4
Aproximação de Funções

Com o objetivo de modelar os resíduos obtidos pelo método *Chain Ladder*, serão utilizadas nesta dissertação algumas técnicas para aproximação de funções.

As técnicas para aproximação de funções são muito estudadas, principalmente os métodos aplicáveis a um grande conjunto de dados e em espaços de altas dimensões. No geral, um problema de aproximação de funções se resume em escolher uma função dentre um conjunto de funções "bem-definidas" que melhor se aproxime da função desejada.

Pode-se destacar duas classes de problemas de aproximação de funções. São elas:

- aproximação para funções "objetivo" conhecidas. Essa classe é contemplada por um ramo da análise numérica que estuda como determinadas funções conhecidas, porém complexas, podem ser aproximadas por uma classe específica de funções como os polinômios, por exemplo, que têm propriedades desejáveis;
- aproximação para funções "objetivo" desconhecidas. Nesta classe, ao invés de uma fórmula específica, o que se conhece sobre uma função desconhecida g é um conjunto de pontos $(x, g(x))$. Dependendo do domínio e do contra-domínio de g, várias técnicas podem ser aplicadas para aproximar os valores de $g(x)$, por exemplo: interpolação, extrapolação, análise de regressão, classificação, redes neurais, lógica *fuzzy*, entre outras.

As aproximações por classificação e regressão recebem um tratamento unificado e diferenciado por meio da teoria da aprendizagem estatística, onde são vistos como problemas de aprendizado supervisionado. Quando as saídas são valores em um espaço discreto trata-se o problema de classificação, quando as saídas pertencem a um espaço contínuo trata-se de regressão. Este último é o caso utilizado nesta dissertação.

A teoria de aprendizado estatístico busca extrair tendências e padrões importantes dos dados disponíveis, estabelecer uma relação entre eles e entender o que eles representam. Destacam-se aqui duas técnicas que vêm recebendo

crescente atenção dos pesquisadores nos últimos anos. São elas: as Máquinas de Suporte Vetorial (SVM's do inglês *Support Vector Machines*) e os Processos Gaussianos (GP's do inglês *Gaussian Process*).

As SVM's constituem uma técnica de aprendizado supervisionado não-paramétrico, aplicada na classificação e regressão de alto desempenho. Esta técnica está embasada na teoria de aprendizagem estatística, desenvolvida por Vapnik em (31).

Os GP's são uma abordagem Bayesiana de aprendizado supervisionado e também não paramétrico. Williams e Rasmussen em (26) observaram a dificuldade de análise Bayesiana de redes neurais devido a complexidade das distribuições *a priori* sobre as funções decorrentes. Ao invés disso, se os processos gaussianos são usados como *a priori* sobre as funções, a análise Bayesiana pode ser realizada normalmente.

4.1 Um Breve Comentário sobre Aprendizagem Estatística

Técnicas de aprendizagem estatística estão baseadas no princípio de inferência indutiva, pelo qual é possível se obter informações genéricas sobre o todo de um conjunto, a partir de um conjunto particular de dados observados. O aprendizado por indução pode ser classificado em: supervisionado ou não-supervisonado. No caso desta dissertação será abordada a aprendizagem supervisionada, cujo o objetivo principal é construir modelos para aproximar funções, ou seja, "aprender" a representação de uma função.

A teoria descrita por Vapnik em (31) tem como objetivo encontrar o valor de uma função para qualquer entrada válida. Na aprendizagem supervisionada, isso é feito utilizando dados de treinamento, isto é, um conjunto de exemplos (dados) representado na forma: entrada - saída desejada. Esses exemplos fornecem a relação geral entre os dados de entrada e saída e a função de um algoritmo de aprendizagem de máquina é extrair uma representação para essa relação. O fundamental desse processo é que a representação gerada tenha capacidade de produzir saídas corretas para entradas que não foram apresentadas previamente.

A função solução, a qual fornece valores para as novas entradas, é escolhida dentre um conjunto de funções candidatas que mapeiam o espaço de entrada no domínio de saída por meio do treinamento. Um conjunto formado pelas funções candidatas escolhidas, denominado *espaço de hipóteses*, é escolhido antes de se "aprender" a função correta. O processo que seleciona uma *hipótese* que melhor relaciona os dados é chamado de algoritmo de

aprendizagem.

Para se resolver um problema de aprendizagem supervisonada é necessário:

1. Obter dados representativos do que se quer como resposta, ou seja, dados para o treinamento.

2. Especificar como os dados de entrada são representados. Uma representação usual um vetor pertencente a um espaço vetorial munido de produto interno, o qual denomina-se *espaço característico*.

3. Definir a estrutura da função que será "aprendida" e o correspondente algoritmo de aprendizagem.

4. Aplicar o algoritmo de aprendizagem no conjunto de dados de treinamento. Para otimizar o desempenho do algoritmo é possível escolher e ajustar parâmetros por meio de testes aplicados a um subconjunto do conjunto de treinamento. Definidos os parâmetros, o desempenho do algoritmo pode ser avalidado em um conjunto de teste, que é um conjunto formado pelos dados iniciais menos do conjunto de treinamento.

4.2
Máquinas de Suporte Vetorial

Inicialmente as Máquinas de Suporte Vetorial foram desenvolvidas para construir hiperplanos separadores em problemas de reconhecimento de padrões. Posteriormente este método foi estendido para construir funções separadoras não-lineares, mas lineares em um espaço característico e também estimar funções de valores reais (regressão) (31). Muitas técnicas de reconhecimento de padrões são fundamentadas na minimização do risco empírico, isto é, buscam otimizar o desempenho do algoritmo sobre o conjunto de treinamento. Já as SVM's se baseiam na minimização do risco funcional, ou seja, procuram minimizar os erros cometidos no reconhecimento de padrões que possuem uma distribuição de probabilidade de dados desconhecida.

As SVM's, constituem um sistema de aprendizado treinado com algoritmo de otimização, onde os vetores do espaço de entrada são mapeados de forma não-linear para um espaço característico de alta dimensionalidade, por meio de um algoritmo escolhido. Nesse espaço uma superfície de decisão linear, que constitui um hiperplano de separação ótima de exemplos ou dados é construída, conforme Figura 4.1. A teoria da otimização, sobre a qual está baseado o treinamento, fornece as técnicas matemáticas necessárias para a formação de tais hiperplanos. A classificação de vetores de suporte tem o objetivo de

criar uma maneira computacionalmente eficiente para maximizar as margens de separação entre os dados, ou seja, que o algoritmo seja capaz de lidar com diferentes tamanhos de amostras e "aprender", em um espaço característico de alta dimensionalidade, hiperplanos de separação ótima, isto é, que tenham grande habilidade de generalização. A teoria da generalização permite controlar e prevenir problemas com excessos de ajustes ("overfitting") por meio do monitoramento de medidas das margens definidas por esses hiperplanos. Para maiores detalhes vide Vapnik em (31) e Smola e Schölkopft em (29).

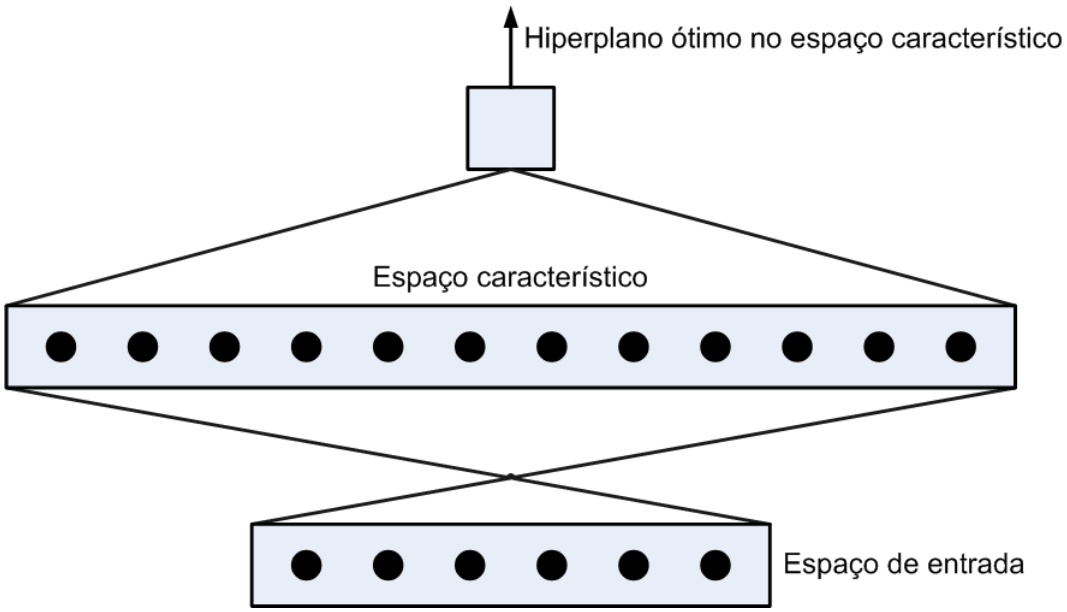

Figura 4.1: Máquina de suporte vetorial. (31)

Para exemplificar, é mencionado aqui o caso mais simples de um modelo SVM, um classificador binário linear com margens rígidas, que possuem classes ou rótulos positivos e negativos. Esses rótulos definem fronteiras lineares somente a partir de dados linearmente separáveis. Esse fato o torna pouco aplicável em problemas reais. No entanto foi a partir dessa formulação que se obteve a extensão paras definir fronteiras lineares sobre conjuntos de dados não-separáveis.

Exemplo 4.2.1 *Considere um conjunto de treinamento* $\chi \subset \mathbb{R}^n \times \mathbb{R}$ *que possua pontos de duas classes. Uma SVM separa estas classes por meio de um hiperplano que é determinado por alguns pontos de* χ*, denominados vetores suporte. No caso separável, este hiperplano maximiza a margem e todos os vetores suporte caem a uma mesma distância a partir do hiperplano. Nos casos não-separáveis, tanto o hiperplano, quanto os vetores são obtidos pela solução de um problema de otimização com restrições, permitindo a ocorrência de erros, porém admitindo uma certa penalização. Vide Figura 4.2.*

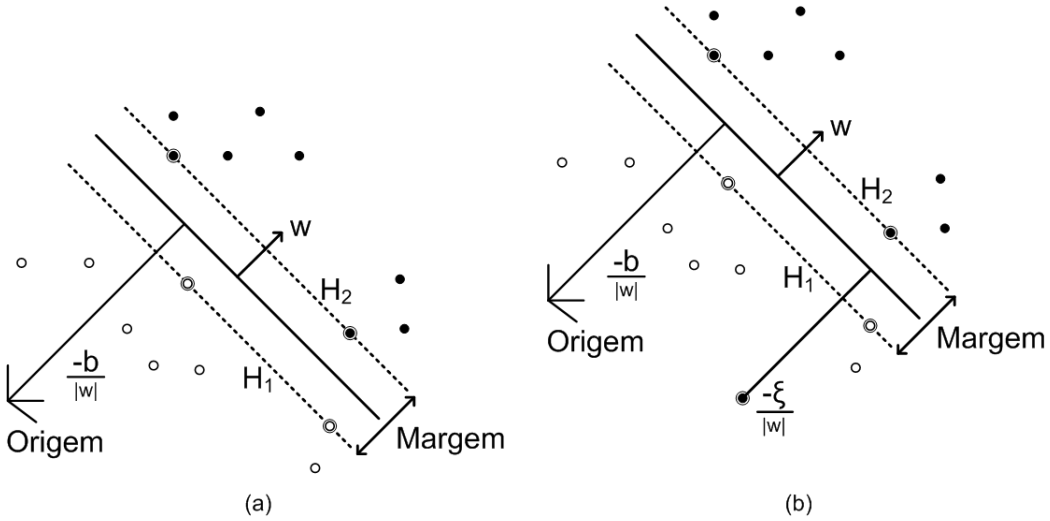

Figura 4.2: Hiperplano linear de separação ótima para (a) dados linearmente separáveis e (b) dados não separáveis e os respectivos vetores suporte (circulados). (3)

As SVM's têm apresentado um bom desempenho em algumas aplicações como, por exemplo, reconhecimento de imagens, problemas de regressão e séries temporais, e por esse motivo constituem uma área de pesquisa ativa. Além disso, têm como uma vantagem a convexidade do problema de otimização definido em seu treinamento e o fato de não apresentarem mínimos locais.

4.2.1 Máquina de Suporte Vetorial para Regressão

Máquinas de Suporte Vetorial para Regressão, ou SVR's (do inglês *Support Vector Regression*), são uma nova metodologia não-paramétrica capaz de lidar com problemas de predição ou regressão de funções. Elas formam um algoritmo de aprendizagem supervisionada baseado no princípio da indução para ajuste de dados multidimensionais e são caracterizadas pelo uso de "kernels" ou funções núcleos, pela ausência de mínimos locais e pela capacidade de controlar as margens de separação e o número de vetores suporte. Essa metodologia possibilita o uso de vários tipos de funções núcleos e por esse motivo podem ajustar funções altamente não-lineares.

O algoritmo de uma SVR é construído por meio do treinamento de um conjunto de amostras (dados observados) $(\mathbf{x}, y)$ onde $\mathbf{x} \in \mathbb{R}^n$ *e* $y \in \mathbb{R}$. Assume-se que os dados são identicamente e independentemente distribuídos e que existe alguma distribuição de probabilidade não-conhecida $P(\mathbf{x}, y)$ de onde os dados são observados. O objetivo aqui é encontrar uma função f que aproxime o mapeamento dos dados de entrada para os dados de saída, ou seja, que relacione $\mathbf{x} \rightarrow f(\mathbf{x})$.

Funções de Perda e Risco Empírico

Para explorar dados de treinamento que apresentam ruídos (erros) é necessário encontrar funções de perda (também conhecidas como funções de custo) que estabeleçam certas penalidades para erros. Uma função de perda determina a importância da precisão na separação de dados e pode ser considerada como o custo decorrido de erros obtidos na previsão de uma certa variável. Em grande parte dos problemas, ela será do tipo $c(\mathbf{x}, y, f(\mathbf{x})) = c(f(\mathbf{x}) - y)$.

Em problemas que utilizam SVR, o objetivo é encontrar a função de perda f que minimiza o risco esperado, denominado risco funcional, em relação aos dados empíricos (treinamento), dado pela Equação 4-1.

$$R[f] = \int c(\mathbf{x}, y, f(\mathbf{x}))dP(\mathbf{x}, y), \tag{4-1}$$

onde $c(\mathbf{x}, y, f(\mathbf{x}))$ corresponde a uma função de perda determinando a penalidade para os erros de previsão e $P(\mathbf{x}, y)$ é um distribuição de probabilidade não-conhecida.

Visto que $P(\mathbf{x}, y)$ é desconhecida, a avaliação do risco funcional torna-se desnecessária e os dados empíricos serão usados apenas na estimação de uma função f' que de alguma forma se "aproxima" daquela que realmente minimiza $R[f]$.

Substituindo a integral da Equação 4-1 por uma estimativa empírica, produz uma aproximação possível para o risco funcional, denominada de risco funcional empírico, dado pela Equação 4-2:

$$R_{emp}[f] := \frac{1}{l}\sum_{i=1}^{l} c(\mathbf{x_i}, y_i, f(\mathbf{x_i})). \tag{4-2}$$

Escrito dessa forma, o risco é definido apenas pela função de perda, o que significa uma maneira de medir do erro médio no conjunto de dados empíricos, além de que o termo da distribuição de probabilidade desaparece.

A questão agora é minimizar o risco funcional empírico. Uma primeira tentativa seria a de encontrar um minimizador de riscos empíricos $f_0 := argmin_{f \in H} R_{emp}[f]$ para alguma classe funções H. O minimizador f_0 só faz sentido somente se $\lim_{l\to\infty} R_{emp}[f] = R[f]$, o que é validado pela Lei dos Grandes Números. Entretanto, ainda há um dificultador quando H é uma classe muito rica, o que ocorre quando se lida com poucos dados em espaços de alta-dimensão; isso pode levar a "overfitting", ou seja, pequenos erros no treinamento, além de propriedades de generalização ruins. Isto significa que pode-se obter uma função $f \in H$ que gere ótimas previsões para os valores

de y_i no treinamento, porém que não garanta um bom desempenho nos dados quando avaliadas nos dados de teste.

Para controlar essa dificuldade, um termo de regularização é adicionado ao risco empírico de forma que complexidade da classe de funções H esteja limitada, constituindo assim a classe de risco funcional regularizado, representado na Equação 4-3. No caso, de vetores suporte, considera-se como termo de regularização, a norma de uma vetor $\mathbf{w}$, definido na seção seguinte, dividida por dois, ou seja,$\frac{1}{2}\|\mathbf{w}\|^2$.

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

onde $\lambda > 0$ é chamada *constante de regularização* que pondera os termos da equação.

Minimizar o risco regularizado significa que objetivando conseguir um risco pequeno, torna-se necessário controlar tanto o erro de treinamento quanto a complexidade do modelo, buscando explicar os dados com uma modelagem simples. A seleção da função de perda deve ser feita cuidadosamente de forma que a escolha seja a que melhor represente as incertezas do problema de otimização proposto. A Figura 4.3 traz alguns modelos de funções de perda.

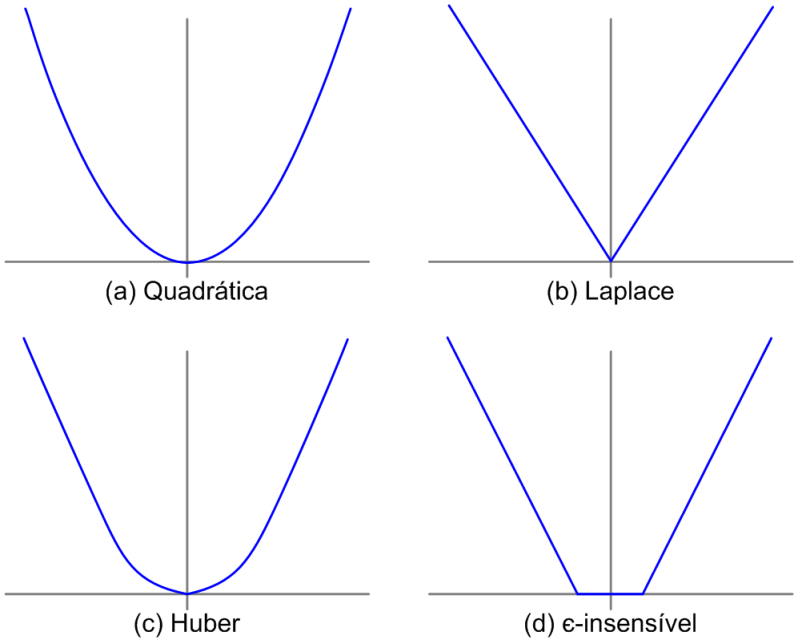

Figura 4.3: Modelos de Funções de Perda.(14)

Na Figura 4.3, (a) corresponde ao critério de erros mínimos quadrados. A função de perda (b) é uma função de perda Laplaciano que é menos sensível a "outliers" do que a função de perda quadrática. Huber propôs a função de perda (c) como uma função de perda robusta que tem propriedades ideais

quando a distribuição dos dados é desconhecida. Estas três funções de perda não irão produzir vetores de suporte esparsos. Para resolver esse problema Vapnik propôs a função de perda (d) como uma aproximação para a função de Huber que permite que seja obtido um conjunto de vetores de suporte esparsos.

Formulando um Problema de Otimização

Em uma regressão ε-SVR, dado um conjunto de dados de treinamento $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_l, y_l)\} \subset \mathbb{R}^n \times \mathbb{R}$, o objetivo é encontrar uma função $f(\mathbf{x})$ tal que, para todos os dados de treinamento, cada um dos valores $(\mathbf{x_i})_{i=1,\cdots,l}$, tenha no máximo um desvio ε dos valores alvo y_i e que, ao mesmo tempo, f seja a mais "plana" possível. Em outras palavras, erros menores que ε são tolerados, mas será penalizado qualquer desvio maior do que ε, isto é, $|f(\mathbf{x_i}) - y_i| \geq \varepsilon, \ \forall \ i = 1, \cdots, l$. Inicialmente, a função a ser obtida é linear e portanto é descrita pela forma dada pela Equação 4-4.

$$f(x) =< \mathbf{w}, \mathbf{x} > +b \ \ com \ \ \mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}, \tag{4-4}$$

onde $<,>$ representa o produto interno em $\mathbb{R}^n$. Para que $f(\mathbf{x})$ seja o mais plana possível é necessário que $\mathbf{w}$ tem o menor comprimento possível. Uma forma de se obter isso é minimizar a norma de $\mathbf{w}$, isto é, $\|\mathbf{w}\|^2 =< \mathbf{w}, \mathbf{w} >$.

Levando-se em conta essas considerações, a busca por um função f se transforma em um problema de otimização convexa:

$$\begin{aligned} & & minimize \ \ \frac{\|\mathbf{w}\|^2}{2} \\ sujeito \ \ a & \begin{cases} y_i - < \mathbf{w}, \mathbf{x_i} > -b \leq \varepsilon, & i = 1, \cdots, l \\ < \mathbf{w}, \mathbf{x_i} > +b - y_i \leq \varepsilon, & i = 1, \cdots, l \end{cases} \end{aligned} \tag{4-5}$$

Da maneira como foi formulado, o Problema 4-5 assume que uma função f que aproxima todos os pares $(\mathbf{x_i}, y_i)$ com precisão ε existe, e que portanto o problema de otimização é possível. Muitas das vezes, isso pode não ocorrer ou ainda, pode-se desejar que haja a ocorrência de um número mínimo de erros na separação dos dados de treinamento. Dessa forma torna-se necessário criar margens flexíveis de erro ("soft margin") utilizando algum tipo de função de perda que introduz variáveis de folga não-negativas, ξ_i, ξ_i^*, que consideram pontos situados fora da margem $|f(\mathbf{x_i}) - y_i|$. Essa funções, denominadas ε-insensível, estabelecem penalidades para a aceitação de pontos fora da margem definida e estabelece restrições que seriam inviáveis para o problema de otimização anterior. Sob essas condições obtém-se a seguinte formulação para o problema de otimização:

$$minimize \quad \frac{\|\mathbf{w}\|^2}{2} + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)$$

$$sujeito \quad a \begin{cases} y_i - < \mathbf{w}, \mathbf{x_i} > -b \leq \varepsilon + \xi_i, & i = 1, \cdots, l \\ < \mathbf{w}, \mathbf{x_i} > +b - y_i \leq \varepsilon + \xi_i^*, & i = 1, \cdots, l \\ \xi_i, \xi_i^* \geq 0 & i = 1, \cdots, l \end{cases} \tag{4-6}$$

As variáveis de folga ξ_i e ξ_i^* se localizam em lados opostos, a uma distância ε dos valores objetivo, conforme Figura 4.4. A constante C é pré-estabelecida e determina o intercâmbio entre a "planicidade" de f e a quantidade de desvios maiores que ε que serão aceitos, ou seja, pondera os dois termos da função de perda. Os valores $\mathbf{w}$ e b são as soluções do problema definido em 4-6. Essa formulação descreve um problema de otimização quadrática, visto que, que possui restrições que são desigualdades lineares e sua função objetivo é quadrática. Ademais, ele é convexo logo, tem solução única.

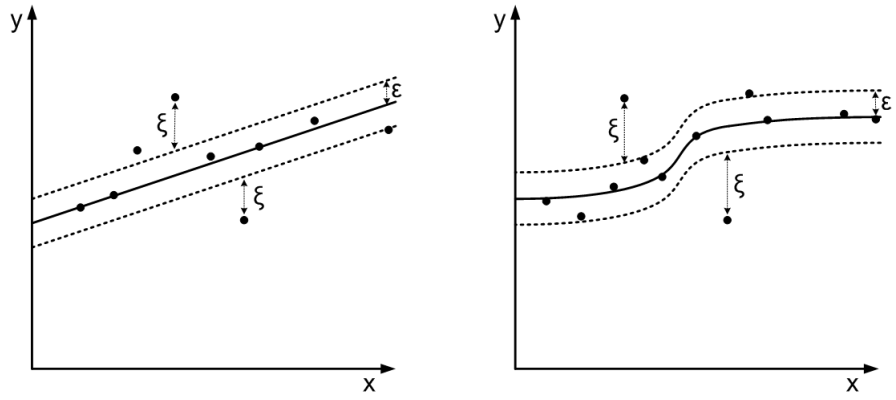

Figura 4.4: A esquerda estão representadas variáveis de folga para regressão linear unidimensional com função ε-insensível. A direita uma situação análoga para o caso não linear.

O problema de otimização descrito pelas Equações 4-6 corresponde a lidar com a função de perda ε-insensível, $|\xi|_\varepsilon = |f(\mathbf{x}) - y|$, descritas pela Expressão 4-7:

$$|\xi|_\varepsilon = \begin{cases} 0 & \text{se } |\xi| \leq \varepsilon, \\ |\xi| - \varepsilon & \text{caso contrário.} \end{cases} \tag{4-7}$$

A Figura 4.5 descreve essa situação graficamente. A precisão ε é pré-especificada, gerando uma região de raio ε em torno dos dados. As variáveis de folga ξ_i e ξ_i^* possibilitam um padronização correta para dados que encontram ligeiramente fora da região. Somente os pontos situados fora dessa região contribuem para a perda, desde que, os desvios sejam penalizados de forma linear.

A minimização da Função 4-7 determina o intercâmbio entre a complexidade do modelo e esses tais pontos.

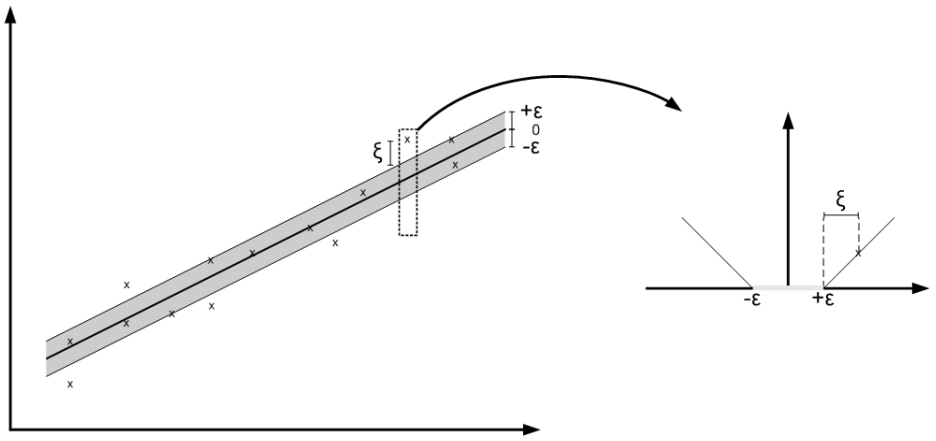

Figura 4.5: A esquerda está representado um hiperplano separador linear e a direita a respectiva função de perda ε-insensível. (29)

A Figura 4.6 mostra uma função de perda ε-insensível considerada na forma quadrática, $|\xi|_{\varepsilon quad} = |f(\mathbf{x}) - y|^2$. Neste caso, o problema de otimização é definido pela formulação 4-8:

$$minimize \quad \frac{\|\mathbf{w}\|^2}{2} + C\sum_{i=1}^{l}(\xi_i^2 + \xi_i^{*2})$$
$$sujeito \quad a \begin{cases} y_i - < \mathbf{w}, \mathbf{x_i} > -b \leq \varepsilon + \xi_i, & i = 1, \cdots, l \\ < \mathbf{w}, \mathbf{x_i} > +b - y_i \leq \varepsilon + \xi_i^*, & i = 1, \cdots, l \\ \xi_i, \xi_i^* \geq 0 & i = 1, \cdots, l \end{cases} \tag{4-8}$$

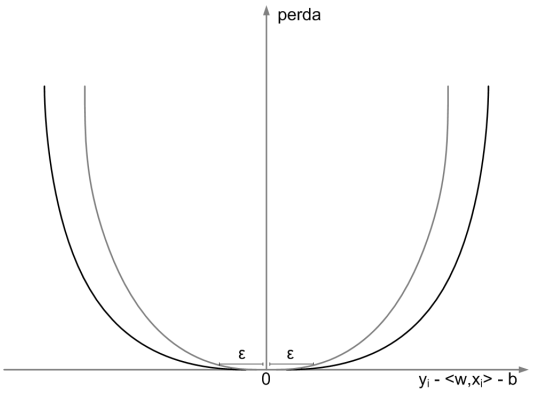

Figura 4.6: Função de perda ε-insensível quadrática. (5)

A escolha das funções de perda influenciam nas estratégias escolhidas para realizar a regressão. É necessário que estas funções sejam convexas para que se garanta a existência e a unicidade da solução do problema de otimização.

Formulação Dual

Um problema descrito por meio de formulações primais como em 4-6 pode ser mais facilmente resolvido se estiver na sua formulação dual. Essa formulação propicia estender a SVR para funções não-lineares.

Para a conversão da formulação primal para a dual introduz-se os multiplicadores de Lagrange ou variáveis duais para cada uma das restrições impostas pelo Problema 4-6. O questão aqui é construir uma função de Lagrange, ou Lagrangiano, a partir de uma combinação entre a função objetivo e cada uma das restrições correspondentes, por meio da introdução de um conjunto de variáveis duais. Neste caso, o Lagrangiano é dado pela Equação 4-9:

$$\begin{aligned} L = \quad & \frac{\|\mathbf{w}\|^2}{2} + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) - \sum_{i=1}^{l}(\eta_i\xi_i + \eta_i^*\xi_i^*) + \\ & - \sum_{i=1}^{l}\alpha_i(-y_i + < \mathbf{w}, \mathbf{x_i} > + b + \varepsilon + \xi_i) + \\ & - \sum_{i=1}^{l}\alpha_i^*(y_i - < \mathbf{w}, \mathbf{x_i} > - b + \varepsilon + \xi_i^*) \end{aligned} \tag{4-9}$$

onde $\eta_i, \eta_i^*, \alpha_i, \alpha_i^*$ são consideradas variáveis duais, ou multiplicadores de Lagrange e portanto, devem ser constantes positivas, ou seja, $\eta_i, \eta_i^*, \alpha_i, \alpha_i^* \geq 0$.

O problema de otimização agora consiste em minimizar o Lagrangiano em relação aos parâmetros primais $\mathbf{w}, b, \xi_i^*, \xi_i$, ou maximizá-lo em relação ao a multiplicadores de Lagrange $\eta_i, \eta_i^*, \alpha_i, \alpha_i^*$. Para minimizar a função de Lagrange, as derivadas parciais de L em relação a cada uma das variáveis primais devem ser calculadas e igualadas a zero. Formalmente:

$$\partial_b L = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0 \tag{4-10}$$

$$\partial_{\mathbf{w}} L = \mathbf{w} - \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \tag{4-11}$$

$$\partial_{\xi_i^*} L = C - \alpha_i^* - \eta_i^* = 0 \tag{4-12}$$

$$\partial_{\xi_i} L = C - \alpha_i - \eta_i = 0 \tag{4-13}$$

Substituindo as Expressões 4-10, 4-11,4-12 e 4-13 no próprio lagrangiano 4-9, se obtem o seguinte problema de otimização dual:

$$maximize \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) < \mathbf{x_i}, \mathbf{x_j} > \\ -\varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*). \end{cases}$$
$$sujeito \;\; a \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \;\; e \;\; \alpha_i, \alpha_i^* \in [0, C] \tag{4-14}$$

As condições 4-12 e 4-13 fornecem os resultados $\eta_i^* = C - \alpha_i^*$ e $\eta_i = C - \alpha_i$ respectivamente, o que provoca a eliminação das variáveis duais η_i^*, η_i. As variáveis de folga não aparecem na formulação dual. A Equação 4-10 resulta na eliminação de b e Equação 4-11 pode ser reescrita da seguinte forma:

$$\mathbf{w} = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\mathbf{x_i} \tag{4-15}$$

onde α_i, α_i^* são soluções do problema dual.

O vetor $\mathbf{w}$ pode ser escrito como combinação linear dos pontos de treinamento $\mathbf{x_i}$ e a substituição do resultado 4-15 na expressão 4-4 resulta na Equação 4-16, denominada como expansão para $f(x)$ em vetores suporte.

$$f(x) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) < \mathbf{x_i}, \mathbf{x} > + b \tag{4-16}$$

A complexidade da função representada pelos vetores de suporte (SV's) independe da dimensão do espaço $\mathbb{R}^n$, dependendo apenas do número de SV's. Vale ressaltar que para se avaliar $f(\mathbf{x})$ não é necessário calcular $\mathbf{w}$ de maneira explícita e que o algoritmo pode ser completamente formulado em termos de produto interno entre os dados. Essas observações são importantes para a formulação de uma extensão não-linear.

O Cálculo do fator b

O cálculo para encontrar o valor da constante b pode ser feito por meio da exploração das condições de *Karush-Kuhn-Tucker*, (KKT) descritas detalhadamente em [Karush de 1939, Kuhn e Tucker, 1951].

As condições de KKT para o problema de otimização tratado até aqui são as seguintes:

$$\begin{cases} \alpha_i(-y_i + < \mathbf{w}, \mathbf{x_i} > + b + \varepsilon + \xi_i) = 0 \\ \alpha_i^*(y_i - < \mathbf{w}, \mathbf{x_i} > - b + \varepsilon + \xi_i^*) = 0 \end{cases} \tag{4-17}$$

$$\begin{cases} (C - \alpha_i)\xi_i = 0 \\ (C - \alpha_i^*)\xi_i^* = 0 \end{cases} \tag{4-18}$$

$$\begin{cases} \xi_i, \xi_i^* = 0 \\ \alpha_i, \alpha_i^* = 0 \end{cases} \tag{4-19}$$

Observações importantes sobre as condições de Karush-Kuhn-Tucker:

1. Somente as amostras $(\mathbf{x_i}, y_i)$ com $\alpha_i = C$ e $\alpha_i^* = C$ estão fora da região ε-insensível.

2. Como $\alpha_i \alpha_i^* = 0$, ou seja, nunca existirá um conjunto de variáveis duais α_i e α_i^*, que sejam simultaneamente não-nulas, pois isso faria com que a função de otimização se movimentasse de forma não-nula em ambas as direções.

3. Os multiplicadores de Lagrange serão não-nulos somente quando $|f(\mathbf{x}) - y_i| \geq \varepsilon$. Isto significa que, para $|f(\mathbf{x}) - y_i| < \varepsilon$, o segundo da fator das Equações 4-17 é não-nulo, e portanto, α_i, α_i^* se anulam para que as condições KKT seja satisfeitas. Dessa maneira existe uma representação esparsa de $\mathbf{w}$ em termos de $\mathbf{x_i}$, isto é, não é preciso todos os x_i para descrever $\mathbf{w}$. Só são considerados os pontos para os quais $\alpha_i, \alpha_i^* > 0$. A esses pontos dá-se o nome de *Vetores Suporte*.

A fórmula explícita de b

As observações 1 e 2 acima fornecem as conclusões 4-20 e 4-21:

$$\varepsilon - y_i + < \mathbf{w}, \mathbf{x_i} > + b \geq 0 \quad e \quad \xi = 0 \quad \alpha_i < C \tag{4-20}$$

$$\varepsilon - y_i + < \mathbf{w}, \mathbf{x_i} > + b \leq 0 \quad e \quad \xi = 0 \quad \alpha_i < C \tag{4-21}$$

Em combinação com uma análise análoga de α_i^* se obtém:

$$\begin{aligned} max\{-\varepsilon + y_i - < \mathbf{w}, \mathbf{x_i} > | \alpha_i < C \ ou \ \alpha_i^* > 0\} \leq b \\ \leq min\{-\varepsilon + y_i - < \mathbf{w}, \mathbf{x_i} > | \alpha_i > 0 \ ou \ \alpha_i^* < C \end{aligned} \tag{4-22}$$

Tomando os pontos de treinamento, para os quais $\alpha_i, \alpha_i^* \in (0, C)$, as Inequações 4-20 e 4-21 tornam-se verdadeiras. Dessa maneira as expressões para encontrar o valor de b, dadas pelas Equações 4-23:

$$\begin{aligned} b = y_i + < \mathbf{w}, \mathbf{x_i} > - \varepsilon \quad para \quad 0 < \alpha_i < C \\ b = y_i - < \mathbf{w}, \mathbf{x_i} > + \varepsilon \quad para \quad 0 < \alpha_i^* < C \end{aligned} \tag{4-23}$$

Para efeitos de estabilidade, recomenda-se tomar uma média sobre todos os pontos x_i e para isso, considera-se $\delta_i = \varepsilon * sinal(\alpha_i - \alpha_i^*)$, onde ε é o erro de predição dado por $f(\mathbf{x}) - y_i$. O fator b é então calculado como sendo a média de $\{\delta_i + y_i - < \mathbf{w}, \mathbf{x_i} >\}$ para todos os $i = 1, \cdots, l$.

Funções Núcleos

A teoria abordada até aqui, considera apenas os modelos lineares no espaço de entrada, que por serem de fácil interpretação, tornam-se convenientes e necessários em alguns casos, como quando há poucos dados disponíveis. Contudo, aplicações complexas do mundo real requerem um espaço de hipóteses mais expressivo do que funções lineares e exigem a exploração de características mais abstratas dos dados. Nesse sentido, representações por núcleos se apresentam como uma alternativa à combinação linear simples dos dados.

Uma maneira de se obter a não-linearidade do algoritmo é através da aplicação de um mapeamento não-linear sobre os padrões de treinamento $\mathbf{x}$, via uma função $\Phi : \mathbb{R}^n \to \Im$ em um espaço de características de alta dimensionalidade $\Im$, onde a máquina linear possa ser utilizada. Pode-se construir máquinas não-lineares seguindo duas etapas:

- primeiramente, é feito um mapeamento não-linear que transforma os dados em um espaço característico de alta dimensionalidade por meio do uso de máquinas lineares na representação dual.
- em seguida, é feita uma regressão linear no novo espaço.

Com o aumento da dimensionalidade do mapeamento por conta do produto interno, a construção de máquinas lineares feita dessa maneira, torna-se impraticável. A solução obtida é o mapeamento implícito por meio das chamadas *funções núcleo* ou *kernels*, que serão correspondentes ao produto interno em algum espaço característico.

Definição 4.1 (Funções Núcleo) *Uma função núcleo é uma função K tal que para todo $\mathbf{x}, \mathbf{z} \in \mathbb{R}^n$ se tem $K(\mathbf{x}, \mathbf{z}) =< \Phi(\mathbf{x}), \Phi(\mathbf{z}) >$ onde Φ corresponde ao mapeamento de $\mathbb{R}^n$ para um espaço característico $\Im$.*

O problema de otimização descrito pela formulação 4-14 depende apenas dos produtos internos entre os diferentes padrões. Portanto ao substituir $< \mathbf{x}, \mathbf{z} >$ por $K(\mathbf{x}, \mathbf{z})$, ao invés de se usar $\Phi(\mathbf{x})$ explicitamente, o problema passa a depender de $< \Phi(\mathbf{x}), \Phi(\mathbf{z}) >$ e as considerações feitas anteriormente estarão satisfeitas, com a condição de que uma regressão linear em um espaço

de dimensão infinita seja realizada. Essa substituição permite que o problema de otimização dual seja reescrito com a formulação 4-24:

$$\begin{aligned} minimize & \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(\mathbf{x},\mathbf{z}), \\ -\varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*), \end{cases} \\ sujeito \;\; a & \sum_{i=1}^{l}(\alpha_i + \alpha_i^*) = 0 \;\; e \;\; \alpha_i, \alpha_i^* \in [0, C]. \end{aligned} \tag{4-24}$$

Ao se utilizar a definição de funções núcleo, o vetor $\mathbf{w}$ não será mais calculado de forma explícita, pois será reescrito da seguinte maneira:

$$\mathbf{w} = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\Phi(\mathbf{x}_i) \tag{4-25}$$

E função de previsão inicialmente definida pela Equação 4-4, será estimada pela Equação 4-26:

$$f(\mathbf{x}) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)K(\mathbf{x_i}, \mathbf{x}) + b \tag{4-26}$$

Um ponto positivo em escrever o problema de otimização na forma dual é que dessa forma o número de parâmetros não depende do número de atributos, mas sim do tamanho da amostra. A escolha de uma função núcleo, possibilita a realização de um mapeamento não-linear de forma implícita sem aumentar o número de parâmetros ajustáveis, visto que uma função núcleo calcula o produto interno de dois vetores característicos que correspondem a dois valores de entradas. Dessa forma, a função núcleo é calculada explicitamente e o espaço característico é calculado de forma implícita.

Condições para as funções núcleos

Funções núcleo podem ser consideradas como generalização de produtos internos. Então, as propriedades pertinentes a elas são aquelas que também ao produto interno. Contudo, é preciso observar que algumas propriedades de produto interno, não ocorrem em funções núcleo, como a linearidade por exemplo. No entanto, ainda é necessário definir os tipos de funções $K(\mathbf{x}, \mathbf{z})$ corresponderão a um produto interno em algum espaço característico $\Im$. O teorema 4.2 caracteriza este tipo de funções.

Teorema 4.2 (Mercer) *Existem um mapeamento* Φ *e uma expansão da forma* $K(\mathbf{x}, \mathbf{z}) = < \Phi(\mathbf{x}), \Phi(\mathbf{z}) > = \sum_t \Phi(\mathbf{x})_t, \Phi(\mathbf{y})_t$ *se e somente se,* $\forall\, g(\mathbf{x})$ *tal que* $\int g(\mathbf{x})^2 d\mathbf{x}$ *é finito, implica que* $\int K(\mathbf{x}, \mathbf{z})g(\mathbf{x})g(\mathbf{z})d\mathbf{x}d\mathbf{z} \geq 0$.

O Teorema 4.2 determina se uma certa função núcleo constitui ou não um produto interno em algum espaço característico $\Im$, porém não mostra como

construir a função Φ, nem explicita o espaço característico em questão. A partir desse resultado se pode derivar regras para a composição de funções núcleo que também satisfazem às condições de Mercer.

Exemplos de funções núcleo

– Função polinomial homogênea:

$$K(\mathbf{x}, \mathbf{z}) =< \mathbf{x}, \mathbf{z} >^p, \ \ com \ \ p \in \mathbb{N}. \tag{4-27}$$

No caso particular de $p = 1$, esta função é denominada *função núcleo linear* e pode ser utilizada para prever séries temporais tendo como modelo resultando um modelo alto regressivo AR.

– Função polinomial não-homogênea:

$$K(\mathbf{x}, \mathbf{z}) = (< \mathbf{x}, \mathbf{z} > +c)^p, \ \ com \ \ p \in \mathbb{N} \ \ e \ \ c \geq 0. \tag{4-28}$$

Esta função pode ser escrita como uma soma de funções núcleo polinomiais homegêneas.

– Função Base Radial Gaussiana (RBF-Gaussiana):

$$K(\mathbf{x}, \mathbf{z}) = exp(-\gamma \left\|\mathbf{x} - \mathbf{z}\right\|^2). \tag{4-29}$$

Funções RBF são funções que podem ser escritas como $K(\mathbf{x}, \mathbf{z}) = f(d(\mathbf{x}, \mathbf{z}))$, onde d é uma métrica em $\mathbb{R}^n$ e f é uma função em $\mathbb{R}^+$.

Todas as funções acima estão disponíveis no pacote `kernlab` da `plataforma R`.

Também é possível criar núcleos mais complexos a partir de núcleos mais simples. Para mais informações sobre funções núcleo consulte em (29).

Arquitetura de um SVR

A idéia básica do SVR é mapear os dados de entrada $\mathbf{x} \in \mathbb{R}^n$ em um espaço característico de alta dimensionalidade $\Im$ via um mapeamento não-linear $\Phi : \mathbb{R}^n \rightarrow \Im$ e aí aplicar um regressão linear sobre os dados mapeados.

A Figura 4.7 mostra um esquema que representa as etapas do processo para regressão por SVR. Os dados de entrada são mapeados não-linearmente para um espaço característico por meio da pela função Φ, onde o produto interno entre os dados mapeados é calculado. Isso corresponde ao cálculo das funções núcleo ou *kernels* denominadas $K(\mathbf{x}, \mathbf{x_i})$. Por fim, os resultados são combinados por meio dos pesos $v_i = (\alpha_i^* - \alpha_i)$ que adicionados ao termo constante b fornecem uma previsão para o valor de $\mathbf{x}$.

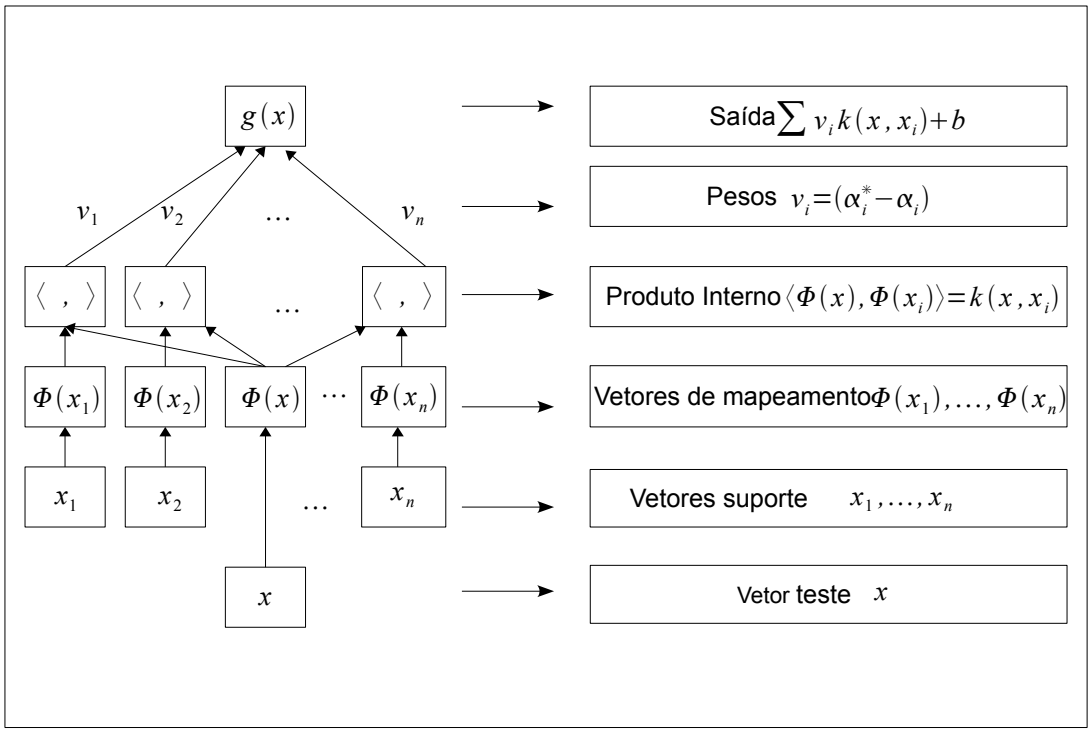

Figura 4.7: Arquitetura de uma máquina de suporte vetorial para regressão. (29)

4.3 Processo Gaussiano

Um processo de gaussiano (GP) é um processo estocástico, cujas realizações consistem em associar valores aleatórios a cada um dos pontos de um intervalo de tempo (ou espaço), de modo que cada variável aleatória tenha uma distribuição normal. Além disso, cada coleção finita dessas variáveis aleatórias possui uma distribuição normal multivariada.

Processos Gaussianos são importantes em modelagem estatística por causa das propriedades herdadas da distribuição normal. Por exemplo, se um processo aleatório é modelado como um processo de Gauss, a distribuição de grandezas como a média e o erro de estimação podem ser obtidas de forma explícita. Na verdade, processo gaussiano, GP, é uma generalização da distribuição de probabilidade Gaussiana. Uma análise Bayesiana de aprendizado estatístico possibilita introduzir *a priori* nos parâmetros das funções base. No entanto, é possível introduzir *a priori* sobre as próprias funções produzidas pelo modelo. Uma vantagem de se usar diretamente o espaço de funções *a priori* é que ele pode estabelecer uma restrição para suavidade sem estar vinculado a

um número limitado de funções base.

Essa técnica une de forma consistente o rigor matemático e o tratamento computacional em problemas tanto de classificação, quanto de regressão. Neste trabalho, como já foi mencionado, é abordado o processo gaussiano para regressão.

Um breve comentário sobre Inferência Bayesiana

A *Inferência Bayesiana* é um tipo de inferência estatística que descreve as incertezas sobre quantidades não observadas diretamente de forma probabilística. As incertezas são modificadas periodicamente após observações de novos dados ou resultados.

Esta abordagem é realizada basicamente seguindo os seguintes passos:

- Escolhe-se uma função densidade $g(\theta)$ chamada distribuição *a priori* que representa todas as informações a respeito do parâmetro θ antes de conhecermos os dados amostrais.
- Escolhe-se um modelo estatístico $g(X|\theta)$ que reflete todas as informações de X dado θ.
- Depois, observam-se os dados $(X_1, \cdots, X_n)$, atualizam-se as informações e calcula-se a distribuição *a posteriori* $g(\theta|X_1, \cdots, X_n)$, que representa a informação resultante a respeito θ de após o conhecimento dos dados.

4.3.1 Processo Gaussiano para Regressão

Existem várias maneiras de interpretar processos gaussianos para regressão, conhecidos pela sigla GPR's, derivada do inglês *Gaussian Process Regression*. Neste trabalho é considerada a abordagem onde os GPR's são usados para definir uma distribuição sobre funções e inferir sobre ela usando diretamente o espaço de funções.

O algoritmo de um GPR também é construído por meio de treinamento. Nesse caso, são considerados como dados de treinamento os valores $(\mathbf{x_i}, y_i)_{i=1,\cdots,l}$ onde $\mathbf{x_i} \in \mathbb{R}^n \;\; e \;\; y_i \in \mathbb{R}$. O objetivo é construir uma função $f : \mathbb{R}^n \to \mathbb{R}$ que aproxime os dados futuros de acordo com uma relação estabelecida pelo treinamento. Em outras palavras, o que se quer é fazer a previsão de valores de saída $f(\mathbf{x}_*)$ para um novo valor de entrada $\mathbf{x}_*$, dado o conjunto de treinamento $D = \{(\mathbf{x_i}, y_i), i = 1, \cdots, l\}$.

Definição 4.3 *Um processo gaussiano é uma coleção de variáveis aleatórias, onde qualquer conjunto finito delas possui uma distribuição conjunta Gaussiana. (26).*

Um GP é completamente especificado por sua função média $m(\mathbf{x})$ e sua função covariância $k(\mathbf{x}, \mathbf{x}')$. Essas funções, para um processo gaussiano real escrito como $f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, são definidas pelas Equações 4-30 e 4-31:

$$m(\mathbf{x}) = E[f(\mathbf{x})], \tag{4-30}$$

$$k(\mathbf{x}, \mathbf{x}') = E[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \tag{4-31}$$

Definir um processo gaussiano como uma coleção de variáveis aleatórias implica na necessidade de *consistência* ou *propriedade de marginalização*. Essa propriedade significa que se um GP, especifica $(y_1, y_2) \sim N(\boldsymbol{\mu}, \sum)$, deve especificar também ($y_1 \sim N(\mu_1, \sum_{11})$, onde $\sum_{11}$ é uma submatriz de $\sum$, ou seja, a distribuição obtida para um grande conjunto de variáveis não modifica para uma quantidade pequena de variáveis. Se a função covariância determina as entradas da matriz de covariância, a *consistência* exigida é satisfeita.

A função covariância determina a covariância entre pares de variáveis aleatórias de forma que a covariância das saídas é escrita em função das entradas, isto é, $Cov(f(\mathbf{x_p}), \mathbf{f}(\mathbf{x_q})) = k(\mathbf{x_p}), \mathbf{f}(\mathbf{x_q})$. Pode ser mostrado que para toda função covariância $k(\ ,\ )$ definida positiva existe uma expansão (possivelmente infinita) em termos de funções bases. Existem muitas escolhas razoáveis para essa função. Informações adicionais são encontradas em (26).

A determinação da função covariância resulta em uma distribuição sobre funções, pois pode-se conseguir amostras de uma distribuição de funções avaliando quaisquer quantidade de pontos. Em outras palavras, pode-se escolher valores de entrada para teste, formando assim um conjunto $X_* = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_l\}$, representado por uma matriz $l \times k$, onde cada $\mathbf{x}_i$ com $i = 1, \cdots, l$ corresponde a um vetor linha. Em seguida avaliar a função nesses pontos e consequentemente escrever sua matriz de covariância $\mathbf{K}$. Dessa forma dá-se origem a um vetor gaussiano aleatório 4-32, o que torna possível a interpretação dos valores gerados como uma função de entradas.

$$f_* \sim N(0, \mathbf{K}(X_*, X_*)) \tag{4-32}$$

Previsão considerando observações sem ruídos

Considere que o conjunto $\{(\mathbf{x_i}, f_i), i = 1, \cdots, l\}$ seja conhecido. A distribuição conjunta *a priori* entre as saídas do treinamento $\mathbf{f} \subset \mathbb{R}$ e as saídas de

teste $\mathbf{f}_* \subset \mathbb{R}$ é dada pela forma:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} \mathbf{K}(X,X) & \mathbf{K}(X,X_*) \\ \mathbf{K}(X_*,X) & \mathbf{K}(X_*,X_*) \end{bmatrix}\right) \tag{4-33}$$

onde $\mathbf{f} = [f(x_1), \cdots, f(x_l)]^T$, X corresponde a um conjunto de entradas $\mathbf{x_i}$ para o treinamento e X_* corresponde a uma conjunto de entradas $\mathbf{x_{i*}}$ para teste.

Se existe l pontos de treinamento e l_* pontos de teste então $\mathbf{K}(X, X_*)$ representa uma matriz de covariância $l \times l_*$ avaliada em todos os pares $(\mathbf{x}, \mathbf{x}_*)$, onde $\mathbf{x}$ é um ponto de treinamento e $\mathbf{x}_*$ é um ponto de teste. Assim de maneira análoga, defini-se as matrizes $\mathbf{K}(X,X)$, $\mathbf{K}(X_*,X_*)$ e $\mathbf{K}(X_*,X)$. Para se obter uma distribuição *a posteriori* é necessário restringir essa distribuição conjunta *a priori* somente às funções que estejam de acordo com os dados observados. Isto corresponde a utilizar como *a priori* a distribuição conjunta condicional Gaussiana, como descrito na Equação 4-34.

$$\begin{aligned} f_*|X_*, X, \mathbf{f} \sim N(\mathbf{K}(X_*,X)\mathbf{K}(X,X)^{-1}\mathbf{f}, \\ \mathbf{K}(X_*,X_*) - \mathbf{K}(X_*,X)\mathbf{K}(X,X)^{-1}\mathbf{K}(X,X_*)). \end{aligned} \tag{4-34}$$

O valores de f_* correspondentes às entradas X_* podem ser amostrados a partir da distribuição conjunta *a posteriori* avaliando a média e a matriz covariância da distribuição condicional 4-34.

Previsão considerando observações com ruídos

Existem situações onde é preciso considerar ruídos (erros) de forma que as saídas sejam escritas como $y = f(\mathbf{x}) + \varepsilon$, onde ε corresponde aos ruídos. Assumindo que os ruídos são independente e identicamente distribuídos e que possuem variância σ_n^2, a distribuição *a priori* das observações ruidosas torna-se:

$$Cov(f(\mathbf{x}_p), f(\mathbf{x}_q)) = k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2 \delta_{pq} \quad ou \quad Cov(\mathbf{y}) = \mathbf{K}(X,X) + \sigma_n^2 \mathbf{I} \tag{4-35}$$

onde δ_{pq} corresponde a um *Delta de Kronecker* que é igual a um se $p = q$ e igual a zero, caso contrário.

Introduzindo o termo de ruídos na Expressão 4-33 pode-se escrever a distribuição conjunta sobre os valores observados e os valores de teste, como *a priori*:

$$\begin{bmatrix} \mathbf{y} \\ \boldsymbol{f_*} \end{bmatrix} \sim N\left(0, \begin{bmatrix} \mathbf{K}(X,X) + \sigma_n^2 \mathbf{I} & \mathbf{K}(X,X_*) \\ \mathbf{K}(X_*,X) & \mathbf{K}(X_*,X_*) \end{bmatrix}\right) \tag{4-36}$$

Assim, pode-se extrair a distribuição conjunta condicional correspondente a Expressão 4-34 e encontrar as equações fundamentais para as previsões

de um GPR. Essas equações são descritas pelas Expressões 4-37, 4-38 e 4-39:

$$\mathbf{f}_*|X, \mathbf{y}, X_* \sim N(\bar{\mathbf{f}}_*, Cov(\mathbf{f}_*)), \quad onde \tag{4-37}$$

$$\bar{\mathbf{f}}_* \doteq E[\mathbf{f}_*|X, \mathbf{y}, X_*] = \mathbf{K}(X_*, X)[\mathbf{K}(X, X) + \sigma_n^2\mathbf{I}]^{-1}\mathbf{y} \tag{4-38}$$

$$Cov(\mathbf{f}_*) = \mathbf{K}(X_*, X_*) - \mathbf{K}(X_*, X)[\mathbf{K}(X, X) + \sigma_n^2\mathbf{I}]^{-1}\mathbf{K}(X, X_*) \tag{4-39}$$

Para efeitos de simplicidade pode ser utilizada a seguinte notação: $\mathbf{K} = \mathbf{K}(X, X)$ e $\mathbf{K}_* = \mathbf{K}(X, X_*)$. No caso em que há somente um dado de teste, define-se $\mathbf{k}(\mathbf{x}_*) = \mathbf{k}_*$ para denotar o vetor de covariâncias entre o ponto de teste e os l pontos de treinamento, ou seja, $\mathbf{k}(\mathbf{x}_*) = [k(\mathbf{x}_*, \mathbf{x}_1), \cdots, k(\mathbf{x}_*, \mathbf{x}_l)]$. Dessa forma as Equações 4-38 e 4-39 podem ser reescritas da seguinte forma:

$$\begin{aligned} \bar{f}(\mathbf{x}_*) &= \mathbf{k}^T(\mathbf{x}_*)(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y} \\ Var[f(\mathbf{x}_*)] &= k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}^T(\mathbf{x}_*)(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{k} \end{aligned} \tag{4-40}$$

Para analisar a distribuição prevista pelas Equações 4-40 deve-se observar que a previsão da média $\bar{f}(\mathbf{x}_*)$ é uma combinação linear das observações $\mathbf{y}$. Outra maneira de olhar para esta equação é considerá-la como uma combinação linear de l funções núcleo k, cada uma centrada em um ponto de treinamento. Formalmente:

$$\bar{f}_* = \sum_{i=1}^{l} \alpha_i k(\mathbf{x}_i, \mathbf{x}_*), \tag{4-41}$$

onde $\alpha = (\mathbf{K} + \sigma_n^2 I)^{-1} y$.

Vale ressaltar também que a variância de f_*, descrita em 4-40, não depende dos dados de saída, mas apenas dos dados de entrada. Esse fato corresponde a uma propriedade da distribuição Gaussiana. Ela é calculada pela diferença entre dois termos: a covariância *a priori*, $\mathbf{K}(\mathbf{x}_*, \mathbf{x}_*)$, e um termo que representa as observações dadas sobre a função.

A distribuição prevista pelo modelo GP fornece mais do que apenas os erros ponto-a-ponto da equação simplificada 4-40. Observe que a Equação 4-39 se mantém inalterada quando X_* denota várias entradas de teste e nesse caso as covariâncias das saídas de teste são calculadas e os elementos da diagonal são as variâncias ponto-a-ponto. Em suma, a Equação 4-38 é a função média e a Equação 4-39 é a função covariância do processo gaussiano *a posteriori*.

Função de Verossilhança Marginal

De fato, um processo estocástico *a priori* sobre funções, dado um conjunto, $x_1, \cdots, x_n$, fornece uma distribuição de probabilidade sobre suas corres-

pondentes saídas $\mathbf{f} = [f(x_1), \cdots, f(x_n)]$. Defini-se a distribuição *a priori* sobre funções por $P(\mathbf{f})$, $P(f_*, \mathbf{f})$ para a distribuição conjunta de f_* e $\mathbf{f}$), e $P(X|\mathbf{f})$, a probabilidade condicional de valores particulares $X = ((x_1), \cdots, (x_n))^T$ dados os valores de $\mathbf{f}$. Assim para um modelo que considera ruídos, se obtém a seguinte probabilidade condicional:

$$P(y|X) = \int P(y, \mathbf{f}|X)d\mathbf{f} = \frac{1}{P(X)} \int P(y|\mathbf{f})P(\mathbf{f})P(X|\mathbf{f})d\mathbf{f} = \int P(y|\mathbf{f})P(\mathbf{f}|X)d\mathbf{f}. \tag{4-42}$$

Portanto um previsão para a distribuição para $\mathbf{f}$ é encontrada por meio da marginalização do produto entre função *a priori* e a função de verossimilhança. Se $P(X|\mathbf{f})$ e $P(y|\mathbf{f})$ são gaussianas então $P(\mathbf{f}|X)$ também é, sendo que sua média e variância são calculadas usando cálculos de matrizes $n \times n$. A integral descrita em 4-42 é definida como *função de verossilhança marginal*. Esse termo se refere exatamente à marginalização sobre funções. De acordo com o modelo de Processo Gaussiano, *a priori* sobre funções é gaussiana. Sendo assim, $\mathbf{f}|X \sim N(\mathbf{0}, \mathbf{K})$ ou:

$$\log\ P(\mathbf{f}|X) = -\frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f} - \frac{1}{2}\log|\mathbf{K}| - \frac{n}{2}\log 2\pi. \tag{4-43}$$

Dessa forma a função de verossilhança é uma Gaussiana fatorada $\mathbf{y}|\mathbf{f} \sim N(\mathbf{f}, \sigma_n^2 I)$ e por meio do uso das identidades gaussianas, pode-se calcular a integral 4-43 e obter o seguinte resultado:

$$\log\ P(\mathbf{f}|X) = -\frac{1}{2}\mathbf{y}^T(\mathbf{K} + \sigma_n^2 I)^{-1}\mathbf{y} - \frac{1}{2}\log\left|\mathbf{K} + \sigma_n^2 I\right| - \frac{n}{2}\log 2\pi. \tag{4-44}$$

Esse resultado também pode ser observado diretamente do fato de que $y \sim N(0, \mathbf{K} + \sigma_n^2 I)$.

Na prática, para a implementação de um algoritmo GPR utiliza-se a decomposição de Cholesky, ao invés de calcular diretamente as matrizes inversas, visto que esse processo é computacionalmente mais rápido e estável.

Arquitetura de um GPR

Um Processo Gaussiano para Regressão (GPR) é especificado por uma média e uma função de covariância. A média é uma função de $\mathbf{x}$ (que frequentemente é nula) e a covariância é uma função $k(\mathbf{x}, \mathbf{x}')$ que expressa a covariância entre o valor esperado da função $f(x)$ avaliada nos pontos $\mathbf{x}$ e $\mathbf{x}'$. A função $f(\mathbf{x})$ em qualquer problema de modelagem de dados é assumida como sendo uma única amostra das distribuição de Gauss. Como exposto na Seção 4.3.1, o objetivo de um GPR é construir uma função $f : \mathbb{R}^n \to \mathbb{R}$ que aproxime os dados de saída futuros. Em outras palavras, o que se quer é fazer

a previsão de valores de saída $f(\mathbf{x}_*)$ para um novo valor de entrada $\mathbf{x}_*$, dado um certo conjunto de treinamento $D = \{(\mathbf{x}_i, y_i), i = 1, \cdots, l\}$.

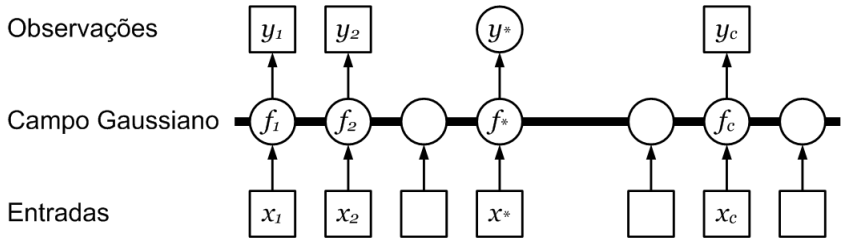

Figura 4.8: Modelo gráfico de um GPR. (26)

O modelo de gráfico em cadeia da Figura 4.8 representa um GP para a regressão. Os quadrados representam as variáveis observadas e os círculos representam incógnitas. A barra horizontal grossa representa um conjunto de nós totalmente conectado. Note que uma observação y_i é condicionalmente independente de todos os outros nós, dada a variável correspondente, f_i. Por causa da propriedade marginalização dos GP's, a adição de novas entradas, $\mathbf{x}$, variáveis f_i, e as saídas desconhecidas, y_*, não alteram a distribuição de qualquer outra variável.

4.4 SVR e GPR no software R

R é uma linguagem computacional e um ambiente para a análise estatística de dados e gráficos. Esse software fornece uma ampla variedade de pacotes estatísticos (modelagem linear e não linear, testes estatísticos clássicos, análise de séries temporais, classificação, entre outros) e técnicas gráficas de qualidade. Ele pode ser usado em vários sistemas computacionais e está disponível no sítio http://www.r-project.org/, onde se encontram textos, tutoriais e códigos-fonte para compilação, além de executáveis já compilados.

Nas simulações feitas nesta dissertação foram utilizadas as implementações "ksvm" e "gausspr" para aplicações de Máquinas de Suporte Vetorial e Processos Gaussianos, respectivamente, disponíveis no pacote estatístico **kernlab** descritos por Karatzoglou e Smola em (16). Ambas a implementações permitem que o usuário escolha a finalidade do algoritmo, ou seja, se é um aplicação de classificação ou de regressão. E em ambas são definidos uma matriz de dados $\mathbf{X}$ e um vetor de respostas $\mathbf{y}$.

Na implementação "ksvm", por exemplo, é possível escolher as alternativas *eps-svr e nu-svr* para regressão, entre outras. A implementação "gausspr" também disponibiliza algoritmos para regressão e possibilita restrições para os parâmetros. Na implementação "gausspr", dependendo do tipo do vetor $\mathbf{y}$,

define-se uma aplicação de classificação ou de regressão. Se **y** for um fator o padrão (default) para "gausspr" é classificação, mas se for um vetor numérico, o padrão (default) é regressão.

A escolha padrão (default) de "ksvm" para regressão é a função *eps-svr*. Neste comando é permitido a definição de vários parâmetros, como: núcleo, sigma, ε, a constante C, entre outros. No comando "gausspr" também é permitido ao usuário a escolha de parâmetros de restrições como: núcleo, sigma, hiper-parâmetros (sinal de variância, ruídos de variância e escala).

Neste trabalho é escolhida para função-núcleo a função radial base gaussiana, ou núcleo gaussiano, cujo comando é "rbfdot" e a expressão matemática é dada pela Equação 4-45. Para os demais parâmetros são utilizados as escolhas padrões de cada implementação do pacote.

$$K(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|}{2\gamma^2}\right). \tag{4-45}$$