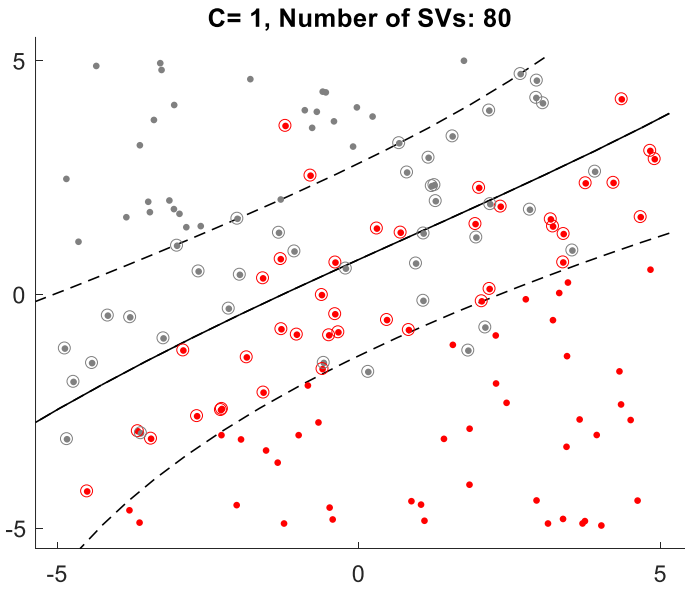
Aprendizado de Máquina e Reconhecimento de Padrões (UTFPR/CPGEI) - Lista de Exercícios 4

Tópicos: Classificadores Não-Lineares.

1. Este problema é dividido em cinco etapas, listadas a seguir:
   I. Gere um conjunto de dados bidimensionais **X**1 (treinamento) conforme segue. Selecione 150 pontos em um espaço bidimensional na região definida por [−5, 5] × [−5, 5], seguindo uma distribuição uniforme (na função rand do matlab, selecione "seed" igual a zero). Rotule o ponto **x** = [x(1), x(2)]^T com a classe +1 (−1) de acordo com a regra 0.05(x^3(1) +x^2(1) +x(1) + 1) > (<)x(2) (Claramente, as classes não são linearmente separáveis. De fato, elas são separadas pela curva associada à equação 0.05 (x^3(1) + x^2(1) + x(1) + 1) = x(2). Plote os dados de **X**1 (treinamento). Gere um conjunto adicional **X**2 (teste) usando a mesma descrição de **X**1 (na função rand do matlab, selecione "seed" igual a 100).
   II. Construa um classificador SVM linear com os parâmetros C = 2 e tol = 0.001. Calcule o erro de classificação de treinamento, de teste e o número de vetores suporte em cada caso. Plote as regiões de decisão definidas pelo classificador e as margens correspondentes.
   III. Construa um classificador SVM não-linear (RBF kernel) com os parâmetros C = 2, tol = 0.001, σ = 0,1 e 2. Calcule o erro de classificação de treinamento, de teste e o número de vetores suporte em cada caso. Plote as regiões de decisão definidas pelo classificador.
   IV. Construa um classificador SVM não-linear com kernel RBF e os parâmetros σ = 1,5 e com kernel polinomial e os parâmetros n = 3 and β = 1. Em ambos os casos, use tol = 0,001 e C = 0,2, 20, 200. Compare e comente os resultados.

A figura a seguir mostra um exemplo da distribuição dos dados e resultados para um caso particular do classificador:



2. Considere um problema de classificação bidimensional envolvendo três classes ω_1, ω_2 e ω_3. Os exemplos da classe ω_1 tem uma distribuição composta por duas Gaussianas com os parâmetros $\boldsymbol{\mu}_{11}$, $\boldsymbol{\mu}_{12}$, $\boldsymbol{\Sigma}_{11}$ e $\boldsymbol{\Sigma}_{12}$. De forma similar, os exemplos da classe ω_2 tem uma distribuição composta por

duas Gaussianas com os parâmetros $\boldsymbol{\mu}_{21}$, $\boldsymbol{\mu}_{22}$, $\boldsymbol{\Sigma}_{21}$ e $\boldsymbol{\Sigma}_{22}$. Já a classe ω_3 tem uma distribuição composta por uma Gaussiana com os parâmetros $\boldsymbol{\mu}_3$ e $\boldsymbol{\Sigma}_3$. Os valores dos parâmetros estão representados abaixo:

$$\Sigma_{11} = \begin{bmatrix} 0.2 & 0 \\ 0 & 2 \end{bmatrix} \quad \Sigma_{12} = \begin{bmatrix} 3 & 0 \\ 0 & 0.5 \end{bmatrix}$$

$$\Sigma_{21} = \begin{bmatrix} 5 & 0 \\ 0 & 0.5 \end{bmatrix} \quad \Sigma_{22} = \begin{bmatrix} 7 & 0 \\ 0 & 0.5 \end{bmatrix}$$

$$\Sigma_3 = \begin{bmatrix} 8 & 0 \\ 0 & 0.5 \end{bmatrix} \quad \mu_{11} = [0,\ 3]^{\mathrm{T}}$$

$$\mu_{12} = [11,\ -2]^{\mathrm{T}}$$

$$\mu_{21} = [3,\ -2]^{\mathrm{T}} \quad \mu_{22} = [7.5,\ 4]^{\mathrm{T}}$$

$$\mu_3 = [7,\ 2]^{\mathrm{T}}$$

I. Gere um conjunto de dados **X** (treinamento) com 1000 pontos para ω_1 (sendo 500 de cada distribuição), 1000 pontos para ω_2 (sendo 500 de cada distribuição) e 500 pontos para ω_3 (use 0 no parâmetro seed para a inicialização do gerador de números aleatórios seguindo distribuição Gaussiana do Matlab). De forma similar, gere um conjunto de teste $\mathbf{X}_{\text{teste}}$ (use 100 no parâmetro seed para esse conjunto). Plote os resultados.
II. Implemente e visualize uma árvore de decisão (Decision Tree) usando o conjunto **X**.
III. Calcule os erros de classificação de treinamento e de teste e comente os resultados.
IV. Pode a árvore nos níveis 0 (sem poda), 1, ..., 11. Para cada árvore podada, calcule o erro de classificação para o conjunto de teste.
V. Plote os erros de classificação e os níveis de poda e aponte qual o nível que retorna o melhor desempenho de classificação. Que conclusões pode-se realizar ao se analisar esse gráfico?
VI. Visualize as árvores de decisão (a do item II e a do melhor resultado do item V).

Dica: No Matlab (Statistics and Machine Learning Toolbox), é possível utilizar as funções “classregtree”, “view”, “prune” e “eval” para gerar uma árvore de decisão, visualizar, podar e avaliar a performance, respectivamente. Em uma versão mais nova do MATLAB (2017-diante), utilize as funções correlatas ao fitctree. No OCTAVE, utilize as funções da toolbox M5PrimeLab: http://www.cs.rtu.lv/jekabsons/Files/M5PrimeLab.pdf