# 3
# Máquinas de Suporte Vetorial para regressão

## 3.1
## Introdução à teoria de aprendizagem supervisionada

Na aprendizagem supervisionada, o objetivo é prever o valor de uma função para qualquer entrada válida utilizando um certo número de exemplos, isto é, dados de treinamento. Tais dados exemplificam as relações entre a entrada e saída. A função de saída pode ser um valor em um espaço discreto, utilizado para classificação ou pode ser um valor em um espaço contínuo, chamado regressão. A solução é escolhida dentre um conjunto de funções candidatas que mapeiam o espaço de entrada no domínio de saída. Usualmente um conjunto particular de funções candidatas é escolhido e denominado *hipótese* antes de se aprender a função correta. O algoritmo que tem com entrada os dados de treinamento e seleciona uma hipótese do espaço de *hipóteses* é referido como algoritmo de aprendizagem.

Para resolver um problema de aprendizagem supervisionada, deve-se seguir os passos:

1. Obter dados de entrada que sejam representativos do que se quer como resposta.

2. Determinar a representação dos dados de entrada, uma vez que, a precisão da função aprendida depende de como os objetos de entrada estão representados. Geralmente, o dado de entrada é transformado em um vetor num espaço vetorial com produto interno que chamamos de *espaço característico*.

3. Determinar a estrutura da função a ser aprendida e o correspondente algoritmo de aprendizagem. Nesta dissertação, utilizamos SVR.

4. O modelo. Utilizar um algoritmo de aprendizagem no conjunto de treinamento. Os parâmetros do algoritmo podem ser ajustados otimizando o desempenho num subconjunto do conjunto de treinamento, chamado conjunto de validação, ou via validação cruzada. Após a escolha dos

parâmetros e aprendizagem, o desempenho do algoritmo pode ser avaliado em um conjunto de teste, o qual é um conjunto separado do conjunto de treino.

### 3.1.1 Vantagens e desvantagens da metodologia

Um ponto marcante da teoria de aprendizagem supervisionada é o grande número de problemas práticos que podem ser resolvidos. Ademais, nesta metodologia basta coletar alguns pares de dados de entrada/saída e usar o algoritmo para aprendizagem da função que relaciona a entrada com a saída.

No entanto, muitas dificuldades merecem uma análise e estudos cuidadosos. Conforme destacado por Cristianini e Shawe-Taylor (7), uma das dificuldades está na escolha da classe de funções das quais o mapeamento dos dados de entrada e saída deve ser procurado, já que a classe deve ser rica o suficiente a fim de que o mapeamento ou uma aproximação seja encontrada, porém a classe não deve ser muito grande para que a complexidade do aprendizado dos exemplos não se torne proibitiva, particularmente quando se leva em consideração o número de exemplos necessários para fazer inferências estatisticamente confiáveis em um ampla classe de funções.

Na prática, esses problemas se traduzem em dificuldades de aprendizagem específicas. Uma delas se refere a ineficiência do algoritmo no caso de existência de mínimos locais. Em segundo lugar, a complexidade da descrição da função de saída, que pode freqüentemente ficar muito grande e sem utilidade na prática. O terceiro problema ocorre se apenas um número limitado de exemplos de treinamento está disponível. Então enriquecer a classe de hipóteses levará a problemas de excesso de ajuste ( "*overfitting* ") e, portanto, a uma generalização ruim. O quarto problema decorre do fato de que o algoritmo de aprendizado freqüentemente é controlado por um grande número de parâmetros cuja escolha é baseada em heurísticas, tornando o sistema difícil e não confiável para usar.

Apesar das dificuldades apresentadas, a metodologia de aprendizagem tem obtido muito sucesso em problemas de interesse prático. A Máquina de Suporte Vetorial é uma ferramenta proeminente nessa área de estudos no qual os quatro problemas apresentados de eficiência no treinamento, eficiência nos dados de teste, excesso de ajuste ("*overfitting* ") e ajuste de parâmetros do algoritmo são evitados. Deve-se ressaltar que o problema de excesso de ajuste é parcialmente resolvido pelo SVR.

## 3.2
## O Desenvolvimento da teoria SVM

No início da década de 60, o algoritmo de máquinas de suporte vetorial foi desenvolvido para construir hiperplanos separadores para problemas de reconhecimento de padrões (Vapnik e Lerner, 1963 (28), Vapnik e Chervonenkis, 1964 (27)). Embora suas características já tivessem sido apresentadas e utilizadas desde a década de 60, a teoria SVM foi introduzida pela primeira vez em um artigo na Annual Conference on Computational Learning Theory (COLT) em 1992 (2).

Na década de 90, o método foi generalizado para construir funções separadoras não lineares (porém lineares em um espaço característico) (Boser et al. 1992 (2), Cortes (6)e Vapnik 1995 (30)). Em 1995, o método foi generalizado para estimar funções de valor real (regressão) (Vapnik, 1995)(30).

Os dois livros recentes escritos por Vapnik(30, 29) fornecem um conhecimento teórico muito extenso no desenvolvimento do conceito de Máquinas de Suporte Vetorial. Podemos ainda destacar dois tutoriais. Um tutorial sobre classificadores baseados em suporte vetorial foi publicado por Burges (1998) (3) e outro sobre Máquinas de Suporte Vetorial para Regressão foi publicado por Smola e Schölkopft em 2003 (23).

A Máquina de Suporte Vetorial é um sistema de aprendizado treinado com um algoritmo de otimização baseado na teoria estatística de aprendizagem, que implementa a seguinte idéia (veja figura 3.1): vetores do espaço de entrada são mapeados não linearmente para um espaço característico de alta dimensionalidade, através de um mapeamento escolhido a priori e, nesse espaço uma superfície de decisão linear é construída, constituindo um hiperplano de separação ótima de exemplos, como por exemplo, a separação binária entre exemplos que possuem rótulos positivos e negativos, tal que a margem de separação seja máxima.

O objetivo da classificação de vetores de suporte está em conceber uma maneira computacionalmente eficiente, ou seja, capaz de lidar com diferentes tamanhos de amostras e capaz de aprender "bons" hiperplanos com separação ótima em um espaço característico de grande dimensionalidade, onde um bom hiperplano de separação se refere àquele que tem grande habilidade de generalização. A teoria de generalização nos fornece um guia sobre como controlar a capacidade e prevenir problemas de excesso de ajustes ("*overfitting*") através do controle das medidas da margem do hiperplano. Já a teoria de otimização fornece as técnicas matemáticas necessárias para encontrar tais hiperplanos. Para maiores detalhes, consulte Cristianini e Shawe-Taylor (7).

O modelo de SVM mais simples é o classificador binário de margem

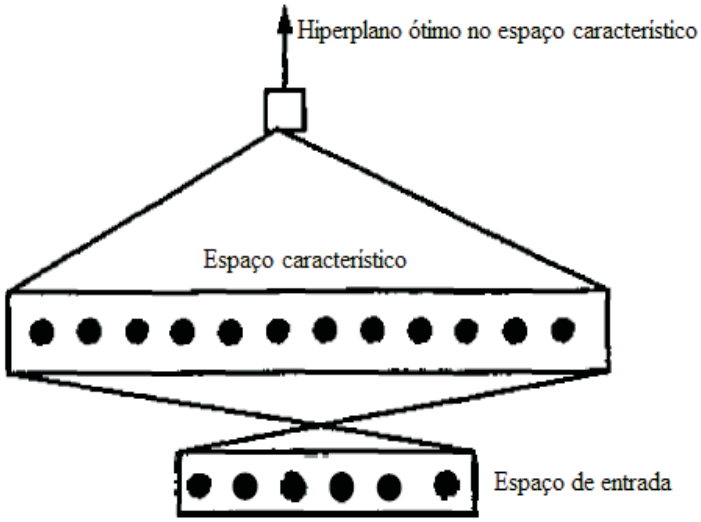


Figura 3.1: A máquina de vetores de suporte (Vapnik, 1998 (29)).



máxima, o qual funciona apenas para dados linearmente separáveis no espaço característico e por isso não se aplica em muitos problemas do mundo real. No entanto, esse classificador é o ponto de partida para o entendimento de SVMs mais complicados, podendo ser estendido para lidar com dados não separáveis, permitindo que alguns erros ocorram, mas com uma certa penalidade (veja figura 3.2).

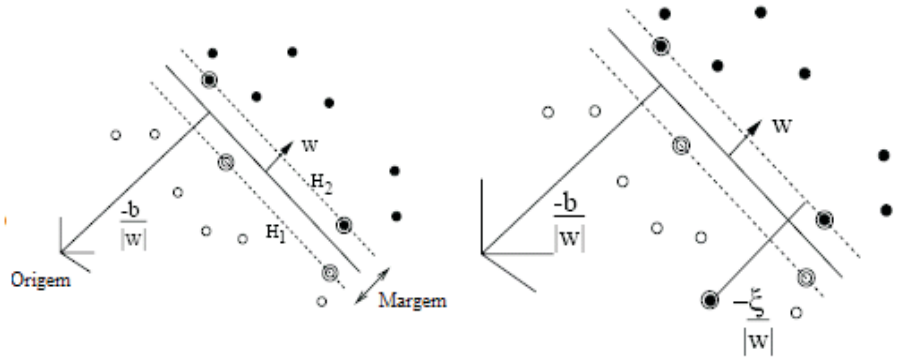


Figura 3.2: Hiperplano linear de separação ótima para dados linearmente separáveis (esquerda) e dados não separáveis (direita). Os vetores suporte estão circulados (Burges (1998) (3)).

O estudo de SVMs constitui uma área de pesquisa ativa. Sua utilização em regressão e em séries temporais é muito promissora, como podemos destacar a dissertação de doutorado de Martin (2005) (14) e o artigo de Silva e Ferreira (2006) (21) na área de energia elétrica.

## 3.3
## Teoria de SVMs na aproximação por regressão

Considere um conjunto de $l$ amostras $\{x_i\}$ onde $x_i \in \mathcal{X}$ (por exemplo, $\mathcal{X} = \mathbb{R}^n$) são os dados de entrada e $y_i \in \mathbb{R}$, os valores alvos para $i = 1, 2, 3, ..., l$. Nesta aplicação, por exemplo, o espaço de entrada é composto por taxas *swap* de diferentes maturidades juntamente com algumas variáveis macroeconômicas em meses subseqüentes e os valores de saída correspondem a uma taxa *swap* com vencimento específico, em meses subseqüentes. Assume-se que os dados são iid (independentes e identicamente distribuídos) e que existe alguma distribuição de probabilidade desconhecida $P(x, y)$ de onde os dados são obtidos.

O objetivo é aprender o mapeamento $x_i \rightarrow y_i$, ou melhor, encontrar uma função a qual aproxime o mapeamento dos dados de entrada (os $x_i's$) para os dados de saída (os $y_i's$), os quais são números reais. Essa aprendizagem é feita com base em dados de treinamento. Em outras palavras, deseja-se encontrar uma função $f$ que relacione a entrada com a saída ($x \rightarrow f(x)$). Também será determinado um critério para verificar a qualidade das estimativas vindas dos dados. A diferença entre o valor dado por $f(x_i)$ e o valor de $y_i$ do arranjo dos dados de treinamento é chamada resíduo da saída e é uma indicação da precisão do ajuste no ponto $x_i$. Deve-se decidir como medir a importância desta precisão, pois resíduos pequenos podem ser aceitáveis mas é desejável que resíduos grandes sejam evitados.

Para considerar dados de treinamento que apresentem ruídos é necessário estabelecer funções de custo (também chamadas de funções de perda) que penalizem tais singularidades. Será determinada uma função de custo a qual determinará como penalizar os erros de estimação. A função de custo determina a importância da precisão e na maioria dos casos será do tipo $c(x, y, f(x)) = c(f(x) - y)$ e pode ser considerada como o custo decorrido de uma falha para prever precisamente uma dada variável.

O objetivo é encontrar a função $f$ que minimiza o valor esperado da perda, fornecido pelo risco funcional:

$$R[f] = \int c(x, y, f(x))dP(x, y), \tag{3-1}$$

onde $c(x, y, f(x))$ denota uma função de custo.

Uma vez que a distribuição $P(x, y)$ é desconhecida, a utilização do risco funcional é inútil. Dessa maneira, pode-se apenas usar os dados de treinamento $(x_i, y_i)$ para estimar uma função $f$ que de alguma maneira esteja "próxima" daquela que minimiza $R[f]$.

### 3.3.1
### Princípio indutivo da minimização do risco empírico

Uma aproximação possível para o risco funcional consiste em substituir a integral por uma estimativa empírica, a fim de obter o chamado risco funcional empírico,

$$R_{emp}[f] := \frac{1}{l}\sum_{i=1}^{l} c(x_i, y_i, f(x_i)). \tag{3-2}$$

Observe que a distribuição de probabilidade não aparece nesta formulação. Este risco é definido apenas como uma medida do erro médio no conjunto de treinamento (para um número de observações finitas e fixas).

Uma tentativa seria encontrar uma função $f$ que minimizasse o risco empírico (princípio da minimização do risco empírico), ou seja,

$f_0 := \arg\min_{f\in F} R_{emp}[f]$ para uma classe de funções $F$.

É possível fornecer condições a máquina de aprendizagem que garantam que assintoticamente (quando l$\rightarrow \infty$), o risco empírico irá convergir para o risco esperado. No entanto, um dos problemas que decorrem da tentativa de usar uma função $f$ que minimize o risco empírico de uma classe de funções $F$ ocorre quando esta classe de funções é muito rica, o que acontece, por exemplo, quando se tem poucos dados em espaços de grande dimensionalidade, podendo ocorrer grandes desvios, excesso de ajuste ( "*overfitting* ") e propriedades de generalização ruins, ou seja, pode-se obter uma função $f \in F$ que faz uma ótima predição dos valores $y_i$ nos dados de treino e, no entanto, não será possível garantir um bom desempenho nos dados de teste.

Uma tentativa de resolver o problema de minimizar o risco empírico seria restringir a classe de soluções admissíveis, ou seja, reduzir a classe $F$ para um conjunto compacto. Essa técnica foi introduzida por Tikhonov e Arsenin em 1963 (26) para resolver problemas inversos e desde então tem sido aplicada a máquinas de aprendizagem com sucesso.

No entanto, ao invés de especificar o conjunto compacto $F$, um termo de estabilização ou regularização $\Omega[f]$ é adicionado à função objetivo original para limitar a complexidade da classe de funçoes $F$, constituindo a classe de risco funcional regularizada:

$$R_{reg}[f] := R_{emp}[f] + \lambda\Omega[f]. \tag{3-3}$$

onde $\lambda > 0$ é a chamada constante de regularização e especifica a troca entre a minimização do risco empírico e a simplicidade garantida por pequenos valores de $\Omega[f]$.

Usualmente $\Omega[f]$ é escolhido para ser convexo uma vez que o $R_{emp}[f]$ também é convexo e assim obtém-se um único mínimo global.

O princípio de minimização do risco empírico é um dos principais assuntos dos livros de Vapnik (30, 29) e maiores detalhes também podem ser obtidos em (18), onde é feita uma discussão das relações envolvendo regularização, Espaço de Hilbert de Núcleo Reproduzível (Reproducing Kernel Hilbert Spaces - RKHS), espaços característicos e operadores de regularização e suas relações com núcleos para vetores de suporte.

### 3.3.2
### O problema de otimização

Inicialmente, buscam-se funções lineares $f$ da seguinte maneira:

$f(x) = \langle w, x\rangle + b$, com $w \in \mathcal{X}$, $b \in \mathbb{R}$,onde $\langle\ ,\rangle$ denota o produto interno em $\mathcal{X}$.

Com relação a escolha do termo de regularização, considera-se $\Omega[f] := \frac{1}{2}\|w\|^2$ uma escolha comum para classificação e regressão. Em regressão, a interpretação geométrica para minimizar $\frac{1}{2}\|w\|^2$ é obter uma função linear aderente com qualidades de aproximação suficientes. Logo,

$$R_{reg}[f] := R_{emp}[f] + \frac{\lambda}{2}\|w\|^2 = \frac{1}{l}\sum_{i=1}^{l} c(x_i, y_i, f(x_i)) + \frac{\lambda}{2}\|w\|^2 \qquad (3\text{-}4)$$

onde $\lambda > 0$ tem que ser escolhido baseado nos dados disponíveis $\lambda(l, (x_i, y_i))$, $i = 1, ..., l$.

Minimizar (3-4) captura a principal idéia da teoria de aprendizagem, ou seja, com o objetivo de se conseguir um risco pequeno, é necessário controlar tanto o erro de treinamento quanto à complexidade do modelo através da explicação dos dados com um modelo simples.

A função de custo deve ser escolhida de forma a melhor representar as incertezas (ruídos) do problema de otimização proposto. Para um grande número de funções de custo, a equação (3-4) pode ser minimizada resolvendo-se um problema de otimização quadrática, o qual possui solução única.

Primeiramente, estuda-se uma regressão com vetores de suporte com o intuito de aprender uma função $f$ que tenha no máximo um desvio $\varepsilon$ em $x_i$ em relação aos valores alvos $y_i$ para todos os dados de treinamento, ou seja, a função $f$ poderá apresentar erros no intervalo $[y_i - \varepsilon, y_i + \varepsilon]$, $i = 1, ..., l$ mas valores de erros superiores a $\varepsilon$ não serão aceitos. Ao mesmo tempo, a função a ser aprendida deve apresentar a máxima aderência possível.

A aderência de $f$ pode ser conseguida pela determinação de um fator $w$ que seja o menor possível, o que pode corresponder à minimização da norma euclidiana de $w$, isto é, $||w||^2 = \langle w, w \rangle$. Tal função deve ainda satisfazer as restrições de erro $\mid f(x_i) - y_i \mid \leq \varepsilon$ para todo $i = 1, ..., l$ onde $S = \{(x_1, y_1), ..., (x_l, y_l)\}$ compõem os dados de treinamento. Assim, obtém-se o seguinte problema de otimização convexa:

$$\begin{array}{ll} \text{minimize}_{w,b} & \frac{1}{2} \parallel w \parallel^2 \\ \text{sujeito a} & \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon, & i = 1, ...l \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon, & i = 1, ..., l \end{cases} \end{array} \tag{3-5}$$

O problema (3-5) supõe a existência de uma função $f$ que aproxima todos os pares $(x_i, y_i)$ com precisão $\varepsilon$ especificada a priori, em outras palavras, que o problema de otimização é viável. Mas, algumas vezes esse pode não ser o caso ou deseja-se que alguns erros ocorram. Por essa razão, estuda-se um tipo de função denominada função de custo $\varepsilon$-insensível, a qual introduz variáveis de folga não-negativas $\xi_i$,$\xi_i^*$, $i = 1, ..., l$ que consideram os pontos situados fora da margem $\mid f(x_i) - y_i \mid \leq \varepsilon$ com uma certa penalidade, separando os dados de treinamento com um número mínimo de erros além de manter restrições que de outra forma seriam inviáveis no problema de otimização anterior. Essa função de custo foi desenvolvida por Cortes e Vapnik,1995 [6]. A função de custo é descrita a seguir (Veja figura 3.3):

$$\mid \xi \mid_\varepsilon := \begin{cases} 0 & \text{se } \mid \xi \mid \leq \varepsilon \\ \mid \xi \mid - \varepsilon & \text{caso contrário} \end{cases} \tag{3-6}$$

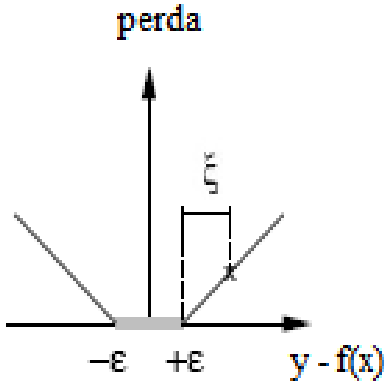


Figura 3.3: Função de custo $\varepsilon$-insensível linear (Schölkopf, Smola, 2001(18)).

Consegue-se a seguinte formulação do problema:

$$
\begin{array}{ll}
minimize_{w,b} & \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{l} (\xi_i + \xi_i^*) \\
sujeito\ a & \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i, & i = 1, ...l, \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^*, & i = 1, ...l, \\ \xi_i, \xi_i^* \geq 0, & i = 1, ...l. \end{cases}
\end{array} \tag{3-7}
$$

As duas variáveis de folga $\xi_i$ e $\xi_i^*$ são introduzidas, uma por estar mais de $\varepsilon$ abaixo do valor alvo e a outra por exceder o valor alvo por mais de $\varepsilon$, respectivamente. O vetor $w$ e o escalar $b$ são a solução do problema de otimização apresentado. A constante $C > 0$ é escolhida pelo usuário e determina o compromisso entre a aderência de $f$ e o montante de desvios maiores do que $\varepsilon$ tolerados. Deve-se atentar para o fato de que cada escolha de $C$ corresponde à escolha de um valor para $\parallel w \parallel^2$ e então minimizar $\xi_i, \xi_i^*$ para aquela escolha de $w$.

Consegue-se um problema cuja função objetivo é quadrática e possui restrições que são desigualdades lineares, logo, um problema de otimização quadrática. Além disso, ele é convexo e, portanto, terá solução única.

Na regressão com vetores suporte uma precisão epsilon é especificada a priori, gerando um região tubular com raio $\varepsilon$ em torno dos dados. O compromisso entre a complexidade do modelo e os pontos situados fora da região tubular (com variáveis de folga $\xi_i, \xi_i^*$ positivas) é determinado pela minimização de (3-7). A figura 3.4 descreve a situação.

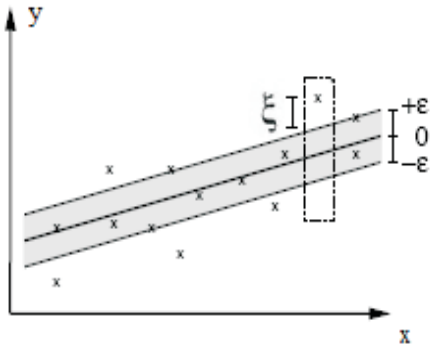


Figura 3.4: Hiperplano separador linear com a função de custo $\varepsilon$-insensível (linear)(Schölkopf, Smola, 2001(18)).

No caso anterior os desvios são penalizados de forma linear. Analogamente, pode-se considerar uma função de custo $\varepsilon$-insensível quadrática conforme mostra a figura 3.5 .

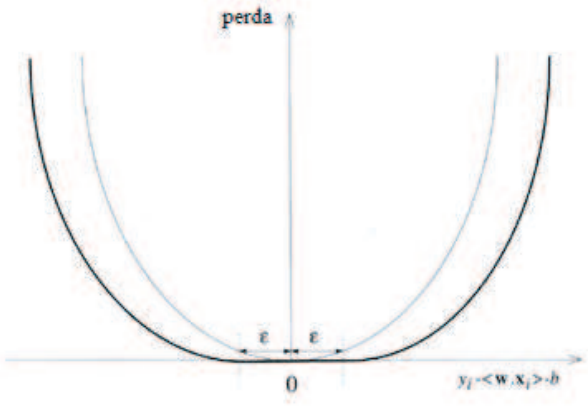


Figura 3.5: A função de custo $\varepsilon$-insensível quadrática (Cristianini, Shawe-Taylor, 2000 (7)).

O problema primal baseado na função de custo $\varepsilon$-insensível quadrática pode ser definido da seguinte maneira:

$$\begin{aligned} &minimize_{w,b} \quad \| w \|^2 + C \sum_{i=1}^{l} (\xi_i^2 + \xi_i^{*2}), \\ &sujeito\ a \quad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i, & i = 1, ...l, \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^*, & i = 1, ...l, \\ \xi_i, \xi_j^* \geq 0, & i = 1, ...l. \end{cases} \end{aligned} \tag{3-8}$$

Cada escolha da função de custo irá resultar em uma estratégia diferente para realizar a regressão. Tais funções devem ser convexas a fim de garantir a existência e unicidade (para convexidade estrita) da solução do problema de otimização. Na tabela seguinte, serão apresentadas algumas funções de custo comumente usadas.

| | Função de custo |
|---|---|
| $\varepsilon$-insensível | $c(\xi) = \mid \xi \mid_\varepsilon$ |
| Laplaciano | $c(\xi) = \mid \xi \mid$ |
| Gaussiano | $c(\xi) = \frac{1}{2}\xi^2$ |
| Perda Robusta de Huber | $c(\xi) = \begin{cases} \frac{1}{2\sigma}(\xi)^2 & se \mid \xi \mid \leq \sigma \\ \mid \xi \mid - \frac{\sigma}{2} & caso\ contrário \end{cases}$ |
| Polinomial | $c(\xi) = \frac{1}{p} \mid \xi \mid^p$ |
| Polinomial por partes | $c(\xi) = \begin{cases} \frac{1}{p\sigma^{p-1}}(\xi)^p & se \mid \xi \mid \leq \sigma \\ \mid \xi \mid - \sigma\frac{p-1}{p} & caso\ contrário \end{cases}$ |

Tabela 3.1: Funções de custo comumente usadas.

No modelo SVR, faz-se uso da função de custo $\varepsilon$-insensível linear, embora outras funções possam ser usadas e o desempenho com cada uma dessas funções comparada, como pode ser verificado em (15) onde são usadas as funções de custo $\varepsilon$-insensível e a robusta de Hubber em dois conjuntos de dados diferentes.

### 3.3.3 Conversão para o Dual

Sabe-se que trabalhar com restrições de desigualdades diretamente é difícil. Assim, será obtida uma descrição dual alternativa que geralmente é mais fácil de resolver que o primal, através de uma reformulação Lagrangiana do problema com a introdução de multiplicadores de Lagrange ou variáveis duais para cada uma das restrições em (3-7). Lembre-se de que para restrições da forma $g_i \geq 0$, as equações são multiplicadas por multiplicadores de Lagrange positivos e subtraídas da função objetivo para formar o Lagrangiano. Para restrições de igualdade, os multiplicadores de Lagrange são irrestritos. O Lagrangiano é dado a seguir:

$$\begin{aligned} L := \ & \tfrac{1}{2} \parallel w \parallel^2 + C \textstyle\sum_{i=1}^{l} (\xi_i + \xi_i^*) - \sum_{i=1}^{l} (\eta_i \xi_i + \eta_i^* \xi_i^*) \\ & - \textstyle\sum_{i=1}^{l} \alpha_i (\varepsilon + \xi_i - y_i + \langle w, x_i \rangle + b) \\ & - \textstyle\sum_{i=1}^{l} \alpha_i^* (\varepsilon + \xi_i^* + y_i - \langle w, x_i \rangle - b) \end{aligned} \tag{3-9}$$

onde $\eta_i, \eta_i^*, \alpha_i, \alpha_i^*$ são os multiplicadores de Lagrange. Portanto, as variáveis duais devem satisfazer as restrições de positividade, isto é, $\eta_i, \eta_i^*, \alpha_i, \alpha_i^* \geq 0$.

O problema de otimização pode ser visto como sendo a minimização da função de Lagrange com relação às variáveis primais ou maximização em relação aos multiplicadores de Lagrange.

A função de Lagrange $L$ será minimizada em relação às variáveis primais $w, b,\ \xi_i, \xi_i^*$ calculando as derivadas parciais de $L$ com respeito a cada uma dessas variáveis primais e igualando-as a zero , obtendo assim:

$$\partial_b L = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) = 0 \tag{3-10}$$

$$\partial_w L = w - \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) x_i = 0 \tag{3-11}$$

$$\partial_{\xi_i} L = C - \alpha_i - \eta_i = 0 \tag{3-12}$$

$$\partial_{\xi_i^*} L = C - \alpha_i^* - \eta_i^* = 0 \tag{3-13}$$

Substituindo 3-10, 3-11, 3-12 e 3-13 em 3-9 resolve-se o problema de otimização dual 3-14. A condição 3-10 implica na eliminação da variável primal $b$. Através das condições 3-12 e 3-13 temos $\eta_i = C - \alpha_i$ e $\eta_i^* = C - \alpha_i^*$ e, a eliminação das variáveis duais $\eta_i$ e $\eta_i^*$.

$$\begin{array}{ll} maximize & \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle x_i, x_j\rangle \\ -\varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) \end{cases} \\ sujeito\ a & \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \ \text{ e } \alpha_i, \alpha_i^* \in [0, C] \end{array} \tag{3-14}$$

A condição (3-11) mostra que:

$$w = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)x_i, \quad portanto, \quad f(x) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\langle x_i, x\rangle + b. \tag{3-15}$$

A observação da equação 3-15 permite destacar alguns pontos que serão importantes na extensão para o caso não-linear:

- $w$ é descrito como a combinação linear de pontos de treinamento $x_i$ onde $\alpha_i, \alpha_i^*$ são soluções do problema dual. Essas variáveis duais possuem uma interpretação intuitiva como forças que agem na estimativa $f(x_i)$ a fim de que a mesma se aproxime das medições alvo desejadas $y_i$.
- o algoritmo completo pode ser descrito em termos de produto interno entre os dados. Ao se obter $f(x)$ não será preciso obter $w$ explicitamente.

### 3.3.4 Condições complementares de Karush-Kuhn-Tucker (KKT)

Pode-se obter uma relação importante entre as soluções primal e dual através das condições complementares de KKT (veja 2.14).

As condições de KKT para o problema de otimização apresentado são:

$$\alpha_i(\varepsilon + \xi_i - y_i + \langle w, x_i\rangle + b) = 0, \quad i = 1, \cdots, l, \tag{3-16}$$

$$\alpha_i^*(\varepsilon + \xi_i^* + y_i - \langle w, x_i\rangle - b) = 0, \quad i = 1, \cdots, l, \tag{3-17}$$

$$(C - \alpha_i)\xi_i = 0, \quad i = 1, \cdots, l, \tag{3-18}$$

$$(C - \alpha_i^*)\xi_i^* = 0, \quad i = 1, \cdots, l, \tag{3-19}$$

$$\xi_i\xi_i^* = 0, \quad i = 1, \cdots, l, \tag{3-20}$$

$$\alpha_i\alpha_i^* = 0, \qquad i = 1, \cdots, l \tag{3-21}$$

Pode-se extrair algumas conclusões úteis:

- Da condição (3-21), nunca existirá um conjunto de variáveis duais $\alpha_i$ e $\alpha_i^*$ que sejam ambas não-nulas, pois isto implicaria em uma movimentação não nula da função de otimização em ambas as direções.
- Apenas exemplos $(x_i, y_i)$ com o correspondente $\alpha_i = C$, $\alpha_i^* = C$ estão fora do tubo $\varepsilon$-insensível por causa das condições (3-18) e (3-19).
- Das condições (3-16) e (3-17), as variáveis duais serão não-nulas somente quando $\mid f(x_i) - y_i \mid \geq \varepsilon$, ou seja, apenas quando as amostras estiverem fora do tubo $\varepsilon$-insensível. Em outras palavras, para $\mid f(x_i) - y_i \mid < \varepsilon$, o segundo fator é não-nulo e, portanto $\alpha_i, \alpha_i^*$ se anulam para satisfazer as condições complementares de KKT. Logo, consegue-se uma representação esparsa de $w$ em termos de $x_i$ já que não será preciso todos os $x_i$ a fim de descrever $w$, só serão considerados os pontos para os quais $\alpha_i, \alpha_i^* > 0$. Tais pontos são chamados de **vetores suporte**.

**Observação:** Quanto menos ruído os dados tiverem, mais esparsa será a representação do problema, uma vez que mais dados estarão na região $\mid f(x_i) - y_i \mid < \varepsilon$. Assim, a representação esparsa dos dados se deve ao uso da função de custo especial $\varepsilon$-insensível, a qual também introduz um viés sistemático desde que, se os valores de $\varepsilon$ são muito grandes, há uma tendência de falta de ajuste ( "*underfitting* "), por exemplo, no caso extremo de $\varepsilon$ muito grande a regressão resultante será uma constante. Entretanto, outra função de custo poderia ter sido usada sacrificando a dispersão dos dados, como por exemplo, a função de custo robusta de Hubber que tem a vantagem de não introduzir um viés adicional ao contrário da $\varepsilon$-insensível .

## 3.4 Como obter b

Nota-se que $w$ é explicitamente determinado pelo procedimento de treinamento, ao contrário de $b$. No entanto, usando as condições complementares de KKT pode-se determinar $b$. Primeiramente, observa-se das condições (3-18) e (3-19) que $\xi_i = \xi_i^* = 0$ se $\alpha_i, \alpha_i^* < C$. Logo, pode-se simplesmente tomar todos os pontos de treinamento para os quais $0 < \alpha_i, \alpha_i^* < C$ e usar as condições (3-16) e (3-17) para obter $b$ da seguinte maneira:

$$b = y_i - \langle w, x_i \rangle - \varepsilon \quad \text{para } 0 < \alpha_i < C$$
$$b = y_i - \langle w, x_i \rangle + \varepsilon \quad \text{para } 0 < \alpha_i^* < C$$

Um único $x_i$ seria suficiente, no entanto para propósitos de estabilidade, é recomendável tomar uma média sobre todos os pontos $x_i$. Para isso, considera-se $\delta_i = \varepsilon * sinal(\alpha_i - \alpha_i^*)$ onde $\varepsilon$ é o erro de predição dado por $f(x_i) - y_i$ e $b$ é calculado como sendo a média em todos os $i = 1, ..., l$ de $\{\delta_i + y_i - < w, x_i >\}$.

## 3.5 Os Núcleos

Até o momento foram considerados modelos lineares no espaço de entrada. Representar $f$ como um modelo linear pode ser uma aproximação mais conveniente e até necessária em alguns casos, visto que o modelo linear é fácil de interpretar além de ser a aproximação de 1 $^a$ ordem de Taylor para $f$ e ainda, nos casos em que se tem poucos dados, o modelo linear pode ser uma alternativa para ajustar os dados sem excesso de ajuste (*"overfitting"*). No entanto, aplicações complexas do mundo real requerem um espaço de hipóteses mais expressivo do que funções lineares. Muitas vezes, ao invés de uma combinação linear simples dos atributos dados, características mais abstratas dos mesmos precisam ser exploradas e as representações por núcleos oferecem uma solução alternativa.

Com o objetivo de aprender relações não-lineares com uma máquina linear, precisa-se selecionar um conjunto de características não-lineares e reescrever os dados nesta nova representação, o que é equivalente a aplicar um mapeamento não-linear fixo dos dados para um espaço característico de alta dimensionalidade onde uma máquina linear possa ser usada. Assim, pode-se construir máquinas não-lineares em dois passos: um mapeamento não-linear fixo transforma os dados em um espaço característico com alta dimensionalidade - o uso de máquinas lineares na representação dual torna possível realizar este passo implicitamente. Em seguida, uma regressão linear neste novo espaço é realizada.

A idéia básica em SVR consiste em mapear os dados de entrada $x \in \mathbb{R}^n$

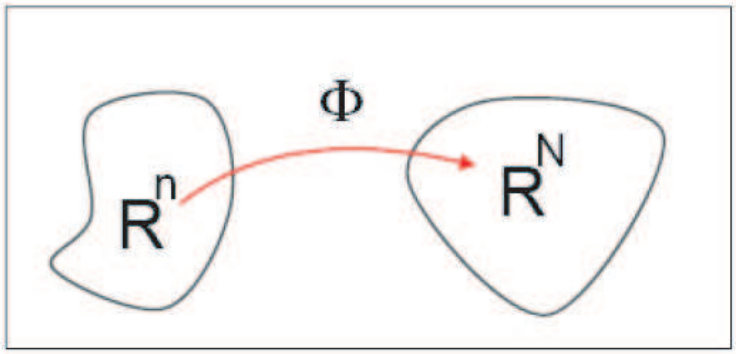


Figura 3.6: O mapeamento dos dados de entrada do espaço de dimensão $R^n$ para um espaço característico com alta dimensionalidade $R^N$, onde uma regressão linear é efetuada (Smola, 1996 (22)).

em um espaço característico com alta dimensionalidade $\mathcal{F}$ via um mapeamento não-linear $\phi : \mathbb{R}^n \to \mathcal{F}$ aumentando o poder de generalização da máquina de aprendizagem e, aplicar uma regressão linear neste novo espaço. (Veja figura 3.6, onde $\mathcal{F} = \mathbb{R}^N$).

**Definição 3.1 (Função núcleo)** *Uma função núcleo é uma função $K$ tal que para todo $x, y \in \mathcal{X}$ tem-se $K(x,y) = \langle \phi(x), \phi(y) \rangle$ onde $\phi$ representa o mapeamento de $\mathcal{X}$ para um (produto interno) espaço característico $\mathcal{F}$.*

Ao substituir $< x_i, x_j >$ por $K(x_i, x_j)$ no algoritmo de vetores de suporte 3-14, todas as considerações anteriores serão válidas desde que ainda será calculada uma regressão linear, porém em um espaço de dimensão infinita e o tempo computacional necessário será praticamente o mesmo. O problema dual de otimização será dado da maneira a seguir:

$$\begin{array}{ll} maximize & \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i, x_j) \\ -\varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) \end{cases} \\ \text{sujeito a} & \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \quad e \quad \alpha_i, \alpha_i^* \in [0, C] \end{array} \tag{3-22}$$

e a função estimada será obtida do seguinte modo:

$$w = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\phi(x_i) \quad \text{e} \quad f(x) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)K(x_i, x) + b. \tag{3-23}$$

Ao introduzir a idéia de função núcleo no problema, $w$ não será mais descrito explicitamente.

Um dos pontos marcantes da teoria do SVR é a não necessidade de conhecer a função $\phi$ explicitamente, pois através da representação dual do problema de otimização, a função decisão pode ser expressa usando o produto interno entre os pontos de treinamento e os pontos de teste $f(x) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\langle\phi(x_i), \phi(x)\rangle + b$. Ademais, existe uma maneira de calcular diretamente o produto interno $\langle\phi(x_i), \phi(x)\rangle$ no espaço característico como uma função dos valores de entrada. Este método de cálculo direto é chamado de função núcleo, cuja definição foi dada anteriormente.

A vantagem de escrever o problema de otimização na forma dual ocorre, pois, nessa reformulação o número de parâmetros independe do número de atributos e sim do tamanho da amostra. A partir da escolha de uma função núcleo, pode-se realizar um mapeamento não-linear implicitamente para um espaço característico de grande dimensionalidade sem aumentar o número de parâmetros ajustáveis, pois a função núcleo calcula o produto interno de dois vetores característicos correspondentes a duas entradas. Logo, defini-se a função núcleo diretamente e, implicitamente, o espaço característico.



### 3.5.1
### Propriedades das funções núcleo

Pela definição de função núcleo consegue-se uma função que retorna o produto interno entre as imagens de dois valores de entrada em um espaço característico. Assim, as propriedades que podem ser atribuídas às funções núcleo são aquelas que satisfazem as do produto interno.

Funções núcleo podem ser consideradas como produtos internos generalizados. Pode-se mostrar que qualquer produto interno é uma função núcleo, mas, a propriedade de linearidade do produto interno não ocorre em funções núcleo. No entanto, algumas propriedades de produto interno que possuem uma generalização para funções núcleo serão verificadas.

– Simetria:
$K(x, y) = \langle\phi(x), \phi(y)\rangle = \langle\phi(y), \phi(x)\rangle = K(y, x)$

– Desigualdade de Cauchy-Schwarz:
$$\begin{aligned} K(x, y)^2 &= \langle\phi(x), \phi(y)\rangle^2 \leq \| \phi(x) \|^2 \| \phi(y) \|^2 \\ &= \langle\phi(x), \phi(x)\rangle\langle\phi(y), \phi(y)\rangle = K(x, x)K(y, y) \end{aligned}$$

O núcleo $K(x, y)$ pode ainda ter uma interpretação de medição de distância no espaço de entrada entre os exemplos $x$ e $y$ conforme estudos de Schölkopf (17).

$$d^2(x,y) = (\phi(x) - \phi(y))^2 = K(x,x) - 2K(x,y) + K(y,y).$$

É preciso definir quais funções $K(x,y)$ corresponderão a um produto interno em algum espaço característico $\mathcal{F}$. O teorema de Mercer caracteriza essas funções.

**Teorema 3.2 (Mercer)** *Existe um mapeamento $\phi$ e uma expansão da forma:*

$$K(x,y) = \langle \phi(x), \phi(y) \rangle = \sum_i \phi(x)_i \phi(y)_i \tag{3-24}$$

*se e somente se, $\forall g(x)$ tal que*

$$\int g(x)^2 dx \quad \textit{é finito} \tag{3-25}$$

$$\textit{implica} \int K(x,y)g(x)g(y)dxdy \geq 0 \tag{3-26}$$



Em muitos casos não deve ser fácil satisfazer a condição de Mercer. No entanto, a condição (3-26) deve valer para toda $g(x)$ com norma $L_2$, ou seja, que satisfaça a equação (3-25).

A condição de Mercer informa se um determinado núcleo constitui ou não um produto interno em algum espaço característico $\mathcal{F}$, embora não mostre como construir a função $\Phi$ e até mesmo que espaço característico teremos.

Alguns núcleos usados em SVR para análise de séries temporais são estudados por Stefan Rüping (24). Exemplos de núcleos:

– Função Núcleo linear:

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

Esta é a função núcleo mais simples. A função decisão toma a forma $f(x) = \langle w, x \rangle + b$. O núcleo linear utilizado para prever séries temporais, isto é, $x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, ..., x_{t-k})$ tem como modelo resultante um modelo auto-regressivo de ordem k (AR(k)).

– Função Núcleo polinomial:

$$K(x,y) = \langle x, y \rangle^d.$$
$$K(x,y) = (\langle x, y \rangle + 1)^d.$$

– Função Núcleo RBF: (Função Base Radial): São funções núcleo que podem ser escritas na forma $K(x, y) = f(d(x, y))$, onde d é uma métrica em $\mathcal{X}$ e $f$ é uma função em $\mathbb{R}^+$. Usualmente a métrica vem do produto interno $d(x, y) =\| x - y \|= \sqrt{\langle x - y, x - y\rangle}$.Como exemplo de função base radial, podemos definir o núcleo função base radial Gaussiano:

$$K(x, y) = \exp(-\gamma \| x - y \|^2).$$

Neste caso, a similaridade entre dois exemplos é julgada pela distância euclidiana.

– Função Núcleo de Fourier :

$$K_F(x, y) = \frac{1-q^2}{2(1-2q\cos(x-y))+q^2}$$

Uma transformação que pode ser feita em dados de séries temporais é a transformada de Fourier. Tal representação é útil se a informação das séries temporais não recaem em valores individuais em cada ponto de tempo, porém na freqüência de alguns eventos. O produto interno da expansão de Fourier de duas séries temporais pode ser calculado diretamente pela expansão de Fourier regularizada conforme descrito por Vapnik em (29) com o resultado do cálculo mostrado acima.

Pode-se ainda criar núcleos mais complexos a partir de outros mais simples. As demonstrações serão omitidas podendo ser encontradas em (7).

**Proposição 3.3** *Sejam $K_1$ e $K_2$ funções núcleo em $X \times X$ , $X \subseteq \mathbb{R}^n$, $a \in \mathbb{R}^+$ e $f$ é uma função de valor real em $X$,$\Phi : X \to \mathbb{R}^m$, com o núcleo $K_3$ definido em $\mathbb{R}^m \times \mathbb{R}^m$ e $\mathbf{B}$ uma matriz $n\times n$ simétrica (semi)definida positiva. Então as seguintes funções são funções núcleo:*

1. $K(x, z) = K_1(x, z) + K_2(x, z)$,
2. $K(x, z) = aK_1(x, z)$,
3. $K(x, z) = K_1(x, z)K_2(x, z)$,
4. $K(x, z) = f(x)f(z)$,
5. $K(x, z) = K_3(\phi(x), \phi(z))$,
6. $K(x, z) = x\prime\mathbf{B}z$.

***Corolário 3.4*** *Dado $K_1$ uma função núcleo em $X \times X$ , $x, z \in X$ e $p(x)$ um polinômio com coeficientes positivos. Então as seguintes funções também são funções núcleo:*

1. $K(x,z) = p(K_1(x,z))$,
2. $K(x,z) = \exp(K(x,z))$,
3. $K(x,z) = \exp(-\gamma||x-z||^2)$.

Para maiores informações sobre núcleos, consulte Schölkopf e Smola (18).

## 3.6 Arquitetura de um SVR

A figura 3.7 descreve os passos do SVR. A entrada $x$ (para a qual uma predição deve ser feita) e os vetores suporte $x_i, i = 1, ...n$ são mapeados não linearmente pela função $\Phi$ para um espaço característico, onde os produtos internos são calculados. Isto corresponde ao cálculo de funções núcleo $K(x, x_i)$. Finalmente, os resultados são combinados através dos pesos $\upsilon_i = (\alpha_i^* - \alpha_i)$ e somados com o termo constante $b$ para fornecer a previsão de $x$.

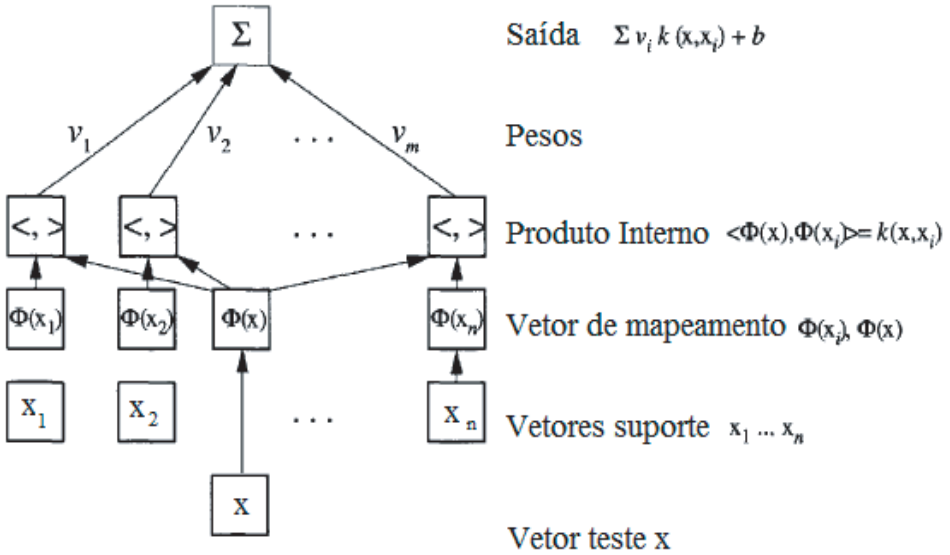


Figura 3.7: Arquitetura de máquina de vetor suporte (Schölkopf, Smola, 2001 (18)).

## 3.7
## O SVR no software R

O R (16) é um software livre usado para análise estatística de dados, oferecendo grande variedade de técnicas estatísticas (modelos lineares e não-lineares, modelos de séries temporais, testes estatísticos clássicos, entre outros) além de produzir gráficos de qualidade. O programa está disponível em http://www.r-project.org , onde também encontram-se textos e tutoriais, além do código fonte para compilação e os executáveis já compilados para diferentes sistemas operacionais.

Utiliza-se a implementação LIBSVM de Chang e Lin (4) empregando o software do R-project (16), após a instalação do pacote e1071 para estimar modelos SVR. Este pacote pode ser usado tanto para classificação quanto para regressão. A matriz de dados $\mathbf{X}$ e um vetor de respostas $\mathbf{y}$ são definidos e, dependendo do tipo de vetor $\mathbf{y}$ definido, o tipo padrão a ser empregado é uma $C$-classificação ou uma $\varepsilon$-regressão. Também é necessário definir a função núcleo a ser utilizada. Neste estudo, utiliza-se a função núcleo linear e a função núcleo Gaussiano (default).

Deve-se ainda definir os parâmetros livres do núcleo (para o núcleo Gaussiano, o valor default: $\gamma$=1/(dimensão dos dados)), o valor do $\varepsilon$ da função de custo $\varepsilon$-insensível linear (default: 0.1) e o valor da constante $C$ (default: 1).

Nesse pacote, pode-se ainda utilizar a função *tune* que ajusta os hiperparâmetros do SVR fazendo uma pesquisa em um domínio retangular com os valores da busca fornecidos pelo usuário. Essa função usa como avaliação da performance para regressão, o erro quadrado médio.