

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

Tópicos: Classificadores Não-Lineares.

Parte 1 – Exercícios Teóricos:

1. Considere o mapeamento do espaço de entrada para um espaço de maior dimensionalidade dado por:

$$x \in \mathcal{R} \rightarrow y \equiv \phi(x) \in \mathcal{R}^{2k+1}$$

sendo

$$\phi(x) = \left[\frac{1}{\sqrt{2}}, \cos x, \cos 2x, \dots, \cos kx, \sin x, \sin 2x, \dots, \sin kx \right]^T$$

Mostre que o produto interno é dado por:

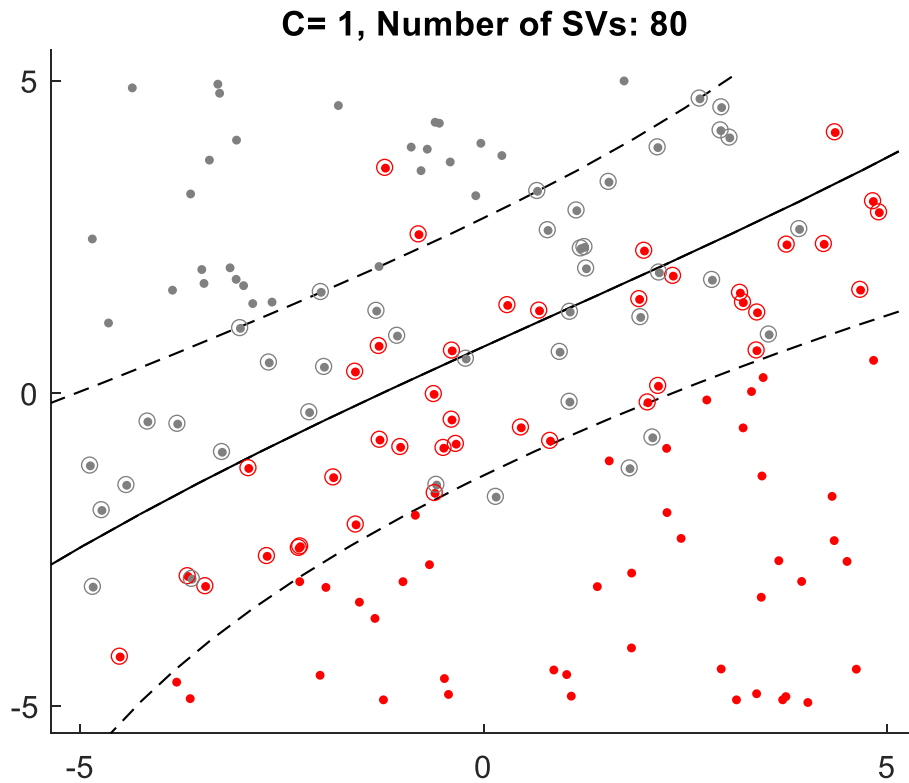
$$\begin{aligned} y_i^T y_j &= K(x_i, x_j) \\ &= \frac{\sin\left(\left(k + \frac{1}{2}\right)(x_i - x_j)\right)}{2 \sin\left(\frac{x_i - x_j}{2}\right)} \end{aligned}$$

Parte 2 – Exercícios Práticos:

1. Este problema é dividido em cinco etapas, listadas a seguir:
 - I. Gere um conjunto de dados bidimensionais **X1** (treinamento) conforme segue. Selecione 150 pontos em um espaço bidimensional na região definida por $[-5, 5] \times [-5, 5]$, seguindo uma distribuição uniforme (na função rand do matlab, selecione “seed” igual a zero). Rotule o ponto $\mathbf{x} = [x(1), x(2)]^T$ 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 **X1** (treinamento). Gere um conjunto adicional **X2** (teste) usando a mesma descrição de **X1** (na função rand do matlab, selecione “seed” igual a 100).
 - II. Construa um classificador SVM linear com os parâmetros $C = 2$ e $\text{tol} = 0.001$. 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 $C = 2$, $\text{tol} = 0.001$, $\sigma = 0,1$ 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. Repita o passo III usando um kernel polynomial dado por $(x^T y + \beta)^n$ para $(n, \beta) = (5, 0)$ e $(3, 1)$. Compare com o passo III e comente os resultados.

- V. Construa um classificador SVM não-linear com kernel RBF e os parâmetros $\sigma = 1,5$ e com kernel polinomial e os parâmetros $n = 3$ and $\beta = 1$. Em ambos os casos, use $\text{tol} = 0,001$ 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 μ_{11} , μ_{12} , Σ_{11} e Σ_{12} . De forma similar, os exemplos da classe ω_2 tem uma distribuição composta por duas Gaussianas com os parâmetros μ_{21} , μ_{22} , Σ_{21} e Σ_{22} . Já a classe ω_3 tem uma distribuição composta por uma Gaussianiana com os parâmetros μ_3 e Σ_3 . Os valores dos parâmetros estão representados abaixo:

$$\begin{aligned} \Sigma_{11} &= \begin{bmatrix} 0.2 & 0 \\ 0 & 2 \end{bmatrix} & \Sigma_{12} &= \begin{bmatrix} 3 & 0 \\ 0 & 0.5 \end{bmatrix} \\ \Sigma_{21} &= \begin{bmatrix} 5 & 0 \\ 0 & 0.5 \end{bmatrix} & \Sigma_{22} &= \begin{bmatrix} 7 & 0 \\ 0 & 0.5 \end{bmatrix} \\ \Sigma_3 &= \begin{bmatrix} 8 & 0 \\ 0 & 0.5 \end{bmatrix} & \mu_{11} &= [0, 3]^T \\ & & \mu_{12} &= [11, -2]^T \\ \mu_{21} &= [3, -2]^T & \mu_{22} &= [7.5, 4]^T \\ & & \mu_3 &= [7, 2]^T \end{aligned}$$

- I. Gere um conjunto de dados \mathbf{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 \mathbf{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>

3. Considere um problema de classificação bidimensional envolvendo duas classes ω_1 e ω_2 seguindo a distribuição das classes ω_1 e ω_2 do exercício anterior.
 - I. Gere um conjunto de dados \mathbf{X} (treinamento) com 200 pontos para ω_1 (sendo 100 de cada distribuição), 200 pontos para ω_2 (sendo 100 de cada distribuição). 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. Treine um modelo Random Forest no conjunto de treinamento e avalie o erro no conjunto de teste. Varie o número de árvores e verifique o impacto no resultado final.

Dica: No Matlab (Statistics and Machine Learning Toolbox) existe uma função para implementar Random Forest. Pesquise no Help para identificá-la. No OCTAVE, utilize as funções da toolbox M5PrimeLab.