

Documentação PIBIC 2017

Daniele Silva Reis

Sumário

1	Transformada de Fourier	2
2	Transformada Wavelet	4
3	Pré-tratamentos	5
4	Regressão	6
4.1	PLS	6
4.2	SVM	8
5	Classificação com PLS-DA e SVM	9
6	Thresholds pelo Teorema de Bayes	10
7	Calculando métricas para classificação	10

Em todos os *scripts* desenvolvidos, considera-se que o arquivo .xlsx de entrada contém uma matriz $[n + 1, m + 3]$, em que:

- n é o número de amostras;
- m é o número de variáveis (comprimentos de onda).

As colunas extras correspondem ao número da coleta, código da amostra e um atributo alvo, enquanto que a linha extra corresponde aos nomes das colunas.

1 Transformada de Fourier

Para realizar a filtragem passa-baixas, selecione o arquivo *tela_inicial.m* (na pasta *fourier*) dentro do ambiente MATLAB. A seguinte tela aparecerá:

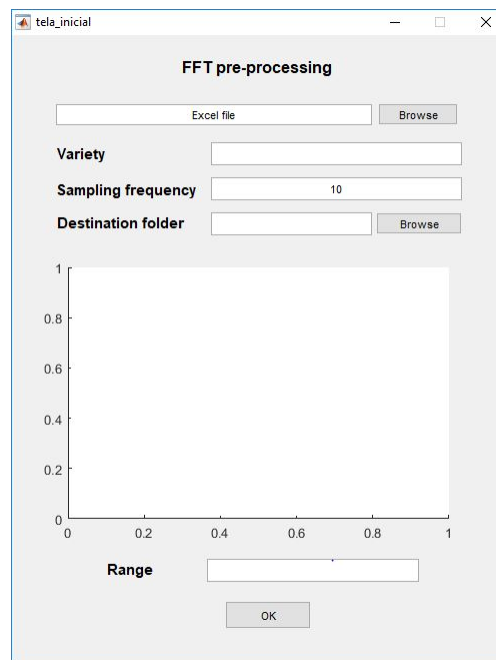


Figura 1: Tela inicial do *script* para *denoising* por filtragem passa-baixas.

O arquivo .xlsx de entrada deve ser selecionado através do primeiro botão *Browse*. O campo *variety* é opcional, mas, se a variedade for inserida, esta será utilizada no nome do arquivo que será gerado. No campo *Sampling frequency*, cujo valor *default* é 10, deve-se inserir a frequência de amostragem desejada. No segundo botão *Browse* deve-se selecionar a pasta de destino onde o arquivo de saída será salvo.

O usuário deve então inserir a faixa de comprimento de onda em que a filtragem será aplicada. Deve-se seguir a notação *val1-val2*, em que *val1* corresponde ao primeiro comprimento de onda desejado e *val2* ao último. Caso o usuário desejar aplicar a filtragem em faixas separadamente, ele deve inserir o ponto e vírgula entre elas. Ex.: 450-600;700-900.

Após clicar em *OK*, a tela exibida na Figura 2 será exibida.

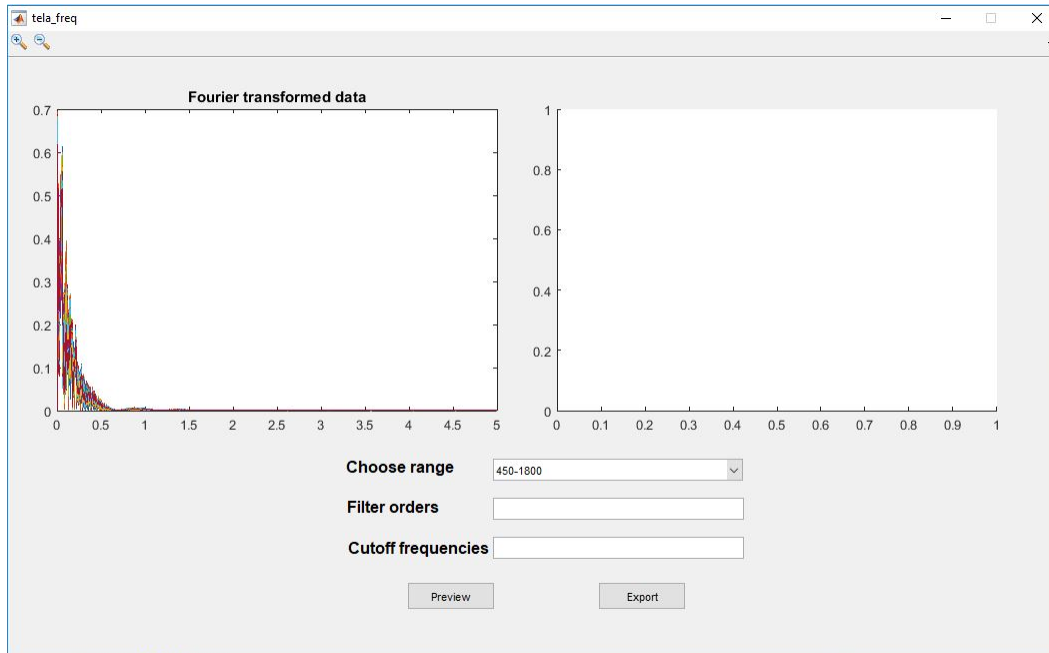


Figura 2: Segunda tela do *script* para *denoising* por filtragem passa-baixas.

Caso o usuário tenha inserido mais de uma faixa de comprimento de onda, ele pode selecionar uma deles através da *combobox Choose range* para uma visualização adequada.

A ordem do filtro passa-baixas é inserida no campo *Filter orders*. Para a filtragem em mais de uma faixa, deve-se inserir as ordens dos filtros com um ponto e vírgula entre eles, ex.: 10;16. Da mesma forma, as frequências de corte devem ser separadas por ponto e vírgula.

Obs.: Mesmo que for desejado um mesmo filtro para todas as faixas, deve-se inserir uma ordem e frequência de corte para cada faixa, que nesse caso serão repetidas. Ex.: 10;10 e 1;1.

Para visualizar o sinal tratado, deve-se clicar no botão *Preview* e, para exportá-lo, no botão *Export*.

ATENÇÃO: o sinal só será exportado corretamente se o botão *Preview* for clicado antes do *Export*.

O arquivo resultante será salvo na pasta de destino inserida anteriormente, com o nome possuindo a seguinte notação:

$$variety_o_o1_f_f1.xlsx$$

Em que *variety* é a variedade inserida e *o1* e *f1* são a ordem do filtro e frequência de corte respectivamente. Para a filtragem em mais de uma faixa, as demais ordens e frequências aparecem seguidas de *underscore*.

O arquivo salvo contém a mesma estrutura do arquivo de entrada, com a única diferença sendo os valores de absorbância por comprimento de onda, que foram tratados.

2 Transformada Wavelet

Para realizar o *denoising* por transformada *wavelet*, abra o arquivo *tela_inicial2.m* (na pasta *wavelet*) dentro do ambiente MATLAB. A seguinte tela aparecerá:

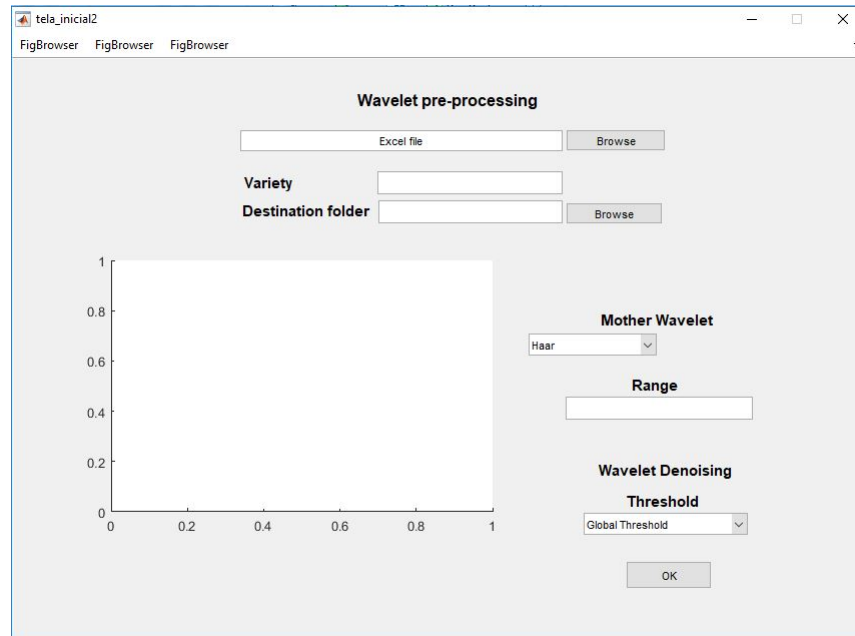


Figura 3: Primeira tela do *script* para *denoising* por transformada *wavelet*.

De forma semelhante ao *script* para filtragem passa-baixas, deve-se selecionar o arquivo de entrada, a variedade e pasta de destino.

A diferença encontra-se na configuração da transformada *wavelet*. Deve-se inicialmente selecionar o tipo de *mother wavelet* e, dependendo da escolhida, outra *combobox* será mostrada, possibilitando a escolha de um subtipo de *mother wavelet*.

Insere-se então a faixa de comprimentos de onda em que o alisamento será realizado, em que também é possível colocar mais de uma, utilizando a mesma notação do *script* anterior: 400-600;1400-1800, por exemplo.

Por fim, seleciona-se o tipo de *threshold* que será aplicado. Ao clicar em *OK*, a seguinte tela mostrada na Figura 4 aparecerá, contendo o sinal alisado para os 12 primeiros níveis de decomposição:

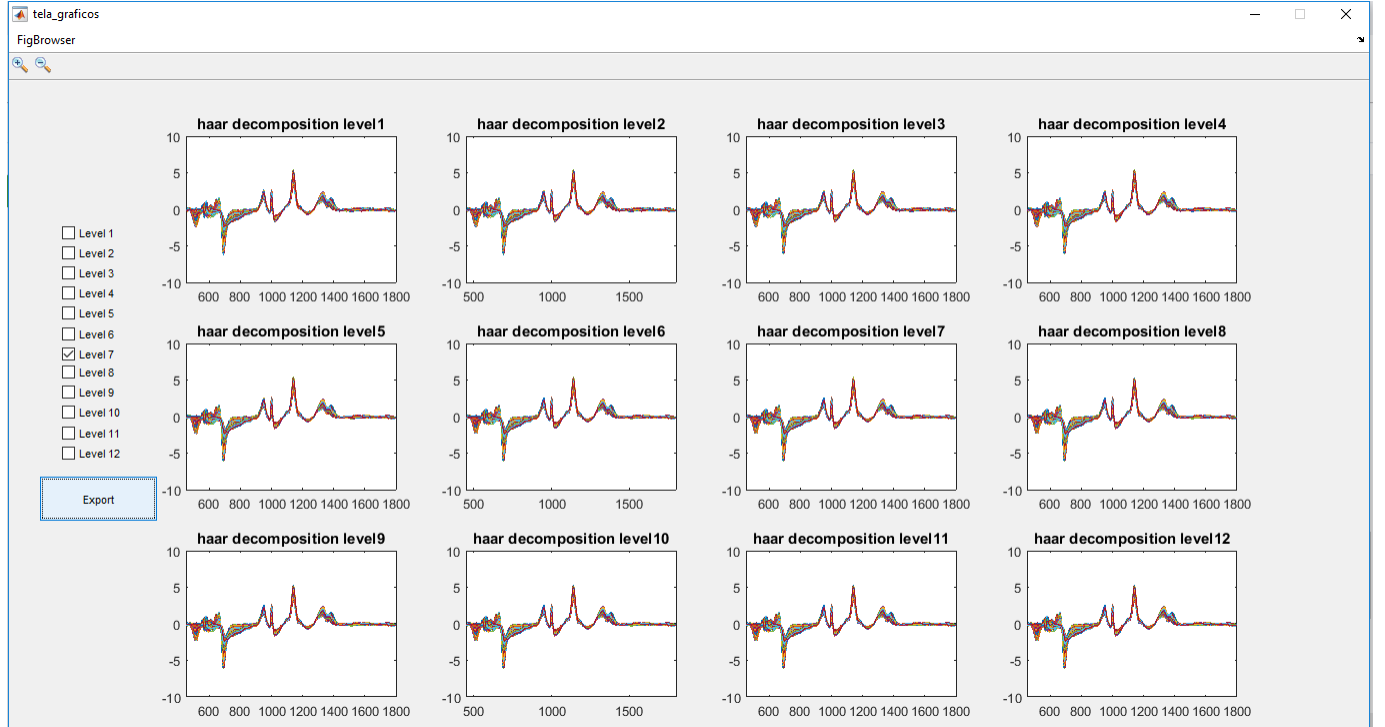


Figura 4: Segunda tela do *script* para *denoising* por transformada *wavelet*.

O usuário pode então exportar o(s) sinal(s) tratado(s) que ele desejar, através da seleção do(s) respectivo(s) *checkbox(es)* e clique no botão *Export*.

O arquivo resultante será salvo na pasta de destino inserida anteriormente, com o nome possuindo a seguinte notação:

$$variety_rn\ thr_wname\ lm.xlsx$$

Em que *variety* é a variedade inserida, o *n* em *rn* consiste no número de faixas inseridas, *thr* é o tipo de *threshold* selecionado, *wname* a *mother wavelet* e seu subtipo escolhidos e o *m* de *lm* é o nível de decomposição.

O arquivo salvo contém a mesma estrutura do arquivo de entrada, com a única diferença sendo os valores de absorbância por comprimento de onda, que foram tratados.

3 Pré-tratamentos

O *script* para pré-tratamento localiza-se na pasta *pre-treatment*. Deve-se executar o arquivo *tela_inicial.m* no MATLAB, de forma que a tela na Figura 5 será exibida.

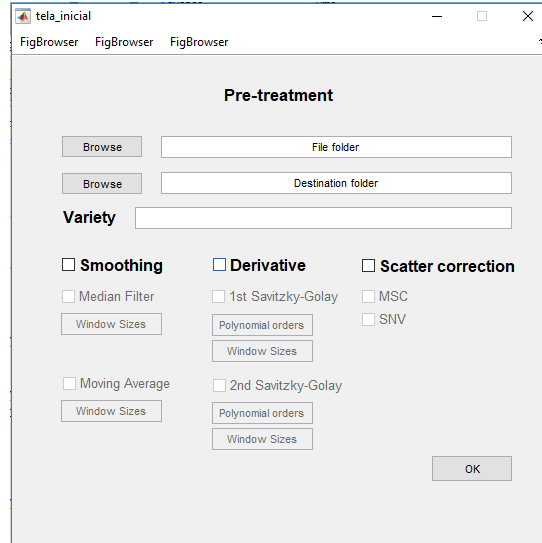


Figura 5: Tela do *script* para tratamento por alisamento, derivadas e normalização.

No primeiro botão *Browse*, seleciona-se uma pasta que contenha arquivos *.xlsx* que deseja-se processar. No segundo botão *Browse*, o usuário deve escolher a pasta de destino onde os dados tratados serão salvos. Por fim, o usuário deve digitar a variedade da fruta analisada.

O usuário então escolherá os tipos de tratamento que deseja submeter aos seus dados: alisamento, diferenciação e normalização, com seus respectivos algoritmos. Nos campos de texto deve-se inserir os parâmetros de cada técnica, como tamanho da janela e ordem do polinômio. Caso seja desejado gerar o pré-tratamento para vários parâmetros pro vez, insira-os com uma vírgula os separando. Ex.: 7,9,11 para tamanhos de janela.

Após clicar em *OK*, o *script* gerará todas as combinações possíveis de pré-tratamentos dados os *checkboxes* selecionados. Os nomes dos arquivos resultantes serão os mesmos dos arquivos originais mais a abreviação dos pré-tratamentos aplicados.

ATENÇÃO: Os pré-tratamentos por diferenciação e normalização só são executados se a PLS Toolbox estiver disponível!

4 Regressão

4.1 PLS

Para realizar a regressão com PLS, vá até a pasta *pls*, abra o arquivo *tela_inicial.m* no MATLAB e o execute. A seguinte tela será exibida:

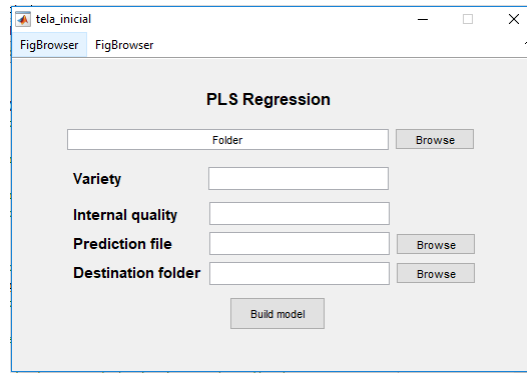


Figura 6: Tela do *script* para regressão por PLS.

Clique no primeiro botão *Browse* para selecionar uma pasta que contenha arquivos .xlsx que serão utilizados para calibração dos modelos. Cada um destes arquivos, que contêm a matriz de tamanho $[n + 1, m + 3]$, será utilizado para a construção de um modelo de regressão cada.

Após isso, indique a variedade da amostra e o atributo alvo nos campos *Variety* e *Internal quality* respectivamente. Estes campos são opcionais mas serão úteis na exportação dos resultados.

Selecione então o arquivo de predição, que será utilizado para o teste dos modelos construídos. Este campo também é opcional. Se não for escolhido nenhum arquivo, o *script* somente dará as métricas para calibração e *cross-validation*.

Por fim, escolha a pasta de destino onde os resultados serão salvos. Ao clicar em *OK*, a seguinte tela será exibida:

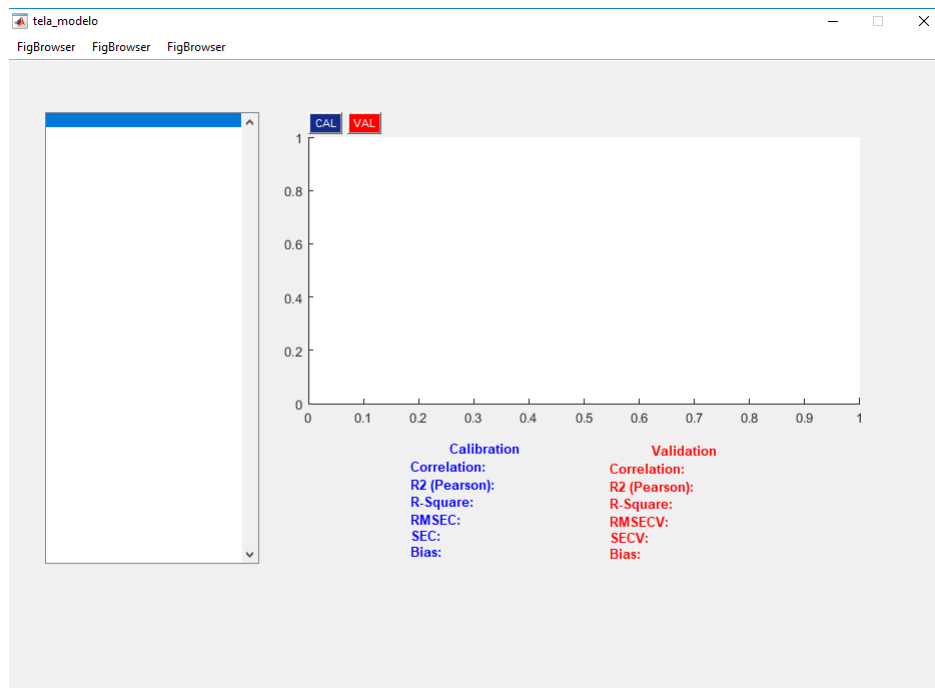


Figura 7: Segunda tela do *script* para regressão por PLS.

No menu à esquerda serão exibidos os modelos construídos. Ao clicar em cada um, serão exibidos os respectivos gráficos de dispersão métricas para calibração e *cross-validation*. A exportação dos resultados é feita automaticamente.

ATENÇÃO: A regressão por PLS só será realizada se a PLS Toolbox estiver disponível!

4.2 SVM

O *script* para construção de modelos SVM está localizado na pasta *svm*. Execute o arquivo *tela_inicial.m* no ambiente MATLAB e a seguinte janela será exibida:

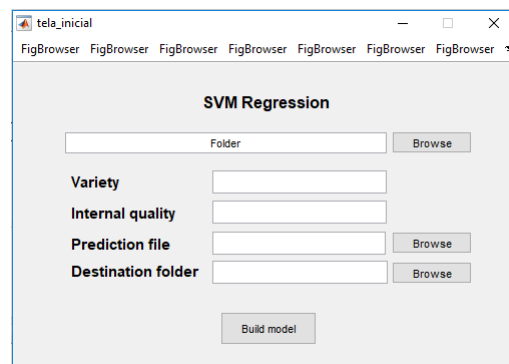


Figura 8: Tela do *script* para regressão por SVM.

Os campos mostrados na Figura 8 são os mesmos mostrados na Figura 6: arquivos de entrada, pasta de destino, arquivo de teste, variedade e atributo alvo. Da mesma forma, não é necessário inserir um arquivo para teste; na sua ausência, apenas as métricas para calibração e *cross-validation* serão computadas.

A execução do modelo é realizada após clicar no botão *OK*, que abre a tela mostrada na Figura 9.

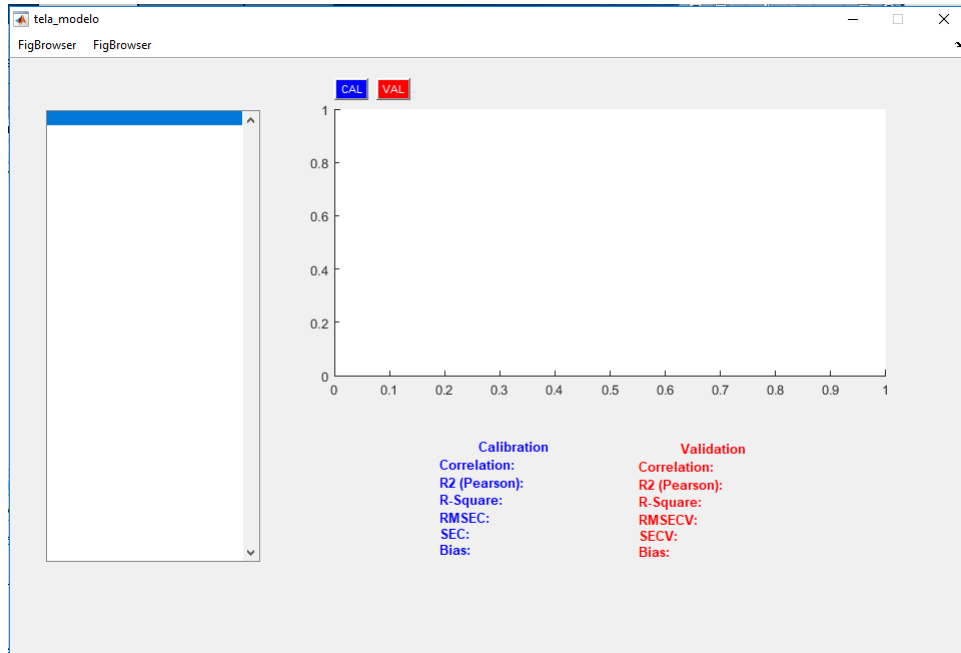


Figura 9: Segunda tela do *script* para regressão por SVM.

Neste *script* os resultados e gráficos de dispersão são exibidos quando o usuário clicar no modelo desejado, localizado no menu à esquerda.

Os resultados são exportados automaticamente para a pasta de destino informada na tela anterior.

ATENÇÃO: A regressão por SVM só será realizada se a PLS Toolbox estiver disponível!

5 Classificação com PLS-DA e SVM

Os *scripts* para classificação com as técnicas PLS-DA e SVM estão localizados na pasta *classification*. Abra o arquivo *tela_inicial.m* no ambiente MATLAB e o execute. A tela mostrada na Figura 10 será exibida.

Figura 10: Tela do *script* para classificação por PLS-DA e SVM.

Inicialmente selecione, através do primeiro botão *Browse*, uma pasta que contém o(s) arquivo(s) .xlsx de entrada. Cada um destes arquivos, que contém a matriz de tamanho $[n + 1, m + 3]$, será utilizado para a construção de um modelo de classificação cada.

Após clicar no segundo botão *Browse*, selecione a pasta em que serão salvos os resultados da predição.

Por fim, através do terceiro e último botão *Browse*, selecione o arquivo que será utilizado como teste para cada modelo construído. Por fim, clique em *OK*.

Os arquivos de resultado para cada método (PLS-DA e SVM) conterão a matriz de confusão para os dados de teste, assim como as métricas acurácia, sensibilidade, sensibilidade e taxa de falsos positivos. O nome do arquivo é dado pela concatenação do nome original com o método de classificação empregado.

ATENÇÃO: A classificação por ambos algoritmos só será realizada se a PLS Toolbox estiver disponível!

6 Thresholds pelo Teorema de Bayes

O arquivo contendo o *script* para determinação de *thresholds* pelo Teorema de *Bayes* está localizado na pasta *classification*, sob o nome *bayesian.m*. Após abrir o arquivo no MATLAB ou apenas a pasta em que o arquivo está localizado, você deve criar duas variáveis através da janela de comando:

- Um *array* que contém os valores de referência;
- Um *array* que contém as classes existentes (representadas por números).

Considerando que as duas variáveis criadas são denominadas *reference* e *classes* respectivamente, chame a função *bayesian.m* pela janela de comando como mostrado a seguir:

```
thresholds = bayesian(reference , classes)
```

Os *thresholds* computados pelo *script* serão exibidos também na janela de comando.

7 Calculando métricas para classificação

O arquivo *calculate_metrics.m*, localizado na pasta *classification*, contém o *script* que calcula métricas de classificação como acurácia, sensibilidade, sensibilidade e taxa de falsos positivos, além da matriz de confusão.

Para executá-lo, abra a pasta *classification* ou o próprio arquivo no MATLAB e crie as seguintes variáveis:

- Um *array* que contém os valores previstos por um modelo;
- Um *array* que contém os valores de referência;
- Um *array* que contém as classes existentes (representadas por números).

Considerando que as três variáveis criadas são denominadas *predicted*, *reference* e *classes* respectivamente, chame a função *calculate_metrics.m* pela janela de comando como mostrado a seguir:

```
metrics = calculate_metrics(classes , predicted , reference)
```

O resultado será então exibido na tela. Como a variável *metrics* retornada é do tipo *struct*, você pode verificar a matriz de confusão ao inserir o seguinte comando:

```
metrics.conf_matrix
```