

MAE 5905: Introdução à Ciência de Dados

Pedro A. Morettin

Instituto de Matemática e Estatística
Universidade de São Paulo
pam@ime.usp.br
<http://www.ime.usp.br/~pam>

Aula 10

8 de maio de 2023

Sumário

- 1 Classificador de Margem Não Linear
- 2 Noções da Teoria
- 3 Regressão por SVM

CMNL

- Na seção anterior apresentamos um algoritmo de classificação (CMF), usado quando as fronteiras são lineares. Para fronteiras não lineares, precisamos aumentar a dimensão do espaço de dados por meio de outras funções, polinomiais ou não, para determinar as fronteiras de separação.
- Pode-se demonstrar que um classificador linear como aquele definido anteriormente (CMF) depende somente dos vetores suporte e pode ser escrito na forma

$$f(\mathbf{x}) = \sum_{i \in S} \gamma_i \langle \mathbf{x}, \mathbf{x}_i \rangle + \delta, \quad (1)$$

em que S indica o conjunto dos vetores suporte, os γ_i são funções de α e β e $\langle \mathbf{x}, \mathbf{y} \rangle$ indica o produto interno dos vetores \mathbf{x} e \mathbf{y} .

- Uma das vantagens de se utilizar **kernels** na construção de classificadores é que eles dependem somente dos vetores suporte e não de todas as observações o que implica uma redução considerável no custo computacional.

CMNL

- O classificador CMF usa um *kernel* linear, da forma

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^p x_{ik} x_{jk} = \mathbf{x}_i^\top \mathbf{x}_j.$$

- Se quisermos usar um CMF em um espaço característico de dimensão maior, podemos incluir polinômios de grau maior ou mesmo outras funções na definição do classificador.
- Os *kernels* mais utilizados na prática são:
 - a) lineares: $K(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1^\top \mathbf{x}_2$;
 - b) polinomiais: $K(\mathbf{x}_1, \mathbf{x}_2) = (a + \mathbf{x}_1^\top \mathbf{x}_2)^d$;
 - c) radiais: $K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\gamma \|\mathbf{x}_1 - \mathbf{x}_2\|^2)$, com $\gamma > 0$ constante.
 - d) tangentes hiperbólicas: $K(\mathbf{x}_1, \mathbf{x}_2) = \tanh(\theta + \mathbf{x}_1^\top \mathbf{x}_2)$.

CMNL

Os **classificadores CMNL** são obtidos combinando-se CMF com **kernels** não lineares, de modo a obter

$$f(\mathbf{x}) = \alpha + \sum_{i \in S} \gamma_i K(\mathbf{x}, \mathbf{x}_i) + \delta. \quad (2)$$

em que os γ_i são funções de α e β .

Exemplo CMNL

- **Exemplo.** Consideremos uma análise alternativa para dados do exemplo anterior, utilizando um *kernel* polinomial, de grau 3.
- Os comandos e resultados da reanálise dos dados por meio do classificador de margem não linear são:

```
escolhaparam <- tune.svm(grupo ~ altfac + proffac, data = face,  
                        kernel = "polynomial", degree=3,  
                        gamma = 2^(-1:2), cost = 2^2:6)  
  
> summary(escolhaparam)
```

Parameter tuning of svm:

- sampling method: 10-fold cross validation
- best parameters:
degree gamma cost
3 0.5 4
- best performance: 0.1681818

Exemplo CMNL

- Detailed performance results:

	degree	gamma	cost	error	dispersion
1	3	0.5	4	0.1681818	0.09440257
2	3	1.0	4	0.1772727	0.12024233
3	3	2.0	4	0.1872727	0.11722221
4	3	4.0	4	0.1872727	0.11722221
5	3	0.5	5	0.1972727	0.11314439
6	3	1.0	5	0.1772727	0.12024233
7	3	2.0	5	0.1872727	0.11722221
8	3	4.0	5	0.1872727	0.11722221
9	3	0.5	6	0.1872727	0.12634583
10	3	1.0	6	0.1772727	0.12024233
11	3	2.0	6	0.1872727	0.11722221
12	3	4.0	6	0.1872727	0.11722221

Exemplo CMNL

```
svm.model <- svm(grupo ~ altfac + proffac, data=face,
                  type='C-classification', kernel='polynomial',
                  degree=3, gamma=1, cost=4, coef0=1, scale=FALSE)
summary(svm.model)
```

Parameters:

SVM-Type: C-classification

SVM-Kernel: polynomial

cost: 4

degree: 3

coef.0: 1

Number of Support Vectors: 40

(11 10 19)Number of Classes: 3

Levels:

braq dolico meso

A tabela de classificação é

	true		
pred	braq	dolico	meso
braq	29	0	4
dolico	0	26	3
meso	4	5	30

acurácia=0,84

Exemplo CMNL

O gráfico correspondente está apresentado na Figura 1.

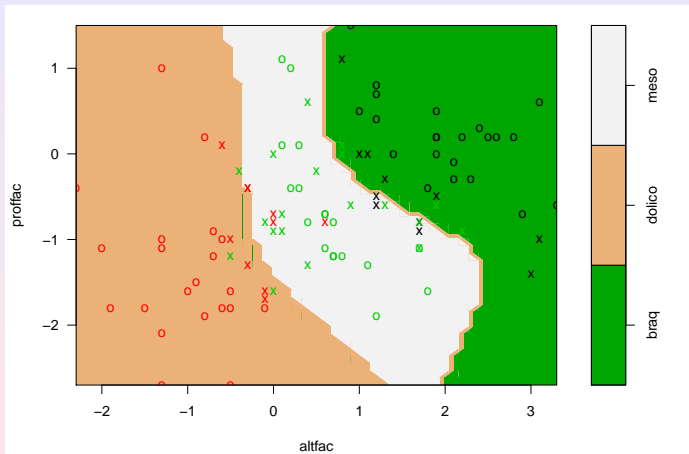


Figura: Classificação do tipo facial obtida pelo classificador de margem não linear.

Exemplo CMNL

- Neste caso, o número de classificações erradas (16) é igual ao caso do classificador de margem flexível. A TEC é 0,16.
- Com base nesses resultados, podemos classificar indivíduos para os quais dispomos apenas dos valores das variáveis preditoras. Com essa finalidade, consideremos o seguinte conjunto de previsão com 4 indivíduos:

	paciente	altfac	proffac
1	102	1.4	1.0
2	103	3.2	0.1
3	104	-2.9	-1.0
4	105	0.5	0.9

CMNL

Por meio dos seguintes comandos

```
svm.model <- svm(grupo ~ altfac + proffac, data=face, type='C-classification',
                 kernel='polynomial', degree=3, gamma=1, cost=4, coef0=1,
                 scale=FALSE, probability=TRUE)
prednovos <- predict(svm.model, teste, probability=TRUE)
```

obtemos a tabela com as probabilidades de classificação de cada um dos 4 indivíduos

```
      1      2      3      4
braq  braq  dolico  meso
attr(,"probabilities")

      braq      dolico      meso
1 0.954231749 0.0193863931 0.0263818582
2 0.961362058 0.0006154201 0.0380225221
3 0.008257919 0.9910764215 0.0006656599
4 0.254247666 0.1197179567 0.6260343773
```

Levels: braq dolico meso

O processo classifica os indivíduos 102 e 103 como braquicéfalos, o indivíduo 104 como dolicocéfalo e o 105, como mesocéfalo.

Hiperplano separador

Um hiperplano definido num espaço de dimensão p é um **subespaço** de dimensão $p - 1$ definido por

$$\alpha + \beta_1 X_1 + \dots + \beta_p X_p = 0. \quad (3)$$

Um ponto com coordenadas (x_1, \dots, x_p) satisfazendo (3) situa-se no hiperplano. Se $\alpha + \beta_1 x_1 + \dots + \beta_p x_p > 0$, esse ponto situa-se num lado do hiperplano e se $\alpha + \beta_1 x_1 + \dots + \beta_p x_p < 0$, o ponto situa-se no outro lado desse hiperplano. Dessa forma, o hiperplano separa o espaço p dimensional em duas metades.

Teoria-CMM

- Consideremos o espaço característico $\mathcal{T} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ e as respostas y_1, \dots, y_n com $y_i \in \{-1, 1\}$, definindo o conjunto de treinamento. Novos dados \mathbf{x}_0 são classificados de acordo com o sinal de $f(\mathbf{x}_0)$.
- Suponha que exista um hiperplano separador, de modo que α e β são tais que $f(\mathbf{x}) > 0$, para pontos com $y = +1$ e $f(\mathbf{x}) < 0$, para pontos com $y = -1$, de modo que $yf(\mathbf{x}) > 0$, para qualquer dado de treinamento.
- O CMM tem como objetivo maximizar a margem que é a menor distância entre o hiperplano e qualquer ponto do conjunto de treinamento.
- Para entender o procedimento de otimização, considere a distância de um ponto \mathbf{x} ao hiperplano cuja equação é $f(\mathbf{x}) = 0$, nomeadamente

$$d = |f(\mathbf{x})| / \|\beta\|,$$

em que denominador indica a norma do vetor β .

Teoria–CMM

- Como o interesse está nos pontos que são corretamente classificados, devemos ter $y_i f(\mathbf{x}_i) > 0$, $i = 1, \dots, n$. Logo, a distância entre qualquer ponto \mathbf{x}_i e o hiperplano é

$$\frac{y_i f(\mathbf{x}_i)}{\|\boldsymbol{\beta}\|} = \frac{y_i(\alpha + \boldsymbol{\beta}^\top \mathbf{x}_i)}{\|\boldsymbol{\beta}\|}. \quad (4)$$

- A margem é a distância do hiperplano ao ponto \mathbf{x} mais próximo e queremos escolher α e $\boldsymbol{\beta}$ de modo a maximizar essa distância. A margem máxima é obtida por meio da resolução de

$$\operatorname{argmax}_{\alpha, \boldsymbol{\beta}} \left\{ \frac{1}{\|\boldsymbol{\beta}\|} \min \left[y_i(\alpha + \boldsymbol{\beta}^\top \mathbf{x}_i) \right] \right\}. \quad (5)$$

- A solução de (5) é complicada mas é possível obtê-la por meio da utilização de **Multiplicadores de Lagrange**. Note que se multiplicarmos α e $\boldsymbol{\beta}$ por uma constante, a distância de um ponto \mathbf{x} ao hiperplano separador não se altera.

Teora–CMM

- Logo podemos considerar a transformação $\alpha^* = \alpha/f(\mathbf{x})$ e $\beta^* = \beta/f(\mathbf{x})$ e para o ponto mais próximo do hiperplano, digamos \mathbf{x}^* , obtendo

$$y^*(\alpha + \beta^\top \mathbf{x}^*) = 1, \quad (6)$$

e consequentemente, $d = \|\beta\|^{-1}$.

- Desse modo, todos os pontos do conjunto de treinamento satisfarão

$$y_i(\alpha + \beta^\top \mathbf{x}_i) \geq 1, \quad i = 1, \dots, n. \quad (7)$$

Esta relação é chamada **representação canônica do hiperplano separador**.

- Dizemos que há uma **restrição ativa** para os pontos em que há igualdade; para os pontos em que vale a desigualdade, dizemos que há uma **restrição inativa**. Como sempre haverá um ponto que está mais próximo do hiperplano, sempre haverá uma restrição ativa.

Teoria-CMM

- Então, o problema de otimização implica maximizar $\|\beta\|^{-1}$, que é equivalente a minimizar $\|\beta\|^2$.
- Na linguagem de Vapnik (1995), isso equivale a escolher $f(\mathbf{x})$ de maneira que seja a mais achatada (*flat*) possível, que por sua vez implica que β deve ser pequeno.
- Isso corresponde à resolução do problema de **programação quadrática**

$$\operatorname{argmin}_{\alpha, \beta} \left\{ \frac{1}{2} \|\beta\|^2 \right\}, \quad (8)$$

sujeito a (7). O fator $1/2$ é introduzido por conveniência.

- Com esse objetivo, para cada restrição em (7), introduzimos os Multiplicadores de Lagrange $\lambda_i \geq 0$, obtendo a função lagrangeana

$$L(\alpha, \beta, \lambda) = \frac{1}{2} - \sum_{i=1}^n \lambda_i [y_i(\alpha + \beta^\top \mathbf{x}_i) - 1], \quad (9)$$

em que $\lambda = (\lambda_1, \dots, \lambda_n)^\top$. O sinal negativo no segundo termo de (9) justifica-se por que queremos minimizar em relação a α e β e maximizar em relação a λ .

Teoria-CMM

- Derivando L em relação a β e a λ , obtemos

$$\beta = \sum_{i=1}^n \lambda_i y_i \mathbf{x}_i \quad \text{e} \quad \sum_{i=1}^n \lambda_i y_i = 0. \quad (10)$$

- Eliminando α e β em (9) e usando (10), obtemos a chamada **representação dual** do problema da margem máxima, no qual maximizamos

$$\tilde{L}(\lambda) = \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), \quad (11)$$

com respeito a λ , sujeito às restrições

$$\lambda_i \geq 0, \quad i = 1, \dots, n, \quad (12)$$

$$\sum_{i=1}^b \lambda_i y_i = 0. \quad (13)$$

- Em (11), $K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y}$ é um *kernel* linear, que será estendido para algum *kernel* mais geral com a finalidade de ser aplicado a espaços característicos cuja dimensionalidade excede o número de dados. Esse *kernel* deve ser positivo definido.

Teoria-CMM

- Para classificar um novo dado \mathbf{x}_0 usando o modelo treinado, avaliamos o sinal de $f(\mathbf{x}_0)$, que por meio de (10), pode ser escrito como

$$f(\mathbf{x}_0) = \alpha + \sum_{i=1}^n \lambda_i y_i K(\mathbf{x}_0, \mathbf{x}_i). \quad (14)$$

- Pode-se demonstrar (veja Bishop, 2006), que esse tipo de otimização restrita satisfaz certas condições, chamadas de **condições de Karush-Kuhn-Tucker** (KKT) que implicam

$$\begin{aligned} \lambda_i &\geq 0, \\ y_i f(\mathbf{x}_i) - 1 &\geq 0, \\ \lambda_i (y_i f(\mathbf{x}_i) - 1) &= 0. \end{aligned} \quad (15)$$

Teoria-CMM

- Para cada ponto, ou $\lambda_i = 0$ ou $y_i f(\mathbf{x}_i) = 1$. Um ponto para o qual $\lambda_i = 0$ não aparece em (14) não tem influência na classificação de novos pontos.
- Os pontos restantes são chamados **vetores suporte** e satisfazem $y_i f(\mathbf{x}_i) = 1$; logo esses pontos estão sobre as fronteiras do espaço separador, como na Figura 3 da Aula 9.
- O valor de α pode ser encontrado a partir de

$$y_i \left(\sum_{j \in S} \lambda_j y_j K(\mathbf{x}_j, \mathbf{x}_i) + \alpha \right) = 1, \quad (16)$$

em que S é o conjunto dos vetores suporte.

- Multiplicando essa expressão por y_i , observando que $y_i^2 = 1$ e tomando a média de todas as equações sobre S , obtemos

$$\alpha = \frac{1}{n_S} \sum_{i \in S} \left(y_i - \sum_{j \in S} \lambda_j y_j K(\mathbf{x}_i, \mathbf{x}_j) \right), \quad (17)$$

em que n_S é o número de vetores suporte.

Teoria-CMF

- Vamos considerar agora, o caso em que as duas classes podem se sobrepor. Precisamos modificar o CMM para permitir que alguns pontos do conjunto de treinamento sejam classificados erroneamente. Para isso introduzimos uma penalidade, que cresce com a distância ao hiperplano separador.
- Isso é conseguido pela introdução de **variáveis de folga** (*slack*) $\xi_i \geq 0, i = 1, \dots, n$, uma para cada dado.
- Então, $\xi_i = 0$ para pontos sobre ou dentro da fronteira correta [delimitada por $f(\mathbf{x}) = -1$ e $f(\mathbf{x}) = 1$] e ξ_i dado pela distância do ponto à fronteira, para os outros pontos.
- Assim, um ponto que estiver sobre o hiperplano $f(\mathbf{x}) = 0$ terá $\xi_i = 1$ e pontos com $\xi_i > 1$ são classificados erroneamente.
- Nesse caso, a restrição para o caso CMM será substituída por

$$y_i(\alpha + \beta^\top \mathbf{x}_i) \geq 1 - \xi_i, \quad i = 1, \dots, n, \quad (18)$$

com $\xi_i \geq 0$.

Teoria - CMF

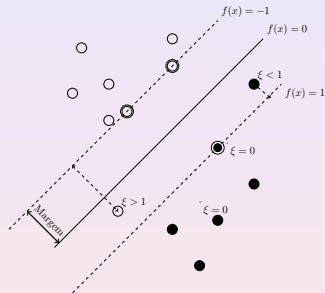


Figura: Detalhes sobre o classificador de margem flexível.

Teoria-CMF

- Pontos para os quais $0 < \xi_i \leq 1$ estão dentro da fronteira da margem, mas do lado correto do hiperplano, e pontos para os quais $\xi_i > 1$ estão do lado errado do hiperplano e são classificados erroneamente. Pontos para os quais $\xi_i = 0$ são corretamente classificados e estão sobre a fronteira da margem ou do lado correto da fronteira da margem.
- Nesse contexto, estamos diante de uma **margem flexível** ou **suave**. O objetivo é maximizar a margem e, para isso, minimizamos

$$C \sum_{i=1}^n \xi_i + \frac{1}{2} \|\beta\|^2, \quad (19)$$

em que $C > 0$ controla o balanço entre a penalidade das variáveis de folga e a margem.

- Como qualquer ponto classificado erroneamente satisfaz $\xi_i > 1$, segue-se que $\sum_{i=1}^n \xi_i$ é um limite superior do número de classificações errôneas. No limite, quando $C \rightarrow \infty$, obtemos o CMM.

Teoria-CMF

- Para minimizar (19) sujeito a (18) e $\xi_i > 0$ consideramos o lagrangeano

$$\begin{aligned} L(\alpha, \beta, \mathbf{x}_i, \lambda, \mu) = & \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^n \xi_i \\ & - \sum_{i=1}^n \lambda_i [y_i f(\mathbf{x}_i) + \xi_i - 1] - \sum_{i=1}^n \mu_i \xi_i, \end{aligned} \quad (20)$$

em que $\lambda_i \geq 0, \mu_i \geq 0$ são multiplicadores de Lagrange.

- Derivando (21) com relação a β, α, ξ_i , obtemos

$$\beta = \sum_{i=1}^n \lambda_i y_i \mathbf{x}_i, \quad \sum_{i=1}^n \lambda_i y_i = 0 \quad (21)$$

e

$$\lambda_i = C - \mu_i. \quad (22)$$

Teoria - CMF

- Substituindo (21) - (22) em (21), temos

$$\tilde{L}(\boldsymbol{\lambda}) = \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), \quad (23)$$

que é uma expressão idêntica ao caso separável, com exceção das restrições, que são diferentes.

- Como $\lambda_i \geq 0$ são multiplicadores de Lagrange e como $\mu_i \geq 0$, de (22) segue que $\lambda_i \leq C$. Logo, precisamos maximizar (23) com respeito às variáveis duais λ_i , sujeito a

$$0 \leq \lambda_i \leq C, \quad (24)$$

$$\sum_{i=1}^n \lambda_i y_i = 0, \quad i = 1, \dots, n. \quad (25)$$

- Novamente, estamos diante de um problema de programação quadrática.

Teoria - CMF

- A previsão para um novo ponto \mathbf{x} é obtida avaliando o sinal de $f(\mathbf{x})$ na equação do hiperplano (eq. 3, Aula 9). Substituindo (21) nessa mesma equação, obtemos

$$f(\mathbf{x}) = \alpha + \sum_{i=1}^n \lambda_i y_i K(\mathbf{x}, \mathbf{x}_i). \quad (26)$$

- Dados para os quais $\lambda_i = 0$ não contribuem para (26). Os dados restantes formam os vetores de suporte. Para esses, $\lambda_i > 0$ e, por (28) abaixo, devem satisfazer

$$y_i f(\mathbf{x}_i) = 1 - \xi_i. \quad (27)$$

- No caso de CMF, as condições de KKT são dadas por

$$\begin{aligned} \lambda_i &\geq 0, & y_i f(\mathbf{x}_i) - 1 + \xi_i &\geq 0, \\ \lambda_i (y_i f(\mathbf{x}_i) - 1 + \xi_i) &= 0, \end{aligned} \quad (28)$$

$$\begin{aligned} \mu_i &\geq 0, & \xi_i &\geq 0, \\ \mu_i \xi_i &= 0, & i &= 1, \dots, n. \end{aligned} \quad (29)$$

Teoria - CMF

- Procedendo como no caso de CMM, obtemos

$$\alpha = \frac{1}{N_{\mathcal{M}}} \sum_{i \in \mathcal{M}} \left(y_i - \sum_{j \in S} \lambda_j y_j K(\mathbf{x}_i, \mathbf{x}_j) \right), \quad (30)$$

em que \mathcal{M} é o conjunto do pontos tais que $0 < \lambda_i < C$.

- Se $\lambda_i < C$, então, por (22), $\mu_i > 0$ e por (29), temos $\xi = 0$ e tais pontos estão na fronteira de separação. Pontos com $\lambda_i = C$ estão dentro da fronteira de separação e podem ser classificados corretamente se $\xi_i \leq 1$ e erroneamente se $\xi_i > 1$.

Teoria-CMNL

- Seja \mathcal{X} o conjunto de dados (ou de **padrões**). A função $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ é um **kernel** se existir um espaço vetorial com produto interno, \mathcal{H} (usualmente um espaço de Hilbert) e uma aplicação $\Phi : \mathcal{X} \rightarrow \mathcal{H}$, tal que, para todos $x, y \in \mathcal{X}$, tivermos

$$K(x, y) = \langle \Phi(x), \Phi(y) \rangle. \quad (31)$$

Φ é a **aplicação característica** e \mathcal{H} , o **espaço característico**.

- Por exemplo, tomemos $\mathcal{X} = \mathbb{R}^2$ e $\mathcal{H} = \mathbb{R}^3$ e definamos

$$\begin{aligned} \Phi : \mathbb{R}^2 &\rightarrow \mathbb{R}^3, \\ (x_1, x_2) &\rightarrow (x_1^2, x_2^2, \sqrt{2}x_1x_2). \end{aligned}$$

Então, se $x = (x_1, x_2)$ e $y = (y_1, y_2)$, é fácil verificar que $\langle \Phi(x), \Phi(y) \rangle = \langle x, y \rangle$; logo $K(x, y) = \langle \Phi(x), \Phi(y) \rangle = \langle x, y \rangle$ é um *kernel*.

Teoria-CMNL

- Para tornar o algoritmo de suporte vetorial não linear, notamos que ele depende somente de produtos internos entre os vetores de \mathcal{X} ; logo, é suficiente conhecer $K(\mathbf{x}, \mathbf{x}^\top) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}^\top) \rangle$, e não Φ explicitamente. Isso permite formular o problema de otimização, substituindo a derivada do Lagrangeano no caso de CMF por

$$\beta = \sum_{i=1}^n \alpha_i \Phi(\mathbf{x}_i). \quad (32)$$

- Agora, β não é mais dado explicitamente como antes. Também, o problema de otimização é agora realizado no espaço característico e não em \mathcal{X} .
- Os *kernels* a serem usados têm que satisfazer certas condições de admissibilidade. Veja Smola e Schölkopf (2004) para detalhes. Os *kernels* mencionados anteriormente são admissíveis.

Regressão via SVM

- Dado um conjunto de treinamento, $\mathcal{T} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$, o objetivo é obter uma função $f(\mathbf{x}_i)$, a mais achatada (*flat*) possível tal que $|y_i - f(\mathbf{x}_i)| < \epsilon$, $i = 1, \dots, n$ em que $\epsilon > 0$ é o maior erro que estamos dispostos a cometer. Por exemplo, ϵ pode ser a máxima perda que admitimos ao negociar com ações dadas certas características obtidas do balanço de um conjunto de empresas.
- No caso de funções lineares, o objetivo é determinar α e β tais que $|f(\mathbf{x}_i)| = |\alpha + \beta^\top \mathbf{x}_i| \leq \epsilon$. A condição de que $f(\mathbf{x})$ seja a mais achatada possível corresponde a que β seja pequeno, ou seja o problema a resolver pode ser expresso como

$$\text{minimizar } \frac{1}{2} \|\beta\|^2 \text{ sujeito a } \begin{cases} y_i - \beta^\top \mathbf{x}_i - \alpha \leq \epsilon, \\ \alpha + \beta^\top \mathbf{x}_i - y_i \leq \epsilon \end{cases} . \quad (33)$$

Regressão via SVM

- Nem sempre as condições (33) podem ser satisfeitas e nesse caso, assim como nos modelos de classificação, podemos introduzir variáveis de folga ξ_i e ξ_i^* , $i = 1, \dots, n$, que permitem flexibilizar a restrição de que o máximo erro permitido seja ϵ . O problema a resolver nesse contexto é

$$\text{minimizar } \frac{1}{2} \|\beta\|^2 + \sum_{i=1}^n C(\xi + \xi^*) \text{ sujeito a } \begin{cases} y_i - \beta^\top \mathbf{x}_i - \alpha \leq \epsilon + \xi_i, \\ \alpha + \beta^\top \mathbf{x}_i - y_i \leq \epsilon + \xi_i^*, \\ \xi_i, \xi_i^* > 0. \end{cases} \quad (34)$$

- A constante $C > 0$ determina um compromisso entre o achatamento da função f e o quanto estamos dispostos a tolerar erros com magnitude maior do que ϵ .

Regressão via SVM

- As soluções de (33) ou (34) podem ser encontradas mais facilmente usando a formulação dual (ver Nota de Capítulo 3). No caso de modelos lineares, a previsão para um elemento com valor das variáveis preditoras igual a \mathbf{x}_0 é obtida de

$$f(\mathbf{x}_0) = \sum_{i=1}^n \hat{\lambda}_i K(\mathbf{x}_0, \mathbf{x}_i) + \hat{\alpha},$$

em que $\hat{\lambda}_i$ são multiplicadores de Lagrange, $K(\mathbf{x}_0, \mathbf{x}_i)$ é um *kernel*, $\hat{\alpha} = y_i - \varepsilon - \hat{\beta}^\top \mathbf{x}_i$ e $\hat{\beta} = \sum_{i=1}^n \hat{\lambda}_i \mathbf{x}_i$.

- Os vetores suporte são aqueles para os quais os multiplicadores de Lagrange $\hat{\lambda}_i$ são positivos.
- Se optarmos por um *kernel* linear, $K(\mathbf{x}, \mathbf{x}_i) = \langle \mathbf{x}_0, \mathbf{x}_i \rangle$.

Regressão via SVM - Exemplo

Consideremos os dados de **distancia** com o objetivo de estudar a relação entre a distância com que motoristas conseguem distinguir um certo objeto e sua idade. O diagrama de dispersão e a reta de mínimos quadrados ajustada ($y = 174,2 - 1,0x$) correspondentes estão apresentados na Figura 3.

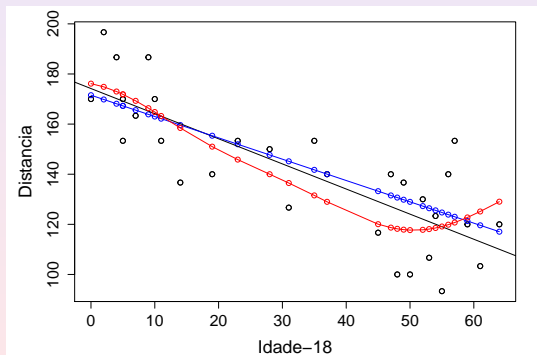


Figura 3: Regressão SVM para os dados de distância.

Regressão via SVM - Exemplo

O ajuste de uma regressão com suporte vetorial baseada num *kernel* linear com os parâmetros *default* pode ser obtido por meio dos comandos

```
model1<- svm(x, y, kernel="linear")  
summary(model1)
```

Parameters:

SVM-Type: eps-regression

SVM-Kernel: linear

cost: 1

gamma: 1

epsilon: 0.1

Number of Support Vectors: 23

```
betahat <- model1$rho
```

```
[1] -0.08572489
```

```
coef1 <- sum(model1$coefs*x[model1$index])
```

```
alfahat <- coef1/model1$rho
```

```
[1] 172.8264
```

de forma que a função previsora corresponde à $f(x) = 172,9 - 0,09x$.

Regressão via SVM - Exemplo

- A previsão para as distâncias segundo esse modelo pode ser obtida por meio do comando `yhat1 <- predict(model1, x)`. O *RMSE* correspondente pode ser obtido por meio do comando `rmse(yhat1, y)` é 16,51464 (maior do que o *RMSE* associado ao ajuste por meio de mínimos quadrados, que é 16,02487).
- Um modelo mais flexível pode ser ajustado com um *kernel* radial do tipo $K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\gamma ||\mathbf{x}_1 - \mathbf{x}_2||^2)$ com $\gamma > 0$ constante. Nesse caso, convém realizar uma análise de sensibilidade com validação cruzada para a seleção da melhor combinação dos valores do máximo erro ϵ que estamos dispostos a cometer e do custo de penalização, C . Isso pode ser concretizado por meio dos comandos

```
sensib <- tune(svm, y ~ x, ranges = list(epsilon = seq(0,1,0.1),
  cost = 2^(2:9)))
```

Parameter tuning of svm:

- ```
- sampling method: 10-fold cross validation
- best parameters:
 epsilon cost
 0.8 8
- best performance: 275.8086
```

## Regressão via SVM - Exemplo

Com esses resultados, realizamos um ajuste por meio de um *kernel* radial com parâmetros  $C = 8$  e  $\epsilon = 0.8$ , obtendo

```
model2 <- svm(x, y, kernel="radial", cost=8, epsilon=0.8)
summary(model2)
```

Parameters:

SVM-Type: eps-regression

SVM-Kernel: radial

cost: 8

gamma: 1

epsilon: 0.8

Number of Support Vectors: 6

O *RMSE* para esse modelo é 15,84272, menor do que aqueles obtidos por meio dos demais ajustes. Um gráfico com os ajustes por mínimos quadrados (em preto) e por regressões com suporte vetorial baseadas em *kernels* linear (em azul) e radial (em vermelho) está apresentado na Figura 3.

## Regressão via SVM - Observações

- Algoritmos de suporte vetorial no contexto de regressão também podem ser utilizados com o mesmo propósito de suavização daquele concretizado pelo método **Lowess** (veja a Nota de Capítulo 2 do Capítulo 5).
- Nesse contexto, a suavidade do ajuste deve ser modulada pela escolha do parâmetro  $\epsilon$ . Valores de  $\epsilon$  pequenos (próximos de zero) geram curvas mais suaves e requerem muitos vetores suporte, podendo produzir sobreajuste. Valores de  $\epsilon$  grandes (próximos de 1,0, por exemplo) geram curvas menos suaves e requerem menos vetores suporte.
- O parâmetro  $C$  tem influência no equilíbrio entre as magnitudes da margem e das variáveis de folga. Em geral, o valor desse parâmetro deve ser selecionado por meio de uma análise de sensibilidade concretizada por validação cruzada.

## Referências

- Morettin, P. A. e Singer, J. M. (2022). *Estatística e Ciência de Dados*. LTC: Rio de Janeiro.
- Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, **14**, 199–222.
- Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. New York: Springer.
- Vapnik, V. (1998). *Statistical Learning Theory*. New York: Wiley.