

Lista 6
Support Vector Machines
Instruções

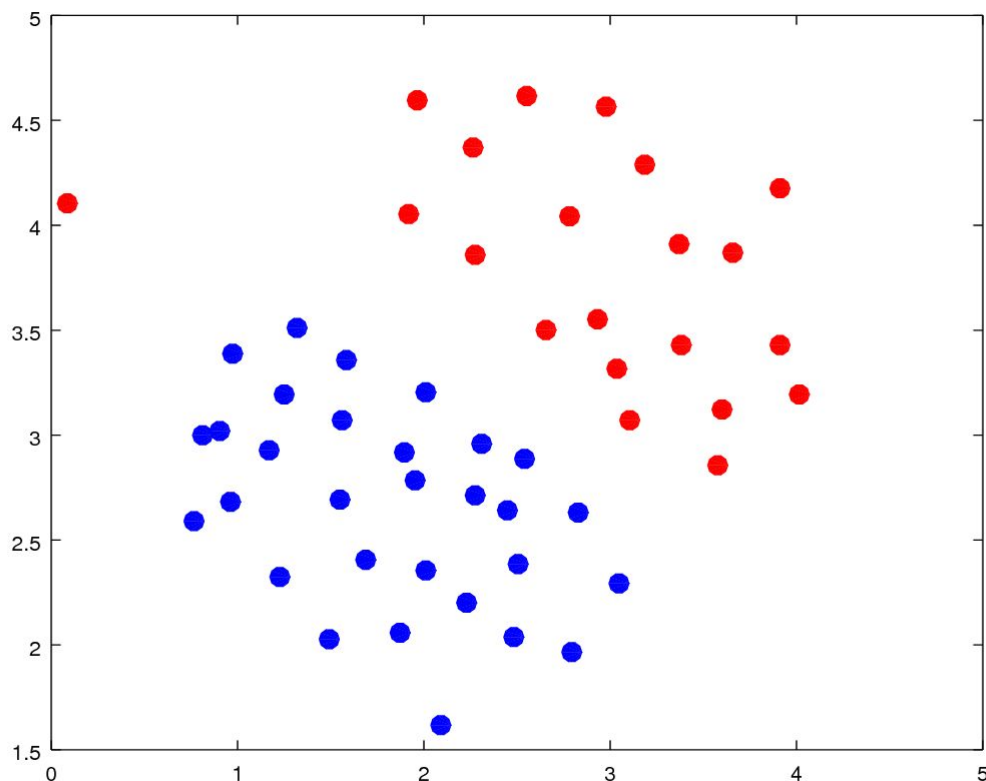
Deverá ser enviado ao professor, um arquivo texto contendo os gráficos, resultados e comentários requeridos em cada item.

1. Classificação usando SVM e ajuste do parâmetro C

- Carregue os dados contidos no arquivo ex6data1.data.

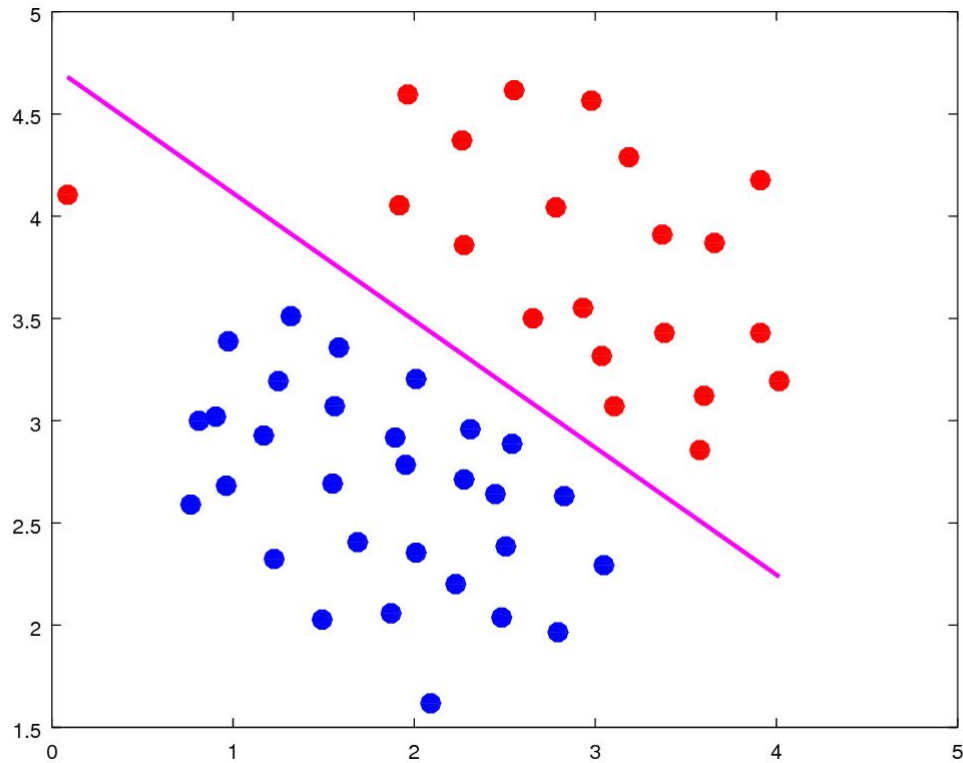
O arquivo contém uma matriz e um vetor de dados. A matriz X é composta de 51 linhas e 2 colunas e representa um conjunto de dados de dimensão 2. O vetor y dá a classe a qual pertence cada vetor. Este exemplo consiste em um problema de duas classes.

Apresentar: Figura com o conjunto de dados

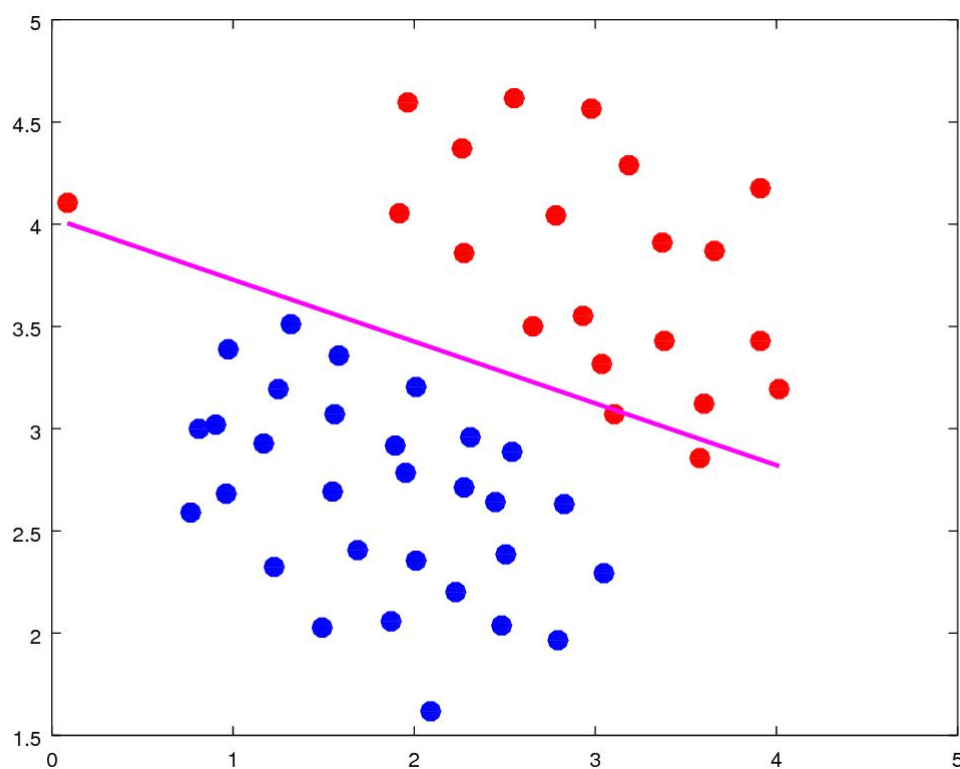


- Utilize a função `svmTrain` para treinar um modelo. Para esta função utilize o Kernel linear. Utilize valores de $C = 1$ e $C = 100$.
- Plote o resultado utilizando a função `visualizeBoundaryLinear`. **Apresentar:** Figuras com o conjunto de dados e as superfícies de separação

Com $C=1$



Com $C=100$



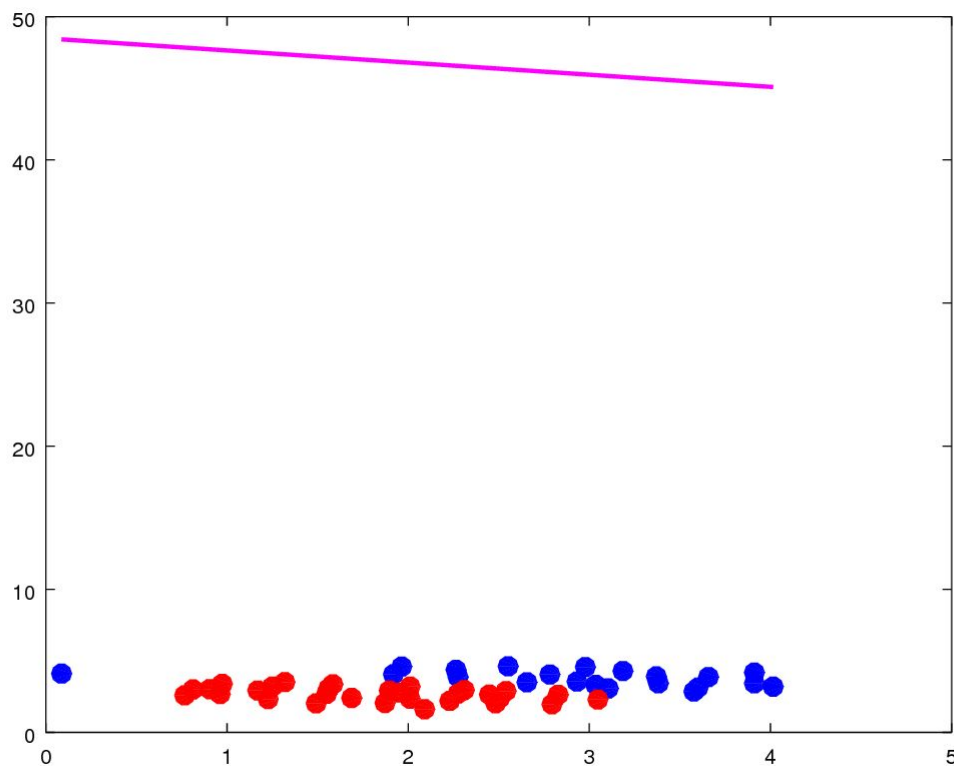
Comentários: Comente sobre as superfícies de separação obtidas para os dois valores de C

O parâmetro C permite uma regularização sobre a convergência do modelo. Com o valor muito elevado, o custo da folga concedida se torna muito elevada, fazendo com que o algoritmo tente evitar usar o valor do erro e considere apenas a distância máxima entre a reta e os elementos. Aquele elemento que estava isolado poderia ser apenas um outlier, e não significa que os elementos do conjunto vermelho estejam espalhados naquela direção. Com o valor de $C = 1$ a folga se torna maior, permitindo que alguns elementos passem para o outro lado da reta, desde que isso diminua o erro total, tornando o modelo mais genérico.

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

Apresentar: Os valores de w

Comentários: Comente sobre os valores de w obtidos



$$w = [0.018560, 0.022392]$$

Com o valor de C muito baixo, para maximizar a função de custo se torna necessário que o erro seja muito grande. O custo da folga do algoritmo é muito baixo, e permite que a reta se afaste demais do valor desejado. O parâmetro C deve ser usado com cautela para evitar overfitting permitindo que outliers sejam ignorados, mas ao ser usado da forma que foi pode quebrar totalmente a lógica do algoritmo, se tornando um algoritmo de maximização de erros.

- 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

Comentários: Comente sobre o tempo de convergência do método.

Deixei rodando por algumas horas e não convergiu. Ai eu desisti e não coloquei uma imagem associada.

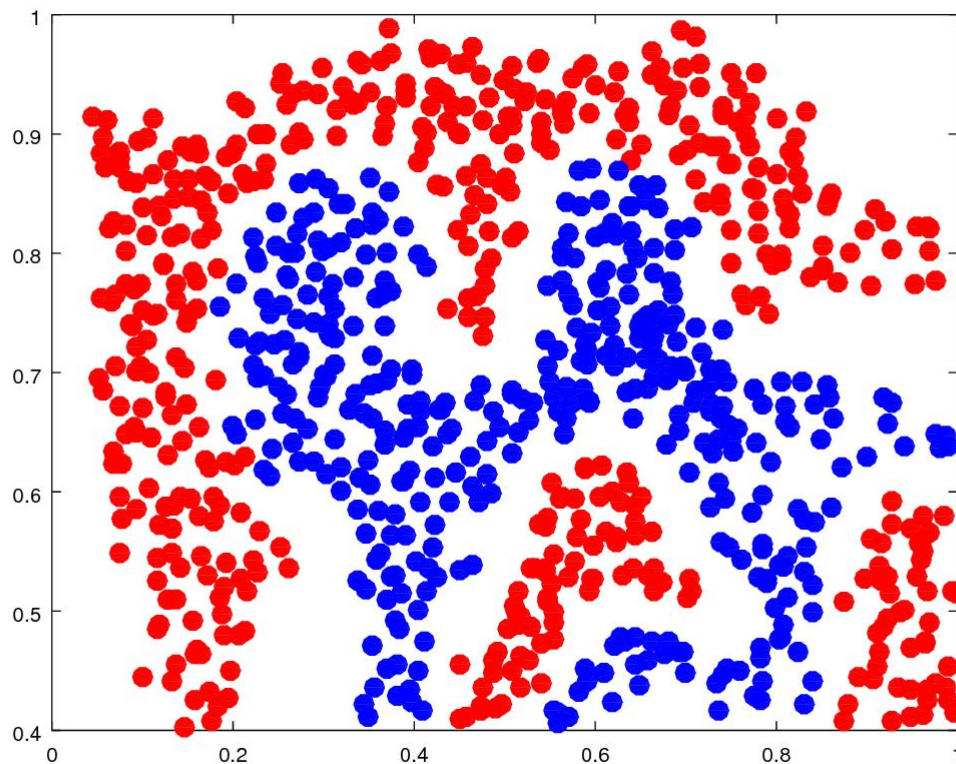
O valor elevado de C pode estar afetando a convergência do método. Talvez o tamanho do passo do algoritmo de otimização esteja muito grande, e a tolerância desproporcionalmente pequena, fazendo com que o algoritmo nunca chegue em um resultado.

1. 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

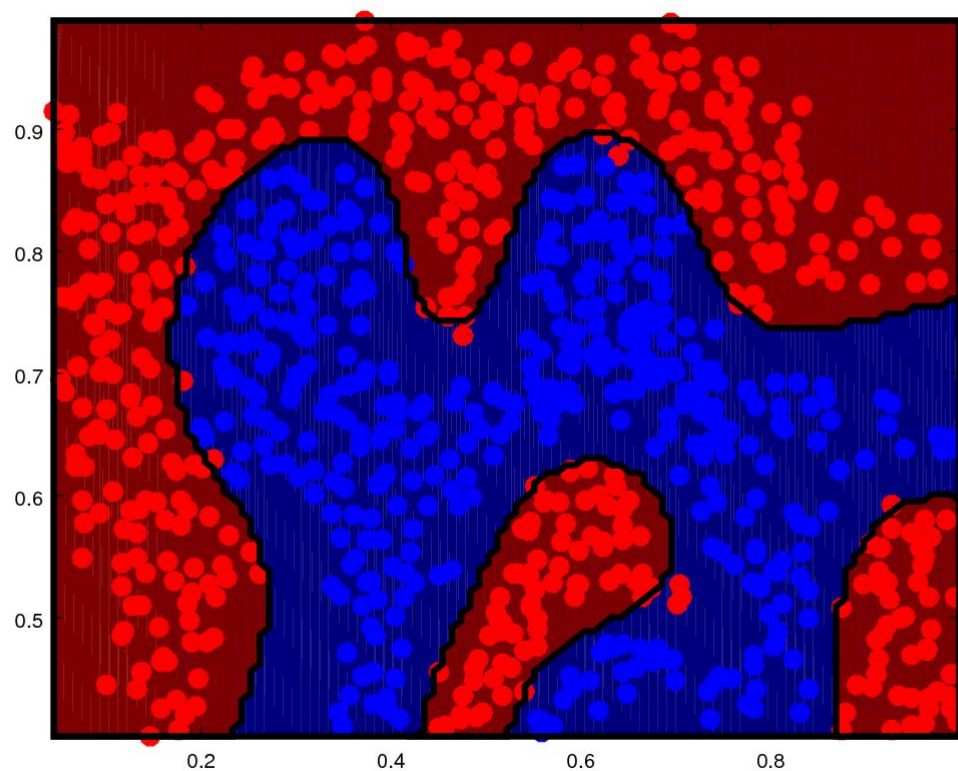


Comentários: Comente sobre qual tipo de Kernel deve ser utilizado neste problema.

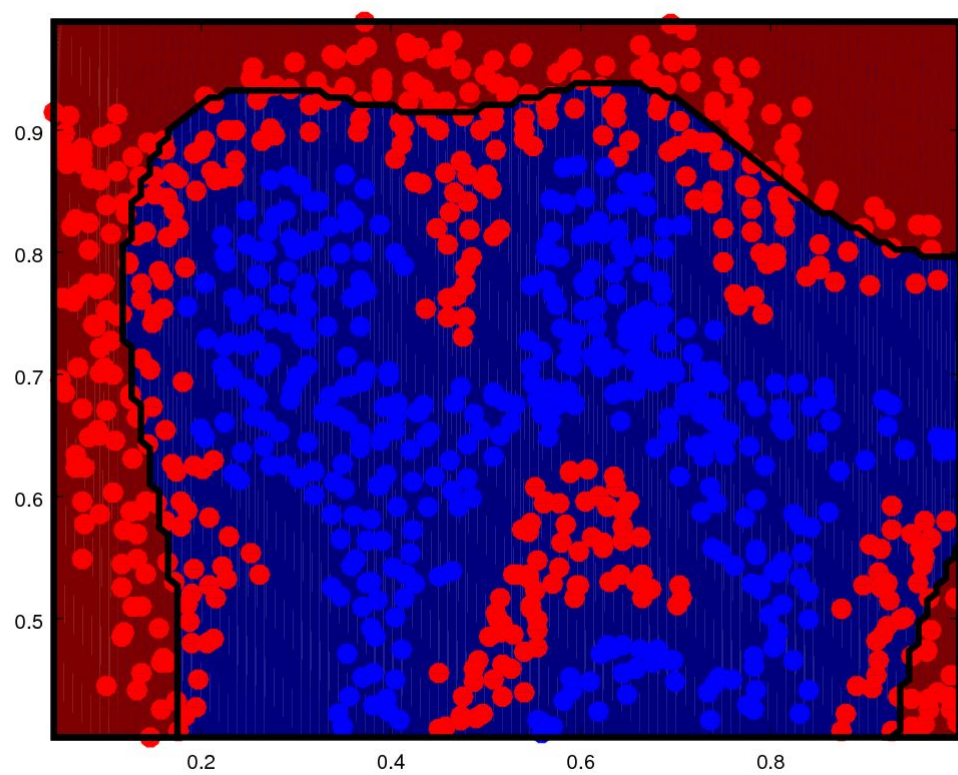
Com certeza não é o kernel linear. O jeito é tentar usar o kernel gaussiano.

- 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

Para $\sigma = 0.1$



para $\sigma = 0.2$



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

Ao mudar o valor de sigma a função gaussiana se torna mais aberta, menos restrita, diminuindo o overfitting.