14/07/2021

Primeiro Trabalho de Inteligência Artificial e Sistemas Inteligentes

Prof. Flávio Miguel Varejão

1. *Descrição*

Este trabalho consiste em realizar uma comparação experimental entre um conjunto pré-definido de técnicas de aprendizado para classificação automática, baseadas na ideia de combinados de classificadores, aplicadas a alguns problemas de classificação. As técnicas escolhidas são: Bagging, AdaBoost, RandomForest e HeterogeneousPooling. As bases de dados a serem utilizadas são digits, wine e breast cancer, todas disponíveis em https://scikit-learn.org/stable/datasets/toy_dataset.html.

Para cada base, o procedimento experimental de treinamento, validação e teste será realizado através 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. A busca em grade (grid search) do ciclo interno deve considerar os seguintes valores de hiperparâmetros de cada técnica de aprendizado:

Bagging: [n_estimators = 10, 25, 50, 100]

AdaBoost: [n_estimators = 10, 25, 50, 100]

RandomForest: [n_estimators = 10, 25, 50, 100]

HeterogeneousPooling: [n_samples = 1, 3, 5, 7]

Os resultados de cada classificador devem ser apresentados numa tabela contendo a média das acurácias obtidas em cada fold do ciclo externo, o desvio padrão e o intervalo de confiança a 95% de significância dos resultados, e também através do boxplot dos resultados de cada classificador em cada fold.

Um exemplo de uma tabela para uma base de dados hipotética é mostrado a seguir.

Método	Média	Desvio Padrão	Limite Inferior	Limite Superior
Bagging	0.95	0.04	0.94	0.96
AdaBoost	0.92	0.01	0.91	0.93
RandomForest	0.96	0.08	0.90	0.99
HeterogeneousPooling	0.93	0.01	0.93	0.94

O método HeterogeneousPooling deve ser implementado, mas usará as implementações dos métodos KNeighbors, DecisionTree e GaussianNB do sklearn. Os métodos Bagging, AdaBoost e RandomForest também estão disponíveis no scikit-learn. As descrições dos métodos implementados no sklearn podem ser acessadas respectivamente em:

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html
https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Os dados utilizados no conjunto de treino em cada rodada de teste devem ser padronizados (normalização com z-score). Os valores de padronização obtidos nos dados de treino devem ser utilizados para padronizar os dados do respectivo conjunto de teste.

Além das tabelas e dos gráficos bloxplot, será necessário apresentar também a tabela pareada dos resultados (p-values) dos testes de hipótese entre os pares de métodos. Na matriz triangular superior devem ser apresentados os resultados do teste t pareado (amostras dependentes) e na matriz triangular inferior devem ser apresentados os resultado do teste não paramétrico de wilcoxon. Os valores da célula da tabela que rejeitarem a hipótese nula para um nível de significância de 95% devem ser escritos em negrito.

Um exemplo de uma tabela pareada para uma base de dados hipotética é mostrado a seguir.

Bagging	0.0941	**0.045**	0.096
0.032	AdaBoost	0.074	0.084
0.084	0.096	RandomForest	0.105
0.105	0.084	**0.049**	HeteroGeneous

2. *Heterogenous Pooling*

O classificador Heterogeneous Pooling é um combinado de classificadores heterogêneos que usa como classificadores base: Árvore de Decisão (DT) (DecisionTreeClassifier), Naive Bayes Gaussiano (NB) (GaussianNB) e Vizinho Mais Proximo (NN) (KNeighborsClassifier), sempre com valores default do sklearn para seus hiperparâmetros, com exceção do K do NN, que terá o valor 1. O único parâmetro do método Heterogeneous Pooling é o *n_samples*, que indica o número de vezes que os classificadores base serão usados para gerar o combinado. Por exemplo, se *n_samples* é igual a 3, o combinado será composto por 9 classificadores: 3 árvores de decisão, 3 naive bayes e 3 vizinhos mais próximos. Para diferenciar os classificadores de mesmo tipo em um combinado, o primeiro deles será treinado com a base de treino original e os demais serão treinados com uma base de treino diferente, obtida a partir da base de treino original através da função resample do sklearn. Ela pode ser acessada em:
https://scikit-learn.org/stable/modules/generated/sklearn.utils.resample.html

O critério de decisão para classificar uma instância é votação majoritária, ou seja, deve-se escolher a classe mais escolhida dentre os classificadores que compõem o combinado. Em

caso de empate, a classe escolhida deve ser a mais frequente na base de dados de treino original dentre as que empataram na votação.

O pseudo código a seguir mostra como o HP é obtido a partir de uma base de dados de treino:

- Obter e armazenar a ordenação das classes de acordo com a ocorrência nos exemplos na base de treino (ordenar decrescentemente da mais frequente para a menos frequente)
- Para cada um dos *n_samples* faça
    - Se for a primeira iteração então
        - Usar a base original para treino dos classificadores
    - Senão
        - Montar uma base de treino de mesmo tamanho da original coletando aleatoriamente exemplos da base original com reposição
    - Fim-se
    - Treinar os classificadores NN, NB, DT na base de treino corrente e incluí-los no combinado de classificadores
- Fim-para

O pseudo código seguinte mostra como o combinado HP é usado para classificar um exemplo da base de dados de teste:

- Para cada um dos classificadores individuais do combinado faça
    - Obter a classificação do exemplo usando o classificador individual e armazenar a classe selecionada
- Fim-para
- Contar quantas vezes cada classe foi selecionada e obter a(s) mais votada(s)
- Se mais de uma classe for a mais votada então
    - Retornar a classe mais votada mais frequente na base de treino
- Senão
    - Retornar a classe mais votada
- Fim-se

---

3. *Informações Complementares*

a. Use o valor 36851234 para o parâmetro random_state (`random_state=36851234`) nas chamadas a RepeatedStratifiedKFold para que os resultados sejam reproduzíveis.

b. Para que os resultados do método HeterogeneousPooling sejam reproduzíveis use o

valor 0 para o parâmetro random_state (`random_state=0`) na primeira chamada a resample. A partir daí, use o valor corrente incrementado de 1 nas chamadas sucessivas de resample (random_state = 1, random_state = 2, ...).

c. Os gráficos bloxplot requeridos no treino e no teste devem ser gerados usando função específica do pacote seaborn.

d. O apêndice deste enunciado apresenta instruções de instalação e uso do overleaf para a escrita do artigo.

4. *Artigo*

Após a realização dos experimentos, um artigo descrevendo todo o processo experimental realizado deverá ser escrito em latex usando o software overleaf. O artigo deve ser estruturado contendo os seguintes componentes:

1. Título
2. Resumo
3. Seção 1. Introdução
4. Seção 2. Descrição do Método Implementado (Heterogeneous Pooling)
5. Seção 3. Descrição dos Experimentos Realizados
    a. Seção 3.1 Digits
    b. Seção 3.2 Wine
    c. Seção 3.3 Breast Cancer
6. Seção 4. Conclusões
    a. Análise geral dos resultados
    b. Contribuições do Trabalho
    c. Melhorias e trabalhos futuros
7. Referências Bibliográficas

Na subseção de análise geral dos resultados é importante discutir, dentre outras coisas, se houve diferença estatística significativa entre quais métodos em quais bases e responder se teve um método que foi superior em mais de uma base.

5. *Condições de Entrega*

O trabalho deve ser feito individualmente e submetido pelo sistema da sala virtual até a data limite (11 de agosto de 2021).

O trabalho deve ser submetido em três arquivos: um arquivo pdf com o artigo produzido no trabalho, um arquivo ipynb com o notebook jupyter para ser carregado e executado no jupyter e todos os arquivos com código fonte em python utilizados em um arquivo zip. Tanto o arquivo pdf quanto os arquivos ipynb e zip devem possuir o mesmo nome Trab1_Nome_Sobrenome.

Note que a data limite já leva em conta um dia adicional de tolerância para o caso de

problemas de submissão via rede. Isso significa que o aluno deve submeter seu trabalho até no máximo um dia antes da data limite. Se o aluno resolver submeter o trabalho na data limite, estará fazendo isso assumindo o risco do trabalho ser cadastrado no sistema após o prazo. Em caso de recebimento do trabalho após a data limite, o trabalho não será avaliado e a nota será ZERO. Plágio ou cópia de trabalhos serão verificadas automaticamente por sistemas como o moss. Trabalhos em que se configure cópia receberão nota zero independente de quem fez ou quem copiou.

6. Requisitos da implementação

a. Modularize seu código adequadamente.

b. Crie códigos claros e organizados. Utilize um estilo de programação consistente, Comente seu código.

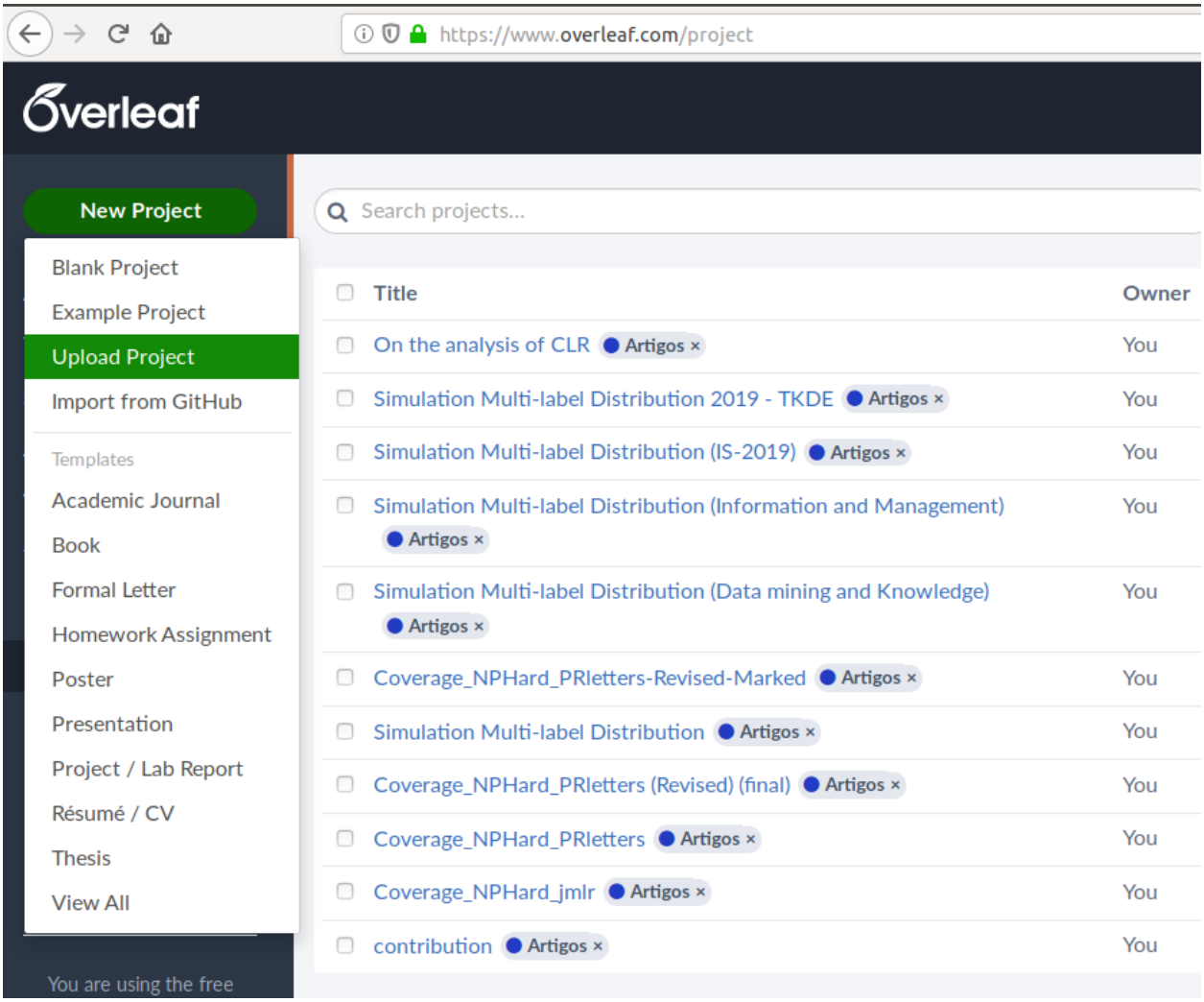
c. Os arquivos do programa devem ser lidos e gerados na mesma pasta onde se encontram os arquivos fonte do seu programa.

Observação importante

Caso haja algum erro neste documento, serão publicadas novas versões e divulgadas erratas em sala de aula. É responsabilidade do aluno manter-se informado, freqüentando as aulas ou acompanhando as novidades na página da disciplina na Internet.

Apêndice. Artigo em Latex usando Overleaf

Juntamente com este enunciado foi disponibilizado um arquivo zip com o template de latex para confecção do artigo. O primeiro passo a ser feito é criar uma conta pessoal no Overleaf (https://www.overleaf.com/register). Uma vez criada sua conta, deve-se entrar nela. Para incluir o template no overleaf, basta apenas selecionar "New Project>Upload Project" e selecionar o arquivo zip, como mostrado na figura abaixo. Não é necessário descompactar, faça o upload do zip direto. Lembrar de renomear o artigo após o upload do arquivo.


https://www.overleaf.com/project
Overleaf
New Project
Blank Project
Example Project
Upload Project
Import from GitHub
Templates
Academic Journal
Book
Formal Letter
Homework Assignment
Poster
Presentation
Project / Lab Report
Résumé / CV
Thesis
View All
You are using the free
Search projects...
Title
Owner
On the analysis of CLR Artigos × You
Simulation Multi-label Distribution 2019 - TKDE Artigos × You
Simulation Multi-label Distribution (IS-2019) Artigos × You
Simulation Multi-label Distribution (Information and Management) Artigos × You
Simulation Multi-label Distribution (Data mining and Knowledge) Artigos × You
Coverage_NPHard_PRletters-Revised-Marked Artigos × You
Simulation Multi-label Distribution Artigos × You
Coverage_NPHard_PRletters (Revised) (final) Artigos × You
Coverage_NPHard_PRletters Artigos × You
Coverage_NPHard_jmlr Artigos × You
contribution Artigos × You