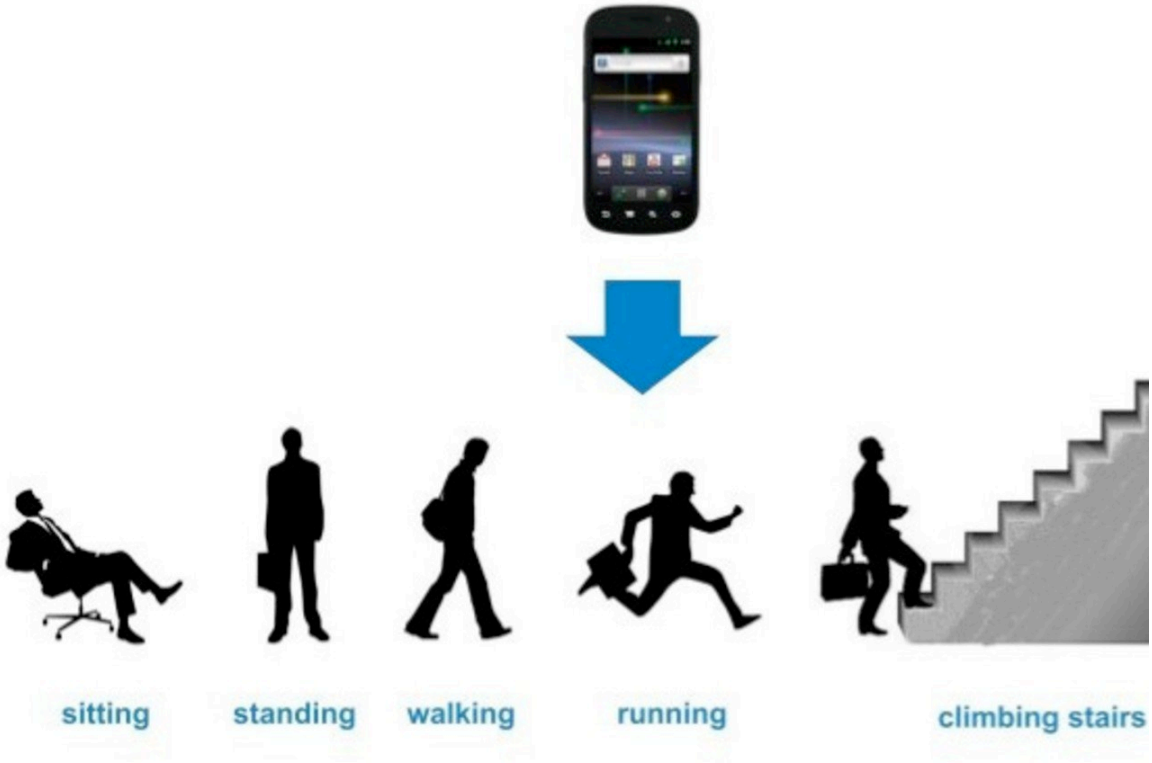
Tópicos de Ciência de Dados

Trabalho Laboratorial

Classificação de Atividades Humanas



Introdução

Período de execução: 11 aulas práticas laboratoriais

Esforço extra aulas previsto: 32h

Datas de Metas:

- Entrega da componente A: 04-11-2022
- Entrega da componente B: 16-12-2022

Objectivo: O objectivo central deste trabalho prático é que o aluno exercite conceitos centrais de um pipeline de análise de dados, passando pelas fases de preparação de dados, a sua limpeza, a extração de características descritivas, a sua seleção/redução e a aprendizagem computacional.

Trabalho Prático

O problema proposto no presente trabalho prático é um problema típico de classificação com que comummente se deparam os cientista de dados. O contexto do exercício proposto é o reconhecimento de atividades humanas. Este é um contexto com uma importância crescente em múltiplas situações, abrangendo, por exemplo, aplicações médicas, aplicações recreativas e de bem-estar. Independentemente do problema específico e das suas potenciais aplicações, o presente contexto irá permitir exercitar e interiorizar conceitos centrais em qualquer pipeline de análise dados com que um cientista de dados se confronta: dado um volume (elevado) de dados reais, desenvolver um classificador (não-linear) que permita identificar um conjunto de estados.

Figura 1: localização dos sensores.

No presente trabalho iremos usar o dataset FORTH-TRACE benchmark[1]. Este dataset foi adquirido usando 5 sensores (vide Figura 1) e inclui sensores de aceleração, velocidade angular e variação do campo magnético, quer da parte superior, quer da parte inferior, do corpo. O dataset é composto por dados adquiridos de 15 participantes usando um protocolo que envolvia 16 atividades distintas listadas na tabela Tabela 1: Atividades. O dataset original pode ser descarregado usando o seguinte link: https://github.com/spl-icsforth/FORTH_TRACE_DATASET. O dataset contém os seguintes ficheiros:

- partX/partXdev1.csv
- partX/partXdev2.csv
- partX/partXdev3.csv
- partX/partXdev4.csv
- partX/partXdev5.csv

em que X corresponde ao ID do participante e 1 a 5 corresponde ao ID do dispositivo (vide Tabela 2).

Cada ficheiro CSV segue o formato seguinte:

- Coluna 1: Device ID
- Coluna 2: accelerometer x
- Coluna 3: accelerometer y
- Coluna 4: accelerometer z
- Coluna 5: gyroscope x
- Coluna 6: gyroscope y
- Coluna 7: gyroscope z
- Coluna 8: magnetometer x
- Coluna 9: magnetometer y
- Coluna 10: magnetometer z
- Coluna 11: Timestamp
- Coluna 12: Activity Label

Tabela 1: Atividades

Etiqueta	Atividade
1	Stand
2	Sit
3	Sit and Talk
4	Walk
5	Walk and Talk
6	Climb Stair (up/down)
7	Climb Stair (up/down) and talk
8	Stand-> Sit

[1] Katerina Karagiannaki,Athanasia Panousopoulou,Panagiotis Tsakalides, A Benchmark Study on Feature Selection for Human Activity Recognition, UBICOMP/ISWC '16, (https://dl.acm.org/doi/pdf/10.1145/2968219.2971421)

9	Sit-> Stand
10	Stand-> Sit and talk
11	Sit->Stand and talk
12	Stand-> walk
13	Walk-> stand
14	Stand -> climb stairs (up/down), stand -> climb stairs (up/down) and talk
15	Climb stairs (up/down) -> walk
16	Climb stairs (up/down) and talk -> walk and talk

Tabela 2: Identificadores dos dispositivos

ID	Atividade
1	Pulso esquerdo
2	Pulso direito
3	Peito
4	Perna superior direita
5	Perna inferior esquerda

A . Elaboração de um conjunto de scripts e funções em Python, NumPy e SciPy para realizar as tarefas de preparação dos dados e *Feature Engineering*

1. 🖮 Crie um script e grave-o com o nome **'mainActivity.py'**. Este script será utilizado na chamada de todas as funções indicadas abaixo.

2. Descarregue os dados do site https://github.com/spl-icsforth/FORTH_TRACE_DATASET.
   - 🖮 Elabore uma rotina que carregue os dados relativos a um indivíduo e os devolva num Array NumPy. Poderá usar, por exemplo, a biblioteca CSV (https://docs.python.org/3/library/csv.html).

3. **Análise e tratamento de *Outliers***: o objectivo será identificar e tratar *outliers* no *dataset* usando diferentes abordagens univariável e multivariável. Para o efeito iremos utilizar os módulos dos vectores aceleração, giroscópio e magnetómetro. Seja

$$\vec{t} = \left(t_x, t_y, t_z\right)$$

   o vector aceleração, giroscópio e magnetómetro. O respectivo módulo é determinado recorrendo:

$$\left\|\vec{t}\right\| = \sqrt{t_x^2 + t_y^2 + t_z^2}$$

   3.1. 🖮 Elabore uma rotina que apresente simultaneamente o *boxplot* de cada atividade (coluna 12 – eixo horizontal)

relativo a todos os sujeitos e a uma das seguintes variáveis transformadas: módulo do vector de aceleração, módulo do vector de giroscópio e módulo do vector de magnetómetro). Sugere-se o uso da biblioteca *matplotlib* (veja, por exemplo, matplotlib.pyplot.boxplot - https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.boxplot.html).

3.2. ✍ Analise e comente a densidade de *Outliers* existentes no *dataset* transformado, isto é, nos módulos dos vectores aceleração, giroscópio e magnetómetro para cada atividade (use somente os sensores do pulso direito). Observe que a densidade é determinada recorrendo

$$d = \frac{n_o}{n_r} \times 100$$

em que n_o é o número de pontos classificados como *outliers* e n_r é o número total de pontos.

3.3. ⌨ Escreva uma rotina que receba um *Array* de amostras de uma variável e identifique os *outliers* usando o teste Z-Score para um *k* variável (parâmetro de entrada).

3.4. ✍ Usando o Z-score implementado, assinale todos as amostras consideradas *outliers* nos módulos dos vectores de aceleração, giroscópio e magnetómetro. Apresente *plots* em que estes pontos surgem a vermelho, enquanto que os restantes surgem a azul. Use *k*=3, 3.5 e 4.

3.5. ✍ Compare e discuta os resultados obtidos em 3.1 e 3.4.

3.6. ⌨ Elabore uma rotina que implemente o algoritmo K-means para *n* (valor de entrada) clusters.

3.7. ✍ Determine os *outliers* no *dataset* transformado usando o k-means. Experimente diferentes números de *clusters* e compare com os resultados obtidos em 3.4. Ilustre graficamente os resultados usando plots 3D para cada vetor (veja, por exemplo, https://towardsdatascience.com/an-easy-introduction-to-3d-plotting-with-matplotlib-801561999725).

3.7.1. Bónus: poderá realizar um estudo análogo usando o algoritmo DBSCAN (sugere-se que recorra à biblioteca *sklearn*[2])

3.8. ⌨ Implemente uma rotina que injete outliers com uma densidade igual ou superior a x% nas amostras da variável fornecida. Para o efeito deverá:

- A calcular a densidade de outliers existente no Array fornecido com n_r pontos; observe que a densidade *d* é obtida por

[2] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html#sklearn.cluster.DBSCAN

$$d = \frac{n_o}{n_r} \times 100$$

em que

$$n_o \equiv \#\left\{ p \notin \left[\mu - k\sigma, \mu + k\sigma \right] \right\}$$

- Se a densidade *d* for inferior a x, então deverá sortear (x-d)% dos pontos não *outliers* de forma aleatória e para cada ponto selecionado deverá transformá-lo tal que

  $$p \leftarrow \mu + s \times k \times \left(\sigma + q \right)$$

  em que μ e σ representam, respectivamente, os valores médio e o desvio padrão da amostra, *k* é o limite especificado no ponto 3.3, $s \in \{-1,1\}$ é uma variável escolhida de forma aleatória usando uma distribuição uniforme e *q* é uma variável aleatória uniforme no intervalo $q \in [0,z[$ em que z é a amplitude máxima do *outlier* relativamente a $\mu \pm k\sigma$. (observe que ao acrescentar outliers poderá adulterar as características da distribuição o que poderá induzir alterações da classificação de pontos previamente classificados como outliers; tal obriga a um processo iterativo)

3.9. Elabore uma rotina que determine o modelo linear de ordem *p*. Para o efeito, a sua rotina deverá receber *n* amostras de treino de um vector de dimensão *p*, i. e., ($x_{i,1}$, $x_{i,2}$, $x_{i,2}$, ... , $x_{i,p}$) e a respectiva saída y_i. A sua rotina deverá determinar o melhor vector de pesos β tal que

$$\underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{p} \left(y_i - \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ldots + \beta_p x_{i,p} \right)^2 = \underset{\beta}{\operatorname{argmin}} \left\| Y - X\beta \right\|^2$$

3.10. Determine o modelo linear para o módulo aceleração usando uma janela com *p* valores anteriores. Usando a rotina desenvolvida no ponto 3.9 injete 10% de *outliers* no módulo da aceleração. Elimine esses *outliers* e substitua-os pelos valores previstos pelo modelo linear. Analise o erro de predição apresentando i) a distribuição do erro e ii) exemplos de plots contendo o valor previsto e real. Determine o melhor p para o seu modelo (sugestão: poderá usar estratégias LOO – leave one out ou mesmo GCV – generalized cross validation).

3.11. Repita 3.10 usando uma janela de dimensão *p* centrada no instante a prever. Deverá usar não só os p/2 valores anteriores e seguintes da variável que pretende prever bem como das restantes variáveis disponíveis (módulos disponíveis). Compare com os resultados obtidos em 3.10.

4. Extração de informação característica: o objectivo será comprimir o espaço do problema, extraindo informação característica discriminante que permita implementar soluções eficazes do problema de classificação.

4.1. ✍ Usando as variáveis aplicadas na alínea 3.1, determine a significância estatística dos seus valores médios nas diferentes atividades. Observe que poderá aferir a gaussianidade da distribuição usando, por exemplo, o teste Kolmogorov-Smirnov (vide documentação do SciPy). Para rever a escolha de testes estatísticos sugere-se a referência[3]. Comente.

4.2. ⌨ Desenvolva as rotinas necessárias à extração do feature set temporal e espectral sugerido no artigo[4]. Para o efeito deverá:

- Ler o artigo e identificar o conjunto de features temporais e espectrais identificadas por estes autores
- Para cada feature deverá elaborar uma rotina para a respectiva extração
- Usando as rotinas elaboradas no item anterior, deverá escrever o código necessário para extrair o vetor de features em cada instante.
  - Nota: Poderá usar as bibliotecas NumPy e SciPy. Qualquer outra biblioteca deverá ser identificada.

4.3. ⌨ Desenvolva o código necessário para implementar o PCA de um feature set; poderá usar implementações existentes.

4.4. ✍ Determine a importância de cada vetor principal na explicação da variabilidade do espaço de features. Note que deverá normalizar as features usando o z-score. Quantas dimensões deverá usar para explicar 75% do feature set?

4.4.1. Indique como poderia obter as features relativas a esta compressão e exemplifique para um instante à sua escolha.

4.4.2. Indique as vantagens e as limitações desta abordagem.

4.5. ⌨ (Este ponto será transferido para a parte B deste trabalho; apresenta-se aqui meramente por uma questão de coerência) Desenvolva o código necessário para implementar o Fisher feature Score e o ReliefF; poderá usar implementações existentes.

---

[3] Jean-Baptist du Prel, Dr. med.,1 Bernd Röhrig, Dr. rer. nat.,2 Gerhard Hommel, Prof. Dr. rer. nat.,3
3 Jean-Baptist du Prel, Bernd Röhrig, Gerhard Hommel, and Maria Blettner, Choosing Statistical Tests, Deutsches Arzteblatt, v107(19), 2010. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2881615/

[4] Mi Zhang, Alexander A Sawchuk, A. Sawchuk, A feature selection-based framework for human activity recognition using wearable multimodal sensors, : BodyNets '11: Proceedings of the 6th International Conference on Body Area Networks, November 2011 Pages 92–98. https://pdfs.semanticscholar.org/8522/ce2bfce1ab65b133e411350478183e79fae7.pdf

4.6. ✍ Identifique as 10 melhores features de acordo com o Fisher Score e o ReliefF e compare os resultados.

4.6.1. Indique como poderia obter as features relativas a esta compressão e exemplifique para um instante à sua escolha.

4.6.2. Indique as vantagens e as limitações desta abordagem.

B . Elaboração de um conjunto de scripts e funções em Python, NumPy, SciPy e Scikit-learn para realizar as tarefas de Aprendizagem Computacional e Avaliação

(O enunciado deste componente será disponibilizado mais tarde)