



INF 1771 – Inteligência Artificial

Aula 14 – Support Vector Machines (SVM)

2016.1



Prof. Augusto Baffa
<abaffa@inf.puc-rio.br>



Formas de Aprendizado

- **Aprendizado Supervisionado**
 - Árvores de Decisão.
 - K-Nearest Neighbor (KNN).
 - **Support Vector Machines (SVM).**
 - Redes Neurais.
- Aprendizado Não Supervisionado
- Aprendizado Por Reforço

Aprendizado Supervisionado

- Observa-se alguns pares de **exemplos de entrada e saída**, de forma a aprender uma **função que mapeia a entrada para a saída**.
- Damos ao sistema a **resposta correta** durante o processo de treinamento.
- É eficiente pois o sistema pode trabalhar diretamente com informações corretas.

Introdução

- 1968: base matemática
 - Teorema de Lagrange (Teoria dos Grupos)
- Proposto em 1992 pelo russo **Vladimir Vapnik**.
 - [Vapnik et al, 1992] Primeiro artigo
 - [Vapnik et al, 1998] Definição detalhada
- Última década
 - Série de artigos com aplicações de SVM
 - Série de artigos com otimizações de SVM
 - SMO, por exemplo (weka)

Introdução

- SVM são utilizadas em diversas áreas:
 - Bioinformática
 - Reconhecimento de assinaturas
 - Classificação de texto e imagens
 - Identificação de spams
 - Reconhecimento de padrões diversos
 - Identificação de dados replicados

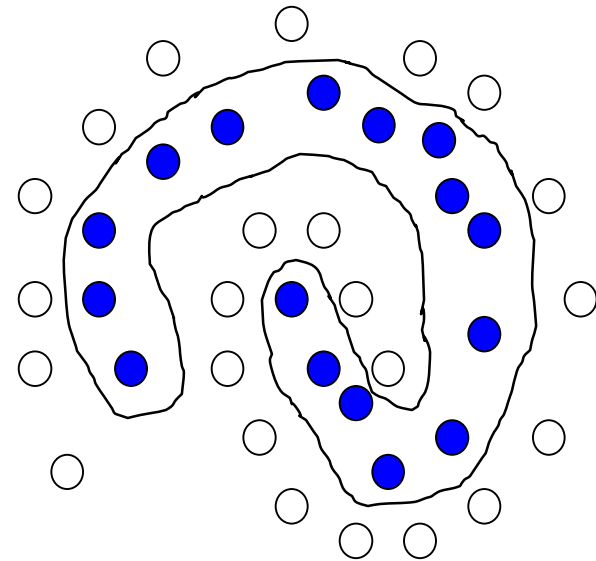
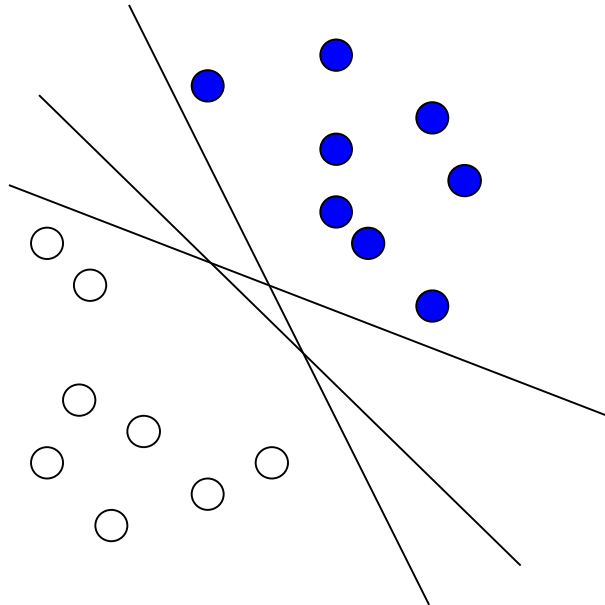
Support Vector Machine

- Consiste em um método de aprendizado que tenta encontrar a **maior margem** para separar diferentes classes de dados.
- Pertence à classe de algoritmos de **aprendizado supervisionado**.
- A essência do SVM é a construção de um **hiperplano ótimo**, de modo que ele possa separar diferentes classes de dados com a maior margem possível.

Motivação da SVM

Reta / Plano / Hiperplano

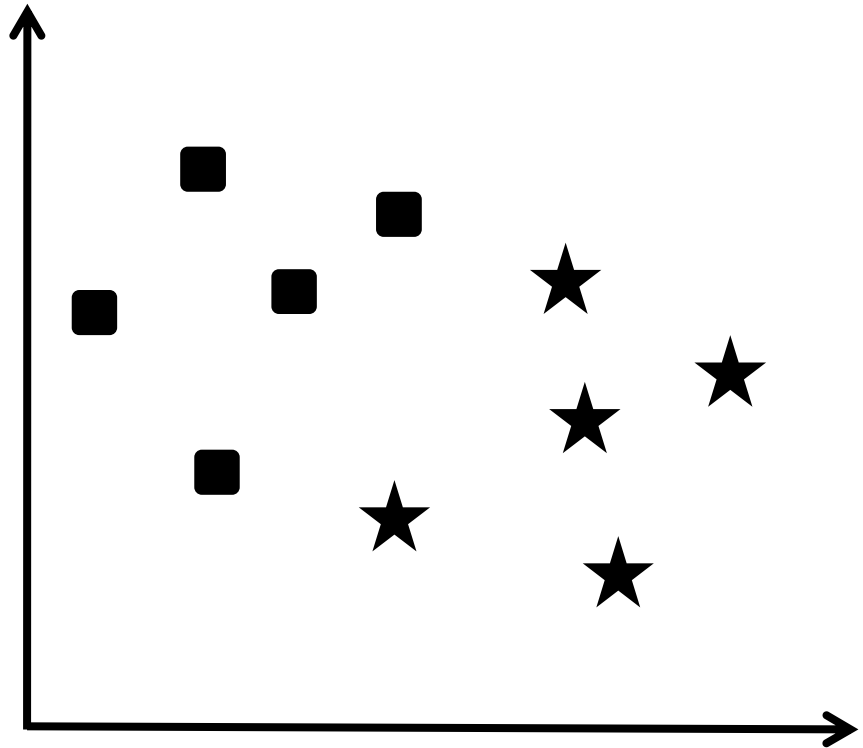
?



Qual o hiperplano ótimo?
Menor erro de classificação

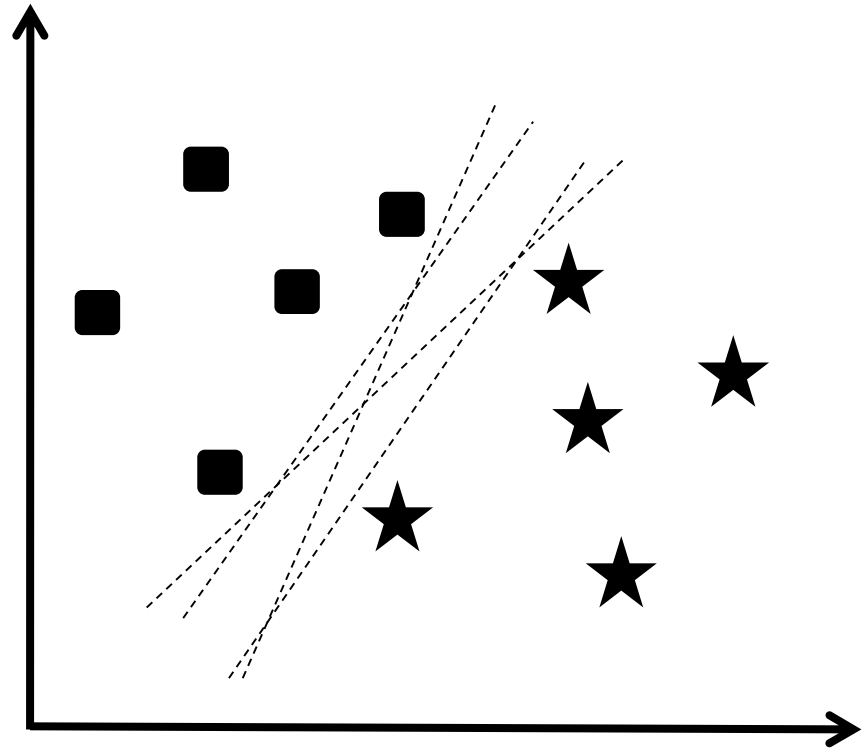
Support Vector Machine

- Como separar essas duas classes?
- Qual o **hiperplano** ótimo?
 - Menor erro de classificação
 - Maior **margem**
 - Distância entre **vetores de suporte** e o hiperplano



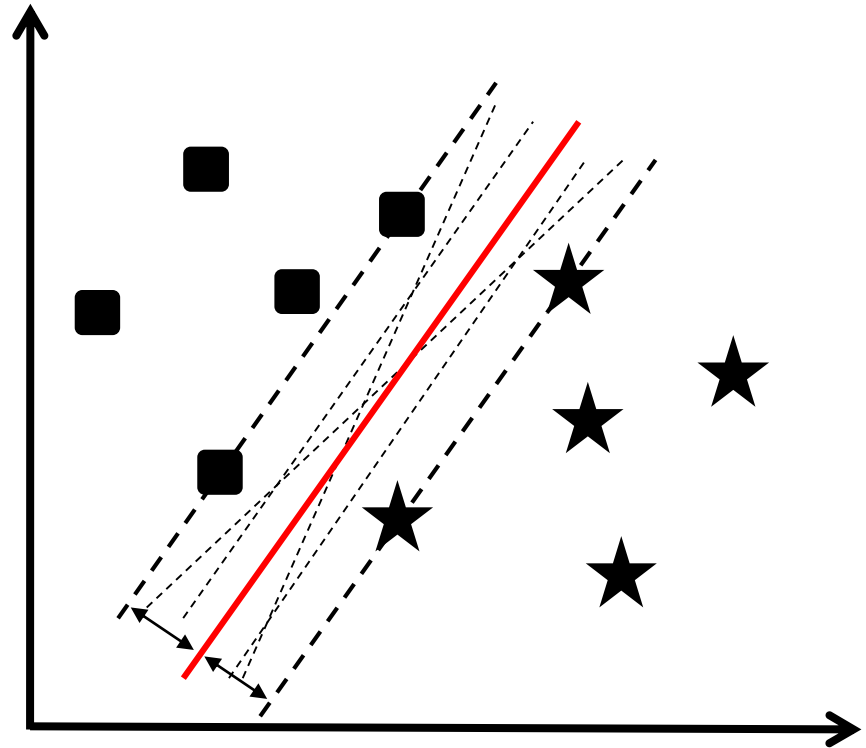
Support Vector Machine

- Como separar essas duas classes?
 - Existem diversas retas que podem ser traçadas para separar os dados.
- Qual delas é a melhor opção?



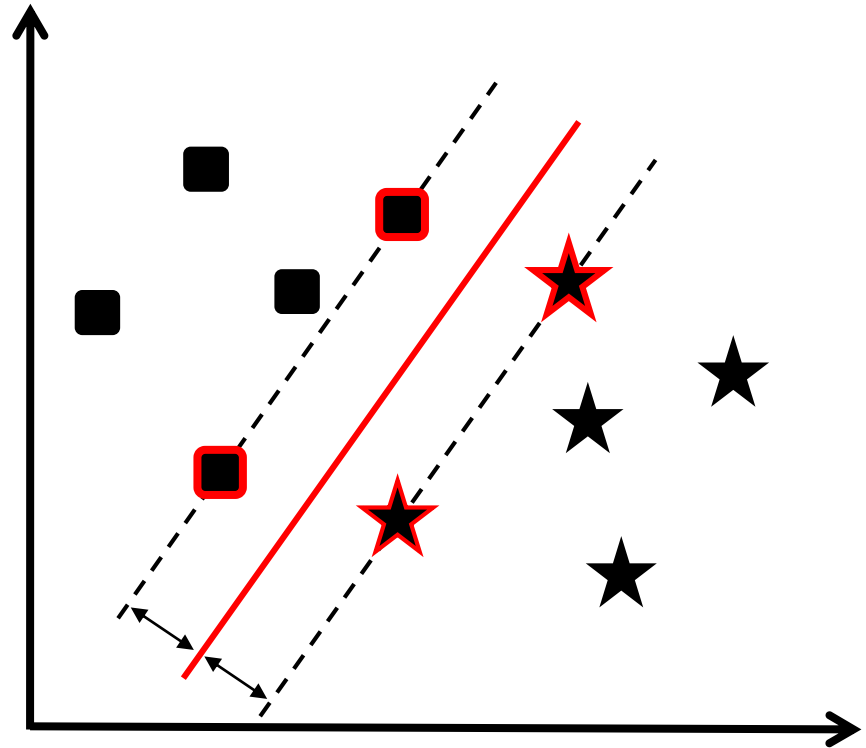
Support Vector Machine

- Como separar essas duas classes?
 - Existem diversas retas que podem ser traçadas para separar os dados.
- Qual delas é a melhor opção?
 - **Hiperplano ótimo!**



Vetores de Suporte

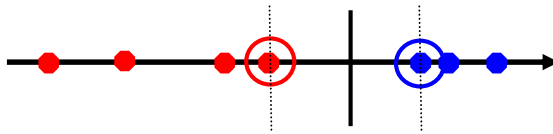
- Servem para definir qual será o hiperplano.
- São encontrados durante a fase de treinamento.
- Os vetores de suporte são os exemplos de treinamento realmente importantes. Os outros exemplos podem ser ignorados.



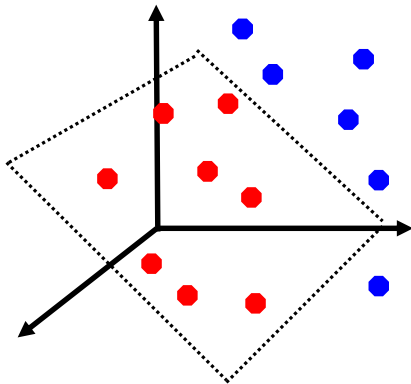
Support Vector Machine

- Hiperplano:

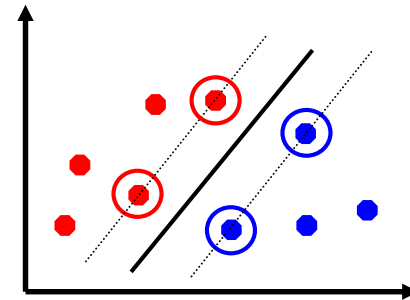
- Espaço 1D = Ponto



- Espaço 3D = Plano



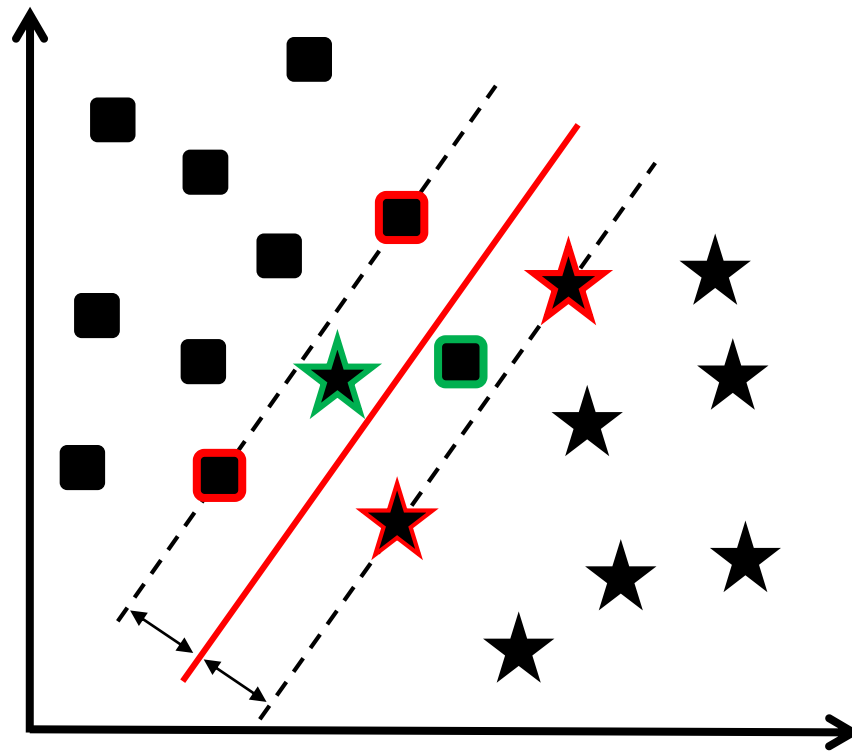
Espaço 2D = Reta



Support Vector Machine

- A aplicação de um método puramente linear para classificar um conjunto de dados pode sofrer com **dois problemas** bastante comuns:
 - Outliers
 - Exemplos rotulados erroneamente
- Mesmo assim o SVM ainda assim pode ser aplicado através do uso do parâmetro C (**soft margin** - variáveis de folga)

Soft Margin

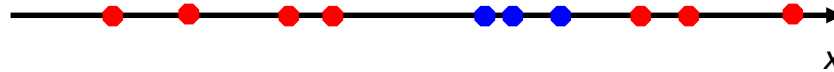


Support Vector Machine

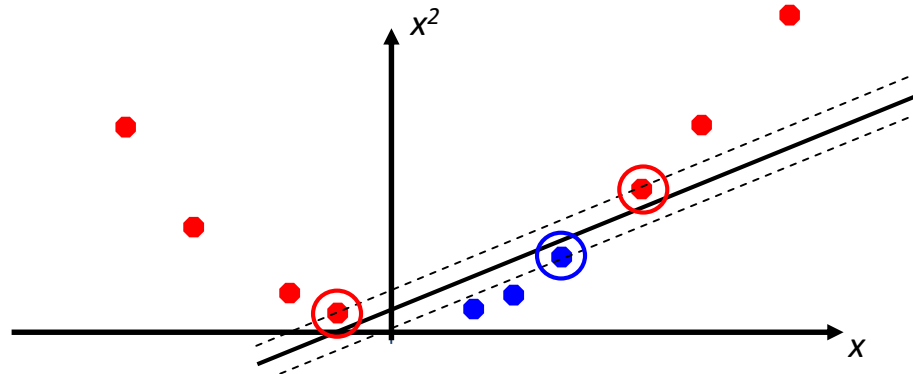
- Em alguns problemas não é possível separar as classes linearmente mesmo utilizando a margem de folga.
- Na realidade, a grande maioria dos problemas reais não são separáveis linearmente.
- O que fazer?

SVM Não-Linear

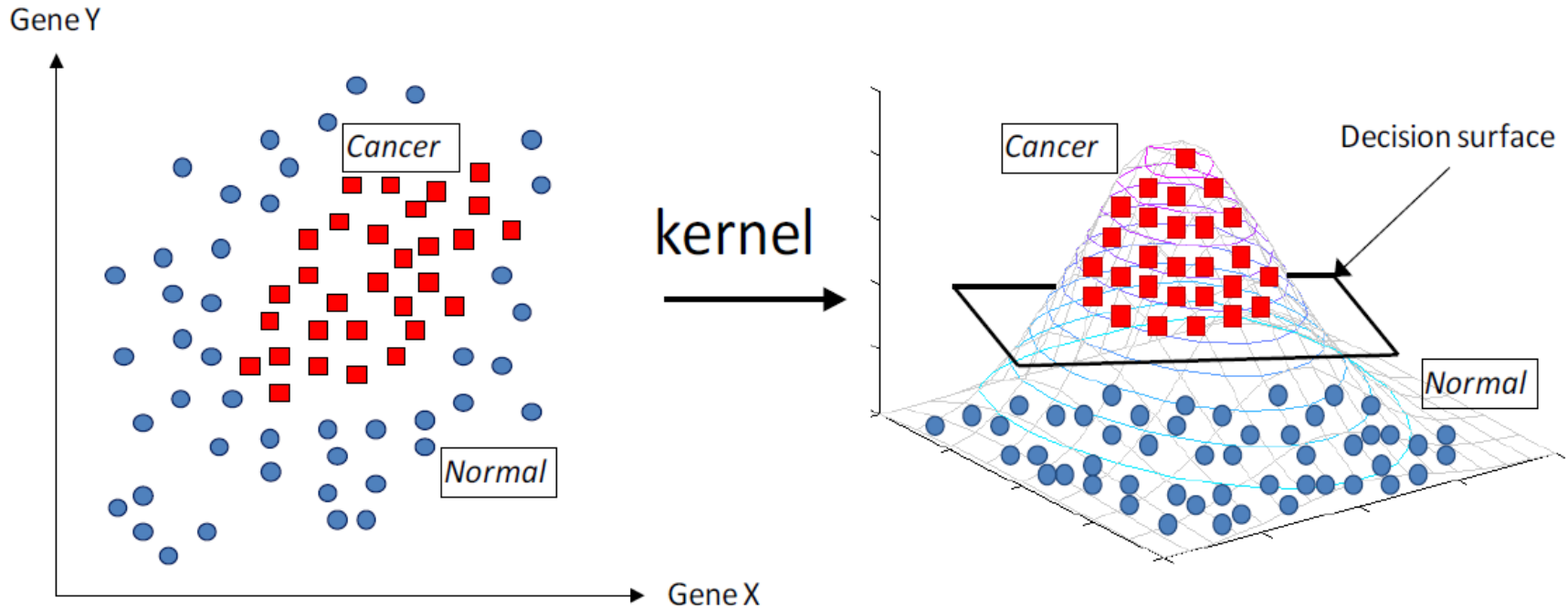
- O que fazer quando os dados não são linearmente separáveis?



- A abordagem utilizada pelo SVM para resolver esse tipo de problema consiste em mapear os dados para um espaço de dimensão maior:



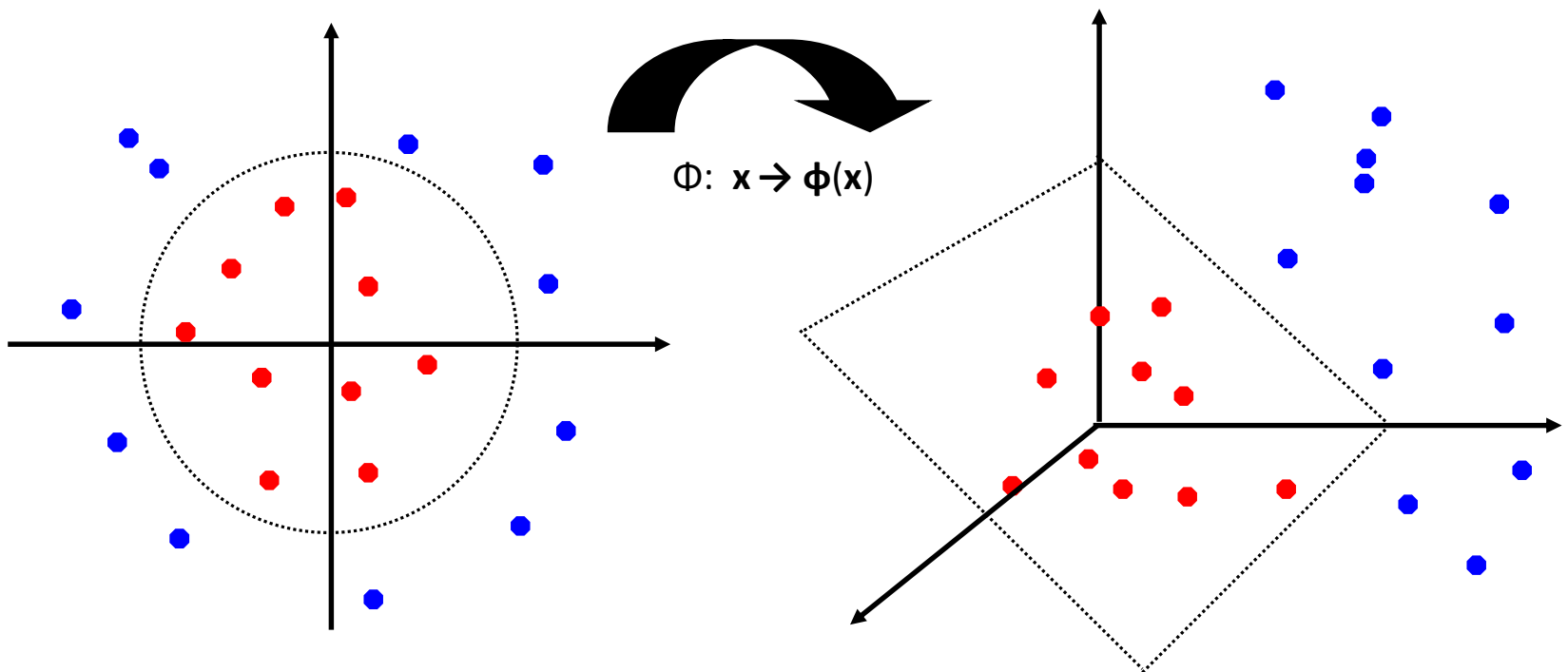
SVM Não-Linear



- Os dados são mapeados em um espaço dimensional maior ("espaço de características").
- O espaço característica é construído através de uma projeção matemática chamada "Kernel Trick" (Função Φ).

SVM Não-Linear

- O espaço de atributos original pode ser mapeado em um espaço de atributos de dimensão maior onde o conjunto de treinamento é linearmente separável:

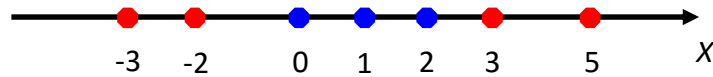


Kernel Trick

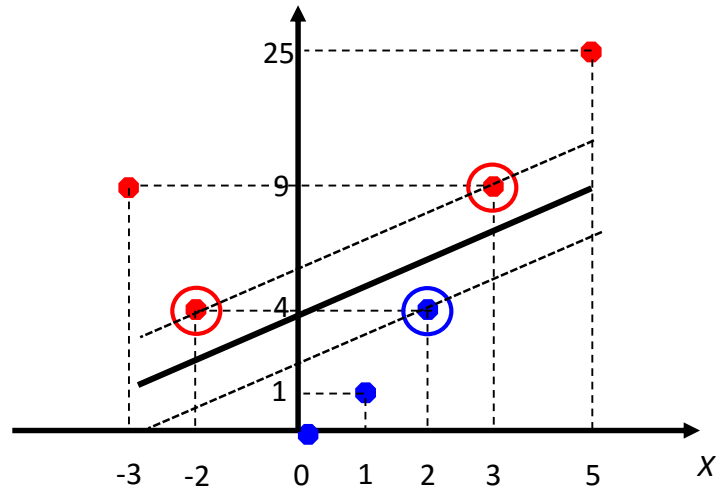
- A função que projeta o espaço de entrada no espaço de características é conhecida com *Kernel*
- Baseado no teorema de Cover
 - Dados no espaço de entrada são transformados (transf. não linear) para o espaço de características, onde são linearmente separáveis.
- O vetor $\phi(x_i)$ representa a “imagem” induzida no espaço de características pelo vetor de entrada

SVM Não-Linear Exemplo

- Considerando o seguinte conjunto de exemplos de treinamento que não são linearmente separáveis:



- Elevando para uma dimensão linearmente separável ($\mathbb{R}^1 \rightarrow \mathbb{R}^2$):

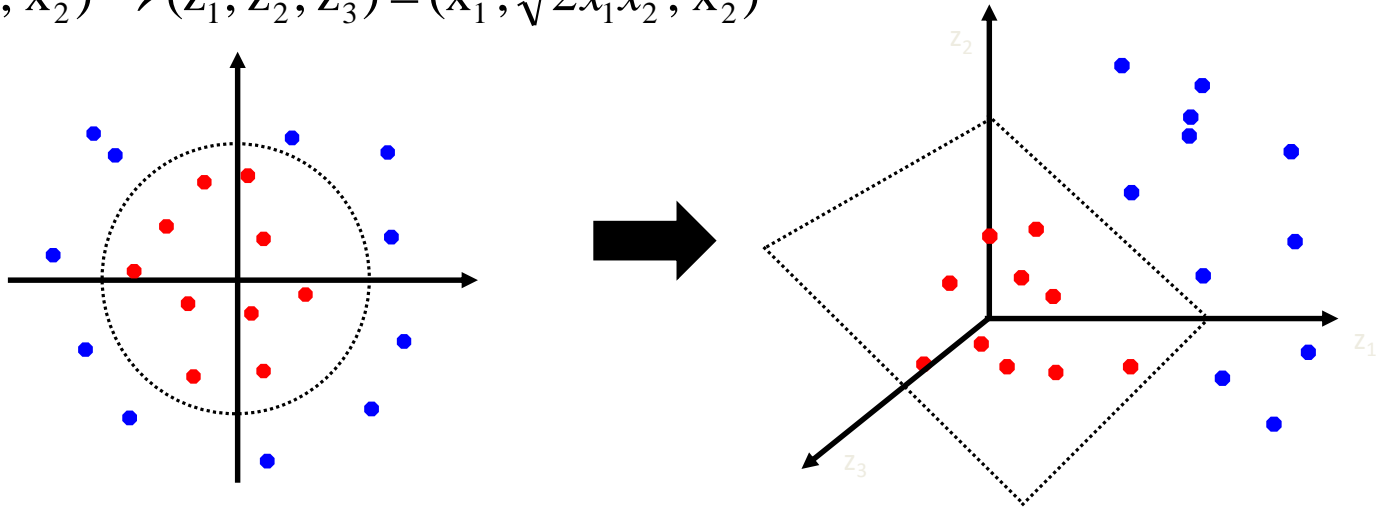


- Kernel:** $\phi(x) = (x, x^2)$

SVM Não-Linear Exemplo

- A mesma metodologia pode ser aplicada em um espaço 2D de características ($\mathbb{R}^2 \rightarrow \mathbb{R}^3$).
- A única diferença é a necessidade de uma nova função de kernel. Um exemplo de função de kernel aplicável nesse caso seria:

$$\phi(x_1, x_2) \rightarrow (z_1, z_2, z_3) = (x_1^2, \sqrt{2x_1x_2}, x_2^2)$$

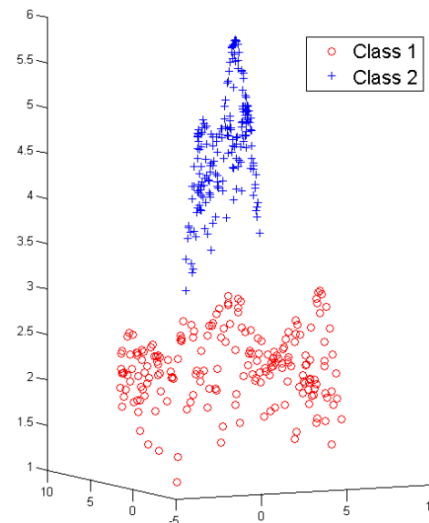
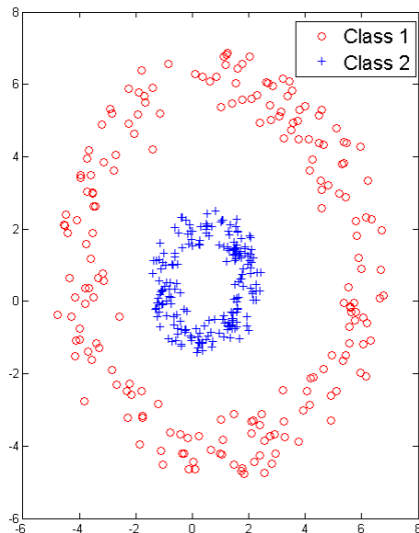


Problema

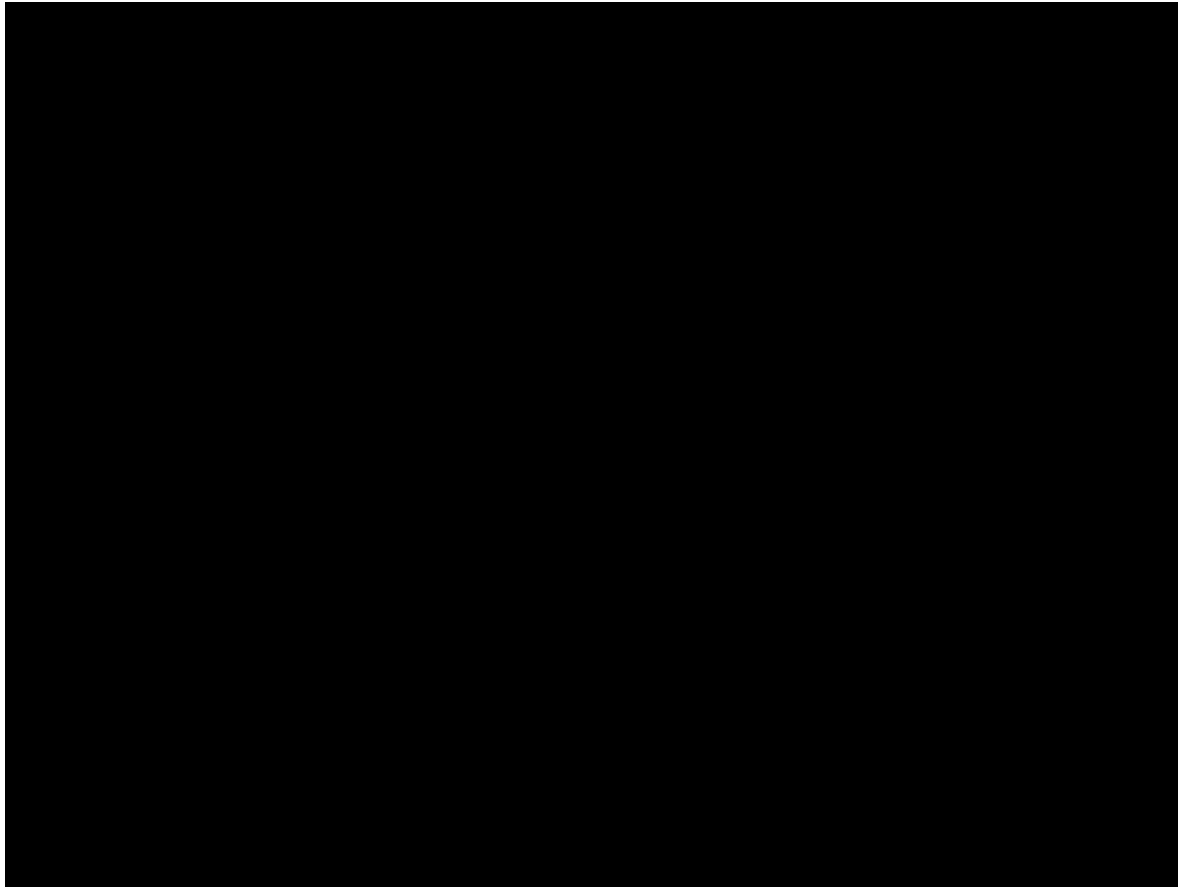
- Como escolher a função $\phi(x_i)$ tal que o espaço de características transformado seja eficiente para classificação e não possua custo computacional alto demais?
- Funções de Núcleo (kernel)
 - Polinomial
 - Gaussiano
 - Sigmoid
- Sempre aumentam o número de dimensões
 - Algumas vezes aumentam bastante!

Funções de Kernel

Kernel	Função $\phi(x_i, x_j)$
Polinomial	$(\delta(x_i \cdot x_j) + k)^d$
Gaussiano	$\exp(-\sigma \ x_i - x_j\ ^2)$
Sigmoidal	$\tanh(\delta(x_i \cdot x_j) + k)$



Polynomial Kernel



Support Vector Machine

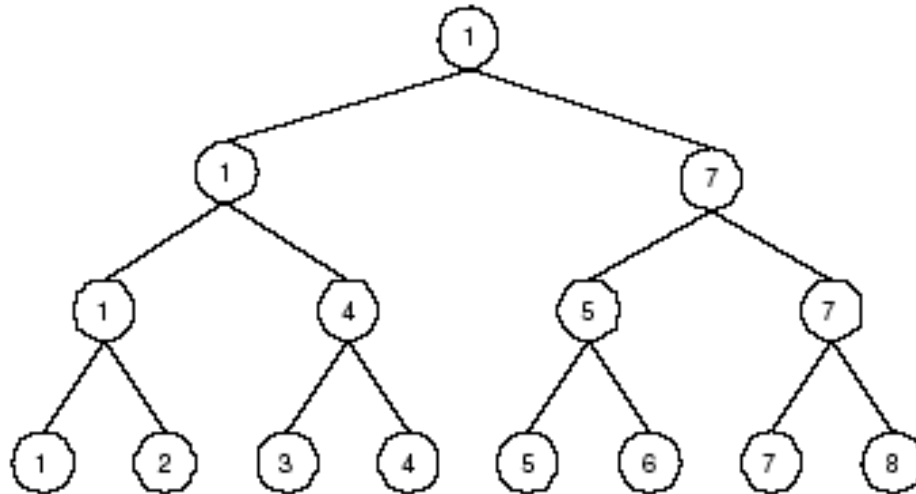
- O SVM foi originalmente concebido para lidar com **classificações binárias**.
- Entretanto, a maior parte dos problemas reais requerem **múltiplas classes**.
- Para se utilizar uma SVM para classificar múltiplas classes é necessário transformar o problema multi-classe em vários problemas de classes binárias
 - Um contra o resto.
 - Pairwise.

Tomada de Decisão

- SVMs são classificadores binários, ou seja, separam duas classes.
- Entretanto, a grande maioria dos problemas reais possuem mais que duas classes.
- Como utilizar os SVMs nesses casos?
 - Pairwise, um-contra-todos

Pairwise

- Consiste em treinar classificadores *pairwise* e arranjá-los em uma árvore



A competição se dá nos níveis inferiores, e o ganhador chegará ao nó principal da árvore.

Número de classificadores para q classes = $q(q-1)/2$.

Um-Contra-Todos

- Neste método, o número de classificadores é igual a q .
- Treina-se um classificador c_i para a primeira classe, usando-se como contra exemplos as outras classes, e assim por diante.
- Para se obter a decisão final pode-se utilizar uma estratégia de votos.

Aplicação

- Antes de aplicar uma SVM para classificar um conjunto de dados é necessário responder algumas questões:
 - Quais funções de kernel utilizar?
 - Qual o valor do parâmetro C (Soft Margin)?
- Validações cruzadas (cross-validations).

Vantagens e Desvantagens

- **Vantagens:**

- Consegue lidar bem com grandes conjuntos de exemplos.
- Trata bem dados de alta dimensão.
- O processo de classificação é rápido.

- **Desvantagens:**

- É necessário definir um bom Kernel.
- O tempo de treinamento pode ser bem longo dependendo do número de exemplos e dimensionalidade dos dados.

Aplicações

- Categorização de texto
 - Filtragem de email
 - Web-searching
 - Classificação/Indexação de documento
 - Gestão Eletrônica de Documentos (GED).

Aplicações

- Imagens
 - Indexação de imagens
 - Aplicações médicas
 - Recuperação de imagens (Image Retrieval)

Aplicações

- Reconhecimento da escrita.
 - Geralmente com resultados melhores que outros tipos de classificadores.
- Bio-Informática
 - Categorização automática de genes em DNA.
 - Seqüências de aminoácidos.
 - Classificação de proteínas.

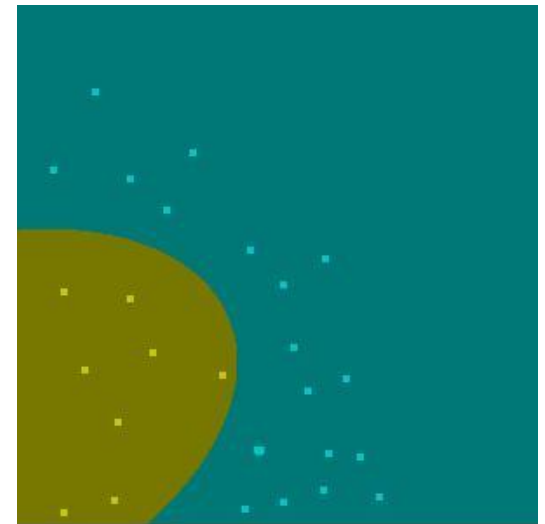
LIBSVM

- <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
 - <http://www.python.org/download/>
 - <http://www.gnuplot.info/download.html>
- Disponível em Python, Java e C++
 - C# via wrapper
- **Bases de Exemplos:**
 - <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>
 - <http://archive.ics.uci.edu/ml/datasets.html>

LIBSVM

- Applets para brincar:

- <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- http://neuron.eng.wayne.edu/java/AHK/EPM_pp.html



Change Run Clear Save Load -t2 -c 580

Leitura Complementar

- Mitchell, T. **Machine Learning**, McGraw–Hill Science/Engineering/Math, 1997.
- Duda, R., Hart, P., Stork, D., **Pattern Classification**, John Wiley & Sons, 2000
- Cristianini, N., Shawe-Taylor, J., **An Introduction to Support Vector Machines and Other Kernel-based Learning Methods**, Cambridge University Press, 2000.

