

TIP8311 - Reconhecimento de Padrões

Programa de Pós-Graduação em Engenharia de Teleinformática
Universidade Federal do Ceará (UFC)

Responsável: Prof. Guilherme de Alencar Barreto

2º. Trabalho Computacional - 03/02/2022

Questão Única – Acesse através do link abaixo o conjunto de dados para uso neste trabalho.

<https://archive.ics.uci.edu/ml/datasets/Parkinsons>

Pede-se:

1. Implementar o classificador bayesiano gaussiano cuja função discriminante mostrada na Eq. (12) da nota de aula “Critério MAP e Classificadores Bayesianos Gaussianos”.

2. Implementar o classificador bayesiano gaussiano segundo a função discriminante mostrada na Eq. (44) da mesma nota de aula e compará-lo com a implementação do Item 1. Há diferenças de desempenho entre as duas implementações?

3. Implementar o classificador bayesiano gaussiano cuja função discriminante mostrada na Eq. (17) da nota de aula “Critério MAP e Classificadores Bayesianos Gaussianos”. Usar a matriz de covariância agregada (*pooled*).

4. Implementar o classificador bayesiano gaussiano cuja função discriminante mostrada na Eq. (39) da nota de aula e compará-lo com a implementação do Item 3. Há diferenças de desempenho entre as duas implementações?

5. Implementar também o classificador linear de mínimos quadrados (LMQ) e comparar seu desempenho com os classificadores dos itens 1 a 4.

6. Aplicar PCA ao conjunto de dados e repetir o estudo comparativo.

Preencher a tabela de resultados abaixo. Número de rodadas independentes de treino/teste: 100. Separação treino/teste: 80/20% por classe.

Classificador	Média ± Desvio-padrão	Mediana	[Mínimo, Máximo]
Item1			
Item2			
Item3			
Item4			
Item5			
Item1+PCA			
Item2+PCA			
Item3+PCA			
Item4+PCA			
Item5+PCA			

7. Construa as matrizes de confusão para as classes para o melhor e pior caso ao longo das 100 rodadas. Quais classes são mais fáceis de classificar? Quais são mais difíceis? Elencar possíveis explicações para os resultados.

Obs. 1: Para estimar as matrizes de covariância das classes, podem ser usados comandos prontos disponíveis no Matlab/Octave e em bibliotecas e pacotes de outras linguagens de programação.

Obs. 2: Avaliar a invertibilidade das matrizes de covariância das classes através dos seus postos. Usar comandos `rank` e `rcond` no Octave/Matlab ou seus equivalentes em outras linguagens de programação.

Obs. 3: Em caso de problemas de inversão das matrizes de covariância, usar estratégias de regularização discutidas nas Variantes 1 e 3 do classificador quadrático.

Boa Sorte!