

## Lista 6

### Support Vector Machines

#### Instruções

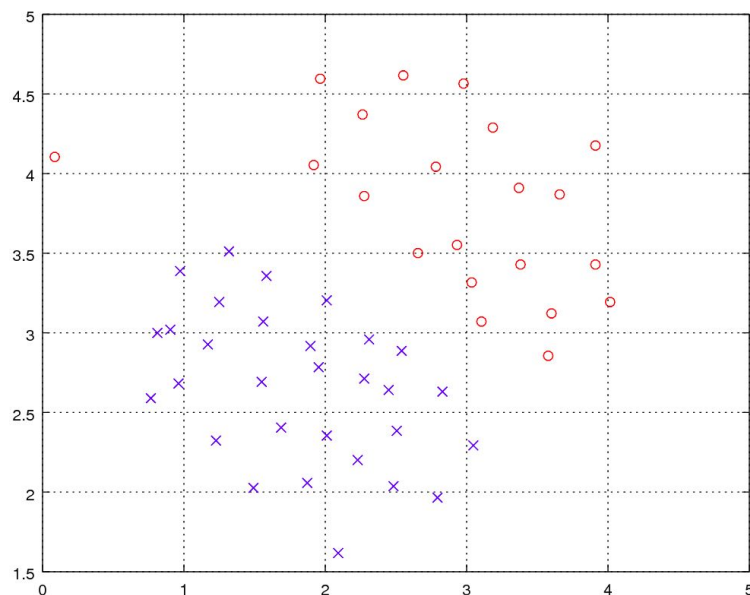
Deverá ser enviado ao professor, um arquivo texto contendo os gráficos, resultados e comentários requeridos em cada item.

#### 1. Classificação usando SVM e ajuste do parâmetro C

- Carregue os dados contidos no arquivo `ex6data1.data`.

O arquivo contém uma matriz e um vetor de dados. A matriz  $X$  é composta de 51 linhas e 2 colunas e representa um conjunto de dados de dimensão 2. O vetor  $y$  dá a classe a qual pertence cada vetor. Este exemplo consiste em um problema de duas classes.

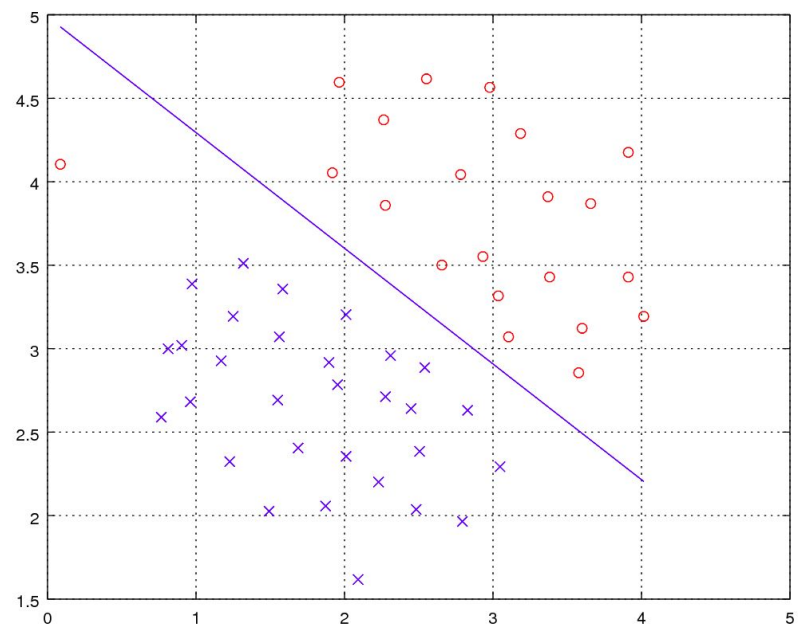
**Apresentar:** Figura com o conjunto de dados



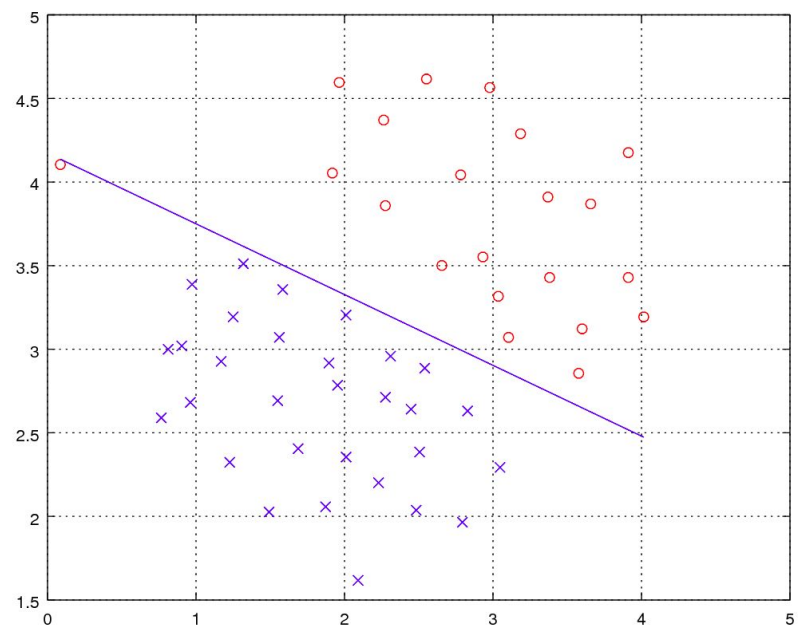
- Utilize a função `svmTrain` para treinar um modelo. Para esta função utilize o Kernel linear. Utilize valores de  $C = 1$  e  $C = 100$ .

- Plote o resultado utilizando a função `visualizeBoundaryLinear`. **Apresentar:** Figuras com o conjunto de dados e as superfícies de separação

$C = 1$



$C=100$



**Comentários:** Comente sobre as superfícies de separação obtidas para os dois valores de  $C$

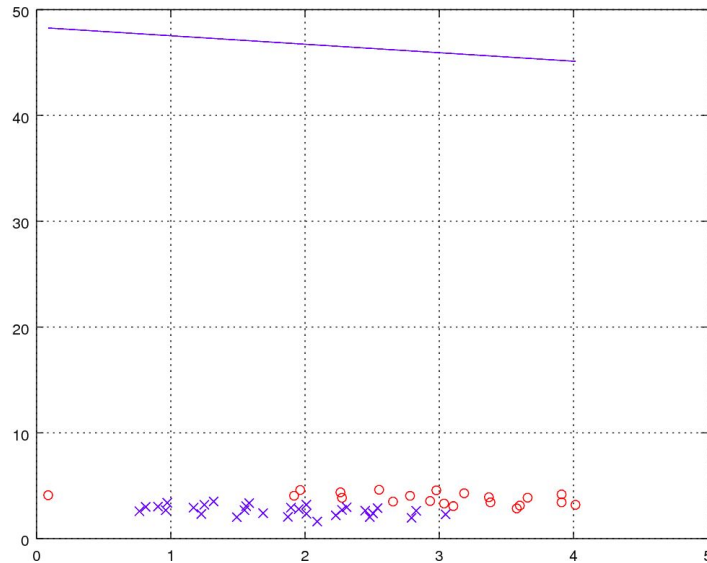
Para  $C=1$ , a penalidade para o erro é menor. Consequentemente, o overfitting diminui. Para  $C=100$ , a penalização para o erro é bem maior, o que faz com que o modelo fique sobre-ajustado para o conjunto de treino (que se evidencia ao vermos a inclinação da reta para incluir o outlier do canto superior esquerdo no gráfico).

- Utilize  $C=0.001$  e refaça o experimento.

**Apresentar:** Os valores de  $w$

Pesos: **0.017904 e 0.022330**

Gráfico:



**Comentários:** Comente sobre os valores de  $w$  obtidos

Bem baixos. A penalização pelo erro é quase nula, logo algoritmo praticamente não regularizou os pesos com os erros a cada passo de treinamento.

- Altere a classe do elemento 37 do conjunto de dados. Para isso, faça  $y(37) = 1$ . Treine o modelo utilizando  $C = 1e9$

**Apresentar:** Figuras com o conjunto de dados e a superfície de separação.

Nops...

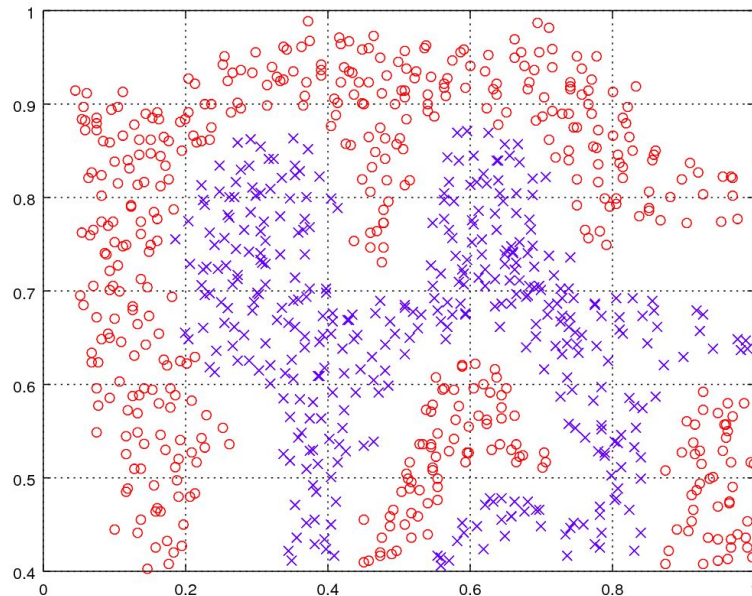
**Comentários:** Comente sobre o tempo de convergência do método. Infinito. Era impossível de convergir já que os dados ficaram linearmente inseparáveis. Por maior que fosse  $C$ , penalizando muito o erro, é impossível de traçar uma reta que divida perfeitamente tal conjunto.

## 2. Classificação usando SVM e Kernel RBF

- Carregue os dados contidos no arquivo `ex6data2.data`.

O arquivo contém uma matriz e um vetor de dados. A matriz  $X$  é composta de 863 linhas e 2 colunas e representa um conjunto de dados de dimensão 2. O vetor  $y$  dá a classe a qual pertence cada vetor. Este exemplo consiste em um problema de duas classes.

**Apresentar:** Figura com o conjunto de dados



**Comentários:** Comente sobre qual tipo de Kernel deve ser utilizado neste problema.

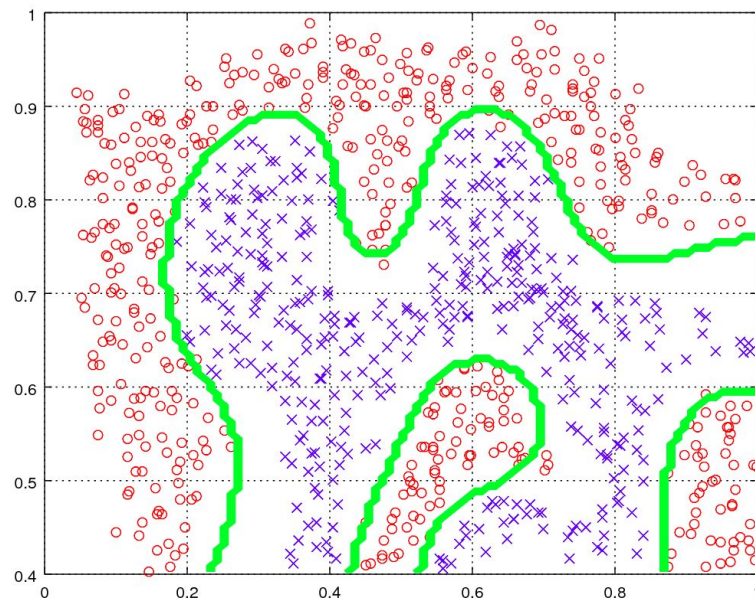
Com certeza um não-linear. Acredito que algum com polinômio de grau bem alto ou um kernel gaussiano.

- Utilize a função `svmTrain` para treinar um modelo. Para esta função utilize o Kernel RBF. Utilize valores de  $\sigma = 0.1$  e  $\sigma = 0.2$ . Utilize  $C = 1$ .

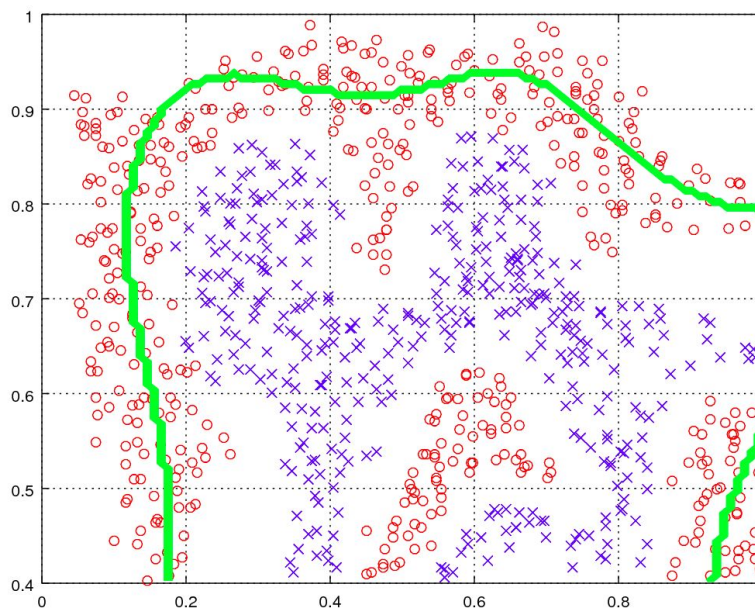
- Plote o resultado utilizando a função `visualizeBoundary`.

**Apresentar:** Figuras com o conjunto de dados e as superfícies de separação.

$\sigma = 0.1$



$\sigma = 0.2$



**Comentários:** Comente sobre as superfícies de separação obtidas para os dois valores de sigma

Bem diferentes. Aparentemente, o sigma é um parâmetro bem sensível e afeta na no ajuste dos pesos. Com o sigma ligeiramente maior, a superfície de separação ficou menos enviesada ao conjunto de teste.