Sensores espectroscópicos e modelos de regressão aplicados na análise de solos

Aula 3 – Máquinas de Vetores de Suporte

Me. José Vinícius Ribeiro

Universidade Estadual de Londrina

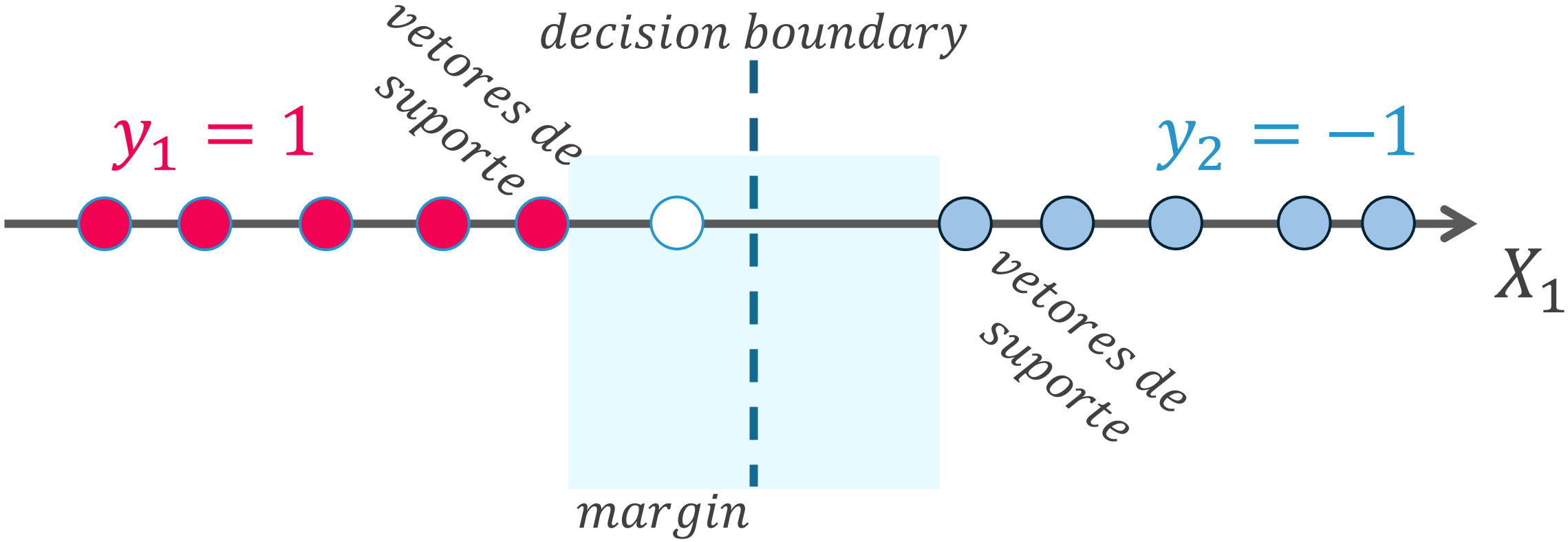
Pós Graduação Física UEL

SUMÁRIO

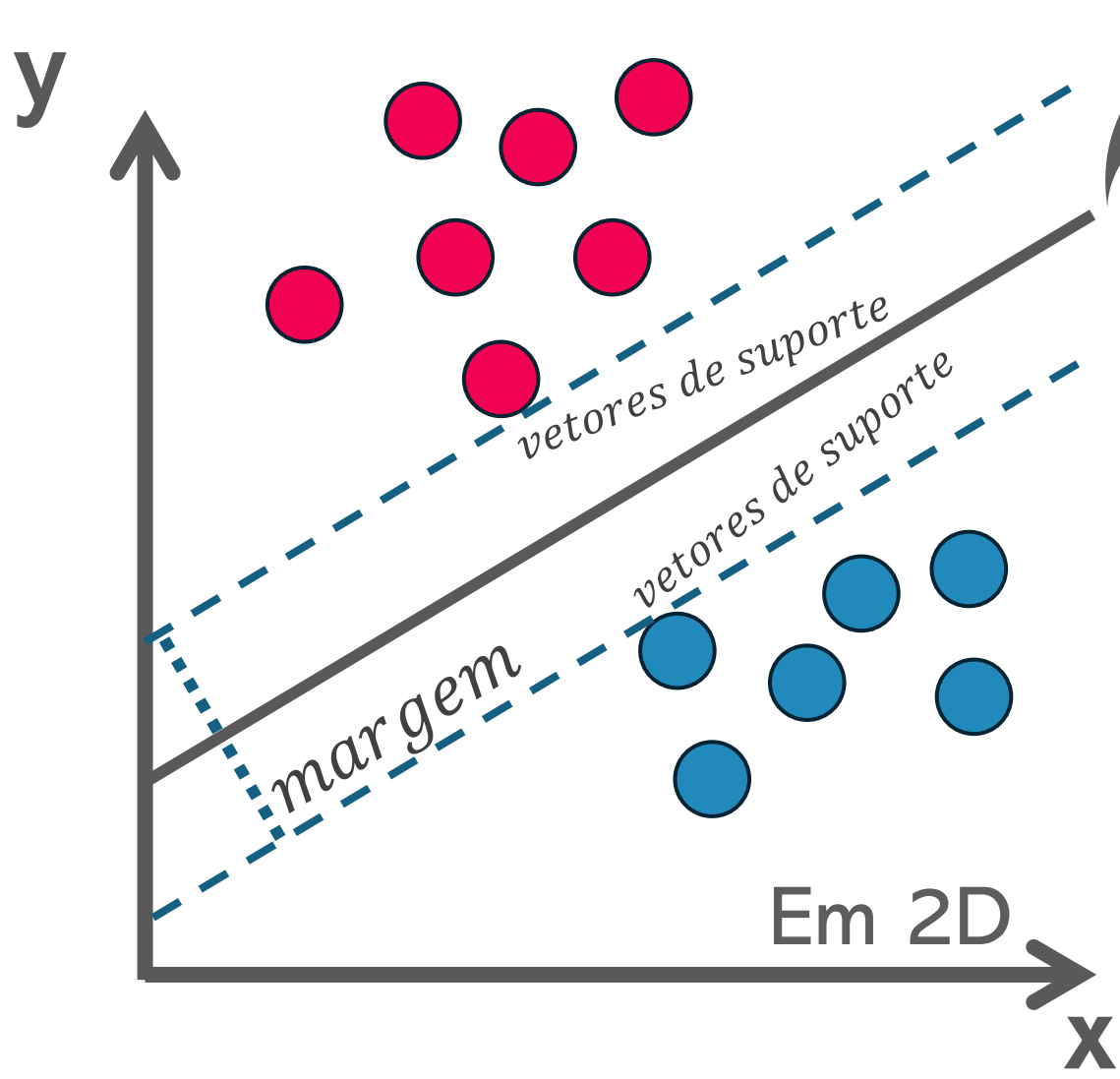
- Ideia principal
- Um pouco de Geometria Analítica
- SVM linear – obtendo o hiperplanno
- Truque de Kernel
- SVR
- Método de permutação (Explicabilidade)
- Prática no python (vscode)

Ideia principal das SVMs

As máquinas de vetores de suporte (SVMs) exploram uma sofisticada ideia de *decision boundary* associada a *margin.*



Ideia principal das SVMs



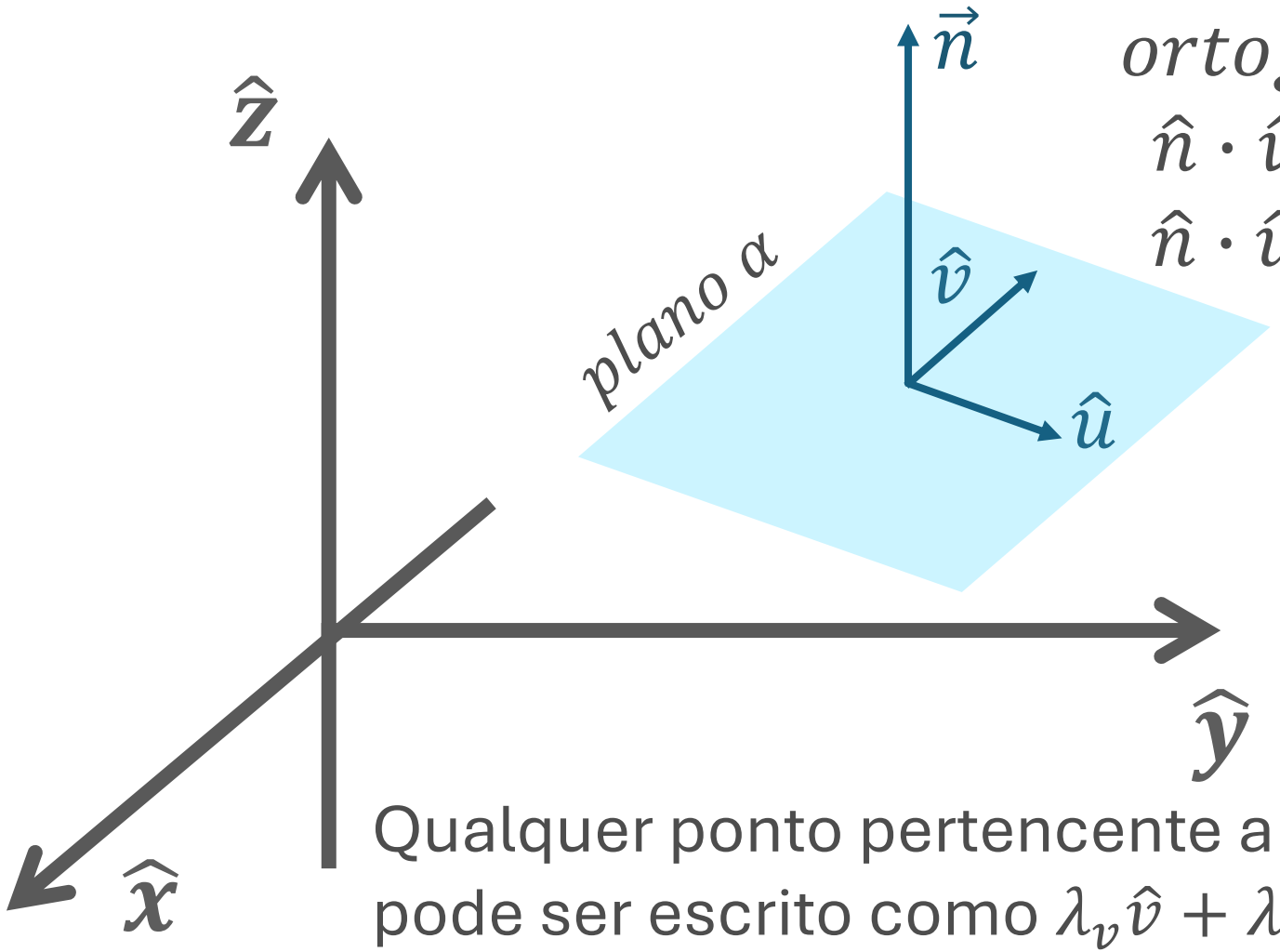
hiperplano

Depende do numero de variáveis

Vai ser construído através do sub-conjunto amostral de vetores de suporte

Um pouco de G. A. - planos

$todo\ vetor\ de\ \alpha\ são\ ortogonais\ ao\ vetor\ \hat{n}$



$ortogonal$

$\hat{n} \cdot \hat{v} = 0$

$\hat{n} \cdot \hat{u} = 0$

$\hat{v}, \hat{u}\ linearmente\ independentes\ definem\ o\ plano$
(vetores diretores)

$definição\ de\ \alpha$

$\hat{\alpha} = v\hat{v} + u\hat{u}$

$u, v \in \mathbb{R}$

Qualquer ponto pertencente a α pode ser escrito como $\lambda_v \hat{v} + \lambda_u \hat{u}$

Um pouco de G. A. - planos

Com essas informações, podemos representar qualquer plano por meio do seu vetor normal

Considerando um vetor $\vec{r}$ do espaço $\mathbb{R}^3$ qualquer, pertencente a um plano também $\mathbb{R}^3$ e dado por:

$$\vec{r} = (x - x_0)\hat{x} + (y - y_0)\hat{y} + (z - z_0)\hat{z}$$

$$\vec{r} = (x - x_0, y - y_0, z - z_0)$$

$$\vec{r} \cdot \hat{n} = 0 = (x - x_0, y - y_0, z - z_0) \cdot (n_x, n_y, n_z) =$$

$$n_x(x - x_0) + n_y(y - y_0) + n_z(z - z_0) = 0$$

Um pouco de G. A. - planos

Equação geral do plano (versão escalar):

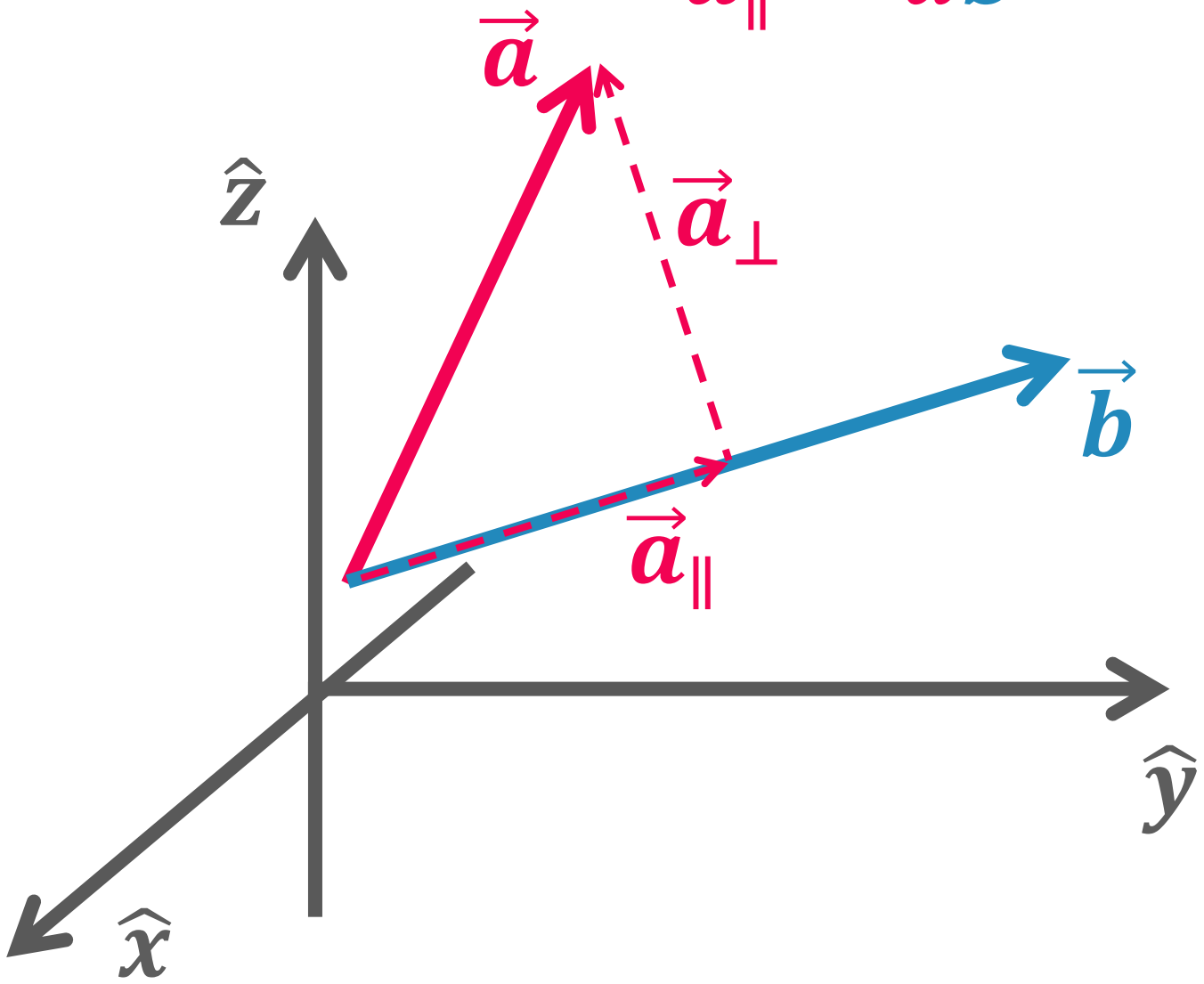
$$xn_x + yn_y + zn_z = cte$$

$$a_1x_1 + a_2x_2 + \cdots + a_mx_m = cte$$

Dessa forma, qualquer hiperplano (independente da dimensão) pode ser obtido através do vetor normal a ele.
O SVM utiliza essa ideia para calcular os hiperplanos ótimos

Um pouco de G. A. – projeção entre vetores

$$\vec{a}_{\parallel} = \alpha\vec{b} \qquad \vec{a} = \vec{a}_{\parallel} + \vec{a}_{\perp} \qquad \vec{a} = \alpha\vec{b} + \vec{a}_{\perp}$$



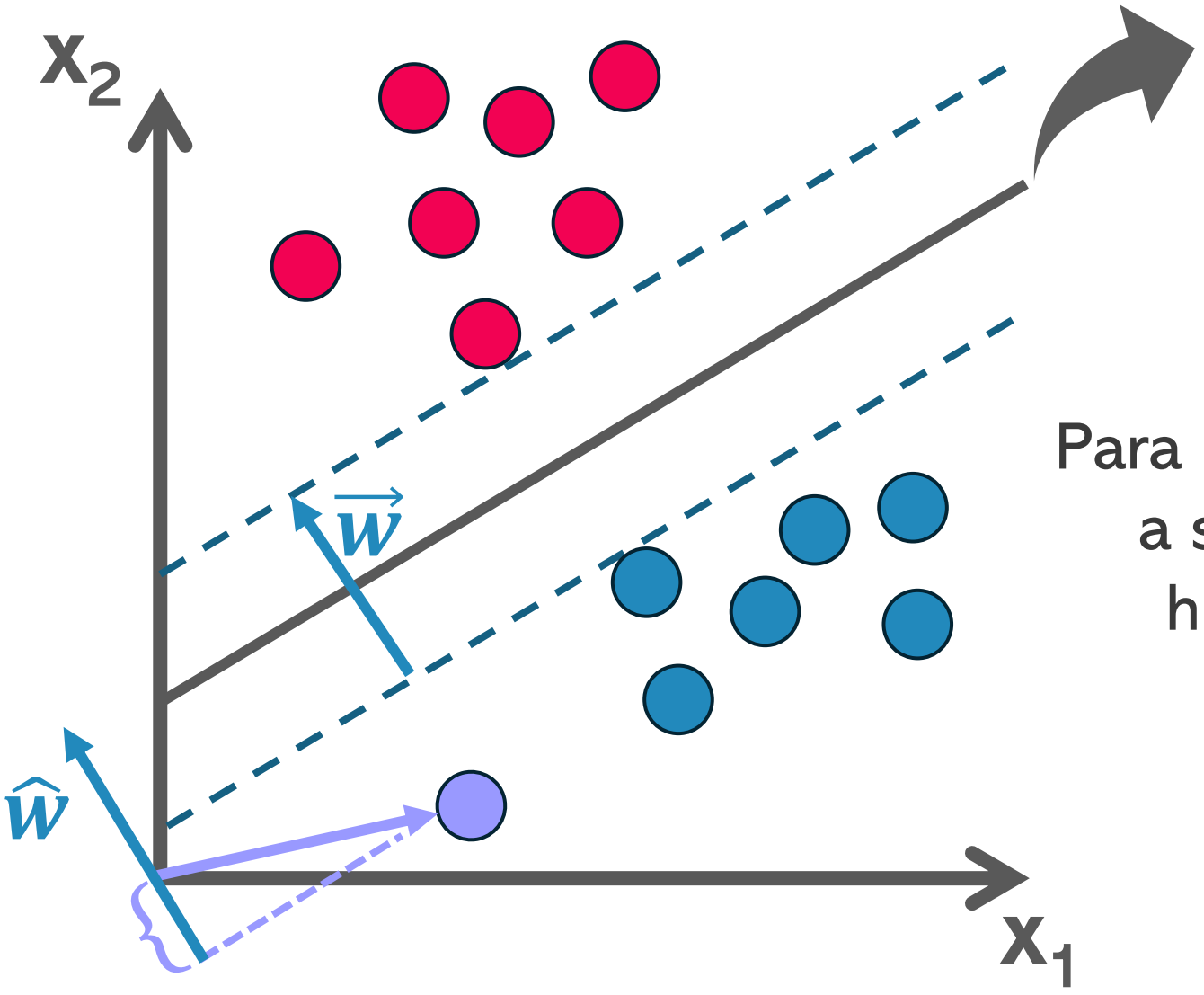
$$\vec{b} \cdot \vec{a} = \vec{b} \cdot \alpha\vec{b} + \vec{b} \cdot \vec{a}_{\perp}$$

$$\vec{b} \cdot \vec{a} = \alpha \parallel \vec{b} \parallel^2$$

$$\alpha\vec{a}_{\parallel} = \frac{\vec{b}\,\vec{b}\vec{a}\,\vec{a}}{\parallel \vec{b}\,\vec{b}^2 \parallel^2}\vec{b}$$

$$\parallel \vec{a}_{\parallel} \parallel = \frac{\vec{b} \cdot \vec{a}}{\parallel \vec{b} \parallel}$$

SVM e G. A.

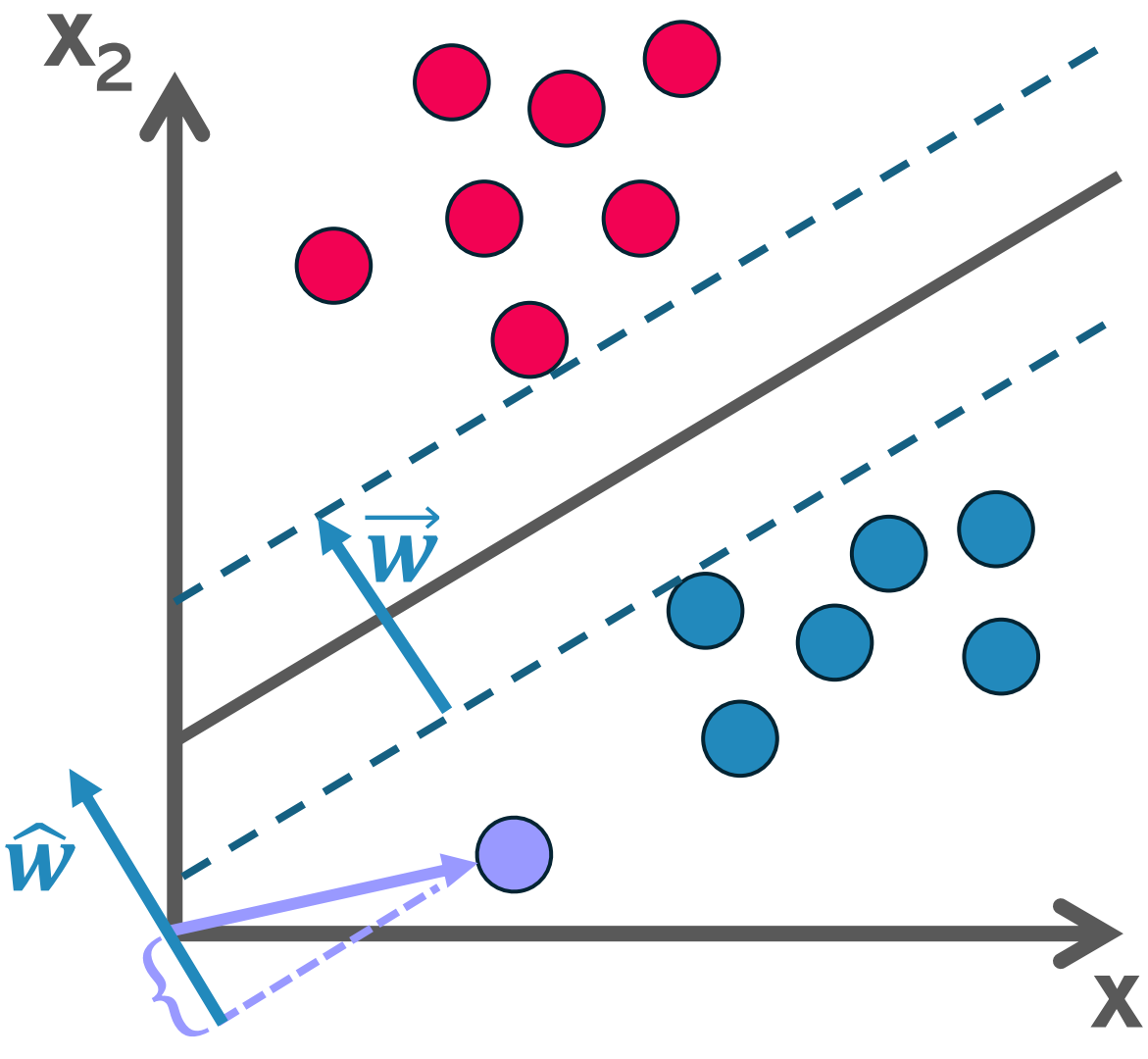


$$w_{x_1} x_1 + w_{x_2} x_2 + b = 0$$

Para classificar novos dados, calculamos a sua projeção no vetor normal ao hiperplano e somamos o viés (*b*)

SVM e G. A.

Relembrando $\| \vec{r}_{\|} \| = \frac{\vec{w} \cdot \vec{r}}{\|\vec{w}\|}$



Se o vetor $\vec{r} = x_{10}\hat{x}_1 + x_{20}\hat{x}_2$ localiza o novo ponto, fazemos:

$$= \hat{\boldsymbol{w}} \cdot \vec{\boldsymbol{r}} + b$$

$$w_{x_1} x_{10} + w_{x_2} x_{20} + b =$$

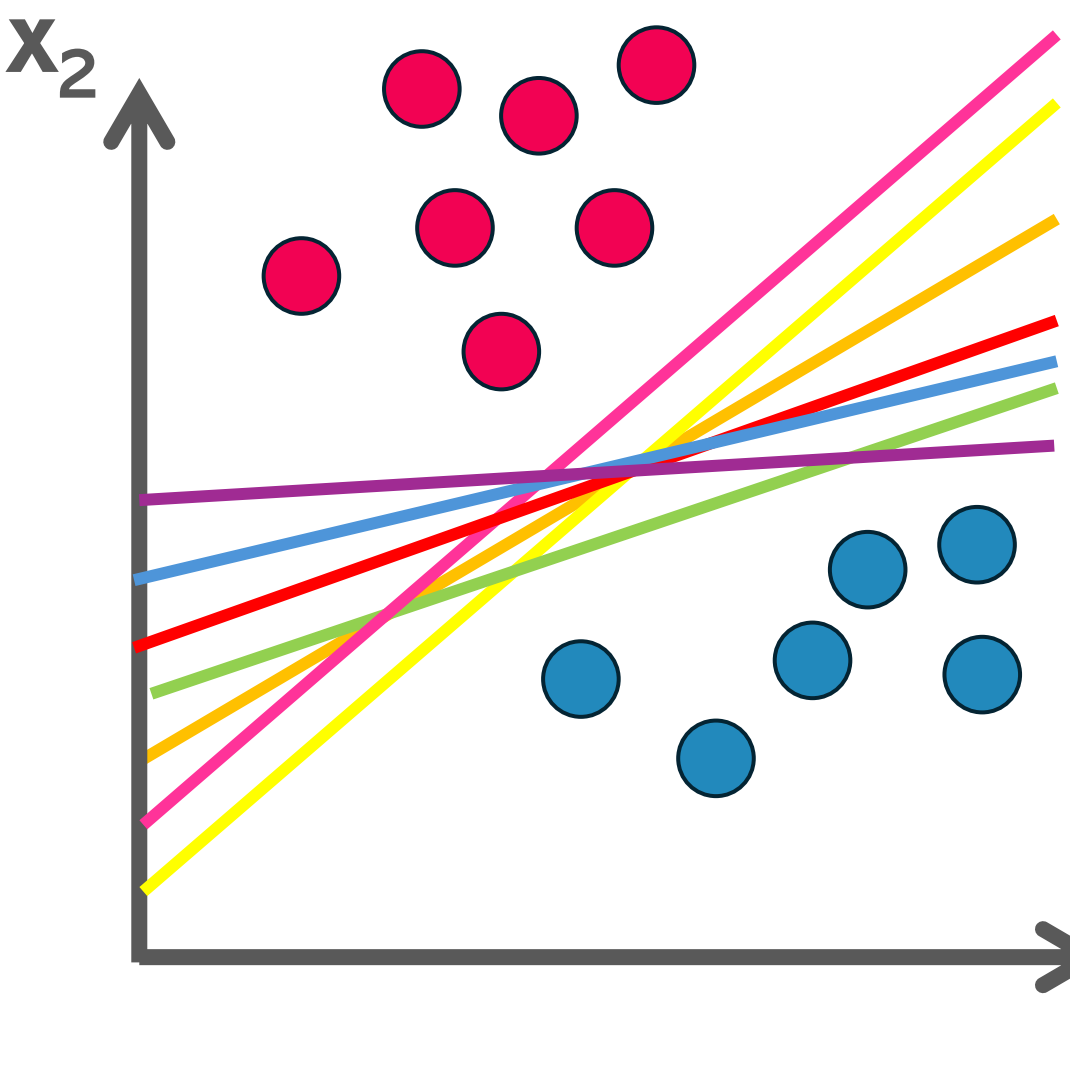
> 0 Vermelha

< 0 Azul

$=0$ Indefinido

Obtendo o hiperplano

SVM linear – obtendo o hiperplano

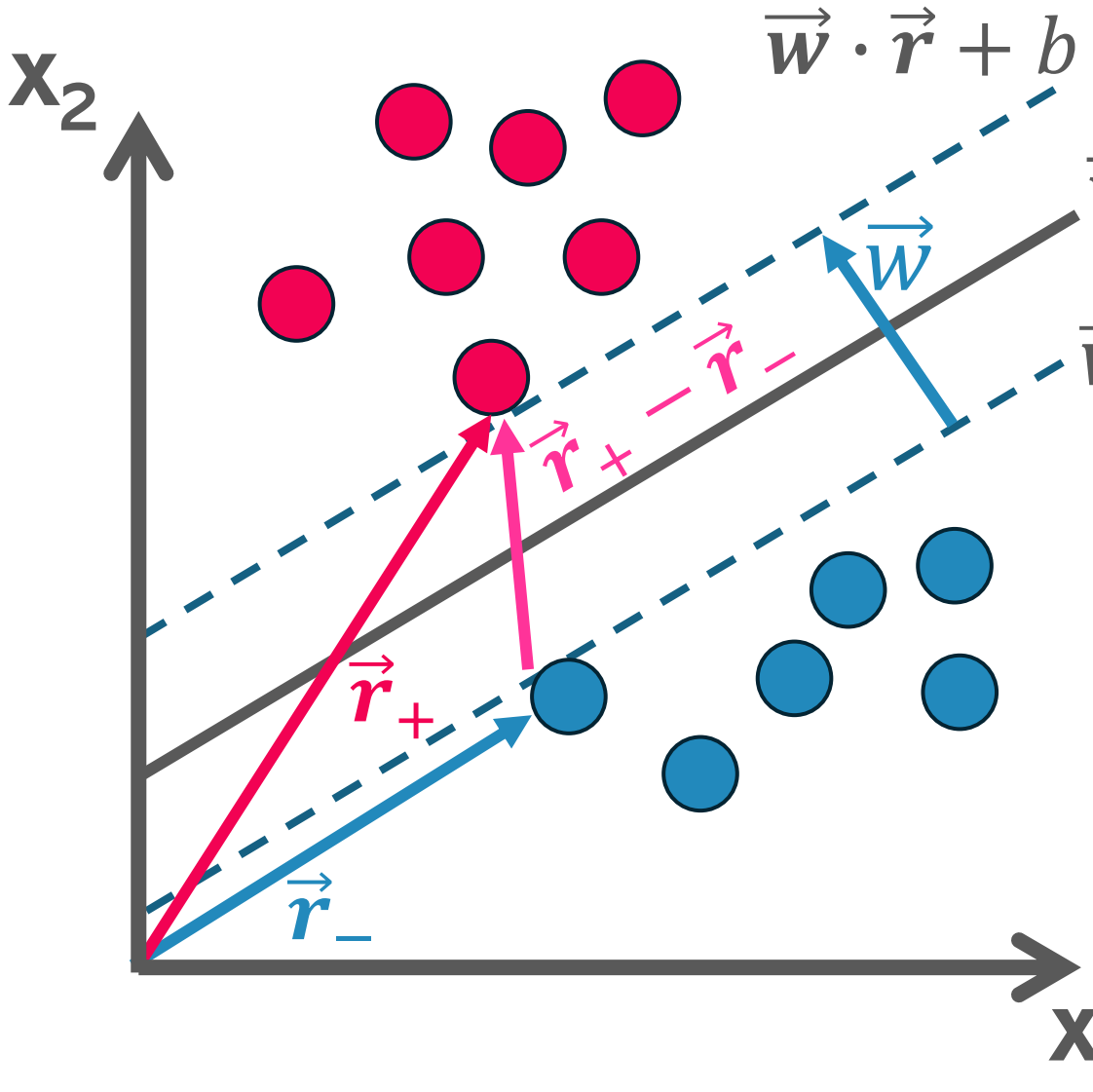


Infinitos planos são possíveis de serem calculados

Qual o mais eficiente?

Aquele que maximiza a margem!

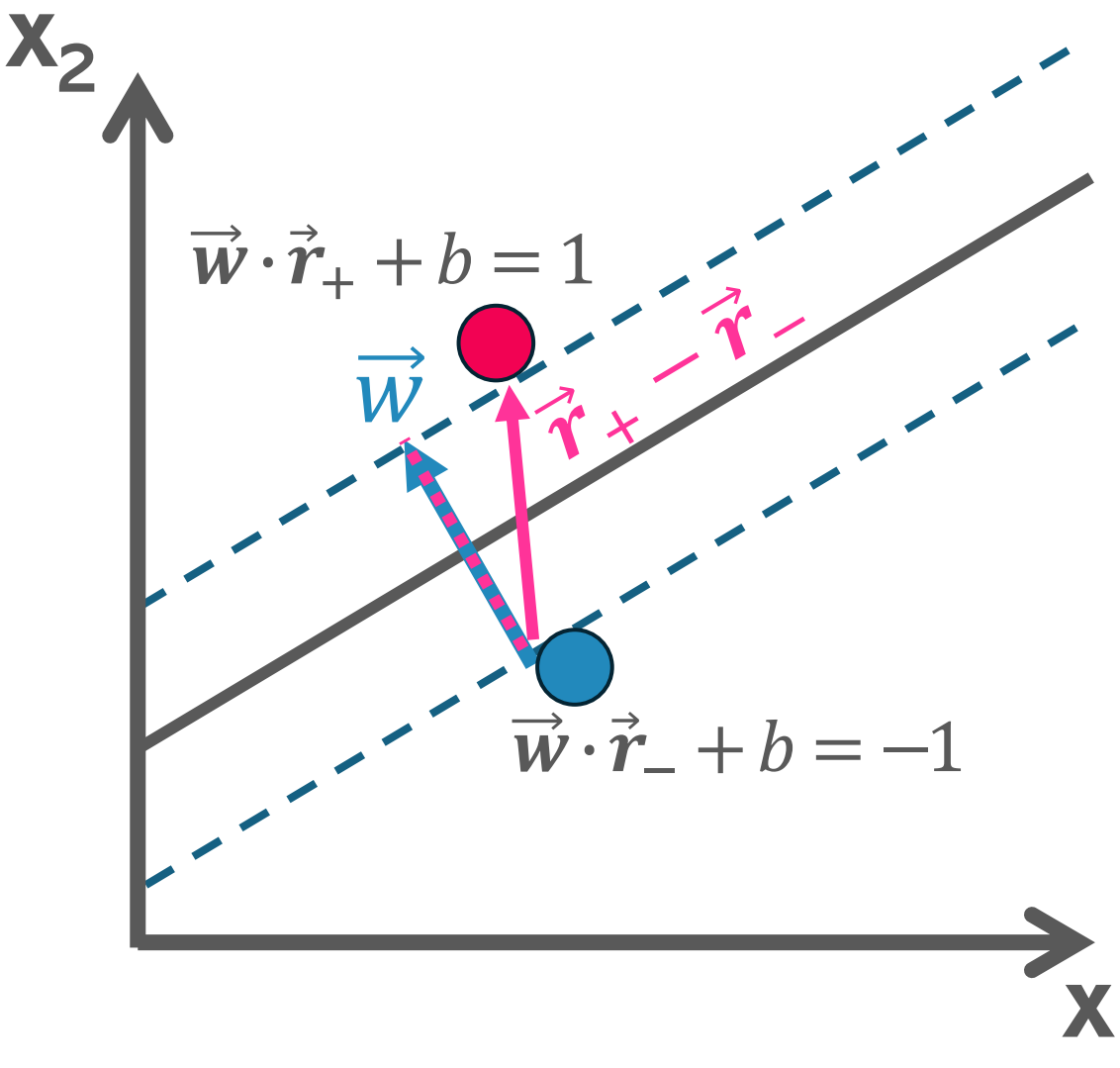
SVM linear – obtendo o hiperplano



$\vec{r} = x_1\hat{x}_1 + x_2\hat{x}_2$

Vamos calcular a margem com o auxílios dos vetores de suporte e buscar maximiza-la

SVM linear – obtendo o hiperplano

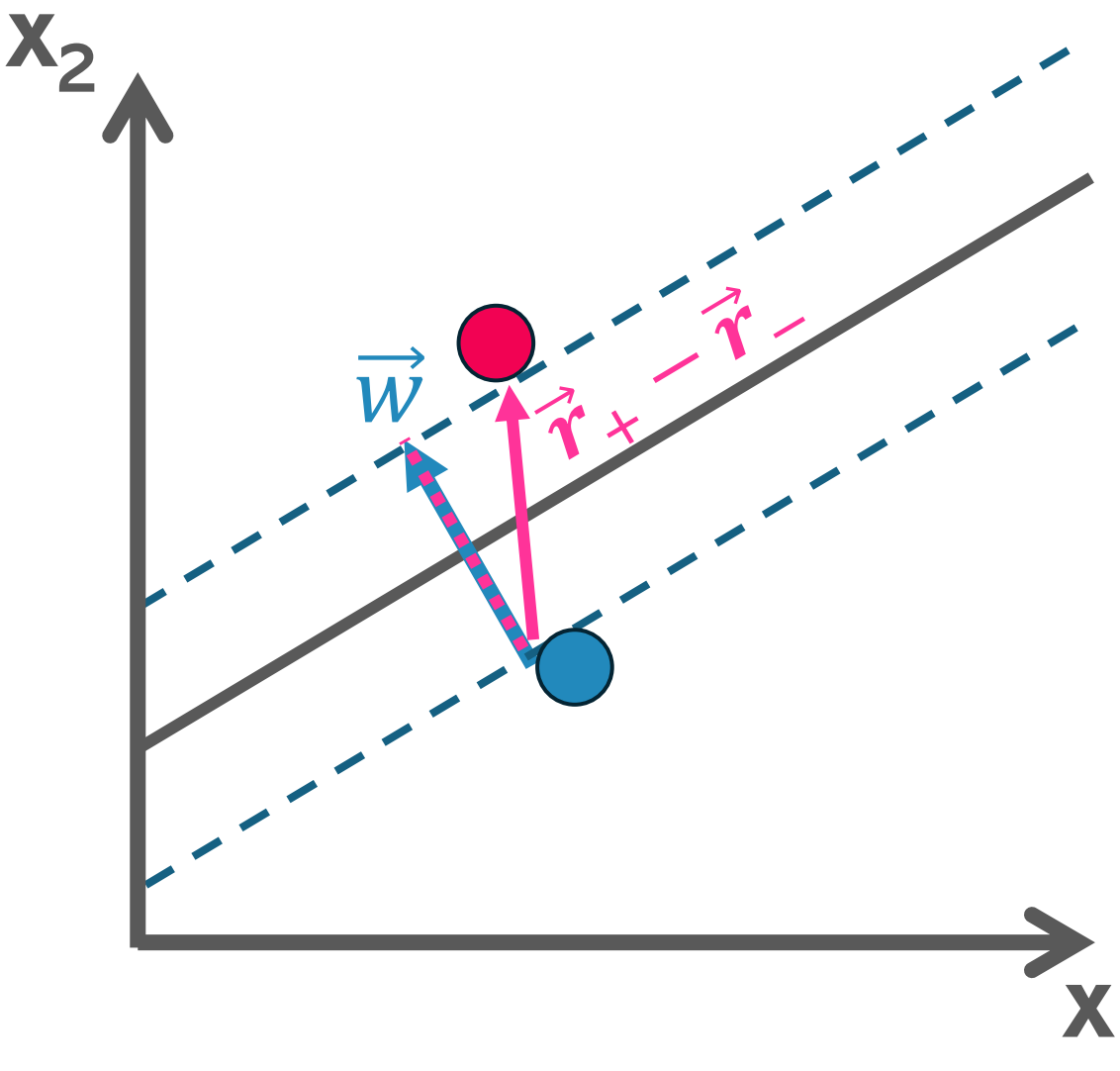


A projeção de $\vec{r}_+ - \vec{r}_-$ em $\vec{w}$ fornece a margem

$$\frac{(\vec{r}_+ - \vec{r}_-) \cdot \vec{w}}{\| \vec{w} \|} = \frac{\vec{r}_+ \cdot \vec{w} - \vec{r}_- \cdot \vec{w}}{\| \vec{w} \|}$$

$$= \frac{1 - b - (-1 - b)}{\| \vec{w} \|} = \frac{2}{\| \vec{w} \|}$$

SVM linear – obtendo o hiperplano



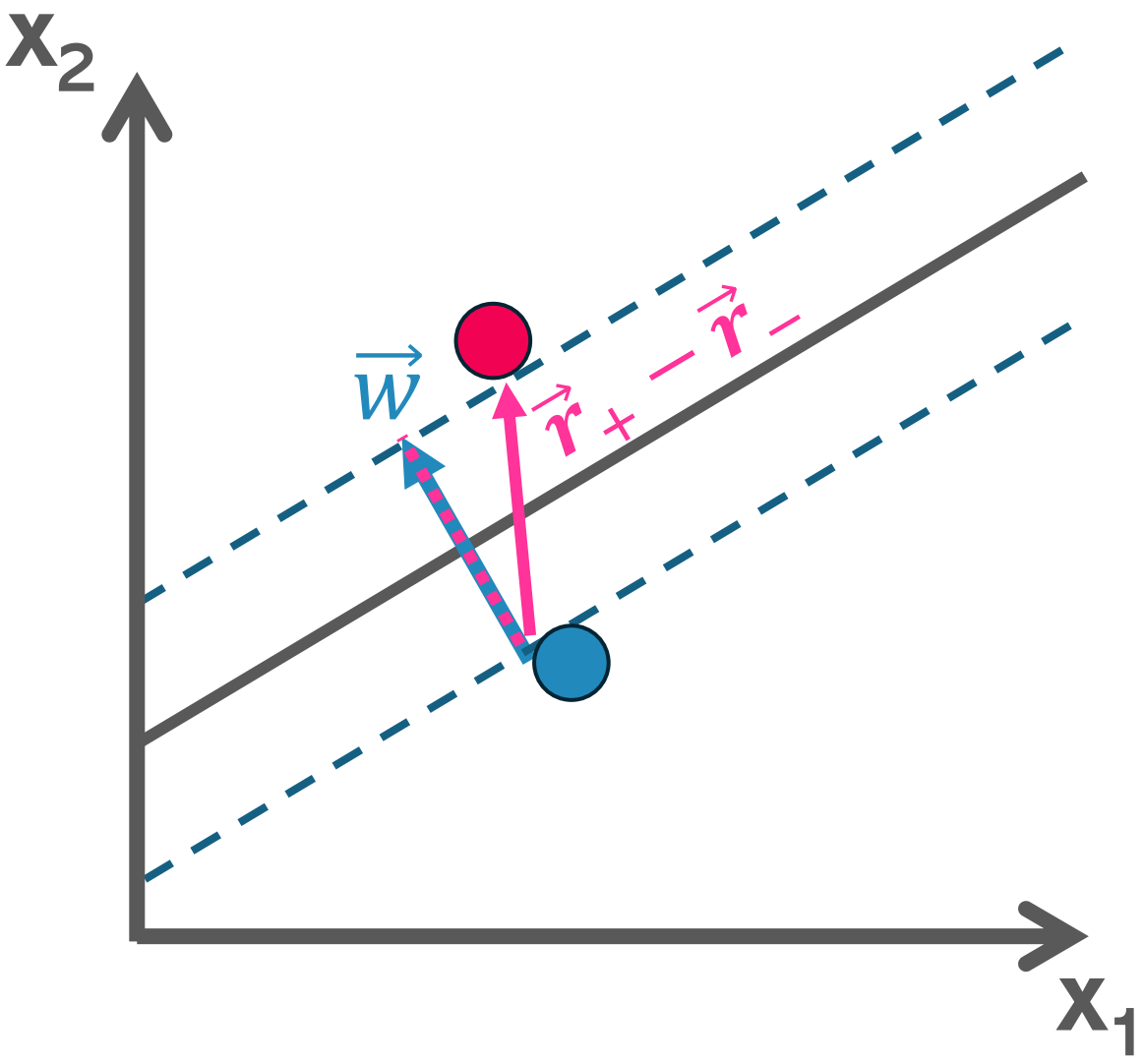
$$Máx_{\|\vec{w}\|}\left(\frac{2}{\|\vec{w}\|}\right)$$

Na prática é mais conveniente

$$Min_{\|\vec{w}\|}\left(\frac{1}{2}\|\vec{w}\vec{w}\|^2\right)$$

SVM linear – obtendo o hiperplano

Equação é válida para mais dimensões $Min_{\|\vec{w}\|,b}(\frac{1}{2} \| \vec{w} \|^2)$



Restrição: $y_i(\vec{w} \cdot \vec{r}_i + b) - 1 \geq 0$

No hiperplano: $y_i(\vec{w} \cdot \vec{r}_i + b) - 1 = 0$

Ou seja, temos um problema de otimização de parâmetros ($\| \vec{w} \|$ e b) com restrições.

A técnica dos multiplicadores de Lagrange transforma problemas com restrições em um sem restrições, englobando multiplicadores α

SVM linear – obtendo o hiperplano

$$Min_{\|\vec{w}\|,b}\left(\frac{1}{2}\| \vec{w} \|^2\right) \qquad \text{Restrição: } y_i(\vec{w}\cdot\vec{r}_i + b) - 1 = 0$$

Multiplicadores de Lagrange (minimizar)

$$L(x,\alpha) = f(x) - \alpha[g(x) - \lambda] \qquad \nabla_x L = 0 \quad g(x) = \lambda$$

Montamos a Lagranngiana e a extremizamos $(\nabla L = 0)$

$$L(\| \vec{w} \|, b, \alpha) = \frac{1}{2}\| \vec{w} \|^2 - \sum_{i=1}^{n} \alpha_i [y_i(\vec{w}\cdot\vec{r}_i + b) - 1]$$

$$\frac{\partial}{\partial b} L(\| \vec{w} \|, b, \alpha) = 0 \qquad \frac{\partial}{\partial \vec{w}} L(\| \vec{w} \|, b, \alpha) = 0$$

SVM linear – obtendo o hiperplano

$$Min_{\|\vec{w}\|,b}\left(\frac{1}{2}\| \vec{w} \|^2\right) \qquad \text{Restrição: } y_i(\vec{w}\cdot\vec{r}_i + b) - 1 = 0$$

$$L(\| \vec{w} \|, b, \alpha) = \frac{1}{2}\| \vec{w} \|^2 - \sum_{i=1}^{n} \alpha_i [y_i(\vec{w}\cdot\vec{r}_i + b) - 1]$$

$$\frac{\partial L}{\partial \vec{w}} = \vec{w} - \sum_{i=1}^{n} \alpha_i y_i \vec{r}_i = 0 \qquad \vec{w} = \sum_{i=1}^{n} \alpha_i y_i \vec{r}_i$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i b = 0 \qquad \sum_{i=1}^{n} \alpha_i y_i = 0$$

SVM linear – obtendo o hiperplano

$$L(\| \vec{w} \|, b, \alpha) = \frac{1}{2} \| \vec{w} \|^2 - \sum_{i=1}^{n} \alpha_i [y_i (\vec{w} \cdot \vec{r}_i + b) - 1]$$

$$\vec{w} = \sum_{i=1}^{n} \alpha_i y_i \vec{r}_i \qquad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\frac{1}{2} \| \vec{w} \|^2 = \frac{1}{2} \left(\sum_{i=1}^{n} \alpha_i y_i \vec{r}_i \right) \left(\sum_{j=1}^{n} \alpha_j y_j \vec{r}_j \right) = \frac{1}{2} \sum_{i}^{n} \sum_{j}^{n} \alpha_i \alpha_j y_i y_j \vec{r}_i \cdot \vec{r}_j$$

SVM linear – obtendo o hiperplano

$$L(\| \vec{w} \|, b, \alpha) = \frac{1}{2} \| \vec{w} \|^2 - \sum_{i=1}^{n} \alpha_i [y_i (\vec{w} \cdot \vec{r}_i + b) - 1]$$

$$\vec{w} = \sum_{i=1}^{n} \alpha_i y_i \vec{r}_i \qquad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\sum_{i=1}^{n} \alpha_i [y_i (\vec{w} \cdot \vec{r}_i + b) - 1] = \sum_{i=1}^{n} \alpha_i y_i \left(\sum_{j=1}^{n} \alpha_j y_j \vec{r}_j \cdot \vec{r}_i \right) + \sum_{i=1}^{n} \alpha_i y_i b - \sum_{i=1}^{n} \alpha_i$$

$$- \sum_{i=1}^{n} \alpha_i [y_i (\vec{w} \cdot \vec{r}_i + b) - 1] = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \vec{r}_i \cdot \vec{r}_j$$

SVM linear – obtendo o hiperplano

$$L(\| \vec{w} \|, b, \alpha) = \frac{1}{2} \| \vec{w} \|^2 - \sum_{i=1}^{n} \alpha_i [y_i (\vec{w} \cdot \vec{r}_i + b) - 1]$$

$$L(\| \vec{w} \|, b, \alpha) = \frac{1}{2} \sum_{i}^{n} \sum_{j}^{n} \alpha_i \alpha_j y_i y_j \vec{r}_i \cdot \vec{r}_j + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \vec{r}_i \cdot \vec{r}_j$$

$$L(\| \vec{w} \|, b, \alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \vec{r}_i \cdot \vec{r}_j$$

SVM linear – obtendo o hiperplano

Agora só resta maximizar os multiplicadores de modo que

$$max_\alpha \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \vec{r}_i \cdot \vec{r}_j \right\}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\alpha_i \geq 0 \;\; \forall \, i = 1,2, \ldots, n$$

$\alpha_i > 0$ se é vetor de suporte

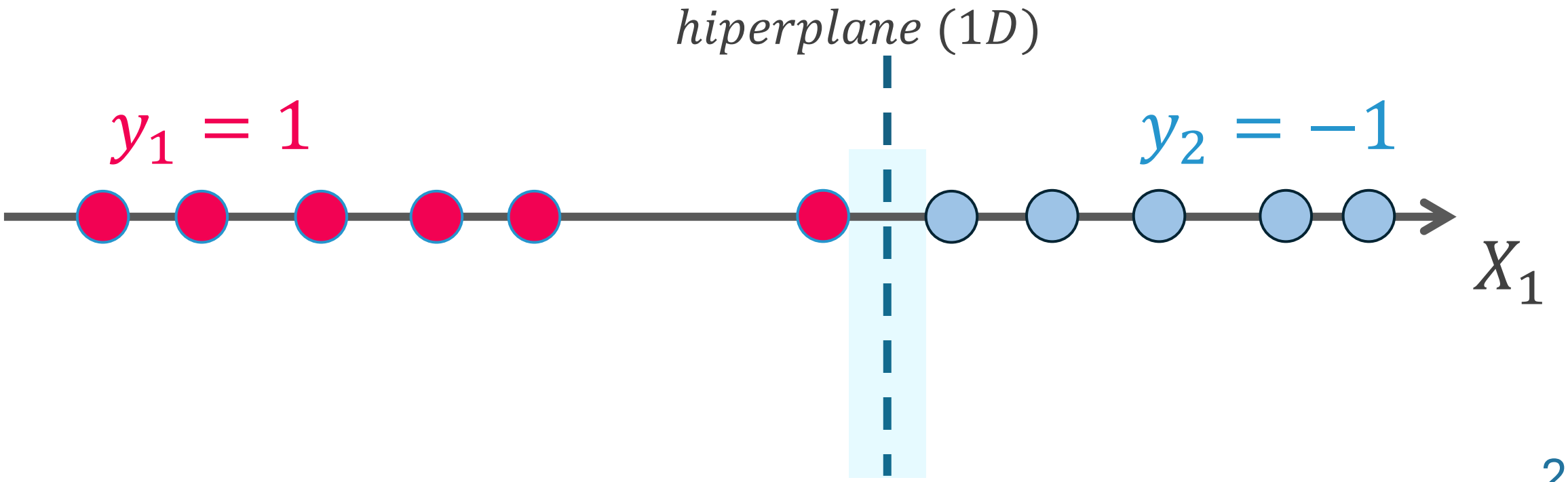
(cálculo geralmente feito numericamente)

Entretanto, perceba que teremos uma dependência em $\vec{r}$ apenas em termos do produto interno $\vec{r}_i \cdot \vec{r}_j$. Isso será muito útil

SVM linear – Hard-margin vs Soft-margin

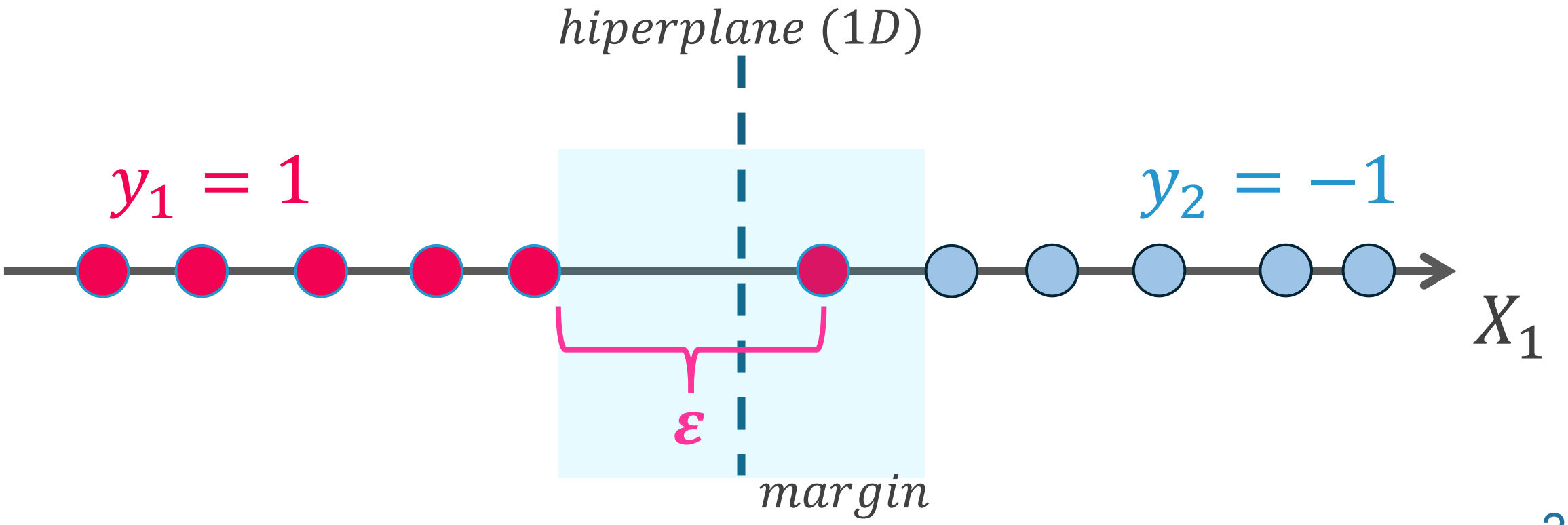
Em dados reais, é praticamente impossível sempre encontrar hiperplanos que linearmente dividem os dados

SVM hard-margin busca um hiperplano perfeito, não admite erros. Logo ele tem grandes chances de levar a *overfitting*

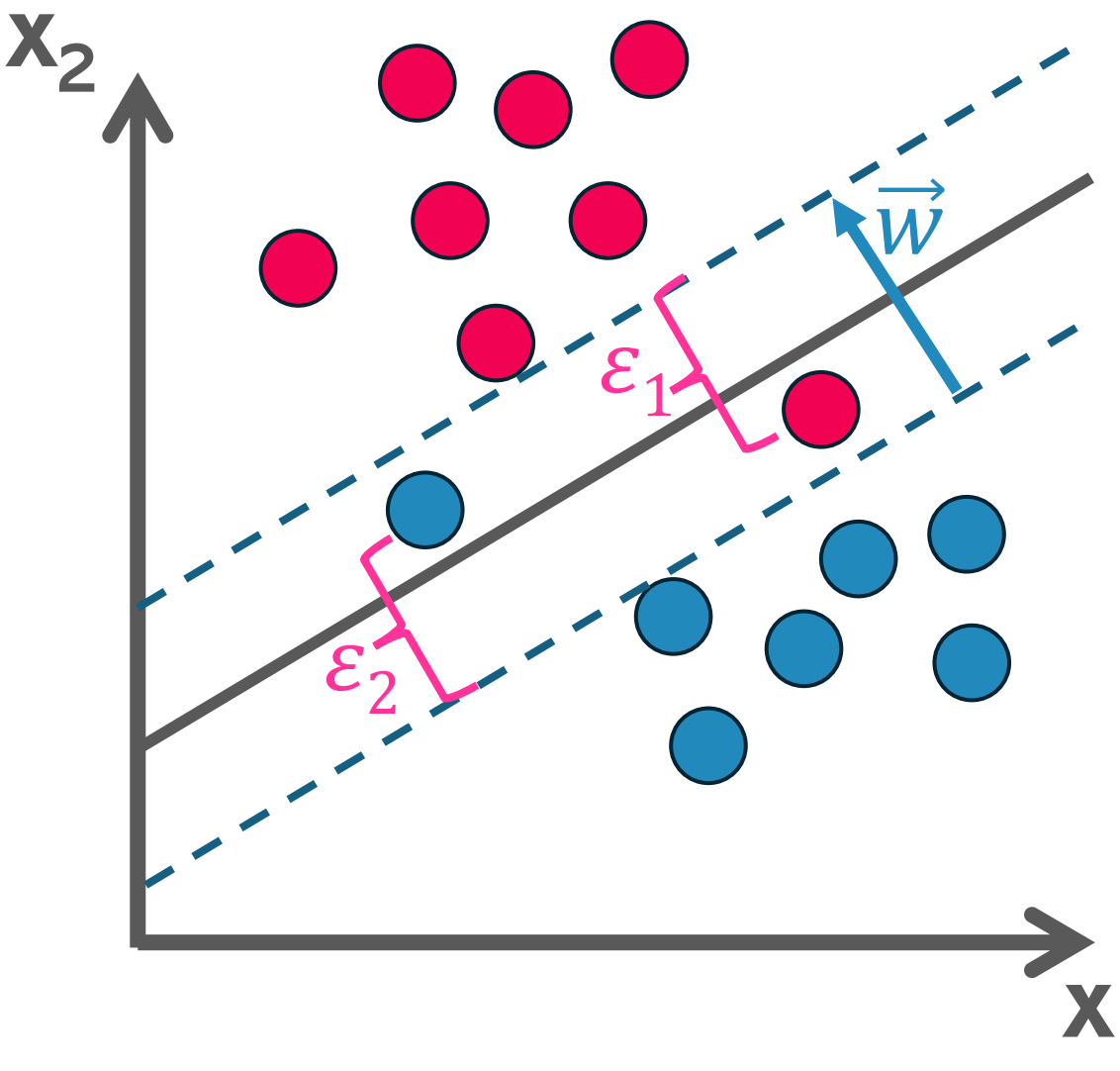


SVM linear – Hard-margin vs Soft-margin

O SVM soft-margin foi projetado para admitir erros devido ao ruído dos dados, se adequando mais aos casos reais.
Com isso, um novo termo de penalização é introduzido (***C***)



SVM linear – Hard-margin vs Soft-margin



$$Min_{\|\vec{w}\|,b}\left(\frac{1}{2} \| \vec{w} \|^2\right)$$

Restrição do SVM hard-margin

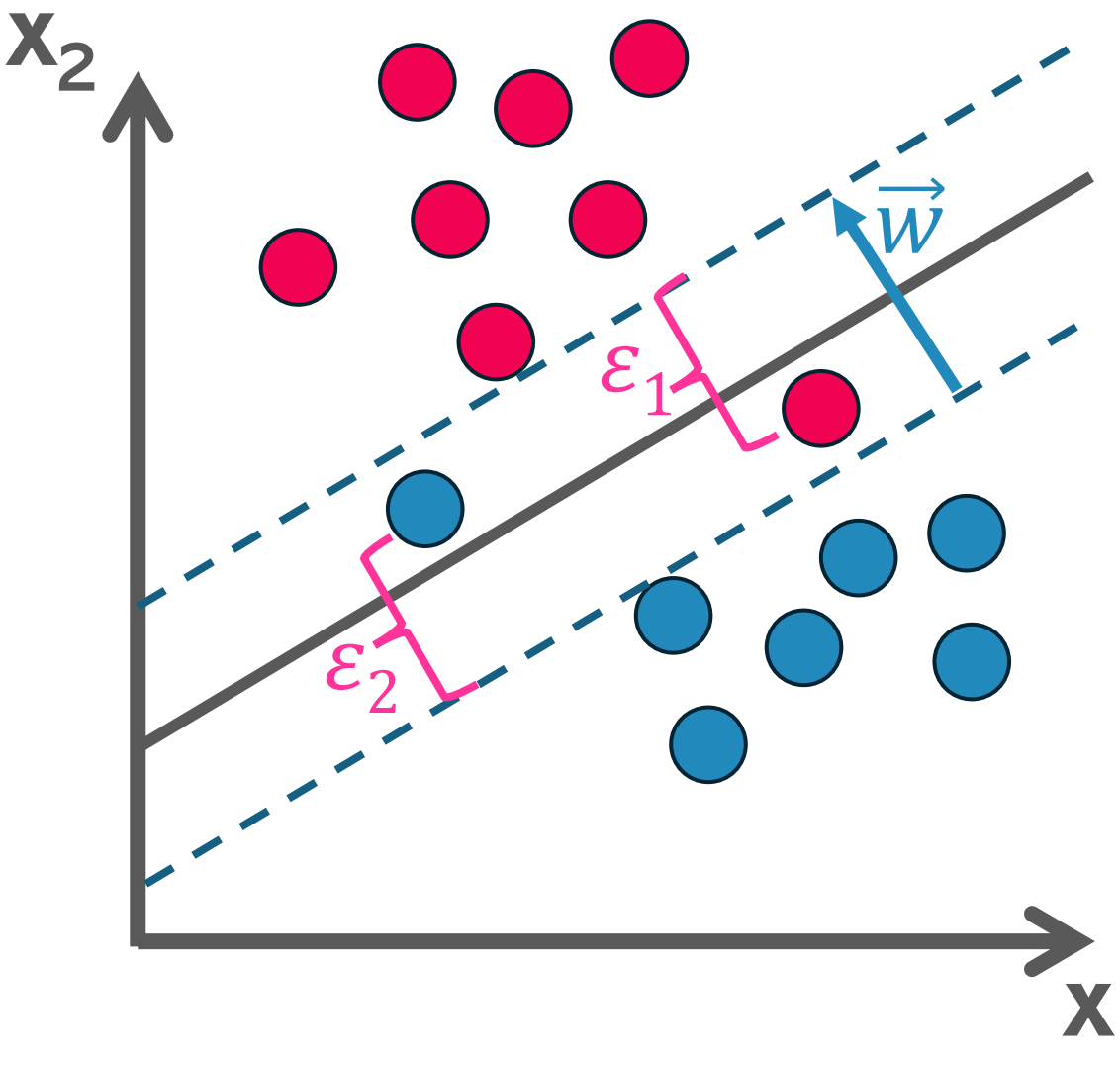
$$y_i(\vec{w} \cdot \vec{r}_i + b) \geq 1$$

Restrição do SVM soft-margin

$$y_i(\vec{w} \cdot \vec{r}_i + b) \geq 1 - \varepsilon_i$$

Incorporar o erro permite flexibilidade na hora de obter o hiperplano

SVM linear – Hard-margin vs Soft-margin



Restrição do SVM soft-margin

$$y_i(\vec{\boldsymbol{w}} \cdot \vec{\boldsymbol{r}}_i + b) \geq 1 - \varepsilon_i$$

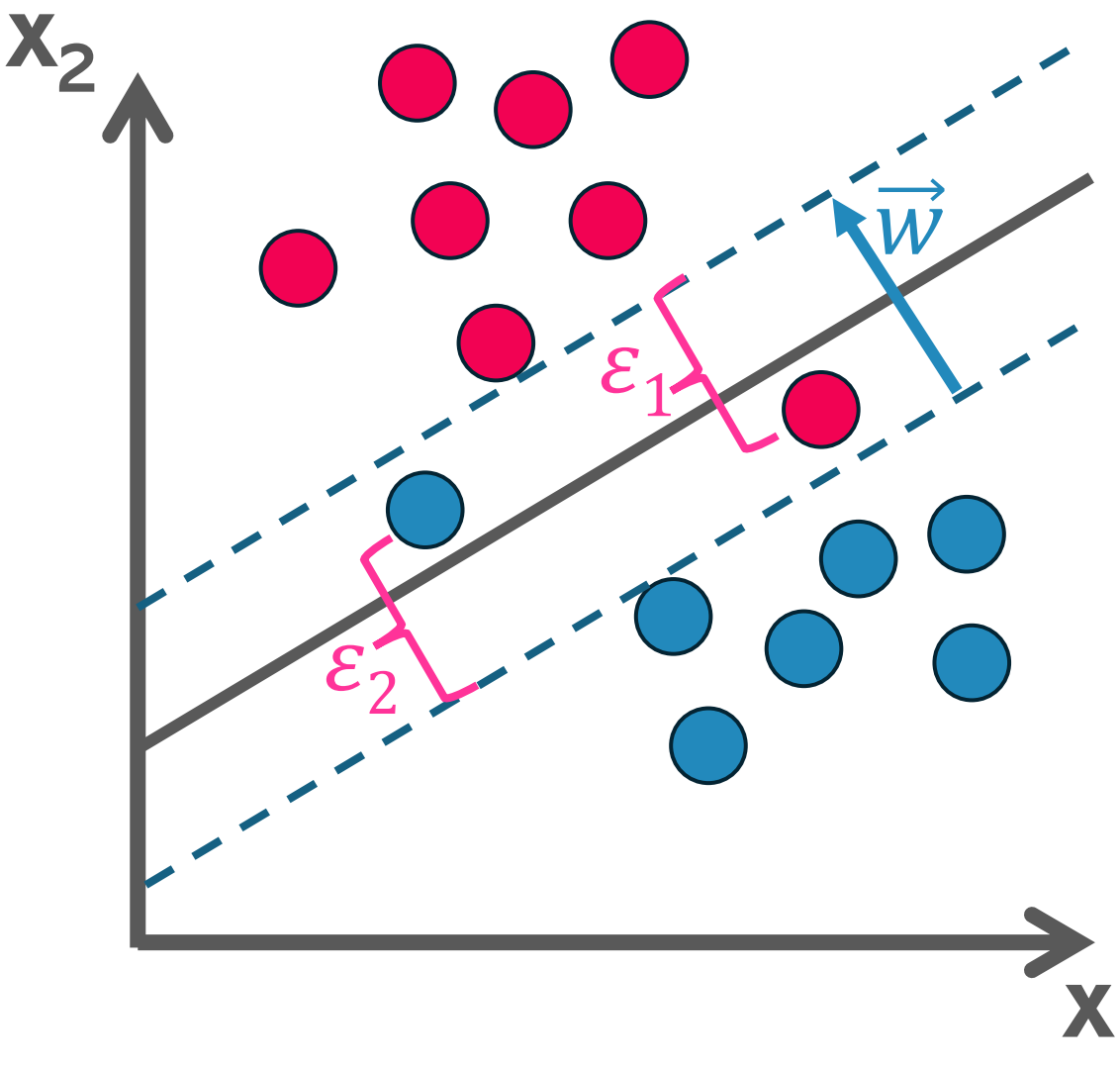
Entretanto, nós não podemos simplesmente assumir qualquer ε. Sua escolha também deve ser otimizada

$$Min_{\|\vec{w}\|,b,\varepsilon_i}\left(\frac{1}{2}\|\vec{\boldsymbol{w}}\|^2 + C\sum_{i=1}^{m}\varepsilon_i\right)$$

$$\varepsilon_i \in [0,1]$$

C é chamado de hiperparâmetro de regularização (penalização ou custo)

SVM linear – Hard-margin vs Soft-margin



Analogamente ao anterior, desenvolvendo os multiplicadores de Lagrange chegamos em:

$$Máx_{\alpha_i}\left(\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j}^{n}\alpha_i\alpha_j y_i y_j\,\vec{r}_i\cdot\vec{r}_j\right)$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

$$\alpha_i \geq C \;\; \forall\, i = 1,2,\ldots,n$$

SVM linear – Hard-margin vs Soft-margin

$$Máx_{\alpha_i}\left(\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j}^{n}\alpha_i\alpha_j y_i y_j \vec{r}_i \cdot \vec{r}_j\right) \qquad \sum_{i=1}^{n}\alpha_i y_i = 0$$

$$\alpha_i \geq C \;\; \forall\, i = 1,2,\ldots,n$$

Quanto menor o C, maior tolerância a erros
(permite que ε_i aumente, margem grande, *underfitting*).

Quanto maior o C, menor a tolerância a erros
(permite que ε_i diminua, margem pequena, *overfitting*).

Ou seja, o C é uma espécie de ponderador dos erros, e deve ser **escolhido** a priori adequadamente (**cross-validation**)

SVM linear – Hard-margin vs Soft-margin

Por outro lado, agora temos um termo extra (ε_i) que deve ser incluído no problema de otimização, resolvido também com multiplicadores de Lagrande

$$Min_{\|\vec{w}\|,b,\varepsilon_i}\left(\frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n}\varepsilon_i\right) \qquad \varepsilon_i \in [0,1]$$

Restrição do SVM soft-margin

$$y_i(\vec{w}\cdot\vec{r}_i + b) \geq 1 - \varepsilon_i$$

$$Máx_{\alpha_i}\left(\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j}^{n}\alpha_i\alpha_j y_i y_j\, \vec{r}_i\cdot\vec{r}_j\right)$$

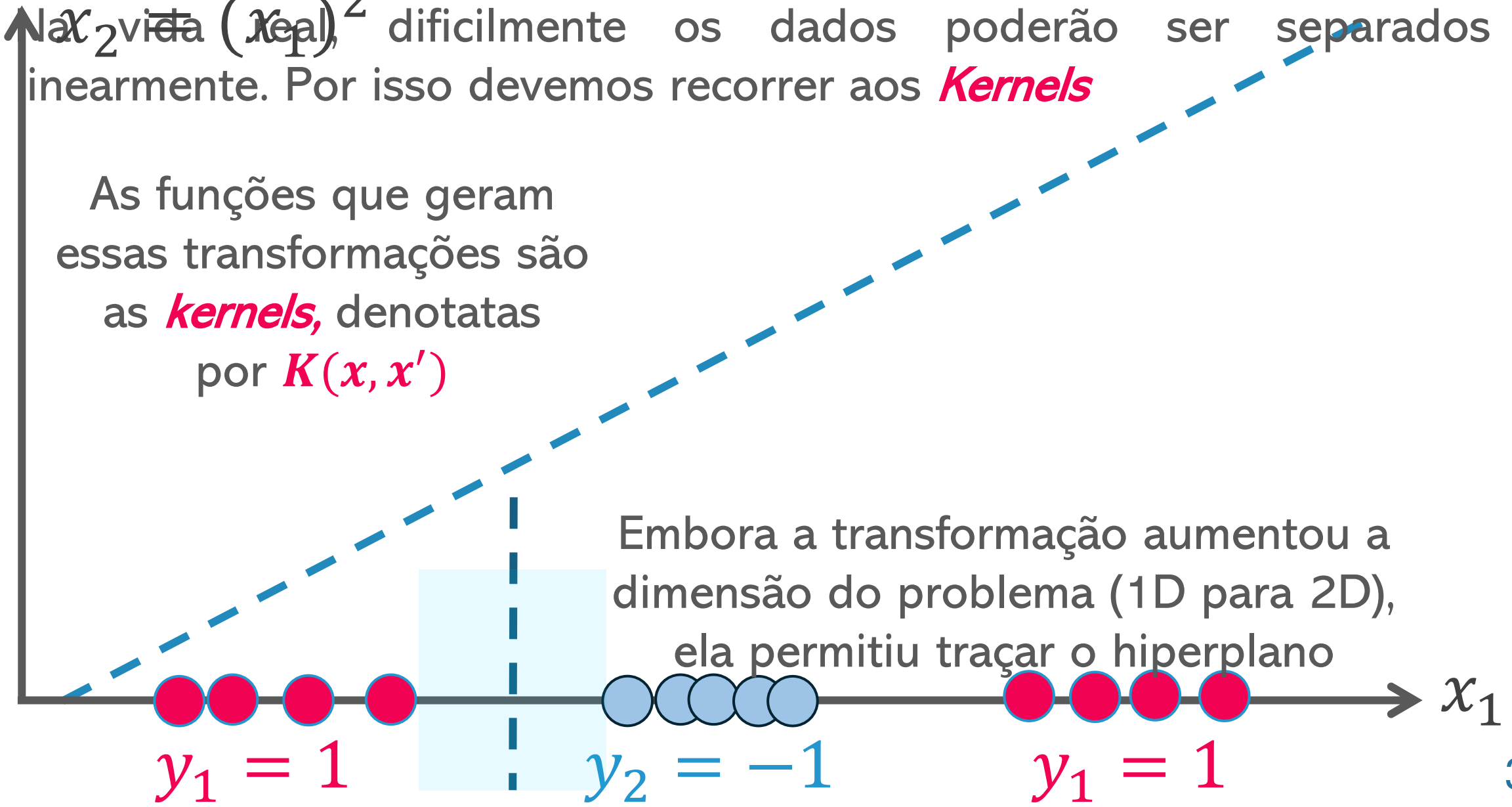
Ao final, também teremos que a dependência em $\vec{r}$ será do tipo $\vec{r}_i\cdot\vec{r}_j$

Truque de Kernel

Truque de kernel

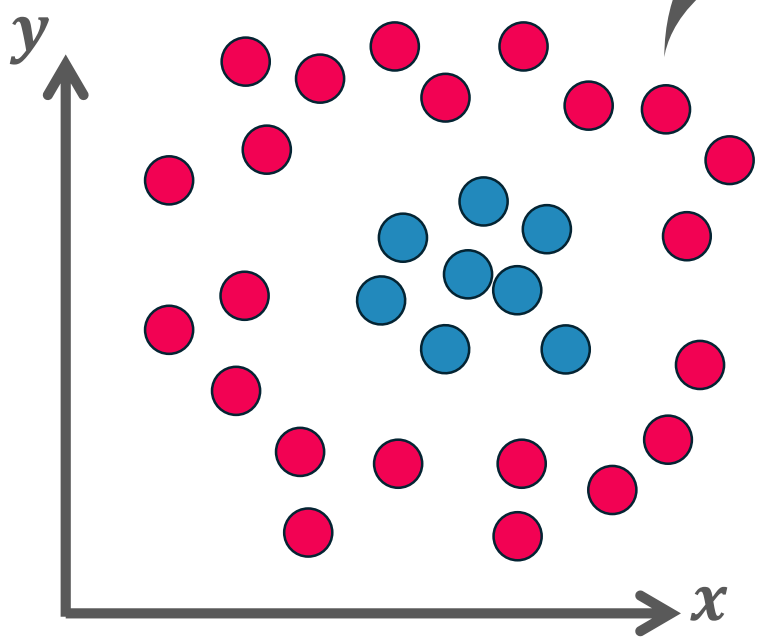
Na vida real, $x_2 = (x_1)^2$ dificilmente os dados poderão ser separados inearmente. Por isso devemos recorrer aos ***Kernels***

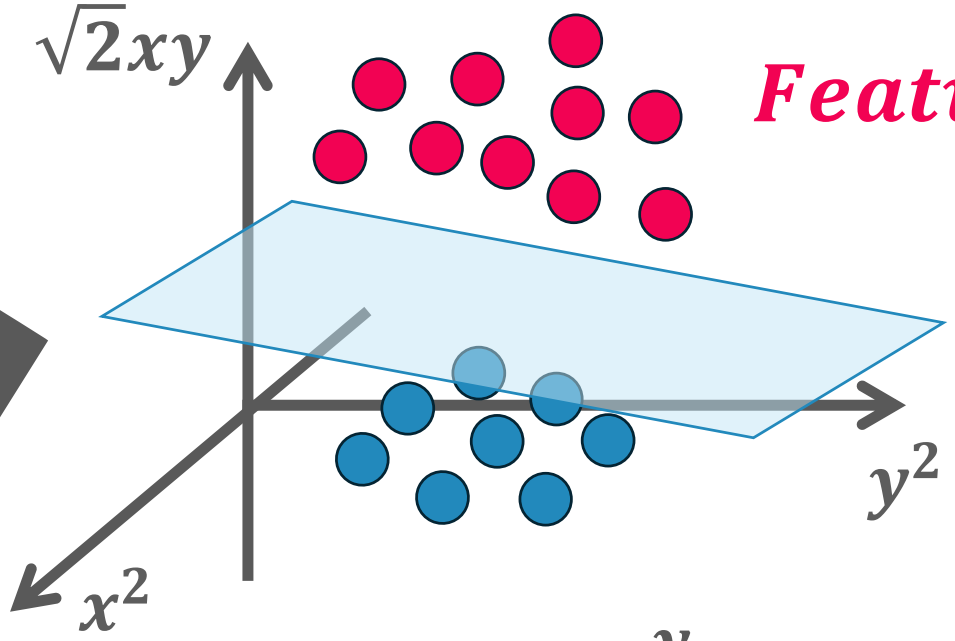
As funções que geram essas transformações são as ***kernels,*** denotatas por $K(x, x')$



Truque de kernel

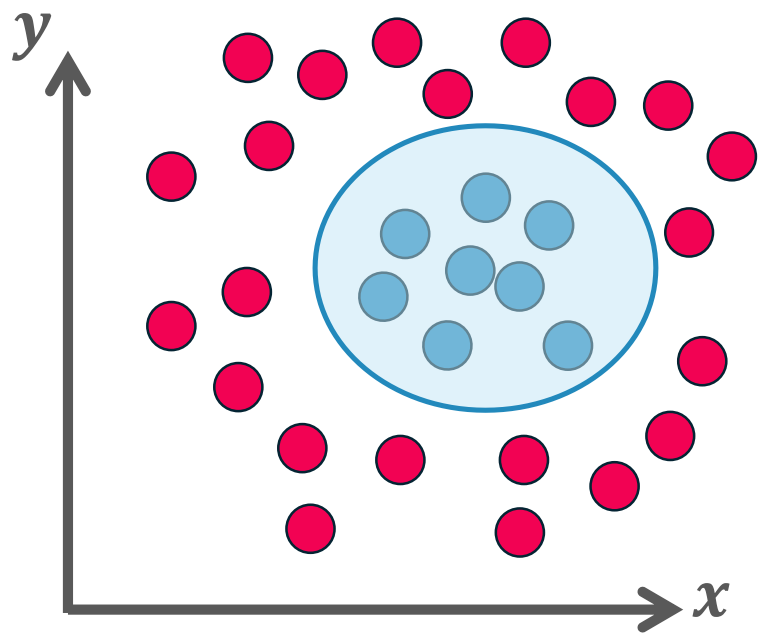
ϕ: $\mathbb{R}^2 \Rightarrow \mathbb{R}^3 \quad \forall (x,y) \in \mathbb{R}^2$





Feature space

Original space



Ex: $\phi(x,y) = \left(x^2, \sqrt{2}xy, y^2\right)$

Truque de kernel – SVM não-linear

$\phi: \mathbb{R}^m \Rightarrow \mathbb{R}^{m'} \quad (m' > m)$

É a função que mapeia os dados iniciais no espaço de alta dimensão

$$Min_{\|\vec{w}\|,b,\varepsilon_i} \left(\frac{1}{2} \| \vec{w} \|^2 + C \sum_{i=1}^{n_{treino}} \varepsilon_i \right) \qquad \varepsilon_i \in [0,1]$$

Incluímos a transformações na restrição:

$y_i(\vec{w}\phi(\vec{x}_i) + b) \geq 1 - \varepsilon_i$

Truque de kernel – SVM não-linear

$$Min_{\|\vec{w}\|,b,\varepsilon_i}\left(\frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n}\varepsilon_i\right)$$

Restrição

$$y_i(\vec{w}\cdot\phi(\vec{r}_i)+b) \geq 1-\varepsilon_i$$

Lagrange

$$Máx_{\alpha_i}\left(\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j}^{n}\alpha_i\alpha_j y_i y_j\,\phi(\vec{r}_i)\cdot\phi(\vec{r}_j)\right)$$

Ao longo de todo o problema só teremos que calcular o produto escalar das transformações $\phi(\vec{r}_i)\cdot\phi(\vec{r}_j)$, em vez de calcular a transformação $\phi(.)$ ou, em outras palavras, transformar os dados de entrada para o espaço de alta-dimensionalidade (demorado e custoso)

Truque de kernel – SVM não-linear

Explorando isso, utilizam-se as funções Kernel, que **reproduzem** como seria o resultado do produto escalar $\phi(\vec{r}_i) \cdot \phi(\vec{r}_j)$ no espaço de alta dimensão, porém, utilizando para isso os dados no espaço original. Esse é o famoso truque de kernel

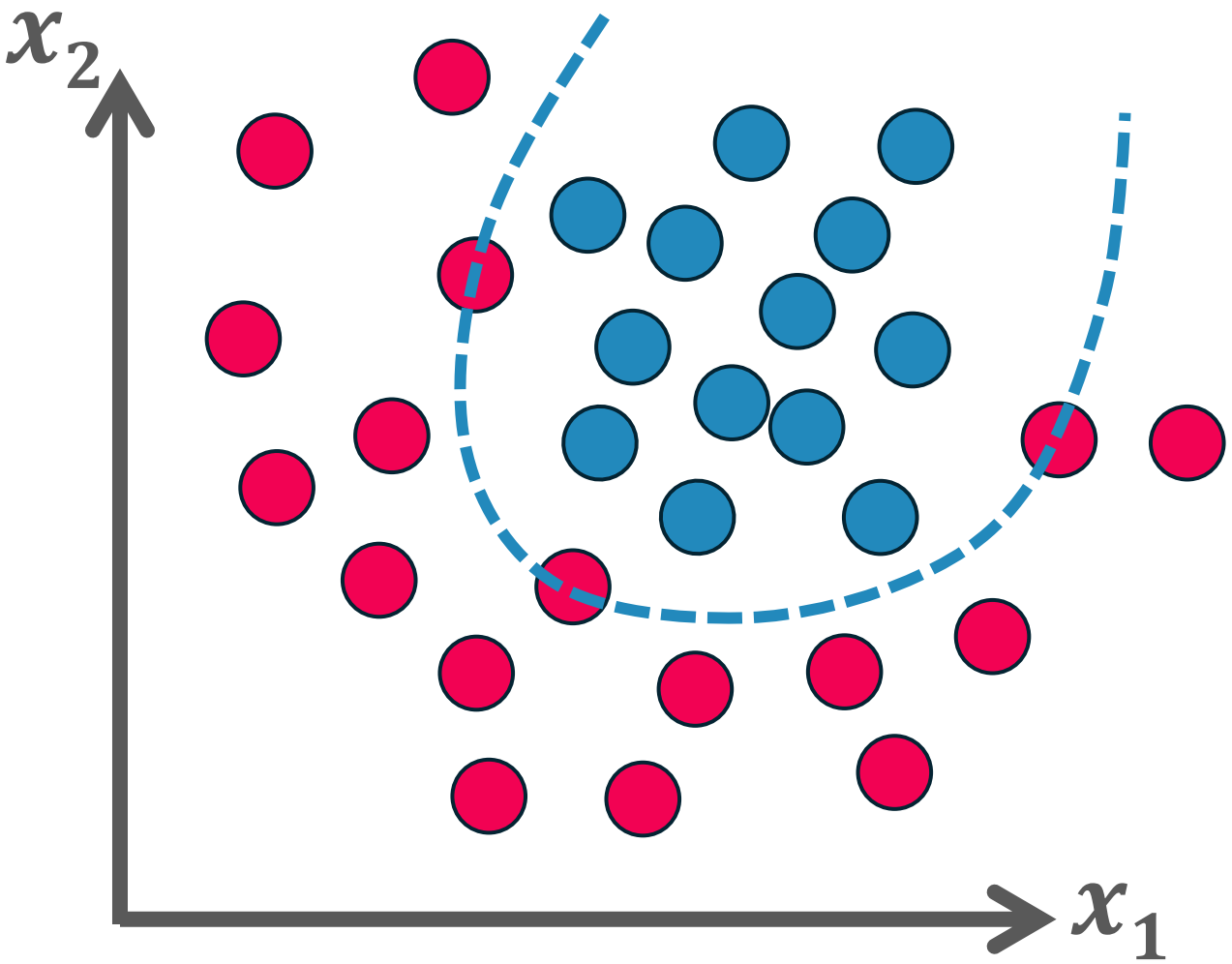
$$K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j) \begin{cases} = \vec{x}_i \cdot \vec{x}_j & \text{(linear)} \\ = (\gamma \vec{x}_i \cdot \vec{x}_j + coef)^d & \text{(polinomial)} \\ = exp[-\gamma \parallel \vec{x}_i - \vec{x}_j \parallel^2] & \text{(RBF)} \\ = tanh(\gamma \vec{x}_i \cdot \vec{x}_j + coef) & \text{(Sigmoid)} \end{cases}$$

Truque de kernel – SVM não-linear

Em outras palavras, a função $K(\vec{x}_i, \vec{x}_j)$ mapeia pares de vetores $(\vec{x}_i, \vec{x}_j)$ direto em um produto escalar. Dessa forma, não é necessário calcular explicitamente o mapeamento dos dados no espaço de maior dimensao pois ele é feito implicitamente no espaço original usando K

$$K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j) \begin{cases} = \vec{x}_i \cdot \vec{x}_j & \text{(linear)} \\ = (\gamma \vec{x}_i \cdot \vec{x}_j + coef)^d & \text{(polinomial)} \\ = exp[-\gamma \parallel \vec{x}_i - \vec{x}_j \parallel^2] & \text{(RBF)} \\ = tanh(\gamma \vec{x}_i \cdot \vec{x}_j + coef) & \text{(Sigmoid)} \end{cases}$$

Truque de kernel – SVM não-linear

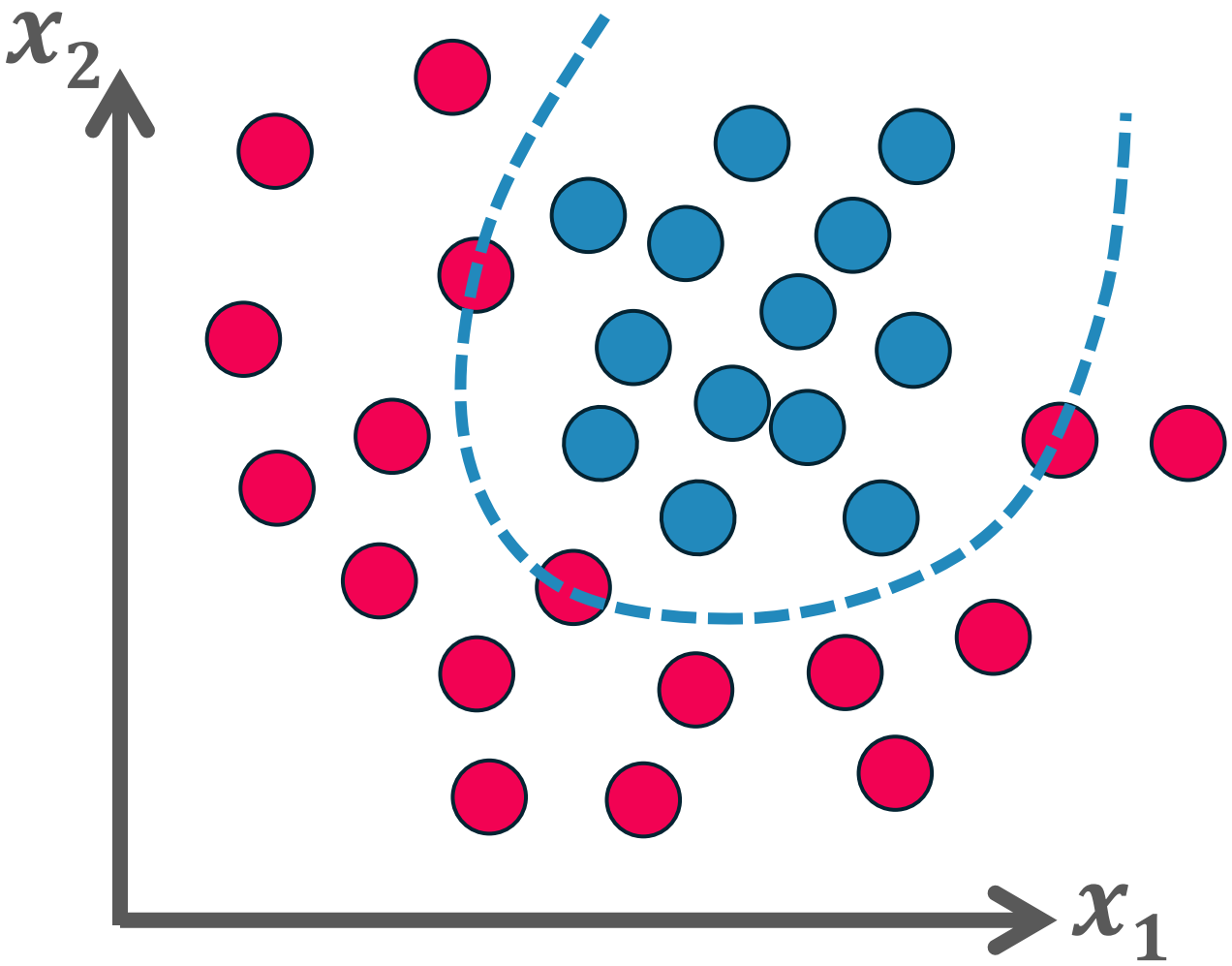


2º Polynomial

$$K(x_i, x_j) = (\gamma x_i x_j + coef)^2$$

O termo ***coef*** atua como um deslocamento (ou bias). Ao adicioná-lo à multiplicação interna, mesmo quando o produto interno é baixo ou nulo, o kernel ainda produz um valor não nulo.

Truque de kernel – SVM não-linear



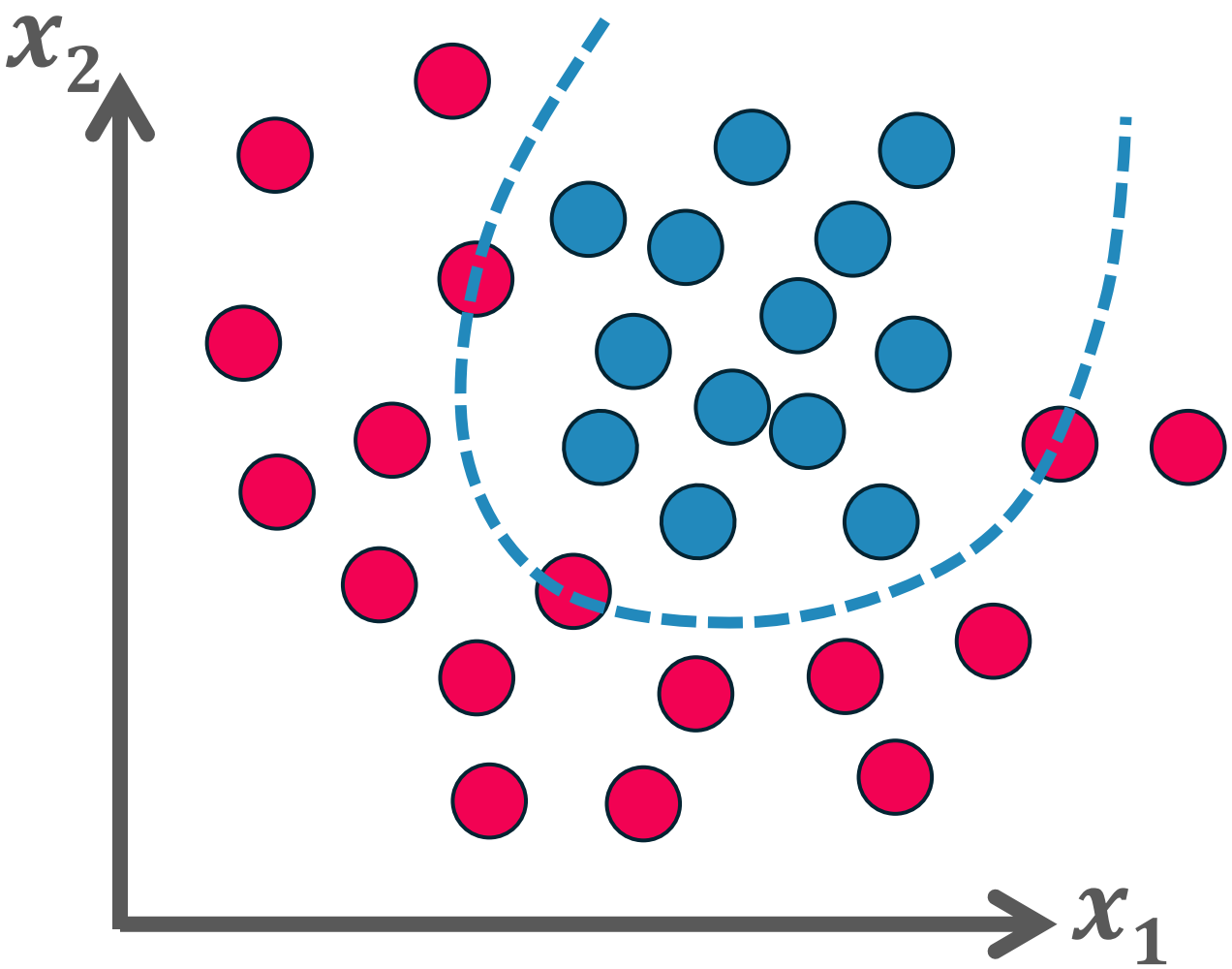
2º Polynomial

$$K(x_i, x_j) = (\gamma x_i x_j + coef)^2$$

γ é um coeficiente de escalamento

Um valor pequeno diminui a importância do produto interno. Isso faz com que os pontos sejam vistos como mais "próximos" na transformação polinomial, levando a uma superfície de decisão mais suave e menos complexa.

Truque de kernel – SVM não-linear



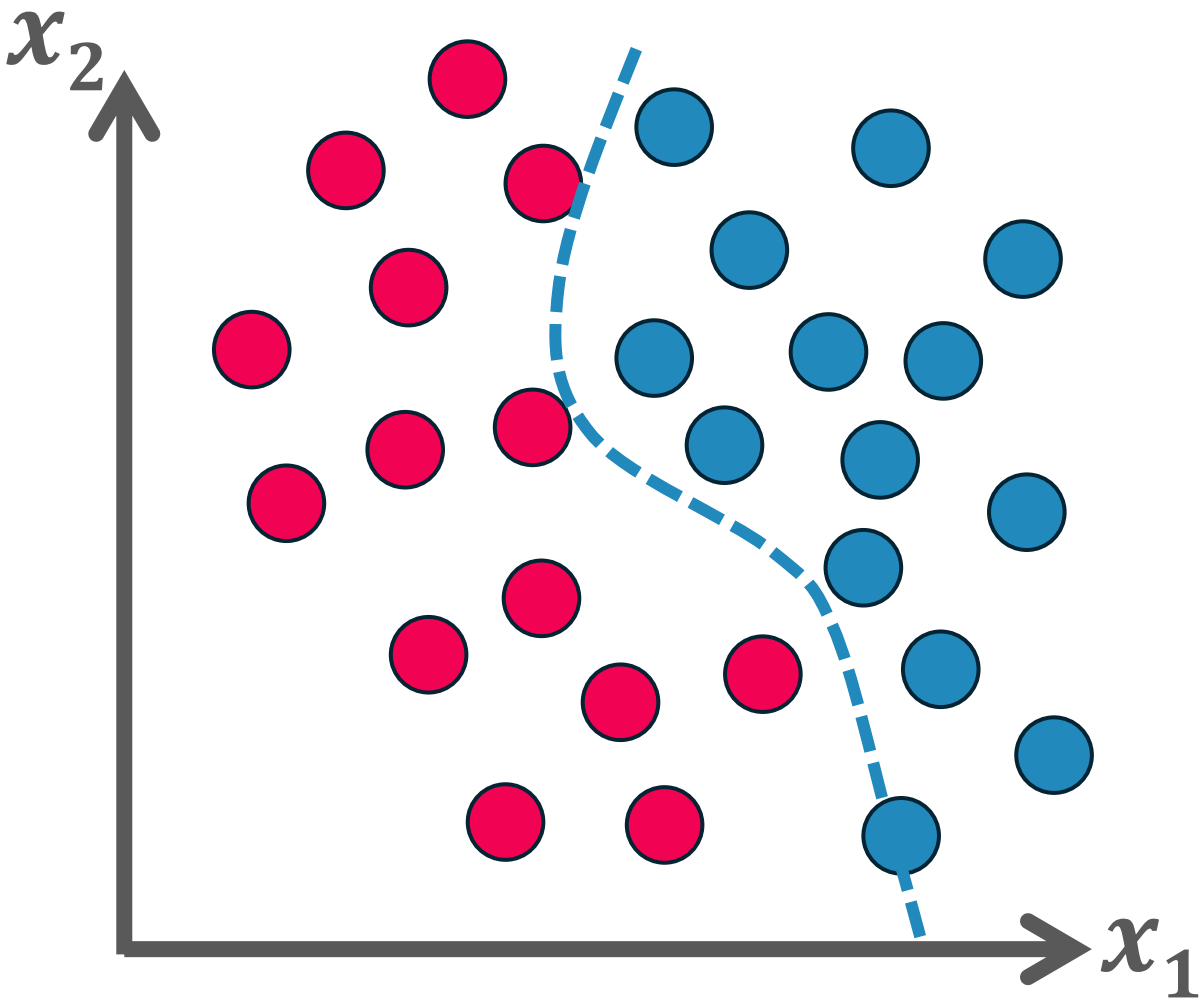
2º Polynomial

$$K(x_i, x_j) = (\gamma x_i x_j + coef)^2$$

γ é um coeficiente de escalamento

Valores altos amplificam o produto interno, intensificando a distinção entre pontos com ângulos abruptos. Isso pode resultar em uma fronteira de decisão mais complexa e adaptada aos detalhes particulares (risco de overfitting)

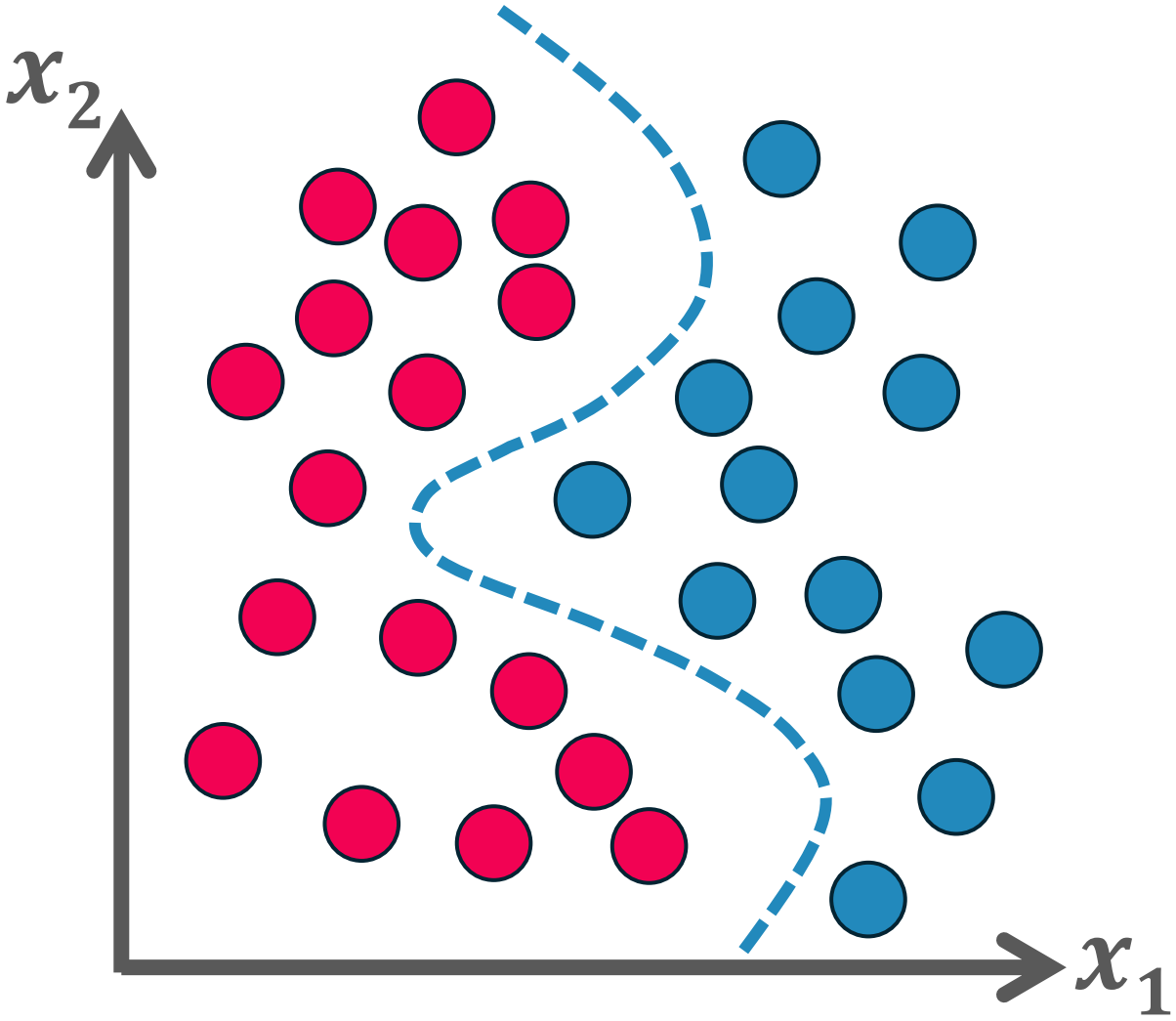
Truque de kernel – SVM não-linear



3º Polynomial

$$K(x_i, x_j) = (\gamma x_i x_j + coef)^3$$

Truque de kernel – SVM não-linear

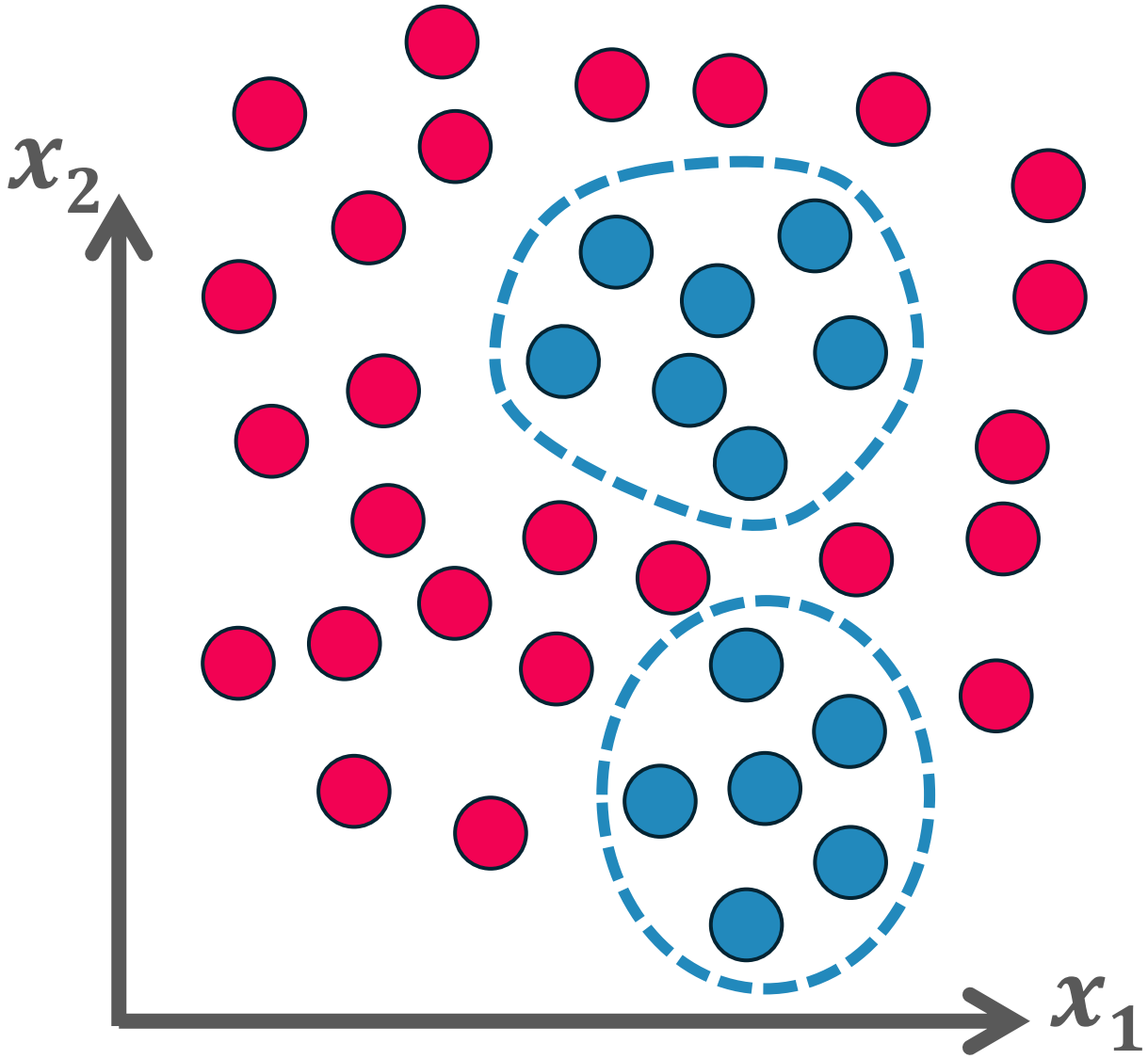


Dº Polynomial

$$K(x_i, x_j) = (\gamma x_i x_j + coef)^D$$

Um grau ***D*** menor mantêm a complexidade do modelo relativamente baixa. Quanto maior ***D***, mais angulações o hiperplano terá (risco de overfitting). Em geral, grau 2 ou 3 pode capturar interações simples entre as variáveis sem exagerar a complexidade (mais usados)

Truque de kernel – SVM não-linear

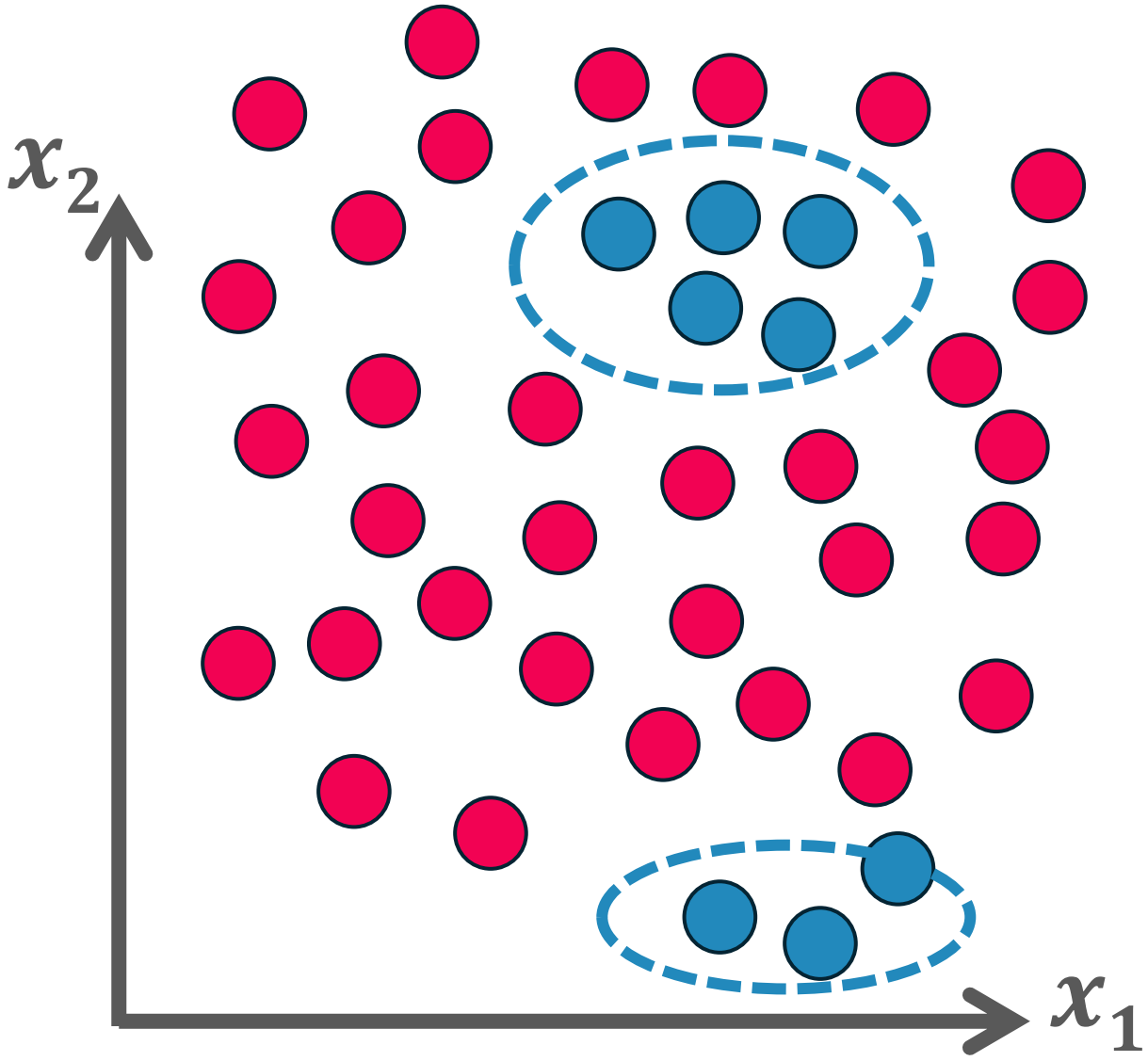


Radial basis

$$K(x_i, x_j) = \exp[-\gamma \parallel x_i - x_j \parallel^2$$

Com um valor baixo de γ diminui o argumento da exponencial, fazendo com que ela caia mais lentamente.
Isso delimita uma área maior no espaço original. Logo, o modelo resultante tende a ser mais suave e pode subestimar as variações locais nos dados (underfitting).

Truque de kernel – SVM não-linear



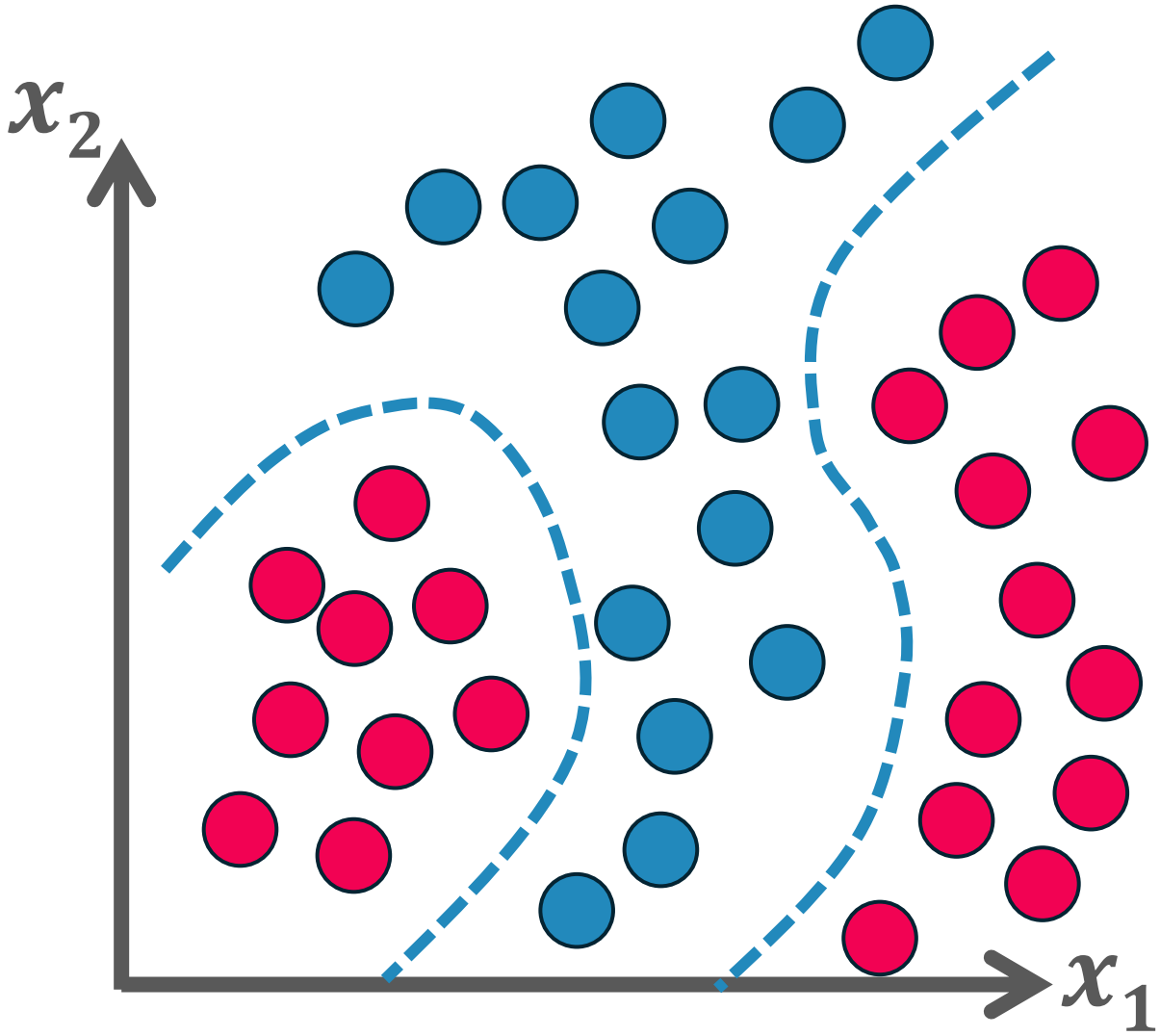
Radial basis

$$K(x_i, x_j) = \exp[-\gamma \parallel x_i - x_j \parallel^2$$

Com um valor alto de γ causa uma rápida decaída do expoente – mesmo pequenas distâncias entre os pontos podem levar a valores de kernel muito próximos a zero.

Isso faz com que a fronteira de decisão se ajuste finamente aos dados de treinamento (menos área delimitada), mas aumenta o risco de overfitting

Truque de kernel – SVM não-linear

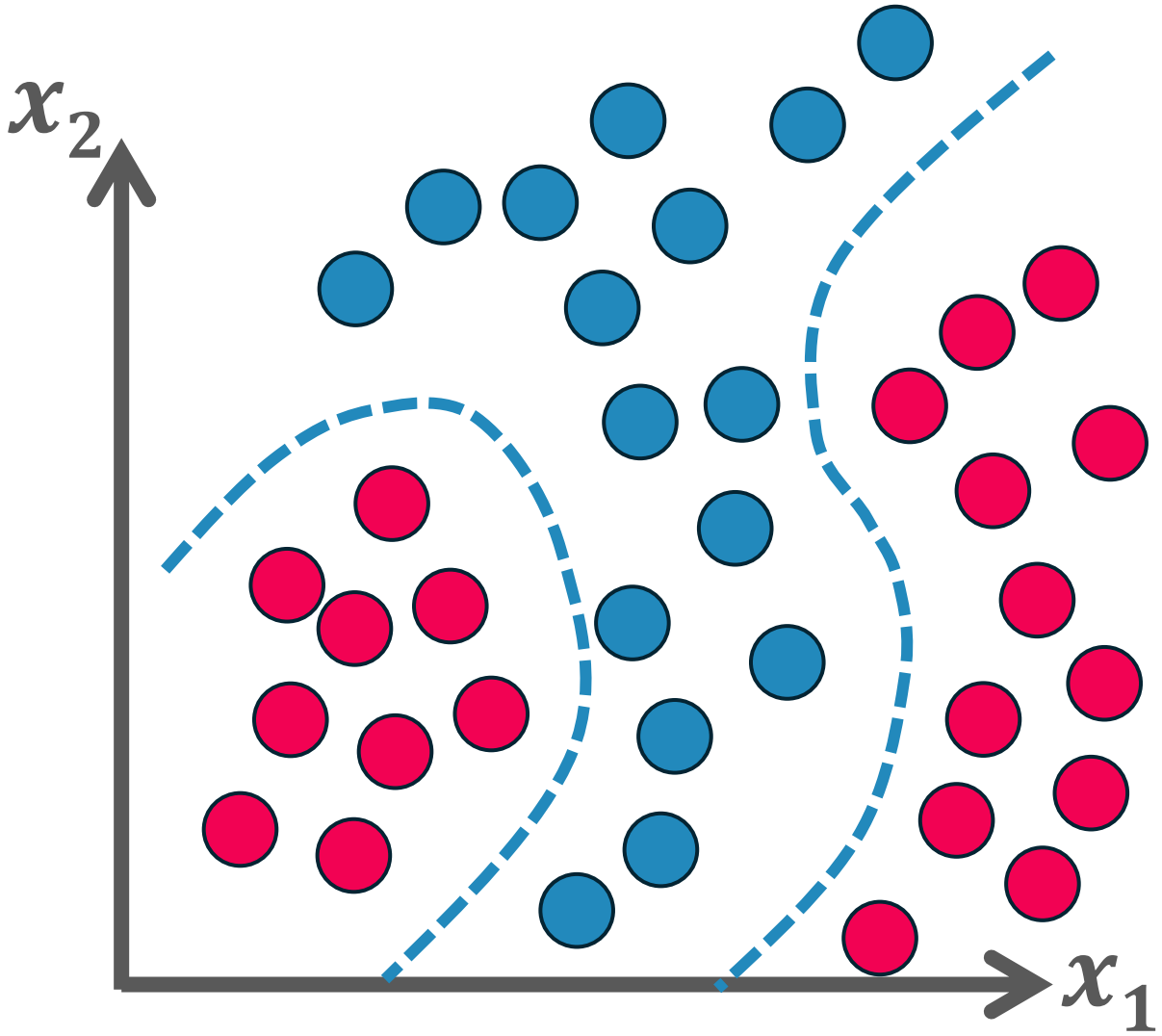


Sigmoid

$$K(x_i, x_j) = tanh(\gamma x_i x_j + coef)$$

Um γ baixo reduz a influência do produto interno antes de aplicar a função tangente hiperbólica.
Isso resulta em uma função kernel que varia de forma mais suave (curvas com angulações menores), mas que pode não capturar suficientemente as não linearidades presentes nos dados

Truque de kernel – SVM não-linear



Sigmoid

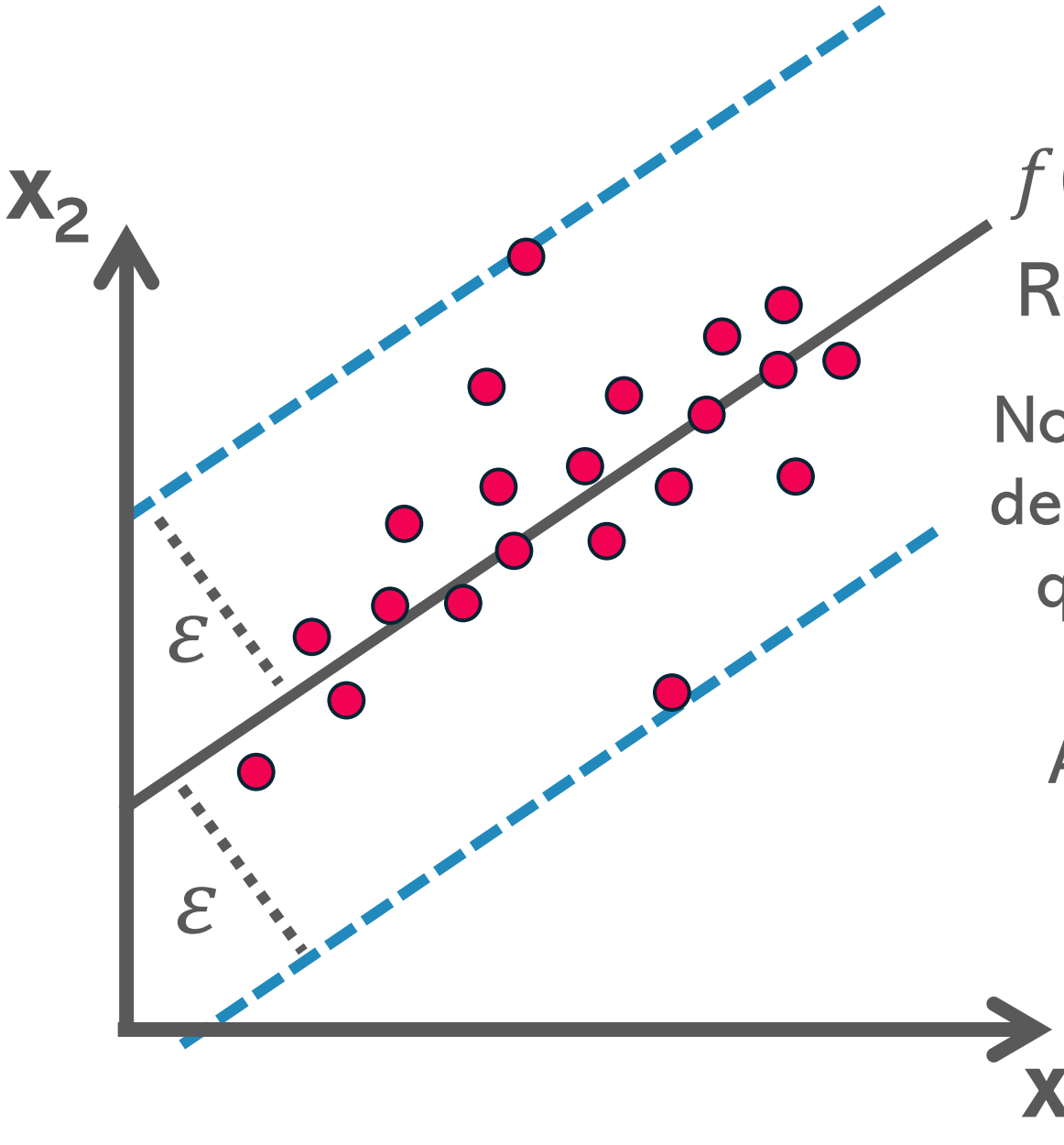
$$K(x_i, x_j) = tanh(\gamma x_i x_j + coef)$$

Um γ alto amplificam o produto interno e pode levar curvas muito complexas (muitas angulações), novamente, com risco de overfitting.

O *coef* deslocando o argumento da função *tanh*. Um ajuste apropriado pode ajudar a centralizar a função de forma que ela trace curvas melhores (partindo de regiões mais proícias para diferenciar os dados)

SVM aplicado a regressão

SVM aplicado a regressão - SVR



$$f(r_i) = (\vec{w} \cdot \vec{r} + b)$$

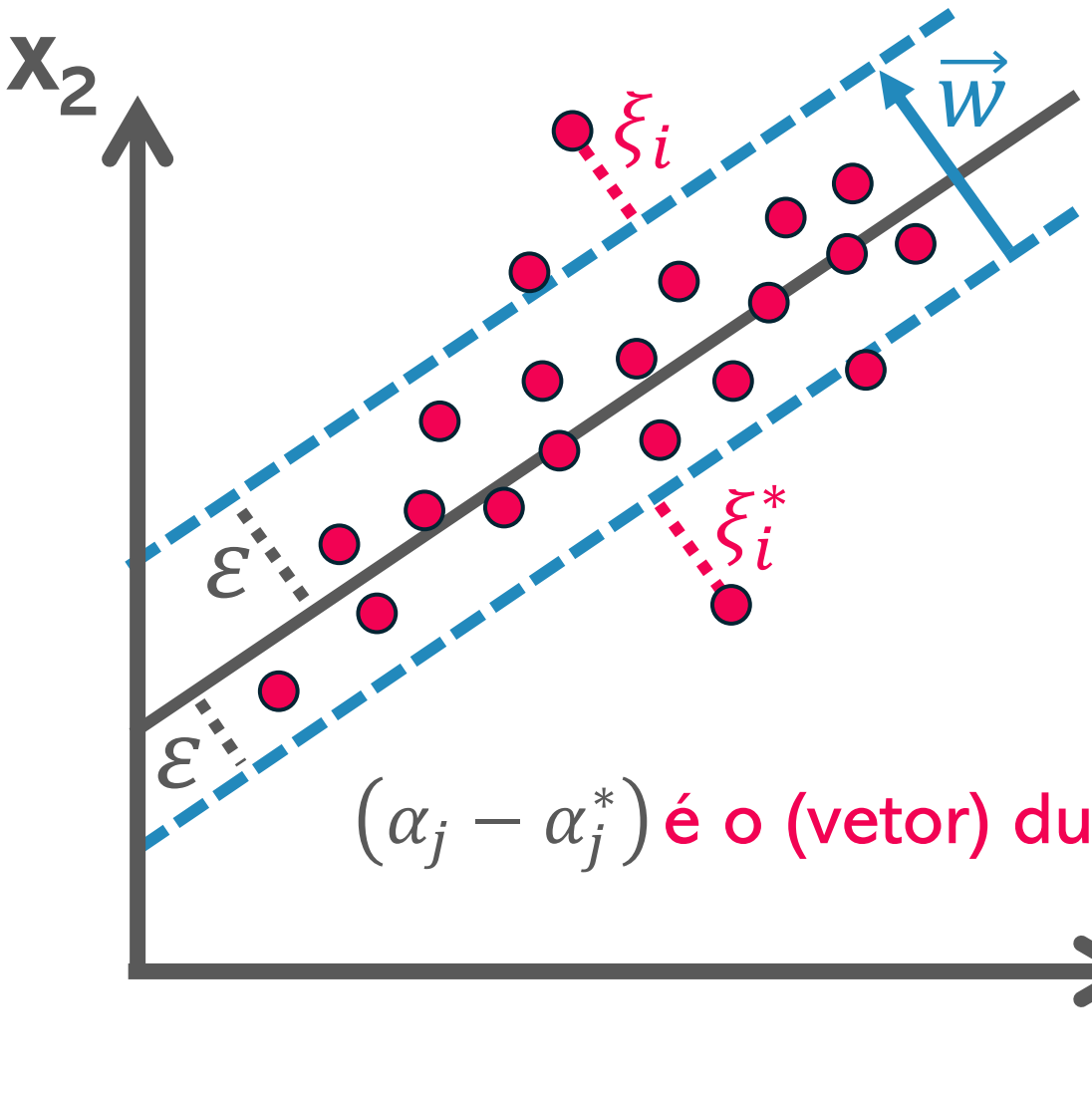
Restrição: $|y_i - f(r_i)| \leq \varepsilon$

No SVR aplicamos o conceito de "tubo de dispersão", para buscar uma função que aceite amostras cujo erro é $\leq \varepsilon$

A formulação do problema dessa forma não gera resultados muito precisos, pois assumimos uma margem ε muito grande (Analogo a hard-margin)

SVM aplicado a regressão - SVR

Note que ξ_i não precisa necessariamente ser igual ξ_i^*.



Introduzimos então variáveis de tolerância (ou folga) ξ_i e ξ_i^*. Em outras palavras, aceitamos predições erradas além da margem

$$Min_{\|\vec{w}\|,b,\xi_i^*,\xi_i}\left(\frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{m}\xi_i^* + \xi_i\right)$$

Restrições:

$$y_i - f(r_i) \leq \varepsilon + \xi_i$$
$$y_i - f(r_i) \geq \varepsilon + \xi_i^*$$
$$\xi_i^*\ \xi_i \geq 0\ \forall i$$

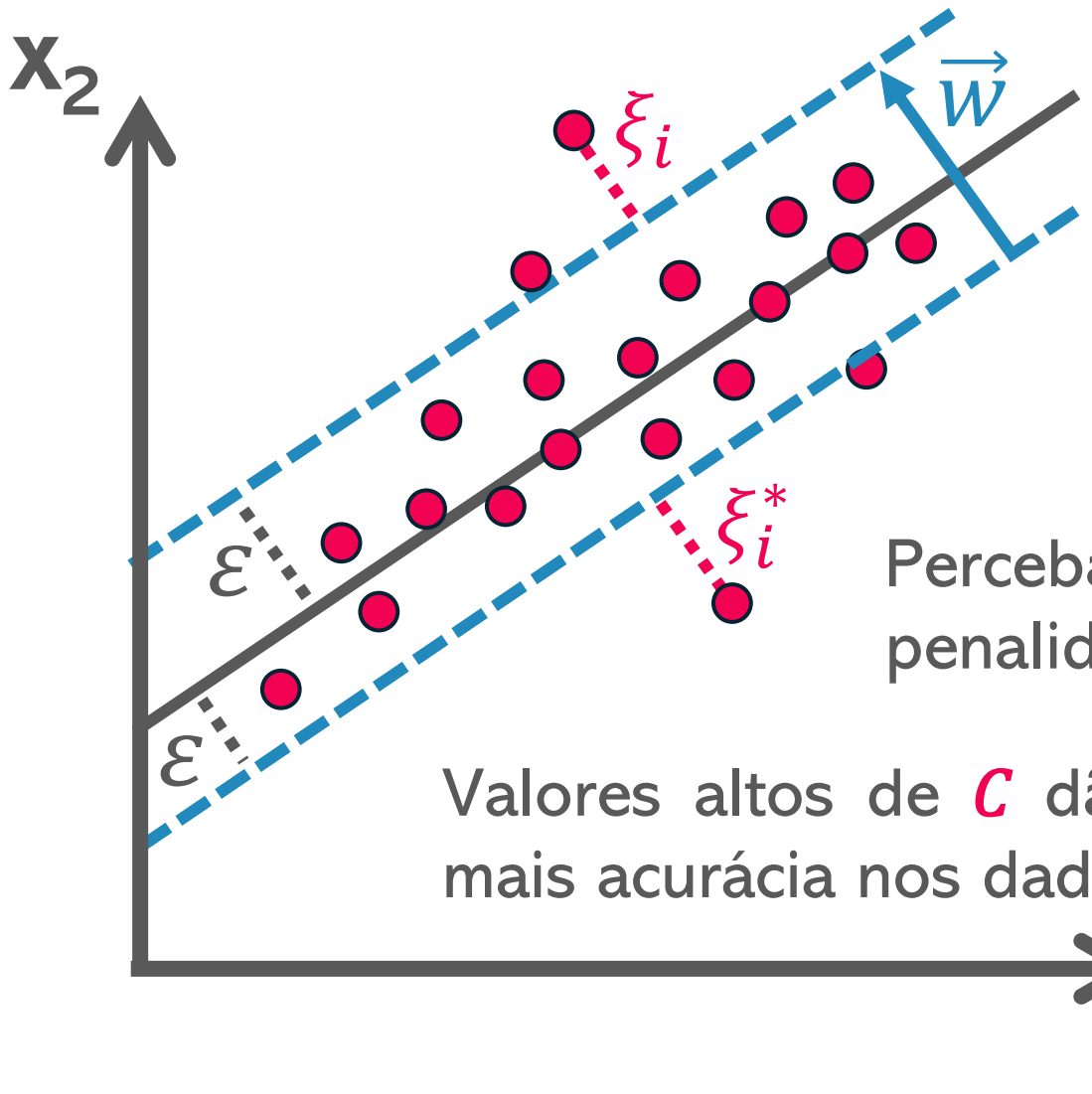
Multuplicadores de Lagrange

$$\hat{y}_i = \sum_{j}^{n}(\alpha_j - \alpha_j^*)\,K(\vec{r}_i, \vec{r}_j) + b$$

$(\alpha_j - \alpha_j^*)$ é o (vetor) dual

$$0 < \alpha_i^*, \alpha_i \leq C \quad \forall\, i = 1,2,\ldots,n_{vetsup}$$

SVM aplicado a regressão - SVR



$$Min_{\|\vec{w}\|,b,\xi_i^*,\xi_i}\left(\frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{m}\xi_i^* + \xi_i\right)$$

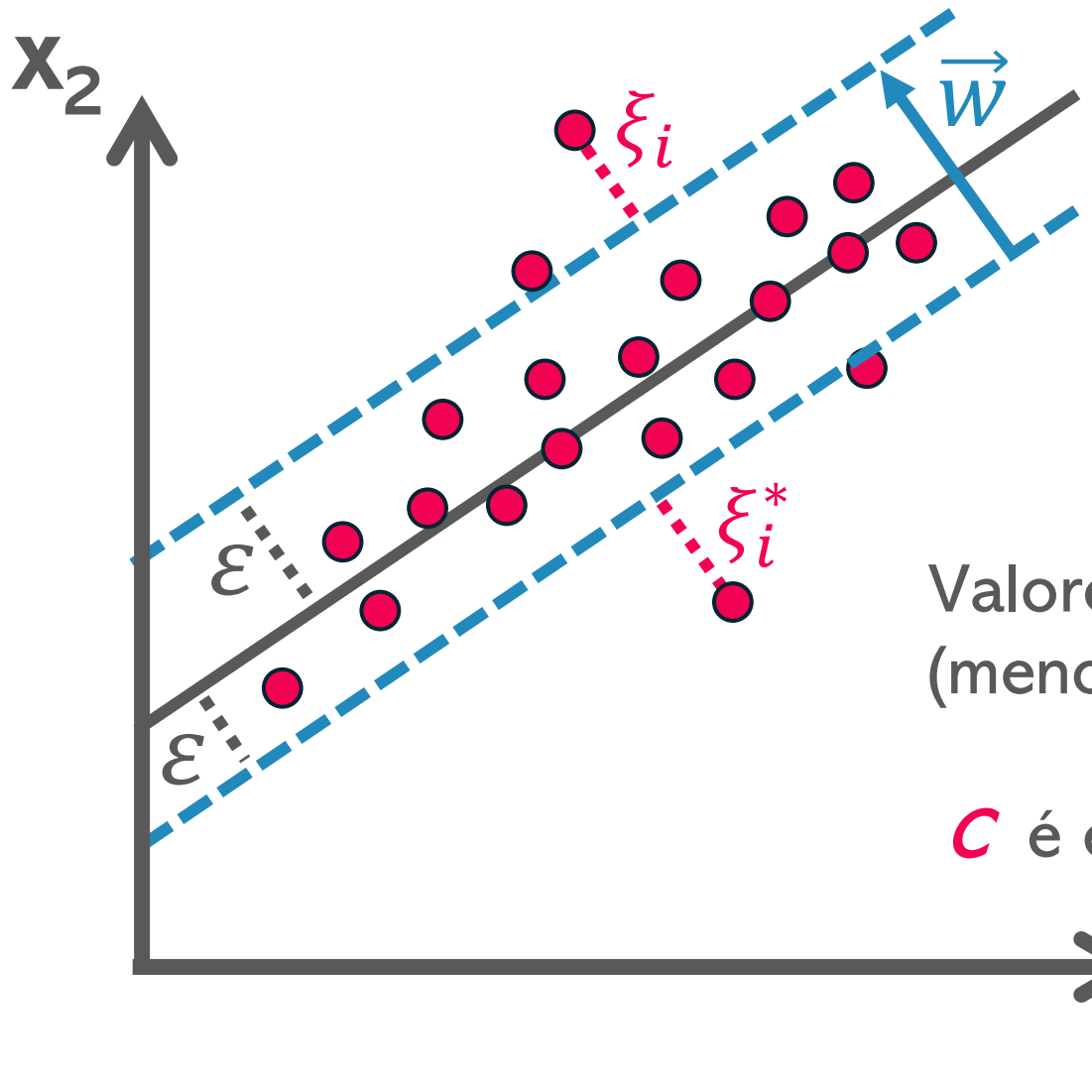
$$y_i - f(r_i) \leq \varepsilon + \xi_i$$
$$y_i - f(r_i) \geq \varepsilon + \xi_i^*$$
$$\xi_i^*\,\xi_i \geq 0 \forall i$$

$$\hat{y}_i = \sum_{j}^{n}(\alpha_j - \alpha_j^*)\,K(\vec{r}_i,\vec{r}_j) + b$$

Perceba que no SVR o parâmetro C controla a penalidade para os pontos que ultrapassam o limite.

Valores altos de C dão menos tolerância a erros (menor margem, mais acurácia nos dados de treinamento: controla o overfitting)

SVM aplicado a regressão - SVR



$$Min_{\|\vec{w}\|,b,\xi_i^*,\xi_i}\left(\frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{m}\xi_i^* + \xi_i\right)$$

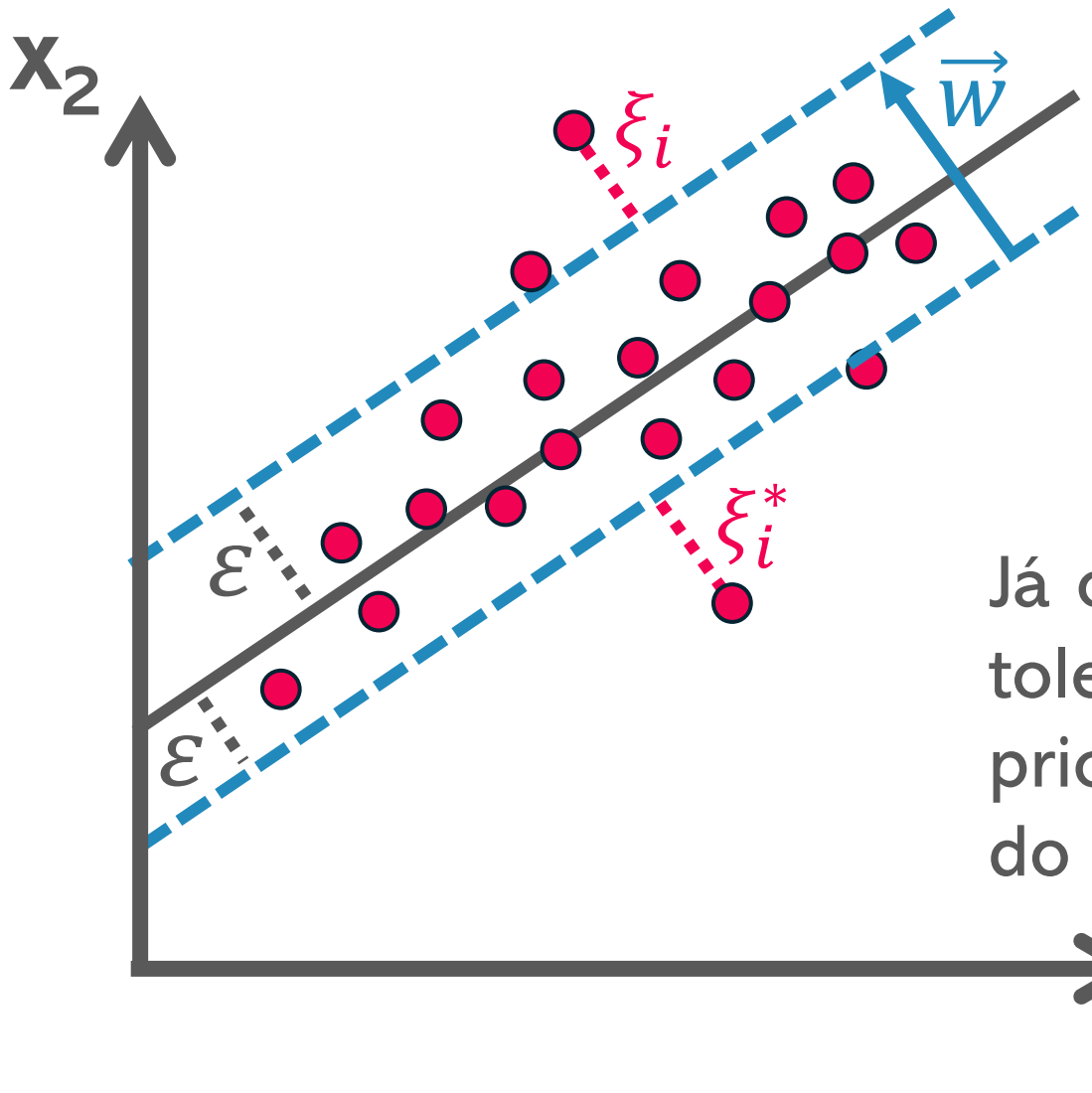
$$y_i - f(r_i) \leq \varepsilon + \xi_i$$
$$y_i - f(r_i) \geq \varepsilon + \xi_i^*$$
$$\xi_i^*\,\xi_i \geq 0 \forall i$$

$$\hat{y}_i = \sum_{j}^{n}(\alpha_j - \alpha_j^*)\,K(\vec{r}_i,\vec{r}_j) + b$$

Valores baixos de C permitem uma margem maior (menos penalização: controla o *underfitting*).

C é escolhido a priori (**cross-validation**)

SVM aplicado a regressão - SVR



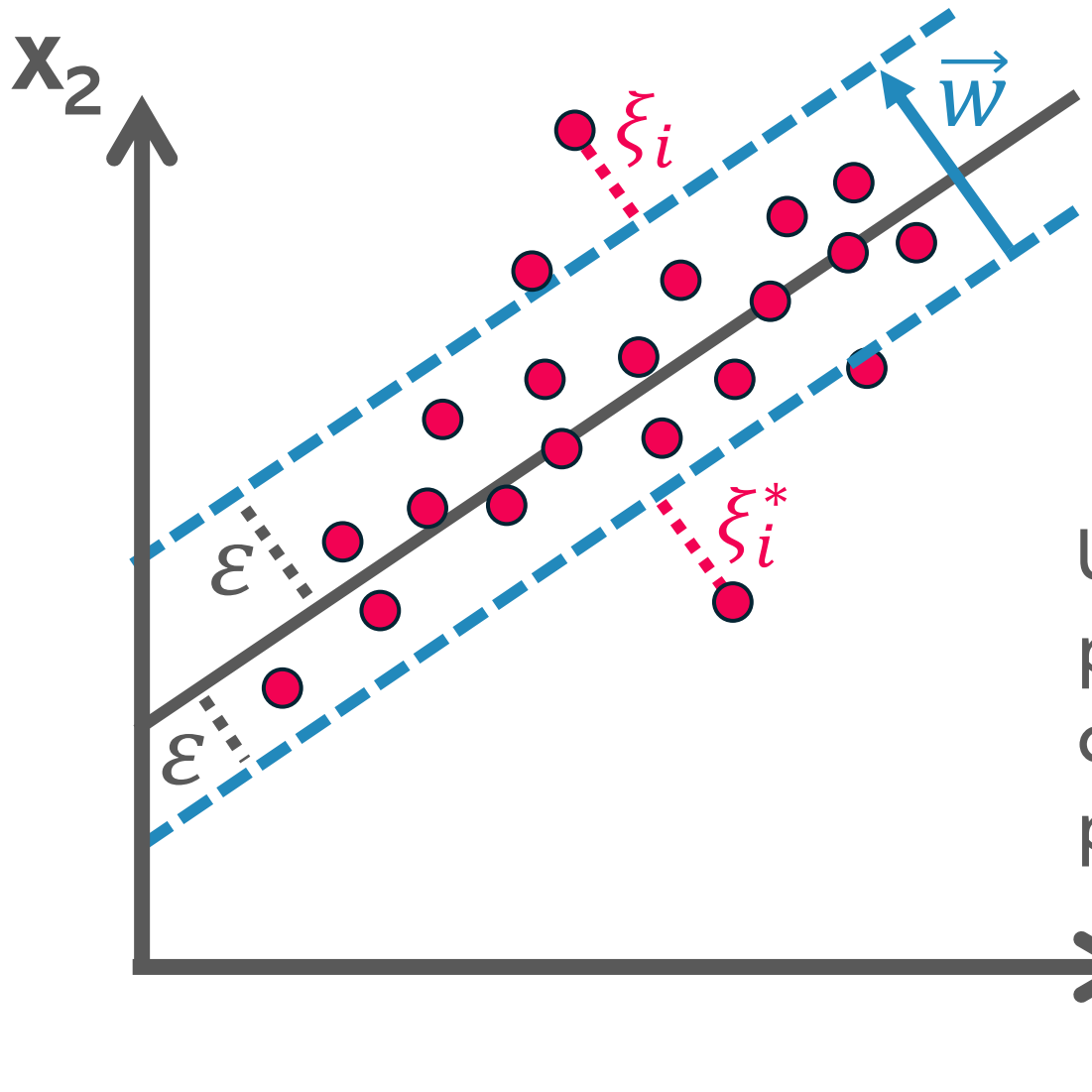
$$Min_{\|\vec{w}\|,b,\xi_i^*,\xi_i}\left(\frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{m}\xi_i^* + \xi_i\right)$$

$$y_i - f(r_i) \leq \varepsilon + \xi_i$$
$$y_i - f(r_i) \geq \varepsilon + \xi_i^*$$
$$\xi_i^*\,\xi_i \geq 0 \forall i$$

$$\hat{y}_i = \sum_{j}^{n}(\alpha_j - \alpha_j^*)\,K(\vec{r}_i,\vec{r}_j) + b$$

Já o ε, que determina a largura do tubo de tolerância sem penalização, é **escolhido** a priori (**utilizamos cross-validation**), diferente do caso de classificação

SVM aplicado a regressão - SVR



$$Min_{\|\vec{w}\|,b,\xi_i^*,\xi_i}\left(\frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{m}\xi_i^* + \xi_i\right)$$

$$y_i - f(r_i) \leq \varepsilon + \xi_i$$
$$y_i - f(r_i) \geq \varepsilon + \xi_i^*$$
$$\xi_i^*\,\xi_i \geq 0 \forall i$$

$$\hat{y}_i = \sum_{j}^{n}\left(\alpha_j - \alpha_j^*\right) K(\vec{r}_i, \vec{r}_j) + b$$

Um ε maior permite mais erro sem penalização, o que pode reduzir a complexidade do modelo, mas também pode levar a um ajuste menos preciso

Extraindo explicações

SVM como caixa preta

Diferente dos modelos que vimos até o momento, o SVM exibe um caráter de "caixa preta". Muito se deve ao uso das funções de Kernel. Não é uma tarefa trivial extrair explicações sobre seus resultados.

Entretanto, existem algumas tentativas de torná-lo mais transparente

ANALYTICA CHIMICA ACTA

www.elsevier.com/locate/aca

Visualisation and interpretation of Support Vector Regression models

B. Üstün, W.J. Melssen, L.M.C. Buydens *

Institute for Molecules and Materials, Analytical Chemistry, Radboud University of Nijmegen, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands

Received 9 October 2006; received in revised form 1 March 2007; accepted 2 March 2007
Available online 18 March 2007

Link: doi:10.1016/j.aca.2007.03.023

Üstün *et al.* 2007

Seção 2.2.2 - Interpretation of SVR models

$$\hat{y}_i = \sum_j^n \left(\alpha_j - \alpha_j^*\right) K(\vec{\boldsymbol{r}}_i, \vec{\boldsymbol{r}}_j) + b \qquad 0 < \alpha_i^*, \alpha_i \leq C \;\; \forall\, i = 1,2,\ldots,n_{vetsup}$$

$$\alpha_i^{[dual]} = \alpha_i - \alpha_i^*$$
$$i = 1,2,\ldots,n_{vetsup}$$

Quanto maior o coeficiente dual, maior a influência daquela amostra na construção do hiperplano do SVR, e portanto, maior a influência na modelagem

$$\boldsymbol{X}_{[vetsup]} = [\boldsymbol{X}]_{n_{vetsup} \times var}$$

$$\boldsymbol{p} = \boldsymbol{X}_{[vetsup]}^t \boldsymbol{\alpha}^{[dual]}$$

$$[\boldsymbol{X}]_{var \times n_{vetsup}} [\boldsymbol{\alpha}]_{n_{vetsup} \times 1} = [\boldsymbol{p}]_{var \times 1}$$

Üstün *et al.* 2007

Seção 2.2.2 - Interpretation of SVR models

$$\boldsymbol{p} = \boldsymbol{X}^{t}_{[vetsup]}\boldsymbol{\alpha}^{[dual]} \qquad p_j = \sum_{i=1}^{n_{vetsup}} \alpha_i^{[dual]} X_{ij} \qquad j = 1, 2, .., variáveis$$

Ou seja, cada componente do p_j é a soma de todos os valores da variável j das amostras vetores de suporte ponderada pelos respectivos $\alpha_i^{[dual]}$. Logo, o vetor $\boldsymbol{p}$ pode ser utilizado para indicar o quanto cada variável contribui para o ajuste do hiperplano ou para a formação/criação do modelo (ideia de importância), pois os maiores valores serão devidos aos $\alpha_i^{[dual]}$ grandes.

Üstün *et al.* 2007

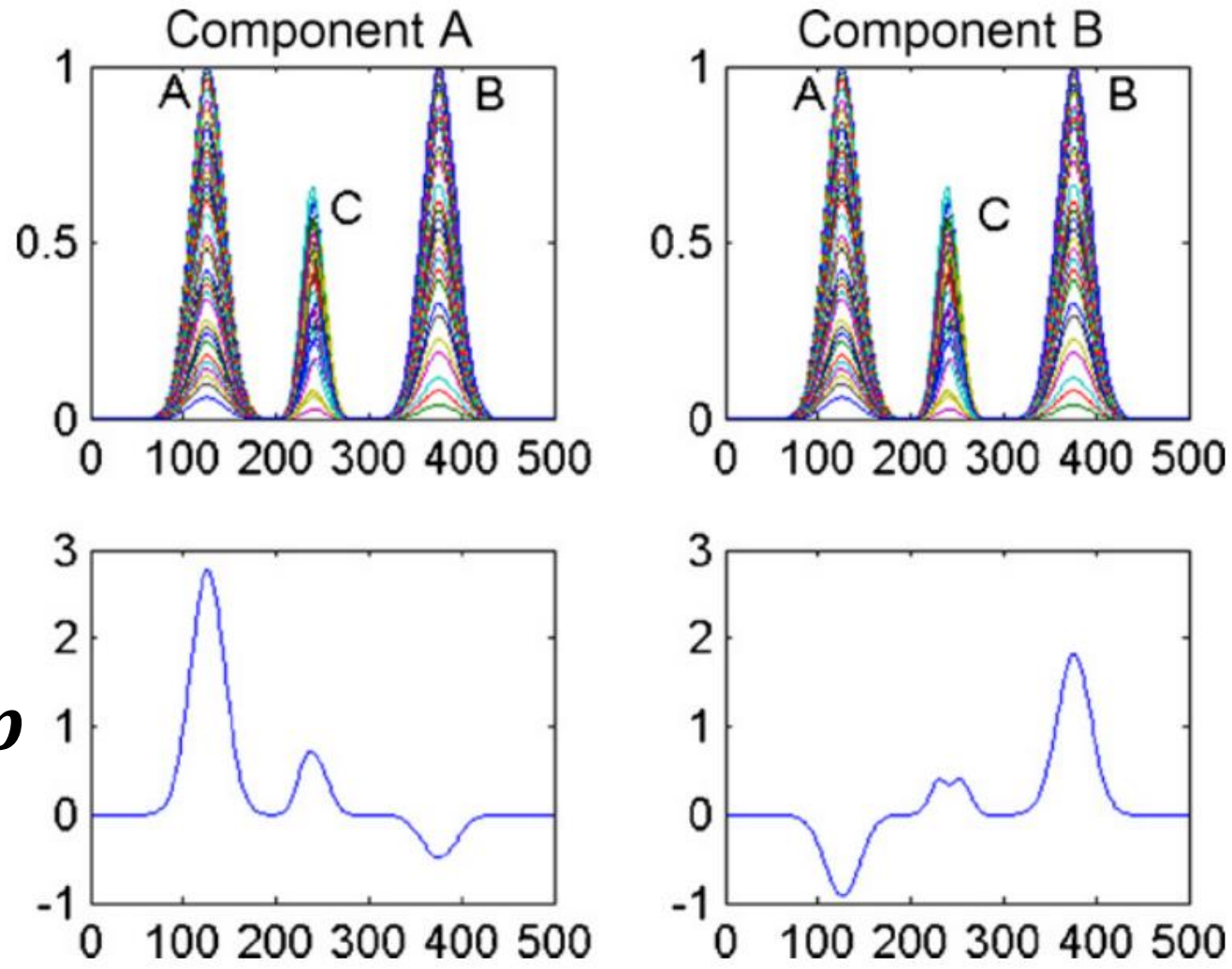
Seção 2.2.2 - Interpretation of SVR models

$$\boldsymbol{p} = \boldsymbol{X}^{t}_{[vetsup]}\boldsymbol{\alpha}^{[dual]} \qquad p_j = \sum_{i=1}^{n_{vetsup}} \alpha_i^{[dual]} X_{ij} \qquad j = 1, 2, .., variáveis$$

A ideia é similar o loadings do PLS, que mostra como cada variável original pesa nos nas variáveis latentes. Aqui o pvetor exibe as amostras vetor de suporte que mais contribuem para o hiperplano ótimo, e portanto, os padrões que regem as variáveis originais associadas a elas são importantes para o modelo

Üstün *et al.* 2007

Seção 2.2.2 - Interpretation of SVR models



$$p_j = \sum_{i=1}^{n_{vetsup}} X_{ij}\, \alpha_i^{[dual]}$$

$$j = 1, 2, .., variáveis$$

Fig. 7. Os dois gráficos superiores representam os espectros originais do problema simulado. A parte inferior representa o gráfico de linhas do vetor p para os modelos SVR para quantificar os componentes A e B.

SVM como caixa preta

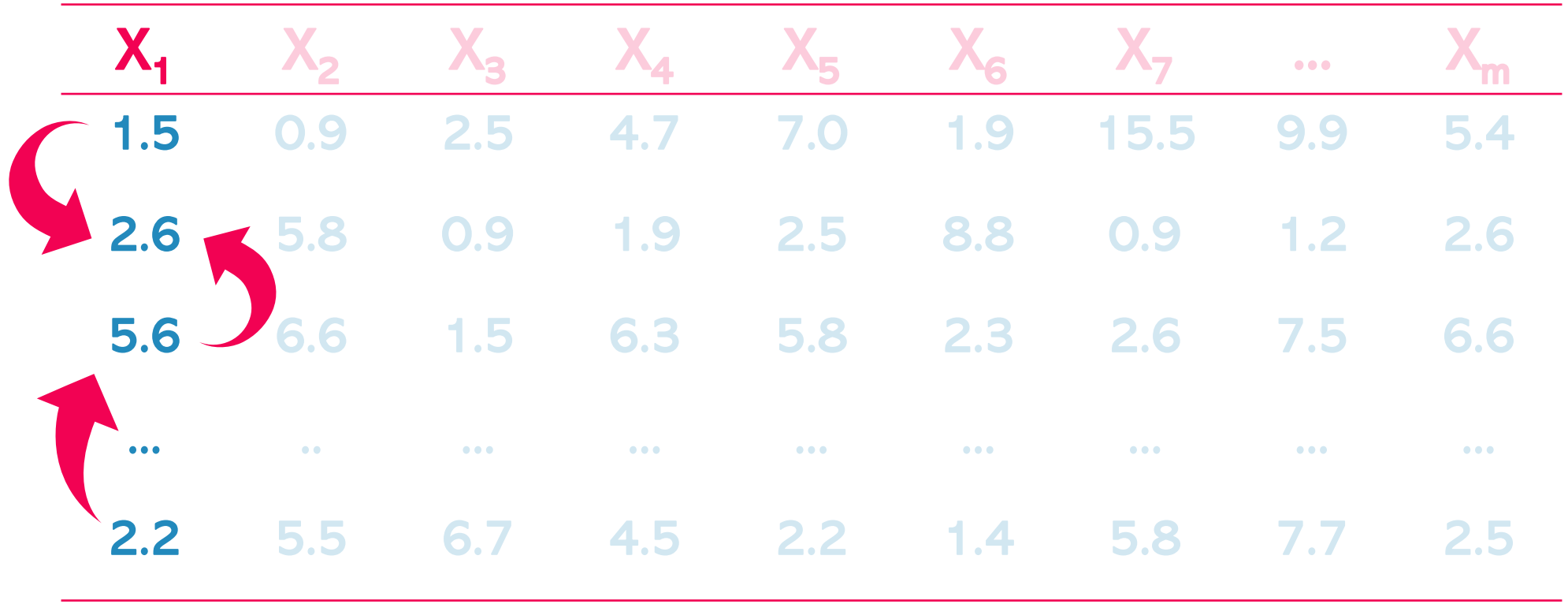
Por outro lado, embora estudos abordando teoria sobre os mecanismos internos sobre o SVM estejam sendo feitos, outros algoritmos criados para lidar com modelos "caixa preta" também têm sido propostos

Eles são mais amplamente abordados no campo da "eXplainable AI (XAI)". No geral, três instâncias são importantes:

- Agnosticismo: Algoritmos agnósticos vs. Específicos
- Timing: Ad-Hoc vs. Post-Hoc
- Escopo: Global vs. Local

Variáveis importantes via Permutação

A permutação é uma técnica agnóstica que mede o impacto de cada variável na performance do modelo treinado (Post-hoc) e fornece explicações globais. Uma métrica do modelo geral é extraída e então permuta-se a ordem das amostras de acordo com cada variáveis individualmente como forma de perturbação, então a mesma métrica é extraída e comparada com a original



X_1	X_2	X_3	X_4	X_5	X_6	X_7	...	X_m
1.5	0.9	2.5	4.7	7.0	1.9	15.5	9.9	5.4
2.6	5.8	0.9	1.9	2.5	8.8	0.9	1.2	2.6
5.6	6.6	1.5	6.3	5.8	2.3	2.6	7.5	6.6
...	..	...	...	...	...	...	...	...
2.2	5.5	6.7	4.5	2.2	1.4	5.8	7.7	2.5

Variáveis importantes via Permutação

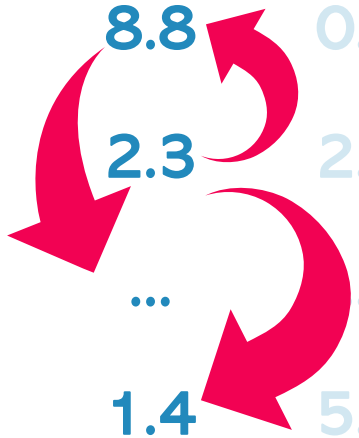
A permutação é uma técnica agnóstica que mede o impacto de cada variável na performance do modelo treinado (Post-hoc) e fornece explicações globais. Uma métrica do modelo geral é extraída e então permuta-se a ordem das amostras de acordo com cada variáveis individualmente como forma de perturbação, então a mesma métrica é extraída e comparada com a original

X_1	X_2	X_3	X_4	X_5	X_6	X_7	...	X_m
1.5	0.9	2.5	4.7	7.0	1.9	15.5	9.9	5.4
2.6	5.8	0.9	1.9	2.5	8.8	0.9	1.2	2.6
5.6	6.6	1.5	6.3	5.8	2.3	2.6	7.5	6.6
...	..	..	...	...	...	...	...	...
2.2	5.5	6.7	4.5	2.2	1.4	5.8	7.7	2.5

Variáveis importantes via Permutação

A permutação é uma técnica agnóstica que mede o impacto de cada variável na performance do modelo treinado (Post-hoc) e fornece explicações globais. Uma métrica do modelo geral é extraída e então permuta-se a ordem das amostras de acordo com cada variáveis individualmente como forma de perturbação, então a mesma métrica é extraída e comparada com a original

X_1	X_2	X_3	X_4	X_5	X_6	X_7	...	X_m
1.5	0.9	2.5	4.7	7.0	1.9	15.5	9.9	5.4
2.6	5.8	0.9	1.9	2.5	8.8	0.9	1.2	2.6
5.6	6.6	1.5	6.3	5.8	2.3	2.6	7.5	6.6
...	..	...	...	...	...	..	...	...
2.2	5.5	6.7	4.5	2.2	1.4	5.8	7.7	2.5



Variáveis importantes via Permutação

A permutação é uma técnica agnóstica que mede o impacto de cada variável na performance do modelo treinado (Post-hoc) e fornece explicações globais. Uma métrica do modelo geral é extraída e então permuta-se a ordem das amostras de acordo com cada variáveis individualmente como forma de perturbação, então a mesma métrica é extraída e comparada com a original

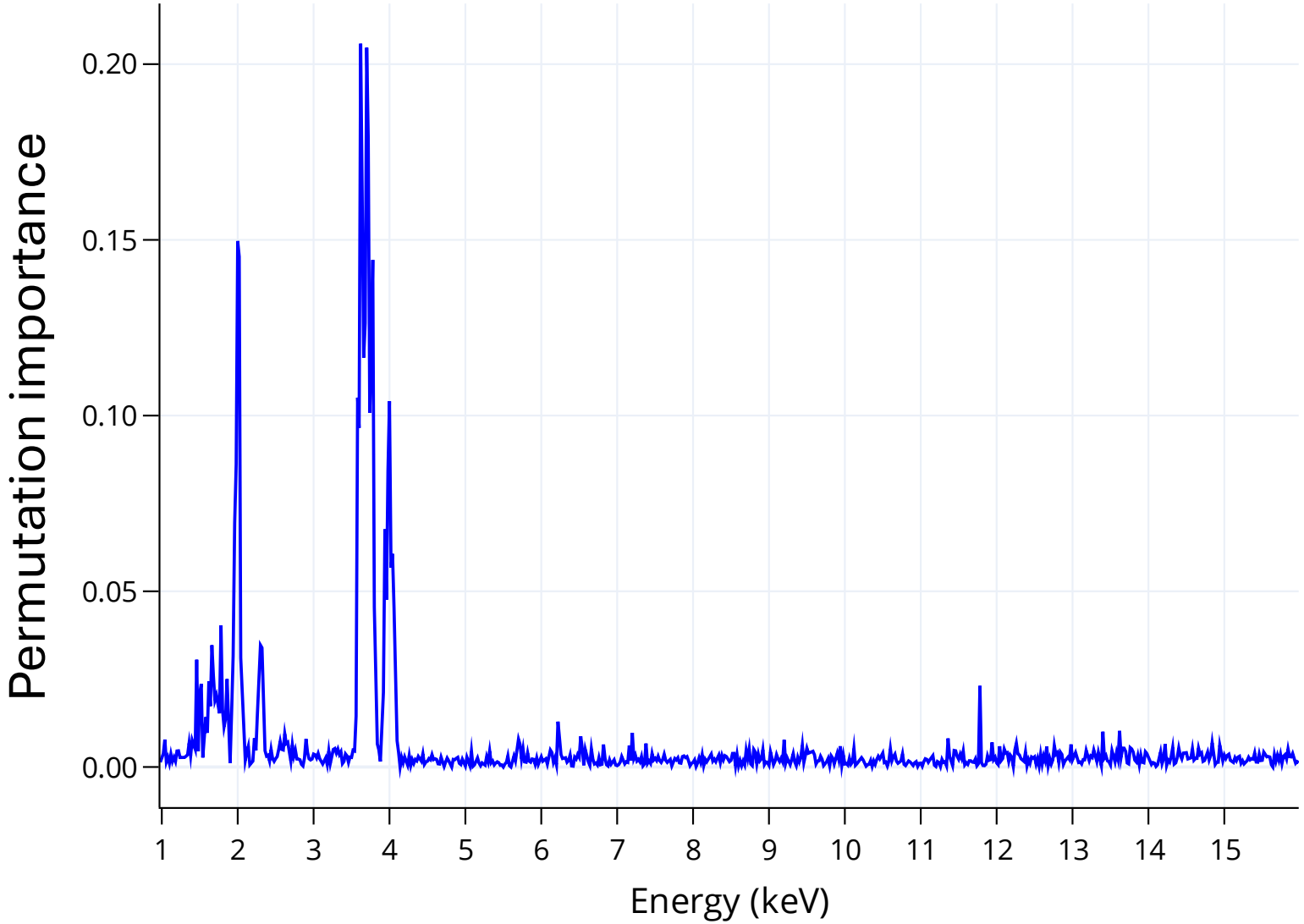
X_1	X_2	X_3	X_4	X_5	X_6	X_7	...	X_m
1.5	0.9	2.5	4.7	7.0	1.9	15.5	9.9	5.4
2.6	5.8	0.9	1.9	2.5	8.8	0.9	1	2.6
5.6	6.6	1.5	6.3	5.8	2.3	2.6	7.5	6.6
...	..	...	...	...	...	...	...	...
2.2	5.5	6.7	4.5	2.2	1.4	5.8	7.7	2.5

Variáveis importantes via Permutação

O processo é repetido k-vezes para reduzir o efeito do acaso. Então o erro médio dos modelos em relação a cada variável permutada é obtido e comparado com o original para se obter uma estimativa da importância das variáveis

X_1	X_2	X_3	X_4	X_5	X_6	X_7	...	X_m
1.5	0.9	2.5	4.7	7.0	1.9	15.5	9.9	5.4
2.6	5.8	0.9	1.9	2.5	8.8	0.9	1	2.6
5.6	6.6	1.5	6.3	5.8	2.3	2.6	7.5	6.6
...	..	...	...	...	...	...	...	...
2.2	5.5	6.7	4.5	2.2	1.4	5.8	7.7	2.5

Variáveis importantes via Permutação



Prática no VS Code

Üstün *et al.* 2007

Seção 2.2.1 - Visualisation of the information in the kernel matrix

$$K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j) = K_{ij}$$

Gera uma matriz quadrada $(n \times n)$ que engloba o grau de similaridade entre $\vec{x}_i$ e $\vec{x}_j$ no espaço de alta dimensão

A ideia está no produto interno $\phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$, que aumenta quando $\phi(\vec{x}_i)$ e $\phi(\vec{x}_j)$ têm direções semelhantes e diminui quando eles são diferentes, caracterizando intuitivamente uma medida de similaridade

Portanto, ao visualizar K entendemos diretamente quais pares de amostras o modelo considera próximos ou distantes **no espaço de kernel**, fornecendo uma intuição sobre a geometria interna do SVR.

Üstün *et al.* 2007

Seção 2.2.1 - Visualisation of the information in the kernel matrix

A estratégia é correlacionar cada linha de K (onde estão os padrões de similaridades das amostras no espaço de kernel) com cada variável original para gerar uma informação sobre quais as variáveis mais contribuíram para gerar os padrões similaridade entre as amostras no espaço de Kernel (utilizadas diretamente para o cálculo do SVR)

$$R_{ij} = np.corrcoef(K[i,:], X[:,j])[0,1] \qquad \begin{matrix} i = 1,2,\ldots n\,(amostras) \\ j = 1,2,\ldots m\,(variáveis) \end{matrix}$$

Isso vinculo os padrões mais importantes para o modelo SVR (K) com as variáveis originais

Üstün *et al.* 2007

Seção 2.2.1 - Visualisation of the information in the kernel matrix

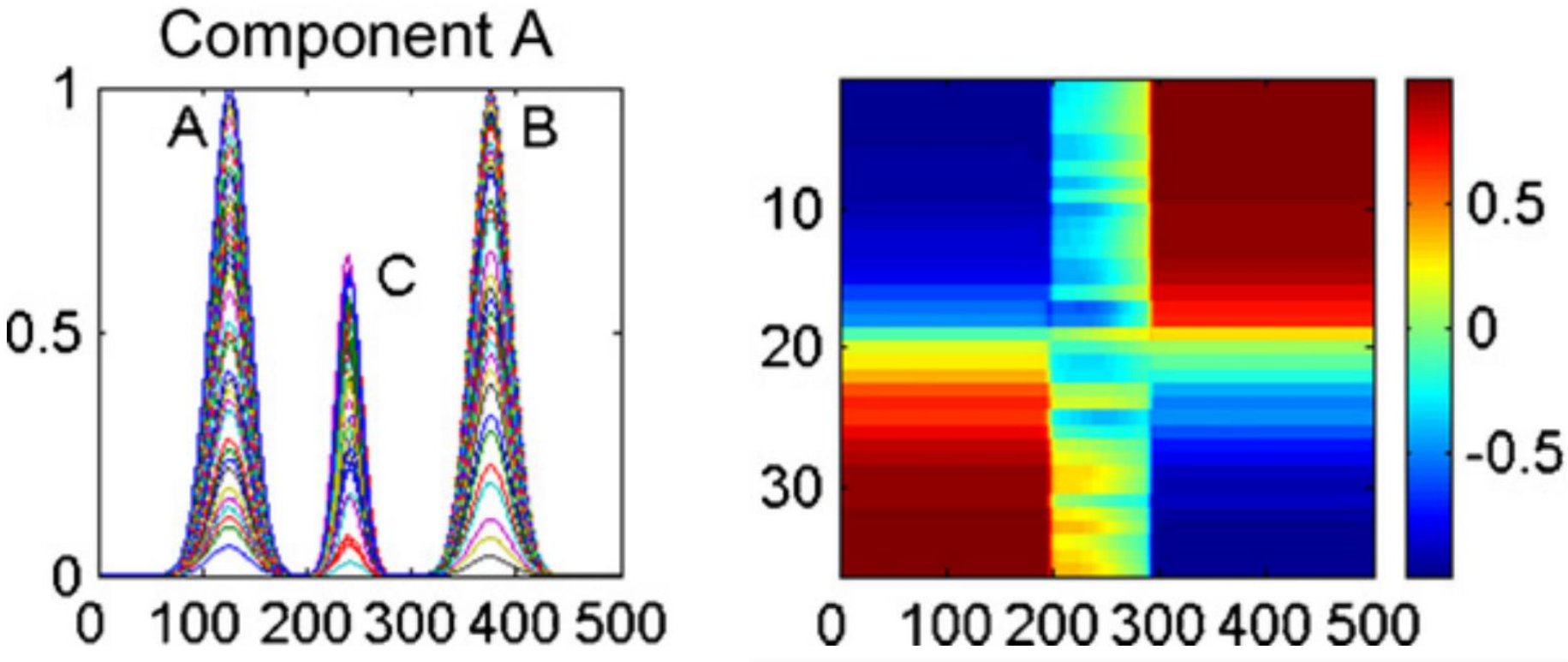
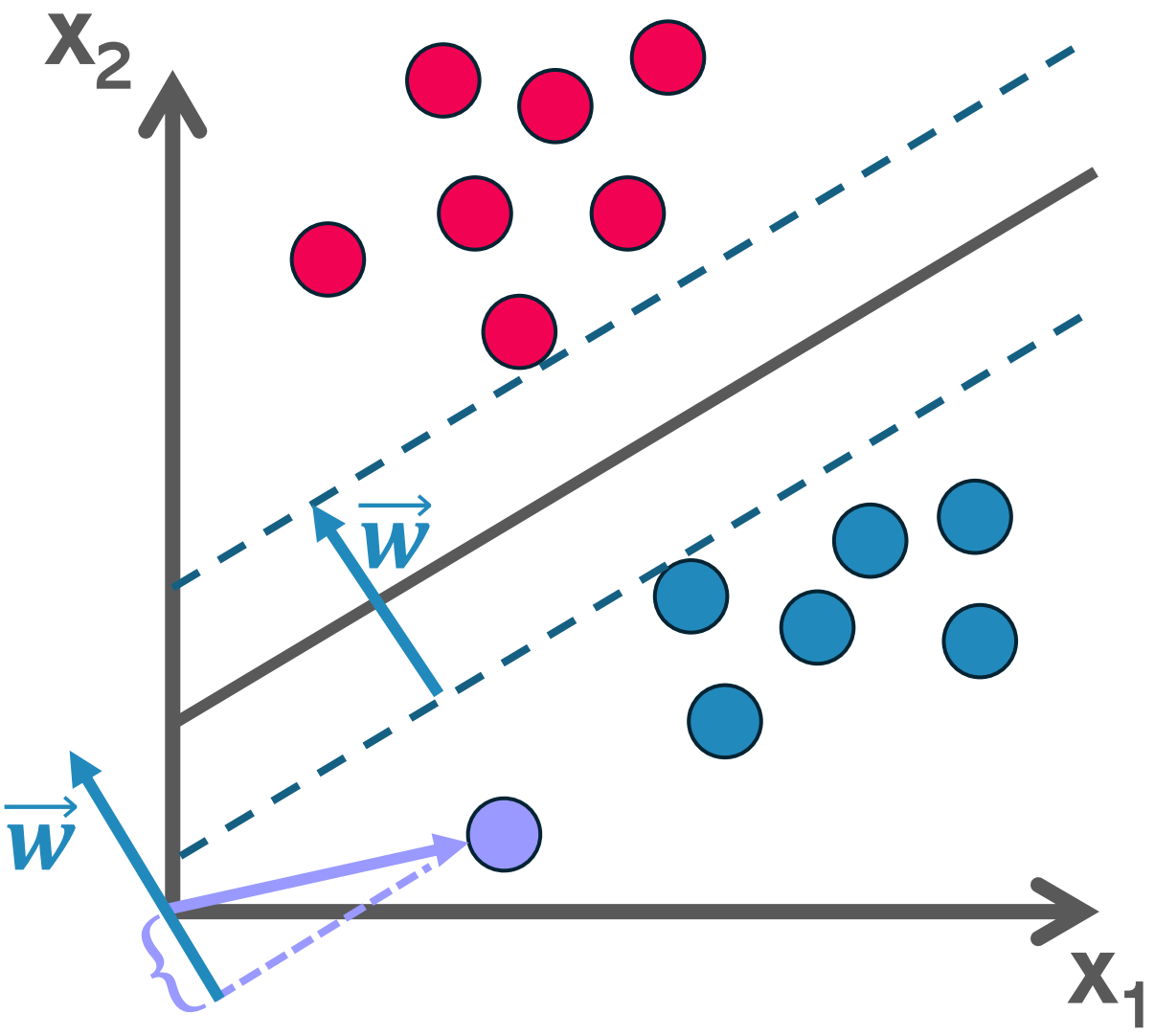


Fig. 5. O gráfica a esquerda representa os espectros originais. A imagem da direita é a Imagem de Correlação (IC), onde o eixo y representa as amostras distribuídas por concentração (de baixa para alta concentração) e o eixo x representa as variáveis de entrada.

SVM e G. A.



Se o vetor $\vec{r} = x_{10}\hat{x}_1 + x_{20}\hat{x}_2$ localiza o novo ponto:

$$w_{x_1}x_{10} + w_{x_2}x_{20} + b =$$

$$= \vec{w} \cdot \vec{r} + b$$

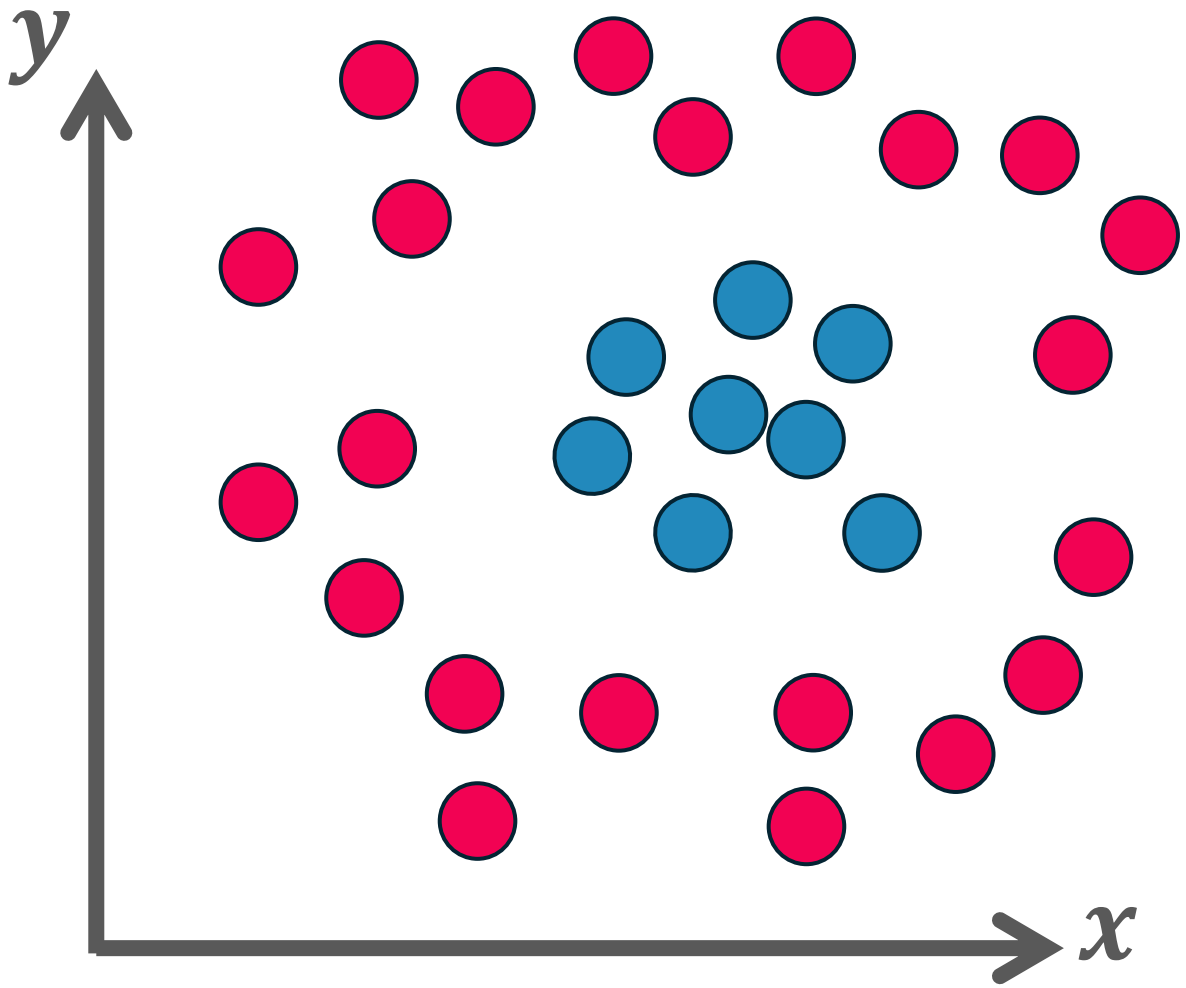
$$= \| \vec{r} \|^2 \| \vec{w} \|^2 + b$$

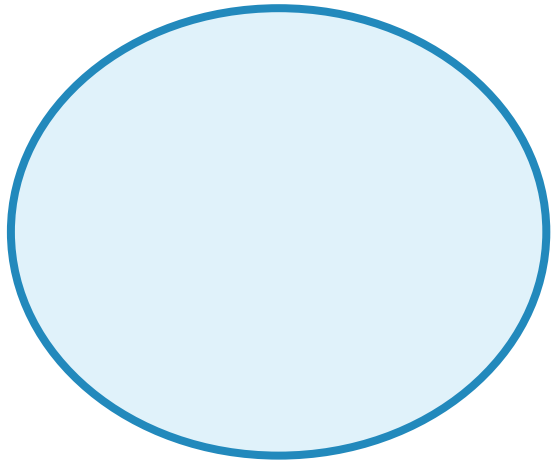
> 0 Vermelha

< 0 Azul

$= 0$ Indefinido

Truque de kernel





Truque de kernel