**Prática de Laboratório 3: classificação com máquinas de vectores de suporte**

Nesta prática examinar-se-à a utilização das máquinas de vectores de suporte (SVM) em problemas de classificação.

Na mesma serão utilizados três conjuntos de dados. Dois dos conjuntos são sintéticos, tendo como finalidade a análise de conceitos básicos acerca das SVM, enquanto que o terceiro é um conjunto de dados real. Este último será utilizado para o estudo do procedimento de treino e teste de um classificador.

Com a prática se pretende alcançar os seguintes objectivos:

* Utilizar as máquinas de vectores de suporte para o treino de classificadores para conjuntos diversos.
* Analisar conceitos básicos relacionados ao método, tais como superfície de decisão, vectores de suporte, função *kernel*, etc.
* Empregar um procedimento para a busca de parâmetros adequados e treino de um classificador baseado em SVM.

Como pré-requisitos é importante o estudo de diversos aspectos:

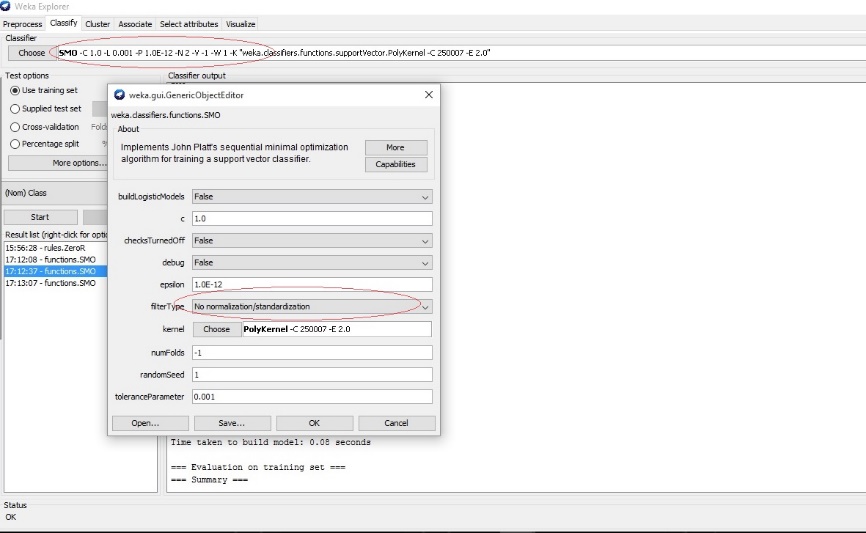
* Avaliação dos modelos e métodos de validação
  + Medidas de desempenho, *holdout*, validação cruzada, matriz de confusão
  + Consultar: Witten, pgs. 147 – 150; 152 – 157; 180 - 182
* Algoritmo SVM
  + Consultar: Hsu, C. W., Chang, C. C. e Lin, C. J., “A Practical Guide to Support Vector Classification”
  + Consultar: Witten, pg. 124 – 131, 223 – 227

**Tarefa 1: classificação com dados linearmente separáveis**

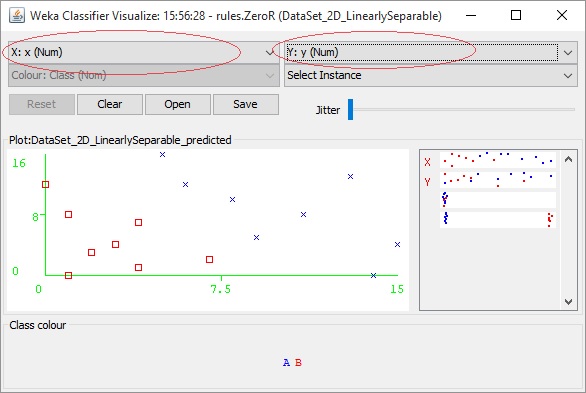
Nesta actividade utilizaremos o Weka para treinar um classificador SVM para o caso de um problema simples, cujo objectivo é o reforço do entendimento básico dos conceitos relacionados às máquinas de vectores de suporte.

Treinemos primeiro uma SVM para um problema linearmente separável. Para tal utilizaremos o método SMO. SMO, abreviatura de *Sequential Minimal Optimization*, é um algoritmo que permite resolver o problema de optimização quadrática subjacente ao treino das SVM.

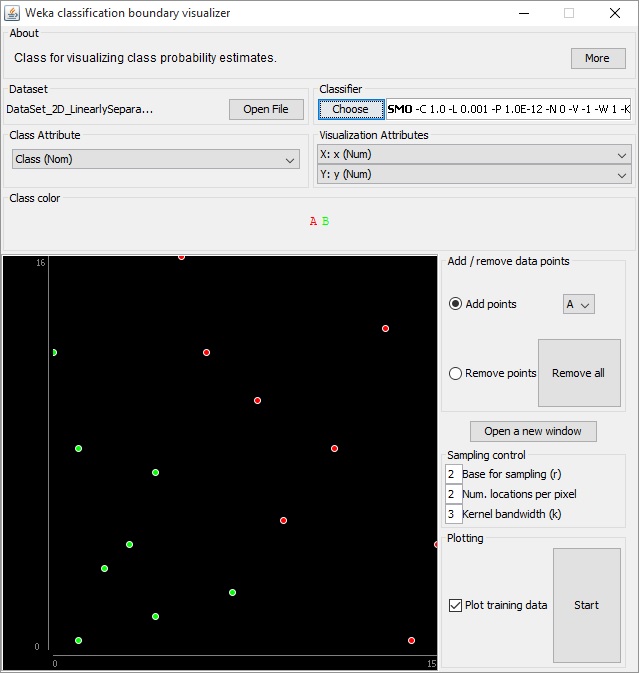
1. No painel *Preprocess*, carregamos o conjunto de dados *DataSet\_2D\_LinearlySeparable*. Observamos quais são os atributos e os valores correspondentes e as classes. Inspeccione os dados utilizando a secção *Visualize*.
2. Na secção *Classify*, faça click no botão *Choose* e seleccione *classifiers* > *functions* > *SMO*.
3. Na caixa de diálogo do classificador (click sobre a barra existente à frente do botão *Choose*) na propriedade *filterType* selecione a opção *No normalization/standardization*.



1. No painel *Test Options*, seleccione *Use Training Set*, para que o algoritmo, uma vez treinado, seja testado sobre o mesmo conjunto utilizado para o treino.
2. Faça click no botão *Start* para executar o classificador.
3. Visualizemos agora os resultados de duas formas diferentes.
   1. Primeiro, na lista de resultados (por baixo do botão *Start*) faça click com o botão direito do rato e seleccione a opção *Visualize classifier error*. Na janela correspondente, seleccione o atributo x para o eixo X e o atributo y para o eixo Y.



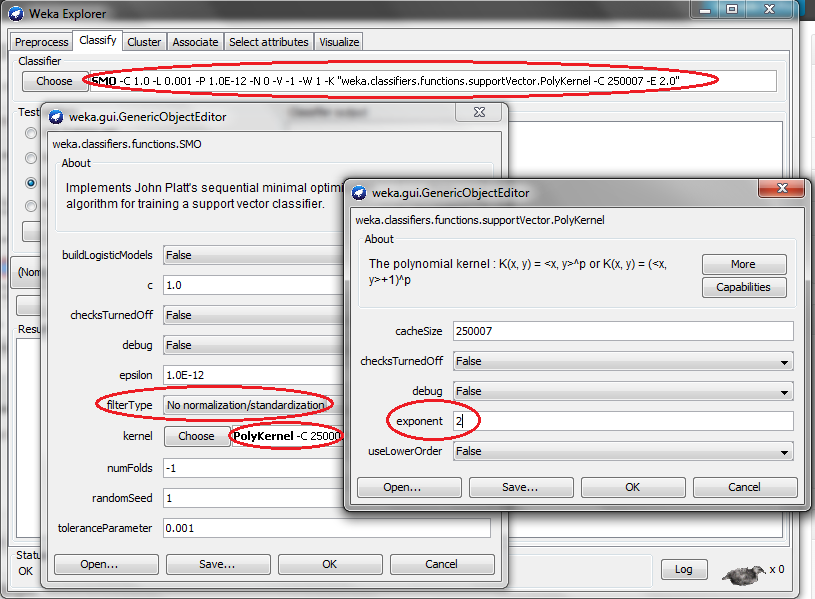
* 1. A outra forma consiste em utilizar a aplicação *Classification Boundary Visualizer* do WEKA. Esta não está incluída no Explorador, por isso, é necessário abrir a aplicação numa nova janela. Construa a superfície de separação para o mesmo conjunto de dados e classificador utilizado no passo anterior.
     1. No WEKA abra a interface de comandos (*Simple CLI*) e introduza o comando *java weka.gui.boundaryvisualizer.BoundaryVisualizer*.
     2. Seleccione o conjunto de dados e o classificador.
     3. Active a opção *Plot training data* e pressione o botão *Start*.



1. Analise as superfícies de decisão obtidas e descreva os erros de classificação.
2. Experimente acresecentar e remover alguns pontos, no *Boundary Visualizer*, e observe o que acontece com a superfície de decisão. Se altera em algum caso? Descreva a quantidade de pontos que acrescentou/removeu, em que áreas e as alterações que ocorrem na superfície de separação.
3. Ilustre todas as suas conclusões com os gráficos que achar conveniente.

Provemos agora treinar o classificador com um *kernel* polinomial primeiro com expoente 2 e depois com expoente 4.

1. Para tal abra a caixa de diálogo de opções do classificador (click sobre a barra existente à frente do botão *Choose*) e nas opções do *kernel* altere o valor do expoente (click na caixa de diálogo correspondente à propriedade *kernel*).



1. Execute o classificador.
2. Visualize os resultados da classificação utilizando os dois métodos anteriores.
3. Observe o que acontece com a quantidade de vectores de suporte e com o erro de classificação em cada caso.
4. Ilustre igualmente as suas conclusões.

**Tarefa 2: dados não separáveis linearmente**

Nesta actividade vamos experimentar a classificação de um conjunto de dados não separáveis no espaço de entradas. Provaremos o efeito da alteração do parâmetro de penalização C e das diferentes opções de *kernel* nos resultados da classificação.

1. Abra o ficheiro *DataSet\_2D\_NoLinearlySeparable*
2. Visualize e analise os dados correspondentes.
3. Seleccione como classificador SMO e execute-o com as opções por defeito, usando *Use training set* nas opções de teste. As opções por defeito correspondem a um *kernel* linear (*kernel* polinómico com expoente 1).
4. Visualize os resultados da classificação. O que é que se observa?
5. Na caixa de diálogo de opções do classificador experimente alterar o valor do parâmetro de penalização (C) para 10, 100 e depois para 1000. Se observa algum efeito nos resultados da classificação?
6. Independentemente da opção utilizada poderá observar que em nenhum dos casos foi possível classificar correctamente a totalidade dos dados. Isto se deve a que o *kernel* utilizado por defeito é linear e o conjunto de dados utilizado não é separável linearmente no espaço de entradas. Será separável noutro espaço? Provemos para tal outras opções de *kernel*.
7. Treine o mesmo classificador, utilizando desta vez o *kernel* polinomial. Seleccione nas opções do *kernel* o expoente 2.
8. Observe os resultados da classificação. Foram separados correctamente os dados?
9. Prove valores altos de C, 1000, 10000, 100000, etc. e verifique se é possível separar completamente os dados de treino.

**Tarefa 3: conjunto de dados real**

Investiguemos agora o processo de classificação de um conjunto de dados real, o conjunto p*ima indian diabetes*. Pode ver uma descrição do conjunto de dados no repositório da UCI.

Ao treinar uma SVM é necessário definir os valores adequados para uma série de parâmetros, o parâmetro de penalização C e do *kernel*, caso seja aplicável. Nesta tarefa veremos como fazer a determinação dos referidos parâmetros.

Dividamos primeiro o conjunto de dados em duas partes, conjuntos de treino e de teste.

1. Para tal, no Explorador basta fazer o seguinte:
   1. Conjunto de treino:
      1. Carregue o conjunto de dados *diabetes*.
      2. Seleccione o filtro *RemovePercentage* no painel de pré-processamento (*filters.unsupervised.instance.RemovePercentage*).
      3. Defina a percentagem correcta para a divisão dos dados, 60 % neste caso. Para tal faça click no espaço diante do botão *Choose* para abrir as opções do filtro.
      4. Aplique o filtro.
      5. Guarde os dados gerados como um novo ficheiro (*diabetes\_trn*)
   2. Conjunto de teste:
      1. Recarregue o conjunto de dados completo (ou use apenas o botão *Undo* para reverter as alterações para o conjunto de dados).
      2. Selecione o filtro *RemovePercentage* se ainda não estiver selecionado.
      3. Defina a propriedade como *invertSelection* como verdadeira. Para tal faça click no espaço diante do botão *Choose* para abrir as opções do filtro.
      4. Aplique o filtro.
      5. Guarde os dados gerados como novo ficheiro (*diabetes\_tst*).

Determinemos o melhor conjunto de parâmetros com base nos resultados de uma validação cruzada em 5 folhas realizada sobre o conjunto de treino.

1. Carregue o conjunto de dados de treino (*diabetes\_trn*).
2. Efectue de forma sucessiva a validação cruzada para as seguintes combinações de parâmetros:
   1. Kernel RBF: C = 0,01; 0,1; 1; 10; 100 G = 0,001; 0,01; 0,1; 1,0;10
   2. Kernel polinomial: C = 0,01; 0,1; 1; 10; 100; p = 2, 4, 8, 16
   3. Kernel linear: C = 0,01; 0,1; 1; 10; 100
3. Registe os valores das taxas de erro correspondentes nas tabelas. Tenha em conta que para os *kernel* RBF e polinómico as tabelas deverão ser de dupla entrada. Em que casos se registam as menores taxas de erro?

Treinemos agora o classificador utilizando as melhores combinações dos parâmetros correspondentes a cada *kernel* e apliquemos o mesmo aos dados de teste.

1. Carregue o conjunto de treino.
2. Seleccione o classificador e defina os valores dos parâmetros a utilizar.
3. Seleccione *Supplied test set* nas opções de teste e carregue o ficheiro de teste.
4. Treine o classificador e anote os resultados da classificação para cada alternativa testada.
5. Os resultados da classificação estão em concordância com os obtidos ao realizar a validação cruzada?