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 9

27 de março de 2026

Sumário

1. Algoritmos de Suporte Vetorial
2. Classificador de Margem Máxima
3. Classificador de Margem Flexível

Peliminares

- No problema da modelagem de dados, a quantidade e a qualidade das observações governarão o desempenho do modelo.
- Contudo, a escolha de uma abordagem para a modelagem também influenciará o desempenho do modelo.
- Dados são finitos e amostrados, em geral, não uniformemente, e dada a natureza de alta dimensão frequentemente existente, os dados formarão, somente, uma distribuição esparsa no espaço de entrada.
- Consequentemente, o problema é frequentemente mal-posto (*ill-posed*) ou mal colocado.
- Para tratar a natureza mal-posta de um problema, é necessário convertê-lo para um problema bem-posto(*well-posed*).
- Para um problema ser bem-posto, uma única solução deve existir que varie continuamente com os dados.
- Conversão para um problema bem-posto é tipicamente obtida por meio de alguma forma de controle, que tem por objetivo um compromisso do ajuste do modelo com restrições sobre a flexibilidade do modelo, produzindo um modelo robusto que generaliza-se com sucesso.

Peliminares

- Uma abordagem para obter modelos bem-postos é o método de regularização, já abordado em aulas anteriores.
- Uma possibilidade é o uso de algoritmos de suporte vetorial (SVM: support vector machines) em conjunção com métodos usando kernels. Vapnik (1998).
- SVM é uma técnica de classificação/aproximação desenvolvida por Vapnik no contexto de Minimização do risco estrutural (*Structural risk minimization*) (SRM), que tem por objetivo construir modelos parcimoniosos no sentido da teoria de Aprendizado Estatístico (*Statistical Leaning*) (SL) de Vapnik.
- A formulação considera o princípio SRM, que parece ser superior ao princípio de Minimização do erro empírico(*Empirical Risk minimization*)(ERM), tradicionalmente empregado em redes neurais.
- SRM minimiza um limite superior do risco esperado, enquanto o ERM minimiza o erro sobre o conjunto de treinamento.
- É essa diferença que faz com que o SVM tenha uma habilidade grande de generalizar, que é o objetivo do SL.
- SVM foi proposto para reconhecimento de padrões, estimação de regressões, expansões com kernels, métodos de regularização etc.

Kernel SV

- Algoritmos de suporte vetorial (ASV) exploram a ideia de mapear os dados em um espaço de dimensão maior (chamado feature space) onde se pode aplicar um algoritmo linear.
- Normalmente, esta aplicação e suas propriedades são desconhecidas. No lugar de obter esta aplicação explicitamente, usa-se um kernel $K(x, y)$ (*integral operator kernel*) que corresponde ao produto interno dos dados mapeados em um espaço de alta dimensão, ou seja,

$$K(\mathbf{x}, \mathbf{y}) =< \Phi(\mathbf{x}), \Phi(\mathbf{y}) >, \tag{1}$$

em que $\Phi : \mathbb{R}^n \to \mathcal{F}$ denota a aplicação no *feature espace* $\mathcal{F}$, com produto interno $< \cdot, \cdot >$. O ASV pode, então, calcular uma função não linear no espaço dos dados de entrada $\mathbb{R}^n$ e esta função toma a forma

$$f(\mathbf{x}) =< \mathbf{w}, \Phi(\mathbf{x}) > +b, \tag{2}$$

em que $\mathbf{w} \in \mathcal{F}$ é um vetor nesse espaço de funções e $b \in \mathbb{R}$ é o chamado **viés** (*bias*) do modelo.

Kernel SV

- Dado um conjunto de treinamento $D = \{(\mathbf{x}_i, y_i), \mathbf{x}_i \in \mathbb{R}^n, y_i \in \mathbb{R}, i = 1, \ldots, N\}$, queremos minmizar o funcional de risco regularizado (RRF)

$$\begin{aligned} R_{reg}[f] &= C R_{emp}[f] + \frac{1}{2}||\mathbf{w}||^2 \\ &= \frac{C}{N}\sum_{i=1}^{N} c(f(\mathbf{x}_i), y_i) + \frac{1}{2}||\mathbf{w}||^2, \end{aligned} \tag{3}$$

na qual $c(f(\mathbf{x}_i, y_i)$ é uma função convexa determinando a perda relativa a desvios entre o valor estimado $\hat{f}$ e o verdadeiro valor y_i e $C \geq 0$ é uma constante de regularização.

- Para a classe de funções custo convexas, (3) pode ser resolvida usando um algoritmo de programação convexa.
- A solução do algoritmo SVM pode ser representada como uma uma combinação linear poderada de kernels,

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b, \tag{4}$$

na qual os kernels são centrados no dados. Ou seja, f pode ser expressa em termos somente de K, a aplicação Φ aparece somente implicitamente por meio do produto interno em $\mathcal{F}$.

Kernel SV

- Propriedades de K incluem que seja positivo semidefinido (PSD), não negativo e simétrico.
- No argumento apresentado, a função kernel define a classe de funções no espaço do modelo. A questão é: é possível reverter o raciocínio, ou seja, sob quais condições um kernel simétrico, PSD, não negativo K corresponde a um produto interno em algum feature space $\mathcal{F}$.
- A resposta é que K deve satisfazer o Teorema de Mercer. Detalhes em Smola (1998), Smola et al. (1998).

Smola, A. (1998). Learning with kernels. Ph.D. Thesis, Technischen Universität Berlin, Germany.

Smola, A., Schölkopf, B. and Müller, K. (1998). The connection between regularization operators and support vector kernels. *Neural Networkss*, **11** 637–649.

ASV/SVM

- Algoritmos de Suporte Vetorial (ASV), conhecidos na literatura anglo-saxônica como Support Vector Machines (SVM) foram introduzidos por Cortes and Vapnik (1995), que desenvolveram essa classe de algoritmos para classificação binária e englobam técnicas úteis para classificação, com inúmeras aplicações, dentre as quais destacamos reconhecimento de padrões, classificação de imagens, reconhecimentos de textos escritos à mão, expressão de gens em DNAs etc.
- Vapnik and Chervonenkis (1964, 1974) foram, talvez, os primeiros a usar o termo Aprendizado Estatístico (Statistical Learning) em conexão com problemas de reconhecimento de padrões e inteligência artificial.
- Algoritmos de suporte vetorial são generalizações não lineares do algoritmo *Generalized Portrait*, desenvolvido por Vapnik e Chervonenkis (1964).
- Embora a tradução literal do termo proposto por Vapnik seja Máquinas de Suporte Vetorial, optamos por utilizar Algoritmos de Suporte Vetorial para que não se pense que algum tipo de máquina esteja ligado a essas técnicas. Aparentemente, Vapnik utilizou esse termo para enfatizar o aspecto computacional intrínseco à aplicação dos algoritmos.

ASV/SVM

- Uma propriedade importante dos ASV é que a determinação dos parâmetros do modelo corresponde a um problema de otimização convexa, de modo que qualquer solução local é também uma solução global.
- Fazem uso extensivo de Multiplicadores de Lagrange.
- Os algoritmos de suporte vetorial competem com outras técnicas bastante utilizadas, como Modelos Lineares Generalizados (MLG), Modelos Aditivos Generalizados (MAG), Redes Neuronais (Neurais), modelos baseados em árvores etc.
- A comparação com esses métodos é baseada em três fatores: interpretabilidade do modelo usado, desempenho na presença de valores atípicos e poder preditivo.
- Por exemplo, os MLG têm baixo desempenho na presença de valores atípicos, valor preditivo moderado e boa interpretabilidade.
- Por outro lado, os ASV têm desempenho moderado na presença de valores atípicos, alto poder preditivo e baixa interpretabilidade.
- ASV são usados para AE Supervisionado (regressão e classificação).

ASV/SVM

- O princípio operacional fundamental dos ASV é que um kernel (núcleo) é usado para mapear os dados de entrada (ou padrões) em um espaço de dimensão mais alta (feature space), de tal sorte que o problema de classificação, por exemplo, torna-se separável.
- O sucesso da aplicação dos ASV depende da escolha, a priori, desse kernel.
- Os kernels mais populares são:

  Gaussiano
  Polinomial
  Exponential radial basis
  Splines
- Recentemente, têm sido usados kernels baseados em ondaletas.
- Essencialmente, um ASV é implementado por um código computacional que realiza essas tarefas. No Repositório R há pacotes como e1071 e a função svm desenvolvidos com essa finalidade. Outras alternativas são o pacote kernlab e a função ksvm.

ASV/SVM

A abordagem de Cortes and Vapnik (1995) para o problema de classificação baseia-se nas seguintes premissas:

a) Separação de classes: procura-se o melhor hiperplano separador entre as classes, maximizando-se a margem entre os pontos mais próximos das duas classes. Os pontos sobre as fronteiras dessas classes são chamados vetores suporte (**support vectors**).

b) Superposição de classes: pontos de uma classe que estão no outro lado do hiperplano separador são ponderados com baixo peso para reduzir sua influência.

c) Não linearidade: quando não pudermos encontrar um separador linear, utilizamos um **kernel** para mapear os dados de entrada em um espaço de dimensão mais alta (feature space) de tal forma que nesse espaço, são construídos os hiperplanos separadores.

d) Solução do problema: o problema envolve otimização quadrática e pode ser resolvido com técnicas conhecidas.

Fundamentação dos ASV

- Apresentaremos as ideias básicas sobre algoritmos de suporte vetorial (ASV), concentrando-nos no problema de classificação dicotômica, *i.e.*, em que as unidades amostrais devem ser classificadas em uma de duas classes possíveis. Para ideias sobre o caso de mais de duas classes, veja o Texto.
- Adotaremos uma abordagem heurística, mais próxima daquela usualmente empregada em Estatística, deixando para as Notas de Capítulo do Texto a abordagem original (e mais formal) dos ASV.
- Seja $\mathcal{X}$ o espaço dos dados (ou dos padrões); em geral, $\mathcal{X} = \mathbb{R}^d$ e seja a resposta $y \in \{-1, 1\}$.
- Por exemplo, podemos ter dados de várias variáveis explicativas (idade, peso, taxa de colesterol etc.) e uma variável resposta (doença cardíaca, com $y = 1$ em caso afirmativo e $y = -1$ em caso negativo) observadas em vários indivíduos (o conjunto de treinamento). O problema de classificação consiste na determinação de dois subconjuntos (classes) de $\mathcal{X}$, um das quais estará associado a indivíduos com doença cardíaca. O classificador indicará em qual das classes deveremos incluir novos indivíduos (o conjunto de teste) para os quais conhecemos os valores das variáveis explicativas.

Fundamentação dos ASV

Vamos considerar três situações:

1) As classes são perfeitamente separáveis por uma fronteira linear; nesse caso, o separador (hiperplano) é conhecido como classificador de margem máxima (CMM).

- Para duas variáveis, o separador é uma reta; para três variáveis, o separador é um plano. No caso de p variáveis, o separador é um hiperplano de dimensão $p-1$. A Figura 1 é um exemplo. Note que podemos ter mais de uma reta separando as duas classes.

2) Não há um hiperplano que separe as duas classes, como no exemplo apresentado na Figura 2, que corresponde à Figura 1 com pontos trocados de lugar. O separador, neste caso é o classificador de margem flexível (CMF).

3) Um separador linear não conduz a resultados satisfatórios exigindo a definição de fronteiras de separação não lineares. Para isso, recorremos ou a funções não lineares das observações ou a kernels, para mapear o espaço dos dados em um espaço de dimensão maior. O separador, neste caso é o classificador de margem não linear (CMNL).

Fundamentação dos ASV

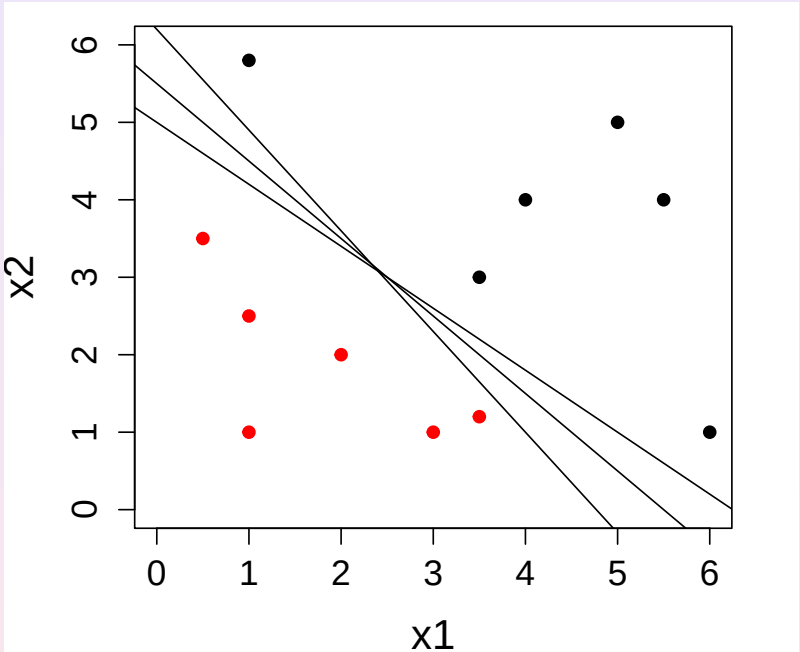


Figura 1: Dois conjuntos de pontos perfeitamente separáveis por um hiperplano (reta).

Fundamentação dos ASV

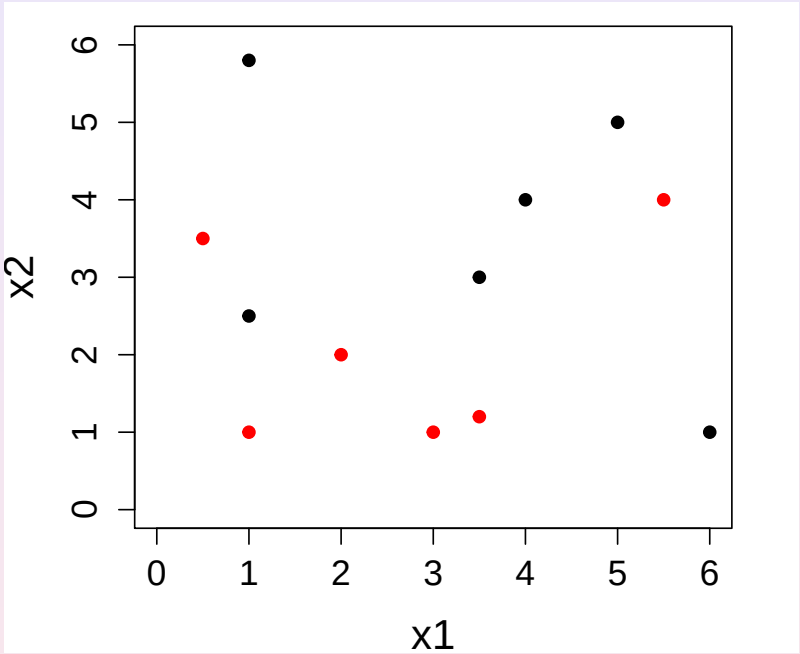


Figura 2: Dois conjuntos de pontos não separáveis por um hiperplano (reta).

Margem e Vetores Suporte

- No caso de duas variáveis, o hiperplano é uma reta com equação $\alpha + \beta_1 X_1 + \beta_2 X_2 = 0$.
- Essa reta separa o plano em duas regiões, uma em que $\alpha + \beta_1 X_1 + \beta_2 X_2 > 0$ e outra em que $\alpha + \beta_1 X_1 + \beta_2 X_2 < 0$.
- Consideremos n observações das variáveis $X_1, \ldots, X_p$, dispostas na forma de uma matriz $\mathbf{X}$, de ordem $n \times p$. Seja $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\top$, o vetor correspondente à i-ésima coluna de $\mathbf{X}$.
- Além disso, sejam $y_1, \ldots, y_n \in \{-1, 1\}$, definindo o conjunto de treinamento $\mathcal{T} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ e seja $\mathbf{x}_0 = (x_{10}, \ldots, x_{p0})^\top$ um **vetor de teste**.

Margem e Vetores Suporte

- Queremos desenvolver um classificador usando um hiperplano separador no espaço $\mathbb{R}^p$ com base no conjunto de treinamento.
- Definindo $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top$, teremos

$$\alpha + \boldsymbol{\beta}^\top \mathbf{x}_i > 0, \quad \text{se} \quad y_i = 1, \tag{5}$$

$$\alpha + \boldsymbol{\beta}^\top \mathbf{x}_i < 0, \quad \text{se} \quad y_i = -1. \tag{6}$$

- Chamemos

$$f(\mathbf{x}) = \alpha + \boldsymbol{\beta}^\top \mathbf{x}. \tag{7}$$

Então, classificaremos $\mathbf{x}_0$ a partir do sinal de $f(\mathbf{x}_0) = \alpha + \boldsymbol{\beta}^\top \mathbf{x}_0$; se o sinal for positivo, $\mathbf{x}_0$ será classificado na Classe 1 (para a qual $y = 1$, digamos), e se o sinal for negativo, na Classe 2 (para a qual $y = -1$). Em qualquer situação, $y_i(\alpha + \boldsymbol{\beta}^\top \mathbf{x}_i) \geq 0$.

Margem e Vetores Suporte

- Como vimos, podem existir infinitos hiperplanos separadores, se os dados de treinamento estiverem perfeitamente separados.
- A sugestão de Vapnik e colaboradores é escolher um hiperplano que esteja o mais afastado das observações de treinamento, chamado de hiperplano de margem máxima.
- A margem é a menor distância entre o hiperplano e os pontos de treinamento.
- O classificador de margem máxima (CMM) é a solução (se existir) do seguinte problema de otimização:

$$\text{maximizar}_{(\alpha,\boldsymbol{\beta})} \quad m(\alpha,\boldsymbol{\beta}) \tag{8}$$

sujeito a

$$\sum_{i=1}^{p} \beta_i^2 = 1, \tag{9}$$

$$y_i(\alpha + \boldsymbol{\beta}^\top \mathbf{x}_i) \geq m(\alpha,\boldsymbol{\beta}), \quad i = 1,\ldots,n. \tag{10}$$

- Dizemos que $m = m(\alpha,\boldsymbol{\beta})$ é a margem do hiperplano e cada observação estará do lado correto do hiperplano se $m > 0$.

Margem e Vetores Suporte

- Os chamados vetores suporte são definidos pelos pontos cujas distâncias ao hiperplano separador sejam iguais à margem e se situam sobre as fronteiras de separação, que são hiperplanos "paralelos" cujas distâncias ao hiperplano separador é igual à margem.
- O classificador depende desses vetores, mas não das demais observações.
- A distância m do hiperplano separador a um ponto do conjunto de treinamento é

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

em que o denominador indica a norma do vetor $\boldsymbol{\beta}$.

Margem e Vetores Suporte

- Como o interesse está nos pontos que são corretamente classificados, devemos ter $y_i f(\mathbf{x}_i) > 0$, $i = 1, \ldots, n$. Então

$$\frac{y_i f(\mathbf{x}_i)}{||\boldsymbol{\beta}||} = \frac{y_i(\alpha + \boldsymbol{\beta}^\top \mathbf{x}_i)}{||\boldsymbol{\beta}||}, \tag{11}$$

  e queremos escolher α e $\boldsymbol{\beta}$ de modo a maximizar essa distância.
- A margem máxima é encontrada resolvendo

$$\text{argmax}_{\alpha,\boldsymbol{\beta}} \left\{ \frac{1}{||\boldsymbol{\beta}||} \text{min}_i \left[y_i(\alpha + \boldsymbol{\beta}^\top \mathbf{x}_i) \right] \right\}. \tag{12}$$

- A solução de (12) é complicada e sua formulação canônica pode ser convertida num problema mais fácil por meio do uso de Multiplicadores de Lagrange.

Exemplo CMM

Consideremos os 12 pontos dispostos na Figura 1, sendo 6 em cada classe. Usando a função svm do pacote e1071 e o comando summary(svm.model) obtemos o seguinte resultado:

```
Call:

svm(formula = type ~ ., data = my.data, type = "C-classification",
kernel = "linear", scale = FALSE)

Parameters:

   SVM-Type:  C-classification

 SVM-Kernel:  linear

       cost:  1

      gamma:  0.5

Number of Support Vectors:  3

 ( 1 2 )

Number of Classes:  2

Levels:

 -1 1
```

Exemplo CMM

- Observe que a função usa o kernel linear, que corresponde ao CMM.
- As opções *cost* e *gamma* serão explicadas adiante.
- Os coeficientes do hiperplano separador, que nesse caso é uma reta, podem ser obtidos por meio dos comandos

```
\alpha = svm.model$ rho

\beta = t(svm.model$coefs) %*% svm.model $ SV
```

e são

```
> alpha

[1] 5.365853

> beta

          x1           x2

[1,] -0.8780489    -1.097561
```

Exemplo CMM

- A equação do hiperplano separador, disposto na Figura 3 é $5,366 - 0,878X_1 - 1,098X_2 = 0$.
- Na mesma figura, indicamos as fronteiras de separação e os vetores suporte, dados pela solução de (8).Note que os coeficientes $\beta_1 = 0,8780489$ e $\beta_2 = -1,097561$ não satisfazem a restrição indicada em (8), pois foram obtidos por meio da formulação canônica do problema em que a restrição é imposta ao numerador de (11). Para detalhes, consulte a Nota de Capítulo 3.
- Neste caso há três vetores suporte (indicados por círculos azuis), um na Classe 1 (ponto em vermelho) e dois na Classe 2 (pontos em preto). Os demais pontos estão em lados separados, delimitados pelas fronteiras de separação (não há pontos entre as fronteiras).
- A margem é $m = 0,71$.

Exemplo CMM

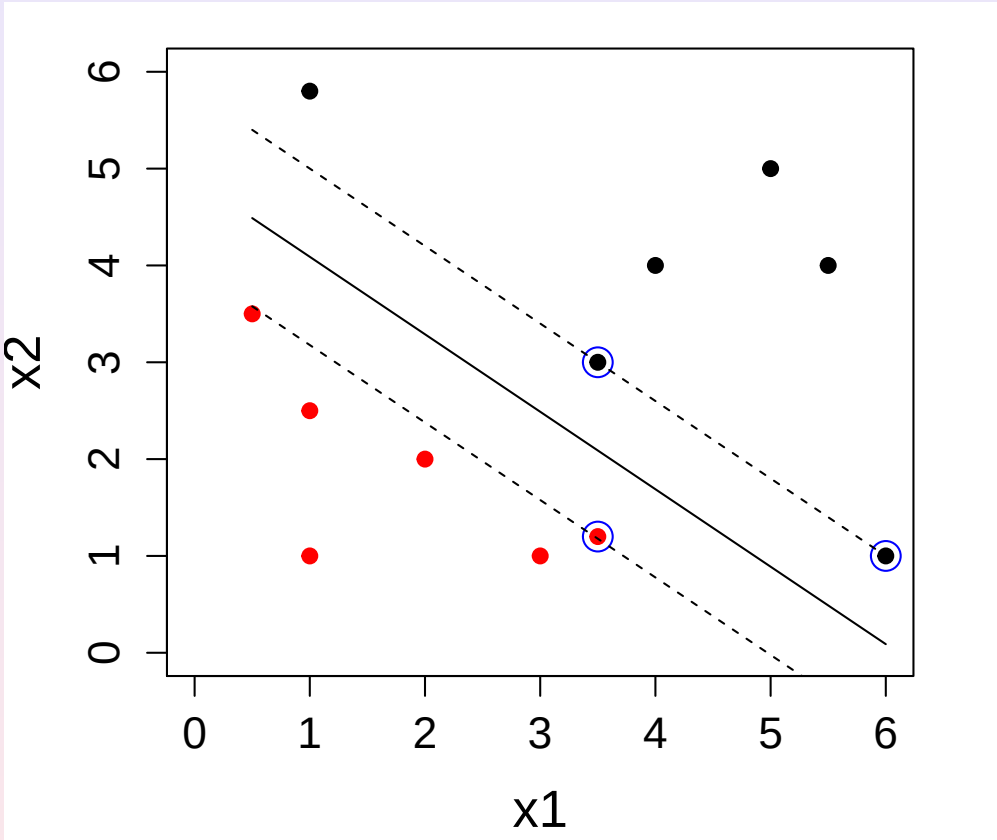


Figura 3: Hiperplano (reta) separador, margem, fronteiras e vetores suporte.

Exemplo CMM

- Consideremos agora dois pontos, $\mathbf{x}_0^*$ e $\mathbf{x}_1^*$, o primeiro na Classe 1 e o segundo na Classe 2 e vamos classificá-los, usando o algoritmo.
- Por meio da função predict, obtemos a Figura 4, que mostra a classificação correta de ambos os pontos (representados nas cores verde e azul).
- Se o problema acima não tiver solução não existirá hiperplano separador, como é o caso apresentado na Figura 2. Nesse caso precisamos recorrer a um classificador que quase separa as duas classes. É o que veremos a seguir.

Exemplo CMM

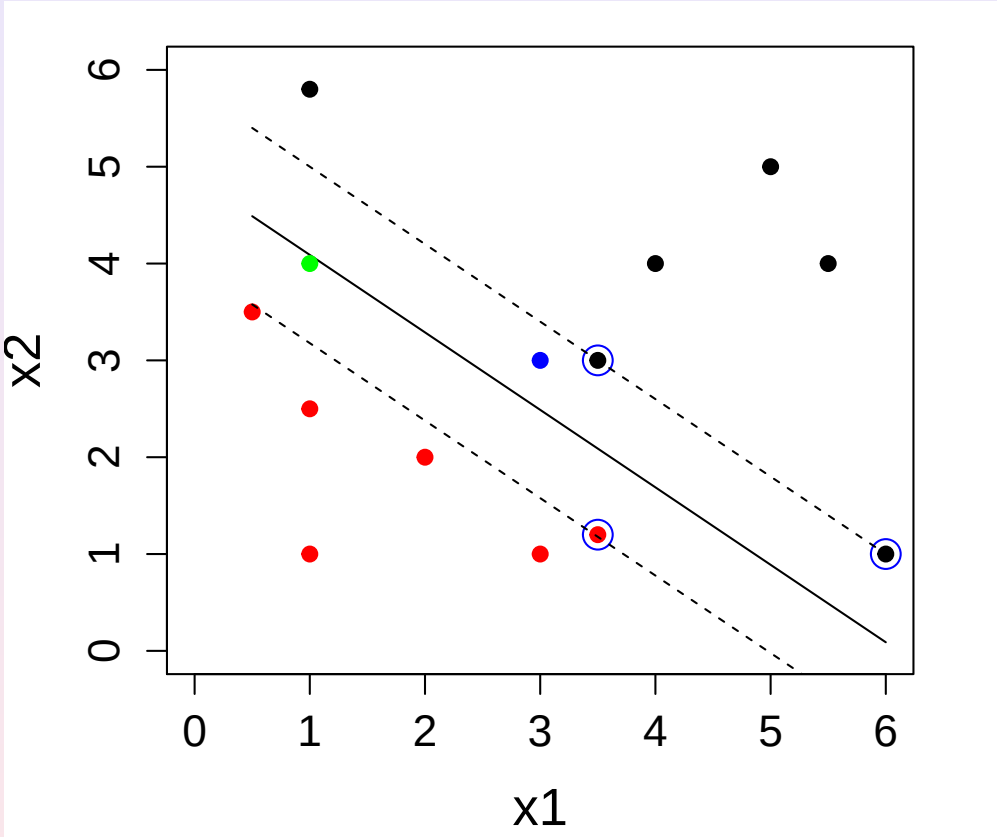


Figura 4: Classificação dos pontos indicados pelas cores verde e azul.

CMF

- Se não existir um hiperplano separador, como aquele do Exemplo anterior, observações podem estar do lado errado da margem ou mesmo do hiperplano, correspondendo nesse caso a classificações erradas.
- O classificador de margem flexível (CMF), também conhecido como classificador baseado em suporte vetorial, é escolhido de modo a classificar corretamente a maioria das observações, o que se consegue com a introdução de variáveis de folga, $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)^\top$, no seguinte problema de otimização:

$$\text{maximizar}_{(\alpha, \boldsymbol{\beta}, \boldsymbol{\xi})} \quad m(\alpha, \boldsymbol{\beta}, \boldsymbol{\xi}), \tag{13}$$

sujeito a

$$\sum_{i=1}^{p} \beta_i^2 = 1, \tag{14}$$

$$y_i(\alpha + \boldsymbol{\beta}^\top \mathbf{x}_i) \geq m(\alpha, \boldsymbol{\beta}, \boldsymbol{\xi})(1 - \xi_i), \tag{15}$$

$$\xi_i \geq 0, \quad \sum_{i=1}^{n} \xi_i \leq C.$$

em que C é uma constante positiva. Veja abaixo para mais detalhes sobre C.

CMF

- Embora esse tipo de classificador seja conhecido como support vector classifier ou soft margin classifier, optamos por denominá-lo "classificador de margem flexível" para diferenciá-lo do "classificador de margem máxima", que também é baseado em vetores suporte.
- As variáveis de folga permitem que observações estejam do lado errado da margem ou do hiperplano. Pontos tais que $\xi_i = 0$ são corretamente classificados e estão sobre a fronteira de separação ou do lado correto da fronteira. Pontos para os quais $0 < \xi_n \leq 1$ estão dentro da fronteira da margem, mas do lado correto do hiperplano, e pontos para os quais $\xi_n > 1$ estão do lado errado do hiperplano e serão classificados erroneamente. Veja a Figura 5, extraída de Bishop (2006). Nessa figura, m está normalizada apropriadamente, veja as Notas de Capítulo 3 e 4.
- O objetivo é maximizar a margem e, então, minimizamos

$$C \sum_{i=1}^{n} \xi_i + \frac{1}{2} ||\boldsymbol{\beta}||^2, \tag{16}$$

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

CMF

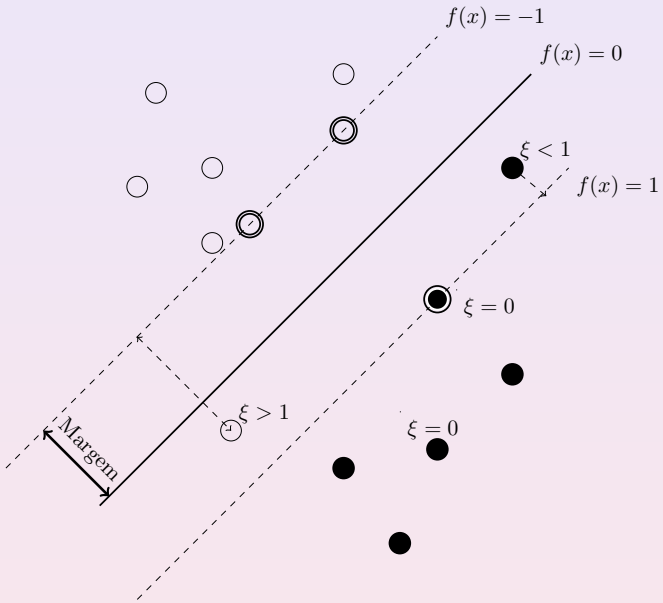


Figura 5: Detalhes sobre o classificador de margem flexível.

CMF

- Como qualquer ponto classificado erroneamente satisfaz $\xi_i > 1$, segue-se que $\sum_{i=1}^{n} \xi_i$ é um limite superior para o número de classificações errôneas. No limite, quando $C \to \infty$, obtemos o CMM.
- Queremos minimizar(16) sujeito a (13). Veja a Nota de Capítulo 4.
- A constante $C \geq 0$ deve ser escolhida apropriadamente e determina o número de violações (classificações erradas) permitidas pelo algoritmo. Se $C = 0$, então não há violações e $\xi_1 = \ldots = \xi_n = 0$. Se C aumenta, a margem fica mais larga e o contrário ocorre se C decresce. O valor de C tem a ver com a relação viés–variância: quando a constante C é pequena, o viés é pequeno e a variância é grande; se C é grande, o viés é grande e a variância é pequena.
  Pode-se dizer que C representa o custo do classificador.
- A constante C normalmente é escolhida por **validação cruzada** O pacote e1071 tem uma função, tune(), que realiza esse procedimento para escolher o melhor modelo, para diferentes valores de C.

Exemplo 1 CMF

Exemplo 1. Consideremos agora os dados dispostos na Figura 3 em que as duas classes não são perfeitamente separáveis. Nesse caso, a utilização da função tune() do pacote e1071 gera o seguinte resultado (editado) indicando que a melhor opção é considerar $C = 4$ e gamma $= 0.5$.

```
Parameter tuning of svm:
- sampling method: 10-fold cross validation
- best parameters:
 gamma   cost
   0.5    4
- best performance: 0.5


- Detailed performance results:
  gamma cost error dispersion
1   0.5    4  0.50  0.4714045
2   1.0    4  0.60  0.4594683
3   2.0    4  0.70  0.4216370
4   0.5    8  0.65  0.4743416
5   1.0    8  0.65  0.4743416
6   2.0    8  0.70  0.4216370
7   0.5   16  0.65  0.4743416
8   1.0   16  0.65  0.4743416
9   2.0   16  0.70  0.4216370
```

Exemplo 1 CMF

Com esses parâmetros, as funções svm e summary geram o seguinte resultado, indicando que há 8 vetores suporte, 4 em cada classe.

```
svm(formula = type ~ ., data = my.data, type = "C-classification",
kernel = "linear", gamma = 0.5, cost = 4, scale = FALSE)
Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  4
      gamma:  0.5
Number of Support Vectors:  8
 ( 4 4 )
Number of Classes:  2
Levels:
 -1 1
```

Exemplo 1 CMF

Um gráfico indicando os vetores suporte e as regiões de classificação correspondentes está apresentados na Figura 6.

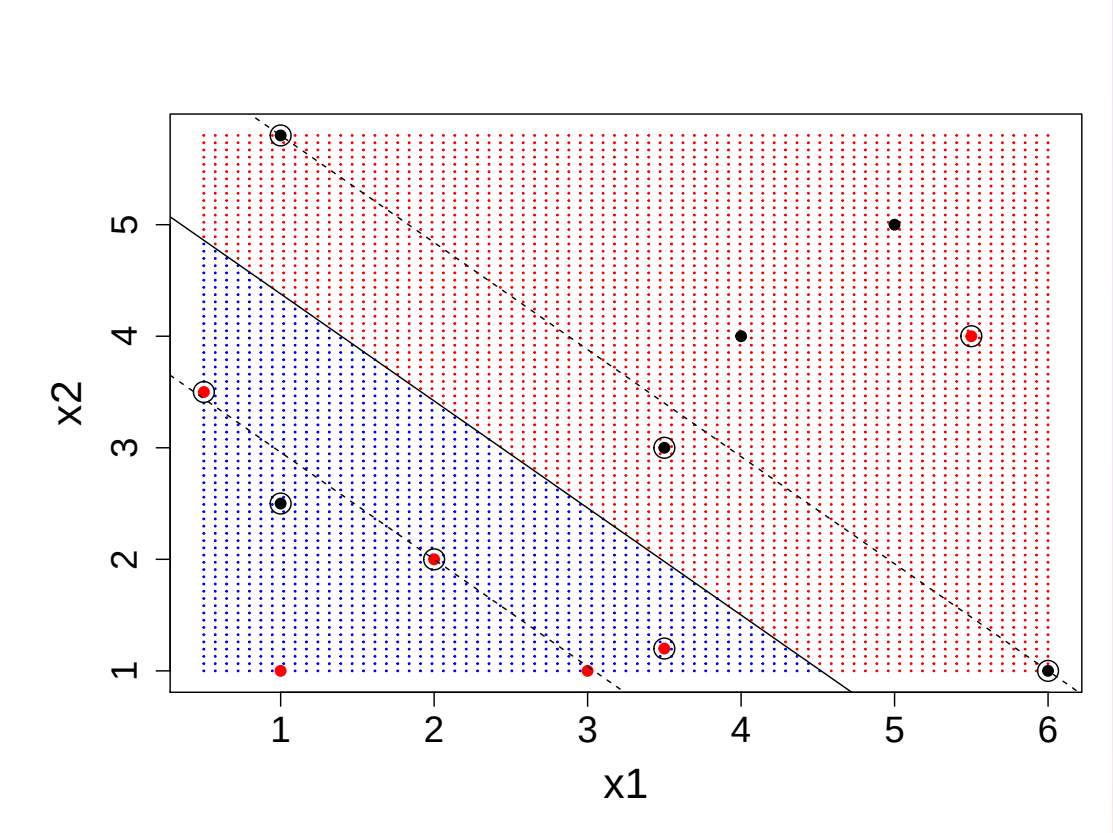


Figura 6: Vetores suporte para os dados da Figura 3.

Exemplo 1 CMF

- A equação do hiperplano classificador é $3,760 - 0,676x_1 - 0,704x_2 = 0$ ou equivalentemente, $x_2 = 3,760/0,704 - 0,676/0,704x_1 = 5,339 - 0,960x_1$. A margem correspondente é $m = (0,676^2 + 0,704^2)^{1/2} = 0,976$. Para detalhes, consulte as Notas de Capítulo 3 e 4.
- Com os comandos svm.pred $<-$ predict(svm.model, my.data) e table(svm.pred, ys) podem-se obter uma tabela com as classificações certas e erradas assim como as classificações determinadas pelo algoritmo. No exemplo, há 2 classificações erradas conforme indicado na Tabela 1.

Exemplo 1 CMF

Tabela 1: Coordenadas e classificação dos pontos do Exemplo 8.1 com classificação predita pelo algoritmo

observação	x1	x2	y	y predito
1	0.5	3.5	1	1
2	1.0	1.0	1	1
3	1.0	2.5	-1	**1**
4	2.0	2.0	1	1
5	3.0	1.0	1	1
6	3.5	1.2	1	1
7	1.0	5.8	-1	-1
8	3.5	3.0	-1	-1
9	4.0	4.0	-1	-1
10	5.0	5.0	-1	-1
11	5.5	4.0	1	**-1**
12	6.0	1.0	-1	-1

Exemplo 2 CMF

- Os dados do arquivo tipofacial foram extraídos de um estudo odontológico realizado pelo Dr. Flávio Cotrim Vellini. Um dos objetivos era utilizar medidas entre diferentes pontos do crânio para caracterizar indivíduos com diferentes tipos faciais, a saber, braquicéfalos, mesocéfalos e dolicocéfalos.
- O conjunto de dados contém observações de 11 variáveis em 101 pacientes. Para efeitos didáticos, utilizaremos apenas a altura facial e a profundidade facial como variáveis preditoras.
- A Figura 7 mostra os três grupos (correspondentes à classificação do tipo facial).

Exemplo 2 CMF

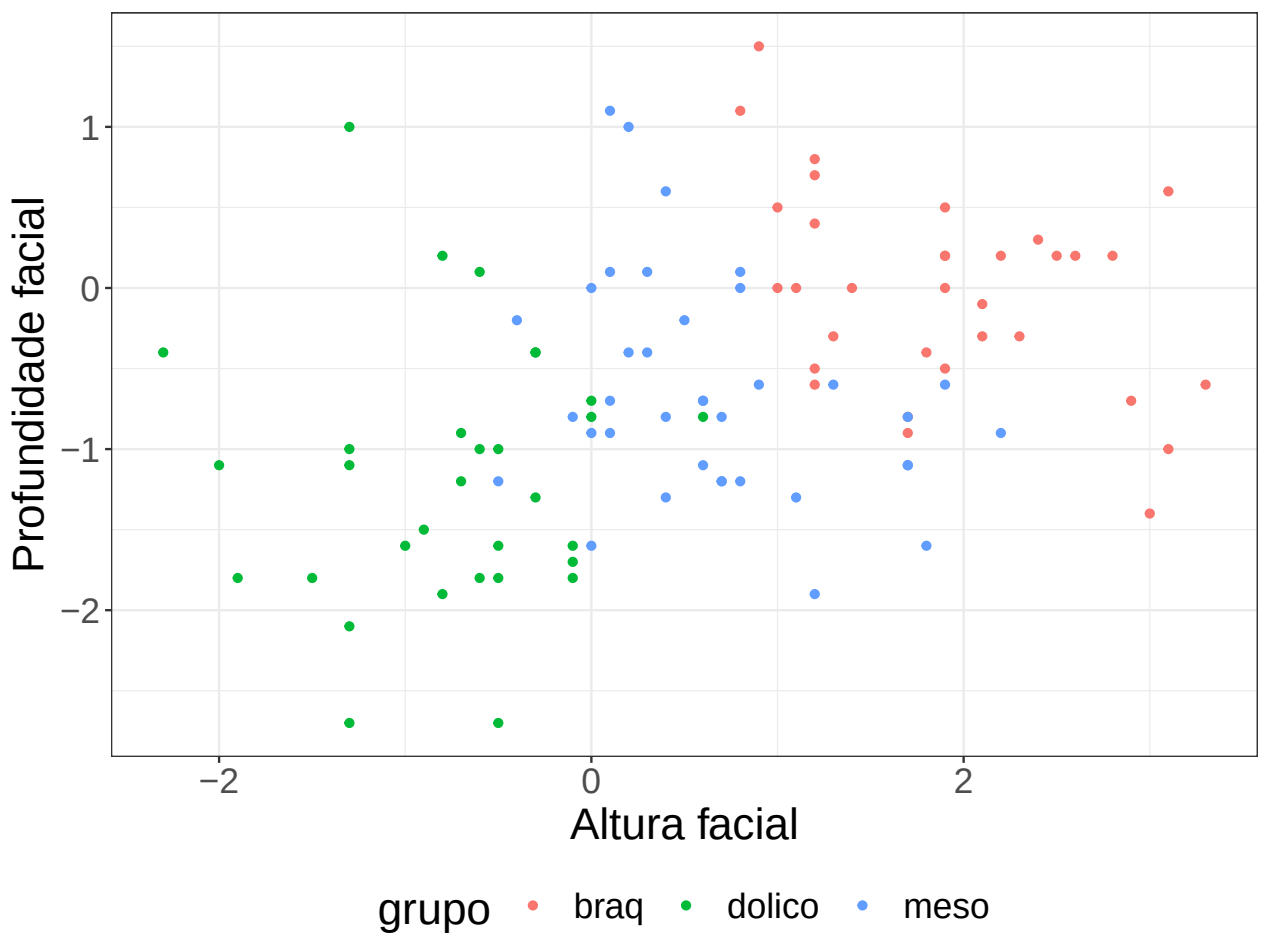


Figura 7: Gráfico de dispersão com identificação dos três tipos faciais.

Exemplo 2 CMF

Utilizando a função tune.svm() do pacote e1071 por meio dos seguintes comandos

```
> escolhaparam <- tune.svm(grupo ~ altfac + proffac, data = face,
                           gamma = 2^(-2:2), cost = 2^2:5,
                           na.action(na.omit(c(1, NA))))
> summary(escolhaparam)
```

obtemos os resultados, apresentados abaixo, que indicam que as melhores opções para os parâmetros C e gamma (obtidas por meio de validação cruzada de ordem 10) para o classificador de margem flexível são $C = 4$ e gamma=2.

Exemplo 2 CMF

```
Parameter tuning of svm:
- sampling method: 10-fold cross validation
- best parameters:
 gamma cost
     2    4
- best performance: 0.1281818
- Detailed performance results:
   gamma cost     error dispersion
1   0.25    4 0.1481818  0.1774759
2   0.50    4 0.1681818  0.1700348
3   1.00    4 0.1681818  0.1764485
4   2.00    4 0.1281818  0.1241648
5   4.00    4 0.1581818  0.1345127
6   0.25    5 0.1481818  0.1774759
7   0.50    5 0.1681818  0.1700348
8   1.00    5 0.1481818  0.1503623
9   2.00    5 0.1281818  0.1148681
10  4.00    5 0.1772727  0.1453440
```

Exemplo 2 CMF

Por intermédio da função svm com os parâmetros $C = 4$ e gamma=2 obtemos o seguinte resultado com o classificador de margem flexível:

```
svm.model <- svm(grupo ~ altfac + proffac, data = face,
                 kernel = "linear", gamma=2, cost=4)
summary(svm.model)

Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  4
Number of Support Vectors:  43
 ( 12 10 21 )
Number of Classes:  3
Levels:
 braq dolico meso
```

Exemplo 2 CMF

A tabela de classificação obtida com os comandos apresentados abaixo, indica o número de classificações certas e erradas.

```
svm.pred  <- predict(svm.model, face)
table(pred = svm.pred, true = face\$grupo)
```

```
              true
pred     braq  dolico  meso
braq      26       0     2
dolico     0      28     4
meso       7       3    31
```

Acurácia=0,84

Exemplo 2 CMF

- Na Figura 8 apresentamos o gráfico de classificação correspondente, obtido por meio do comando

```
plot(svm.model, face, proffac ~ altfac, svSymbol = 4,
dataSymbol = 4,  cex.lab=1.8, main="",
color.palette = terrain.colors)
```

- Uma das características importantes dos classificadores baseados em vetores suporte é que apenas as observações que se situam sobre a margem ou do lado errado da mesma afetam o hiperplano.
- Observações que se situam no lado correto da margem podem ser alteradas (mantendo-se suas classificações) sem que o hiperplano separador seja afetado.

Exemplo 2 CMF

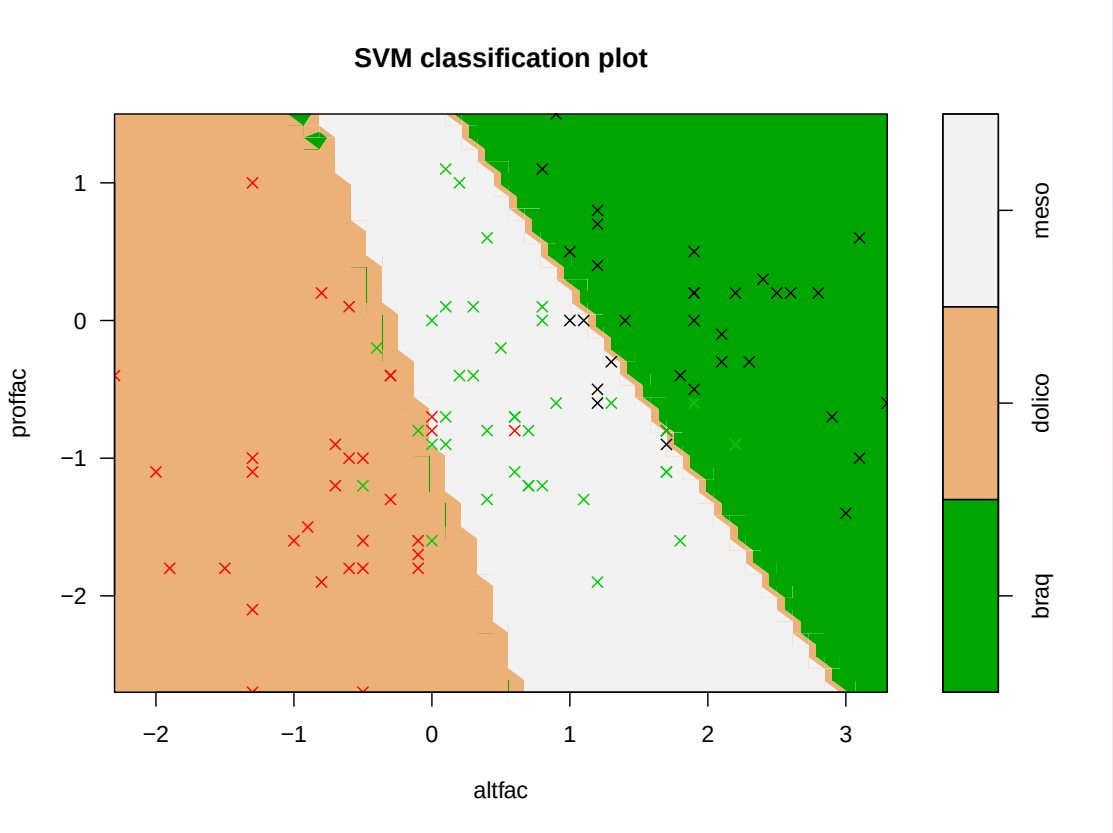


Figura 8: Classificação do tipo facial obtida pelo classificador de margem flexível.

Referências

Cortes, C. and Vapnik, V. (1995). Support vector networks. *Machine Learning*, **20**, 273-297.

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2017). *Introduction to Statistical Learning*. Springer.

Morettin, P. A. e Singer, J. M. (2024). *Estatística e Ciência de Dados*. 2a. edição. Rio de Janeiro: LTC.

Vapnik, V. and Chervonenkis, A. (1964). A note on a class of perceptrons. *Automation and Remote Control*, **25**.

Vapnik, V. and Chervonenkis, A. (1974). *Theory of Pattern recognition* [in Russian]. Moskow: Nauka.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory.* New York: Springer.

Vapnik, V. (1998). *Statistical Learning Theory*. New York: Wiley.