Support Vector Machines - SVM

M. Fernanda P. Costa

Departamento de Matemática
Universidade do Minho

Outline

Introdução

Máquinas de Vetores de Suporte (*Support Vector Machines* (SVMs)) é um dos algoritmos de classificação, teoricamente mais bem motivados e mais eficazes em aprendizagem automática (machine learning).

▷ Pode produzir classificadores lineares (SVM linear) ou classificadores não lineares (SVM com kernel).

▷ Muito eficaz na prática:

- Dá boa generalização para os casos de teste.
- O problema de otimização é convexo e tem uma única solução - solução global.
- Pode ser treinado em conjuntos de dados de grande dimensão.
- Kernels especiais podem ser definidos para muitas aplicações.

▷ Pode ser estendido para além da classificação para resolver problemas de regressão, redução de dimensionalidade, deteção de outlier, entre outros.

Classificação binária: Caso linearmente separável

▷ Seja $D = \{(x^n, y^n)_{n=1}^N\}$, um dataset de treino de classificação binária

- $x^{n\,T} = (x_1^n, \ldots, x_d^n) \in \mathbb{R}^d$: vector dos atributos
- $y^n \in \{+1, -1\}$: *label*, denotando a classe positiva ou negativa.

▷ **Definição Hiperplano**: Chama-se hiperplano em $\mathbb{R}^d$, ao conjunto de todos os pontos $x \in \mathbb{R}^d$ que verificam a equação

$$w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + w_0 = 0$$

ou seja,

$$\underbrace{w^T x + w_0}_{p(x)} = 0$$

onde

- $w = (w_1, w_2, \ldots, w_d)^T \in \mathbb{R}^d$ é um vetor perpendicular ao hiperplano;
- $w_0 \in \mathbb{R}$ é um escalar, chamado viés;
- $p(x)$ é a função do hiperplano.

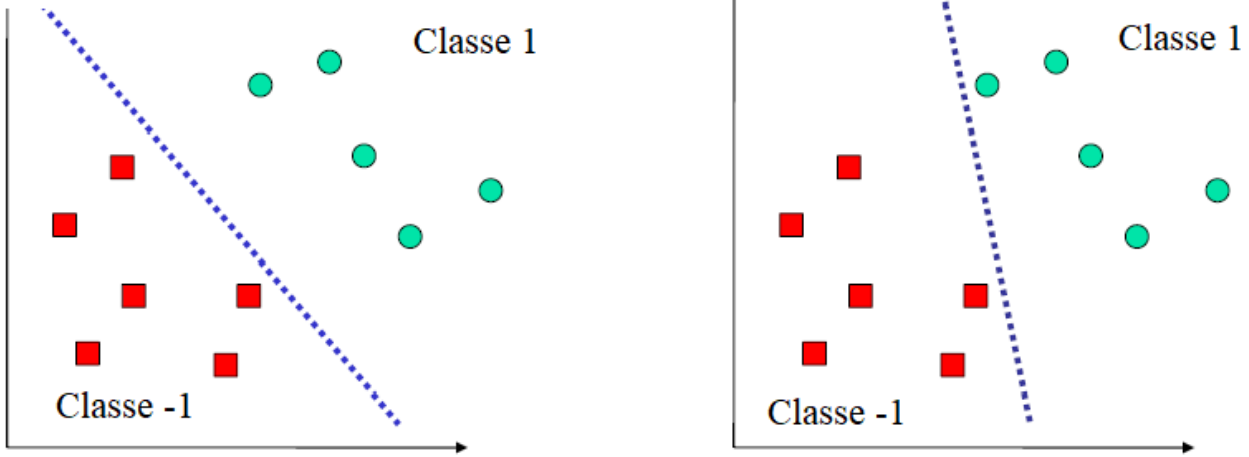
Vamos assumir que $D = \{(x^n, y^n)_{n=1}^N\}$ é linearmente separável.

▷ Se D é linearmente separável então existe um hiperplano

$$\underbrace{w^T x + w_0}_{p(x)} = 0, \quad w \in \mathbb{R}^d, \; w_0 \in \mathbb{R}$$

que separa corretamente os pontos de D das classes $+1$ e -1.

Exemplos



▷ O hiperplano representado por (w, w_0) divide o espaço dos atributos $\mathbb{R}^d$ em duas regiões:

$$w^T x + w_0 \geq 0$$
$$w^T x + w_0 < 0$$

▷ O classificador pode ser definido por uma função sinal

$$\hat{y}(x; w, w_0) = \text{sinal}(w^T x + w_0).$$

Aqui, $\hat{y}(x; w, w_0) = 1$ se $w^T x + w_0 \geq 0$, e $\hat{y}(x; w, w_0) = -1$ caso contrário.

▷ Os parâmetros (w, w_0) do hiperplano são obtidos pelo processo de aprendizagem, a partir dos dados de treino $D = \{(x^n, y^n)_{n=1}^N\}$, e tem de satisfazer as restrições

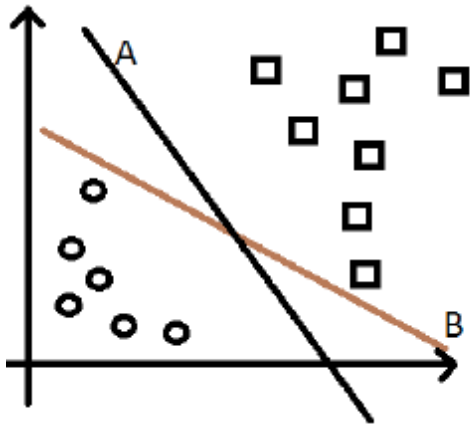
$$w^T x^n + w_0 \geq 0 \quad \text{se} \quad y^n = 1$$
$$w^T x^n + w_0 < 0 \quad \text{se} \quad y^n = -1$$

ou seja,

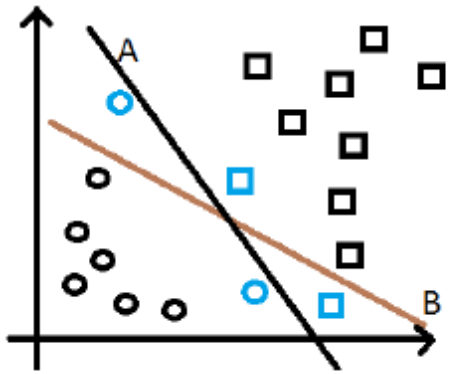
$$y^n (w^T x^n + w_0) \geq 0, \ \forall (x^n, y^n) \in D$$

▷ Se D é linearmente separável, existem vários hiperplanos que separam corretamente os dados de treino. Embora sejam todos igualmente bons no conjunto de treino, eles diferem com os dados do conjunto de teste. Qual deles será o melhor?

Exemplo:



a) hiperplanos A e B



b) conjunto de teste (dados a azul)

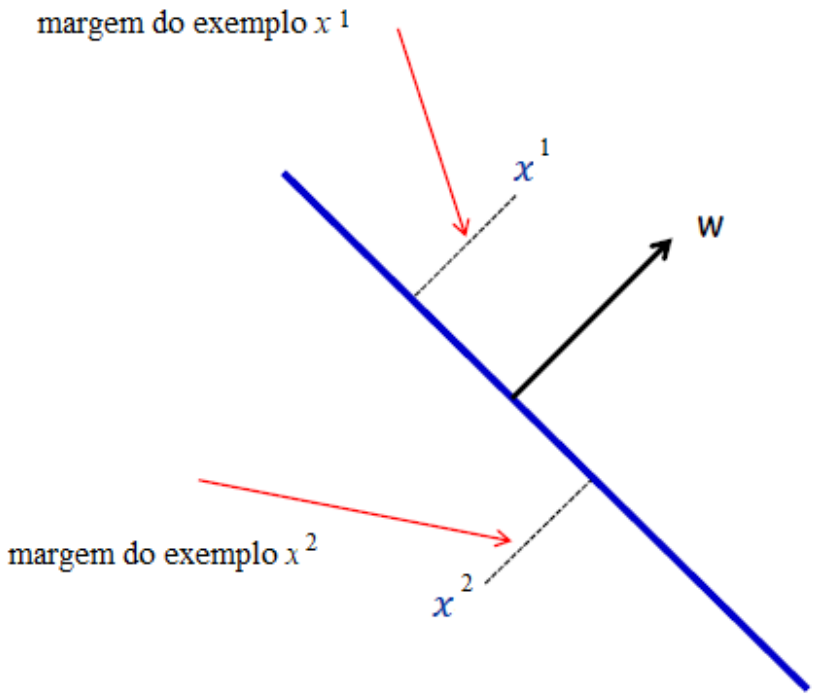
▷ Se os hiperplanos estão muito próximos de alguns pontos do conjunto de treino D, existe um maior risco de ocorrer erros de classificação com pontos $(x^i, y^i) \notin D$.

Margem geométrica

Definição: A margem de um ponto x^n ao hiperplano H, $w^T x + w_0 = 0$, é definida pela distancia do vetor x^n ao hiperplano H:

$$d(x^n, H) = \frac{|w^T x^n + w_0|}{\|w\|}$$

Exemplo:



Definição: A margem geométrica do hiperplano H, $w^T x + w_0 = 0$, ao conjunto de treino D é definida pela a menor das distancias dos vetores x^n de D ao hiperplano:

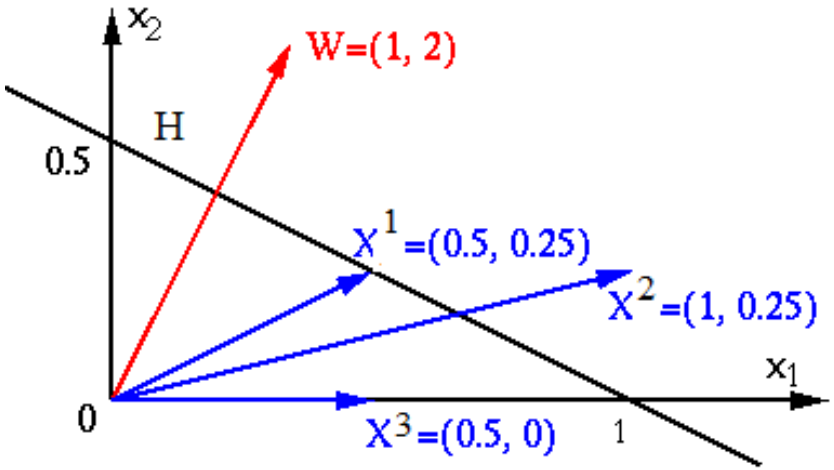
$$m = \min_{n=1...,N} d(x^n, H)$$

Observações:

- Todos os pontos x^n que atingem essa distância mínima são chamados de vetores de suporte para o hiperplano (vetores de suporte são os pontos mais próximos ao hiperplano).

- A margem m pode ser vista como uma medida de quanto bem um hiperplano H executa a tarefa de separar as duas classes binárias.
- Se o valor de m for pequeno significa que H está próximo de um ou mais pontos x^n de uma ou das duas classes. Portanto, existe uma probabilidade razoavelmente alta de que pontos não incluídos no conjunto de treino possam cair no lado errado da região de classificação.
- Por outro lado, se m for grande, essa probabilidade é significativamente reduzida.

Exemplo: Considere a seguinte figura e determine o hiperplano em $\mathbb{R}^2$, H.



Resolução: como $w^T = (1, 2)$ é perpendicular a H e $(0, 0.5) \in H$, então o hiperplano H é dada por:

$$w^T x + w_0 = 0 \quad \Leftrightarrow$$

$$(1, 2) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + w_0 = 0 \quad \Leftrightarrow$$

$$x_1 + 2x_2 + w_0 = 0$$

Usando agora o ponto $(0, 0.5)$, obtém-se $w_0 = -1$. Portanto, o hiperplano H é: $x_1 + 2x_2 - 1 = 0$

Função do hiperplano: $p(x) = x_1 + 2x_2 - 1$

Norma de w: $\|w\| = \sqrt{w_1^2 + w_2^2} = \sqrt{1^2 + 2^2} = \sqrt{5}$

Distância da origem $\mathbf{0} = (0, 0)$ a H: $d(\mathbf{0}, H) = \dfrac{|p(\mathbf{0})|}{\|w\|} = \dfrac{|-1|}{\sqrt{5}} = 0.447$

Distância de $x^1 = (0.5, 0.25)$, $x^2 = (1, 0.25)$ e $x^3 = (0.5, 0)$ a H:

- $p(x^1) = 0.5 + 2 \times 0.25 - 1 = 0$, i.é, x^1 pertence a H; a distância a H é

$$d(x^1, H) = \frac{|p(x^1)|}{\|w\|} = 0$$

- $p(x^2) = 1 + 2 \times 0.25 - 1 = 0.5 > 0$, i.é, x^2 está acima do hiperplano H; a distância a H é

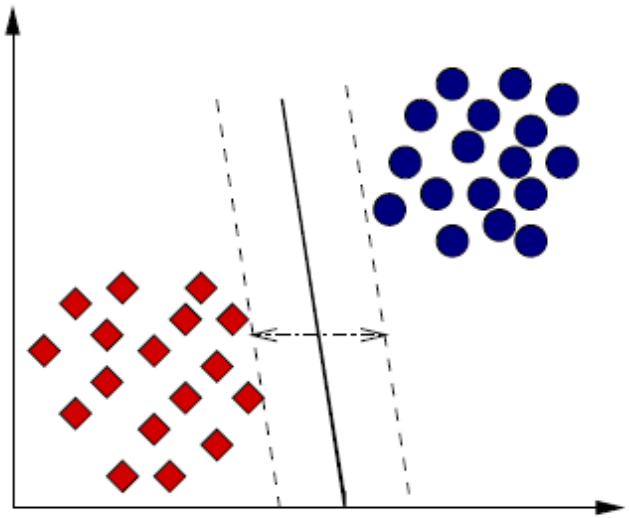
$$d(x^2, H) = \frac{|p(x^2)|}{\|w\|} = \frac{|0.5|}{\sqrt{5}} = 0.2235$$

- $p(x^3) = 0.5 + 0 - 1 = -0.5 < 0$, i.é, x^3 está abaixo do hiperplano H; a distância a H é

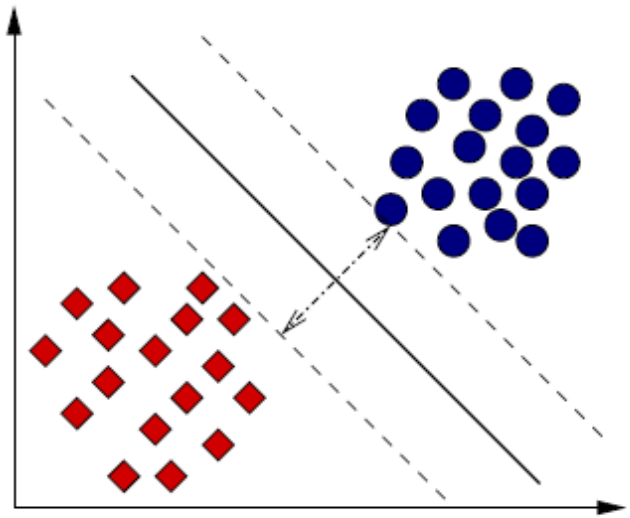
$$d(x^3, H) = \frac{|p(x^3)|}{\|w\|} = \frac{|-0.5|}{\sqrt{5}} = 0.2235$$

Hiperplano de margem máxima

▷ Um hiperplano $w^T x + w_0 = 0$ é considerado de Margem Máxima (ou de Separação Ótima) se separa as duas classes no conjunto de treino D e a distância entre os pontos x^n (das classes opostas) mais próximos ao hiperplano é máxima.



a) Hiperplano com margem pequena



b) Hiperplano com margem máxima

Hiperplano canónico

▷ Dado um hiperplano $w^T x + w_0 = 0$, é possível obter um número infinito de hiperplanos todos iguais multiplicando o hiperplano por um escalar $k \in \mathbb{R}$:

$$k[w^T x + w_0] = 0 \Leftrightarrow (kw)^T x + kw_0 = 0$$

▷ Para obter o Hiperplano único ou canónico, escolhemos o escalar k de modo que a distancia absoluta de um vetor de suporte $x^n \in D$ ao hiperplano seja 1. Ou seja,

$$k|w^T x^n + w_0| = 1 \;\Leftrightarrow\; k = \frac{1}{|w^T x^n + w_0|}$$

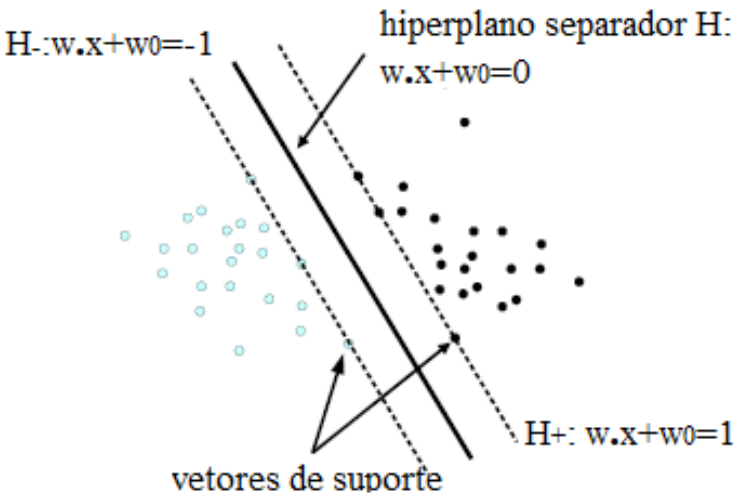
Definição: Um Hiperplano Canónico em relação ao de conjunto de treino D é aquele em que o vetor (w, w_0) é escalonado por forma a que os pontos x^n de D mais próximos ao hiperplano $w^T x + w_0 = 0$ satisfaçam a condição

$$|w^T x^n + w_0| = 1.$$

Tais vetores x^n são chamados de vetores de suporte.

Denota-se por H_+ e H_- os dois hiperplanos que são paralelos a H e que passam nos vetores de suporte em ambos os lados de H, e que são dados por:

$$H_+ : \quad w^T x + w_0 = 1$$
$$H_- : \quad w^T x + w_0 = -1$$



Distâncias dos hiperplanos à origem:

- $d(H, \mathbf{0}) = \dfrac{|w_0|}{\|w\|}$
- $d(H_+, \mathbf{0}) = \dfrac{|w_0 - 1|}{\|w\|}$
- $d(H_-, \mathbf{0}) = \dfrac{|w_0 + 1|}{\|w\|}$

- Distancia dos hiperplanos H_+ e H_- ao hiperplano H é:

$$|d(H_\pm, \mathbf{0}) - d(H, \mathbf{0})| = \left| \frac{|w_0 \pm 1|}{\|w\|} - \frac{|w_0|}{\|w\|} \right| = \frac{1}{\|w\|}$$

- Margem é $\dfrac{1}{\|w\|}$.

SVM

O objetivo de um SVM linear é encontrar o hiperplano ótimo H que separa as duas classes no conjunto de treino D, de tal forma que H tenha margem máxima aos vetores de suporte e que satisfaz as seguintes condições, para todo $(x^n, y^n) \in D$:

$$\begin{aligned} w^T x^n + w_0 &\geq 1 \quad \text{se} \quad y^n = 1 \\ w^T x^n + w_0 &\leq -1 \quad \text{se} \quad y^n = -1 \end{aligned}$$

Estas condições podem ser reescritas de modo compacto na forma:

$$y^n(w^T x^n + w_0) \geq 1 \quad \text{para} \quad n = 1 \ldots, N \tag{1}$$

Maximizar a margem $\frac{1}{\|w\|}$ é equivalente a minimizar $\|w\|$.

Problema de otimização (primal)

Portanto, o hiperplano separador H de margem máxima é dado pelo seguinte problema de otimização:

$$\begin{aligned} \underset{w\in\mathbb{R}^d, w_0\in\mathbb{R}}{\text{minimizar}} \quad & \frac{1}{2}\|w\|^2 \equiv \frac{1}{2}w^T w \\ \text{sujeito a} \quad & y^n(w^T x^n + w_0) \geq 1, \quad n = 1, \ldots, N \end{aligned} \qquad (\mathrm{P}_{primal})$$

Observações:

- (P_{primal}) é um **problema de Programação Quadrática (PQ)**: a função objetivo é uma função quadrática e as N funções de restrição são lineares.
- Como a função objetivo e as funções de restrição são funções convexas, o problema de otimização é convexo. Portanto, a sua solução ótima (w^*, w_0^*) é uma solução global!
- (w^*, w_0^*) pode ser obtido resolvendo (P_{primal}) por um método de otimização para PQ. Porém, é mais comum resolver o Problema dual para (P_{primal}).

A Função Lagrangiana associada ao problema (P_{primal}), é definida por

$$L(w, w_0, \alpha) = \frac{1}{2} w^T w - \sum_{n=1}^{N} \alpha_n \left[y^n (w^T x^n + w_0) - 1 \right]$$

onde $\alpha = (\alpha_1, \ldots, \alpha_N)^T$ é o vetor dos multiplicadores de Lagrange associados às restrição de desigualdade ($\geq$).

Notar que, as condições KKT para o problema (P_{primal}) são (n=1...,N):

$$\begin{aligned}
\nabla_{w,w_0} L(w, w_0, \alpha) &= 0 && \text{condição de 1ª ordem} \\
y^n (w^T x^n + w_0) - 1 &\geq 0 && \text{admissibilidade primal} \\
\alpha_n &\geq 0 && \text{admissibilidade dual} \\
\alpha_n \left(y^n (w^T x^n + w_0) - 1 \right) &= 0 && \text{condição de complementaridade}
\end{aligned}$$

Problema dual

Dado que o problema (P_{primal}) é convexo, o problema dual tem a seguinte forma:

$$\begin{aligned} \underset{w\in\mathbb{R}^d, w_0\in\mathbb{R}, \alpha\in\mathbb{R}^N}{\text{maximizar}} \quad & L(w, w_0, \alpha) \\ \text{sujeito a} \quad & \nabla_{w,w_0} L(w, w_0, \alpha) = 0 \\ & \alpha_n \geq 0 \end{aligned}$$

Pela condição de 1ªordem: $\nabla_{w,w_0} L(w, w_0, \alpha) = 0$ tem-se:

$$\nabla_w L(w, w_0, \alpha) = 0 \Leftrightarrow w - \sum_{n=1}^{N} \alpha_n y^n x^n = 0 \Leftrightarrow w = \sum_{n=1}^{N} \alpha_n y^n x^n \qquad (2)$$

$$\nabla_{w_0} L(w, w_0, \alpha) = 0 \Leftrightarrow \sum_{n=1}^{N} \alpha_n y^n = 0$$

Substituindo w na função Lagrangeana pela expressão dada na 1ª equação de (2), e simplificando-se obtém-se a função Lagrangeana expressa simplesmente em termos dos multiplicadores de lagrange:

$$L(w, w_0, \alpha) = \frac{1}{2} w^T w - w^T \underbrace{\left(\sum_{n=1}^{N} \alpha_n y^n x^n \right)}_{w} - w_0 \underbrace{\sum_{n=1}^{N} \alpha_n y^n}_{0} + \sum_{n=1}^{N} \alpha_n \quad \Leftrightarrow$$

$$L(w, w_0, \alpha) = -\frac{1}{2} w^T w + \sum_{n=1}^{N} \alpha_n \quad \Leftrightarrow$$

$$L(w, w_0, \alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y^n y^m \mathrm{x}^{n\,T} \mathrm{x}^m$$

Portanto, o problema dual para o problema (P_{primal}) é dado por:

Problema dual

$$\begin{array}{ll} \underset{\alpha\in\mathbb{R}^N}{\text{maximizar}} & \sum_{n=1}^{N}\alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y^n y^m x^{n\,T}x^m \equiv \mathbf{1}\alpha - \frac{1}{2}\alpha^T\mathbf{Q}\alpha \\ \text{sujeito a} & \sum_{n=1}^{N}\alpha_n y^n \equiv \mathrm{Y}\alpha = 0 \\ & \alpha_n \geq 0,\ \ n = 1,\ldots,N \end{array}$$

(P_{dual})

- $\mathbf{Q}$ é uma matriz simétrica $N \times N$ em que os seus elementos são dados por $Q(n,m) = y^n y^m \mathbf{x}^{n\,T}\mathbf{x}^m;\ \ (n,m = 1,\cdots,N)$;
  $\mathbf{Y}_{1\times N} = (y^1, y^2, \ldots, y^N);\ \ \mathbf{1}_{1\times N} = (1,1,\ldots,1)$
- na forma matricial: $\mathrm{Q}_{N\times N} = (\mathrm{Y}^{\mathrm{T}}\mathrm{Y}).*(\mathrm{X}^{\mathrm{T}}\mathrm{X})$, onde

$$\mathrm{X} = \begin{pmatrix} x_1^1 & x_1^2 & \ldots & x_1^N \\ x_2^1 & x_2^2 & \ldots & x_2^N \\ \vdots & \vdots & \ldots & \vdots \\ x_d^1 & x_d^2 & \ldots & x_d^N \end{pmatrix}$$

$.*$ representa o produto elemento a elemento de matrizes

Observações:

▷ O problema (P_{dual}) é um problema de PQ, o qual pode ser resolvido por um:

- método de otimização de minimização para PQ . Para tal basta reescrever o problema dual na forma:

$$\begin{array}{ll} \text{- minimizar}_{\alpha \in \mathbb{R}^N} & +\frac{1}{2}\alpha^T Q \alpha - \mathbf{1}\alpha \\ \text{sujeito a} & Y\alpha = 0 \\ & \alpha_n \geq 0, \ \ n = 1, \ldots, N \end{array}$$

ou

- método de otimização especificamente desenvolvido no contexto do SVM para resolver o problema dual. Destes, um dos mais conhecido é o "Sequential Minimal Optimization (SMO) algorithm".

▷ Calculada a solução ótima $\alpha^* = (\alpha_1^*, \ldots, \alpha_N^*)^T$ de (P_{dual}), $(n = 1, \ldots, N)$

- $\alpha_n^* > 0 \Rightarrow x^n$ é vetor de suporte;
- $\alpha_n^* = 0 \Rightarrow x^n$ não é vetor de suporte (irrelevante);

a seguir e usando a equação (2), obtém-se a solução ótima w^*:

$$w^* = \sum_{n=1}^{N} \alpha_n^* y^n x^n \Leftrightarrow w^* = \sum_{\alpha_n^* > 0} \alpha_n^* y^n x^n \tag{3}$$

▷ Tendo-se w^* e dado que qualquer vetor de suporte x^n verifica:

$$w^{*T} x^n + w_0 = 1 \text{ ou } w^{*T} x^n + w_0 = -1 \Leftrightarrow$$

$$\Leftrightarrow y^n (w^{*T} x^n + w_0) = 1 \Leftrightarrow w^{*T} x^n + w_0 = y^n$$

(note que $(y^n)^2 = 1$). Resolvendo esta equação para w_0, obtém-se

$$\begin{aligned} w_0^* &= y^n - w^{*T} x^n \Leftrightarrow \\ w_0^* &= y^n - \sum_{\alpha_m^* > 0} \alpha_m^* y^m x^{mT} x^n \end{aligned} \tag{4}$$

Nota: Seja n_{sv} o número de vetores de suporte. Em (4), todos os vetores de suporte produzem o mesmo resultado. Na prática, simplesmente usamos todos vetores de suporte e fazemos a média.

$$w_0^* = \frac{\sum_{\alpha_n^*>0} \left(y^n - w^{*T} x^n\right)}{n_{sv}} = \frac{\sum_{\alpha_n^*>0} \left(y^n - \sum_{\alpha_m^*>0} \alpha_m^* y^m x^{mT} x^n\right)}{n_{sv}} \tag{5}$$

Usando notação vetorial:

$$w_0^* = mean\left(Y_{sv} - (\alpha_{sv}. * Y_{sv})\left(X_{sv}^T X_{sv}\right)\right) \tag{6}$$

- X_{sv} matriz $d \times n_{sv}$ dos vetores de suporte
- Y_{sv} vetor $1 \times n_{sv}$ das labels associadas aos vetores de suporte
- α_{sv} vetor $1 \times n_{sv}$ dos alfas associados aos vetores de suporte
- *mean* - dá a média do vetor

Finalmente, obtido o plano separador ótimo H, $w^{*T}x + w_0^* = 0$, o classificador é dado pela **função sinal**. Assim, para um novo vetor de atributos $z \in \mathbb{R}^d$, a classe prevista para z, y_p, é:

$$\begin{aligned} y_p = & \quad \text{sinal}(w^{*T}z + w_0^*) = \\ = & \quad \text{sinal}\left(\sum_{\alpha_n^* > 0} \alpha_n^* y^n \mathrm{x}^{nT}\mathrm{z} + w_0^*\right) \end{aligned} \tag{7}$$

Usando notação vetorial:

$$y_p = \quad \text{sinal}\left(w_0^* + (Y_{sv}.*\alpha_{sv})\,(X_{sv}^T z)\right) \quad \rightarrow \begin{cases} +1, & z \in \{+1\} \\ -1, & z \in \{-1\} \end{cases} \tag{8}$$

▷ Notar que, após o cálculo de α^*, o classificador depende apenas do produto interno entre z e os vetores de suporte do conjunto de treino D.

Algoritmo 1: Algoritmo SVM (dual)

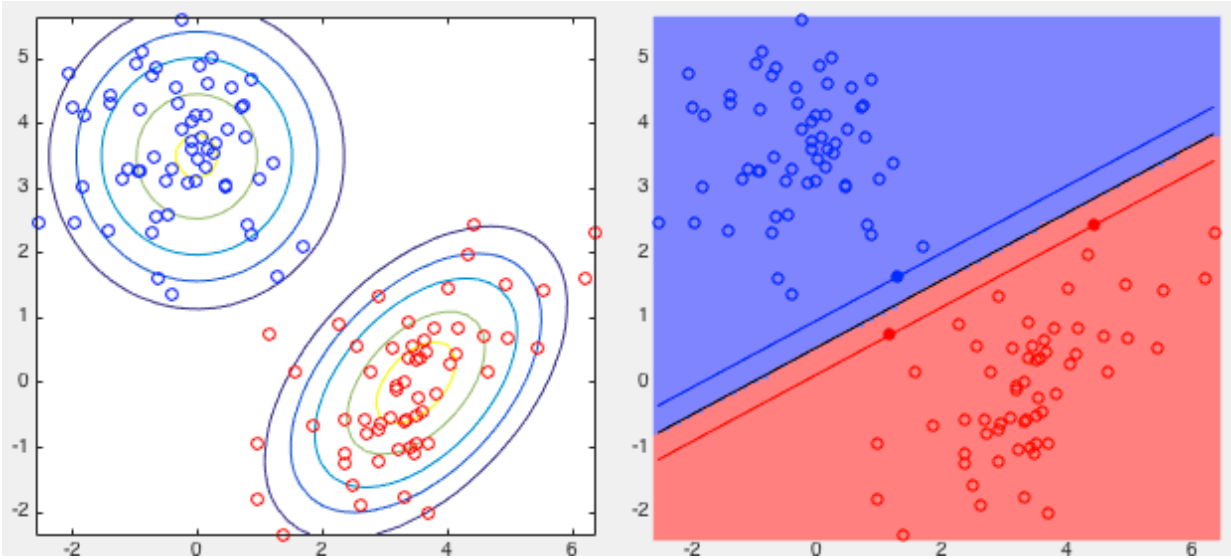
1. Dar: $D = (x^n, y^n)_{n=1}^N$, $x^n \in \mathbb{R}^d$, $y^n \in \{-1, 1\}$,
2. Fazer: $X_{d\times N} = \begin{pmatrix} x_1^1 & x_1^2 & \dots & x_1^N \\ x_2^1 & x_2^2 & \dots & x_2^N \\ \vdots & \vdots & \dots & \vdots \\ x_d^1 & x_d^2 & \dots & x_d^N \end{pmatrix}$; $\mathrm{Y}_{1\times \mathrm{N}} = \left(y^1, y^2, \dots, y^N\right)$
3. Formar a matriz $\mathrm{Q}_{N\times N} = (\mathrm{Y}^{\mathrm{T}}\mathrm{Y}) .* (\mathrm{X}^{\mathrm{T}}\mathrm{X})$
4. Aplicar um método de otimização de minimização para PQ para resolver o problema dual:

$$\begin{aligned} \text{- } \underset{\alpha\in\mathbb{R}^{\mathrm{N}}}{\text{minimizar}} \quad & +\frac{1}{2}\alpha^T \mathrm{Q}\alpha - \mathbf{1}\alpha \\ \text{sujeito a} \quad & \mathrm{Y}\alpha = 0 \\ & \alpha_n \geq 0, \ \ n = 1, \dots, N \end{aligned}$$

   $\Rightarrow$ obtém-se a solução ótima $\alpha^* = \left(\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*\right)^T$
5. Identificar os vetores de suporte: $\alpha_n^* > 0 \Rightarrow \mathrm{x}^n$ é vetor de suporte
6. Formar: X_{sv}, Y_{sv}, α_{sv}
7. Calcular w_0^* usando (6), e definir o classificador usando (8)

Exemplo:

A figura abaixo à esquerda mostra o conjunto de treino $\{(x^n, y^n)\}$, $x^n \in \mathbb{R}^2$ e $y^n \in \{-1, 1\}$, gerados aleatoriamente por uma distribuição Gaussiana. A figura à direita mostra o hiperplano ótimo obtido pelo algoritmo SVM dual.



O hiperplano ótimo é determinado pelos seguintes três vetores de suportes.

n	α_n^*	x^n	y^n
40	0.52	$(4.43, 2.44)^T$	-1
52	3.11	$(1.17, 0.74)^T$	-1
103	3.64	$(1.30, 1.64)^T$	1

Escreva o hiperplano ótimo obtido pelo algoritmo SVM.

Resolução:

- Calcular os parâmetros $w = (w_1, w_2)^T$ do hiperplano, usando (3):

$$\begin{aligned} w &= \sum_{\alpha_n > 0} \alpha_n y^n x^n \\ &= -0.52 \begin{pmatrix} 4.43 \\ 2.44 \end{pmatrix} - 3.11 \begin{pmatrix} 1.17 \\ 0.74 \end{pmatrix} + 3.64 \begin{pmatrix} 1.30 \\ 1.64 \end{pmatrix} \\ &= \begin{pmatrix} -1.2103 \\ 2.3994 \end{pmatrix} \end{aligned}$$

Calcular w_0 como a média dos valores de w_0 obtidos a partir de cada vetor de suporte, usando (5):

n	x^n	y^n	$w^T x^n$	$w_0 = y^n - w^T x^n$
40	$(4.43,2.44)^T$	-1	0.4929	-1.4929
52	$(1.17,0.74)^T$	-1	0.3595	-1.3595
103	$(1.30,1.64)^T$	1	2.3616	-1.3616
				$mean(w_0) = -1.4047$

e portanto $w_0 = -1.4047$

- Hiperplano: $w^T x + w_0 = 0 \Leftrightarrow -1.2103 x_1 + 2.3994 x_2 - 1.4047 = 0$

Vantagem da formulação dual

$$\begin{aligned} \underset{\alpha \in \mathbb{R}^N}{\text{maximizar}} \quad & \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y^n y^m \mathbf{x}^{n\,T}\ \mathbf{x}^m \\ \text{sujeito a} \quad & \sum_{n=1}^{N} \alpha_n y^n = 0 \\ & \alpha_n \geq 0, \ \ n = 1, \ldots, N \end{aligned}$$

Vantagem:

- no problema dual, os vetores dos atributos aparecem apenas num produto interno; e
- fomos capazes de escrever todo o algoritmo SVM em termos de apenas produtos internos entre vetores de atributos!
- Esta vantagem, permite que seja possível aplicar Kernels ao problema de classificação!

Kernels

▷ O algoritmo SVM descrito na secção anterior converge apenas se o conjunto de treino $D = \{(x^n, y^n)_{n=1}^N\}$, com $x^n \in \mathbb{R}^d$ e $y^n \in \{-1, 1\}$, é linearmente separável.

▷ Se D não é linearmente separável, é possível usar um método de Kernel para mapear todos os pontos x^n $(n = 1, \ldots, N)$ do **espaço dos atributos original** para um **novo espaço de atributos** de maior dimensão:

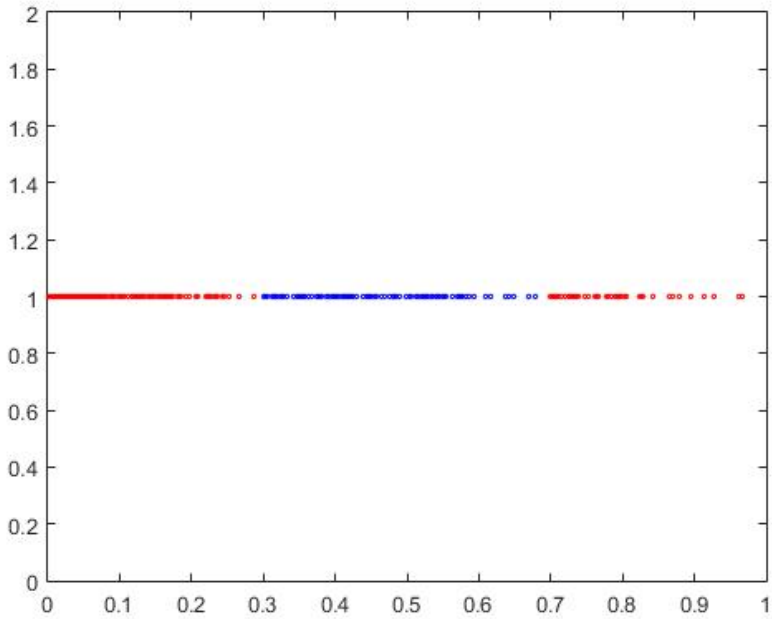
$$x^n \Longrightarrow \phi(x^n)$$

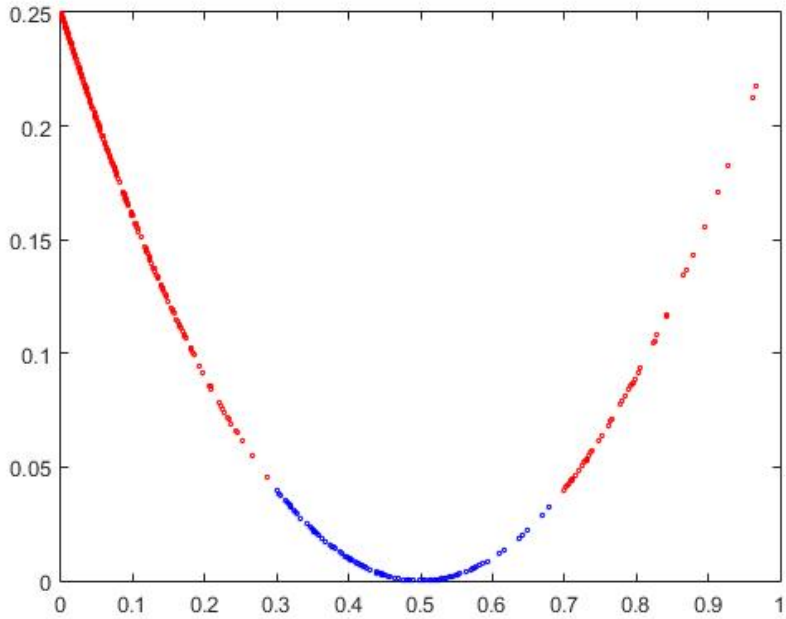
em que as duas classes se tornam linearmente separáveis.

Exemplo:

Em $\mathbb{R}$, as classes $C_- = \{x \in \mathbb{R} : a \leq x \leq b\}$ e $C_+ = \{x \in \mathbb{R} : x \leq a \text{ ou } x \geq b\}$ não são linearmente separáveis. Pelo mapeamento seguinte $\phi : \mathbb{R} \to \mathbb{R}^2$

$$x \in \mathbb{R} \Longrightarrow \phi(x) = \left[x, (x - (a+b)/2)^2\right]^T$$



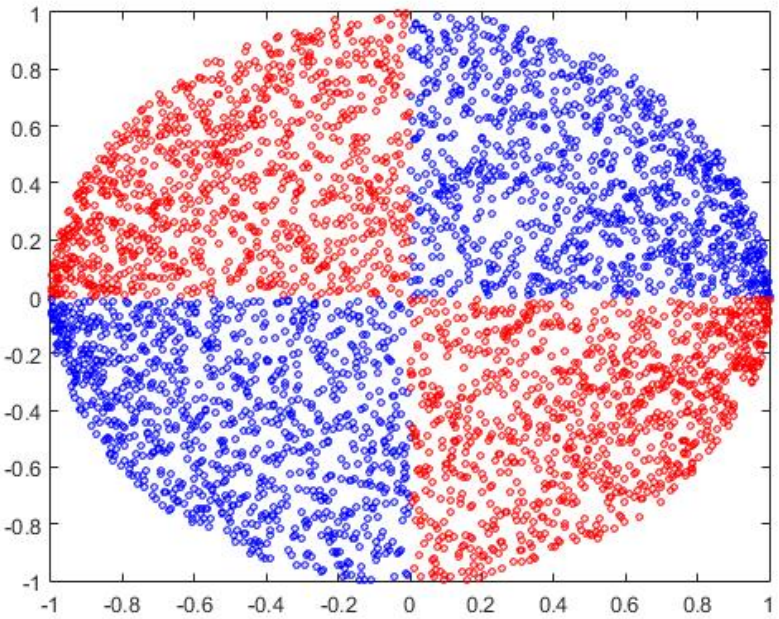


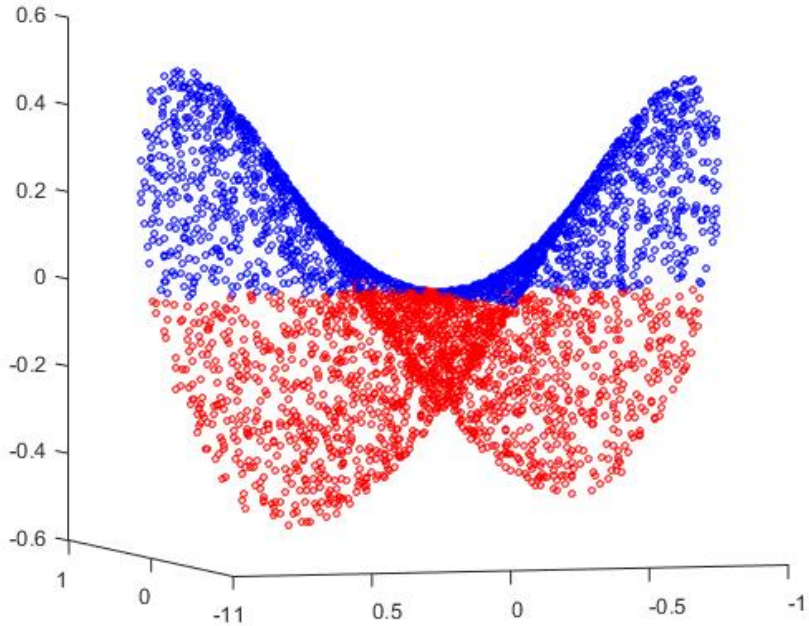
as duas classes podem ser linearmente separadas no espaço $\mathbb{R}^2$.

Exemplo:

No espaço $\mathbb{R}^2$, o conjunto de dados do XOR, a classe C_- contendo os pontos nos quadrantes I e III e a classe C_+ contendo pontos nos quadrantes II e IV, não são linearmente separáveis. Pelo mapeamento seguinte $\phi : \mathbb{R}^2 \to \mathbb{R}^3$

$$x \in \mathbb{R}^2 \Longrightarrow \phi(x) = [x_1, x_2, x_1 x_2]^T$$





as duas classes podem ser linearmente separadas no espaço $\mathbb{R}^3$.

Seja χ o espaço dos atributos $x^1, x^2, \ldots, x^N$, e seja $\mathcal{F}$ o novo espaço dos atributos $\phi(x^1), \phi(x^2), \ldots, \phi(x^N)$ dado pelo mapeamento $\phi : \mathbb{R}^d \to \mathcal{F}$.

Definição:

Um *Kernel* é uma função K, tal que para todo $x, z \in \chi$ retorna o produto interno das suas imagens no espaço $\mathcal{F}$:

$$K(x, z) = \phi(x)^T \phi(z)$$

Observações

- Na prática, a função ϕ não precisa de ser explicitamente especificada, e a sua dimensão nem precisa ser conhecida, uma vez que está implicitamente definida pela escolha do Kernel: $K(x, z)$.

Apresentam-se alguns kernels $K(x,z) = \phi(x)^T \phi(z)$, existentes na literatura, mais usados:

- Kernel Linear (sem mapeamento): $K(x,z) = x^T z$
- Kernels polinomiais:
  - $K(x,z) = (x^T z)^m; \;\; m \in \{2,3,\dots\}$
  - $K(x,z) = (x^T z + c)^m; \;\; c \in \mathbb{R}$
- Kernel Gaussiano/Kernel RBF (radial base function):

$$K(x,z) = e^{-\|x-z\|^2/2\sigma^2} = e^{-\gamma\|x-z\|^2}; \;\; \gamma = 1/2\sigma^2$$

- Kernel Sigmoidal: $K(x,z) = \tanh\left(\beta x^T z + b\right); \;\; \beta, b \in \mathbb{R}$
- ...

Exemplo:
Quando $m = 2$ e $d = n$, o kernel polinomial definido sobre os vetores $x, z \in \mathbb{R}^n$ é:

$$\begin{aligned} K(x,z) &= (x^T z)^2 \\ &= (x_1 z_1 + \cdots + x_n z_n)^2 \\ &= \left(\sum_{i=1}^n x_i z_i\right)\left(\sum_{j=1}^n x_j z_j\right) \\ &= \sum_{i,j=1}^n (x_i x_j)(z_i z_j) \\ &= \phi(x)^T \phi(z) \end{aligned}$$

Assim, $\phi(x)$ é dado por (mostra-se aqui para o caso de n=3):

$$\phi(x) = (x_1^2, x_1x_2, x_1x_3, x_2x_1, x_2^2, x_2x_3, x_3x_1, x_3x_2, x_3^2)^T$$

(mapeamento do espaço $\mathbb{R}^3$ para o espaço $\mathbb{R}^9$).

Observação:
Calcular $\phi(x)$ requer n^2 operações, calcular $K(x,z)$ precisa apenas de n.

SVM com função Kernel

O método do kernel pode ser aplicado ao algoritmo SVM, pois todos os pontos de dados exibidos no algoritmo estão na forma de um produto interno. Assim, durante o processo de treino, substituímos o produto interno $x^{nT}x^m$ em (P_{dual}) e (5) pela função kernel $K(x^n, x^m)$:

Problema dual com Kernel

$$\begin{aligned} \underset{\alpha\in\mathbb{R}^N}{\text{maximizar}} \quad & \sum_{n=1}^{N}\alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y^n y^m K(x^n,x^m) \equiv \mathbf{1}\alpha - \tfrac{1}{2}\alpha^T\,\mathbf{Q}\alpha \\ \text{sujeito a} \quad & \sum_{n=1}^{N}\alpha_n y^n \equiv Y\alpha = 0 \\ & \alpha_n \geq 0,\ \ n = 1,\ldots,N \end{aligned}$$

$(P_{dual[kernel]})$

onde $\mathbf{Q}_{N\times N} = (Y^T Y).*K(X^T, X)$

Obtida a solução ótima $\alpha^* = (\alpha_1^*, \ldots, \alpha_N^*)^T$ de $(\mathrm{P}_{dual[kernel]})$, substitui-se o produto interno $x^{nT}x^m$ em (5) por $K(x^n, x^m)$:

$$w_0^* = \frac{\sum\limits_{\alpha_n^*>0} \left(y^n - \sum\limits_{\alpha_m^*>0} \alpha_m^* y^m \ \mathbf{K}(x^m, x^n) \right)}{n_{sv}}$$

ou seja, usando a notação vetorial:

$$w_0^* = mean\left(Y_{sv} - (\alpha_{sv}. * Y_{sv}) \ \mathbf{K}(X_{sv}^T, X_{sv}) \right) \tag{9}$$

- X_{sv} matriz $d \times n_{sv}$ dos vetores de suporte
- Y_{sv} vetor $1 \times n_{sv}$ das labels associadas aos vetores de suporte
- α_{sv} vetor $1 \times n_{sv}$ dos alfas associados aos vetores de suporte
- *mean* - dá a média do vetor

Por último, também se substitui no classificador em (7), o produto interno $x^{n\,T} z$ por $K(x^n, z)$:

$$y_p = \text{sinal}\left(\sum_{\alpha_n^*>0} \alpha_n^* y^n K(x^n, z) + w_0^*\right)$$

ou seja, usando a notação vetorial:

$$y_p = \text{sinal}\left(w_0^* + (Y_{sv} .* \alpha_{sv}) K\left(X_{sv}^T, z\right)\right) \rightarrow \begin{cases} +1, & z \in \{+1\} \\ -1, & z \in \{-1\} \end{cases} \quad (10)$$

Observação:

- O vetor normal w^* nunca precisa de ser explicitamente calculado.
- Como tanto o treino como a classificação são realizados num espaço de maior dimensão, em que as classes são mais provavelmente linearmente separáveis, a classificação pode ser mais eficaz.
- Em geral, o método de kernel pode ser aplicado a qualquer algoritmo desde que os pontos $x \in \mathbb{R}^d$ apareçam sempre na forma de um produto interno.

Algoritmo 2: Algoritmo SVM com Kernel (dual)

1. Dar: $D = (x^n, y^n)_{n=1}^N$, $x^n \in \mathbb{R}^d$, $y^n \in \{-1, 1\}$, $z \in \mathbb{R}^d$
2. Fazer: $\mathrm{X}_{\mathrm{d}\times\mathrm{N}} = \begin{pmatrix} x_1^1 & x_1^2 & \dots & x_1^N \\ x_2^1 & x_2^2 & \dots & x_2^N \\ \vdots & \vdots & \dots & \vdots \\ x_d^1 & x_d^2 & \dots & x_d^N \end{pmatrix}$; $\mathrm{Y}_{1\times\mathrm{N}} = (y^1, y^2, \dots, y^N)$
3. Formar matriz $\mathrm{Q}_{N\times N} = (\mathrm{Y}^{\mathrm{T}}\mathrm{Y}).*\mathrm{K}(\mathrm{X}^{\mathrm{T}}, \mathrm{X})$
4. Aplicar um método de otimização de minimização para PQ para resolver o problema dual:

$$\begin{aligned} \text{- minimizar}_{\alpha\in\mathbb{R}^{\mathrm{N}}} \quad & +\frac{1}{2}\alpha^T \mathrm{Q}\alpha - \mathbf{1}\alpha \\ \text{sujeito a} \quad & \mathrm{Y}\alpha = 0 \\ & \alpha_n \geq 0, \ \ n = 1, \dots, N \end{aligned}$$

   $\Rightarrow$ obtém-se a solução ótima $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)$
5. Identificar os vetores de suporte: $\alpha_n^* > 0 \Rightarrow \mathrm{x}^n$ é vetor de suporte
6. Formar: X_{sv}, Y_{sv}, α_{sv}
7. Calcular w_0^* usando (9), e definir o classificador usando (10)

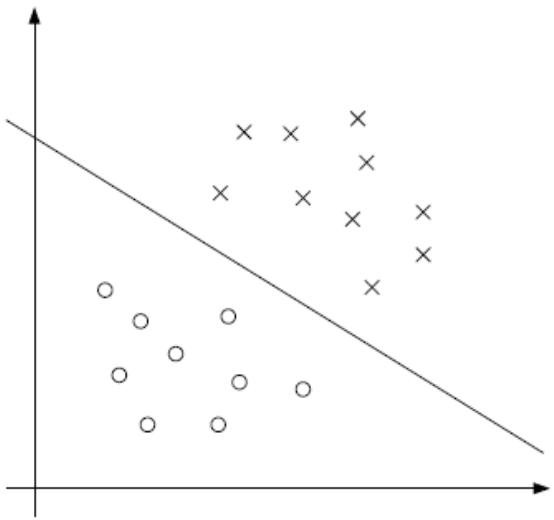
Soft margins SVM: C-SVM

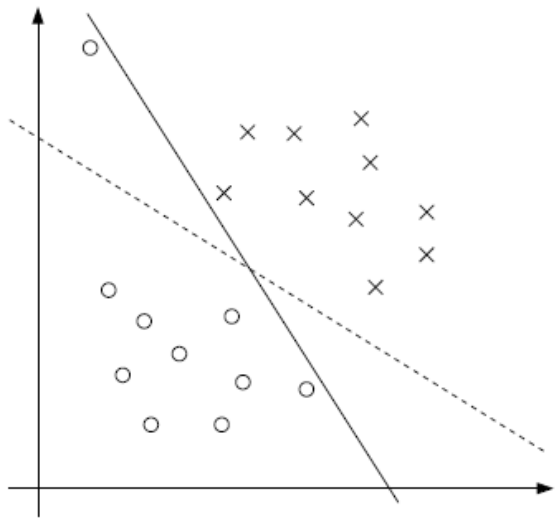
▷ O algoritmo SVM conforme apresentado até agora, assume que os dados são linearmente separáveis. Embora o mapeamento dos dados para um novo espaço dos atributos de maior dimensão via ϕ geralmente aumente a probabilidade de que os dados sejam linearmente separáveis, não podemos garantir que o seja.

▷ Além disso, caso existam *outliers* nos dados, estes podem ter uma influência indevida na orientação e posição do hiperplano.

Exemplo

A figura abaixo à esquerda mostra um classificador de margem ótima, e quando é adicionado um único *outlier* na região superior-esquerda (figura à direita), faz com que o hiperplano faça uma mudança dramática na sua orientação e posição, e o classificador resultante tem um margem muito mais pequena.





Problema de otimização (primal)

Para fazer com que algoritmo SVM trabalhe para dados não-linearmente separáveis bem como ser menos sensível a *outliers*, a condição para o hiperplano ótimo em (1) pode ser relaxada, incluindo um termo de erro extra ($\xi_n \geq 0$):

$$y^n(w^T x^n + w_0) \geq 1 - \xi_n, \quad \text{para} \quad n = 1 \ldots, N$$

Para melhor resultado de classificação, o erro ξ_n $(n = 1, \ldots, N)$ precisa ser minimizado bem como $\|w\|$. Assim, o problema de otimização (primal) em (P_{primal}) é reformulado na forma:

Problema primal (*soft margins*)

$$\begin{aligned} \underset{w \in \mathbb{R}^d,\, w_0 \in \mathbb{R},\, \xi \in \mathbb{R}^N}{\text{minimizar}} \quad & \tfrac{1}{2} w^T w + C \sum_{n=1}^{N} \xi_n \\ \text{sujeito a} \quad & y^n(w^T x^n + w_0) \geq 1 - \xi_n, \quad n = 1, \ldots, N \\ & \xi_n \geq 0 \quad n = 1, \ldots, N \end{aligned} \qquad (P_{primal[soft]})$$

Parâmetro C

O parâmetro $C > 0$ controla o trade-off entre a distribuição global dos pontos das duas classes e os pontos locais próximos à fronteira de cada classe.

- Um valor de C grande, dá ênfase à minimização do termo de erro na função objetivo, de modo que a fronteira de decisão H que maximiza a margem é determinada por um número pequeno de vetores de suporte locais na região entre as duas classes, incluindo possivelmente alguns outliers. O resultado correspondente é semelhante ao de um SVM de margem rígida, considerado anteriormente, e está mais predisposto ao problema de overfitting.
- Um valor de C pequeno, desvaloriza o termo de erro, permitindo erros maiores e que mais pontos se tornam vetores de suporte com base nos quais H é determinado. A fronteira de decisão resultante reflete melhor a distribuição global dos pontos das duas classes.

Problema dual

Como anteriormente, a função Lagrangiana associada ao problema ($P_{primal[soft]}$) é dada por:

$$L(w, w_0, \xi, \alpha, \mu) = \frac{1}{2} w^T w + C \sum_{n=1}^{N} \xi_n - \sum_{n=1}^{N} \alpha_n \left[y^n (w^T x^n + w_0) - 1 + \xi_n \right] - \sum_{n=1}^{N} \mu_n \xi_n$$

onde α_n e μ_n $(n = 1, \ldots, N)$, são os multiplicadores de Lagrange associados às restrições de desigualdade ($\geq$).

Dado que ($P_{primal[soft]}$) é convexo, o problema dual tem a seguinte forma:

$$\underset{w \in \mathbb{R}^d, w_0 \in \mathbb{R}, \xi, \alpha, \mu \in \mathbb{R}^N}{\text{maximizar}} \quad L(w, w_0, \xi, \alpha, \mu)$$

$$\text{sujeito a} \quad \nabla_{w, w_0, \xi} L(w, w_0, \xi, \alpha, \mu) = 0$$
$$\alpha_n \geq 0$$
$$\mu_n \geq 0$$

Pela condição KKT de 1ª ordem: $\nabla_{w,w_0,\xi} L(w, w_0, \xi, \alpha, \mu) = 0$ tem-se

- $\nabla_w L(w, w_0, \xi, \alpha, \mu) = 0 \Leftrightarrow w = \sum_{n=1}^{N} \alpha_n y^n x^n$
- $\nabla_{w_0} L(w, w_0, \xi, \alpha, \mu) = 0 \Leftrightarrow \sum_{n=1}^{N} \alpha_n y^n = 0$
- $\nabla_{\xi_n} L(w, w_0, \xi, \alpha, \mu) = 0 \Leftrightarrow C - \alpha_n - \mu_n = 0, \ (n = 1, \ldots, N)$

Substituindo w na função Lagrangeana pela expressão dada na 1ª equação, e simplificando obtém-se:

$$L(w, w_0, \xi, \alpha, \mu) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y^n y^m \mathrm{x}^{n\,T} \mathrm{x}^m - w_0 \sum_{n=1}^{N} \alpha_n y^n + \sum_{n=1}^{N} (C - \alpha_n - \mu_n) \xi_n$$

Mas pelas 2ª e 3ª equações, os dois últimos termos tem de ser zero, obtendo-se a mesma função objetivo dual, como no caso anterior:

$$L(w, w_0, \xi, \alpha, \mu) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y^n y^m \mathrm{x}^{n\,T} \mathrm{x}^m$$

Como $\mu_n \geq 0$ e $C - \alpha_n - \mu_n = 0$, então $C - \alpha_n \geq 0 \Leftrightarrow \alpha_n \leq C$, e a restrição $0 \leq \alpha_n$ é substituída por $0 \leq \alpha_n \leq C$.

Portanto, o problema dual para o problema $(\mathrm{P}_{primal[soft]})$ é:

Problema dual (*soft margins*)

$$\begin{array}{ll} \underset{\alpha \in \mathbb{R}^N}{\text{maximizar}} & \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y^n y^m \mathrm{x}^{n\,T} \mathrm{x}^m \equiv \mathbf{1}\alpha - \frac{1}{2} \alpha^T \mathrm{Q} \alpha \\ \text{sujeito a} & \sum_{n=1}^{N} \alpha_n y^n = \mathrm{Y}\alpha = 0 \\ & 0 \leq \alpha_n \leq C, \quad n = 1, \ldots, N \end{array}$$

$(\mathrm{P}_{dual[soft]})$

▷ $(\mathrm{P}_{dual[soft]})$ também é um problema de PQ, o qual pode ser resolvido por um dos métodos de otimização acima mencionados.

▷ Tendo-se a solução ótima α^*: w_0^* é obtido por (6) e o classificador é dado por (8). Este algoritmo é designado por C-SVM com Kernel.

Algoritmo 3: Algoritmo C-SVM (dual)

1. Dar: $D = (x^n, y^n)_{n=1}^N$, $x^n \in \mathbb{R}^d$, $y^n \in \{-1, 1\}$, $C \in \mathbb{R}^+$
2. Fazer: $\mathrm{X}_{\mathrm{d}\times\mathrm{N}} = \begin{pmatrix} x_1^1 & x_1^2 & \dots & x_1^N \\ x_2^1 & x_2^2 & \dots & x_2^N \\ \vdots & \vdots & \dots & \vdots \\ x_d^1 & x_d^2 & \dots & x_d^N \end{pmatrix}$; $\mathrm{Y}_{1\times\mathrm{N}} = \left(y^1, y^2, \dots, y^N\right)$
3. Formar matriz $\mathrm{Q}_{N\times N} = (\mathrm{Y}^{\mathrm{T}}\mathrm{Y}).*(\mathrm{X}^{\mathrm{T}}\mathrm{X})$
4. Aplicar um método de otimização de minimização para PQ para resolver o problema dual:

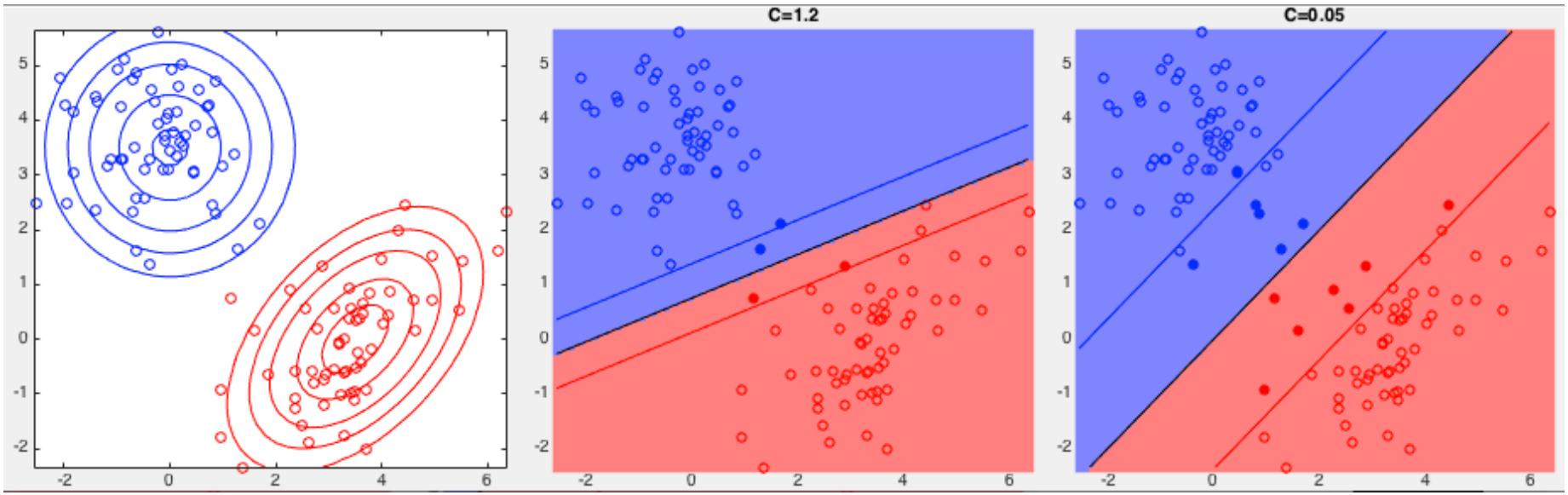
$$\begin{aligned} \text{- minimizar}_{\alpha\in\mathbb{R}^{\mathrm{N}}} \quad & +\frac{1}{2}\alpha^T \mathrm{Q}\alpha - \mathbf{1}\alpha \\ \text{sujeito a} \quad & \mathrm{Y}\alpha = 0 \\ & 0 \le \alpha_n \le C, \ n = 1, \dots, N \end{aligned}$$

   $\Rightarrow$ obtém-se a solução ótima $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)^T$
5. Identificar os vetores de suporte: $\alpha_n^* > 0 \Rightarrow \mathrm{x}^n$ é vetor de suporte
6. Formar: X_{sv}, Y_{sv}, α_{sv}
7. Calcular w_0^* usando (6) e definir o classificador usando (8)

Exemplo:

O algoritmo C-SVM é testado como o mesmo conjunto de treino anterior, ver figura abaixo à esquerda. Na figura, mostra-se dois resultados com os valores $C = 1.2$ e $C = 0.05$.

- quando $C = 1.2$ é grande, o número de vetores de suporte (os quatro pontos sólidos) é pequeno devido ao menor erro ξ_n permitido. O hiperlano de decisão determinado por estes vetores de suporte, independente do resto do conjunto de dados (círculos), é principalmente ditado pelos pontos próximos ao hiperplano, não necessariamente um bom reflexo da distribuição global dos pontos.
- quando $C = 0.05$ é pequeno, o número de vetores suporte (os 13 pontos sólidos) é maior, devido ao maior erro ξ_n permitido, e o hiperplano de decisão resultante determinado por estes suporte vetores reflete melhor a distribuição global dos pontos, e separa melhor as duas classes.



C-SVM com função Kernel

O método do kernel pode ser aplicado ao algoritmo C-SVM, pois todos os pontos de dados exibidos no algoritmo estão na forma de um produto interno. Assim, durante o processo de treino, substituímos o produto interno $x^{nT}x^m$ em $(P_{dual[soft]})$ pela função kernel $K(x^n, x^m)$:

Problema dual (*soft margin*) com Kernel

$$\begin{array}{ll} \underset{\alpha\in\mathbb{R}^N}{\text{maximizar}} & \sum_{n=1}^N \alpha_n - \frac{1}{2}\sum_{n=1}^N\sum_{m=1}^N \alpha_n\alpha_m y^n y^m K(x^n,x^m) \equiv \mathbf{1}\alpha - \frac{1}{2}\alpha^T Q\alpha \\ \text{sujeito a} & \sum_{n=1}^N \alpha_n y^n = Y\alpha = 0 \\ & 0 \le \alpha_n \le C \end{array}$$

$(P_{dual[soft][kernel]})$

onde $Q_{N\times N} = (Y^TY).*K(X^T,X)$

▷ $(P_{dual[soft][kernel]})$ também é um problema de PQ.

▷ Tendo-se a solução ótima α^*: w_0^* é obtido por (9) e o classificador é dado por (10). Este algoritmo é designado por C-SVM com kernel.

Algoritmo 4: Algoritmo C-SVM com Kernel (dual)

1. Dar: $D = (x^n, y^n)_{n=1}^N$, $x^n \in \mathbb{R}^d$, $y^n \in \{-1, 1\}$, $C \in \mathbb{R}^+$
2. Fazer: $\mathrm{X}_{\mathrm{I}\times\mathrm{N}} = \begin{pmatrix} x_1^1 & x_1^2 & \dots & x_1^N \\ x_2^1 & x_2^2 & \dots & x_2^N \\ \vdots & \vdots & \dots & \vdots \\ x_d^1 & x_d^2 & \dots & x_d^N \end{pmatrix}$; $\mathrm{Y}_{1\times\mathrm{N}} = (y^1, y^2, \dots, y^N)$
3. Formar matriz $\mathrm{Q}_{N\times N} = (\mathrm{Y}^\mathrm{T}\mathrm{Y}).*\mathrm{K}(\mathrm{X}^\mathrm{T}, \mathrm{X})$
4. Aplicar um método de otimização de minimização para PQ para resolver o problema dual:

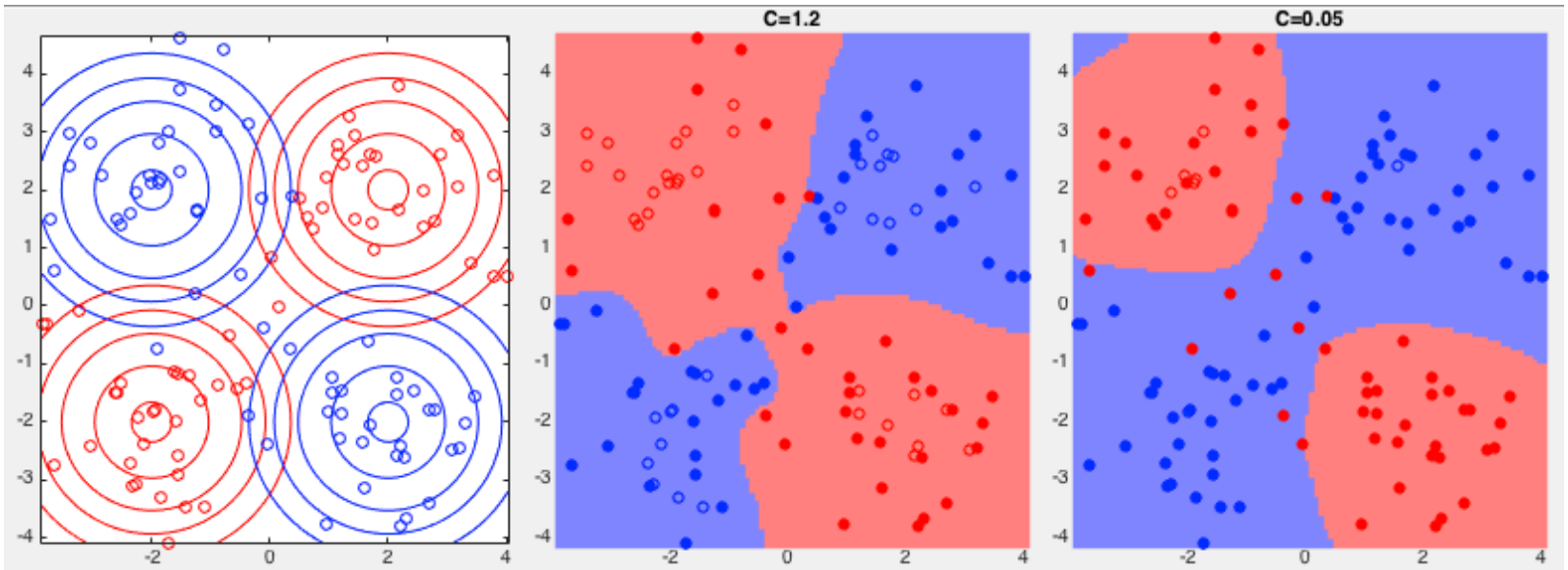
$$\begin{aligned} \text{- minimizar}_{\alpha\in\mathbb{R}^\mathrm{N}} \quad & +\frac{1}{2}\alpha^T \mathrm{Q}\alpha - \mathbf{1}\alpha \\ \text{sujeito a} \quad & \mathrm{Y}\alpha = 0 \\ & 0 \le \alpha_n \le C, \ n = 1, \dots, N \end{aligned}$$

   $\Rightarrow$ obtém-se a solução ótima $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)$
5. Identificar os vetores de suporte: $\alpha_n^* > 0 \Rightarrow \mathrm{x}^n$ é vetor de suporte
6. Formar: X_{sv}, Y_{sv}, α_{sv}
7. Calcular w_0^* usando (9) e definir o classificador usando (10)

Exemplo:

O algoritmo C-SVM é testado com o conjunto de dados XOR, ver figura abaixo à esquerda. Como as duas classes não são linearmente separáveis, usou-se o Kernel RBF para mapear os dados do espaço $\mathbb{R}^2$ para um espaço dimensão infinita. Na figura, mostra-se dois resultados com os valores $C = 1.2$ e $C = 0.05$.

- quando $C = 1.2$ é grande, o número de vetores de suporte (pontos sólidos) é pequeno devido ao menor erro ξ_n permitido. O hiperplano determinado por estes vetores, permite classificar todos os pontos corretamente, mas é mais predisposto ao problema de overfitting, se alguns dos vetores de suporte são outliers.
- quando $C = 0.05$ é pequeno, o número de vetores suporte (pontos sólidos) é maior, devido ao maior erro ξ_n permitido, e o hiperplano de decisão resultante, determinado por estes vetores de suporte, reflete melhor a distribuição global dos pontos. No entanto, como pontos locais próximos ao hiperplano não são muito penalizados, ocorre falha na classificação, cerca de 9 pontos vermelhos são classificados na região azul.



(nota:as cores nas duas figuras mais à direita estão trocadas)

Exercício 1: Implemente o Algoritmo SVM dual (**Algoritmo 1**) e aplique-o aos data sets ex1data1.csv e ex1data2.csv. Dividir o data set em duas partes: 80% para treino Dt e 20% para validação Dv.

- Seleccionar para Dt 80% dos primeiros elementos do data set.
- Para cada um dos data set, indique: os vetores de suporte, w^*, w_0^*, e erro de validação (out-sample error);
- Visualize num gráfico: data set de teino, vetores de suporte e Hiperplano (fronteira de decisão).

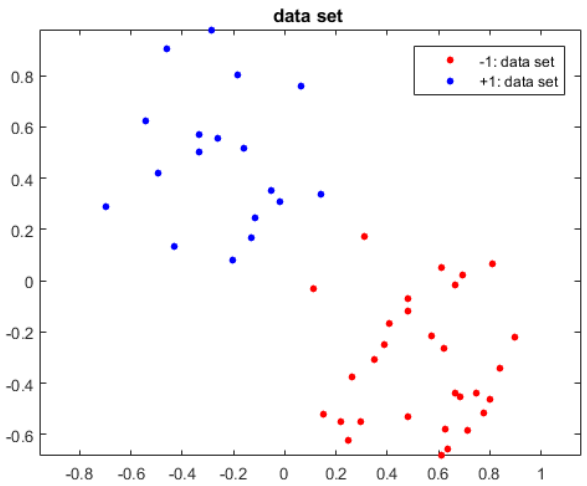
Exercício 2: Faça o exercício 1 mas considere agora que selecciona aleatoriamente para Dt 80% dos elementos do data set.

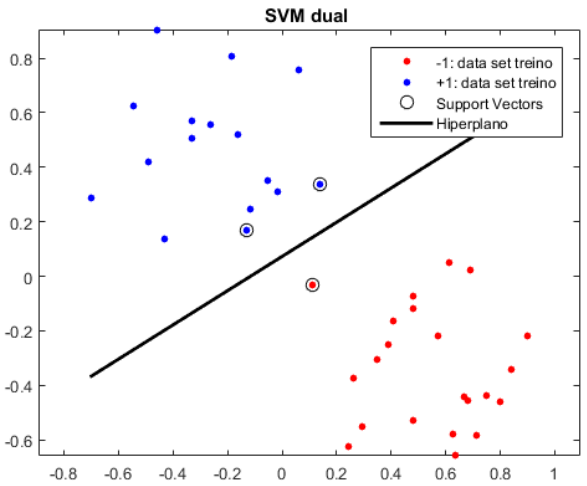
Exercício 3: Implemente o Algoritmo SVM com Kernel (dual) (**Algoritmo 2**) e aplique-o aos data sets ex2data1.csv e ex2data2.csv. Dividir o data set em duas partes: 80% para treino Dt e 20% para validação Dv.

- Seleccionar aleatoriamente para Dt 80% dos elementos do data set.
- Use para Kernel a função RBF.
- Para cada um dos data set, indique: os vetores de suporte, w_0^*, e erro de validação (out-sample error);
- Visualize num gráfico: data set de teino, vetores de suporte e a fronteira de decisão.

Exercício 4: Faça o exercício 3 mas considere agora que usa para Kernel a função polinomial de grau 2 (d=2).

Exercício 1: Solução com ex1data2.csv





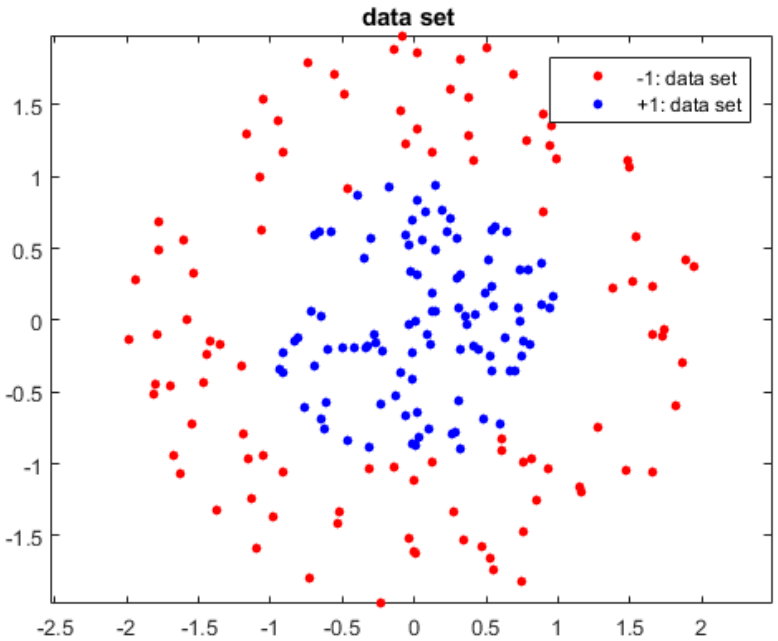
O hiperplano ótimo é determinado por três vetores de suporte e a margem entre eles é máxima. Solução ótima:

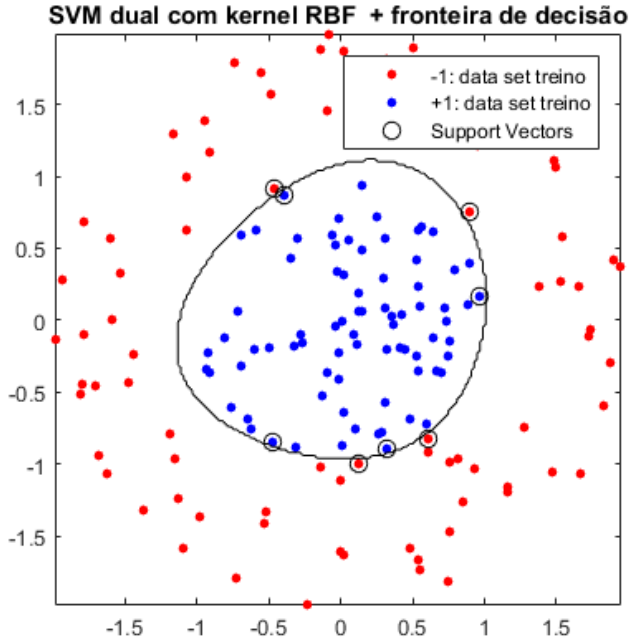
n	α_n^*	x^n	y^n
3	22.60	$(0.1115, -0.0328)^T$	-1
5	15.72	$(-0.1285, 0.1678)^T$	1
31	6.88	$(0.1403, 0.3365)^T$	1

$\mathrm{w}^* = (-3.57, 5.69)^T$, $w_0^* = -0.41$

out-sample error: 0.00

Exercício 3: Solução com ex2data1.csv





A fronteira de decisão ótima é determinado por oito vetores de suporte.
Solução ótima:

Exercício 3: Solução com ex2data1.csv

n	α_n^*	x^n	y^n
4	175.72	[-0.4608, 0.9134]	-1
66	7.22	[0.9006, 0.7536]	-1
88	179.76	[-0.3904, 0.8666]	1
89	531.12	[0.3192, -0.8911]	1
120	5.81	[0.9707, 0.1613]	1
121	209.40	[0.6095, -0.8207]	-1
122	28.49	[-0.4680, -0.8417]	1
150	352.84	[0.1269, -0.9925]	-1

$w_0^* = -11.63$

out-sample error: 0.00

Nota: Como o data set de treino foi seleccionado aleatoriamente, a solução altera sempre que se corre o algoritmo.

Bibliografia

- Mohammed J. Zaki, Wagner Meira, Jr., Data Mining and Machine Learning: Fundamental Concepts and Algorithms, 2nd Edition, Cambridge University Press, March 2020.