Primeiro Trabalho de Inteligência Artificial

Rebeca Cecco De Oliveira

Abstract

Neste trabalho serão apresentados e analisados os resultados da aplicação de várias técnicas de classificação na base de dados *Digits* da biblioteca Sklearn. As técnicas usadas foram *ZeroR*, *Naive Bayes Gaussiano*, *K Vizinhos Mais Próximos* e *Árvore de Decisão* implementadas pela biblioteca já mencionada. Além dos metódos já citados também foi implementado o metódo K-Means Centroides (KMC).

1. Introdução

Problemas de *Classificação* são problemas os quais existe um conjunto de dados e é necessário agrupar esses dados em categorias de acordo com as características observadas. No campo de *Aprendizado de Máquina* problemas de classificação são considerados exemplos de aprendizagem supervisionada, ou seja, existe um conjunto de treinamento o qual observações corretas estão disponíveis. Outro tipo de aprendizagem seria a não supervisionada, por exemplo o problema de *Clustering* que envolve agrupar dados de acordo com a alguma medida de semelhança.

No caso de problemas de classificação temos as variáveis conhecidas, que são as características que serão observadas. As variáveis podem ser de vários tipos, como: categórica ordinal, categórica nominal ou assumir valores númericos inteiros ou reais.[1].Algoritmos que implementam a classificação são conhecidos como classificadores.

Neste trabalho foi realizada uma comparação experimental entre um conjunto de técnicas de aprendizado e classicação automática aplicadas a uma base de dados bem comportada. Os algoritmos utilizados foram *ZeroR*, *Naive Bayes Gaussiano*, *K Vizinhos Mais Próximos*, *Árvore de Decisão* e *KMeans Centroides*. A base de dados utilizada foi a *Digits*

2. Base de Dados

Para a realização do experimento foi utilizada a base de dados *Digits* fornecida pela biblioteca *Scikit Learn*.

Essa base foi gerada a partir de imagens 8x8 pixels de algarismos indo-arábicos escritos a mão por 43 pessoas voluntárias.

No total são 1797 instâncias, sendo elas dividas em 10 classes (digitos 0 a 9), cada uma tendo por volta de 180 instâncias e cada instância possuindo 64 atributos (resultante das imagens de 8x8 pixels).

3. O Método KMC

A ideia desse classificador é utilizar um algoritmo de agrupamento de modo a definir k grupos de exemplo de cada classe da base de treino.Tendo a base n classes, então inicialmente serão formados$k*n$ grupos. Então, é calculado os centróides de cada um dos grupos e este é associado a classe do grupo ao qual foi gerado.

A classificação feita é analizando qual o centróide está mais próximo de cada instância. Para implementar esse classificador foi utilizado o *KMeans*, fornecido pela biblioteca *Scikit Learn* [2], como algoritmo de agrupamento e foram utilizados os valores padrões para os hiperparâmetros

4. Experimentos Realizados

O experimento foi separado em duas etapas já que existem classificadores que possuem hiperparâmetros e os que não possuem.

A primeira etapa consiste no treino e teste com 3 rodadas de validação cruzada estratificada de 10 folds dos classificadores *ZeroR* e *Naive Bayes Gaussiano*.

A segunda etapa procedimento experimental consiste no treinamento, validação e teste atráves de 3 rodadas de ciclos aninhados de validação e teste, com o ciclo interno de validação contendo 4 *folds* e o externo de teste com 10 *folds* dos classificadores *KNeighbors*, *DecisionTree* e *KMeansCentroids*.

A busca em grade do ciclo interno considerou os seguintes valores de hiperparâmetros de cada técnica de aprendizado

- KMeans Centroides: k: 1, 3, 5, 7

- K Vizinhos Mais Próximos: n_neighbors : 1, 3, 5, 7
- Árvore de Decisão: max_depth: None, 3, 5, 10

Os experimentos foram realizados no ambiente Google Colaboratory hospedado numa máquina virtual 12GB de RAM e uma CPU de dois núcleos [3].

A seguir serão apresentados os resultados obitidos e métricas estatísticas do experimento que foi descrito

O tempo computacional discorrido em segundos para ZeroR, GaussianNB, DecisionTree, KNeighbors e KMeansCentroids respectivamente foi 0.079, 0.121, 3.170, 7.103, 257.735

Método	**Média**	**Desvio Padrão**	**Limite Inferior**	**Limite Superior**
ZeroR	3,333e-01	1,129e-16	3,333e-01	3,333e-01
GaussianNB	9.422e-01	6,944e-02	9.162e-01	9,16291e-01
DecisionTree	9,466e-01	4,429e-02	9,301e-01	9,301e-01
KNeighbors	9,511e-01	5,232e-02	9,315e-01	9,315e-01
KMeans Centroid	9,488e-01	5,448e-02	9,285e-01	9,285e-01

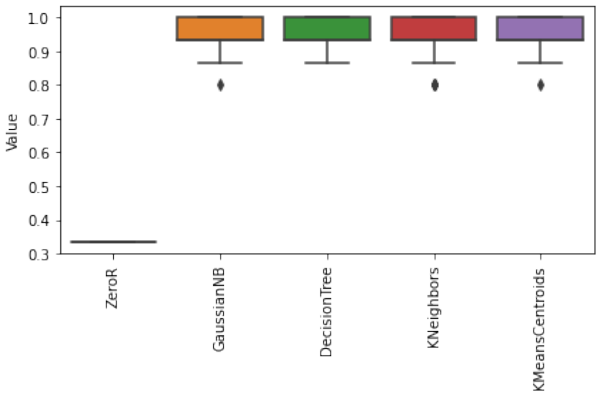
Table 1: Métricas estatísicas dos resultados experimentais

ZeroR	6,614e-33	6,649e-35	3,413e-29	2,342e-32
1,074e-06	GaussianNB	6,014e-01	3,800e-01	8,453e-01
9,877e-07	5,929e-01	DecisionTree	6,623e-01	8,229e-01
1,150e-06	3,604e-01	6,353e-01	KNeighbors	4,479e-01
1,128e-06	7,824e-01	8,509e-01	4,385e-01	KMeans Centroids

Table 2: Resultados pareados ($p-$valores) de testes de hipótese entre os pares de métodos. Na matriz triangular superior estão situados os resultados do teste$-t$ pareado (amostras dependentes) e na matriz triangular inferior os resultado do teste não paramétrico de wilcoxon. Os valores que aceitaram a hipótese nula para um nível de significância de 95% estão destacados em negrito.

No gráfico boxplot a seguir estão apresentadas as acurácias obtidas nos experimentos para cada classificador.

Figure 1: Acurácia dos metódos



5. Conclusões

5.1. Análise dos Resultados

Analisando apenas os tempos computacionais podemos apontar dois extremos, sendo o mais rápido o ZeroR e o mais lento sendo o KMeansCentroids. Isso se deve ao fato do ZeroR ser um algoritmo extremamente simples e o KMeansCentroid ter sido implementado neste trabalho, portanto, é provável que algumas melhorias e otimizações poderiam ser feitas de forma que o tempo de execução não seja tão distoante dos demais. Apesar do ótimo resultado no quesito tempo o ZeroR é péssimo quando se trata de acurrácia, enquanto os outros métodos de classificação estão empatados com um ótimo desempenho.

5.2. Contribuições do Trabalho

Pelo empate estatístico nas acurrácias dos quatro ultimos métodos pode-se perceber que os classificadores prontos podem ser utilizados amplamente, ao menos em base de dados semelhantes a ultizada nesse experimento

5.3. Melhorias e Trabalhos futuros

Primeiramente seria interessante fazer mais experimentos em outras bases de dados, mais diversas e menos bem comportadas, para que os resultados obtidos sejam corroborados ou realizar o desempate de acurrácia.

Além disso é passivel de melhoria e otimização a implementação do algortimo KMeans Centroid ou uma versão alternativa dele utilizando outro algoritmo de agrupamento para comparação de resultados.

References

[1] F. M. Varejao, Notas de aula de inteligência artificial e sistemas inteligentes (2022).

[2] S. Learn, sklearn.cluster.kmeanss, [Online; accessed in 27 Jun. 2022].
URL `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html`

[3] G. Colaboratory, Setting up computing resources, [Online; accessed in 27 Jun. 2022].
URL `https://colab.research.google.com/github/bebi103a/bebi103a.github.io/blob/master/lessons/00/setting_up_your_computer.ipynb`