Fundamentos de Aprendizagem de Máquina

Hélio Pio

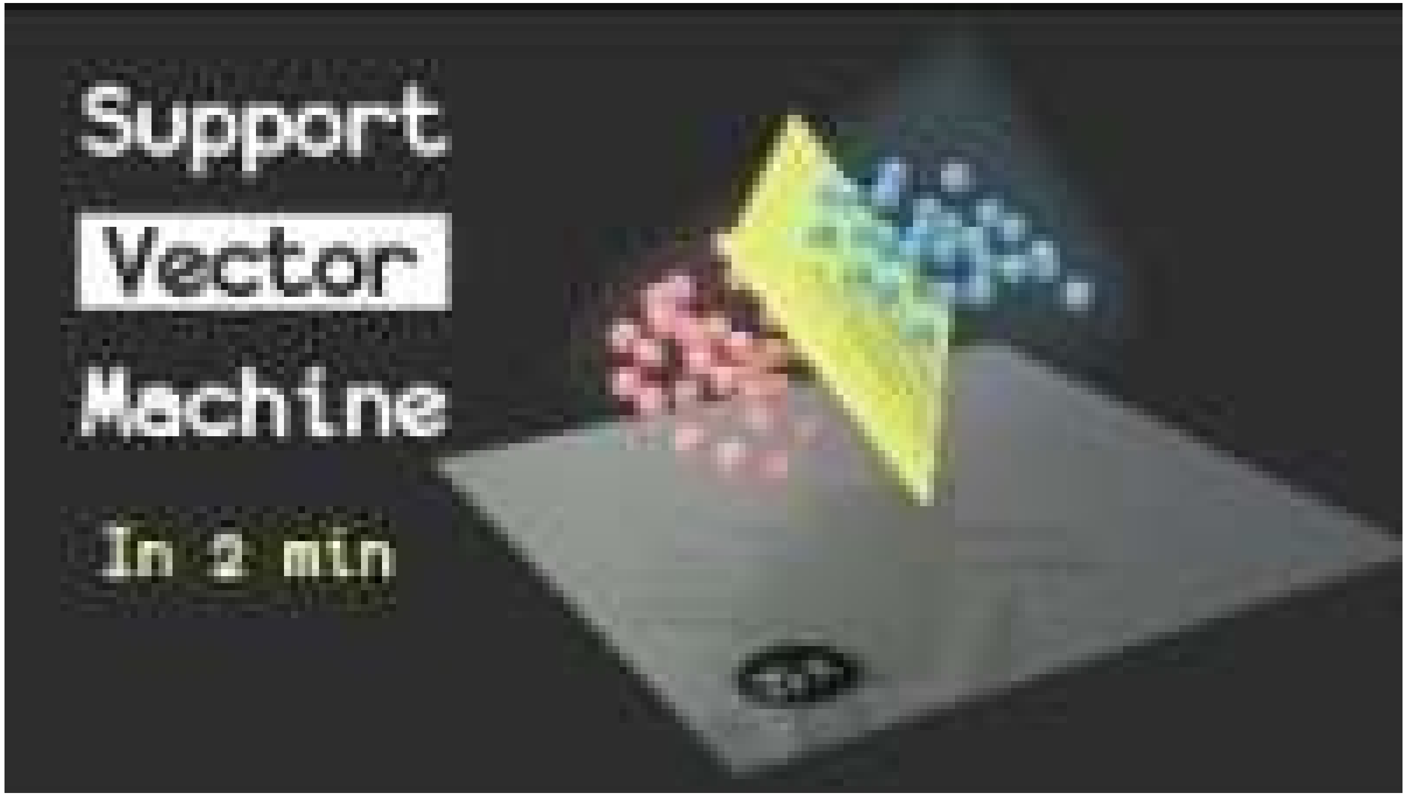
Programação das Aulas

Tópico 1: Introdução a Inteligência Artificial
Tópico 2: Agentes Inteligentes
Tópico 3: Fundamentos de Aprendizagem de Máquina
Tópico 4: Redes Neurais Artificiais
Tópico 5: Atividade em Aula – Primeira Avaliação
Tópico 6: Representação da Incerteza e Lógica Fuzzy
Tópico 7: Redes Bayesianas
Tópico 8: Support Vector Machines
Tópico 9: Atividade em Aula – Segunda Avaliação
Tópico 10: Resolução de Problemas por Meio de Busca e Otimização
Tópico 11: Técnicas de Ensemble
Tópico 12: Atividade em Aula – Terceira Avaliação

O que é *Support Vector Machine*?

Support Vector Machine

O que é *Support Vector Machine*?



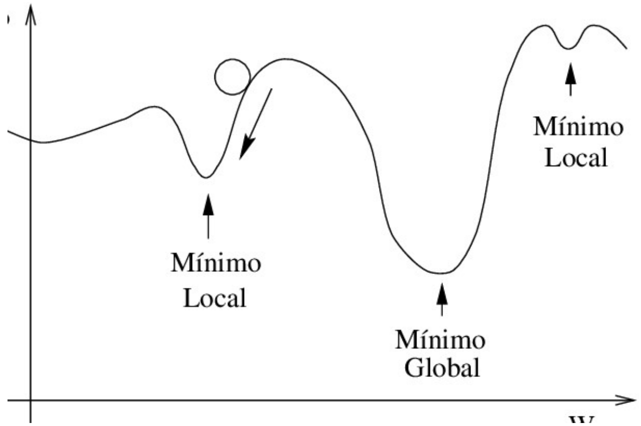
Support Vector Machine

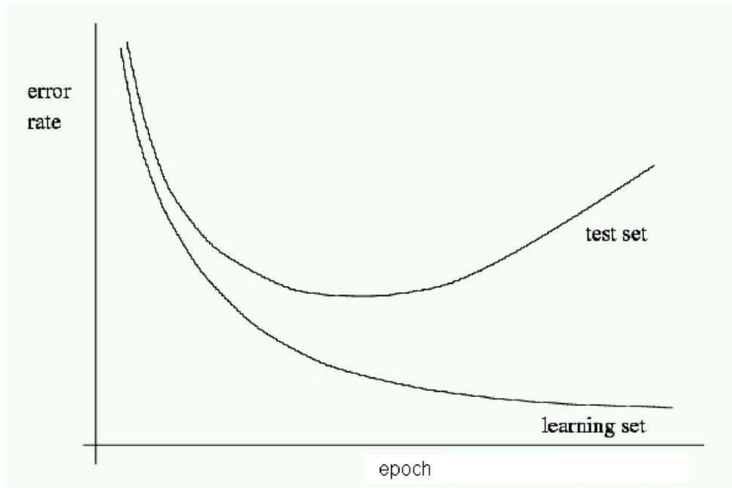
Problemas típicos de RNA

- Mínimo Local x Mínimo Global
  - Menor valor alcançado por uma função em uma vizinhança específica.
  - Menor valor absoluto da função em todo o seu domínio.

- Sobre-ajuste x Capacidade de Generalização.

- "Maldição da Dimensionalidade": Dimensão muito grande dos dados de entrada degrada o aprendizado de máquina.





Support Vector Machine

Problemas típicos de RNA

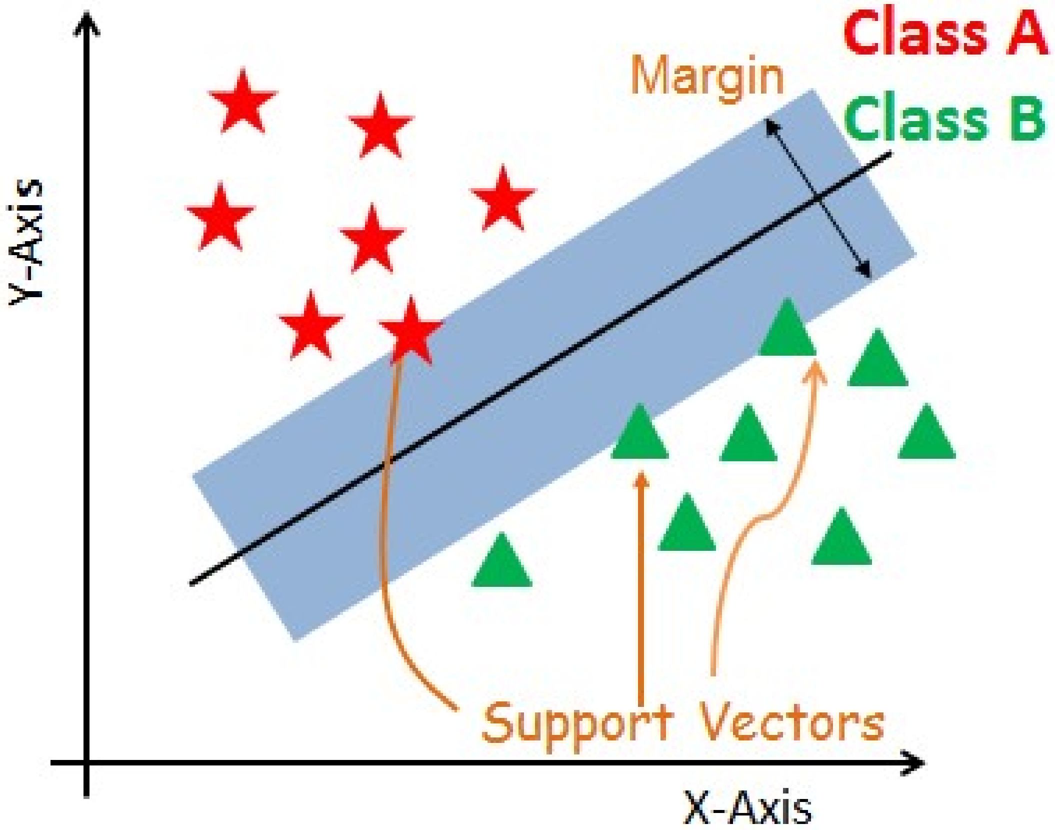
- Soluções possíveis:
  - Reduzir a Dimensionalidade: PCA, NLPCA
  - Aumentar a Dimensionalidade: SVM

AUMENTAR A DIMENSIONALIDADE ??

ISSO NÃO VAI AGRAVAR O PROBLEMA??

Support Vector Machine

Solução: SVM



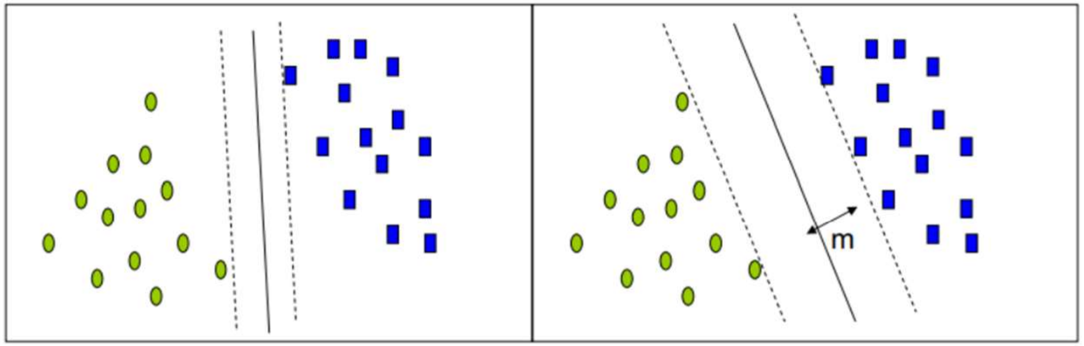
Support Vector Machine

Dados separados linearmente.

- Seja D(x1,y1), ... , (Xd, yd), onde X é o conjunto de tuplas de treinamento associadas à classe y.

- Existem infinitas linhas (hiperplanos) separando duas classes. O alvo é achar o melhor hiperplano que minimize o erro da classificação.

- SVM busca o hiperplano com a maior margem (*maximum marginal hyperplane* – MMH).



Support Vector Machine

Dados separados linearmente.

- O SVM usa um mapeamento não linear para transformar os dados originais (treinamento) em um espaço de dimensão superior, onde é mais fácil encontrar um hiperplano de separação entre as classes.

- Na nova dimensão, SVM busca um hiperplano que realize uma separação linear ótima entre as classes.

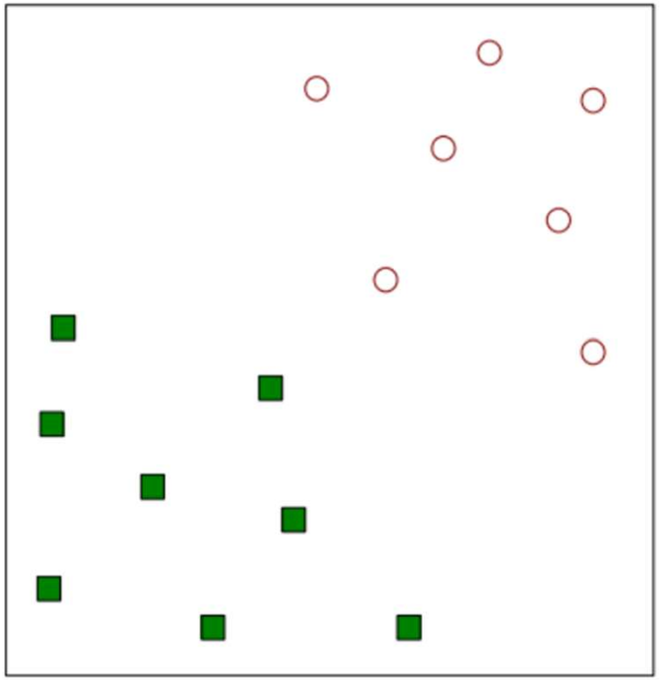
- SVM sempre separa duas classes por meio de um hiperplano com a maior distância entre as classes.

- SVM encontra o hiperplano usando vetores suportes (tuplas especiais no treinamento) e margens definidas pelos vetores suportes.

Support Vector Machine

Dados separados linearmente.

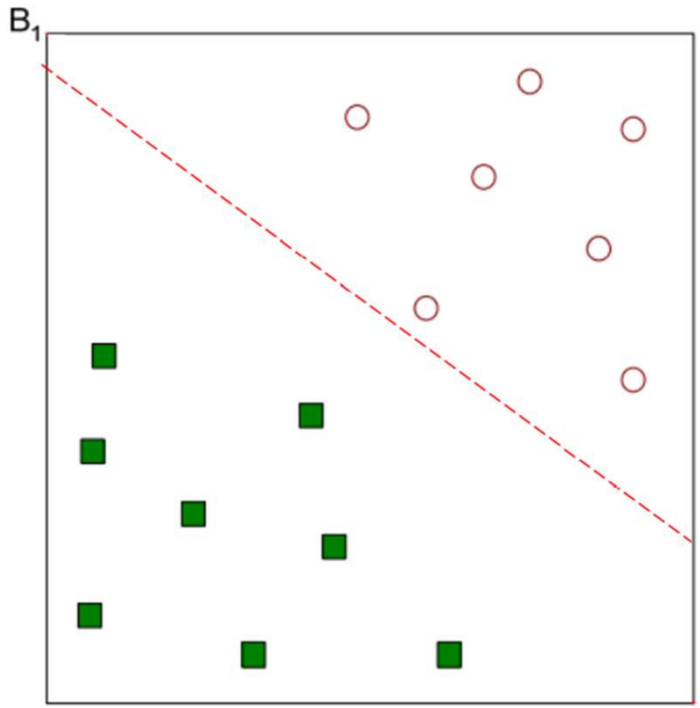
- Encontrar um hiperplano linear (limite de decisão) capaz de separar os dados.



Support Vector Machine

Dados separados linearmente.

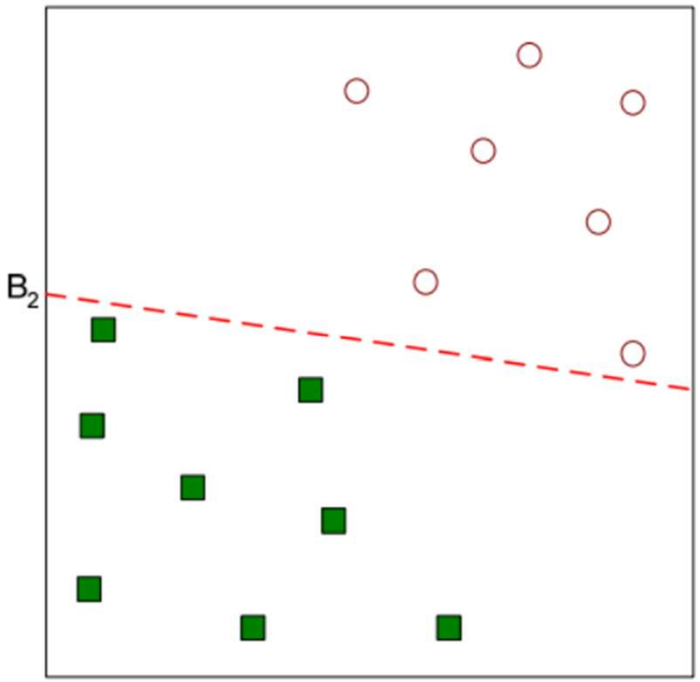
- Uma possível solução.



Support Vector Machine

Dados separados linearmente.

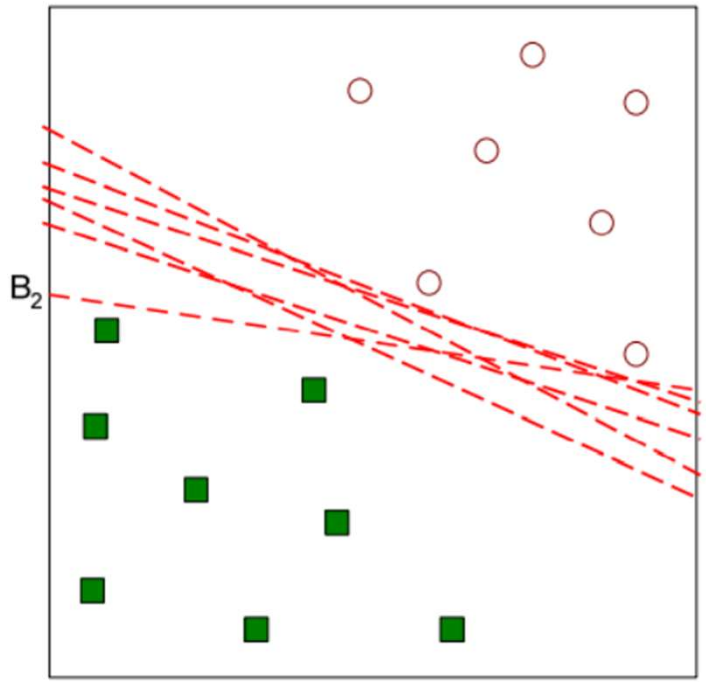
- Outra possível solução.



Support Vector Machine

Dados separados linearmente.

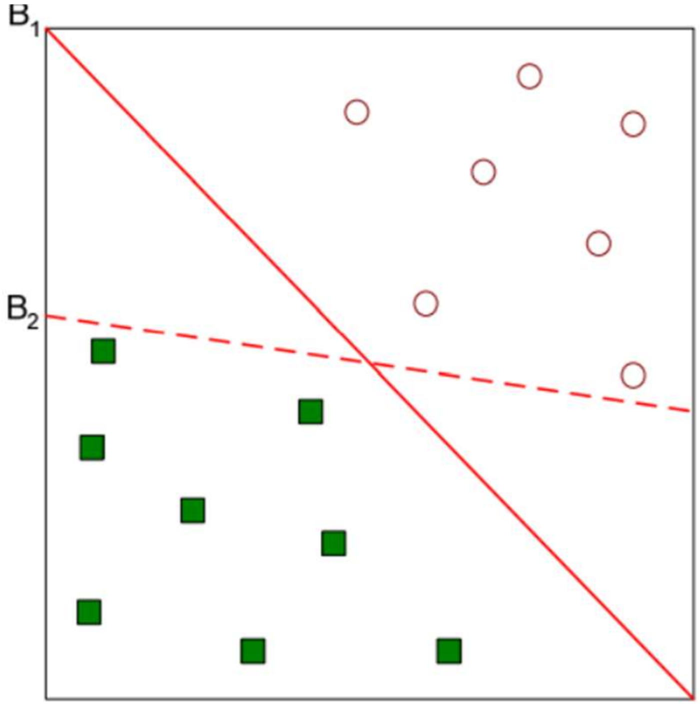
- Outras possíveis soluções.



Support Vector Machine

Dados separados linearmente.

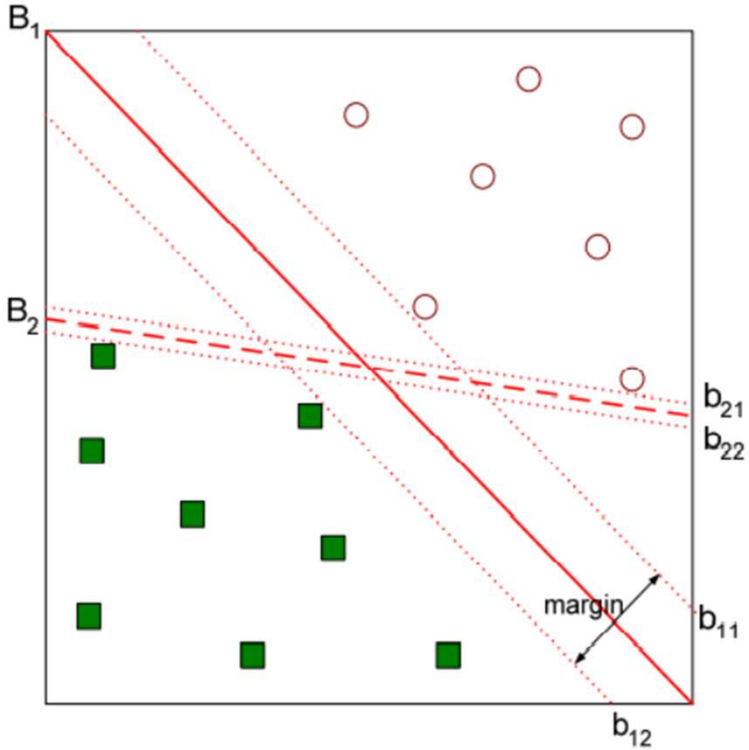
- Qual é o melhor hiperplano? B1 ou B2?



Support Vector Machine

Dados separados linearmente.

- Encontrar o hiperplano que maximize a margem.



Support Vector Machine

Dados separados linearmente.

Algoritmo Determinação do hiperplano ótimo para conjuntos linearmente separáveis (Vert, 2001).

1: Para cada conjunto de treinamento linearmente separável $S = \{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x}_n, y_n)\}$

2: Seja $\alpha^* = (\alpha_1^*, \ldots, \alpha_n^*)$ a solução do seguinte problema de otimização com restrições:

3: Maximizar: $\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \mathbf{x_i} \cdot \mathbf{x_j}$

4: Sob as restrições: $\begin{cases} \sum_{i=1}^{n} y_i \alpha_i = 0 \\ \alpha_i \geq 0, \; i = 1, \ldots, n \end{cases}$

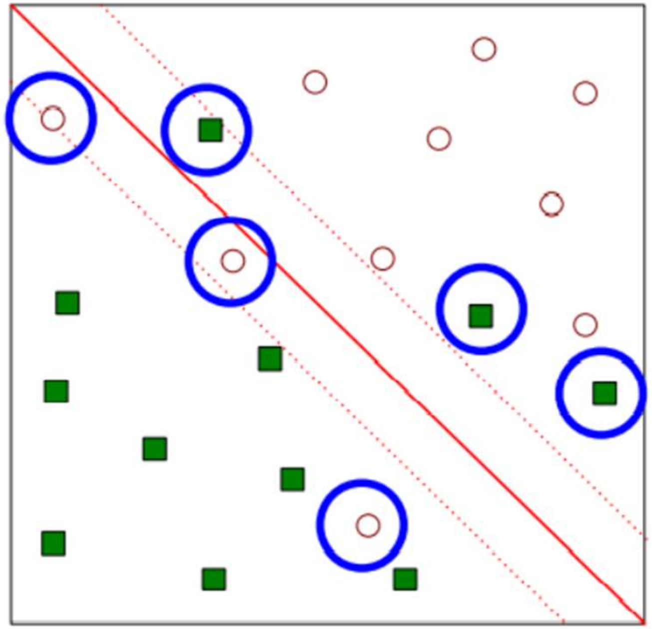
5: O par $(\mathbf{w}^*, b^*)$ apresentado a seguir define o hiperplano ótimo.

6: $\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i{}^* y_i \mathbf{x_i}$

7: $b^* = -{}^1\!/_2 \left[\max_{\{i|y_i=-1\}} (\mathbf{w}^* \cdot \mathbf{x_i}) + \min_{\{i|y_i=+1\}} (\mathbf{w}^* \cdot \mathbf{x_i}) \right]$

Support Vector Machine

E se o problema não for linearmente separável?

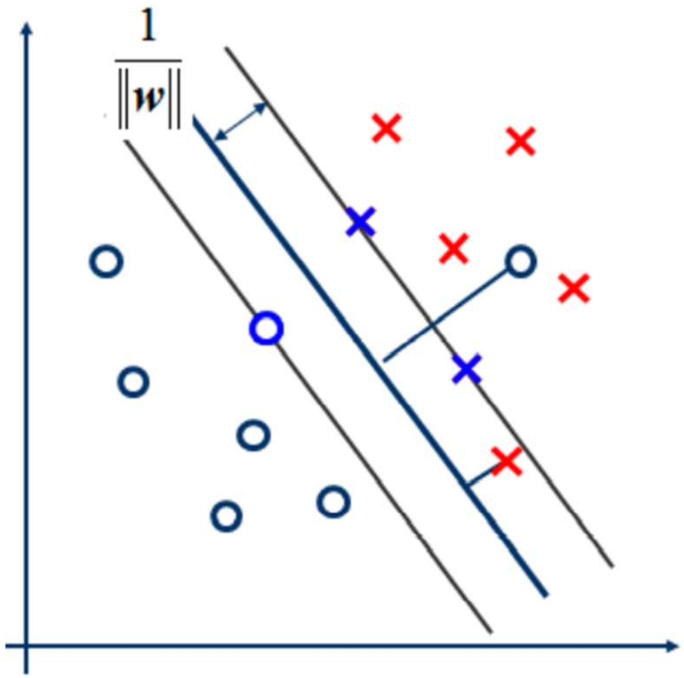


Support Vector Machine

E se o problema não for linearmente separável?

- O coeficiente de penalidade ‘C’ no SVM controla a importância de evitar erros de classificação no treinamento. Valores maiores enfatizam a classificação correta, enquanto menores permitem mais erros.

- O erro de classificação ‘ε’ é a diferença entre a saída prevista e a classe verdadeira do dado, e o SVM busca minimizar essa soma para encontrar o hiperplano de separação ótima.



Support Vector Machine

E se o problema não for linearmente separável?

Algoritmo Determinação do hiperplano ótimo para conjuntos de treinamento gerais (Vert, 2001).

1: Para cada conjunto de treinamento $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$

2: Seja $\alpha^* = (\alpha_1^*, \ldots, \alpha_n^*)$ a solução do seguinte problema de otimização com restrições:

3: Maximizar: $\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \mathbf{x_i} \cdot \mathbf{x_j}$

4: Sob as restrições: $\begin{cases} \sum_{i=1}^{n} y_i \alpha_i = 0 \\ 0 \leq \alpha_i \leq C,\ i = 1, \ldots, n \end{cases}$

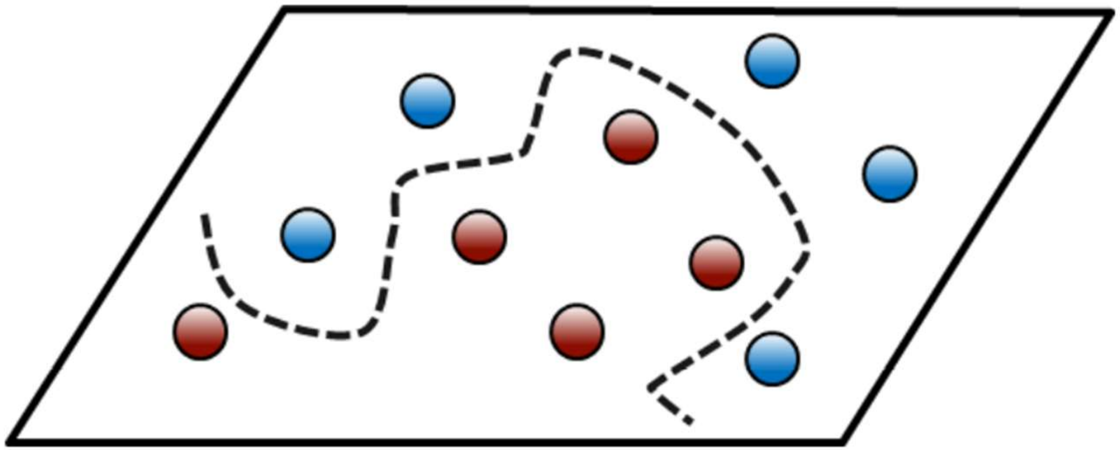
5: O par $(\mathbf{w}^*, b^*)$ apresentado a seguir define o hiperplano ótimo.

6: $\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x_i}$

7: $b^* = -{}^1\!/\!_2 \left[\max_{\{i | y_i = -1\}} (\mathbf{w}^* \cdot \mathbf{x_i}) + \min_{\{i | y_i = +1\}} (\mathbf{w}^* \cdot \mathbf{x_i}) \right]$

Support Vector Machine

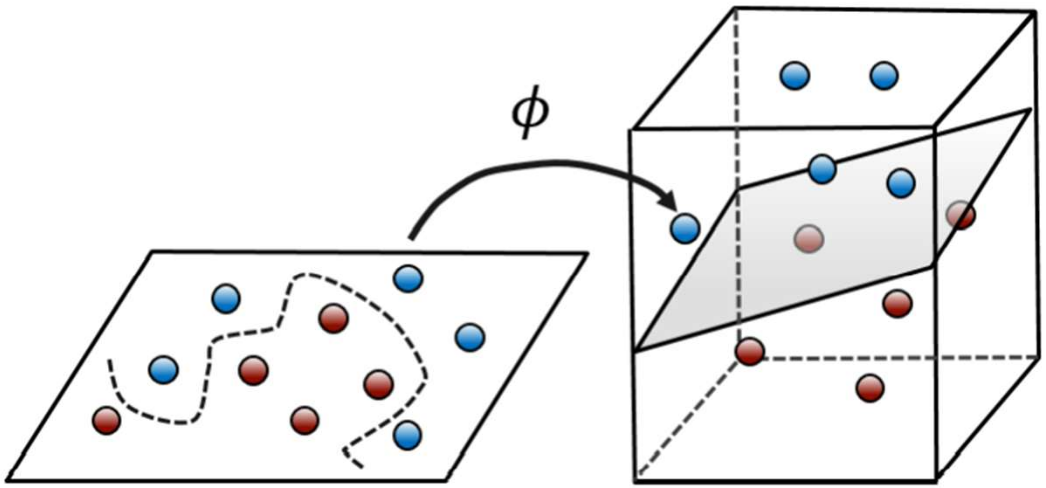
E se o limite de decisão não for linear?



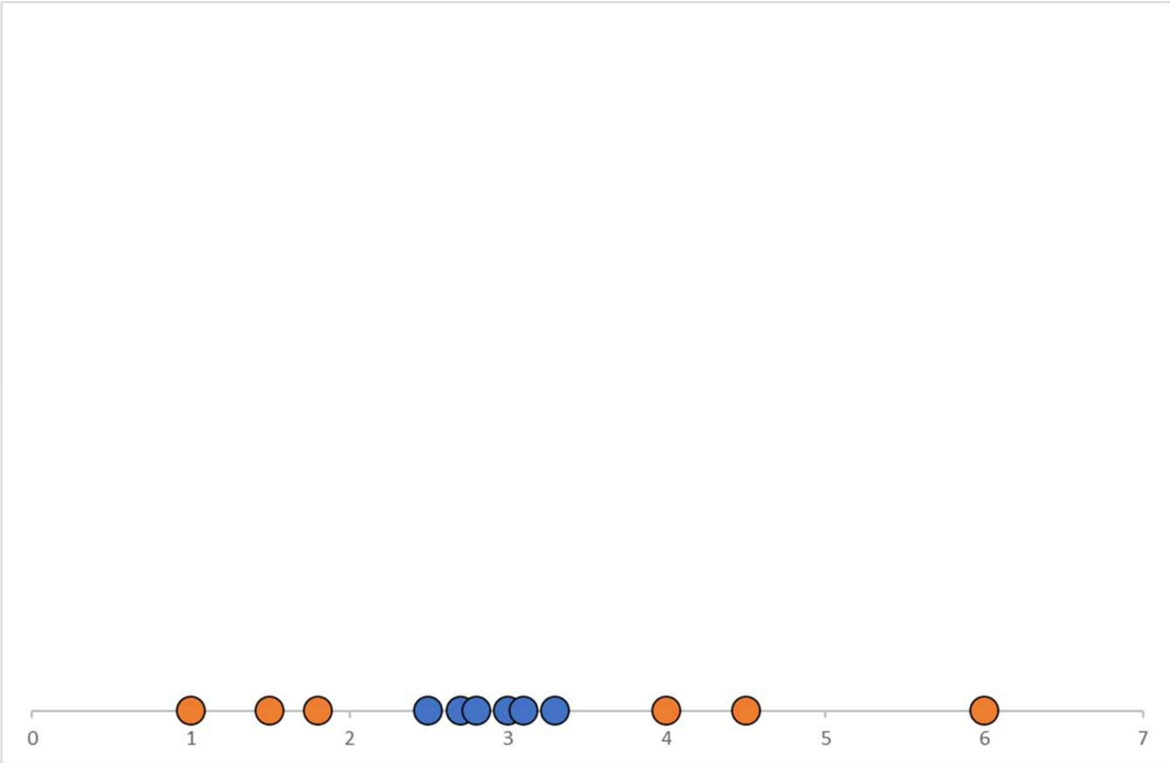
Support Vector Machine

E se o limite de decisão não for linear?

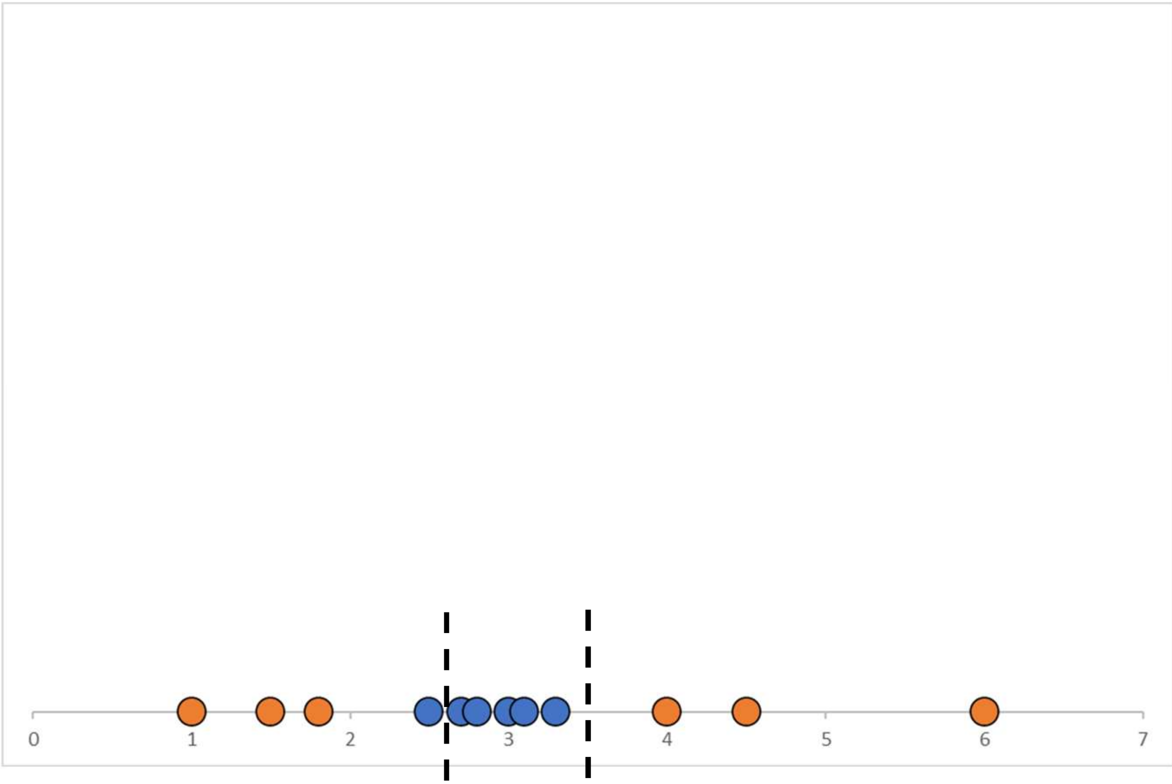
- Transformar os dados originais (treinamento) para uma dimensão maior.



Support Vector Machine

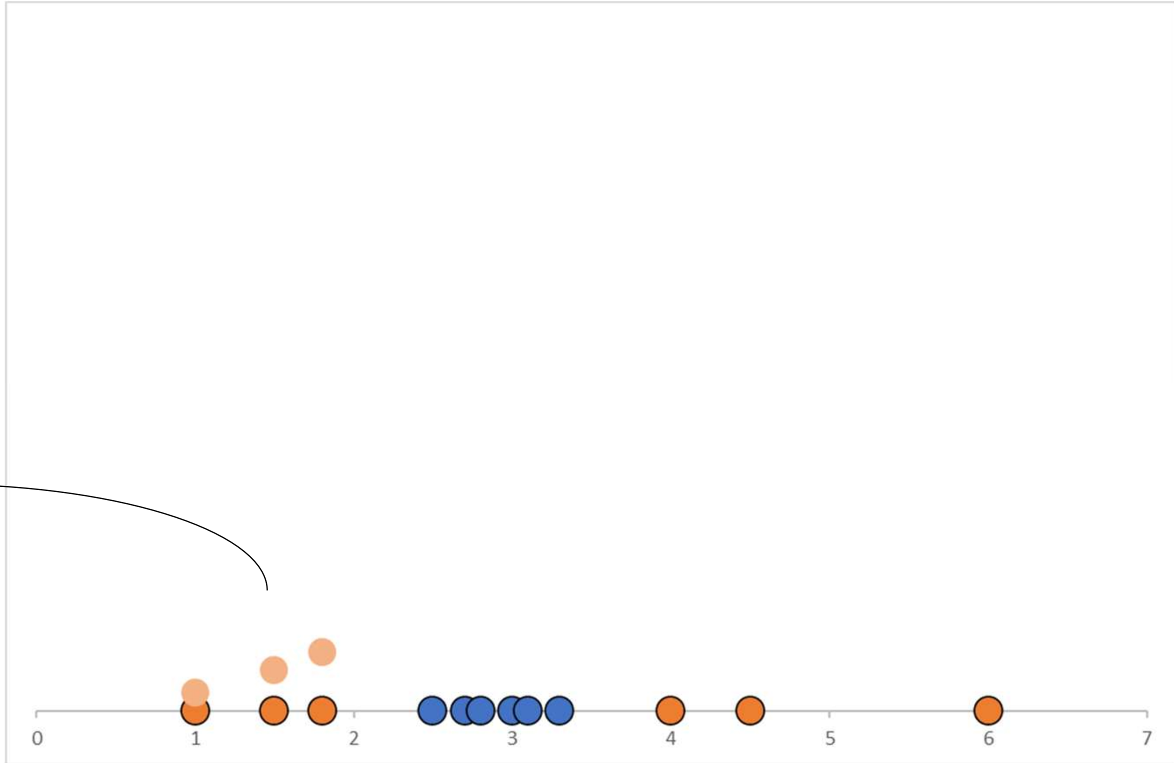


Support Vector Machine

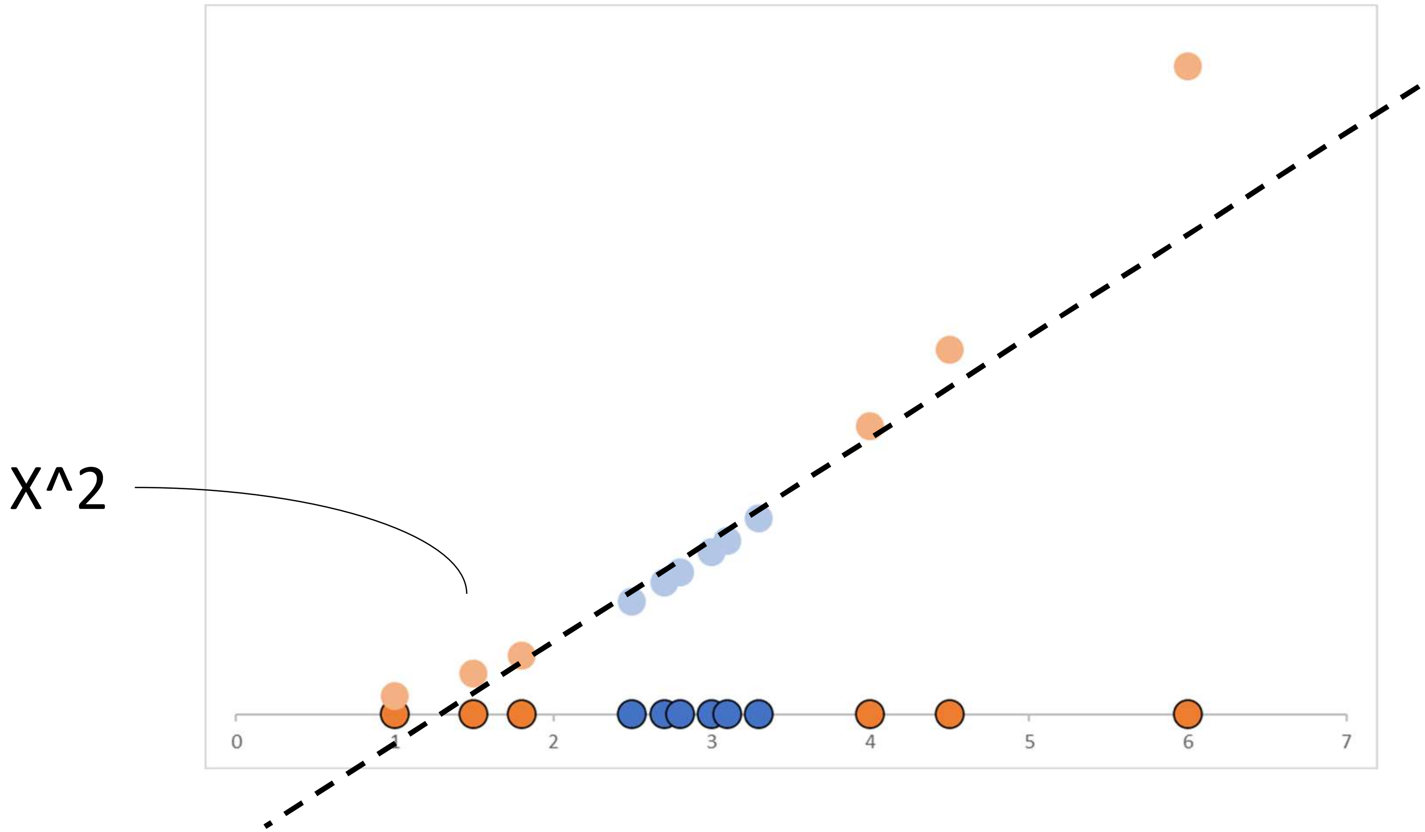


Support Vector Machine

X^2



Support Vector Machine



Support Vector Machine

E se o limite de decisão não for linear?

- Função Kernel

- Kernel polinomial pode ser escrito como:
- $K(X_i, X_j) = (X_i \cdot X_j + 1)^p$
- Kernel gaussiano pode ser escrito como:
- $K(X_i, X_j) = e^{-\frac{1}{2\sigma^2}(X_i - X_j)^2}$
- Kernel RBF pode ser escrito como:
- $K(X_i, X_j) = e^{-\gamma(X_i - X_j)^2}$
- Kernel sigmoid pode ser escrito como:
- $K(X_i, X_j) = tanh(\eta X_i \cdot X_j + \nu)$

Support Vector Machine

E se o limite de decisão não for linear?

- Função Kernel

Algoritmo	Determinação do hiperplano ótimo no espaço de características (Vert, 2001).

1: Para qualquer conjunto de treinamento $\mathbf{\Phi}(S) = \{(\mathbf{\Phi}(\mathbf{x}_1), y_1), \ldots, (\mathbf{\Phi}(\mathbf{x}_n), y_n)\}$

2: Seja $\alpha^* = (\alpha_1^*, \ldots, \alpha_n^*)$ a solução do seguinte problema de otimização com restrições:

3: Maximizar: $\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \mathbf{\Phi}(\mathbf{x_i}) \cdot \mathbf{\Phi}(\mathbf{x_j})$

4: Sob as restrições: $\begin{cases} \sum_{i=1}^{n} y_i \alpha_i = 0 \\ 0 \leq \alpha_i \leq C, \; i = 1, \; \ldots, \; n \end{cases}$

5: O par $(\mathbf{w}^*, b^*)$ apresentado a seguir define o hiperplano ótimo.

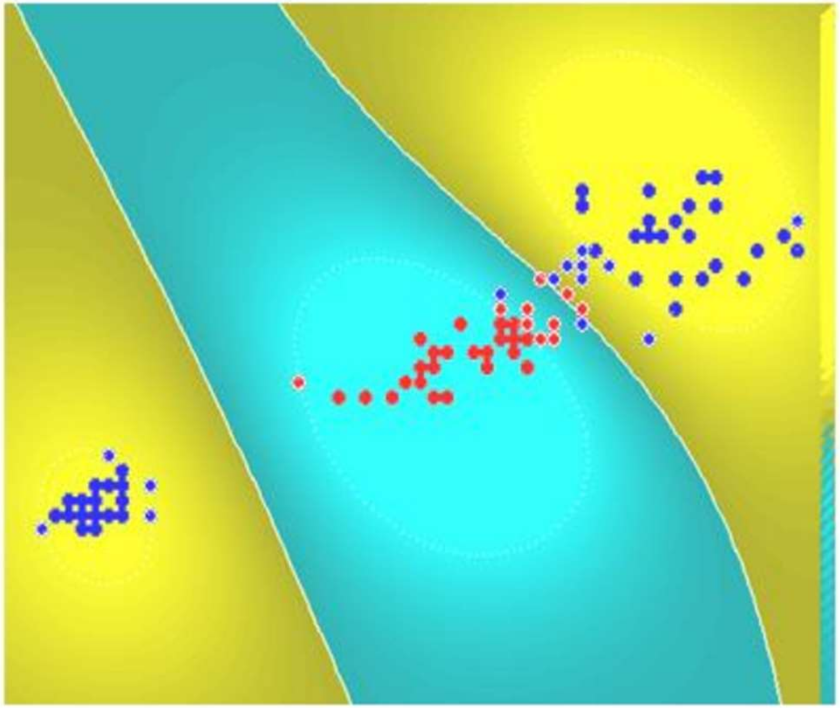
6: $\mathbf{w}^* \leftarrow \sum_{i=1}^{n} \alpha_i^* y_i \mathbf{\Phi}(\mathbf{x_i})$

7: $b^* \leftarrow -\frac{1}{2} \left[\min_{\{i|y_i=+1\}} (\mathbf{w}^* \cdot \mathbf{\Phi}(\mathbf{x_i})) + \max_{\{i|y_i=-1\}} (\mathbf{w}^* \cdot \mathbf{\Phi}(\mathbf{x_i})) \right]$

Support Vector Machine

Resultados:

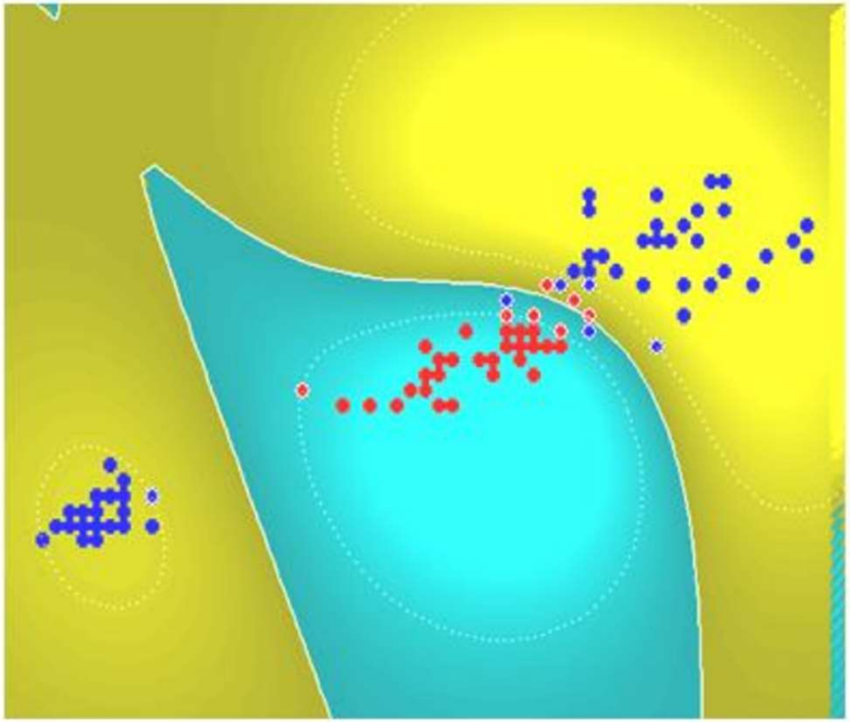
- Dados Iris
  - Kernel RBF, s = 1
  - C = 1



Support Vector Machine

Resultados:

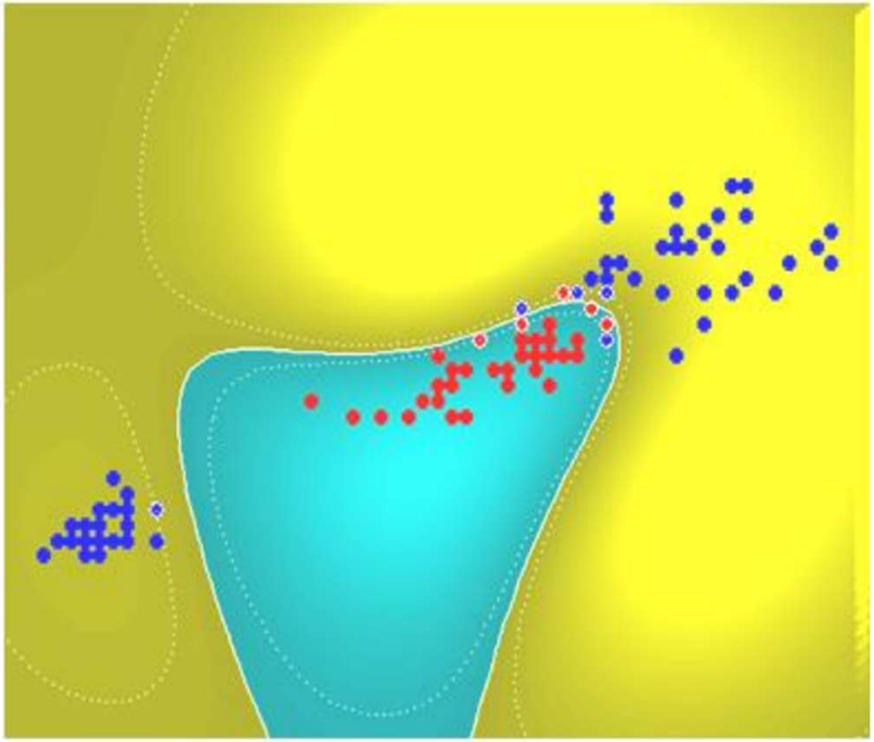
- Dados Iris
  - Kernel RBF, s = 1
  - C = 10



Support Vector Machine

Resultados:

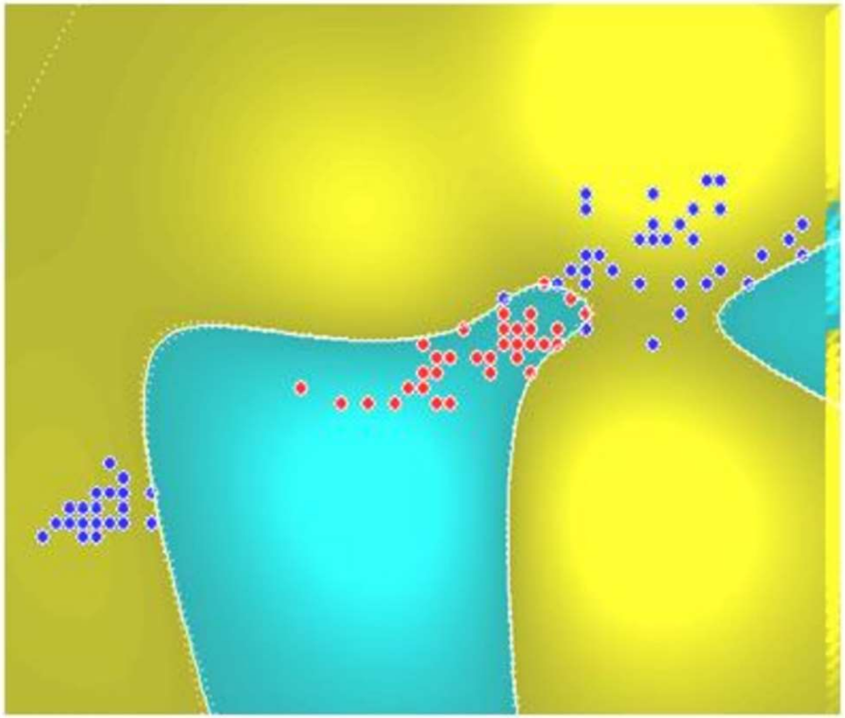
- Dados Iris
  - Kernel RBF, s = 1
  - C = 1000



Support Vector Machine

Resultados:

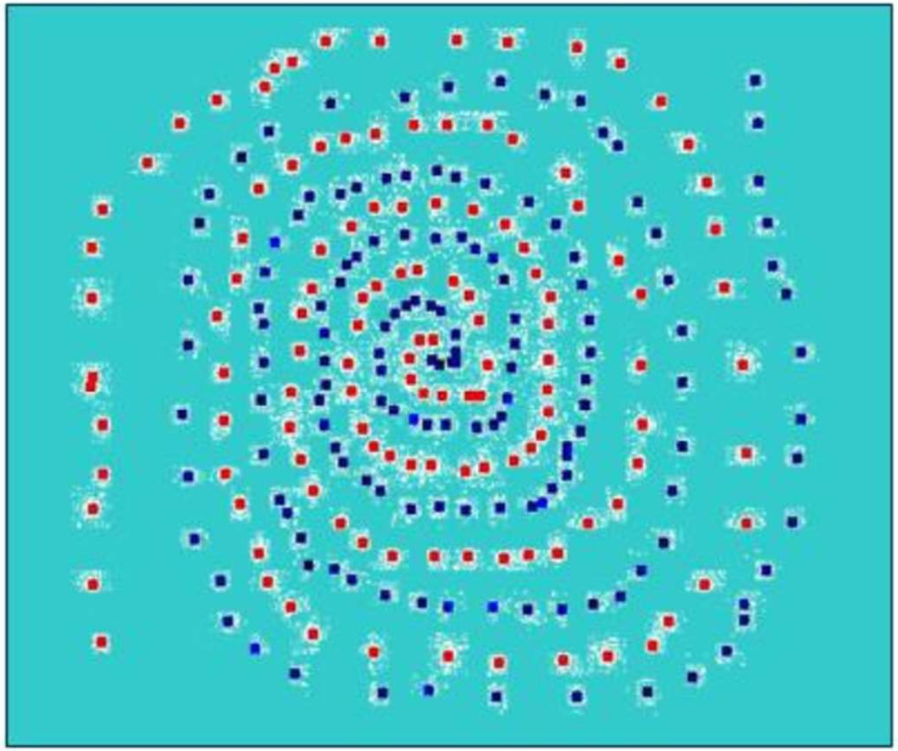
- Dados Iris
  - Kernel RBF, s = 1
  - C = infinito



Support Vector Machine

Resultados:

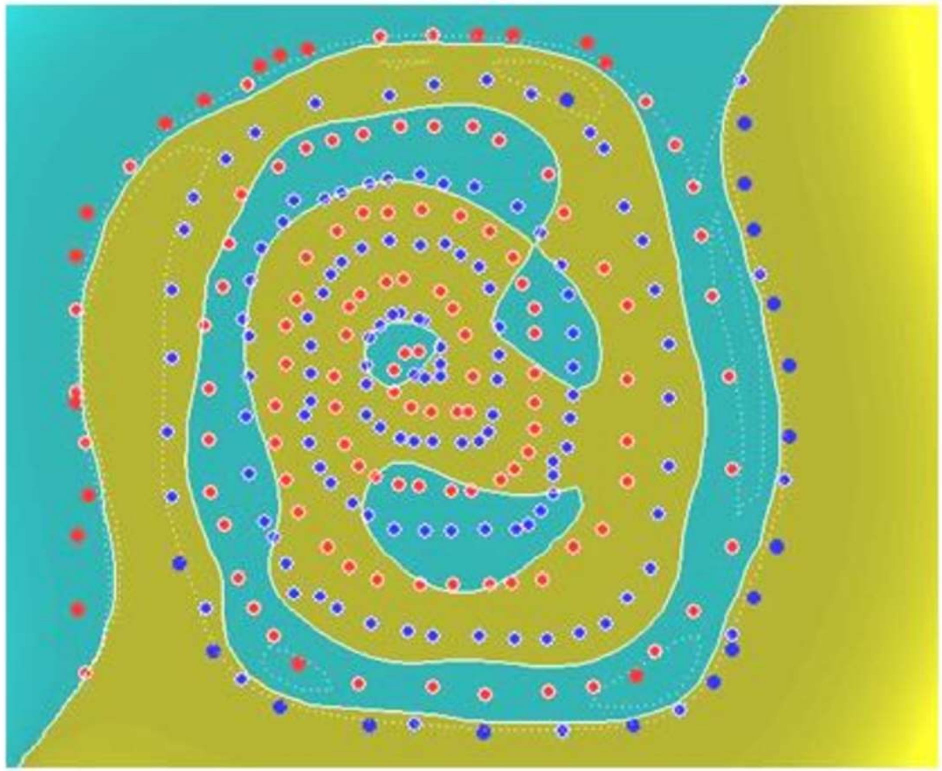
- Espiral complexa



Support Vector Machine

Resultados:

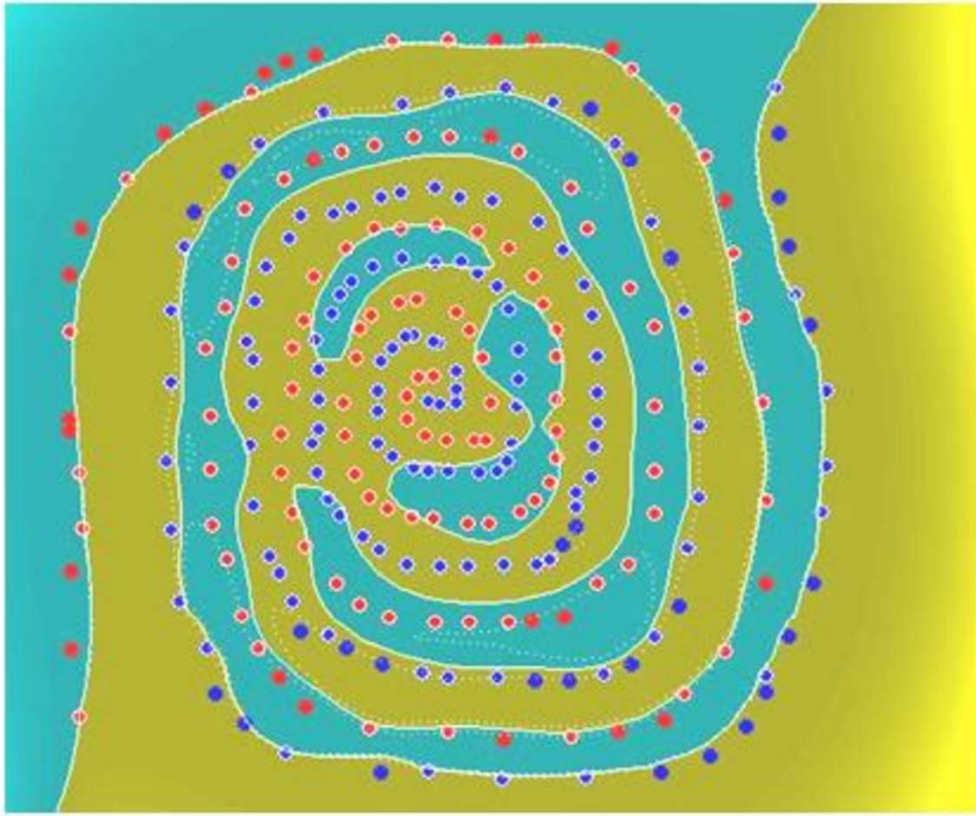
- Espiral complexa
  - o Kernel RBF, s = 1
  - o C = infinito



Support Vector Machine

Resultados:

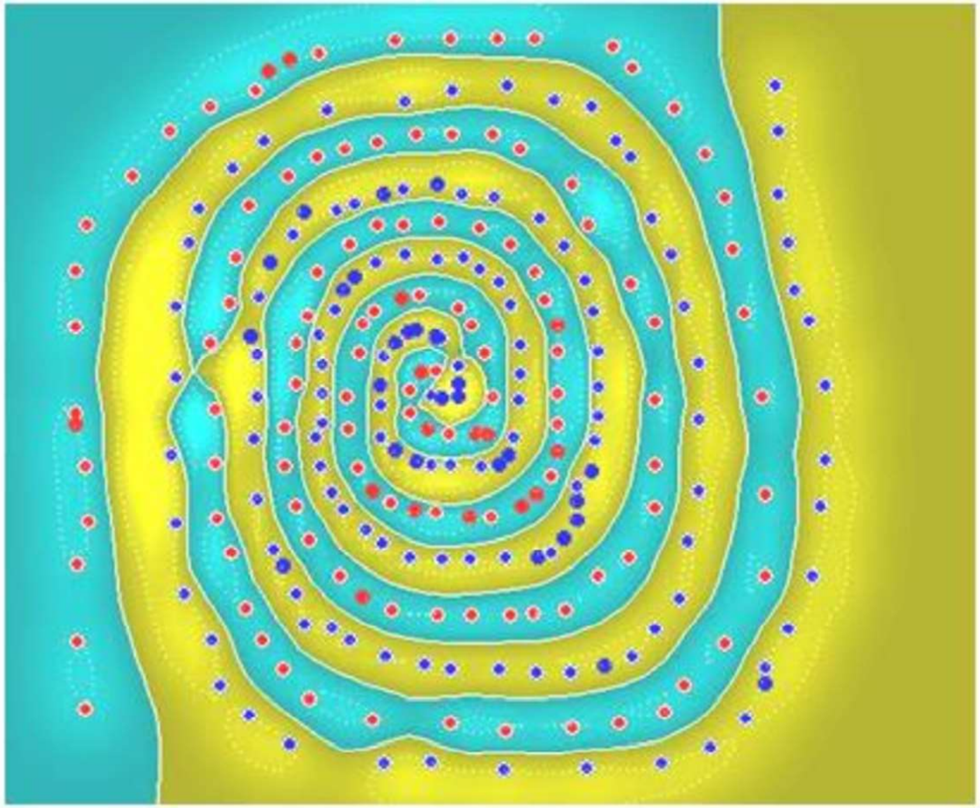
- Espiral complexa
  - Kernel RBF, s = 0.5
  - C = infinito



Support Vector Machine

Resultados:

- Espiral complexa
  - o Kernel RBF, s = 0.1
  - o C = infinito



Support Vector Machine

SVM é eficaz para alta dimensionalidade?

- A complexidade do classificador treinado é caracterizada pelo número de vetores suporte e não pela dimensionalidade dos dados.
- Os vetores suporte são exemplos de treinamentos essenciais ou críticos.
  Se um treinamento for repetido, os mesmos hiperplanos de separação devem ser encontrados (determinístico).
- O número de vetores suporte em um SVM pode ser usado para estimar o limite superior da taxa de erro esperado do classificador, o que é independente da dimensionalidade dos dados.
- Assim, SVM com um pequeno número de vetores suporte pode ter boa generalização, mesmo com a dimensionalidade dos dados alta.

Support Vector Machine

SVM vs RNA

- **SVM**
  - Novo conceito
  - Algoritmo determinístico
  - Ótimas propriedades para generalização
  - Aprendizado difícil – usa técnicas de programação quadrática.
  - Usando ***kernels*** (núcleos) é capaz de aprender funções complexas.

- **Redes Neurais**
  - Relativamente velho
  - Algoritmo não-determinístico
  - Generaliza bem, mas não tem forte fundamento matemático
  - Pode aprender facilmente em um modo incremental.
  - Para aprender funções complexas pode-se usar o multilayer perceptron (**que não é trivial**).

Support Vector Machine

Desvantagens:

- Sensibilidade aos parâmetros: O desempenho do SVM pode ser sensível à escolha dos parâmetros, como o coeficiente de penalidade C e o kernel utilizado, exigindo ajustes cuidadosos.

- Requer escala dos recursos: Antes de aplicar o SVM, é necessário normalizar ou escalar as características para evitar que recursos com escalas muito diferentes afetem o resultado.

- Dificuldade em lidar com dados desbalanceados: Quando as classes são desbalanceadas (uma classe tem muito mais exemplos que outra), o SVM pode favorecer a classe majoritária, resultando em baixo desempenho para a classe minoritária.

Support Vector Machine

Utilidades:

- Classificação: SVM é comumente usado para classificação binária e multiclasse, sendo eficiente em tarefas de classificação.

- Regressão: SVM também pode ser usado para problemas de regressão, estimando valores numéricos ao invés de classes.

- Detecção de Anomalias: SVM é útil na detecção de anomalias, identificando padrões incomuns em dados.

- Classificação em Dados de Alta Dimensão: É adequado para dados com muitas características, como em análise de texto e processamento de imagens.

Support Vector Machine

Aplicações práticas:

- Detecção de Fraudes em Cartões de Crédito: Usado para identificar transações fraudulentas com base em padrões de uso de cartões de crédito.
- Diagnóstico de Câncer: Aplicado para classificar amostras de tecido como benignas ou malignas em testes de biópsia, auxiliando no diagnóstico precoce e tratamento eficiente de pacientes com câncer.
- Reconhecimento de Rostos: Usado para identificar e reconhecer rostos em imagens, sendo utilizado em sistemas de autenticação e vigilância.
- Análise de Sentimento em Redes Sociais: Aplicado para classificar comentários ou postagens em sentimentos positivos ou negativos, permitindo análise de opiniões e feedback de usuários em mídias sociais.
- Previsão de Preços de Ações: O SVM pode ser usado para prever os preços futuros das ações com base em padrões históricos de mercado, auxiliando investidores na tomada de decisões financeiras.

Support Vector Machine

E agora?

Comentários