Leonardo Santos Paulucio

Tarefa Computacional 1

Vitória - ES

15 de Outubro de 2019

1 Introdução

Mineração de dados ou *Data Mining*, em inglês, é um conjunto de ferramentas e técnicas utilizadas para extrair informações ou padrões de dados brutos. Isso permite que sejam aplicadas em uma grande variedade de processos como: classificação, tomada de decisão, predição, entre outros. Além disso, essas técnicas possuem a vantagem de serem aplicadas em uma quantidade muito grande de dados de forma eficiente.

A tarefa de classificação consiste em associar objetos a determinadas classes automaticamente. Para determinar a classe de cada objeto diversas características dos mesmos são analisadas, as chamadas *features*. A detecção de spam em serviços de email pode ser considerada uma tarefa de classificação, onde, nesse caso, existem apenas duas classes: spam ou não spam. Outro exemplo muito conhecido é o de classificação da flor íris, em que são analisadas informações de comprimento e largura de suas sépalas e pétalas. Após a análise, a flor é classificada uma de suas três especies: *iris setosa*, *iris virginica* e *iris versicolor*.

Em alguns casos, onde existem muitas *features* para serem analisadas, a tarefa de classificação acaba se tornando muito complexa e custosa. Esse problema, conhecido em inglês pelo termo *curse of dimensionality*, se deve ao fato de que cada *feature* representa uma dimensão no espaço solução. Assim, a medida que uma nova *feature* é adicionada seu espaço solução cresce exponencialmente. Visando alternativas para contornar esse problema diversas pesquisas têm sido desenvolvidas nos últimos anos.

Feature Selection consiste na tarefa de remover *features* redundantes ou irrelevantes que não agreguem informações aos objetos analisados, de forma que pouca ou quase nenhuma informação seja perdida. Dessa forma, essas técnicas permitem que seja possível: realizar uma simplificação do modelo, diminuir o tempo de treinamento, melhorar a performance do modelo, reduzir a chance de que ocorra *overfitting*. Apesar das vantagens citadas, essa tarefa não é simples de ser feita. Encontrar o melhor conjunto de *features* é uma tarefa difícil. Através de algoritmos de força bruta é possível produzir todos os subgrupos de *features*. Porém, essa abordagem tem um alto custo computacional e se torna impraticável para grandes *datasets*.

Nessa tarefa será feita uma classificação de dados utilizando *feature selection*. Para isso, serão utilizados: redes neurais *feedforward* (classificação) e um algoritmo genético binário (*feature selection*). Os resultados obtidos serão comparados com os obtidos por Hegazy, Makhlouf e El-Tawel (2018).

2 Arquitetura

A arquitetura utilizada é simples e formada por duas partes: o algoritmo genético binário e uma rede neural *feedforward*.

O algoritmo genético binário será o responsável por realizar a seleção das *features*. Nesse caso, cada indivíduo possuirá um cromossomo de tamanho N, onde N corresponde ao total de *features* existentes no *dataset*. A escolha ou não de uma determinada *feature* será determinada pelo valor do cromossomo: se 1 aquela *feature* é selecionada, se 0 a *feature* não é selecionada. A *fitness* para avaliação do indivíduo será calculada por uma função que possui um balanço entre o número de *features* selecionadas e a acurácia do classificador. A função de *fitness* é dada pela seguinte equação:

$$fitness = \rho Erro(C) + \varphi \frac{|F|}{|T|}$$

Onde ρ e φ são constantes utilizadas para controlar a acurácia e seleção de *features*, e o $Erro(C)$ é o erro do classificador, F e T correspondem ao número de *feature* selecionados e o total existente no *dataset* respectivamente.

O classificador formado pela rede neural *feedforward*, irá receber as *features* selecionadas por cada indivíduo e será treinada utilizando *backpropagation* e ELM (*Extreme Learning Machine*). O valor da acurácia obtido pelo classificador será utilizado para calcular o erro que será enviado ao GA para a avaliação do indivíduo. A Figura 1 apresenta um esquema simplificado da arquitetura utilizada.

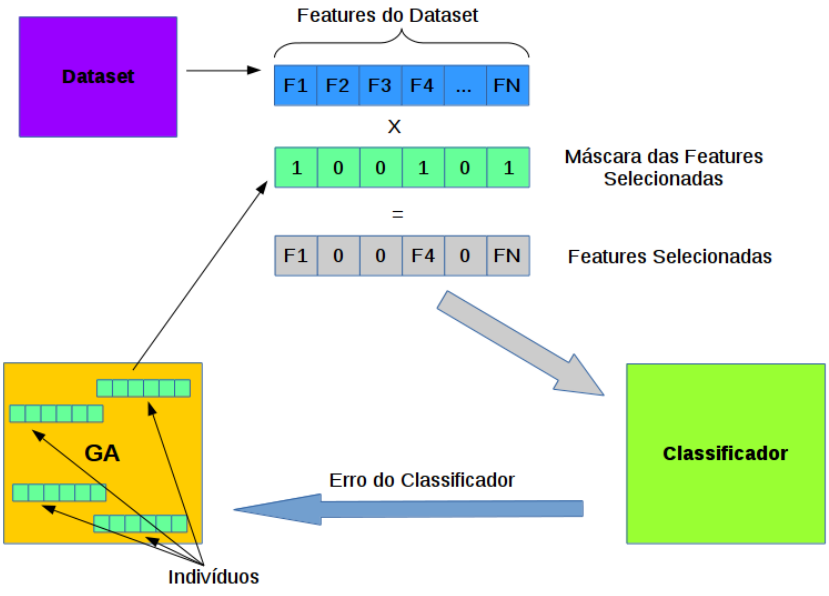


Figura 1 – Arquitetura simplificada utilizada

3 Resultados e Discussão

Com o objetivo de comparar os resultados com os obtidos por Hegazy, Makhlouf e El-Tawel (2018), foram utilizados os mesmos valores nas constantes da função de *fitness* e no número de execuções e épocas. A Tabela 1 apresenta os valores utilizados.

Parâmetro	**Valor**	**Significado**
ρ	0.9	Constante da função *Fitness*
φ	0.1	Constante da função *Fitness*
NumGeraçoes	50	Número de gerações
PopSize	10	Tamanho da População
NumRuns	20	Número de execuções
CrossProb	0.9	Prob. de Crossover
MutProb	0.1	Prob. de Mutação
NumEpochs	50	Num. épocas de treino do classificador
LR	0.001	*Learning rate* do classificador
HidSize	$\sqrt{NumFeatures * NumClasses}$	Tam. camada oculta do classificador

Tabela 1 – Parâmetros utilizados

Para analisar a escalabilidade do método foram escolhidos *datasets* com diferentes quantidades de *features*. Os *datasets* selecionados foram: Wine, Ionosphere e Arrhythmia. A Tabela 2 apresenta com mais detalhers cada um.

Dataset	**Features**	**Amostras**	**Classes**	**Balanceado**
Wine	13	178	3	Sim
Ionosphere	34	351	2	Sim
Arrhythmia	279	452	16	Não

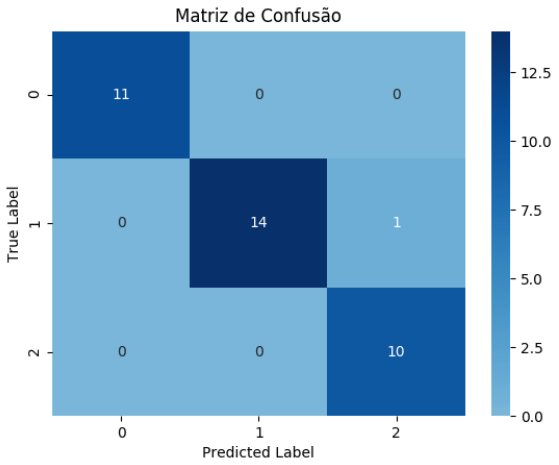
Tabela 2 – Detalhes dos datasets utilizados

3.1 *Baseline*

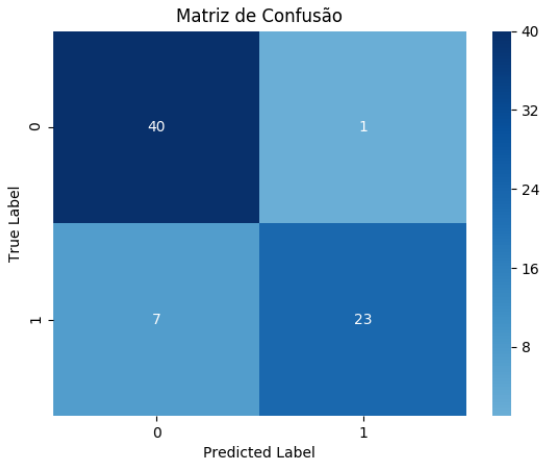
Para o *baseline* todas as *features* foram utilizadas como entrada em um classificador simples. Os resultados obtidos estão ilustrados na Tabela 3. E as matrizes de confusão obtidas estão ilustradas na Figura 2.

Dataset	**Acurácia**
Wine	0.9722
Ionosphere	0.8873
Arrhythmia	0.6593

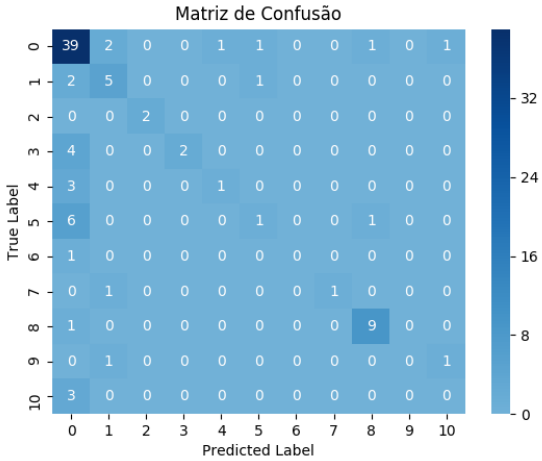
Tabela 3 – Resultados da acurácia média obtida no baseline



(a) Wine



(b) Ionosphere



(c) Arrhythmia

Figura 2 – Matrizes de confusão obtidas no baseline

3.2 Experimento

As tabelas de 4-9 apresentam os resultados obtidos juntamente com os do artigo original para que seja possível realizar uma comparação dos mesmos.

Dataset	**GA+ELM**	**GA+NN**	**ISSA**
Wine	0.0230	0.2747	**0.0000**
Ionosphere	0.5373	0.1192	**0.0770**
Arrhythmia	0.3136	0.2978	**0.0310**

Tabela 4 – Melhor fitness obtidas nas 20 execuções

Dataset	**GA+ELM**	**GA+NN**	**ISSA**
Wine	0.0412	0.1464	**0.0100**
Ionosphere	0.5475	0.1403	**0.0970**
Arrhythmia	0.3398	0.2978	**0.0440**

Tabela 5 – Média das fitness obtidas nas 20 execuções

Dataset	**GA+ELM**	**GA+NN**	**ISSA**
Wine	0.0634	**0.1942**	0.0340
Ionosphere	**0.5550**	0.1602	0.1450
Arrhythmia	**0.3564**	0.3154	0.0510

Tabela 6 – Pior fitness obtidas nas 20 execuções

Dataset	**GA+ELM**	**GA+NN**	**ISSA**
Wine	4.7	6.2	**4.5**
Ionosphere	**9.45**	15.4	11.07
Arrhythmia	136.7	134.05	**95.15**

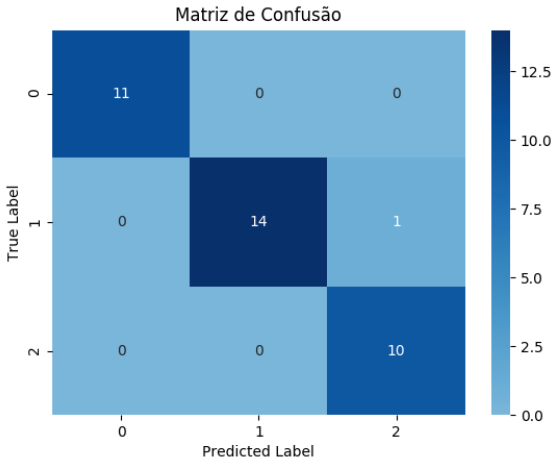
Tabela 7 – Média do número de features selecionadas nas 20 execuções

Dataset	**GA+ELM**	**GA+NN**	**ISSA**	**Baseline**
Wine	**0.9945**	0.9251	0.9780	0.9722
Ionosphere	0.4225	**0.9295**	0.8530	0.8873
Arrhythmia	0.7032	**0.7472**	0.6600	0.6593

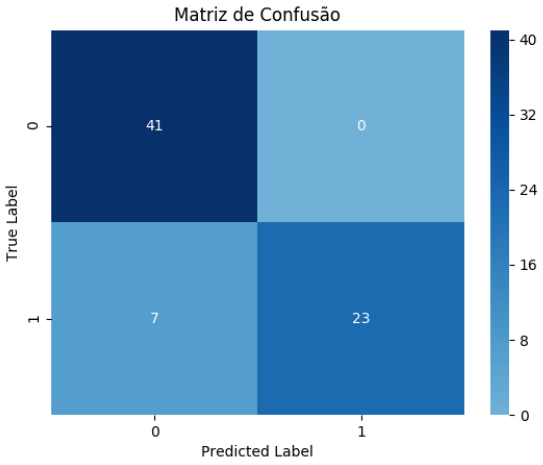
Tabela 8 – Média da acurácia obtida nas 20 execuções

Dataset	**GA+ELM**	**GA+NN**	**ISSA**
Wine	**0.6203**	13.4987	21.13
Ionosphere	**0.6104**	23.3341	32.48
Arrhythmia	**9.8050**	101.9321	122.19

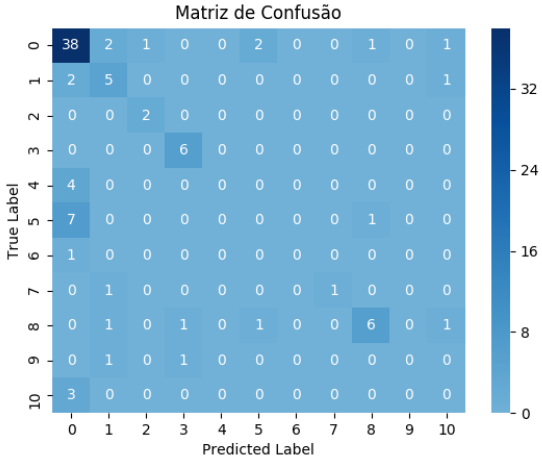
Tabela 9 – Média do tempo de execução nas 20 execuções



(a) Wine utilizando ELM



(b) Ionosphere utilizando NN



(c) Arrhythmia utilizando NN

Figura 3 – Matrizes de confusão obtidas para as melhores acurácias

4 Discussões

A Tabela 4 mostra que apesar do ISSA obter as melhores *fitness* para os três *datasets*, os valores acabam sendo próximos dos obtidos com o novo modelo, isso fica mais evidente na Tabela 5, onde a média das *fitness* obtidas nas 20 execuções são apresentadas. Nota-se, ainda, que o modelo treinado usando ELM fica melhor do que o com *backpropagation* somente para o dataset Wine.

O número médio de *features* selecionadas também é próximo do obtido pelo ISSA, como apresentado na Tabela 7. Vale destacar que o treinamento com ELM conseguiu obter uma quantidade média de *features* selecionadas menor do que a obtida pelo ISSA.

A Tabela 8 compara os resultados de acurácia obtidos nos experimentos com o ISSA e com o *baseline* realizado. Analisando-se a tabela percebe-se que os resultados obtidos conseguiram ser melhores do que o ISSA. Para os *datasets Ionosphere* e *Arrhythmia*, o modelo utilizando GA+NN foi o melhor. Já para o *Wine* o GA+ELM obteve melhor resultado.

Ao se comparar as matrizes de confusão para as melhores acurácias, ilustrada na Figura 3, com as obtidas do *baseline* (Figura 2), percebe-se que elas são bem parecidas. Isso mostra que as *features* selecionadas conseguem representar de forma eficiente e sem perda de muita informação todo o conjunto de dados do *dataset*, visto que o número reduzido de features não compromete muito o resultado do classificador.

5 Conclusão

Nesse trabalho foi desenvolvida uma forma de se realizar *feature selection* utilizando um algoritmo genético binário. O método foi analisado em *datasets* de diferentes tamanhos e comparado com o método ISSA desenvolvido por Hegazy, Makhlouf e El-Tawel (2018).

O modelo obteve bons resultados, conseguindo superar o ISSA em algumas métricas, indicando que seleção de *features* funciona e, pode ser utilizada para simplificar a representação dos dados, tornando o modelo mais simples e mais rápido de ser treinado.

Para tentar melhorar o modelo desenvolvido, poderiam ser feitas mudanças nos parâmetros utilizados, como por exemplo: aumentar o número de épocas no treinamento do classificador utilizando *backpropagation*, alterar os valores das probabilidades de mutação e crossover no algoritmo genético.

Referências

HEGAZY, A. E.; MAKHLOUF, M.; EL-TAWEL, G. S. Improved salp swarm algorithm for feature selection. *Journal of King Saud University - Computer and Information Sciences*, 2018. ISSN 1319-1578. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1319157818303288>. Citado 3 vezes nas páginas 1, 3 e 8.