Possíveis trabalhos futuros seriam modificar os hiperparâmetros das metaheurísticas para aumentar o espaço de busca, com o objetivo de melhorar o desempenho dos classificadores.

References

References

- 170 [1] *Digits, Wine e Breast Cancer* dataset
https://scikit-learn.org/stable/datasets/toy_dataset.html
- [2] Classificadores e *Ensembles*
<https://scikit-learn.org/stable/modules/classes.html>
- [3] Testes de hipóteses
175 <https://docs.scipy.org/doc/scipy/reference/stats.html>
- [4] Notas de aula do professor