

Inteligência Artificial

Aprendizado Supervisionado: Classificação
k-Nearest Neighbor (kNN) e Support Vector Machines (SVM)

Prof. Dr^a. Andreza Sartori

asartori@furb.br

Documentos Consultados/Recomendados

- ARTERO, Almir Olivette. **Inteligência artificial: teórica e prática**. 1. ed. São Paulo: Livraria da Física, 2008.
- COPPIN, Ben. **Inteligência artificial**. Rio de Janeiro: LTC, 2013.
- KLEIN, Dan; ABBEEL, Pieter. **Intro to AI**. UC Berkeley. Disponível em: <http://ai.berkeley.edu>
- LIMA, Edirlei Soares. **Inteligência Artificial**. PUC-Rio, 2015.
- MALIK, Jitendra. Department of EECS - **UC Berkeley**. <http://www-inst.eecs.berkeley.edu/~cs280/sp15/index.html>
- NG, Andrew. **Machine Learning**. Stanford University. <https://www.coursera.org/learn/machine-learning>
<http://cs229.stanford.edu/materials.html>
- RUSSELL, Stuart J. (Stuart Jonathan); NORVIG, Peter. **Inteligência artificial**. Rio de Janeiro: Campus, 2013. 1021p.
- SEBE, Nicu. **Classification**. Universidade de Trento. 2011.

Plano de Ensino da disciplina

Unidade 1: Fundamentos de Inteligência Artificial

Unidade 2: Busca

Unidade 3: Sistemas baseados em conhecimento

Unidade 4: Aprendizado de Máquina e Redes Neurais

Unidade 5: Tópicos especiais



Plano de Ensino da disciplina

Unidade 1: Fundamentos de Inteligência Artificial

Unidade 2: Busca

Unidade 3: Sistemas baseados em conhecimento

Unidade 4: Aprendizado de Máquina e Redes Neurais

Unidade 5: Tópicos especiais



Plano de Ensino da disciplina

Unidade 1: Fundamentos de Inteligência Artificial

Unidade 2: Busca

Unidade 3: Sistemas baseados em conhecimento

Unidade 4: Aprendizado de Máquina e Redes Neurais

4.2 Aprendizado Supervisionado

4.2.1 Regressão

4.2.2 k-Nearest Neighbour (KNN)

4.2.3 Support Vector Machine (SVM)

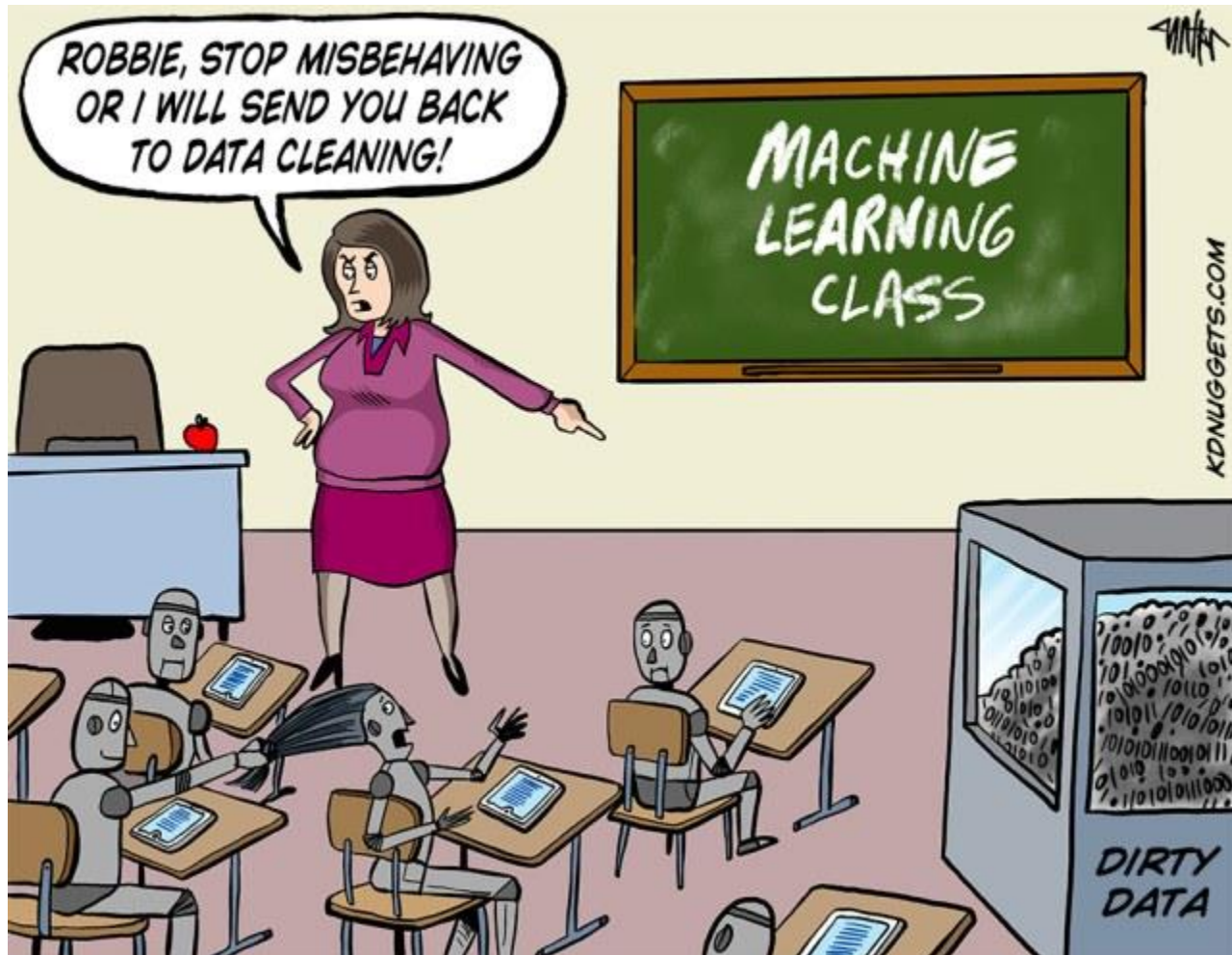
4.2.4 Redes Neurais

4.3 Aprendizado Não-Supervisionado

4.3.1 Clustering: k-means



Recapitulando...

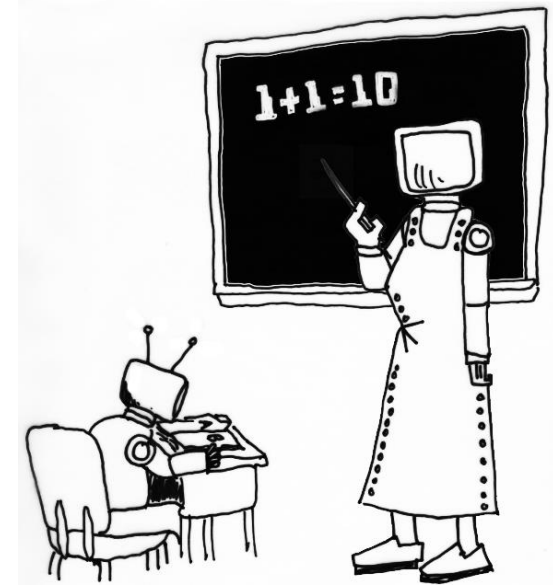


Aprendizado Supervisionado

- Damos ao sistema a “**resposta correta**” durante o processo de treinamento.
- Dado um conjunto de entradas de treinamento e saídas correspondentes, produz os resultados "corretos" para novas entradas.

SUPERVISED MACHINE LEARNING

- É eficiente pois o sistema pode trabalhar diretamente com informações corretas.



Abordagens do Aprendizado Supervisionado

- **Classificação:**

- Responde se uma determinada “entrada” pertence a uma certa classe.
- Dada a imagem de uma fruta: informa que fruta é (dentro um número finito de classes).

- **Regressão:**

- Faz uma predição a partir de exemplos.
- Prever o valor dos imóveis, dados os valores por metro quadrado.

Abordagens do Aprendizado Supervisionado

- Classificação:

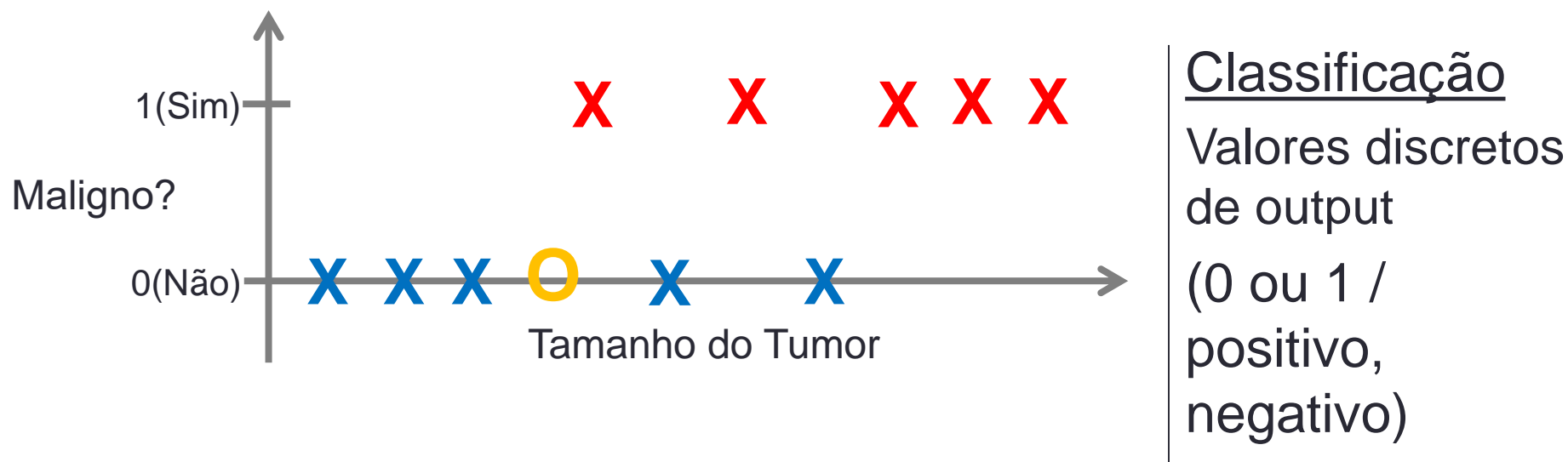
- Responde se uma determinada “entrada” pertence a uma certa classe.
- Dada a imagem de uma fruta: informa que fruta é (dentro um número finito de classes).

- Regressão:

- Faz uma predição a partir de exemplos.
- Prever o valor dos imóveis, dados os valores por metro quadrado.

Aprendizado Supervisionado: Classificação

Prever se tumor na mama é Maligno ou Benigno.



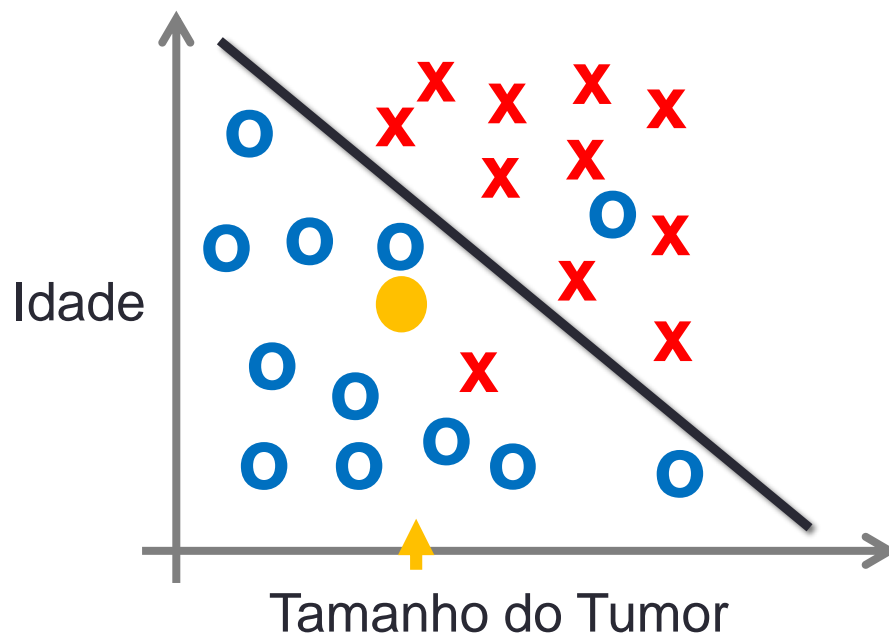
Qual é a probabilidade / chance de um tumor ser maligno ou benigno?

Pode ter mais de dois valores para valores possíveis de saída (multiclasse).

Exemplo: 0 (benigno), 1 (câncer tipo 1), 2 (câncer tipo 2), 3,n

Aprendizado Supervisionado: Classificação

Prever se tumor na mama é Maligno ou Benigno.



Mais de uma característica (feature)

- Espessura
- Uniformidade do tamanho da célula
- Uniformidade da forma celular
- ... (número infinito de características – SVM)

Classificação

Executamos classificação todos os dias:



Cadeira



Mesa

Classificação

Executamos classificação todos os dias:

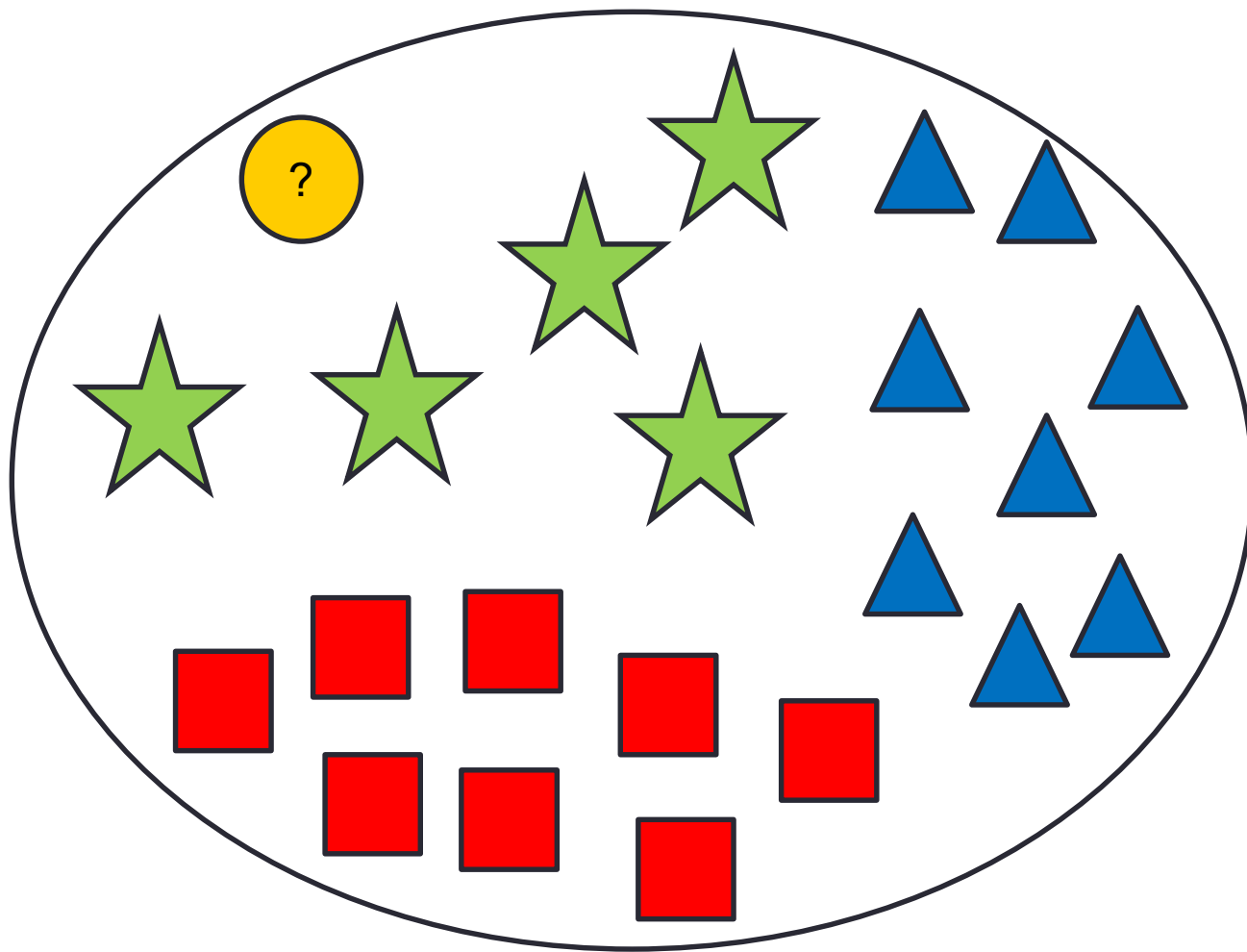


Comestível



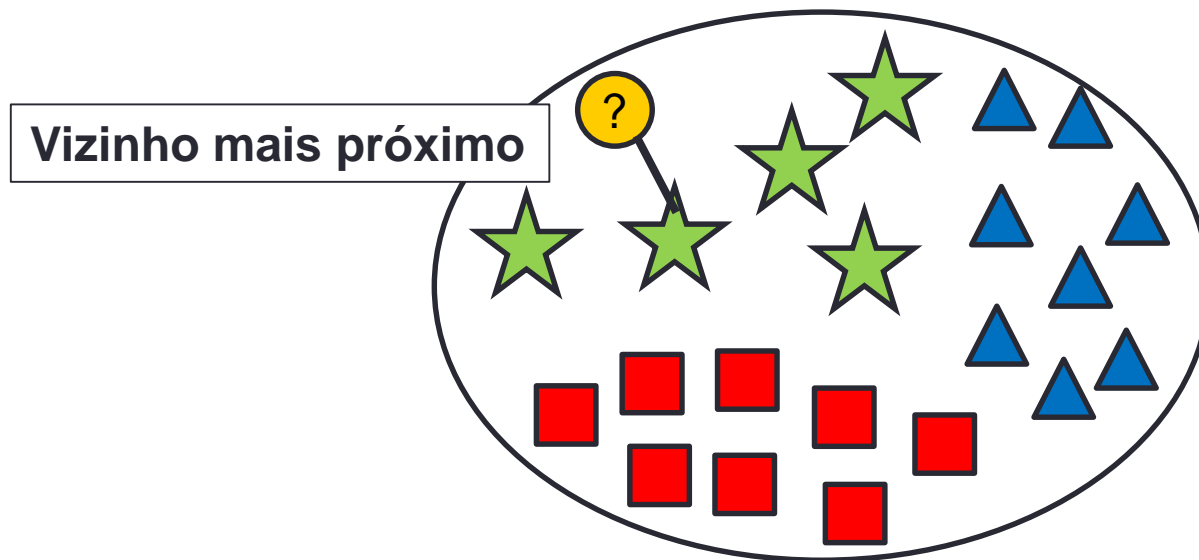
Não Comestível

Classificação



k-Nearest Neighbor (kNN)

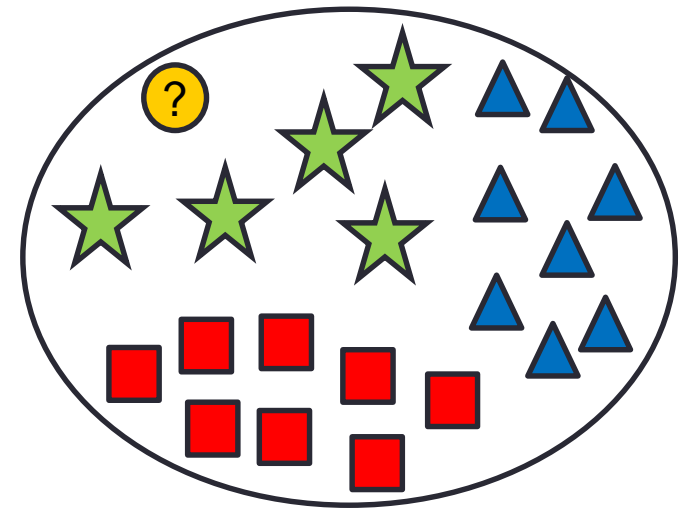
- **k-Vizinhos Mais Próximos**
- É um dos algoritmos de classificação mais **simples**.
- Classifica objetos com base nos **exemplos de treinamento** que se encontram mais próximos no espaço de características.



k-Nearest Neighbor

Para utilizar o kNN é necessário:

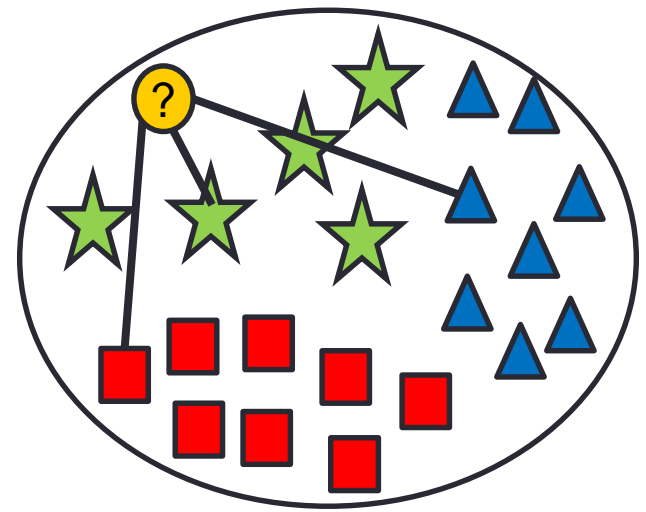
1. Um conjunto (Data Set) de exemplos de treinamento.
2. Definir uma métrica para calcular a distância entre os exemplos de treinamento.
3. Definir o valor de k
 - k é o número de vizinhos mais próximos que serão considerados pelo algoritmo.



k-Nearest Neighbor

Classificação de um exemplo desconhecido com o algoritmo KNN:

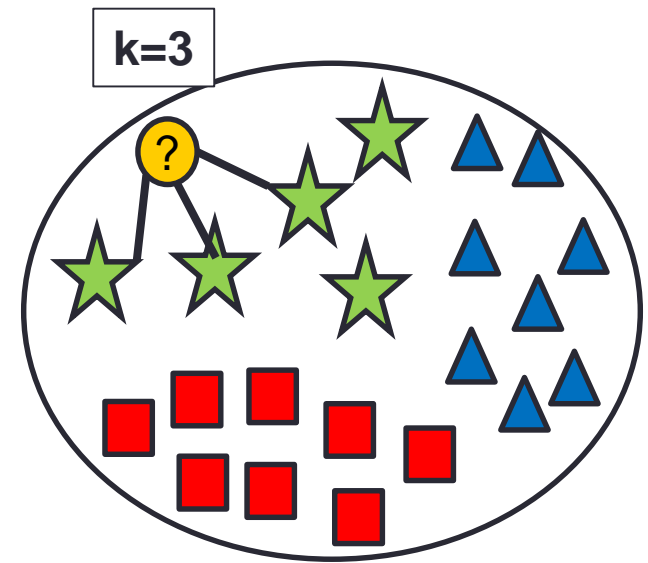
1. Cálculo da distância entre o exemplo desconhecido e os outros exemplos do conjunto de treinamento.



k-Nearest Neighbor

Classificação de um exemplo desconhecido com o algoritmo KNN:

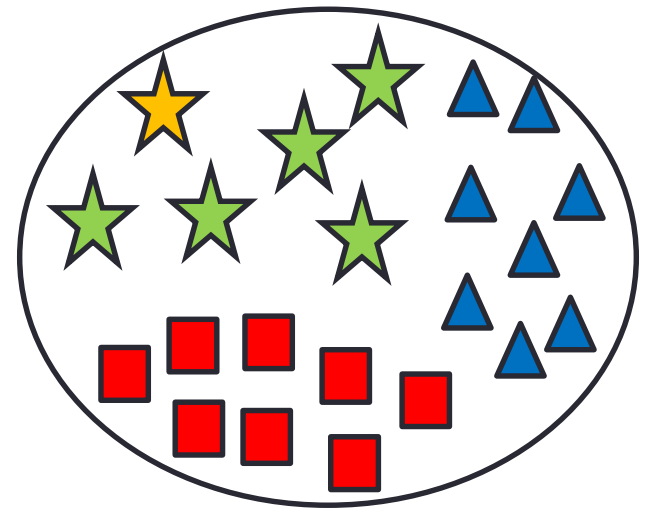
1. Cálculo da distância entre o exemplo desconhecido e os outros exemplos do conjunto de treinamento.
2. Identificar os k vizinhos mais próximos.



k-Nearest Neighbor

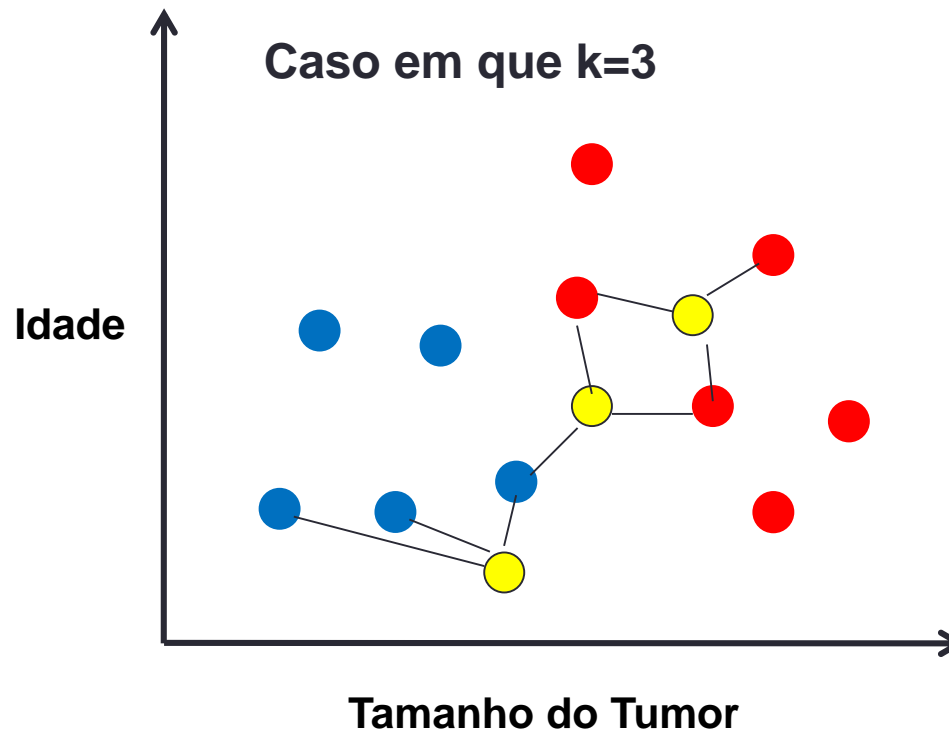
Classificação de um exemplo desconhecido com o algoritmo KNN:

1. Cálculo da distância entre o exemplo desconhecido e os outros exemplos do conjunto de treinamento.
2. Identificar os k vizinhos mais próximos.
3. Utilizar o rótulo (label) da classe dos vizinhos mais próximos para determinar o rótulo de classe do exemplo desconhecido (votação majoritária).



Espaço de Características

Para um exemplo não rotulado x , encontre os k mais próximos a ele na base de dados rotulada e atribua a classe mais frequente para x .

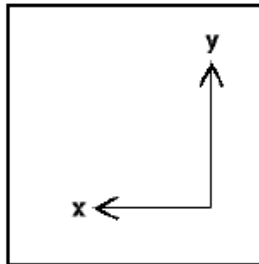


Prever se tumor na mama é Maligno ou Benigno.

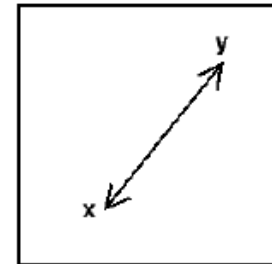
Cálculo da Distância Entre Dois Pontos

- O principal propósito da medida de distância é **identificar os dados que são similares e que não são similares**.
- Existem diferentes maneiras de calcular a distância entre dois pontos. Formas de medir a distância:
 - Distância Euclidiana
 - Distância Manhattan (distância medida entre 2 pontos através dos ângulos de x e y)
 - Distância de Minkowsky (generalização da distância euclidiana)
 - Distância de Hamming (verifica se os 2 atributos são iguais ou não)

- As distâncias mais utilizadas:



Manhattan

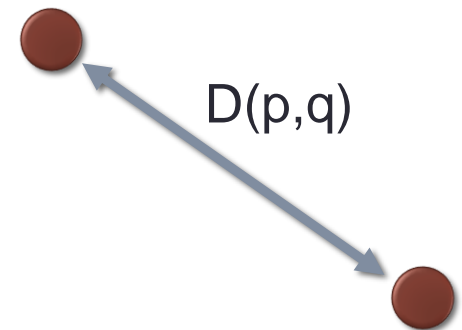


Euclidean

Cálculo da Distância Entre Dois Pontos

- Distância Euclidiana:
 - Dá a mesma importância para todas as características.
 - Ponto negativo: sensibilidade aos valores extremos - outliers

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$



$$D(p,q) = \sqrt{(\text{soma}((p-q)^2))}$$

Porém é importante normalizar os dados!

kNN: Normalização

O dados precisam ser **normalizados** para evitar que as medidas de distância sejam dominadas por uma única característica.

- Exemplos:
 - Altura de uma pessoa pode variar de 1,10 à 2,10.
 - Peso de uma pessoa pode variar de 40 kg à 160 kg.
 - O salário de uma pessoa podem variar de R\$ 800 à R\$ 30.000.
- A Normalização ajusta as escalas de valores das características para o mesmo intervalo:
[-1 a 1] , [0 a 1], etc.

kNN: Normalização

Normalização linear no intervalo [0,1]

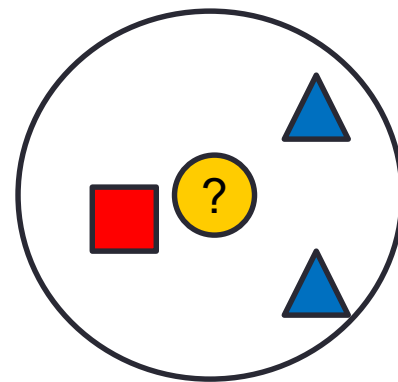
$$N(X) = \frac{X - \text{Min}(X)}{\text{Max}(X) - \text{Min}(X)}$$

CPF	Despesa	Despesa_normalizada
999999999999	1000	0,14
111111111111	2000	0,43
333333333333	3000	0,71
555555555555	1500	0,29
222222222222	1500	0,29
000000000000	1000	0,14
888888888888	3000	0,71
777777777777	500	0
666666666666	4000	1
444444444444	1000	0,14

k-Nearest Neighbor

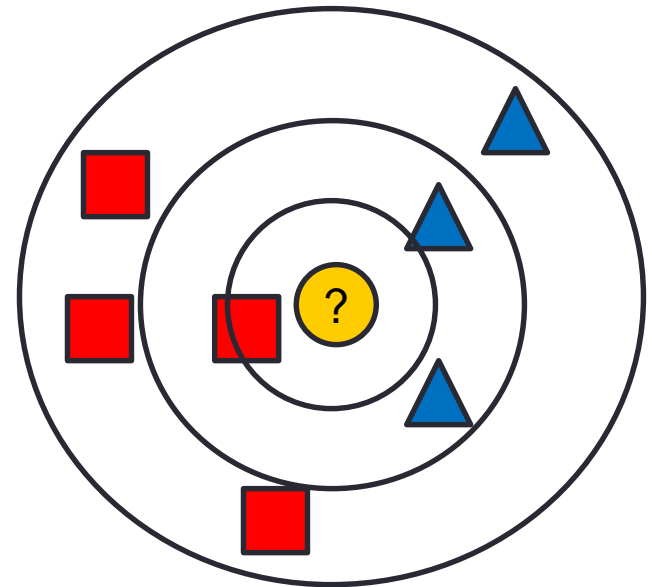
Como determinar a classe do exemplo desconhecido a partir da lista de vizinhos mais próximos:

- Considera-se o voto majoritário entre os rótulos (labels) de classe dos k vizinhos mais próximos.
- Como escolher o valor de k ?



k-Nearest Neighbor

- **K = 1**
 - Pertence a classe de quadrados.
- **K = 3**
 - Pertence a classe de triângulos.
- **K = 7**
 - Pertence a classe de quadrados.



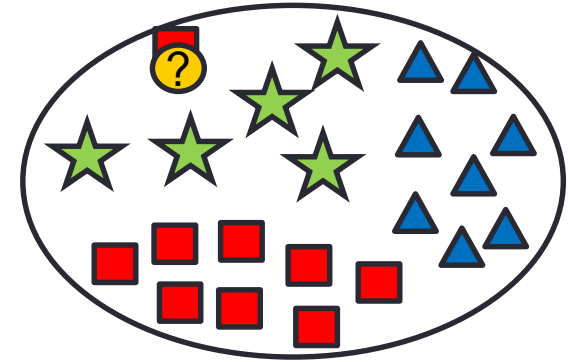
k-Nearest Neighbor

- Como escolher o valor do parâmetro k ?
 - Se k tem valor baixo, a classificação fica sensível a pontos de ruído.
 - Se k tem valor alto, pode aumentar a inclusão de elementos de outras classes.
 - O valor do parâmetro k é escolhido comumente através de tentativa-e-erro.
 - Avaliação empírica com diferentes valores de k .
- Além disso, é necessário sempre escolher um valor ímpar para k , assim se evita empates na votação.

k-Nearest Neighbor

- Vantagens:

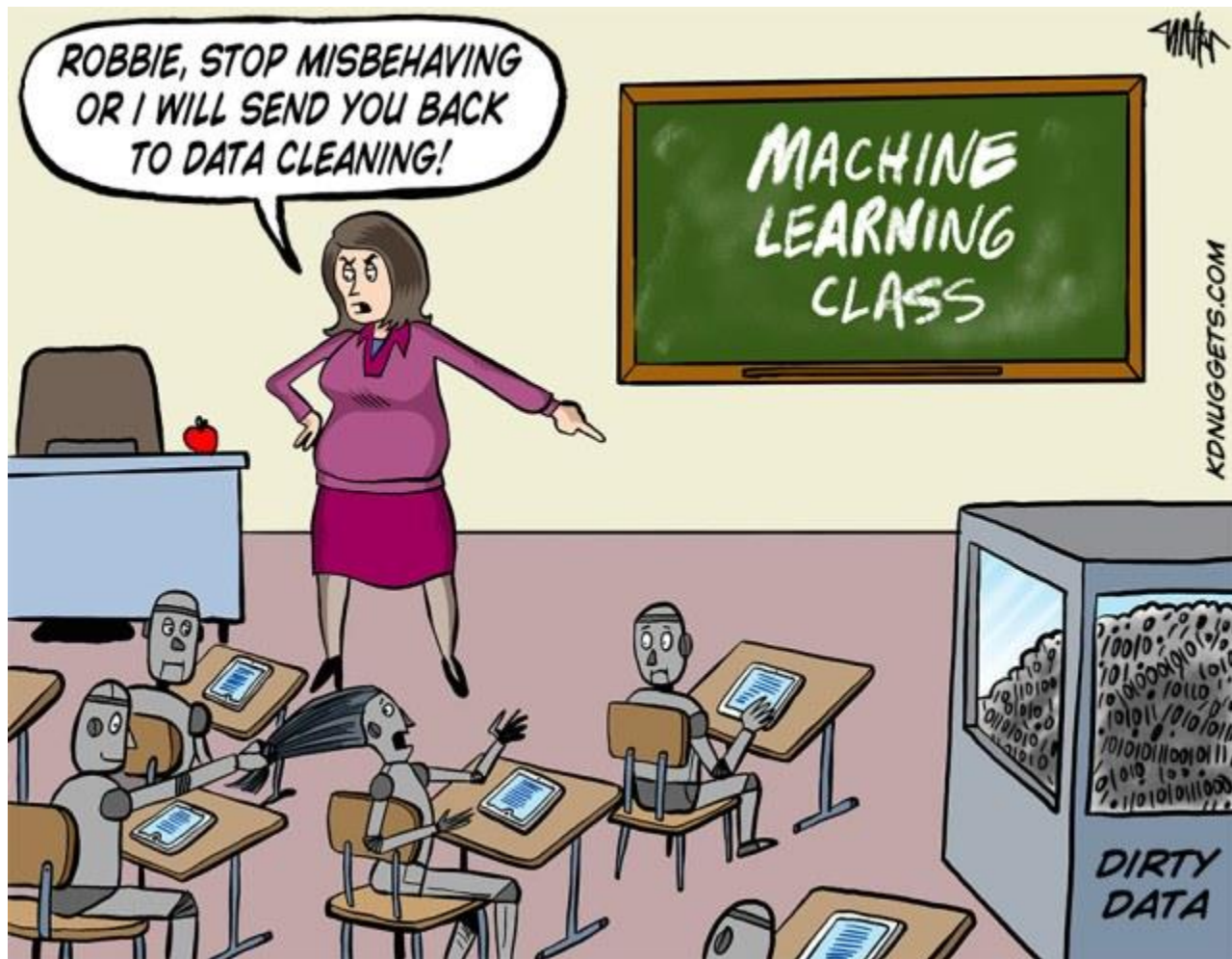
- Técnica simples e facilmente implementada.
- Bastante flexível.
- Em alguns casos apresenta ótimos resultados.



- Desvantagens:

- Pode ser um processo computacionalmente complexo, pois requer um cálculo de distância para cada exemplo de treinamento.
- Pode consumir muito tempo quando o conjunto de treinamento é muito grande.
- A precisão da classificação pode ser severamente degradada pela presença de ruído ou características irrelevantes.

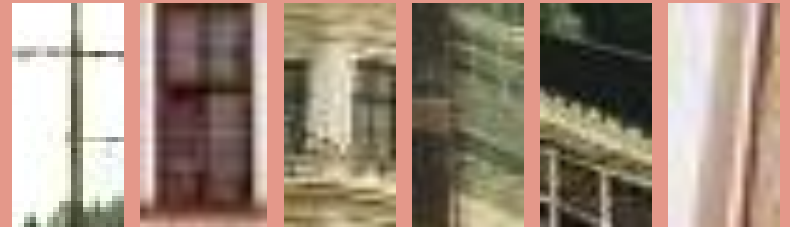
Support Vector Machine (SVM)



Classificação de Pedestres



(+ exemplos)



(- exemplos)

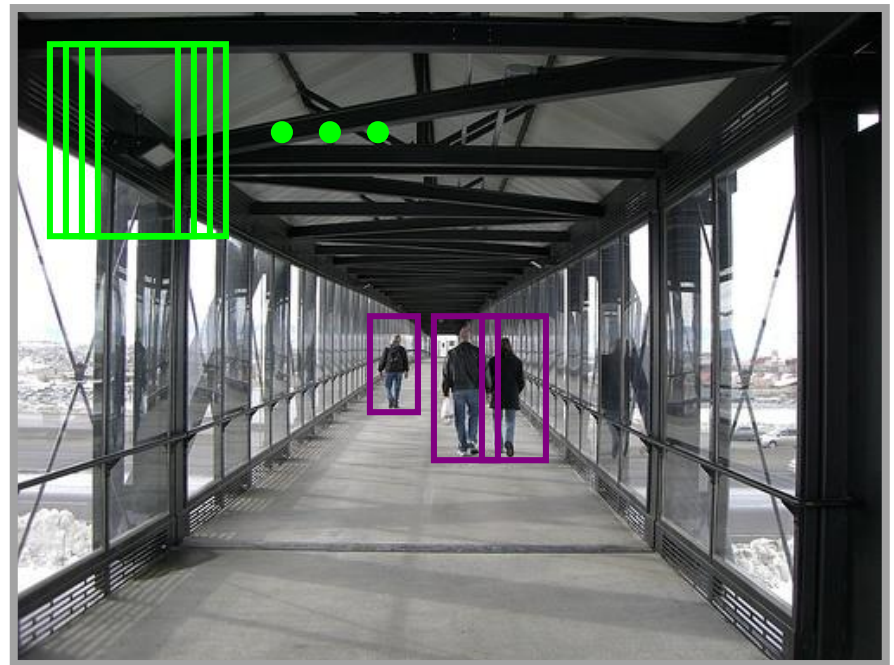
Representação das características

Treinamento

Classificador

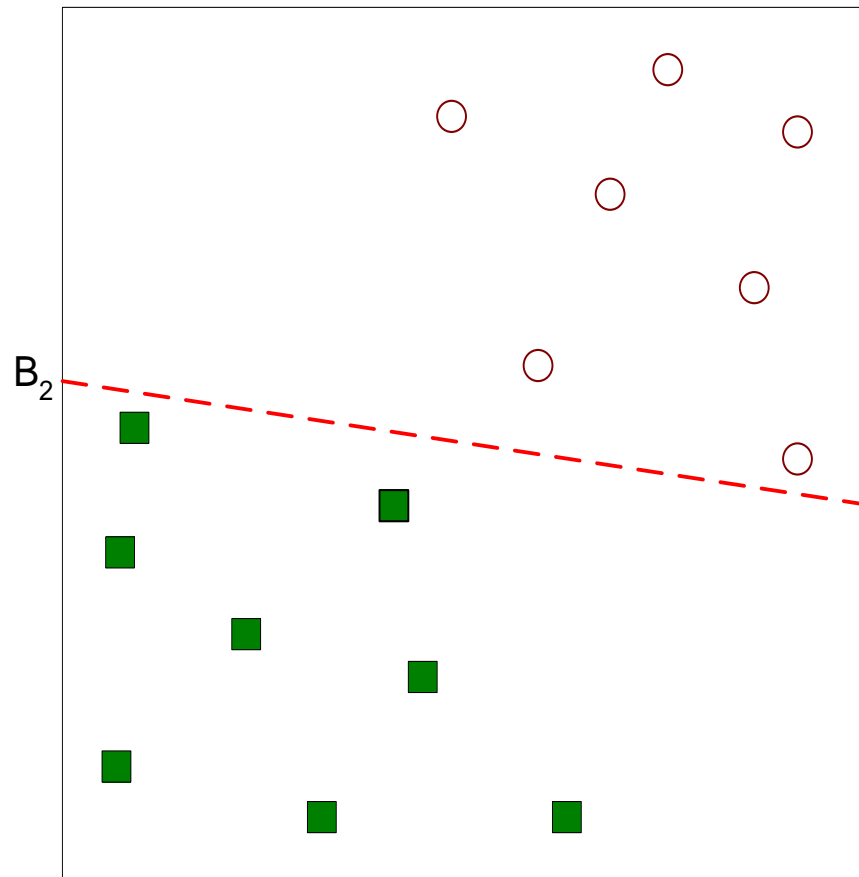
Exemplo: abordagem de Janela Deslizante Multi-Escalar para detecção de objetos

- Complexidade proporcional ao número de pixels
 - Pode ter milhões de posições
- Cálculo com múltiplas escalas - pessoas com tamanhos diferentes;



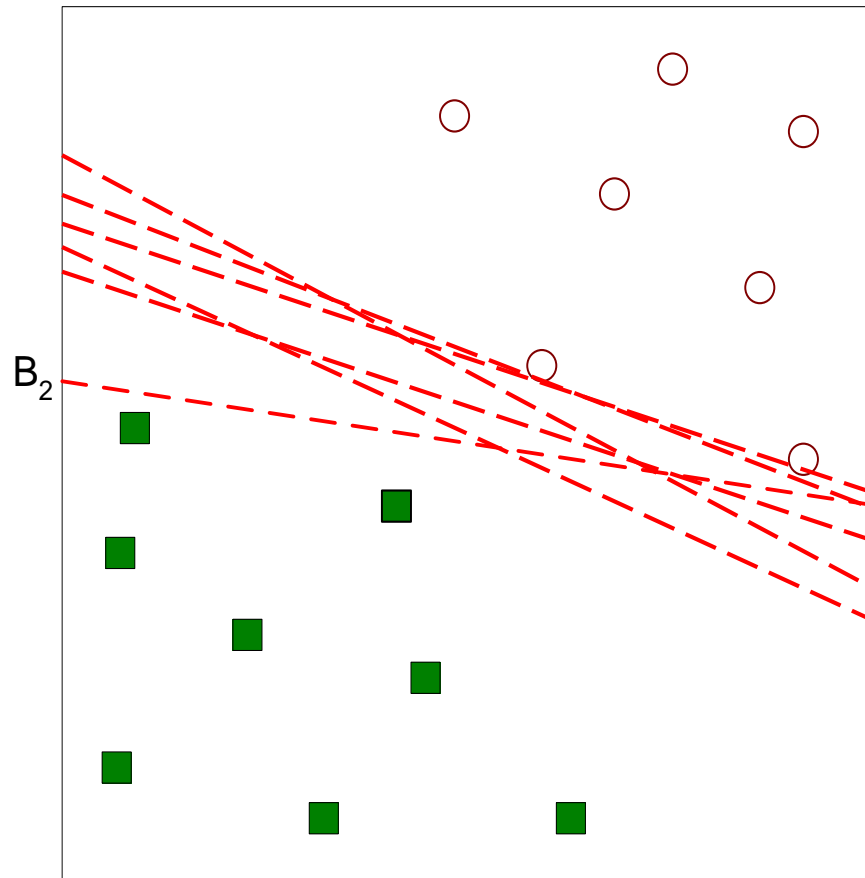
$\text{n}^\circ \text{ de pixels} * \text{n}^\circ \text{ de escalas consideradas} * \text{complexidade de avaliação do classificador}$

Support Vector Machines (SVM)



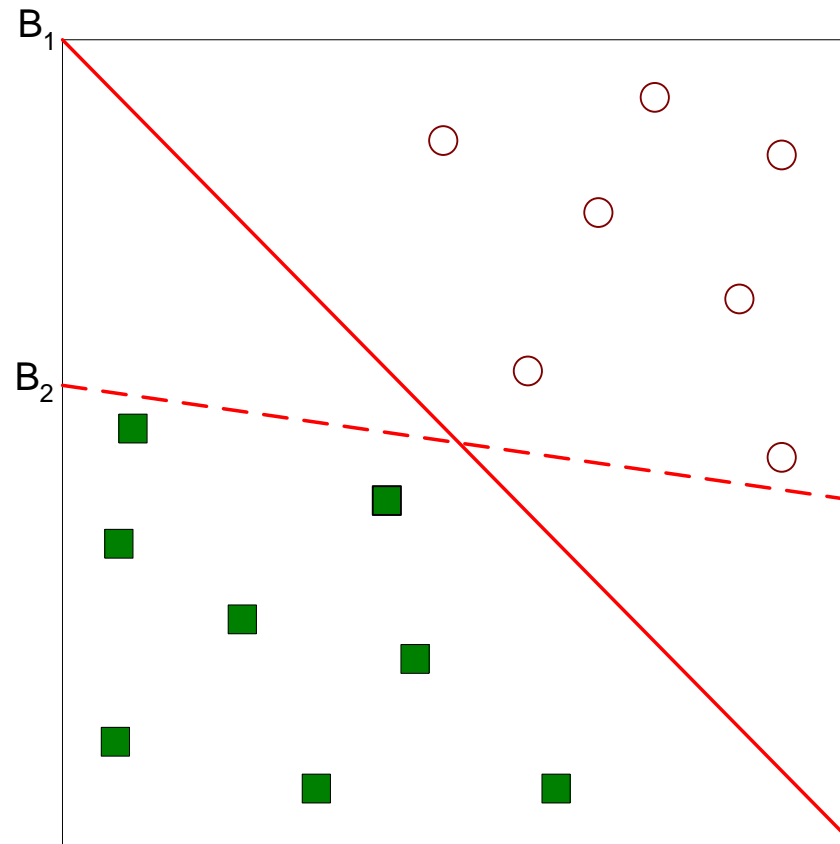
Como podemos separar essas duas classes?

Support Vector Machines (SVM)



Várias soluções possíveis.

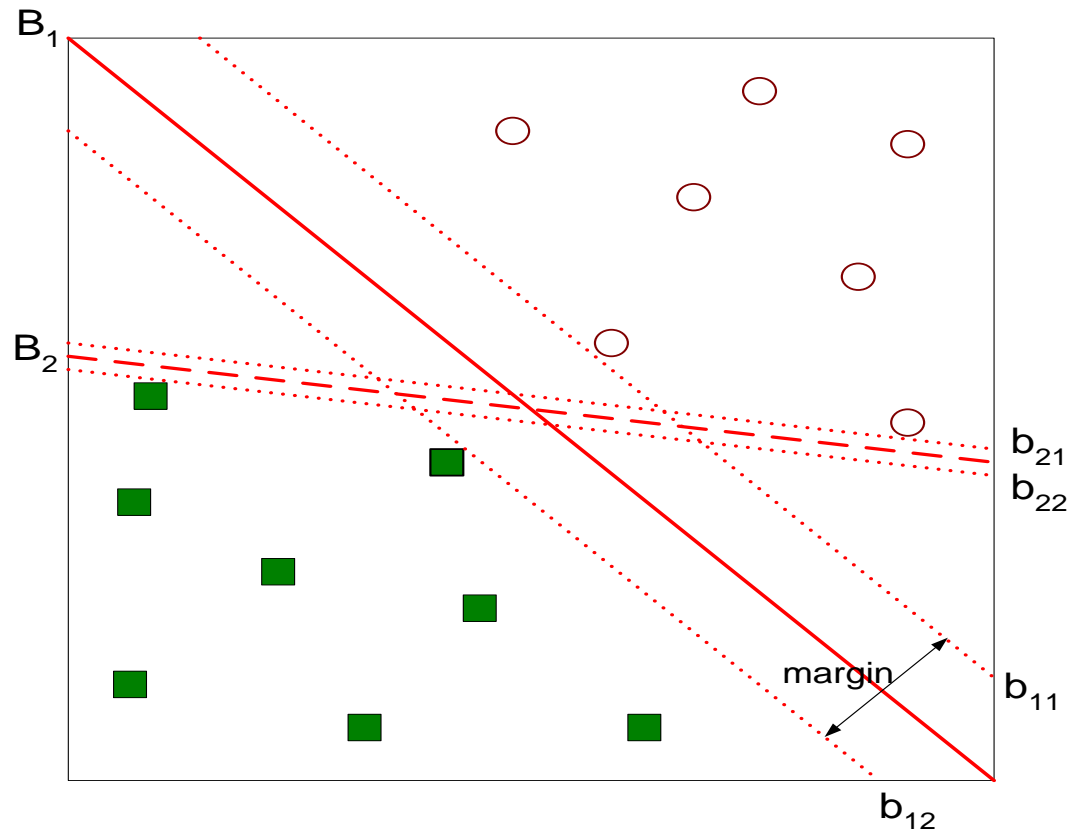
Support Vector Machines (SVM)



- Qual das duas retas divide melhor as classes? B_1 ou B_2 ?
- Como podemos definir o que é melhor?

Support Vector Machines (SVM)

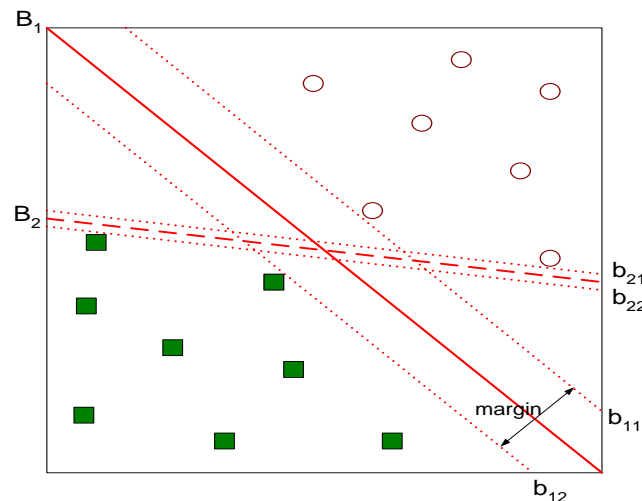
Qual dos dois métodos é mais seguro essa separação?



Encontra o hiperplano que **maximiza** a margem \Rightarrow B1 é melhor que B2

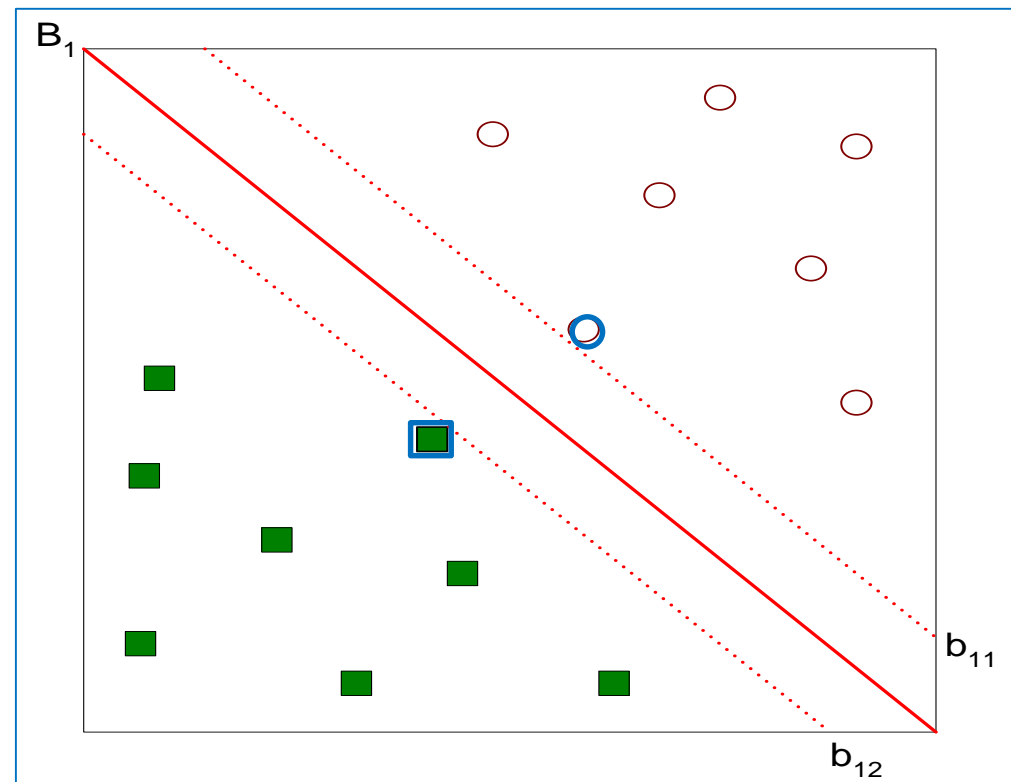
Support Vector Machine (SVM)

- Máquina de Vetores de Suporte
- Procura encontrar a maior margem para separar diferentes classes de dados.
- 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.



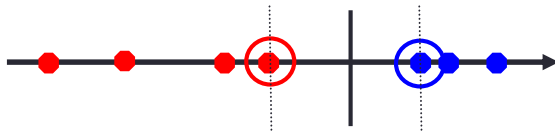
Vetores de Suporte (Support Vectors)

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

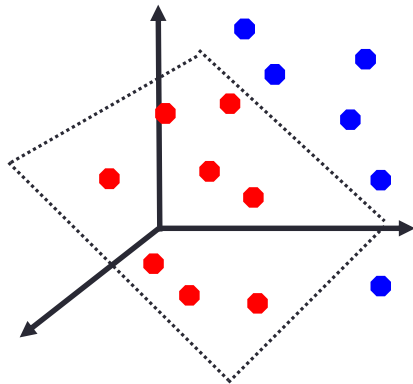


Support Vector Machine (SVM)

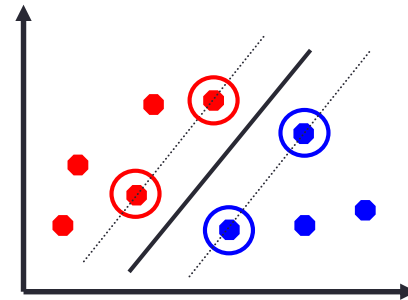
- Hiperplano:
 - Espaço 1D = Ponto



- Espaço 3D = Plano

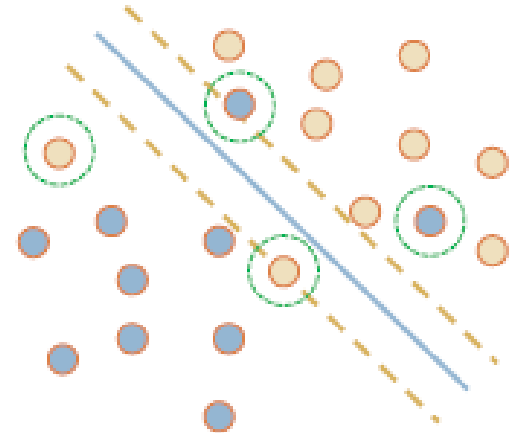


Espaço 2D = Reta

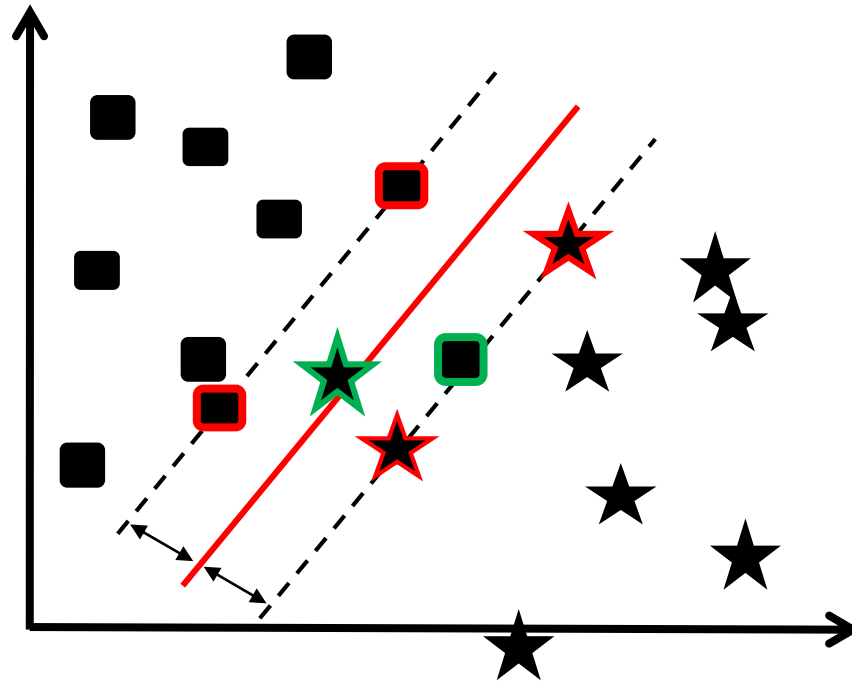


Support Vector Machine (SVM)

- A aplicação de um método puramente linear para classificar um conjunto de dados pode sofrer com **dois problemas**:
 - Outliers (Presença de ruídos e exemplos inconsistentes)
 - Exemplos rotulados erroneamente
- O SVM ainda assim pode ser aplicado através do uso de variáveis de folga (**Soft Margin**)



Soft Margin



- É utilizada quando não há um hiperplano que divida os exemplos exatamente em -1 e +1 (acontece em boa parte dos casos)
- Permite que alguns dados possam violar a restrição, porém há uma penalização para estes casos.

Support Vector Machine (SVM)

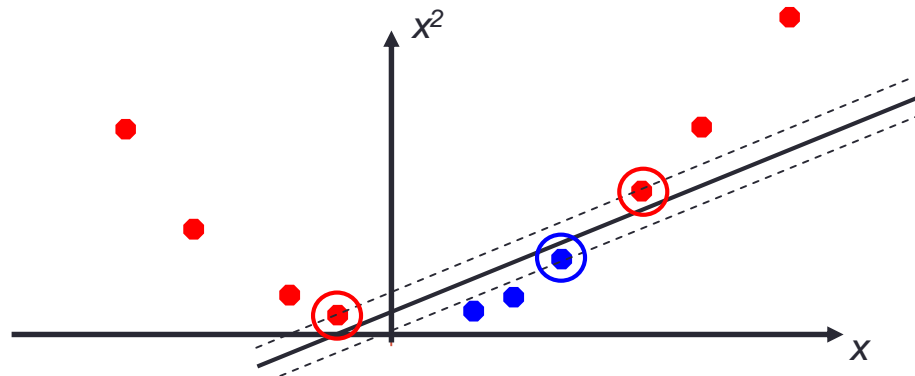
- Em alguns problemas não é possível separar as classes linearmente mesmo utilizando a margem de folga.
- Grande maioria dos problemas reais não são separáveis linearmente.
- O que fazer quando os dados não são linearmente separáveis?

SVM Não-Linear

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

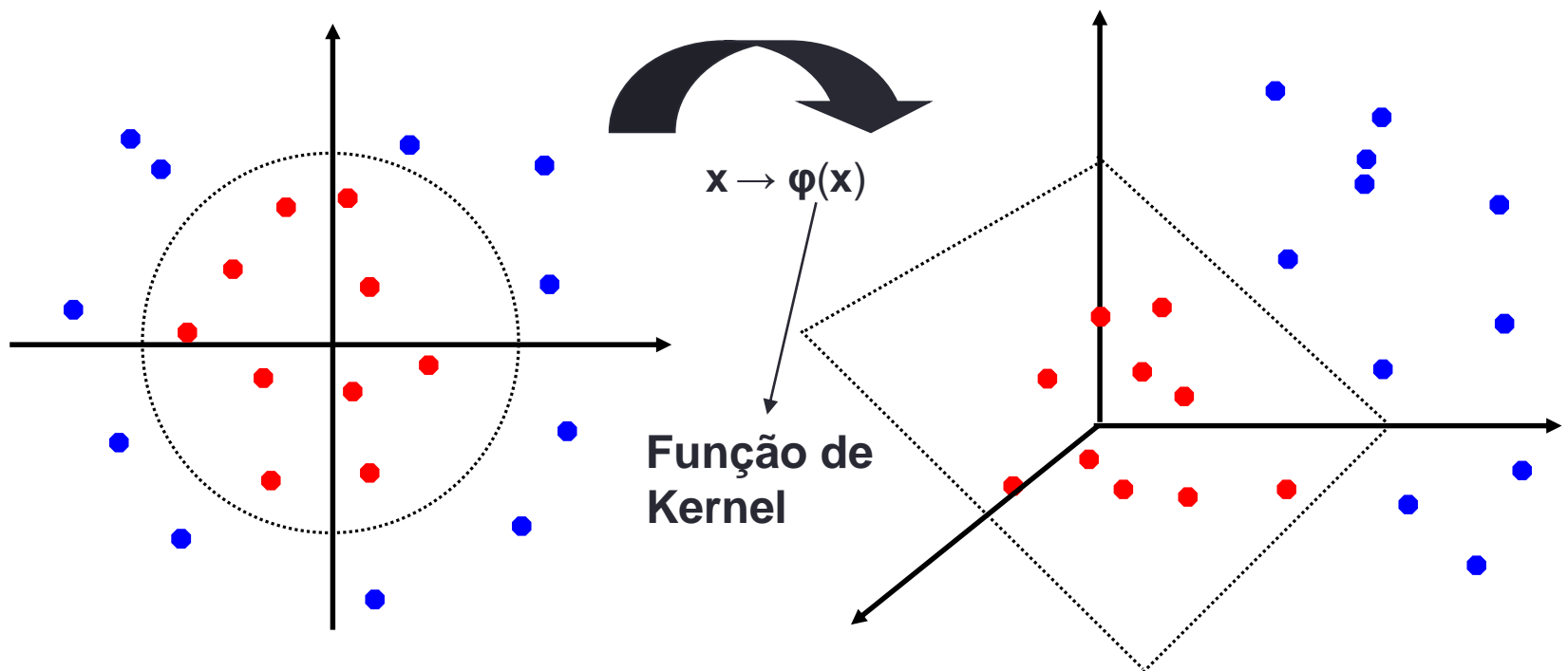


- A abordagem utilizada pelo SVM para resolver esse tipo de problema consiste em mapear os dados para um espaço de dimensão maior:



SVM Não-Linear

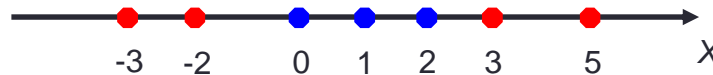
O espaço de características (feature space) original pode ser mapeado em um espaço de características de dimensão maior onde o conjunto de treinamento é linearmente separável:



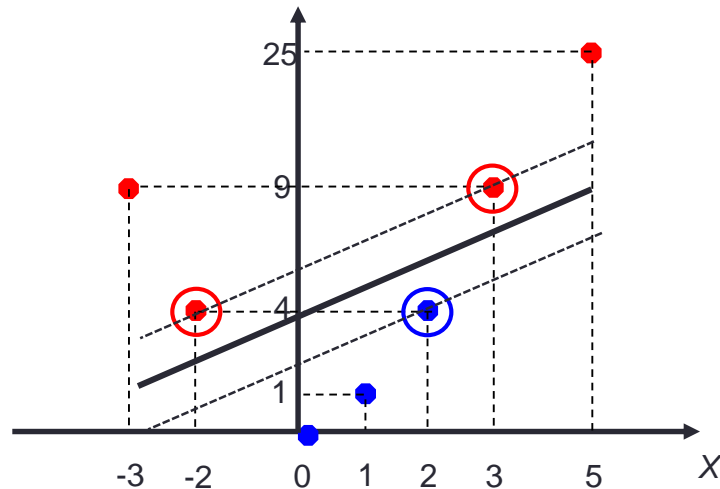
Kernel Trick

SVM Não-Linear Exemplo

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



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

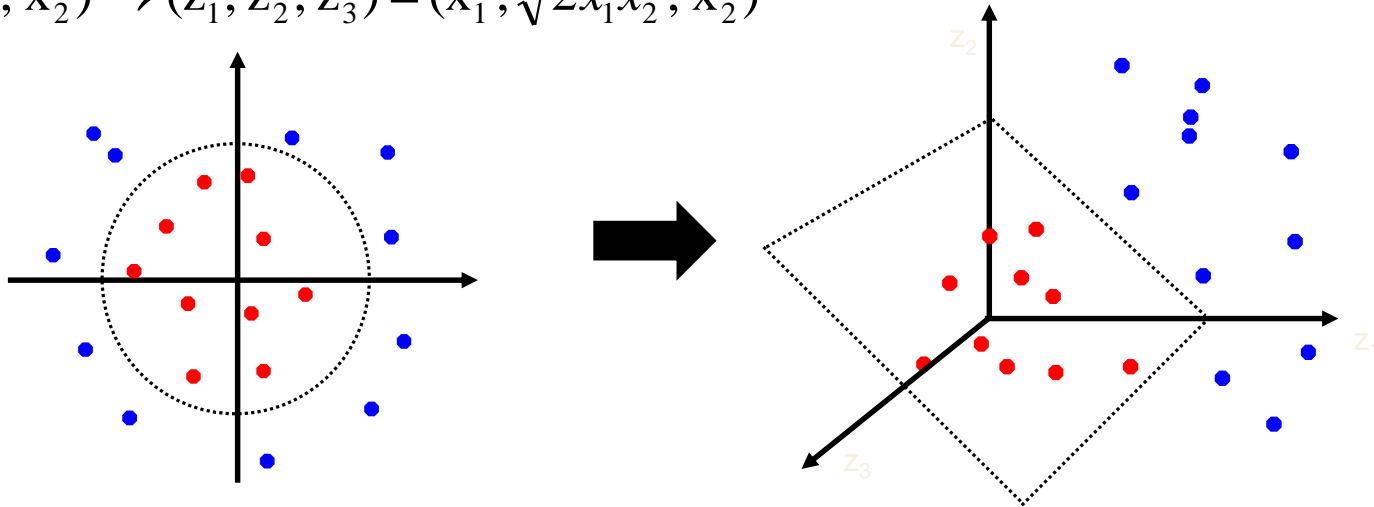


- Kernel:** $\varphi(x) = (x, x^2)$

SVM Não-Linear Exemplo

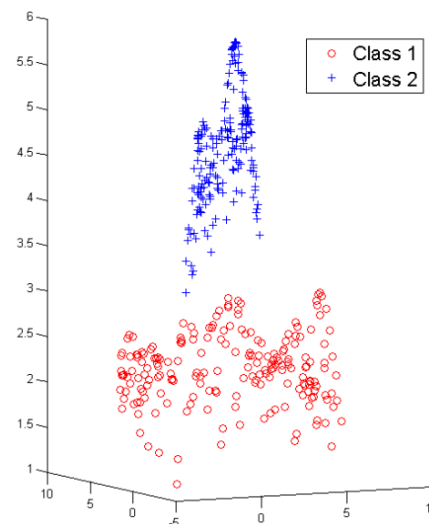
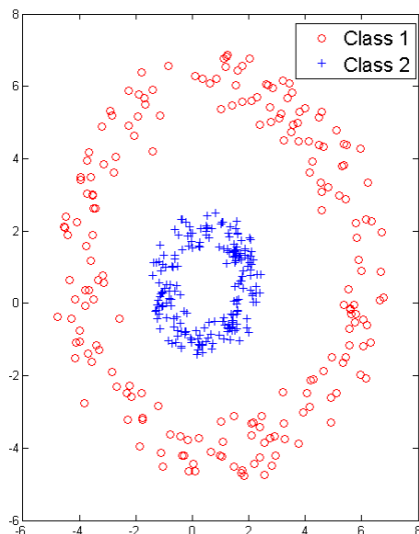
- A mesma metodologia pode ser aplicada em um espaço 2D de características ($\mathbb{R}^2 \rightarrow \mathbb{R}^3$).
- A única diferença é a necessidade de uma nova função de kernel. Um exemplo de função de kernel aplicável nesse caso seria:

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



Funções de Kernel

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



Support Vector Machine (SVM)

- O SVM originalmente lida com **classificações binárias** **$(-1,+1)$** .
- 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 (one-versus-all).
 - Pairwise (one-versus-one)

SVM Multi-Classes

- Um contra o resto (one-versus-all):
 - **Estratégia o vencedor leva tudo:** o classificador com a função de saída mais elevada será atribuído a classe.
- Pairwise (one-versus-one)
 - **Estratégia max-vitórias votantes:** a classe com a maioria dos votos é escolhida.

Aplicação

- Antes de aplicar um SVM para classificar um conjunto de dados é necessário responder algumas questões:
 - Quais funções de kernel utilizar?
 - Qual o valor do parâmetro para a Soft Margin?
- Validações cruzadas (cross-validations)
 - Holdout
 - K-fold
 - Leave-one-out

Validações Cruzadas (Cross-Validations)

- **Holdout:**

- Divide o grupo de dados em 2
 - uma parte para treinamento e outra para teste
- É comum considerar $2/3$ dos dados para treinamento e o $1/3$ para teste

- **K-fold: (Mais utilizado!)**

- Divide o DataSet em k subconjuntos do mesmo tamanho
- Então, um subconjunto é utilizado para teste e os k restantes são utilizados para treinamento.
- Este processo é realizado k vezes alternando de forma circular o subconjunto de teste.

- **Leave-one-out: (alto custo computacional!)**

- Caso específico do k-fold
- k igual ao número total de dados N.
- São realizados N cálculos de erro, um para cada dado.

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 longo dependendo do número de exemplos e dimensionalidade dos dados.
- Não funciona bem em conjunto de dados com grande quantidade de ruídos.

Software e DataSets Disponíveis

LIBSVM: <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>

Python: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC>

- DataSets para Estudo/Exemplo:
 - LIBSVM Data: Classification, Regression, and Multi-label
<https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>
 - Machine Learning Repository
<http://archive.ics.uci.edu/ml/datasets.html>