Universidade de Brasília
Faculdade de Economia, Administração e Contabilidade (FACE)

Pedro Correia Santos Bezerra

SVR-GARCH com misturas de kernels gaussianos

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro da CAPES

Brasília
2016

Pedro Correia Santos Bezerra

SVR-GARCH com misturas de kernels gaussianos

Dissertação apresentada como requisito parcial à obtenção do título de Mestre em Administração ao Programa de Pós-Graduação em Administração da Universidade de Brasília.

Área de concentração: Finanças e Métodos Quantitativos

Orientador: Prof. Dr. Pedro Henrique Melo Albuquerque

Brasília
2016

Bezerra, Pedro Correia Santos.
SVR-GARCH com misturas de kernels gaussianos - Brasília, 2016. 83p.
Dissertação (Mestrado) - Faculdade de Economia, Administração e Contabilidade (FACE). Departamento de Administração. Universidade de Brasília (UnB).

1. Previsão de volatilidade
2. Teoria do aprendizado estatístico
3. Aprendizado de máquina

I. Bezerra, Pedro Correia Santos II. Universidade de Brasília. Faculdade de Economia, Administração e Contabilidade (FACE). Departamento de Administração.

Pedro Correia Santos Bezerra

SVR-GARCH com misturas de kernels gaussianos

Dissertação de Mestrado sob o título " SVR-GARCH com misturas de kernels gaussianos ", defendida por Pedro Correia Santos Bezerra e aprovada em 18 de abril de 2016, em Brasília, Distrito Federal, pela banca examinadora constituída pelos doutores:

Prof. Dr. Pedro Henrique Melo Albuquerque
Departamento de Administração - Universidade de Brasília(UnB)

Prof. Dr. Daniel Oliveira Cajueiro
Departamento de Economia - UnB

Prof. Dr. Vinícius Amorim Sobreiro
Departamento de Administração - UnB

Aos meus pais, Marli e Francisco.
Ao meu irmão e Físico, Thiago.
Aos meus avós, Maria Alice e Damião Bezerra (in memoriam).
Ao meu tio e guerreiro, Djalma Correia (in memoriam).
Ao meu labrador e amigo, Zulu (in memoriam).

Agradecimentos

Agradeço aos meus queridos pais, Marli e Francisco, e ao meu irmão, Thiago, pelo amor e carinho que sempre tiveram por mim e por terem me dado todas as condições de desenvolver minhas habilidades cognitivas e não-cognitivas. À minha namorada pela amizade e apoio incondicional.

Agradeço aos Professores Pedro Albuquerque e Tadeu Ferreira pelo auxílio para o desenvolvimento deste trabalho. Aos membros da banca, Daniel Cajueiro e Vinícius Sobreiro, pelos excelentes comentários e sugestões. Por fim, agradeço à Isabel Sales pela leitura e revisão atenta deste trabalho.

"Finanças, finanças, são tudo finanças"

Machado de Assis

Resumo

A previsão da volatilidade dos retornos financeiros é fundamental em finanças empíricas. Nos últimos 15 anos, a máquina de suporte vetorial para regressão (*Support Vector Regression* (SVR)) foi proposta na literatura para estimação e previsão da volatilidade devido à sua capacidade de modelar as caudas pesadas, agrupamento de volatilidade e efeito de alavancagem dos retornos financeiros (Cavalcante *et al.*, 2016; Santamaría-Bonfil *et al.*, 2015). Evidências empíricas sugerem que o mercado de capitais oscila entre vários estados (ou regimes) (BenSaïda, 2015), em que a distribuição global dos retornos é uma mistura de distribuições normais (Levy e Kaplanski, 2015). Neste contexto, o objetivo deste trabalho foi implementar misturas de kernels gaussianos no modelo SVR com variáveis de entrada do GARCH (1,1) (denominado SVR-GARCH) para capturar os regimes de mercado e aprimorar as previsões da volatilidade. O SVR-GARCH com combinação convexa de um, dois três e quatro *kernels* gaussianos foi comparado com o *random walk*, SVR-GARCH com *kernel* de ondaleta de Morlet, SVR-GARCH com *kernel* de ondaleta de Chapéu Mexicano, GARCH(1,1), EGARCH(1,1) e GJR(1,1) com distribuição normal, t-Student, t-Student assimétrica e distribuição de erro generalizada (GED) para a série de log-retornos diários do Ibovespa de 22 de dezembro de 2007 a 04 de janeiro de 2016. Para selecionar os parâmetros ótimos do SVR e do *kernel*, utilizou-se a técnica de validação combinada com o procedimento de *grid-search* e análise de sensibilidade. Para comparar o desempenho preditivo dos modelos, utilizou-se o Erro Quadrático Médio (MSE), Erro Quadrático Normalizado (NMSE), Raiz Quadrada do Erro Quadrático Médio (RMSE) e o teste de Diebold-Mariano. Os resultados empíricos indicam que o modelo SVR-GARCH com *kernel* de ondaleta de Chapéu Mexicano e o *random walk* têm desempenho preditivo superior em relação aos demais modelos. Ademais, o SVR-GARCH com mistura de dois, três e quatro kernels gaussianos é superior ao SVR-GARCH com *kernel* de ondaleta de Morlet e um *kernel* gaussiano, o que também é uma novidade e contribuição deste trabalho. Por fim, esta dissertação confirma os achados da literatura em relação à superioridade do SVR na modelagem dos fatos estilizados da volatilidade das séries financeiras em relação aos modelos GARCH linear e não-linear com caudas pesadas.

Palavras-chave: Previsão de volatilidade, Aprendizado de máquina, Teoria do aprendizado estatístico, Máquina de suporte vetorial para regressão, Kernel de ondaleta.

Abstract

Volatility forecasting plays an important role in empirical finance. In the last 15 years, a number of studies has used the Support Vector Regression to estimate and predict volatility due to its ability to model leptokurtosis, volatility clustering, and leverage effect of financial returns (Cavalcante *et al.*, 2016; Santamaría-Bonfil *et al.*, 2015). Empirical evidence suggests that the capital market oscillates between several states (or regimes) (BenSaïda , 2015), in which the overall distribution of returns is a mixture of normal distributions (Levy e Kaplanski, 2015). In this context, the objective of this dissertation is to use a mixture of Gaussian kernels in the SVR based on GARCH (1,1) (heretofore SVR-GARCH) in order to capture the regime behavior and to improve the one-period-ahead volatility forecasts. In order to choose the SVR parameters, I used the validation technique (holdout method) based on grid-search and sensitivity analysis. The SVR-GARCH with a linear combination of one, two, three and four Gaussian kernels is compared with *random walk*, SVR-GARCH with Morlet wavelet kernel, SVR-GARCH with Mexican Hat wavelet kernel, GARCH, GJR and EGARCH models with normal, student-t, skew-student-t and Generalized Error Distribution (GED) innovations by using the Mean Squared Error (MSE), Normalized Mean Squared Error (NMSE), Root Mean Squared Error (RMSE) and Diebold Mariano test. The out-sample results for the Ibovespa daily closing price from August 20, 2013 to January 04, 2016 shows that the SVR-GARCH with Mexican Hat wavelet kernel and random walk model provide the most accurate forecasts. The outcomes also highlight the fact that the SVR GARCH with a mixture of two, three and four Gaussian kernels has superior results than the SVR GARCH with Morlet wavelet kernel and a single Gaussian kernel. Moreover, consistent with the findings of the literature, I confirm that the SVR has superior empirical results in modeling financial time series stylized facts than the linear and non-linear GARCH models with fat-tailed distributions.

Keywords: Volatility forecasting, Machine learning, Statistical learning theory, Kernel methods, Support Vector Regression, Wavelet kernels.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

AIC	Akaike Information Criteria
ARCH	Autoregressive Conditional Heteroskedasticity
BIC	Bayesian Information Criteria
DM	estatística do teste Diebold-Mariano
Dimensão VC	Dimensão Vapnik-Chervonenkis
GARCH	Generalized Autoregressive Conditional Heteroskedasticity
GED	Generalized Error Distribution
EAM	Erro Absoluto Médio
ERM	Minimização Empírica do Risco
MSE	Erro Quadrático Médio
NMSE	Erro Quadrático Médio Normalizado
RMSE	Raiz Quadrada do Erro Quadrático Médio
EGARCH	Exponencial Generalized Autoregressive Conditional Heteroskedasticity
FDP	Função Densidade de Probabilidade
Ibovespa	Índice da Bolsa de Valores de São Paulo
LSSVM	Least Square Support Vector Machine
ML	Maximum Likelihood
GARCH-MN	Mixed Normal GARCH
GJR	Modelo de Glosten-Jagannathan-Runkle
QLM	Quasi-Maximum Likelihood
SVM	Support Vector Machine
SRM	Minimização Estrutural do Risco
SVR	Support Vector Regression
SVR-GARCH	SVR com variáveis de entrada do GARCH(1,1)
SVR-GARCH Morlet	SVR-GARCH com kernel de ondaleta de Morlet
SVR-GARCH Mexican	SVR-GARCH com kernel ondaleta de Chapéu Mexicano
TGARCH	Threshold-GARCH
VC	Vapnik-Chervonenkis

Lista de Símbolos

$argmin(.)$	Argumento do mínimo
I (.)	Função indicadora
E (.)	Operador de esperança
P (.)	Medida de probabilidade
$\Gamma(.)$	Função Gamma
F_{t-1}	Conjunto informacional no tempo $t-1$
Var(.)	Operador de variância
sup(.)	Supremo
$X \sim$ N(0,1)	X possui distribuição normal padrão

Capítulo 1

Introdução

> "*Each of the five tribes of machine learning has its own master algorithm, a general-purpose learner that you can in principle use to discover knowledge from data in any domain. The symbolists master algorithm is inverse deduction, the connectionists is backpropagation, the evolutionaries is genetic programming, the Bayesians is Bayesian inference, and the analogizers is the support vector machine. In practice,however, each of these algorithms is good for some things but not others. What we really want is a single algorithm combining the key features of all of them: the ultimate master algorithm* ".
>
> Domingos (2015, p. xvii)

A previsão de séries temporais financeiras é fundamental para os participantes do mercado financeiro e autoridades governamentais. Nos últimos anos, houve um crescimento expressivo da utilização de algoritmos de *machine learning* na modelagem de séries financeiras, em função de suas habilidades em capturar a natureza não linear, dinâmica e caótica dessas séries, sem a necessidade de realizar suposições sobre a distribuição dos dados (Cavalcante *et al.*, 2016).

A previsão de volatilidade é fundamental para o gerenciamento de riscos, apreçamento de ativos e formação de carteiras de investimento (Poon, Huang. Clive, 2003). A popularidade do GARCH (*Generalized Autoregressive Conditional Heteroskedasticity*) é devido a sua fácil aplicação e a capacidade de modelar em alguma extensão: a aglomeração de volatilidade, as caudas pesadas e a ausência de correlação dos retornos. Não obstante, vários estudos apresentam evidências empíricas que o GARCH possui baixo desempenho preditivo (Brailsford e Faff, 1996; Choudhry e Wu, 2008; Dimson e Marsh, 1990; Jorion, 1995). Diante disso, várias modificações foram propostas para melhorar suas previsões como: mudanças na especificação e estimação do modelo, utilização de diferentes *proxies* para a volatilidade (Chen *et al.*, 2010).

Por serem paramétricos e em geral estimados pelo método da máxima verossimilhança

(*maximum likelihood*, ML), os modelos GARCH lineares e não-lineares fazem suposições sobre a forma funcional do processo gerador dos dados e da distribuição do termo de erro. No entanto, quando a distribuição dos dados não é conhecida, a estimação via ML torna-se menos acurada e eficiente (Li, 2014). Para contornar essas limitações, modelos de previsão de volatidade baseados em algoritmos de aprendizado de máquina foram propostos na literatura, pois não especificam uma forma funcional particular, não estabelecem *a priori* hipóteses sobre a distribuição dos dados, são flexíveis e capazes de capturar características não lineares das séries financeiras (Cao e Tay, 2001, 2003). Dentre eles, destaca-se o uso do *Support Vector Regression* (SVR) na estimação e previsão da volatilidade condicional dos retornos financeiros (Santamaría-Bonfil *et al.*, 2015). Na estimação da volatilidade com modelos família GARCH, além de especificar a distribuição do termo de erro, é necessário estimar parâmetros via ML ou quasi-máxima verossimilhança (*quasi-maximum likelihood*, QML). No entanto, no SVR é necessário apenas especificar suas variáveis de entrada e saída (Li, 2014). Dessa maneira, supera-se uma série de limitações computacionais e de ineficiência na estimação que aparecem em modelos da família ARCH e GARCH.

1.1 SVR na previsão da volatilidade

O *Support Vector Machine* (SVM) é uma técnica de *machine learning* criada por Vapnik (1982) e aprimorada por Boser *et al.* (1992). O treinamento do SVM é equivalente a solução de um problema de programação quadrática com restrições lineares. Por conseguinte, a solução é sempre única e global. Além disso, o SVM utiliza o Princípio da Minimização Estrutural do Risco (*Structural Risk Minimization*, SRM), que faz um balanceamento entre o erro de treino e generalização, promovendo, empiricamente, um melhor desempenho de previsão em relação às redes neurais artificiais (Cao e Tay, 2001).

O uso do SVM para regressão (denominado *Support Vector Regression* (SVR)) na modelagem de séries temporais financeiras se justifica pelo fato dessa ferramenta ser fundamentada na teoria do aprendizado estatístico, ser flexível e ter a habilidade de aproximar qualquer função $\int_{-\infty}^{\infty}|f(x)|^2dx < \infty$, realizando poucas suposições sobre o processo gerador dos dados (Cao e Tay, 2001).

No modelo GARCH, a volatilidade é descrita como uma função do retorno e da volatilidade anteriores. Considerando P_t o preço do ativo no instante t, o log-retorno é dado por $r_t = ln(P_t) - ln(P_{t-1})$. Então, fixa-se o modelo AR(1)-GARCH(1,1):

$$r_t = \mu_t + a_t \tag{1.1}$$

em que

$$\mu_t = \phi_0 + \phi_1 r_{t-1}, \tag{1.2}$$

$$a_t = \sqrt{h_t} z_t, \quad z_t \sim i.i.d(0,1) \tag{1.3}$$

$$h_t = \alpha_0 + \alpha_1 a_{t-1}^2 + \beta_1 h_{t-1} \tag{1.4}$$

Nesse modelo, 1.2 é denominado Equação da média para r_t, 1.4 é a Equação da volatilidade e a_t é o choque no instante t. Assim, a volatilidade é o desvio padrão condicional do retorno.

Conforme demonstrado empiricamente por Fernando *et al.* (2003), Chen *et al.* (2010) e Santamaría-Bonfil *et al.* (2015), o SVR baseado na estrutura do GARCH (denominado neste trabalho de SVR-GARCH), além de melhorar as previsões da volatilidade, tem melhor capacidade de aproximar características não-lineares das séries financeiras como aglomeração de volatilidade, caudas pesadas e o efeito alavancagem.

O mercado financeiro oscila entre diferentes regimes ou estados em função de crises financeiras, ciclo de negócios, mudanças na política fiscal ou monetária (Levy e Kaplanski, 2015). Um pressuposto habitual em finanças é que a distribuição dos retornos é uma normal (Wang e Taaffe, 2015). No entanto, como os retornos estão sujeitos às mudanças de regimes (Ang e Timmermann, 2012; BenSaïda, 2015), mesmo que a distribuição do retorno de cada um dos regimes seja normal, a distribuição global, dado a probabilidade de cada regime é uma mistura de normais (Levy e Kaplanski, 2015). Evidências empíricas indicam a oscilação entre dois regimes no mercado financeiro: um regime de alta e outro de baixa volatilidade (Bae *et al.*, 2014). No entanto, o mercado pode apresentar múltiplos regimes escondidos, o que torna necessário o uso de um número maior de misturas (BenSaïda, 2015; Guidolin , 2011).

Dentro desse contexto, o objetivo deste trabalho é aprimorar as previsões do modelo SVR-GARCH, utilizando misturas de *kernels* gaussianos para capturar os regimes de mercado. Optou-se por testar o SVR-GARCH com um, dois, três e quatro kernels gaussianos. Espera-se que a mistura de funções núcleos gaussianas seja capaz de obter resultados preditivos superiores aos modelos com apenas um *kernel* gaussiano, pois além de reunir as vantagens da combinação de *kernels*, a mistura talvez seja capaz de capturar os regimes de mercado e, por conseguinte, melhorar as habilidades preditivas do SVR-GARCH.

Compara-se o SVR-GARCH com um, dois, três e quatro *kernels* gaussianos com o modelo *random walk*, SVR-GARCH com kernel de ondaleta de Morlet, SVR-GARCH com kernel de ondaleta de Chapéu Mexicano, GARCH (1,1), EGARCH(1,1) e GJR(1,1) com distribuição normal, t-Student, GJR (1,1), t-Student assimétrica e distribição de erro generalizada (GED) para série de retornos do Ibovespa. Para comparar o desempenho preditivo dos modelos, utiliza-se o Erro Quadrático Médio (MSE), Erro Quadrático Normalizado (NMSE), Raiz do Erro Quadrático Médio (RMSE) e o teste de Diebold e Mariano (1995).

1.2 Contribuições

As principais contribuições deste trabalho são as seguintes:

- Modelar os regimes de volatilidade por meio de uma misturas de *kernels* gaussianos no SVR-GARCH;
- Implementar o *kernel* de ondaleta de Chapéu Mexicano no SVR-GARCH;
- Revisar a literatura sobre a estimação e previsão de volatilidade com o uso de SVR; e
- Apresentar as vantagens preditivas do modelo SVR-GARCH em relação aos modelos GARCH linear e não-linear;

1.3 Modelagem empírica

Nesta seção descreve-se o processo de modelagem empírica do trabalho. Primeiro, encontra-se os parâmetros ótimos do SVR por meio da validação, busca em grelha (*grid-search*) e análise de sensibilidade. Em seguida, realiza-se as previsões da volatilidade um período a frente no período de teste via SVR-GARCH com misturas de *kernels* gaussianos para a série de retornos do Ibovespa. Em seguida, essas previsões são avaliadas por meio das métricas do Erro Quadrático Médio (MSE), Erro Quadrático Médio Normalizado (NMSE), Raiz Quadrada do Erro Quadrático Médio (RMSE) e o Teste de Diebold-Mariano.

1.3.1 Especificação do modelo

Converte-se a série do índice de preços P_t, usando a seguinte transformação contínua composta:

$$r_t = log\left(\frac{P_t}{P_{t-1}}\right) \tag{1.5}$$

em que r_t é a série dos log-retornos. Em seguida, divide-se a série de retornos em três conjuntos mutuamente exclusivos: treinamento, validação e teste.

Segundo Poon, Huang. Clive (2003), o modelo GARCH (1,1) é o mais popular na modelagem de volatilidade financeira, pois, além de ser mais parcimonioso que o ARCH, é suficiente para capturar as mudanças da variância ao longo de grandes períodos. Além disso, segundo Hansen e Lunde (2005), o GARCH(1,1) tem excelente desempenho preditivo em comparação a 330 modelos ARCH. Assim, neste trabalho a variância condicional é modelada por um processo GARCH(1,1), enquanto a equação da média condicional é modelada por um processo AR(1). Então o modelo linear do GARCH é especificado da mesma maneira que na seção 1.1.

1.3.2 SVR-GARCH

Para estimar a volatilidade, usa-se um SVR com base no modelo GARCH (1,1), dado pelas seguintes equações:

$$r_t = f\left(r_{t-1}\right) + a_t \tag{1.6}$$

sendo f a função de decisão estimada de forma via SVR para a equação da média. Assim como Cao e Tay (2001), Cao e Tay (2003) e Chen *et al.* (2010) faz-se uma análise de sensibilidade para verificar os efeitos da variação dos parâmetros do SVR no Erro Absoluto Médio (EAM) de previsão no período de validação. Para tanto, varia-se um parâmetro do SVR de cada vez, mantendo os outros fixos. Para a variação de cada parâmetro, é feita a previsão no período de validação e, em seguida, calcula-se o EAM de previsão de modo que os parâmetros escolhidos tenham o menor EAM:

$$EAM = \frac{1}{n}\sum_{t=1}^{n}|\epsilon_t| \tag{1.7}$$

em que ϵ_t é o erro de previsão. De posse dos quadrados dos resíduos obtidos do ajuste do SVR-GARCH à Equação da média, realiza-se o ajuste do SVR-GARCH à Equação da volatilidade:

$$\tilde{h}_t = g(\tilde{h}_{t-1}, a_{t-1}^2) \tag{1.8}$$

em que g a função de decisão estimada pelo SVR, a_t^2 é o quadrado do resíduo obtido do ajuste da equação da média e $\tilde{h}$ é a *proxy* da volatilidade. A seleção dos parâmetros do SVR da Equação da volatilidade é feita da mesma forma que na Equação da média.

1.3.3 Escolha do kernel

Para capturar os k regimes de mercado, optou-se por utilizar misturas de $k = 1, 2, 3, 4$ *kernels* gaussianos na Equação da volatilidade do SVR-GARCH:

$$K_{mix}(x, x') = \sum_{k=1}^{K}\rho_k \times K_k(x, x'), \quad \rho_k \geq 0 \quad \text{e} \quad \sum_{k=1}^{K}\rho_k = 1 \tag{1.9}$$

em que ρ_k é o peso da mistura e $K(x, x')_k = \exp(-\gamma \| x - x' \|^2)$. É importante ressaltar que essa combinação linear de *kernels* satisfaz a condição de Mercer (1909). Para a Equação da média, utiliza-se apenas um *kernel* gaussiano.

1.3.4 Seleção e avaliação do modelo via validação

Num problema de aprendizado, deseja-se encontrar o algoritmo que capture as principais características da amostra de treinamento, mas que também seja capaz de prever de forma acurada os dados do conjunto de teste desconhecidos pela máquina. Assim, o objetivo é encontrar o modelo mais simples que se ajusta bem a um conjunto de dados e ainda tem o menor erro de generalização. A capacidade de generalização do algoritmo, dada pela acurácia da previsão do rótulo para um novo conjunto de dados, pode ser analisada com base em dois conceitos: *overfitting* e *underfitting*. Quando o SVR for confrontado com novas observações na fase de teste e apresentar uma baixa taxa de acurácia, então tem-se o superajustamento (*overfitting*) dos dados de treinamento. Caso apresente uma baixa taxa de acerto no conjunto de treinamento, então tem-se o subajustamento (*underfitting*). O objetivo é encontrar o modelo mais simples que não tenha problema de *overfitting*.

O desempenho na generalização dá uma medida da qualidade do modelo escolhido. As técnicas de validação cruzada (*cross-validation*) são usadas para mensurar a capacidade preditiva de um modelo estatístico (Arlot e Celisse, 2010). Em *Machine Learning* utiliza-se a validação cruzada para a avaliação de modelos que têm por finalidade a previsão. Neste trabalho usa-se a técnica de validação, denominada também método *holdout*, que é a técnica mais simples de validação-cruzada (Kohavi, 1995). Para isso é necessário dividir a base de dados em três conjuntos mutuamente exclusivos: treino, validação e teste (Shalev-shwartz e Ben-david, 2014). O conjunto de treinamento serve para treinar o algoritmo, o de validação para selecionar os parâmetros ótimos. Em seguida, o desempenho de previsão do SVR é avaliado no conjunto de teste (período fora da amostra). É comum encontrar na literatura de *Machine Learning*, a seguinte divisão: treino e teste. A única diferença é que a validação está dentro do conjunto de teste. Neste trabalho, optou-se por separar 50% da base de dados para o conjunto de treinamento, os 20% restantes para o conjunto de validação e as últimas 30% observações fazem parte do conjunto de teste.[1]

1.3.5 *Proxy* da volatilidade e métricas de avaliação de previsão

Como a volatilidade não é observável diretamente, é necessário o uso de uma *proxy* para calcular a volatilidade *ex-post*. Neste trabalho utiliza-se a seguinte *proxy*:

$$\tilde{h}_t = (r_t - \bar{r})^2 \tag{1.10}$$

em que r_t são os retornos e $\bar{r}$ é a média dos retornos. O uso dessa *proxy* é comum e já foi utilizada em muitos trabalhos (Brooks, 2001; Brooks e Persand, 2003; Chen *et al.*, 2010). No entanto, segundo Andersen e Bollerslev (1998) as críticas ao baixo desempenho preditivo dos modelos GARCH podem ser decorrentes do uso de *proxies* pouco adequadas na avaliação das previsões. Os autores indicam que a *proxy* mais adequada é a volatilidade realizada, que é calculada com o uso de dados intra-diários. Devido a impossibilidade de acessar esse tipo de dado pelo autor deste trabalho, utiliza-se somente a *proxy* dada pela Equação 1.10.

[1] Segundo Hastie *et al.* (2009, p. 222), não há uma regra geral para determinar o número de observações de cada um dos três conjuntos. É habitual dividir a base em 50% para treino e 50% para validação e teste.

Num tarefa de previsão de volatilidade é necessário avaliar o desempenho preditivo através de alguma função de perda estatística (Amendola e Candila, 2016). A escolha do modelo com melhor desempenho preditivo é sensível à métrica escolhida (Brailsford e Faff, 1996). Não obstante, mesmo que a volatilidade real não seja conhecida e sua *proxy* tenha ruído, Patton (2011) demonstrou as condições suficientes e necessárias para que uma função de perda seja robusta e permita um ranqueamento consistente das previsões. Dentre as funções robustas e não robustas tem-se, por exemplo, o Erro Quadrático Médio (MSE) e o Erro Absoluto Médio (EAM), respectivamente (Amendola e Candila, 2016). Assim, neste trabalhou optou-se por usar o Erro Quadrático Médio (MSE), Erro Quadrático Normalizado (NMSE) e a Raiz Quadrada do Erro Quadrático Médio (RMSE).

Um bom modelo de regressão é aquele que produz o valor mais próximo do real. O erro (ϵ_t) de previsão é a diferença entre o valor real (y_t) e o previsto ($\hat{y}_t$) : $\epsilon_t = y_t - \hat{y}_t$. O Erro Quadrático Médio (MSE) é uma função de perda robusta para a avaliação de previsões de volatilidade e é dado pela seguinte forma:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}\epsilon_t^2 \tag{1.11}$$

O Erro Quadrático Normalizado (*Normalized Mean Squared Error*, NMSE) penaliza erros extremos e é dado pela seguinte expressão Cao e Tay (2003):

$$NMSE = \frac{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{\frac{1}{n-1}\sum_{t=1}^{n}(y_t - \bar{y}_t)^2} = \frac{1}{\sigma^2 n}\sum_{t=1}^{n}\epsilon_t^2 \tag{1.12}$$

em que y_t indica a observação no tempo t , $\bar{y}_t = \sum_{t=1}^{n} y_t$, $\hat{y}_t$ denota a previsão de y_t e σ^2 é a variância amostral. Além dessas duas métricas, utiliza-se a Raiz Quadrada do Erro Quadrático Médio (*Root Mean Squared Error*, RMSE) Brailsford e Faff (1996):

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\epsilon_t^2} \tag{1.13}$$

É importante observar que quanto menor forem os valores do EAM, NMSE e RMSE, melhor é a previsão. Tanto o NMSE quanto o RMSE possuem as mesmas vantagens do Erro Quadrático Médio. Para verificar se as diferenças de previsões entre os modelos são estatisticamente significantes utiliza-se o teste de Diebold e Mariano (1995), que apresenta evidência de que um modelo tem melhor previsão que outro. Neste trabalho, utiliza-se o teste bicaudal para a diferença da função de perda do Erro Quadrático Médio (MSE). Assim, tem-se a seguinte hipótese nula e alternativa:

$$H_0 : MSE_0 - MSE_1 = 0 \quad versus \quad H_1 : MSE_0 - MSE_1 \neq 0$$

em que MSE_0 é o erro absoluto médio do modelo padrão (*benchmark*) e MSE_1 é o erro absoluto médio do modelo testado. A hipótese nula do teste estabelece a igualdade da acurácia de previsão de ambos os modelos. Assim, se a hipótese nula for rejeitada, tem-se evidência que de que o modelo *benchmark* é superior ao outro. Ademais, a estatística do teste

Diebold-Mariano (DM) para uma série temporal com volatilidade σ_t é dada por:

$$DM = \frac{1}{\sqrt{n}} \frac{1}{\sqrt{\hat{V}(d)}} \sum_{t=1}^{T} (\sigma_{t+1}^2 - \hat{\sigma}_{0,t+1}^2)^2 - (\sigma_{t+1}^2 - \hat{\sigma}_{1,t+1}^2)^2 \sim N(0,1) \tag{1.14}$$

em que $\hat{\sigma}_{0,t+1}^2$ é a volatilidade estimada do modelo *benchmark*, $\hat{\sigma}_{1,t+1}^2$ é a volatilidade estimada do modelo testado, $d = \sum_{t=1}^{T} (e_{t_0})^2 - (e_{t_1})^2$ e $\hat{V}(d)$ é uma estimativa da variância assintótica de d (Kisinbay, 2010) . Valores negativos da estatística DM indicam superioridade das previsões do modelo *benchmark*.

Pode-se resumir os passos do SVR-GARCH da seguinte forma:

1. Divide-se a série de log-retornos em três conjuntos mutuamente exclusivos: treinamento, validação e teste.
2. Com a base de treinamento, ajusta-se o SVR-GARCH à Equação da média 1.6.
3. Para a escolha dos parâmetros ótimos do SVR, usa-se a análise de sensibilidade: varia-se um de cada vez os parâmetros do SVR num *grid-search*, mantendo os outros fixos. Para a variação de cada um dos parâmetros, é feita a previsão no período de validação e, posteriormente, calcula-se o Erro Absoluto Médio (EAM) de previsão.
4. De posse dos resíduos obtidos do passo anterior, realiza-se o ajuste do SVR-GARCH à Equação da volatilidade 1.8.
5. Para a escolha dos parâmetros do SVR para a Equação da volatilidade, usa-se a análise de sensibilidade e o *grid-search* da mesma forma que para a Equação da média.
6. De posse dos parâmetros ótimos do SVR, realiza-se a previsão da volatilidade um passo a frente para o período fora da amostra (conjunto de teste). Após cada previsão, calcula-se o erro cometido pelo modelo e, posteriormente, repete-se o processo de previsão um passo à frente.
7. Por fim, utiliza-se as métricas de MSE, NMSE e RMSE e o teste de Diebold-Mariano para comparar os modelos de previsão.

1.4 Organização do trabalho

Este trabalho está organizado da seguinte forma:

- No Capítulo 2 realiza-se uma breve revisão dos modelos univariados de volatilidade condicional: ARCH e GARCH linear e não-linear (EGARCH e GJR);
- O Capítulo 3 trata da mistura finita de distribuições em finanças e dos regimes de volatilidade;
- No capítulo 4 é feita uma síntese sobre a Teoria do Aprendizado Estatístico e os métodos de *kernel*;
- No capítulo 5 deriva-se a máquina de suporte vetorial (SVM) para classificação binária, SVM para regressão não-linear (SVR) e apresenta-se uma revisão da literatura sobre o uso do SVR na estimação e previsão da volatilidade condicional; e
- Os resultados empíricos estão no capítulo 6 e as conclusões no capítulo 7 ;

Capítulo 2

Volatilidade condicional

> "*Engle's ARCH model and subsequent volatility modeling research program provided a workable and elegant solution, solving many problems and stimulating a huge amount of related research that advanced not only the econometrics of dynamic volatility and correlation modeling, but also forecasting, asset pricing, portfolio allocation, risk management, market microstructure modeling, duration modeling and ultra-high-frequency data analysis*".
>
> Diebold (2004, p. 171)

A volatilidade é uma das variáveis fundamentais em finanças empíricas, pois é usada na otimização de carteiras, gerenciamento de riscos, apreçamento de ativos, regulação bancária e análise macroeconômica (Brownlees e Gallo, 2009; Poon, Huang. Clive, 2003). Este capítulo realiza uma breve revisão dos principais conceitos e modelos de volatilidade condicional univariados desenvolvidos na literatura de econometria financeira que são utilizados neste trabalho. Segundo Franses e van Dijk (2000), os modelos de previsão de volatilidade podem ser divididos em duas categorias: baseados apenas em preços históricos e baseados na informação de mercado das opções e/ou em adição aos preços históricos. A primeira categoria conhecida como modelos de volatilidade de séries de tempo consiste nos simples modelos de preços históricos[1], modelos da família GARCH e modelos de volatilidade estocástica (*Stochastic Volatility*) . A segunda categoria é conhecida como modelos de volatilidade implícita. Neste trabalho utiliza-se o modelo GARCH, tendo como *proxy* para a volatilidade diária a Equação 1.10.

2.1 Introdução

A volatilidade dos retornos financeiros é um fenômeno que não pode ser observado diretamente. As séries temporais financeiras apresentam quatro importantes regularidades

[1] *Random walk*, médias móveis, método de suavização exponencial, ARMA etc.

empíricas (fatos estilizados) da volatilidade dos retornos[2]. Primeiro, a existência de *clusters* (agrupamentos) de volatilidade, isto é, períodos de alta (baixa) volatilidade são seguidos de período de alta (baixa) volatilidade. Segundo, o efeito alavancagem, choques negativos tendem a ter um impacto maior na volatilidade que choques positivos. Isso ocorre pois, notícias ruins (choques negativos) tendem a diminuir o preço da ação. Por conseguinte, a razão dívida/patrimônio se eleva, tornando a ação mais volátil, conforme Black (1976). Terceiro, a distribuição incondicional dos retornos apresenta caudas mais pesadas que a distribuição normal, o que é caracterizado pelo excesso de curtose em relação à normal: grandes mudanças ocorrem com mais frequência do que na normal. Quarto, volatilidade segue o processo de reversão à média, isto é, a volatilidade não diverge para o infinito. Qualquer modelo que pretende modelar a volatilidade deve capturar o maior número de fatos estilizados para descrever de forma acurada a volatilidade dos retornos.

Antes de 1982, os modelos econométricos assumiam a variância constante. O modelo ARCH (*Autoregressive Conditional Heteroscedastic*) univariado, criado por Engle (1982) para estimar a variância da inflação, foi o primeiro a reconhecer que a volatilidade (variância condicional) muda ao longo do tempo em função dos erros passados e que variância incondicional é constante. O ARCH é autoregressivo nos retornos quadráticos, considera que a variância não é constante e está condicionada à informação passada. Após o artigo seminal de Engle (1982) , diversas extensões[3] do ARCH foram propostas para representar de maneira adequada os fatos estilizados dos retornos financeiros .

Bollerslev (1986) generalizou o modelo ARCH para permitir um estrutura com *lag* mais flexível. Segundo o autor, esse processo de generalização é semelhante a do AR para o ARMA e, assim, permite uma estrutura mais parcimoniosa, no sentido do GARCH apresentar menos parâmetros que o ARCH para descrever a volatilidade. Os modelos ARCH e GARCH são não-lineares na variância, mas lineares na média.

Os trabalhos com modelos ARCH eram dedicados a previsão da inflação. No entanto, Bollerslev (1987) constatou que esses modelos seriam relevantes na análise da volatilidade condicional dos retornos financeiros mensais ou de frequência maior. A razão disso é que, mesmo ajustando a autocorrelação pelo modelo ARMA, a série temporal dos retornos tem características que são capturadas pelo GARCH. A principal delas é o agrupamento de volatilidade ao longo do tempo, o que resulta numa autocorrelação positiva do quadrado dos retornos.

2.2 Fatos estilizados das séries financeiras

Séries temporais financeiras são caracterizadas por fatos estilizados: achados empíricos consistentes entre diferentes mercados, períodos e instrumentos (Cont, 2001). Sewell (2011) destaca os seguintes fatos:

1. **Dependência**: a autocorrelação linear dos log-retornos dos ativos financeiros é muito insignificante (ou seja, não há dependência linear entre os retornos). Porém para períodos de tempo intra-diários bem curtos isso não é válido. Ademais, a autocorrelação linear dos retornos absolutos e quadráticos é sempre positiva e significante (o que é conhecido como persistência);

2. **Distribuição**: a distribuição (incondicional) dos retornos apresenta caudas mais pesadas (excesso de curtose em relação a distribuição Normal). Apesar de ser aproximada-

[2]Para mais detalhes, consulte Bollerslev *et al.* (1994).

[3]Para um glossário dessas extensões, consulte Bollerslev (2008)

mente simétrica, a distribuição é leptocúrtica. Os retornos anuais são aproximadamente normais. Porém, à medida que a frequência dos dados aumenta, a distribuição apresenta caudas mais pesadas. A série dos resíduos (distribuição condicional) também apresenta caudas pesadas;

3. **Heterogeneidade**: a distribuição dos retornos financeiros não é estacionária (há aglomerações de volatilidade);
4. **Não-linearidade**: a série temporal dos retornos financeiros apresenta não-linearidades na média e na variância. Evidências empíricas indicam a presença de dependência não-linear dos retornos;
5. **Escala** Mercados exibem propriedades de escala não triviais;
6. **Volatilidade**: apresenta autocorrelação positiva (persistência), dependência de longo prazo da função de autocorrelação, possui uma distribuição log-normal não estacionária (aglomeração de volatilidade) e exibe não-linearidades;
7. **Volume**: o nível de negociação no mercado decai segundo uma lei de potência;
8. **Efeitos Calendário**: são anomalias cíclicas dos retornos baseadas no calendário. Dentre esses destaca-se: os efeitos intra-diários, entre meses e janeiro;
9. **Memória Longa**: há 30% de chance da presença de memória longa nos retornos do mercado de ações e 80% de chance da volatilidade de mercado exibir memória longa; e
10. **Caos**: Há pouca evidência de caos de baixa dimensão nos mercados financeiros.

2.3 Modelos de volatilidade condicional univariados

Os retornos financeiros têm média não condicional próxima de zero, excesso de curtose e quase nenhuma correlação. No entanto, o quadrado dos retornos apresentam alta correlação e persistência, o que torna desejável o uso de processos ARCH e GARCH para modelar a volatilidade condicional.

Seja P_t o preço de fechamento de um ativo no dia t. Seja r_t é a série de log retorno definida por: $r_t = \ln \frac{P_t}{P_{t-1}}$, em que r_t não possui autocorrelação serial ou apresenta correlação serial de ordem baixa, mas é dependente. Os modelos de volatilidade têm por objetivo capturar essa dependência na série de retornos. Assim, tem-se a média e a variância condicionais de r_t dado F_{t-1} (conjunto de informação até o instante $t-1$)(Tsay, 2010):

$$\mu_t = E(r_t|F_{t-1}) \quad h_t = Var(r_t|F_{t-1}) = E[(r_t - \mu_t)^2|F_{t-1}] \tag{2.1}$$

Além disso, considerando que média condicional segue um ARMA(p,q):

$$r_t = \mu_t + a_t, \quad \mu_t = + \sum_{i=1}^{p} \phi_i r_{t-i} - \sum_{i=1}^{q} \theta_i a_{t-i} \tag{2.2}$$

Assim, combinando as equações 2.1 e 2.2, tem-se que (Tsay, 2010):

$$h_t = Var(r_t|F_{t-1}) = Var(a_t|F_{t-1}) \tag{2.3}$$

Segundo Tsay (2010), os modelos de volatilidade condicional estão preocupados em modelar a evolução de h_t ao longo do tempo.

2.3.1 Modelo ARCH univariado

Segundo Engle (1982), antes da introdução do ARCH, os modelos econométricos consideravam que a previsão da variância condicional de um período a frente não dependia da informação passada. Assim, Engle (1982) introduziu um novo modelo econométrico denominado ARCH em que a variância condicional do choque no tempo t é função linear do quadrado dos choques passados. Um ARCH (m) é definido por (Tsay, 2010):

$$r_t = \mu_t + a_t \tag{2.4}$$

$$a_t = \sqrt{h_t} z_t, \quad z_t \sim i.i.d(0,1) \tag{2.5}$$

$$h_t = \alpha_0 + \alpha_1 a_{t-1}^2 + \cdots + \alpha_m a_{t-m}^2 \tag{2.6}$$

em que $\alpha_0 > 0$, $\alpha_i \geq 0, i = 1, \cdots, m-1, \alpha_m > 0$. Os coeficientes α_i devem seguir algumas condições de regularidade para garantir que a variância incondicional de r_t seja finita. Em geral, assume-se que $z_t \sim N(0,1)$, porém é comum o uso de alguma distribuição com caudas mais pesadas (Tsay, 2010). Caso a_{t-1}^2 possua valor absoluto grande, espera-se que a variância condicional h_t e o choque a^2 apresentem uma grande magnitude. Em outras palavras, grandes valores (positivos ou negativos) de a_{t-1} tendem a serem seguidos de grandes valores (positivos ou negativos) de a_t. O que implica que o ARCH é capaz de capturar os *clusters* de volatilidade (Tsay, 2010).

O choque do retorno não tem correlação serial, mas é dependente. Ademais, a dependência de a_t é descrita por uma função quadrática do seus valores defasados. A equação 2.6 pode ser reescrita como um processo AR(m) para a_t^2 (Tsay, 2010):

$$a_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2 + \cdots + \alpha_m a_{t-m}^2 \tag{2.7}$$

A variância incondicional de a_t é dada por:

$$\sigma^2 = \frac{\alpha_0}{1 - \alpha_1 - \cdots - \alpha_m} \tag{2.8}$$

Considere um ARCH(1):

$$h_t = \alpha_0 + \alpha_1 a_{t-1}^2 \tag{2.9}$$

em que $\alpha_0, \alpha_1 > 0$, de forma que $\alpha_0 + \alpha_1 a_{t-1}^2 > 0$. Além disso, para a_t ser estacionário com variância finita α_1 deve ser menor que um.

A média de a_t para o ARCH(1) (Tsay, 2010):

$$E(a_t) = E[E(a_t|F_{t-1})] = 0 \tag{2.10}$$

A variância incondicional de a_t do ARCH(1) é:

$$Var(a_t) = \frac{\alpha_0}{1 - \alpha_1} \tag{2.11}$$

em que $Var(a_t) > 0$ e $0 < \alpha_1 < 1$. Além disso, a curtose de a_t no ARCH(1) com z_t distribuído normalmente é dada por:

$$K = \frac{E[a_t^4]}{E[a_t^2]^2} = \frac{3(1 - \alpha_1^2)}{1 - 3\alpha_1^2} > 3 \tag{2.12}$$

Assim, admitindo que a_t segue o ARCH(1), o modelo captura alguma extensão do excesso de curtose presente nas séries de retornos financeiros.

Segundo Tsay (2010) o ARCH possui as seguintes desvantagens. Primeira, o processo ARCH necessita de muitos parâmetros para descrever a volatilidade. Segunda, reage de forma simétrica a retornos positivos ou negativos. Terceira, é um modelo que impõe muitas restrições nos parâmetros. Quarta, tende a superestimar a volatilidade, pois responde de forma lenta a choques isolados da série de retornos.

2.3.2 Modelo GARCH univariado

Com o intuito de facilitar a estimação do ARCH e torná-lo mais parcimonioso, Bollerslev (1986) propôs o modelo GARCH (*Generalized Autoregressive Conditional Heterocedasticty*). Esse modelo é capaz de capturar os *clusters* de volatilidade, mas assim como o ARCH não é capaz de modelar o efeito alavancagem. Além disso, exige que os parâmetros tenham o quarto momento finito da mesma forma que o ARCH. A introdução da variância condicional defasada no modelo GARCH evita a necessidade de adicionar vários retornos quadráticos defasados, como no caso do ARCH, para modelar a volatilidade. Por consequência, há uma redução no número de parâmetros a serem estimados. Assim, a volatilidade é descrita pelo GARCH como uma função dos retornos passados e da própria volatilidade anterior. O GARCH (m, n) pode ser definido da seguinte maneira (Tsay, 2010):

$$a_t = \sqrt{h_t} z_t, \quad z_t \sim i.i.d(0,1) \tag{2.13}$$

$$h_t = \alpha_0 + \sum_{i=1}^{m} \alpha_i a_{t-i}^2 + \sum_{j=1}^{n} \beta_j h_{t-j}, \tag{2.14}$$

em que,

$$\alpha_0 > 0, \alpha_i \geq 0, i = 1, \cdots, m$$

$$\beta_j > 0, j = 1, \cdots, n$$

$$\sum_{i=1}^{q} (\alpha_i + \beta_i) < 1, \quad q = \max(m, n)$$

Como a média condicional é constante, mas a variância condicional não é constante, o GARCH é um processo não correlacionado, mas dependente. O GARCH (m, n) pode ser escrito como um processo AR(∞), o que indica que períodos de grande volatilidade tendem a ser persistentes. É importante destacar que grandes (pequenas) mudanças em a_{t-1}^2 serão seguidas de grandes (pequenas) mudanças em a_t^2. Além disso, quando $q = 0$, tem-se um ARCH (m) (Tsay, 2010).

Considere agora um modelo AR (1) para a média condicional e GARCH (1,1) para a variância condicional com a seguinte notação Tsay (2010):

$$r_t = u_t + a_t \tag{2.15}$$

com

$$u_t = \phi_0 + \phi_1 r_{t-1}, \tag{2.16}$$

$$h_t = \alpha_0 + \alpha_1 a_{t-1}^2 + \beta_1 h_{t-1} \tag{2.17}$$

Escrevendo $\nu_t = a_t^2 - h_t$ e substituindo em 2.17 pode-se reescrever a equação dos resíduos da seguinte forma:

$$a_t^2 = \nu_t + h_t \tag{2.18}$$

$$a_t^2 = \alpha_0 + (\alpha_1 + \beta_1) a_{t-1}^2 + \nu_t - \beta_1 \nu_{t-1}, \tag{2.19}$$

ou seja, o processo GARCH(1,1) pode ser escrito como um ARMA(1,1) dos resíduos quadráticos (a_t^2) com ν_t como ruído branco, que será estacionário de segunda ordem se $\alpha_i + \beta_i < 1$.

Dado que $E[z_t] = 0$ e $Var[z_t] = 1$, a variância de r_t condicionada ao instante anterior é dada por:

$$\begin{aligned} Var(r_t|F_{t-1}) &= E[(r_t - u_t^2)|F_{t-1}] = E[a_t^2|F_{t-1}] \\ &= E[h_t\epsilon^2|F_{t-1}] = h_t Var[\epsilon_t|F_{t-1}] = h_t \end{aligned} \tag{2.20}$$

Como a_t é estacionário, a variância incondicional do choque a_t é dada por:

$$Var(a_t) = \frac{\alpha_0}{1 - \alpha_1 - \beta_1} \tag{2.21}$$

Como $a_t = \sqrt{h_t}z_t$, a variância incondicional dos retornos também é dada por 2.21, pois $E[h_t] = E[a_t^2]$. Ademais, é possível reescrever a equação 2.14 do GARCH(1,1):

$$h_t = (1 - \alpha_1 - \beta_1)E(h) + \alpha_1 a_{t-1}^2 + \beta_1 h_{t-1} \tag{2.22}$$

A previsão da variância(h_t) do GARCH(1,1) para um período a frente é dada por (Tsay, 2010):

$$\begin{aligned} E(h_{t+1}|a_t, h_t) &= E(\alpha_0 + \alpha_1 a_t^2 + \beta_1 h_t) \\ &= \alpha_0 + \alpha_1 E(a_t^2|F_t) + \beta_1 E(h_t|F_t) \\ &= E(h_{t+1}|a_t, h_t) = \alpha_0 + \alpha_1 a_t^2 + \beta_1 h_t \end{aligned}$$

Ademais, a magnitude de $\alpha_1 + \beta_1$, denominada persistência, mede a permanência do impacto de um choque sobre a volatilidade. Como os valores passados da volatilidade entram na equação do GARCH, a volatilidade apresenta períodos mais persistentes em relação ao ARCH.

Bollerslev (1986) mostra as condições para que o quarto momento do GARCH(1,1) exista. Considerando a existência desse momento, o autor demonstra que o GARCH(1,1) apresenta um excesso de curtose em relação a distribuição normal:

$$K = \frac{E[a_t^4]}{E[a_t^2]^2} = \frac{3[1 - (\alpha_1 + \beta_1)^2]}{1 - (\alpha_1 + \beta_1)^2 - 2\alpha_1^2} > 3 \tag{2.23}$$

O GARCH é capaz de capturar os aglomerados de volatilidade e mesmo quando z_t é gaussiano, o GARCH apresenta mais caudas pesadas em relação a distribuição normal. Não obstante, não é capaz de capturar toda a extensão da assimetria e das caudas pesadas dos retornos financeiros. Por isso, muitas vezes assume-se que z_t é um processo ruído branco independente e identicamente com alguma distribuição que tenha caudas mais pesadas. É importante ressaltar que o GARCH captura a aglomeração de volatilidade de maneira simétrica. No entanto, Ning *et al.* (2015) sugere que a alta volatilidade dos retornos tende a se agrupar mais do que a baixa.

A função de autocorrelação de a_t^2 do GARCH(1,1) pode ser expressa da seguinte forma:

$$\rho_{a^2}(1) = \frac{\alpha_1(1 - \alpha_1\beta_1 - \beta_1^2)}{1 - 2\alpha_1\beta_1 - \beta_1^2} \tag{2.24}$$

Para $k \geq 2$ tem-se que:

$$\rho_{a^2}(k) = (\alpha_1 + \beta_1)^{k-1}\rho_{a^2}(1) \tag{2.25}$$

Quanto maior o valor de $\alpha_1 + \beta_1$ menor é o decaimento de ρ_{a^2} depois do primeiro *lag*. Talvez a principal razão do GARCH(1,1) se ajustar bem a séries temporais financeiras seja o fato dele capturar o primeiro *lag* de autocorrelação e a subsequente taxa de decaimento.

É importante ressaltar que, ao compararem 330 modelos do tipo ARCH quanto à capacidade preditiva da variância condicional um passo a frente, Hansen e Lunde (2005) não encontraram evidências de que o modelo GARCH(1,1) possa ser superado por outro modelo para dados de taxa de câmbio. No entanto, para os dados do retorno da IBM, os autores encontraram evidências que o GARCH(1,1) é inferior a outros modelos.

Segundo Morettin e Toloi (2006), como a identificação da ordem do GARCH a ser ajustado a uma série real não é simples, recomenda-se o uso de ordens baixas como (1,1),(1,2),(2,1) e (2,2). Ademais, a escolha do modelo com melhor ajustamento deve ser feito com base nos critérios de AIC (*Akaike Information Criteria*) ou BIC (*Bayesian Information Criteria*), valores de alguma função de perda, log-verossimilhança ou assimetria e curtose.

Apesar da existência de vários métodos para estimação dos parâmetros de modelos GARCH, a estimação em geral é feita pelo estimador de máxima verossimilhança ou de quasi-máxima verossimilhança (QLM), assumindo que a distribuição do termo de erro é gaussiana (Fan *et al.*, 2014). Apesar do estimador de QLM com distribuição gaussiana ser consistente e assintoticamente normal, ele perde eficiência já que os retornos financeiros apresentam caudas pesadas, o que leva a violação da normalidade condicional do termo de erro (Fan *et al.*, 2014).

2.3.3 Extensões do GARCH

Modelos GARCH assimétricos, não-lineares e com distribuição não-normal foram introduzidos na literatura para capturar características não lineares dos retornos financeiros como caudas pesadas, efeito assimetria e excesso de curtose em relação a distribuição normal. Devido a utilização de um modelo SVR-GARCH (1,1), optou-se por mostrar expor algumas extensões do GARCH(1,1).

2.3.4 Distribuição do termo de erro z_t

Com o intuito de modelar as caudas pesadas da distribuição empírica dos retornos financeiros, é possível especificar diferentes distribuições para o termo de erro z_t de um modelo GARCH genérico (Morettin, 2011). Além da normal, dentre as mais utilizadas tem-se Marcucci (2005) : *t-Student*, *Generalized Error Distribution (GED)*, e *t-Student* assimétrica.

1. Uma variável aleatória X que segue uma distribuição t-Student possui a seguinte função densidade de probabilidade Casella e Berger (2001):

$$f(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\,\Gamma(\frac{\nu}{2})}\left(1 + \frac{x^2}{\nu}\right)^{(-\frac{\nu+1}{2})} \tag{2.26}$$

   em que ν é o número de graus de liberdade e Γ é a função gamma.

2. Distribuição de Erro Generalizada: segundo Morettin (2011), a densidade de uma variável aleatória X que segue uma GED com média zero e variância um é dada por:

$$f\left(x\right) = \frac{\nu exp[-\left(\frac{1}{2}\right)|(x/\lambda)|^{\nu}]}{\lambda 2^{(\nu+1/\nu)}\Gamma(1/\nu)}, \tag{2.27}$$

   em que:

$$\lambda = \left[\frac{2^{-(2/\nu)}\Gamma\left(1/\nu\right)}{\Gamma(3/v)}\right]^{1/2} \tag{2.28}$$

em que ν denota a espessura da cauda em relação a distribuição normal, satisfazendo $0 < \nu \leq \infty$. Quando $0 < \nu < 2$ a distribuição tem caudas mais pesadas que a normal.

3. Para modelar o excesso de curtose e os efeitos assimétricos Fernandez e Steel (1998) propuseram a distribuição t-Student assimétrica, que tem a seguinte função de densidade Morettin (2011):

$$f(x|\iota,\nu) = \frac{2}{\iota + 1/\iota}[g(\iota(sx+m)|\nu)I_{(-\infty,0)}(x+m/s)] \tag{2.29}$$

$$+ \frac{2}{\iota + 1/\iota}[g((sx+m)/\iota|\nu)I_{(0,+\infty)}(x+m/s)], \tag{2.30}$$

em que $g(./\nu)$ indica uma t-Student com ν graus de liberdade,

$$m = \frac{\Gamma\left(\left(\nu+1\right)/2\right)\sqrt{\nu-2}}{\sqrt{\pi}\Gamma\left(\nu/2\right)}(\iota - 1/\iota), \tag{2.31}$$

$$s = \sqrt{(\iota^2 + 1/\iota^2 - 1) - m^2} \tag{2.32}$$

em que ι é o parâmetro de assimetria.

Neste trabalho optou-se por estimar os modelos GARCH (1,1) com distribuição normal, GARCH (1,1) com distribuição t-Student, GARCH (1,1) com distribuição t-Student assimétrica, GARCH (1,1) com distribuição GED, EGARCH (1,1) com distribuição normal, EGARCH (1,1) com distribuição t-Student, EGARCH com distribuição t-Student assimétrica, EGARCH com distribuição GED, GJR (1,1) com distribuição normal e GJR (1,1) com distribuição t-Student, GJR (1,1) com distribuição t-Student assimétrica e GJR (1,1) com distribuição GED.

2.3.5 EGARCH

O GARCH tradicional trata choques negativos e positivos de maneira simétrica. No entanto, sabe-se que a volatilidade é maior na presença de retornos negativos. Assim, depois de choques negativos há mais volatilidade. Ou seja, a volatilidade reage de forma assimétrica aos retornos. Para modelar essa característica, Nelson (1991) introduziu o modelo GARCH exponencial (EGARCH). O EGACRH(1,1) é dado pela seguinte parametrização Morettin (2011):

$$a_t = \sqrt{h_t}z_t \tag{2.33}$$

$$\log(h_t) = \alpha_0 + \alpha_1 g(z_{t-1}) + \beta_1 \log(h_{t-1}) \tag{2.34}$$

em que z_t são variáveis aleatórias i.i.d com média zero e $g(.)$ é a curva de impacto de informação (Morettin, 2011):

$$g(z_t) = \theta z_t + \gamma\{|z_t| - E(|z_t|)\} \tag{2.35}$$

em que $E\{g(z_t)\} = 0$.

2.3.6 GJR

O modelo GJR-GARCH de Glosten *et al.* (1993) é similar ao TGARCH de Zakoian (1994) e é capaz de capturar a reação assimétrica da volatilidade aos retornos . O GJR (1,1) é dado pela seguinte parametrização Bollerslev (2008):

$$h_t = \alpha_0 + \alpha_1 a_{t-1}^2 + \gamma_1 a_{t-1}^2 I_{t-1} + \beta_1 h_{t-1} \tag{2.36}$$

$$I_{t-1} = \begin{cases} 1, & \text{se} \quad a_{t-1} < 0 \\ 0, & \text{caso contrário} \end{cases} \tag{2.37}$$

em que α_1, β_1 e γ_1 são parâmetros não-negativos e $I(.)$ é a função indicadora.

2.4 Modelo *random walk*

Um *random walk* é um processo não-estacionário com média constante, que considera que a melhor previsão da volatilidade do dia seguinte é dada pela volatilidade do dia anterior Dimson e Marsh (1990):

$$\hat{h}_t = h_{t-1} + \epsilon_t, \quad \epsilon_t \sim N(0,1) \tag{2.38}$$

em que $\hat{h_t}$ é a previsão da volatilidade e h_t é a volatilidade diária em $t-1$. É um modelo que apresenta excelente acurácia preditiva da volatilidade dos retornos financeiros, especialmente em relação aos modelos econométricos mais sofisticados (Brailsford e Faff, 1996; Dimson e Marsh, 1990).

Capítulo 3

Mistura finita de distribuições

> "*Mixtures of normals are a more general and flexible distribution for fitting phenomena exhibiting heavy tails and nonzero skewness, such as daily changes in market data. Mixtures of normals can properly fit the kurtosis and skewness often found in market variables.*"
>
> (Wang e Taaffe, 2015, p.193)

Misturas finitas de distribuições oferecem uma abordagem flexível para aprimorar a modelagem dos dados. Como qualquer distribuição pode ser bem aproximada por uma mistura finita de distribuições normais (Marron e Wand, 1992), é possível modelar dados cuja a distribuição seja desconhecida (McLachlan e Peel, 2000). A escolha apropriada dos componentes da mistura é capaz de modelar situações complexas em áreas como: biologia, medicina, engenharia, economia, física.

3.1 Mistura univariada de distribuições normais

Quando uma população estatística contém K subpopulações heterogêneas (também denominados regimes), é desejável o uso de misturas finitas de distribuição. Cada k é modelado por uma função densidade de probabilidade (fdp) oriunda de uma família de distribuição paramétrica. Em geral, é feito uma combinação linear das $fdps$. A fdp de cada k é o componente da mistura e o peso de cada uma na mistura é dada pela frequência relativa em relação à população. É importante ressaltar que o número de subpopulações pode ser conhecido ou desconhecido (McLachlan e Peel, 2000).

Seja $X = (X_1, \cdots, X_j)$ uma variável aleatória contínua de dimensão j e $x = (x_1, \cdots, x_j)$ uma observação de X . Assim, a função de densidade de probabilidade de uma mistura de distribuições é definida por uma combinação convexa de k fdps:

$$p(x \mid \Theta) = \sum_{k=1}^{K} \alpha_k p_k(x \mid \Theta_k), \quad \alpha_k \geq 0 \quad \text{e} \quad \sum_{i=1}^{k} \alpha_k = 1 \tag{3.1}$$

em que α_k são os pesos das misturas, $p_k(x \mid \Theta_k)$ é a fdp do k-ésimo componente e $\Theta = (\alpha_1, \alpha_k, \theta_1, \cdots, \theta_k)$ é o conjunto de parâmetros.

A função distribuição acumulada de k variáveis aleatórias gaussianas independentes $X_{i=1,\cdots,k}$ é dada por:

$$F(x) = \sum_{j=1}^{k} p_j \Phi(\frac{x - \mu_j}{\sigma_j}), \tag{3.2}$$

em que Φ é a função de distribuição acumulada $N(0,1)$(Wang e Taaffe, 2015). A função densidade de probabilidade de X é:

$$f(x) = \sum_{j=1}^{k} p_j \frac{1}{\sqrt{2\pi}\sigma_j} \exp \frac{-(x-\mu_j)}{2\sigma_j^2} \tag{3.3}$$

em que $0 \leq pj \leq 1$ e $\sum_{j=1}^{k} p_j = 1$. Suponha o caso em que a variável aleatória X é oriunda de uma mistura de duas distribuições normais em que :

$$x \sim N(\mu, \sigma_1^2)$$

$$x \sim N(\mu, \sigma_2^2)$$

Então, a densidade da mistura pode ser dada por:

$$f(x, p, \mu, \sigma_1, \sigma_2) = p\frac{1}{\sqrt{2\pi}\sigma_1} \exp \frac{-(x-\mu)}{2\sigma_1^2} + (1-p)\frac{1}{\sqrt{2\pi}\sigma_2} \exp \frac{-(x-\mu)}{2\sigma_2^2} \tag{3.4}$$

em que p está entre zero e um. No caso da figura 3.1, em que $p = \frac{1}{2}$.

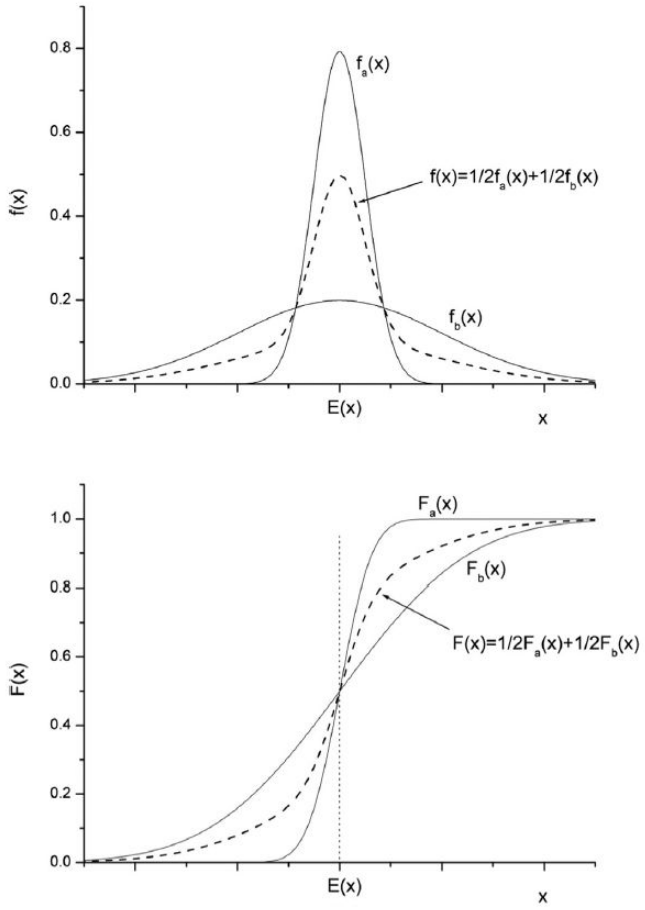

Figura 3.1: *Misturas de distribuições gaussianas. Fonte: Levy e Kaplanski (2015)*

3.2 Misturas de distribuições gaussianas em finanças

Como no longo prazo a distribuição dos retornos aproxima-se de uma distribuição normal, é habitual que modelos em finanças considerem que a distribuição dos retornos financeiros segue um processo estacionário gaussiano (Wirjanto e Xu, 2009). Não obstante, evidências empíricas demonstram que os retornos diários são leptocúrticos (possuem caudas pesadas) e assimétricos em torno da média em relação a curva Gaussiana. Dessa forma, o uso de misturas finitas de distribuições normais foi proposto para capturar alguns dos fatos estilizados das séries financeiras, pois qualquer distribuição contínua pode ser bem aproximada por uma mistura finita de distribuições normais (Wirjanto e Xu, 2009).

O mercado financeiro é um ambiente incerto e desafiador que muda de comportamento devido a uma série de fatores. A série temporal dos retornos é caracterizada por mudanças abruptas (quebras estruturais) em seus parâmetros Guidolin (2011). Diante disso, tanto acadêmicos como profissionais de mercado destacam a existência da oscilação de regimes ou

estados no mercado financeiro (Bae *et al.*, 2014; BenSaïda, 2015).

Segundo Levy e Kaplanski (2015), mesmo que a a distribuição do retorno de cada um dos regimes seja normal, a distribuição global, dado a probabilidade de cada regime, não é normal. Em verdade, ela será uma mistura de normais. Os regimes de mercado podem ser ocasionados por crises financeiras, ciclo de negócios e/ou mudanças abruptas na política fiscal e monetária (Levy e Kaplanski, 2015). Os modelos desenvolvidos para capturar a presença de regimes nas séries financeiras assumem, em geral, a existência de dois regimes (alta e baixa volatilidade) com a distribuição de parâmetros bem definida e probabilidades de transição entre estados [1]. No entanto, o mercado pode apresentar mais de dois regimes. Assim, dado a existência de k regimes, utiliza-se k distribuições normais para modelar cada um dos regimes. Segundo Guidolin (2011), alguns estudos utilizam misturas de até 8 normais para capturar os regimes.

Além do excesso de curtose e assimetria, os retornos financeiros apresentam aglomeração de volatilidade e variação da volatilidade ao longo do tempo. Não obstante, segundo Wirjanto e Xu (2009), os modelos de misturas de normais não foram desenvolvidos para capturarem essas duas características. Assim, modelos de volatilidade condicional que variam no tempo como ARCH e GARCH foram propostos para tal tarefa. Apesar de capturarem as aglomerações de volatilidade, evidências empíricas mostram que o GARCH com inovações seguindo uma distribuição normal ou mesmo uma distribuição com causas pesadas (como *t-Student*, por exemplo) não é capaz de capturar toda a extensão da assimetria e curtose observada na série dos retornos financeiros (Bai *et al.*, 2003). Para contornar esse problema, foram propostos modelos GARCH em que a distribuição da inovação é uma misturas de normais, dando origem ao Mixed Normal GARCH (GARCH-MN), como por exemplo nos trabalho de Wong e Li (2001), Haas *et al.* (2004), Alexander e Lazar (2006). Além disso, para capturar os regimes de volatilidade, foram desenvolvidos modelos GARCH com mudanças de regime markoviano (BenSaïda, 2015; Guidolin, 2011; Marcucci, 2005). Devido a isso, utiliza-se uma mistura de funções núcleos gaussianas no SVR-GARCH para modelar as mudanças de regimes.

[1] Ang e Timmermann (2012) apontam algumas razões para os modelos de mudanças de regime serem utilizados para modelagem de séries financeiras como: habilidade de capturar vários fato estilizados das séries financeiras como caudas pesadas, assimetria, correlações tempo-variantes, efeitos ARCH.

Capítulo 4

Teoria do aprendizado estatístico e métodos de kernels

> "*Statistical learning theory does not belong to any specific branch of science: it has its own goals, its own paradigm, and its own techniques. Statisticians (who have their own paradigm) never considered this theory as part of statistics* ".
>
> Vapnik (1998, p. 720)

O aprendizado de máquina está presente numa gama diversas de empresas, produtos e negócios. Os algoritmos de aprendizagem, conhecidos como aprendizes, realizam inferências dos dados. A principal característica deles é a capacidade de escreverem seus próprios programas. Ou seja, criar novos algoritmos (Domingos, 2015). Segundo Domingos (2015), há cinco tribos (ou escolas de pensamento) em *Machine Learning* e cada uma delas tem um algoritmo mestre: Simbolistas (dedução inversa), Evolucionários (programação genética), Bayesianos (inferência bayesiana), Conexionistas (*backpropagation*) e Analogistas (máquina de suporte vetorial).

As técnicas de aprendizado de máquina têm por objetivo fazer com que um máquina seja capaz de realizar tarefas seguindo algum algoritmo de aprendizado. Para tanto, é necessário construir algoritmos que possam descobrir relações subjacentes, regularidades ou estruturas inerentes aos dados, ou seja, aprender padrões dos dados. Para isso, empregam o princípio da indução. O aprendizado é visto como um problema de inferência com uma amostra de dados de grande dimensão e cheios de ruído. O problema de aprendizado pode ser descrito da seguinte forma: dado uma amostra limitada de exemplos, a máquina deve inferir um regra geral que seja capaz de explicar os exemplos conhecidos e que seja capaz de generalizar para novos exemplos. O aprendizado de máquina trata de três grandes problemas: classificação, regressão e estimação de densidade (Vapnik, 1998). Uma definição mais formal de *Machine learning* é dada por (Mitchell, 1997):

Definição 4.0.1. Um algoritmo computacional $\mathcal{A}$ é dito aprender dos dados (ou experiência) $\mathcal{D}$ com relação a alguma classe de tarefas T e uma medida de desempenho $\mathcal{L}$, se a sua performance nas tarefas T, medida por $\mathcal{L}$, melhora com a experiência $\mathcal{D}$.

Em geral, os problemas de aprendizado de máquina podem ser divididos em três grandes grupos: aprendizado supervisionado, aprendizado não-supervisionado e aprendizado por reforço (Herbrich, 2001). Neste estudo trabalha-se com aprendizado supervisionado, que pode ser descrito da seguinte forma: dado uma amostra de treinamento $(x_1, y_1), \cdots (x_n, y_n)$ com pares de objeto rotulados (classes ou valores reais), a máquina deve encontrar uma função ótima $f : \mathcal{X} \to \mathcal{Y}$ que produza a saída correta para qualquer entrada com o menor erro possível. É importante ressaltar que há uma grande diferença do tratamento dos dados pela Estatística e pelas técnicas de *Machine learning* (Breiman, 2001): os estatísticos assumem um modelo (ex: regressão linear ou logística) para modelar os dados, enquanto as técnicas de aprendizado de máquina assumem que o mecanismo de geração dos dados é complexo e desconhecido, assim o algoritmo de *Machine learning* usa os dados de *input* para prever o *output*, realizando poucas suposições sobre o processo gerador dos dados (Breiman, 2001).

No contexto da tribo dos Analogistas, a Teoria do Aprendizado Estatístico fornece a base teórica de seu algoritmo mestre e provê os elementos teóricos e práticos que permitam retirar conclusões válidas dos dados empíricos. Nas últimas duas décadas, os métodos de *kernel* (ou simplesmente *kernels*) ganharam muita atenção dos pesquisadores da área de aprendizado de máquina devido a sua capacidade de mapear os dados para um espaço de alta dimensão, aumentando o poder computacional dos algoritmos lineares. O *kernel* é equivalente a um produto interno num espaço característico de grande dimensão, onde métodos lineares são utilizados para modelagem dos dados. Qualquer algoritmo que dependa dos dados apenas pelo produto interno é um método de *kernel*.

As vantagens teóricas e computacionais dos métodos de *kernel* para aprendizado de padrões podem ser explicadas pela habilidade em combinar programação matemática, teoria de aprendizado de máquina e análise funcional. Os *kernels* são utilizados em diversas áreas da ciência, como: matemática, estatística, medicina, engenharia, computação etc. O objetivo deste capítulo é dar uma visão intuitiva das ideias e conceitos que serão utilizados neste trabalho. Todas as provas dos resultados estabelecidos aqui podem ser encontradas nas seguintes referências: Steinwart e Christmann (2008), Schölkopf e Smola (2002), Herbrich (2001) e Luxburg e Schölkopf (2008).

4.1 Teoria do aprendizado estatístico

A principal motivação para a Teoria do Aprendizado Estatístico (TAE)[1] é prover os fundamentos matemáticos dos algoritmos de aprendizado de máquina. A TAE surgiu nos anos de 1960 e teve como fundadores os pesquisadores russos Vladimir Vapnik e Alexey Chervonenkis, por isso também é conhecida como Teoria de Vapnik e Chervonenkis. Não obstante, somente nos idos dos anos de 1990 ganhou popularidade devido ao surgimento das máquinas de suporte vetorial (*Support Vector Machine*(SVM)) em seu formato atual (Vapnik, 1999).

Dado um espaço de entrada $\mathcal{X}$ e um espaço de saída $\mathcal{Y}$ oriundos de uma distribuição de probabilidade conjunta $\mathcal{D}$ sobre $\mathcal{X} \times \mathcal{Y}$ e de posse de um conjunto de exemplos que estão rotulados (denominado dados de treinamento ou conjunto de entrada) S=$((x_1, y_1)...(x_m, y_m))$ amostrados de maneira independente de $\mathcal{D}$[2], o objetivo do algoritmo de aprendizagem é encontrar uma função (de alguma classe de funções $\mathcal{F}$) f: $\mathcal{X} \to \mathcal{Y}$ [3] que tenha uma perda esperada baixa para um conjunto de dados desconhecidos e amostrados de $\mathcal{D}$ (Luxburg e Schölkopf

[1] Nesta seção segue-se de perto as explicações presentes em Luxburg e Schölkopf (2008).

[2] Muitas vezes é útil denotar $\mathcal{D} = \mathcal{D}_x \times \mathcal{D}_{y/x}$. Para o uso do aprendizado de máquina na previsão de séries temporais a hipótese de independência é relaxada.

[3] f é denominada regra de decisão ou regra de classificação.

, 2008), ou seja, tenha boa capacidade de generalização. Ademais, a TAE estabelece que:

- Nenhuma suposição é feita sobre a distribuição de $\mathcal{D}$;
- $\mathcal{D}$ é fixa, não se altera ao longo do tempo;
- No momento da aprendizagem, $\mathcal{D}$ é desconhecida pela máquina;
- Devido a ruídos e sobreposição de classes, os rótulos não são determinísticos;

Após a máquina encontrar o classificador f, é preciso mensurar sua qualidade na classificação dos objetos desconhecidos. Para isso, utiliza-se a função de perda ℓ, que mensura a diferença entre o rótulo previsto e o real. No caso de classificação, a função mais simples é dada por Luxburg e Schölkopf (2008):

$$\ell(f(x), y) = \begin{cases} 1 & : f(x) \neq y \\ 0 & : f(x) = y \end{cases}$$

Para o problema de regressão, a função de perda quadrática é muito utilizada: $\ell(f(x), y) = (y - f(x))^2$.

A função de perda mensura o erro de um ponto específico. No entanto, é possível calcular a perda esperada da função f de todos pontos $x \in X$ gerados por $\mathcal{D}$, denominado o risco esperado (erro verdadeiro ou erro de generalização) de f (Luxburg e Schölkopf, 2008):

$$R(f) = E(\ell(f(x), y)) \tag{4.1}$$

O objetivo do aprendizado estatístico é encontrar a função $f \in \mathcal{F}$ que minimize o risco esperado $R(f)$ da função de perda $\ell(f(x), y)$. No entanto, como $\mathcal{D}$ é desconhecido pela máquina, não é possível calcular o risco esperado. Assim, aproxima-se o risco esperado por meio do risco empírico (denominado também erro de treino). Então, busca-se inferir uma função f que minimize o risco empírico $R_{emp}(f)$ na amostra de treinamento. Dessa maneira, o objetivo do princípio indutivo da minimização empírica do risco (ERM) é encontrar um classificador f_n tal que (Luxburg e Schölkopf, 2008):

$$f_n = argmin_{f \in \mathcal{F}} R_{emp}(f) \tag{4.2}$$

Segundo Vapnik (1995), a Teoria do Aprendizado Estatístico deve responder as seguintes questões[4]:

1. Quais são as condições necessárias e suficientes para a consistência (teoria assintótica) do processo de aprendizagem baseada no princípio da minimização empírica do risco (ERM)?
2. Qual é a taxa de convergência do processo de aprendizagem? Como a capacidade de generalização melhorara à medida que a amostra aumenta?
3. Como é possível controlar a taxa de convergência (habilidade de generalização) da aprendizagem?
4. Como é possível construir algoritmos que controlam a habilidade de generalização? Ou seja, existe alguma estratégia que garante, mensura e controla a capacidade de generalização do modelo de aprendizagem?

[4]As questões foram colocadas apenas para dar a motivação da TAE. Este trabalho não tem a intenção de respondê-las. As respostas podem ser encontradas em Vapnik (2006).

4.1.1 Características do espaço de funções

Seja $\mathcal{F}$ o espaço de funções que o algoritmo de aprendizagem encontrará a melhor função de acordo com algum critério. Um algoritmo de aprendizagem realiza o mapeamento dos dados para $\mathcal{F}$. Considere que $\mathcal{F}_{todas}$ contém todas as possíveis funções que mapeiam $\mathcal{X} \rightarrow \mathcal{Y}$. Dentro desse conjunto de funções é possível definir o classificador ótimo, denominado classificador de Bayes (Luxburg e Schölkopf, 2008):

$$f_{Bayes} = \begin{cases} 1, & \text{se} \quad P(Y = 1/X = x) \geq 0.5, \\ -1, & \text{caso contrário} \end{cases} \tag{4.3}$$

Por ser o melhor classificador, f_{Bayes} possui o menor risco esperado, denominado risco de Bayes. Porém, como a distribuição de probabilidade $\mathcal{D}$ é desconhecida da máquina, não é possível calcular o classificador de Bayes. Como não se tem acesso a esse classificador, deseja-se encontrar uma função f que tenha um risco $R(f)$ o mais próximo possível do risco da função ótima (o classificador de Bayes).

4.1.2 Generalização e consistência

Como não há conhecimento sobre $\mathcal{D}$, o $R(f)$ de um classificador f qualquer não pode ser calculado. Porém, é possível calcular o erro cometido por uma função na amostra de treinamento, denominado de erro ou risco empírico (Luxburg e Schölkopf, 2008):

$$R_{emp}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) \tag{4.4}$$

Normalmente, um classificador f aprendido num conjunto de treino particular, possui um risco empírico baixo. No entanto, nada garante que uma função f que produz poucos erros no conjunto de treinamento S, terá um bom desempenho para dados que não pertencem a S (Luxburg e Schölkopf, 2008). Um classificador f_n tem boa capacidade de generalização se a diferença $|R(f_n) - R_{emp}(f_n)|$ é pequena. Isso não implica que f_n tem necessariamente um erro empírico baixo, apenas mostra que $R_{emp}(f_n)$ é uma boa estimativa do verdadeiro erro $R(f)$ (Luxburg e Schölkopf, 2008).

Outro conceito importante da TAE é o de consistência de um conjunto de funções[5]. Um algoritmo de aprendizado quando apresentado a um número crescente de dados de treinamento, deve eventualmente convergir para uma solução ótima, ou seja, está se aproximando de melhor performance de previsão à medida que a amostra aumenta (Luxburg e Schölkopf, 2008).

Suponha que um algoritmo com base numa amostra de treinamento de tamanho n encontre o melhor classificador f_n num espaço funcional $\mathcal{F}$. O melhor classificador em $\mathcal{F}$ é aquele que possui o menor risco. Para demonstrar o conceito de consistência, assume-se que esse classificador é único e é denotado por $f_{\mathcal{F}}$ (Luxburg e Schölkopf, 2008). Além disso, seja $\mathcal{F}_{todas}$ como o espaço que contém o melhor classificador de todos, denominado classificador de Bayes: f_{Bayes}. No entanto, como a máquina desconhece esse classificador, pois provavelmente ele não está no subespaço $\mathcal{F}$. Então, $R(f_{\mathcal{F}}) \geq R(f_{Bayes})$. Com esses conceitos, pode-se construir diferentes tipos de consistência (Luxburg e Schölkopf, 2008, p.7):

Definição 4.1.1. Seja f_n uma função aprendida com base numa amostra n retirada de uma sequência de infinita de pontos de treinamento oriunda de uma distribuição de probabilidade

[5] Veja capítulo 2 Vapnik (1995).

$\mathcal{D}$:

1. Se o risco $R(f_n)$ converge em probabilidade ao risco $R(f_\mathcal{F})$ do melhor classificador em $\mathcal{F}$, $\forall\epsilon > 0$, o algoritmo é consistente em relação a $\mathcal{F}$ e $\mathcal{D}$:

$$P(R(f_n) - R(f_\mathcal{F}) > \epsilon) \to 0, \quad \text{conforme} \quad \text{n} \to \infty \tag{4.5}$$

2. Se o risco $R(f_n)$ converge em probabilidade ao risco $R(f_{Bayes})$, $\forall\epsilon > 0$, o algoritmo é Bayes consistente:

$$P(R(f_n) - R(f_{Bayes})) > \epsilon) \to 0, \quad \text{conforme} \quad \text{n} \to \infty \tag{4.6}$$

3. Se o algoritmo de aprendizado for consistente em relação a $\mathcal{F}$ para qualquer $\mathcal{D}$, ele é universalmente consistente com respeito $\mathcal{F}$.

Os resultados acima exigem convergência do verdadeiro risco $R(f_n)$. Como o risco empírico é um estimador do risco real, então é preciso exigir a convergência do risco empírico. Porém, segundo Luxburg e Schölkopf (2008) não é exigido uma convergência explícita do risco empírico, pois ela surge como um efeito colateral da consistência.

4.1.3 Erro de aproximação e estimação

Considere o espaço $\mathcal{F}_{todas}$ de todas as possíveis funções. Suponha um subespaço $\mathcal{F}$ que possui poucas funções. Nesse caso, a variância é baixa, mas o viés é grande, pois o número de classificadores que é possível obter para um problema é baixo. Caso o $\mathcal{F}$ seja grande e contenha muitas funções, a variância é grande, mas o viés é menor (Luxburg e Schölkopf , 2008). É possível decompor a consistência de Bayes da seguinte forma:

$$R(f_n) - R(f_{Bayes}) = \underbrace{(R(f_n) - R(f_\mathcal{F}))}_{\text{erro de estimação}} + \underbrace{R(f_\mathcal{F}) - R(f_{Bayes})}_{\text{erro de aproximação}} \tag{4.7}$$

O erro de estimação é resultado da incerteza existente nos dados de treinamento. Ele mensura a variação do risco da função f_n, estimada na amostra. O erro de aproximação é resultado do viés do algoritmo de aprendizagem. Ele mensura o viés introduzido no modelo ao escolher uma classe de funções pequena (Luxburg e Schölkopf, 2008).

Percebe-se que através do espaço $\mathcal{F}$, é possível realizar o balanceamento entre o erro de estimação e aproximação. Assim, se for escolhido um espaço $\mathcal{F}$ grande, o erro de aproximação será pequeno, mas o erro de estimação será grande, pois $\mathcal{F}$ conterá funções complexas, o que levará ao subajustamento dos dados. Se espaço $\mathcal{F}$ for pequeno, o erro de estimação é menor, mas o erro de aproximação é grande (Luxburg e Schölkopf, 2008).

4.1.4 Princípio da minimização empírica do risco

O princípio indutivo da minimização do risco é geral: métodos como da máxima-verossimilhança e mínimos quadrados são realizações desse princípio (Vapnik, 1992). A motivação do Princípio da Minimização Empírica do Risco (*Empirical Risk Minimization*, ERM) foi a Lei dos Grandes Números. Essa lei estabelece que, sob algumas condições, a média de variáveis aleatórias ξ_i que foram amostradas de maneira independente e identicamente distribuída de uma distribuição de probabilidade qualquer converge para o seu valor esperado à medida que o tamanho da amostra aumenta (Luxburg e Schölkopf, 2008):

$$\frac{1}{n}\sum_{i=1}^{n}\xi_i \to E(\xi), \quad \text{conforme n} \to \infty \tag{4.8}$$

Pela Lei dos Grandes Números pode-se concluir que para uma função fixa f, o risco empírico converge para o risco esperado à medida que o tamanho amostral tende ao infinito:

$$R_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n}\ell\left(x_i, y_i, f(x_i)\right) \to E(\ell(x, y, f(x)) \quad \text{para n} \to \infty \tag{4.9}$$

Dessa forma, com o erro empírico é possível aproximar muito bem o risco esperado. Com o uso da desigualdade de Chernoff (1952), estendida por Hoeffding (1963) é possível caracterizar quão bem uma média amostral (ou empírica) se aproxima do valor esperado (Luxburg e Schölkopf, 2008):

$$P\left(|\frac{1}{n}\sum_{i=1}^{n}\xi_i - E(\xi)| \geq \epsilon\right) \leq 2\exp\left(-2n\epsilon^2\right) \tag{4.10}$$

em que ξ_i são variáveis aleatórias. Em outras palavras, a desigualdade de Chernoff (1952) indica que a probabilidade da média amostral se desviar do seu valor esperado em mais de ϵ é limitada por uma pequena quantidade $2\exp(-2n\epsilon^2)$. Assim pode-se usar essa desigualdade para obter um limite que define o quanto o risco empírico se aproxima do risco esperado para um f fixo:

$$P(|R_{emp}(f) - R(f)| \geq \epsilon) \leq 2\exp(-2n\epsilon^2)$$

Aparentemente, o limite de Chernoff (1952) é suficiente para provar a consistência do Princípio da Minimização Empírica do Risco. No entanto, ele só é válido para função fixa f que não depende dos dados de treinamento. Sabe-se, porém que f é obtida com base no conjunto de treinamento. Sendo assim, isso invalida o uso da Lei dos Grandes Números para provar que o risco empírico pode ser um bom estimador para o risco esperado e, por conseguinte, leva a inconsistência da Minimização Empírica do Risco (Luxburg e Schölkopf, 2008).

Para tornar o ERM consistente é preciso restringir o espaço de funções admissíveis onde f é escolhido. Em aprendizado de máquina, essa questão é levada em conta por meio da complexidade (ou capacidade) do espaço de funções (Luxburg e Schölkopf, 2008).

4.1.5 Convergência uniforme

Segundo Vapnik (1992), a avaliação da solidez do Princípio da Minimização Empírica do Risco (ERM) exige as respostas das seguintes questões:

- O princípio é consistente? Em outras palavras, o risco empírico converge uniformemente para o risco esperado para todo o conjunto de funções?
- Qual é a taxa de convergência?

A convergência uniforme do conjunto de todas as funções é a condição necessária e suficiente para a consistência do ERM. A teoria do Aprendizado Estatístico mostrou que a consistência da minimização empírica do risco é determinada pelo comportamento do pior caso de todas as funções $f \in \mathcal{F}$ que a máquina pode escolher (Luxburg e Schölkopf, 2008). Se o risco empírico converge para o risco esperado para a pior função, então ele converge para as demais funções em $\mathcal{F}$.

Uma maneira de garantir essa convergência para toda $f \in \mathcal{F}$ é através da convergência uniforme sobre $\mathcal{F}$. Dado uma amostra n suficientemente grande, $\forall\ f \in \mathcal{F}\ |R(f) - R_{emp}(f)|$ deve ser menor que ϵ. De maneira mais formal (Luxburg e Schölkopf, 2008):

$$\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \leq \epsilon \tag{4.11}$$

Assim, para qualquer função $f \in \mathcal{F}$, tem-se que (Luxburg e Schölkopf, 2008):

$$|R(f) - R_{emp}(f)| \leq \sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \tag{4.12}$$

Então para uma função f_n escolhida com base num conjunto de treinamento, pode-se concluir que (Luxburg e Schölkopf, 2008):

$$P(|R(f_n) - R_{emp}(f_n)| \geq \epsilon) \leq P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \geq \epsilon) \tag{4.13}$$

Então, a Lei dos Grandes Números permanece uniforme para uma classe de funções $\mathcal{F}$ para todo $\epsilon \geq 0$,

$$P(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \geq \epsilon) \to 0 \quad \text{quando} \quad n \to \infty \tag{4.14}$$

É possível mostrar por meio de 4.13 que, se a Lei Uniforme dos Grandes Números[6] é válida para algum $\mathcal{F}$, então o princípio da minimização empírica do risco é consistente em relação a $\mathcal{F}$ (Luxburg e Schölkopf, 2008).

$$P\left(|R(f_n) - R(f_{\mathcal{F}})| \geq \epsilon\right) \leq P\left(\sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| \geq \frac{\epsilon}{2}\right) \tag{4.15}$$

Sob a Lei dos Grandes Números, o lado direito dessa desigualdade tende a zero, o que mostra a consistência da Minimização Empírica do Risco (ERM) (Luxburg e Schölkopf , 2008). Em outras palavras, a convergência uniforme sobre $\mathcal{F}$ é uma condição suficiente para o ERM sobre $\mathcal{F}$. A teoria de VC mostrou que a convergência uniforme também é uma condição necessária (Luxburg e Schölkopf, 2008).

Apesar da convergência uniforme ser teoricamente bem fundamentada, é muito difícil saber se a Lei Uniforme dos Grandes Números se aplica a um determinado conjunto de classes (Luxburg e Schölkopf, 2008).

4.1.6 Medidas de capacidade e limites de generalização

Um modelo com baixa capacidade não é capaz de aprender nem os dados de treinamento, enquanto um modelo muito complexo com alta capacidade não possui boa capacidade generalização para dados desconhecidos (Luxburg e Schölkopf, 2008). É possível denotar o risco esperado por (Luxburg e Schölkopf, 2008):

$$R(f) \leq R_{emp}(f) + \text{Capacidade}(\mathcal{F}) \tag{4.16}$$

Dado um conjunto de treinamento S, o objetivo do algoritmo de aprendizagem é produzir uma função f_n com base nesses dados que tenha um risco $R(f_n)$ baixo. Esse risco é uma variável aleatória que não pode ser computada com os dados, pois a distribuição D é desconhecida pela máquina. Assim, a estimativas de $R(f_n)$ tem a forma de limites probabilísticos.

[6] Para mais detalhes, veja p.414 Vapnik (2006)

Esses limites permitem uma melhor compreensão de quais propriedades da classe de funções determinam a existência da convergência uniforme (Luxburg e Schölkopf, 2008).

Existem várias medidas de capacidade para uma classe de funções, como por exemplo: a dimensão VC e o coeficiente de quebra. Além disso, o limite de generalização é um limite probabilístico do erro de generalização com probabilidade de $(1-\delta)$ e tem a seguinte forma geral (Luxburg e Schölkopf, 2008):

$$R(f) = R_{emp}(f) + \text{Capacidade}(\mathcal{F}) + \text{intervalo de confiança}(\delta) \tag{4.17}$$

É importante ressaltar que minimizar somente o risco empírico não garante uma boa capacidade de generalização. Por isso, é necessário minimizar a soma do risco empírico com algum intervalo de confiança.

4.1.7 Coeficiente de quebra

Com os *insights* da simetrização, Vapnik e Chervonenkis derivaram a primeira medida de capacidade (ou complexidade) de uma classe de funções. Seja $Z_n = ((x_1, y_1), \cdots, (x_n, y_n))$ uma amostra de treinamento de tamanho n. Seja $|F_{z_n}|$ a cardinalidade de $\mathcal{F}$ para o conjunto de exemplos Z_n, isto é, o número de funções que produzem classificações distintas para Z_n. Assim, o número máximo de funções que produzem classificações distintas é dado por (Luxburg e Schölkopf, 2008):

$$\mathcal{N}(\mathcal{F}, n) = \max\{|F_{Z_n}|X_1, \cdots, X_n \in \mathcal{X}\} \tag{4.18}$$

$\mathcal{N}(\mathcal{F}, n)$ é denominado coeficiente de *shattering* (quebra) da função de classe $\mathcal{F}$ em relação à amostra de tamanho n (Luxburg e Schölkopf, 2008). Dessa maneira, esse coeficiente permite medir o tamanho e/ou complexidade da classe de funções (Luxburg e Schölkopf, 2008).

4.1.8 Dimensão VC

A dimensão de Vapnik Chernovenkis (VC) é uma das mais importantes medidades de capacidade. A dimensão VC de uma classe de funções $\mathcal{F}$ é definida pelo número máximo de pontos que pode ser classificado de todas as maneiras possíveis por $\mathcal{F}$. Em outras palavras, a dimensão VC mede a capacidade (ou complexidade) de um espaço de funções, que tem por objetivo caracterizar o crescimento do coeficiente de quebra usando apenas um número (Luxburg e Schölkopf, 2008).

Diz-se que uma amostra Z_n de tamanho n é quebrada por uma classe de funções $\mathcal{F}$, se tal classe pode realizar qualquer classificação numa dada amostra, ou seja, a cardinalidade de $\mathcal{F}_{Z_{2n}} = 2^n$ (Luxburg e Schölkopf, 2008). Então, a dimensão de Vapnik e Chervonenkis de $\mathcal{F}$ é definida como o maior número n tal que há uma amostra de tamanho n que pode ser quebrada por $\mathcal{F}$ (Luxburg e Schölkopf, 2008):

$$VC(\mathcal{F}) = \max\{n \in \mathbb{N}|\mathcal{F}_{Z_{2n}}| = 2^n \quad \text{para algum} \quad Z_n\} \tag{4.19}$$

Portanto, se a dimensão VC de uma classe de funções em $\mathcal{F}$ é finita, sabe-se que à medida que a amostra aumenta, o coeficiente de quebra cresce polinomialmente. O que implica na consistência do Princípio de Minimização do Risco Empírico, isto é, aprendizado. Em outras palavras, um algoritmo de aprendizagem será consistente (capacidade de generalização) se, e somente se a função f é oriunda de uma classe de funções $\mathcal{F}$ com dimensão VC finita (Luxburg e Schölkopf, 2008).

Com base na teoria da convergência uniforme, Vapnik (1982) fornece um limite sobre desvio do risco empírico ao risco esperado, que é dado pela soma do risco empírico e um termo de capacidade que pode ser garantido com probabilidade $(1-\eta)$, em que $\eta \in [0,1]$:

$$R(f) \leq R_{emp}(f) + \underbrace{\sqrt{\frac{h\left(\ln\frac{2N}{h}+1\right) - ln\left(\frac{\eta}{4}\right)}{N}}}_{\text{Intervalo de Confiança VC}} \tag{4.20}$$

em que h é a dimensão VC de $\mathcal{F}$, N é o número de exemplos de treinamento. À medida que a razão $\frac{N}{h}$ cresce, o termo de capacidade diminui e o risco esperado (erro de teste) se aproxima do risco empírico (erro de treino). De maneira mais simples:

$$\text{Erro de teste} \leq \text{Erro de treino} + \text{Complexidade do conjunto de funções} \tag{4.21}$$

Assim, é possível mensurar a melhoria da capacidade de generalização à medida que a amostra aumenta.

4.1.9 Limites para margens largas

Uma outra medida de capacidade é de limite de margens largas. Seja um conjunto de pontos num espaço $\mathbb{R}^2$, que se deseja separar em classes com uma linha reta. Dado um conjunto de pontos rotulados e um classificador f_n capaz de separá-los perfeitamente, a margem de f_n pode ser definida como a menor distância entre qualquer ponto e a linha de separação f_n (Luxburg e Schölkopf, 2008).

A dimensão VC de uma classe de funções lineares $\mathcal{F}_\rho$ num espaço arbitrário $\mathbb{R}^d$ de dimensão arbitrária d com uma margem ρ pode ser limitada pela razão do raio R da menor esfera em torno dos pontos com a margem ρ:

$$VC(\mathcal{F}_\rho) \leq \min\left\{d, \frac{4R^2}{\rho^2}\right\} + 1$$

Quanto maior a margem ρ de $\mathcal{F}_\rho$, menor é a dimensão VC. Então, a complexidade do classificador se mantém baixa independente da dimensão d. Portanto, a margem de um classificador pode ser usada como medida de capacidade. A construção do SVM foi motivada por esse resultado (Luxburg e Schölkopf, 2008).

4.1.10 Regularização

A escolha da classe de funções $\mathcal{F}$ é fundamental para o uso do Princípio da Minimização Empírica do Risco. Caso $\mathcal{F}$ seja grande, o risco empírico será baixo, mas o risco de generalização será grande. Assim, com o ERM corre-se o risco do sobreajustamento dos dados de treinamento (Luxburg e Schölkopf, 2008). O ERM é um problema mal colocado (*ill-posed problem*), pois uma pequena mudança no conjunto de treinamento pode gerar uma grande mudança na função estimada, o que gera soluções instáveis. Por meio da regularização é possível resolver o problema do sobreajustamento e da estabilidade da solução (Luxburg e Schölkopf, 2008).

Considere um espaço $\mathcal{F}_n$ formado por uma sequência crescente de espaços funcionais $\mathcal{F}_1$, $\mathcal{F}_2, \cdots, \mathcal{F}_k$. Assim, dado uma amostra n, a máquina deve buscar uma função f_n em $\mathcal{F}_n$ que possui o menor risco empírico e, em seguida, calcular a capacidade de generalização usando alguma medida de capacidade. Segundo Luxburg e Schölkopf (2008), uma forma implícita de trabalhar com espaços funcionais combinados é por meio do princípio da regularização, que visa minimizar o risco regularizado:

$$R_{reg}(f) = R_{emp}(f) + \lambda\Omega(f) \tag{4.22}$$

em que $\Omega(f)$ é o regularizador, que é uma maneira de penalizar funções muito complexas, λ realiza o *trade-off* entre o risco empírico e o regularizador. Caso λ seja grande, a penalização dada por $\Omega(f)$ tem grande importância. Então, é preferível funções que tenha um $\Omega(f)$ pequeno, mesmo que ela tenha um risco empírico grande (Luxburg e Schölkopf, 2008). É importante destacar que classificadores que estão baseados na minimização do risco regularizado podem aprender de forma consistente (assintótica), entre eles as máquinas de suporte vetorial (Steinwart, 2005).

4.1.11 Princípio da minimização estrutural do risco

O Princípio da Minimização Empírica do Risco é destinado a tratar de grandes amostras (Vapnik, 1999). No entanto, quando a amostra é pequena, um risco empírico baixo não garante um risco esperado baixo. Assim, a minimização do risco esperado exige um novo princípio baseado na minimização simultânea de um termo que depende do valor do risco empírico e outro que dependa da dimensão VC do conjunto de funções (Vapnik, 1999).

Uma das grandes preocupações dos algoritmos de aprendizagem de máquina é encontrar uma função que tenha boa capacidade de generalização. Muitas vezes uma função realiza um sobreajustamento dos dados de treinamento, o que leva a uma baixa capacidade de generalização. Para contornar esse problema, o princípio da Minimização Estrutural do Risco (*Structural Risk Minimization*, SRM) tem por objetivo encontrar uma função que minimize, simultaneamente, o risco empírico e a dimensão VC (dada por um termo que mede a complexidade do espaço de funções) Luxburg e Schölkopf (2008):

$$R(f) \leq R_{emp}(f) + \text{Termo de Capacidade}$$

O SRM pode ser descrito da seguinte forma (Sewell, 2008):

1. Com base no conhecimento prévio do problema escolha alguma classe de funções $\mathcal{F}$;
2. Divida $\mathcal{F}$ numa hierarquia de subconjuntos combinados em aumento crescente de complexidade: $\mathcal{F}_1 \in \mathcal{F}_2 \in \cdots \mathcal{F}_K$ com dimensões VC não-descrescentes ($h_1 \leq h_2 \leq \cdots h_k$);
3. Para cada subconjunto F_i, encontre a função f_i que minimize o risco empírico; e
4. Selecione a função (ou modelo) em que a soma do risco empírico e o termo que mede a complexidade da classe de funções seja mínima. Ou seja, escolha a classe de funções $\mathcal{F}_i$ e o respectivo f_i que minimize o lado direito da equação 4.20.

O SRM consiste em encontrar o subconjunto de funções que minimiza o limite sobre o risco esperado. Por conseguinte, esse princípio garante, mensura e controla a capacidade de generalização do algoritmo de aprendizagem. Na Figura 4.1 fica evidente que quando a máquina tem uma grande capacidade (dimensão VC grande), ela apresenta um risco empírico baixo, mas não generaliza bem, pois o intervalo de confiança VC é grande. Com o uso

do limite sobre o risco esperado, é possível escolher a função que tenha o menor erro de generalização (Luxburg e Schölkopf, 2008).

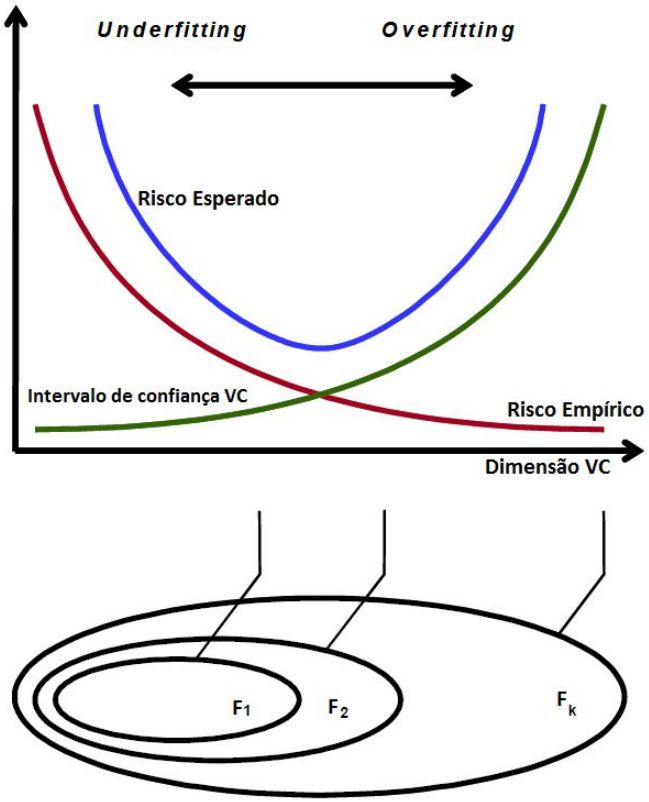

Figura 4.1: *Limite do risco esperado de uma máquina de aprendizado. Fonte: adaptado de Cherkassky e Mulier (2007).*

4.2 Função kernel

Dado um problema de reconhecimento de padrões não-lineares (regressão ou classificação), qualquer algoritmo que utilize o produto interno como medida de similaridade, pode ser substituído por um *kernel* de Mercer, que transforma os dados do espaço de entrada original para um espaço de maior dimensão (denominado espaço característico), em que métodos lineares são usados para facilitar o reconhecimento. O *kernel trick* consiste na transformação (mapeamento) de dados não-separáveis linearmente no espaço de entrada em linearmente separáveis no espaço característico. Como não há restrições no mapeamento

feito pelo *kernel*, o número de dimensões poderia aumentar infinitamente (maldição da dimensionalidade), o que tornaria inviável o cálculo do mapa Φ. Não obstante, o uso do *kernel* dispensa o cálculo explícito de Φ e, por conseguinte, contorna a maldição da dimensionalidade (Steinwart e Christmann, 2008). É importante ressaltar que a escolha do *kernel* é fundamental para o sucesso de qualquer algoritimo baseado em *kernels*. O uso do *kernel* numa tarefa de aprendizagem é dado pelos seguintes passos (Shalev-shwartz e Ben-david , 2014):

1. Dado um conjunto $\mathcal{X}$, escolha um mapa $\Phi : \mathcal{X} \to \mathcal{F}$;
2. Dado uma sequência S de dados de treinamento, crie a imagem da sequência $\hat{\text{S}}$=$(\phi(x_1), y_1), \cdots, (\phi(x_n), y_n)$;
3. Treine uma regra de decisão linear em $\hat{\text{S}}$;e
4. Faça a previsão do rótulo de um ponto, x , ser $h(\phi(x))$.

O sucesso na resolução desse problema consiste na escolha do Φ que seja capaz de tornar a imagem da distribuição dos dados linearmente separável no espaço característico. No entanto, como o cálculo de separadores lineares num espaço de grande dimensão é computacionalmente complexo, usa-se o *kernel* para simplificar esse cálculo.

Definição 4.2.1. Seja uma função bivariada denominada *kernel* k: $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (ou $\mathbb{C}$, dependendo do contexto). Então, para qualquer x e x' num espaço de entrada $\mathcal{X} \subseteq \mathbb{R}^d$, pode-se expressar uma determinada função $k(x, x')$ como um produto interno num espaço característico $\mathcal{H}$:

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}} \tag{4.23}$$

em que $\langle \cdot, \cdot \rangle$ é o produto interno e Φ realiza o mapeamento (linear ou não-linear) o domínio do espaço de entrada $\mathcal{X}$ para um espaço de produto interno $\mathcal{H}$ (denominado espaço característico).

O *kernel* define uma medida de similaridade entre dois dados de entrada através do cálculo do produto interno num espaço característico. Sua principal vantagem é que, antes da aplicação do algoritmo de aprendizado, escolhe-se um *kernel* k em vez de um mapa Φ. Assim, dado um k, pode-se construir um espaço característico de forma que o *kernel* compute o produto interno nesse espaço. É comum o uso da função núcleo sem o conhecimento de Φ, que é gerado de forma implícita. É importante ressaltar que qualquer algoritmo de aprendizado que possa ser escrito como um produto interno pode ser substituído por um *kernel*.

Pela simetria do produto interno, o *kernel* também deve ser simétrico:

$$k(x, x') = k(x', x) \tag{4.24}$$

Além disso deve satisfazer a desigualdade de Cauchy-Schwartz:

$$K^2(x, x') \leq K(x, x) \times k(x', x') \tag{4.25}$$

Ademais, o *kernel* deve ser positivo definido:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} k(x_i, x_j) c_i c_j \geq 0 \tag{4.26}$$

para qualquer subconjunto finito $x_1, ..., x_n$ de $\mathcal{X}$ e um subconjunto $c_1, ..., c_n$ de números reais.

É importante ressaltar que o desempenho da máquina de suporte vetorial (*Support Vector Machine* (SVM)) é extremamente dependente da escolha do *kernel*. No entanto, não há nenhum método para escolha do melhor *kernel* para determinada tarefa (Sangeetha e Kalpana, 2010).

Segundo Genton (2001) funções simétricas e positivas também são denominadas covariâncias na literatura estatística. Uma função simétrica positiva definida é equivalente a uma matriz de Gram simétrica positiva definida:

Definição 4.2.2. (Matriz Kernel) Dado um *kernel* k e as entradas $x_1, \cdots, x_n \in \chi$, então uma matriz $n \times n$:

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{bmatrix}$$

Ou seja,

$$k_{ij} = (k(x_i, x_j))_{ij}$$

é chamada matriz de Gram de k (ou matriz *kernel*) com respeito a $x_1, \cdots, x_n$.

O Teorema de Mercer (1909) diz que toda função *kernel* contínua, simétrica e positiva definida pode ser expressa como um produto interno num espaço de grande dimensão. Assim, o teorema indica se o *kernel* escolhido pelo usuário representa de fato um produto interno em algum espaço e, por consequência é um *kernel* admissível.

Teorema 4.2.1. (Teorema de Mercer) Uma função simétrica $K(x, x')$ pode ser definida como um produto interno:

$$K(x, x') = \langle \phi_i(x) \phi_i(x') \rangle$$

para algum ϕ, se e somente se, $K(x, x')$ é positivo definido:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} k(x_i, x_j) c_i c_j \geq 0$$

Se os autovalores de uma matriz são maiores que zero, então ela é positivamente definida. Segundo Schölkopf e Smola (2002), alguns autores definem funções definidas positivas como positiva semi-definidas. O *kernel* Gaussiano é o *kernel* positivo definido mais conhecido, denominado também função de distribuição normal: $k(x, x') = \exp -\gamma \|x - x'\|, x, x' \in \mathbb{R}^N, \gamma > 0$. Essa função foi introduzida pelo matemático alemão Carl Friedrich Gauss em 1809 (Fasshauer, 2011). A aplicação do kernel gaussiano a um determinado conjunto de dados gera uma matriz kernel. Quando x é igual a x', o valor do *kernel* é igual a 1. Assim, a diagonal da matriz kernel é igual a 1. Enquanto, os valores das demais entradas estão entre 0 e 1. O coeficiente $-\gamma$ implica que as entradas fora da diagonal principal com valores grandes denotam par de observações mais similares.

4.3 Combinações de kernels

Dado alguma função *kernel*, pode-se construir outros *kernels* por meio de regras simples Bishop (2006):

Definição 4.3.1. (Contrução de kernels) Dado dois *kernels* $k_1(x, x')$ e $k_2(x, x')$ admissíveis qualquer. Então as seguintes funções núcleo também serão válidas:

1. $k(x, x') = k_1(x, x') + k_2(x, x')$;
2. $k(x, x') = c \cdot k_1(x, x')$;
3. $k(x, x') = k_1(x, x') + c$;
4. $k(x, x') = k_1(x, x') \cdot k_2(x, x')$;
5. $k(x, x') = f(x) \cdot f(x')$;
6. $k(x, x') = f(x)\, k_1(x, x')\, f(x')$;
7. $k(x, x') = \exp(k_1(x, x'))$; e
8. $k(x, x') = q(k_1(x, x'))$.

em que q é um polinômio com coeficientes não negativo.

Toda função núcleo tem suas vantagens e desvantagens (Smits e Jordaan, 2002). Para melhorar a capacidade de aprendizado e generalização de dados de determinado modelo, utiliza-se misturas de *kernels* que combinam as melhores características de dois ou mais *kernels*. Em geral, a mistura (combinação) pode ser feita de forma linear ou não-linear, mas é importante que o *kernel* resultante seja uma função núcleo admissível. O uso da combinação linear satisfaz essa condição (Smola e Schölkopf, 2004):

$$K_{mix}(x, x') = \rho K_A(x, x') + (1 - \rho) K_B(x, x') \tag{4.27}$$

em que ρ é a mistura ótima que deve ser determinada.

No contexto de combinação linear, Lu *et al.* (2009b) mostrou que a combinação linear de *kernel* gaussiano e polinomial com o uso de um modelo híbrido chamado fuzzy-SVM (FSVM) para classificação apresentou resultados superiores ao *kernel* polinomial e com base radial. Além disso, com o uso do SVR com a combinação linear do *kernel* de ondaleta com diferentes funções núcleo tradicionais, George e Rajeev (2008) apresentou resultados superiores a funções tradicionais sem combinação. Huang *et al.* (2014) realizou uma combinação linear de *kernel* de ondaletas com *kernel* linear para SVM de classificação com intuito de previsão de *financial distress* em empresas chinesas e mostrou que o modelo híbrido proposto apresentou resultados empíricos superiores aos *kernels* polinomial, signóide, ondaleta de Morlet, entre outros.

No contexto da combinação não-linear, Li e Sun (2010) propuseram um SVM basedo na combinação não-linear de vários *kernels* e mostraram suas vantagens empíricas. Ademais, Cortes *et al.* (2009) verificaram que há uma expressiva melhora no desempenho do SVM para regressão com uso de combinações não-lineares polinomiais de funções núcleo base.

4.4 Kernel de ondaleta de Morlet e Chapéu Mexicano

Ondaletas (*wavelets*) são funções que satisfazem determinadas exigências e são utilizadas na análise de séries temporais, processamento de imagens e sinais (Nason, 2008). Uma de suas vantagens é capturar tanto o domínio de frequência quanto o domínio temporal de uma série de dados (Daubechies, 1992). Segundo Zhang *et al.* (2004), as ondaletas aproximam

uma função por meio de uma família de funções oriundas de dilatações e translações de uma ondaleta mãe $\Psi(x) \in L^2(R)$, dada por:

$$\Psi_{k,a}(x) = \frac{1}{\sqrt{a}}\Psi\Big(\frac{x-k}{a}\Big), \quad x, k, a \in R \tag{4.28}$$

em que a é o fator de dilatação e k o de translação.

Com o uso da transformada de ondaleta inversa e da função ondaleta multidimensional, Zhang *et al.* (2004) propuseram o *kernel* de ondaleta com produto escalar:

$$k(x, x') = \prod_{i=1}^{N} \Psi\Big(\frac{x_i - k_i}{a}\Big)\Psi\Big(\frac{x_i' - k_i'}{a}\Big) \tag{4.29}$$

em que $a, x, x' \in \mathbb{R}^N$. Além disso, construíram um kernel de ondaleta com transformação invariante :

$$k(x, x') = \prod_{i=1}^{N} \Psi\Big(\frac{x_i - k_i}{a}\Big) \tag{4.30}$$

em que em que $a, x, x' \in \mathbb{R}^N$ e Ψ é uma função de ondaleta mãe. Combinando a Equação 4.29 com a ondaleta mãe de Morlet $\Psi(x) = cos(1.75x)\exp(x^2/2)$ (Goupillaud *et al.*, 1984), Zhang *et al.* (2004) construíram um *kernel* de ondaleta de transformação invariante com base na ondaleta de Morlet, que satisfaz a condição de (Mercer, 1909), dado pela seguinte expressão Ding *et al.* (2014):

$$k(x, x') = \prod_{i=1}^{N} \left(cos\left(1.75 \times \frac{(x_i - x_i')}{a}\right) \exp\left(\frac{-\| x_i - x_i'\|^2}{2a^2}\right)\right) \tag{4.31}$$

em que em que $a, x, x' \in \mathbb{R}^N$. Segundo Zhang *et al.* (2004), esse *kernel* pode aproximar qualquer função não linear arbitrária, pois é um tipo de função de ondaleta multidimensional. Além disso, é um *kernel* ortonormal. Zhang *et al.* (2004) mostraram por meio de simulações que o SVM para regressão e classificação que o *kernel* com ondaleta de Morlet apresenta resultados superiores ao *kernel* Gaussiano. No contexto de previsão de volatilidade, Com o kernel de ondaleta de Morlet desenvolvido por Zhang *et al.* (2004), Li (2014) mostrou que o SVR com ondaleta obtém melhor desempenho preditivo em relação ao kernel Gaussiano, pois tem menor erro de previsão, apresenta menor custo computacional e melhor capacidade de generalização. Além disso, Tang *et al.* (2009b) mostraram que o kernel com ondaleta de Debauchies pode capturar os agrupamentos de volatilidade e melhorar a capacidade preditiva do SVR em relação ao kernel Gaussiano. Com um kernel de ondaleta *spline* Tang *et al.* (2009a) mostraram a superioridade das ondaletas na previsão da volatilidade em relação ao kernel Gaussiano. O kernel de ondaleta apresenta superioridade preditiva em relação ao Gaussiano, pois este é correlativo e redundante, enquanto o primeiro não o é (Zhang *et al.*, 2004).

Com base na mesma função de ondaleta de transformação invariante dada por 4.29, é possível construir um kernel com a ondaleta mãe de Chapéu Mexicano $\Psi(x) = (1 - x^2)\exp(-1/2x^2)$, que satisfaz a condição de (Mercer, 1909), dado pela seguinte forma Ding *et al.* (2014):

$$k(x, x') = \prod_{i=1}^{N} \left(1 - \left(\frac{x_i - x_i'^2}{a_i}\right)\right) \exp\left(-\frac{1}{2}\left(\frac{x_i - x_i'^2}{a_i}\right)\right) \tag{4.32}$$

em que em que $x, x' \in \mathbb{R}^N$ e a é o parâmetro a ser determinado no período de treinamento. Neste trabalho utiliza-se o procedimento de busca em grelha (*grid-seach*) para encontrar o valor ótimo de a. Segundo Ding *et al.* (2014), quanto maior o valor de a, maior será a capacidade de generalização. Quanto menor, melhor será a capacidade de aprendizado. É importante ressaltar que esta dissertação é o primeiro trabalho a usar a ondaleta mãe de Chapéu Mexicano[7] para a previsão da volatilidade via SVR.

[7] Conhecida também por ondaleta de Ricker e ondaleta de Marr.

Capítulo 5

Máquina de suporte vetorial

"*Nothing is more practical than a good theory*"

Vapnik (1998)

Neste capítulo, apresenta-se o *Support Vector Machine* (SVM) para regressão com função de perda ϵ-insensível (ϵ-SVR). Em seguida, realiza-se a revisão da literatura do uso do SVR na estimação e previsão da volatilidade condicional.

5.1 Introdução

O *Support Vector Machine* (SVM) é uma técnica de aprendizado de máquina supervisionado baseada na Teoria do Aprendizado Estatístico desenvolvida por Vapnik (1982). Ao combinar a função *kernel* com hiperplanos de margem larga, Boser, Guyon, e Vapnik (1992) desenvolveram a forma atual do SVM. As principais características do SVM são: habilidade para lidar com dados em alta dimensão, grande acurácia na classificação e previsão, flexibilidade para trabalhar com vários tipos de dados e resultados teóricos e empíricos superiores aos modelos estatísticos e econométricos tradicionais (Cavalcante *et al.*, 2016; Sankar *et al.*, 2009).

As técnicas de *machine learning* são baseadas no princípio da indução e podem ser divididas em duas classes: aprendizado supervisionado e não-supervisionado (Vapnik, 1995). O SVM é da classe de aprendizado supervisionado, em que, dado uma amostra de treinamento de um conjunto de dados rotulado oriundo de distribuição de probabilidade desconhecida, a máquina infere uma função (também denominada classificador ou hipótese) que é utilizada para predizer o rótulo de outros dados oriundos da mesma distribuição. Os rótulos identificam o fenômeno de interesse. Se os rótulos assumirem valores discretos, então tem-se um SVM para classficação. Caso assumam valores contínuos, tem-se um SVM para regressão (Hastie *et al.*, 2009).

O *Support Vector Regression* (SVR) é uma extensão do SVM para classificação. Por isso possuem propriedades em comum. Nos últimos anos, o SVR está sendo utilizado para previsão em diversas áreas como: biologia, química, engenharia civil, meteorologia, medicina, contabilidade (Song *et al.*, 2014), economia e finanças (Varian, 2014; Zimmermann, 2015).

5.2 Classificador linear

Seja um espaço $\mathcal{X} \in \mathbb{R}^n$, um espaço de saída $\mathcal{Y} = \{-1, +1\}$ e uma função $f : \mathcal{X} \to \mathcal{Y}$. Dado uma classe de funções $\mathcal{F}$, o problema de classificação binária pode ser descrito da seguinte forma (Mohri *et al.*, 2012). A máquina recebe um conjunto de treinamento T, então sua tarefa é encontrar um classificador f com o menor erro de generalização. Como existem várias classes de funções que podem ser escolhidas, é preferível escolher aquela que possua menor complexidade (menor dimensão VC) (Mohri *et al.*, 2012). Uma escolha natural é a classe de classificadores lineares:

$$F = x \mapsto sign(w \cdot x + b) : x, w \in \mathbb{R}^n, b \in \mathbb{R} \tag{5.1}$$

em que w é o vetor de peso e b o termo de viés. Num espaço de duas dimensões o classificador linear é uma reta. Num espaço de três dimensões é um plano e num espaço de dimensão n é um hiperplano. O vetor de pesos w tem sentido perpendicular ao hiperplano, enquanto o termo de viés b move o hiperplano para longe da origem. O hiperplano $w \cdot x + b = 0$ divide o espaço em dois, em que de um lado estão os pontos positivos e do outro os pontos negativos.

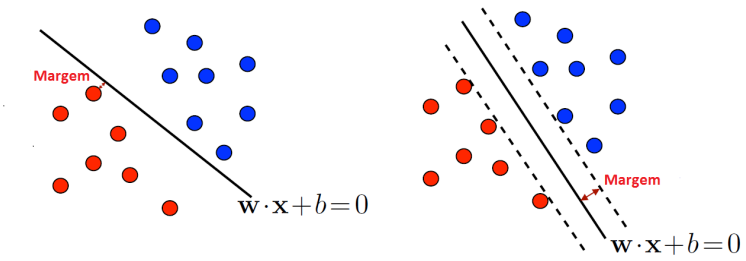

Figura 5.1: *Classificador Linear. Fonte: Adaptado de Mohri et al. (2012).*

A margem é dada pela menor distância entre o hiperplano de separação e os dados de treinamento mais próximos, denominados vetores de suporte que determinam os padrões relevantes e que sozinhos determinam o hiperplano com máxima margem.

5.3 SVM para classificação binária

Suponha que o conjunto T seja linearmente separável. Como existem várias formas de separar os dados, é preciso encontrar alguma forma de encontrar o classificador ótimo. A solução do SVM para esse problema é dada pelo classificador (hiperplano) de máxima margem, que é ótimo pois é robusto a *outliers* e tem excelente capacidade de generalização. A margem ρ é a largura do hiperplano $w \cdot x + b = 0$, que pode ser aumentada antes de atingir um ponto positivo ou negativo no caso da classificação binária:

$$\rho = \frac{|\mathbf{w} \cdot \mathbf{x} + b|}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}^{1} \tag{5.2}$$

[1]Note que maximizar a margem ρ do hiperplano é equivalente a minimizar $\|\mathbf{w}\|$ ou $\frac{1}{2}\|\mathbf{w}\|^2$.

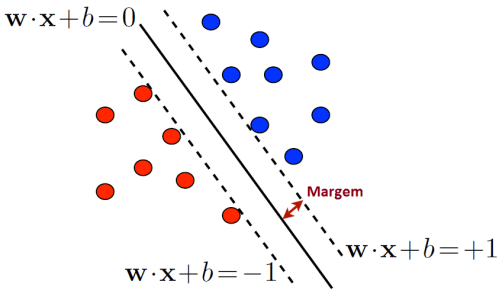

Figura 5.2: *Margem do Hiperplano. Fonte: Adaptado de Mohri et al. (2012).*

Assim, o problema de programação quadrática do SVM linear na forma primal é dado por:

$$Minimize : \frac{1}{2}\|\mathbf{w}\|^2, \tag{5.3}$$

$$\text{sujeito a} \quad y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 \quad i = 1, \cdots, m \tag{5.4}$$

Devido ao teorema de Kuhn e Tucker (1951), como a função objetivo e as restrições são convexas é possível usar os multiplicadores de Lagrange ($\alpha_i \geq 0, i = 1, \cdots, n$) para colocar o problema na sua forma dual. A função lagrangeana associada a forma primal é:

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1) \tag{5.5}$$

em que $\|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w}$.

Para encontrar o mínimo, é preciso minimizar $\mathcal{L}(\mathbf{w}, b, \alpha)$:

$$\frac{\partial \mathcal{L}}{\partial b} = \sum_{i=1}^{m} y_i\alpha_i = 0 \tag{5.6}$$

$$\frac{\partial \mathcal{L}}{\partial w} = w - \sum_{i=1}^{n} y_i\alpha_i = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{n} y_i\alpha_i\mathbf{x_i} \tag{5.7}$$

Assim, chega-se o problema na forma dual é dado por:

$$Maximize : \sum_{i-1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i\alpha_j y_i y_j x_i x_j, \tag{5.8}$$

$$\text{sujeito a} \quad \alpha_i \geq 0 \quad \text{e} \quad \sum_{i=1}^{n} y_i\alpha_i = 0 \tag{5.9}$$

Isso é um problema de programação quadrática que pode ser solucionado com vários métodos, como por exemplo o algoritmo de *Sequencial Minimal Optimization* (SMO). Pela condição de Karush-Kuhn-Tucker (KKT) (Karush (1939); Kuhn e Tucker (1951)), sabe-se que $\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1) = 0$. Então, os pontos $\mathbf{x}_i$ com α_i diferentes de zero são denominados vetores de suporte (SV), pois são os pontos mais próximos do hiperplano separador ótimo e são os únicos pontos de S necessários para determinar esse hiperplano. Por isso são conhecidos como vetores de suporte. Como a solução apresenta vários α_i que são zero, w é uma combinação

linear de uma pequena fração de pontos x_i:

$$\mathbf{w} = \sum_{i \in SV} \alpha_i y_i \mathbf{x}_i \tag{5.10}$$

Caso T não seja linearmente separável, variáveis de folga são introduzidas na restrição 5.4 para permitir que o algoritmo melhore sua capacidade de generalização:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \tag{5.11}$$

Quando o ponto está dentro da margem de erro ($0 \leq \xi_i \leq 1$), ele viola a margem do classificador, mas está no lado correto. Quando $\xi_i \geq 1$, o ponto está mal classificado. Assim, deseja-se encontrar w e b que minimize:

$$Minimize : \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i, \tag{5.12}$$

$$\text{sujeito a} \quad y_i(\mathbf{w}^T \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \tag{5.13}$$

O parâmetro C especifica um *trade-off* entre o erro e a margem. Quanto maior C, menor o número de pontos classificados de forma errada. Quanto menor C, há maximização da margem. Caso $C = \infty$, tem-se o caso de margem rígida.

A forma dual é dada por:

$$Maximize : \sum_{i-1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j, \tag{5.14}$$

$$\text{sujeito a} \quad 0 \leq \alpha_i \leq C \quad \text{e} \quad \sum_{i=1}^{m} y_i \alpha_i = 0 \tag{5.15}$$

Até agora foi considerado um classificador de máxima margem com limite de decisão linear. Porém, é desejável que se possa produzir um limite de decisão não-linear. Para tanto, é preciso transformar o vetor de entrada x para um espaço de maior dimensão (Vapnik , 1995). Note que na forma dual do problema de otimização do SVM, os dados aparecem na forma de produto interno $x_i^T x_j$. Para transformar cada ponto para um espaço do espaço de entrada R^c para alguma espaço de maior dimensão R^n (n > c), utiliza-se um mapa $\Phi : R^c \rightarrow R^n$. Portanto, o produto interno é calculado no espaço característico de grande dimensão $\phi(x_i).\phi(x_j)$. Porém, o cálculo desse produto é computacionalmente custoso. Assim, utiliza-se a função *kernel*: $k(x_i, x_j) = \phi(x_i)^T.\phi(x_j)$[2], evitando o cálculo explícito do mapa $\phi(.)$. Dessa maneira, basta substituir o produto interno por uma função *kernel* nas derivações anteriores do SVM linear. Assim, o operador linear no espaço de maior dimensão é equivalente a um operador não-linear no espaço de entrada (Vapnik, 1995).

Um classificador num espaço de grande dimensão tem muitos parâmetros e é de difícil estimação. Segundo Vapnik (1995), o problema não é o número de parâmetros, mas a flexibilidade do classificador, que pode ser medida pela sua complexidade com a dimensão VC. Quanto maior essa dimensão, mais flexível é o classificador. Não obstante, o cálculo da dimensão VC, em geral, não é factível. Por isso, o SVM é baseado no princípio da Minimização Estrutural do Risco, pois $\sum_i^m \xi_i$ aproxima o erro empírico, enquanto $\frac{1}{2}\|\mathbf{w}\|^2$ está relacionado a complexidade de função classificadora.

[2] $k(x_i, x_j)$ é apenas uma medida de similaridade que compara x_i e x_j. Também é denominado função de covariância (Wilson *et al.*, 2015).

5.4 SVM para regressão não-linear

Inicialmente, o SVM foi utilizado para a classificação de dados. Com a introdução da função de perda ϵ-insensível[3] por Vapnik (1995), o SVM de classificação foi estendido para ser usado em regressões lineares e não-lineares devido a sua acurácia e vantagens computacionais (Smola e Schölkopf, 2004).

Dado um conjunto de treinamento $T = (x_1, y_1), \cdots, (x_n, y_n) \subset \mathbb{R}^N \times \mathbb{R}$, em que $x_n \in \mathcal{X}$ é o vetor de entrada e $y_n \in \mathbb{R}$, o escalar de saída, o objetivo do SVR é encontrar uma função $f(x)$ que aproxima o escalar y_n a menos de um erro de previsão ϵ especificado (Vapnik , 1995). Para tanto, o SVM mapeia de forma não-linear o espaço original para um espaço característico de dimensão mais elevada. Assim, as relações não-lineares do espaço original são aproximadas por uma regressão linear no espaço característico de dimensão mais elevada da seguinte forma (Vapnik, 1995) :

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b, \quad \text{com} \quad \phi : \mathbb{R}^N \to \mathcal{F}, w \in \mathcal{F} \tag{5.16}$$

em que $\mathbf{w}$ é o vetor de pesos, b o termo de viés e $\phi(\mathbf{x})$ é a função mapa não-linear, que projeta os vetores de entrada $\mathbf{x}$ no espaço característico de dimensão elevada $\mathcal{F}$, onde a regressão linear está definida. Quanto maior a dimensão, maior é acurácia do SVR na aproximação suave da função mapa.

Para estimar a regressão é necessário mensurar a diferença entre o valores reais e as respectivas previsões por meio da função de perda ϵ-insensível linear, L_ϵ, proposta por Vapnik (1995). O $\epsilon - SVR$ busca estimar a função $f(\mathbf{x})$ de modo que ela seja o mais suave possível e com erros menores que ϵ no espaço característico. Assim, a norma Euclideana do vetor de pesos $\|\mathbf{w}\|^2$ deve ser minimizada ao mesmo tempo em que se controla o erro sob as restrições de L_ϵ. Então, tem-se o seguinte problema de otimização convexa (Vapnik, 1995) :

$$Minimize : \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n}(L_\epsilon(f(\mathbf{x}), y)); \tag{5.17}$$

em que:

$$L_\epsilon(f(x), y) = \begin{cases} |y_i - f(\mathbf{x})| - \epsilon, & \text{se} \quad |y_i - f(\mathbf{x})| > \epsilon, \\ 0, & \text{caso contrário} \end{cases} \tag{5.18}$$

é a função de perda ϵ-insensível.

Assim, à medida que o valor da função se afasta do erro permitido ϵ, há atribuição de uma penalização linear para o modelo. Apenas as observações que estão em cima e fora da zona (ou banda) de erro insensível, conhecidas como vetores de suporte, irão prover informações para a função de decisão($f(\mathbf{x})$). É importante destacar que a variação de ϵ influencia o número de suportes vetoriais e, por conseguinte, controla a complexidade do modelo (Cherkassky e Ma, 2004). Assim como no SVM com margem suave para classificação, variáveis de folgas (ξ_i, ξ_i^*) são introduzidas para identificar os erros que estão fora da zona (Smola e Schölkopf, 2004). Tem-se assim o problema primal de programação quadrática do SVR (Vapnik, 1995) :

$$Minimize : \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*), \tag{5.19}$$

[3] É importante ressaltar que existem várias funções de perda e que cada uma irá resultar em desempenho distinto das regressões (Smola e Schölkopf, 2004). Neste trabalho utiliza-se apenas a função de perda ϵ-insensível.

$$\text{sujeito a } \begin{cases} y - \mathbf{w^T}\phi(\mathbf{x}) - b \leq \epsilon + \xi_i, \\ \mathbf{w^T}\phi(\mathbf{x}) + b - y \leq \epsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$

em que o primeiro termo (termo de regularização) mede o achatamento da função e indica a norma euclideana do vetor de pesos w. O segundo termo representa a perda de risco empírico determinada pela função de perda ϵ-insensível (Cherkassky e Ma, 2004). Quando o erro é menor que ϵ, as variáveis de folga (ξ_i, ξ_i^*) têm valor zero. O parâmetro de penalização C determina qual extensão do erro empírico ($\epsilon = y - w^T\phi(x) - b$) será tolerado. Quanto maior o valor de C, menor será a margem , menos erros de previsão na amostra de treinamento serão permitidos. Por conseguinte, o algoritmo irá superajustar os dados e terá menor capacidade de generalização(Cherkassky e Ma, 2004). O parâmetro C pode ser visto como uma forma de controlar o superajustamento. Dessa maneira, o SVR especifica o *trade-off* entre os dois termos de forma que a regressão seja capaz de modelar tanto os dados históricos como fazer previsões acuradas de valores futuros desconhecidos. Os parâmetros C e ϵ são os parâmetros livres do SVM e, em geral, são determinados, concomitantemente, pelo método da validação-cruzada (Haykin, 1999). Além disso, a programação quadrática convexa e as restrições lineares do problema primal acima garantem que o SVR sempre obterá a solução única global ótima.

De acordo com a teoria de otimização, sabe-se que a solução do problema da equação 5.19 é complicada devido a um grande conjunto de variáveis. Desse modo, transforma-se o problema para a forma dual com a introdução de um conjunto de variáveis dual e o uso de Multiplicadores de Lagrange (Vapnik, 1995) :

$$\begin{aligned} \mathcal{L} = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) - \sum_{i=1}^{n}\alpha_i(\epsilon + \xi_i - y_i + \langle\, \mathbf{w}, \phi(\mathbf{x})\rangle + b) \\ - \sum_{i=1}^{n}\alpha_i^*(\epsilon + \xi_i + y_i - \langle\, \mathbf{w}, \phi(\mathbf{x_i})\rangle - b) - \sum_{i=1}^{n}(\eta_i\xi_i^* + \eta_i^*\xi_i^*) \\ \text{sujeito a} \quad \alpha_i, \alpha_i^*, \eta_i, \eta_i^* > 0 \end{aligned} \tag{5.20}$$

em que $\mathcal{L}$ é a função lagrangeana e $\alpha_i, \alpha_i^*, \eta_i, \eta_i^*$ os multiplicadores de Lagrange. Segundo Mangasarian (1994), a função acima tem um ponto de sela em relação as variáveis primal e dual. Assim, derivando $\mathcal{L}$ em relação às variáveis de decisão w, b, ξ_i, ξ_i^*, é possível satisfazer a condição do ponto de sela:

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial b} &= \sum_{i=1}^{L}(\alpha_i^* - \alpha_i) = 0 \\ \frac{\partial\mathcal{L}}{\partial w} &= w - \sum_{i=1}^{L}(\alpha_i^* - \alpha_i)\phi(\mathbf{x_i}) = 0 \\ \frac{\partial\mathcal{L}}{\partial\xi_i} &= C - \alpha_i - \eta_i = 0 \\ \frac{\partial\mathcal{L}}{\partial_{\xi_i^*}} &= C - \alpha_i^* - \eta_i^* = 0 \end{aligned} \tag{5.21}$$

Substituindo w e as variáveis duais (η_i, η_i^*) na Equação 5.20, tem-se o seguinte problema

de programação matemática na forma dual:

$$\text{Minimize} : \frac{1}{2}\sum_{j=1}^{n}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle\ \phi(\mathbf{x_i}), \phi(\mathbf{x_j})\rangle - \sum_{i=1}^{n} y_i(\alpha_i - \alpha_i^*) + \epsilon\sum_{i=1}^{n}(\alpha_i + \alpha_i^*) \tag{5.22}$$

$$\text{sujeito a} \sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0 \quad \text{e} \quad \alpha_i, \alpha_i^* \in [0, C]$$

Como o problema de otimização não-linear da Equação 5.20 tem restrições de desigualdade, as condições de Karush-Kuhn-Tucker (Karush (1939);Kuhn e Tucker (1951)) devem ser satisfeitas. Segundo Smola e Schölkopf (2004), essas condições estabelecem que, no ponto de solução, o produto entre as variáveis duais e as restrições devem ser removidas.

$$\begin{gathered}\alpha_i(\epsilon + \xi_i - y_i + \langle \mathbf{w}, \mathbf{x_i}\rangle + b) = 0 \\ \alpha_i^*(\epsilon + \xi_i - y_i + \langle \mathbf{w}, \mathbf{x_i}\rangle - b) = 0 \\ (C - \alpha_i)\xi_i = 0 \\ (C - \alpha_i^*)\xi_i^* = 0\end{gathered} \tag{5.23}$$

As condições acima implicam que, se $|y_i - f(\mathbf{x})| < \epsilon$, então $\alpha_i, \alpha_i^* = 0$. Assim, apenas as observações x_i tais que $\alpha_i, \alpha_i^* \neq 0$ são chamados de vetores de suporte (*support vectors*) e são usados para derivar a função de decisão (Smola e Schölkopf, 2004). De 5.21 tem-se $w = \sum_{i=1}^{n}(\alpha_i^* - \alpha_i)\phi(\mathbf{x_i})$ que é o vetor de suporte em expansão, então a regressão do SVM é dada por (Vapnik, 1995) :

$$f(\mathbf{x}) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\langle\phi(\mathbf{x_i}), \phi(\mathbf{x})\rangle + b \tag{5.24}$$

em que $\langle\phi(\mathbf{x_i}), \phi(\mathbf{x})\rangle$ é o produto interno dos vetores no espaço característico. Conforme dito no 4, devido a complexidade de calcular explicitamente o mapa não-linear, é possível substituí-lo por um *kernel* admissível (Vapnik, 1995) :

$$f(\mathbf{x}) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)K(\mathbf{x_i}, \mathbf{x}) + b^* \tag{5.25}$$

A função *kernel* é de extrema importância para o SVR pois, ao dispensar o cálculo explícito do mapa não-linear, reduz substancialmente o custo computacional do SVR.

5.5 SVR na previsão de séries temporais financeiras

É possível destacar três características de uma série temporal financeira que dificultam sua previsibilidade (Cao e Tay, 2001). Primeira, presença de ruído, caracterizada pela indisponibilidade de informações completas sobre o comportamento passado do mercado para capturar a dependência entre o preço passado e futuro. Segunda, a não estacionaridade, o que implica que a distribuição conjunta da série se altera ao longo do tempo. E, por fim, a presença de caos determinístico, em que no curto prazo a série é aleatória, mas no longo prazo exibe um padrão determinístico (Cao e Tay, 2001). A modelagem de séries financeiras com o SVM visa a superação dessas dificuldades com intuito de aperfeiçoar as previsões.

É possível dividir as aplicações do SVR em finanças empíricas em três partes: formação de carteiras (Huerta *et al.*, 2013), previsão de retorno de ativos e gerenciamento de riscos.

Conforme dito na introdução deste capítulo, o SVM é baseado no Princípio da Minimização Estrutural do Risco (SRM) da Teoria de Aprendizado Estatístico. Esse princípio foi construído sob a hipótese de que os dados do conjunto de treinamento S são independentes e identicamente distribuídos (Ruping e Morik, 2003). No caso de séries temporais, essa hipótese é violada. Não obstante, Fender (2003) demonstra que a maioria dos teoremas centrais envolvidos na minimização do risco estrutural continuam válidos para dados que possuam uma estrutura de dependência fraca. Em que pese essas restrições, evidências empíricas mostram o êxito do SVR em comparação aos modelos tradicionais de previsão de séries temporais(Ferreira, 2011; Sankar *et al.*, 2009).

Ademais, para a análise de séries temporais univariadas e multivariadas, a escolha do *kernel* ótimo é crucial para a qualidade da modelagem dessas séries. Cada *kernel* modela diferentes hipóteses no processo gerador da série temporal. Não obstante, até o momento não há nenhum método para escolha da função núcleo mais adequada para diferentes séries temporais, inclusive séries financeiras.

5.6 Aplicações do SVR na estimação e previsão de volatilidade condicional

O objetivo desta seção é revisar a literatura sobre a utilização do *Support Vector Regression* (SVR) na estimação e previsão da volatilidade condicional. Os artigos que compõe essa revisão foram encontrados por meio de uma pesquisa conduzida até março de 2016 nas bases de dados da Elsevier, Wiley Online Library, IEEE Xplore Digital Library, Google Scholar, SCOPUS, ISI Web of Knowledge, Sciencedirect e ProQuest com as seguintes palavras-chaves: *volatility forecasting, support vector regression e support vector machine.*

5.6.1 Revisão da literatura

Em geral, o GARCH é estimado pelo método da máxima-verossimilhança (ML), que é ótimo quando os resíduos seguem uma distribuição normal. Caso isso não ocorra, haverá mais erro na estimação. Assim, Fernando *et al.* (2003) estimaram um modelo GARCH por meio de ϵ-*Support Vector Regression* (ϵ-SVR), pois essa ferramenta não pressupõe nenhum tipo de distribuição sobre a série de retornos. Para comparar a estimação dos parâmetros GARCH (1,1) via SVR em relação ao ML, Fernando *et al.* (2003) realizaram uma modelagem empírica em seis séries financeiras com observações diárias nos anos de 1990: 4 índices de ações e 2 ações. Para calcular a proxy, usaram o retorno quadrático e retorno intra-diário como medidas da volatilidade realizada *ex-post*, a estimação do GARCH(1,1) por meio do SVR no período *out-sample*, apresentou resultados preditivos superiores ao GARCH(1,1) estimado via ML.

Gavrishchaka e Ganguli (2003) apresentam as vantagens de usar o SVM na previsão da volatilidade para capturar a memória longa e os efeitos multi-escala. Para modelar a volatilidade condicional, os autores utilizam retornos defasados como *input* do SVR e a volatilidade realizada como *proxy* da volatilidade Para dados de taxa de câmbio (dolár/marco alemão) de 1980 a 2000, os autores mostram que o SVM apresenta resultados preditivos superiores aos modelos tradicionais de volatilidade como o GARCH. Mais tarde, Gavrishchaka e Banerjee (2006) aplicaram o modelo proposto por Gavrishchaka e Ganguli (2003) para a série de retornos do $S\&P500$ e também mostraram que o SVM é superior aos modelos de volatilidade

tradicionais da família ARCH.

Fernando *et al.* (2003) usaram um SVM para o GARCH com uma estrutura (*feed-forward*), que possui pouca habilidade para modelar a memória longa e possui uma forma autoregressiva AR(1). Conforme Haykin (1999), boa parte dos modelos de predição com base no SVR eram feitos de forma estática, numa única direção (*feedforward*) e capturavam apenas uma dinâmica AR não-linear. No entanto, é possível melhorar as previsões com *recurrent-loop* SVR e chega-se assim numa estrutura semelhante ao ARMA não-linear. Em face disso, Chen *et al.* (2010) propuseram uma forma recursiva de realizar previsões da volatilidade com *Support Vector Regression*(SVR) baseado no GARCH, denominado SVM-GARCH, que introduz uma estrutura ARMA não-linear na equação da média e na variância condicional. Para avaliar a acurácia do SVM-GARCH na previsão da volatilidade de um período a frente em comparação ao modelo de média móveis, GARCH tradicional, EGARCH assimétrico e modelos de redes neurais artificiais para o GARCH (ANN-GARCH), os autores utilizaram as métricas do Erro Absoluto Médio (MAE) e Acurácia Direcional (DA). Além disso, usaram o teste de Diebold-Mariano para avaliar as previsões. Para a escolha dos parâmetros livres do SVM, os autores utilizaram o método da validação-cruzada e análise de sensibilidade. A comparação dos modelos foi feita numa série simulada e com dados reais. Os resultados empíricos para a série simulada demonstram que, para os três tipos de *kernel* testados (linear, polinomial e gaussiano) o desempenho preditivo da volatilidade é superior a todos os outros modelos. Ademais, constataram que os três *kernels* testados apresentaram resultados semelhantes.

Xu *et al.* (2011) compararam a capacidade preditiva dos modelos de médias móveis, GARCH (1,1), EGARCH (1,1), FIGARCH (1,1), redes neurais e o SVM-GARCH na previsão da volatilidade de dois índices de preço do mercado acionário chinês (*Shangai A shares* e *Shenzen A shares*) no período de janeiro de 2006 a abril de 2010. Com a utilização das métricas de Erro Absoluto Médio (MAE), Raiz do Erro Quadrático Médio (RMSE) e *Hit Rate*(HR) para avaliar o desempenho preditivo dos modelos, os autores demonstraram que o FIGARCH é capaz de capturar a propriedade de memória longa e supera o GARCH e EGARCH em termos de acurácia preditiva. Não obstante, em relação a todos os outros modelos, o SVM-GARCH apresentou a melhor capacidade preditiva.

Ou e Wang (2010b) utilizaram o LLSVM (*Least Square Support Vector Machine*) modificado por Suykens (1999) para a construção de modelos de volatilidade híbridos como GARCH-LSSVM, EGARCH-LSSVM e GJR-LSSVM para previsão da volatilidade de três índices de ações (Cingapura, Filipinas e Kuala Lampur) que compõe a Associação de Nações do Sudeste Asiático (ASEAN). As previsões dos modelos híbridos foram comparados com o GARCH(1,1), EGARCH(1,1) e GJR(1,1), respectivamente. As previsões foram feitas em dois estágios: no ano de 2007 e no ano de 2008. Com a utilização das métricas de Erro Absoluto Médio (MAD), Erro Quadrático Médio Normalizado (NMSE) e *Hit Rate*(HR) e R^2 para mensurar a performance dos modelos, os modelos híbridos apresentaram mais robustez e resistência a períodos de alta volatilidade em relação aos respectivos modelos tradicionais.

Com base no SVM recursivo proposto por Chen *et al.* (2010), Ou e Wang (2010a) propuseram um modelo *Recurrent Relevant Support Vector Machine* (RRVM) para previsão da volatilidade do *Shangai Composite Index* (SSECI). O modelo *Relevant Vector Machine* foi proposto por Tipping (2001). Além de ter um tratamento bayesiano, esse modelo tem a forma funcional idêntica ao SVM, o que permite o aproveitamento das vantagens do SVM, mas por outro lado não necessita da obtenção do valor ótimo dos parâmetros C e ϵ. O objetivo do *paper* foi comparar o modelo RVM recursivo criado por Ou e Wang (2010a) com os modelos SVM-GARCH, LSSVM recursivo e GARCH (1,1) na previsão da volatilidade no período de 2001 a 2006. O período de estimação *in-sample* (treinamento do algoritmo) foi

de Janeiro de 2001 a Dezembro de 2005. Enquanto, o período *out-sample* (teste do RRVM) foi de Janeiro de 2006 a dezembro de 2006. Com a utilização das métricas de Desvio Médio Absoluto (MAD), Erro Quadrático Médio Normalizado (NMSE) e *Hit Rate* para mensurar a performance das previsões, foi constatado que o RRVM apresenta desempenho superior a todos os outros modelos, pois é um modelo dinâmico e possui memória longa. Além disso, o LSSVM recursivo e o SVM-GARCH apresentam resultados bem parecidos e ambos foram superiores ao GARCH(1,1).

Sabe-se que o desempenho da previsão do SVM depende da escolha da função *kernel*. Um dos problemas da previsão da volatilidade condicional via SVM é que os *kernels* habitualmente utilizados (gaussiano, linear e polinomial) não são capazes de capturar de forma acurada os *clusters* de volatilidade. No entanto, segundo Tang, Tang, e Sheng (2009b), teoricamente, a função ondaleta pode descrever os agrupamentos de volatilidade de forma adequada. Assim, Tang, Tang, e Sheng (2009b) combinaram a teoria de ondaletas com o SVM para produzir uma função *kernel* de ondaleta multidimensional para prever a volatilidade condicional dos retornos de mercados com base na estrutura do GARCH. O desempenho preditivo do *kernel* com ondaleta de Debauchies em comparação ao núcleo gaussiano foi avaliado em dois conjuntos de dados simulados e cinco índices diários. Os autores fizeram duas simulações. Uma com o termo de erro seguindo uma distribuição normal. E outra com uma distribuição *t- Student* com 4 graus de liberdade para simular o excesso de curtose presente em séries financeiras. Com a utilização das métricas de distância de Komolgorov-Sirminov (KS) e distância de Anderson-Darling (AD) Erro Quadrático Médio Normalizado para mensurar a performance das previsões, foi constatado que as ondaletas de Debauchies apresentam desempenho superior ao *kernel* gaussiano nas simulações. Para análise em dados reais, Tang *et al.* (2009b) realizaram uma modelagem empírica em cinco índices de ações com observações diárias de 1º de janeiro de 1992 a 31 de dezembro de 1997. Com a utilização das métricas de Erro Absoluto Médio Normalizado (NMAE), Erro Quadrático Médio Normalizado (NMSE) e HitRate (HT) para mensurar a performance das previsões, foi constatado que as ondaletas de Debauchies apresentam desempenho superior ao kernel gaussiano.

Tang *et al.* (2009a) construíram o *Spline Wavelet Kernel Support Vector Machine* (SWSVM) com uso da combinação da teoria de *spline* e ondaletas para previsão de volatilidade com base no modelo GARCH. Um das formas mais simples de construir uma ondaletas envolve a utilização de funções *splines*, que possuem poucos suportes. O desempenho preditivo do *kernel* de ondaleta *spline* em comparação ao *kernel* gaussiano foi avaliado em dois conjuntos de dados simulados e cinco índices diários. Os autores fizeram duas simulações. Uma com o termo de erro seguindo uma distribuição normal. E outra com uma distribuição *t- Student* com 4 graus de liberdade para simular o excesso de curtose presente em séries financeiras. Com a utilização das métricas de Erro Absoluto Médio Normalizado (NMAE), Erro Quadrático Médio Normalizado (NMSE) e HitRate (HT) para mensurar a performance das previsões, foi constatado que as ondaletas com *spline* apresentam desempenho superior ao kernel gaussiano nas duas simulações. Para análise em dados reais, Tang *et al.* (2009a) realizaram uma modelagem empírica em cinco índices de ações com observações diárias de 1992 a 1997. Os autores constaram que o SWSVM apresentam desempenho bem superior ao SVM com *kernel* gaussiano.

Em geral, o GARCH é estimado pelo método da máxima-verossimilhança, em que é necessário especificar uma distribuição para o termo de erro. Diante disso, Hwang e Shin (2010) propuseram a utilização do *kernel machine learning* para estimar os parâmetros do GARCH. Os resultados empíricos demonstraram que o *kernel machine learning* possui melhor desempenho na previsão da volatilidade em relação a estimação pelo método da

máxima-verossimilhança e o SVM.

Shim e Lee (2010) utilizaram o LSSVR num esquema iterativo para estimar a média e a volatilidade condicional num modelo GARCH-M não-linear. O método consistiu num LSSVR balanceado para a média e um LSSVR desbalanceado para a volatilidade condicional. Os resultados empíricos mostraram que o GARCH-M não linear teve uma melhor performance em relação ao GARCH linear e o GARCH-M linear.

Khan (2011b) combinou o SVR com o modelo HAR (Heterogenous Autoregressive) para criar um modelo híbrido chamado SVM-HAR que melhorasse a previsão da volatildade realizada para o índice Nikkei 225. Na comparação do SVM-HAR com HAR clássico foi constatado que o modelo híbrido foi superior ao clássico na previsão da volatilidade para o Nikkei 225.

Khan (2011a) comparou o modelo SVM-HAR-ARCH baseado na volatilidade diária realizada no período de 5 e 15 minutos do índice Nikkei com o modelo HAR-ARCH usando diferentes distribuições para o termo de erro no cálculo do VaR para um período a frente. Os resultados do estudo demonstraram que para os dados intradiários de 15 minutos o SVM-HAR-ARCH é superior HAR-ARCH.

A previsão de séries financeiras é desafiante, por causa das causas pesadas, volatilidade persistente e memória longa. Segundo Wang *et al.* (2011), nos últimos anos processos multifractais foram propostos para a modelagem de séries temporais, pois têm as propriedades de memória longa e caudas pesadas. Assim, os autores propuseram um SVM baseado num modelo *Markov-Switch Multifractal* (MSM) para previsão de volatilidade no curto prazo. O SVM é usado para modelar as inovações e o MSM modela a volatilidade, pois consegue capturar *outliers*, *clusters* de volatilidade e a dependência de longo prazo. Segundo Wang, Huang, e Wang (2011), os modelos MSM tem um excelente desempenho em relação ao GARCH(1,1) na previsão de volatilidade num horizonte de 10 a 15 dias. Para períodos mais longos, os resultados das previsões são pouco superiores ao GARCH(1,1). Assim, os autores desenvolveram um algoritmo para previsão de volatilidade de curto prazo com a utilização de um modelo híbrido de SVM com MSM. O modelo proposto foi avaliado na estimação da volatilidade em dois índices da bolsa chinesa. A análise foi feita para um período de 20 anos (entre 1991 e 2010). Com o uso das métricas de Erro Quadrático Médio (MSE) e coeficiente R^2 para mensurar a acurácia das previsões, foi constatado que o SVM-MSM é superior ao GARCH(1,1) e ao MSM.

Hossain e Mohammed (2011) utilizaram a combinação do SVM e RVM baseados no modelo GARCH e compararam esses modelos híbridos com o GARCH e o ARMA-GARCH na previsão de volatilidade múltiplos períodos a frente de três índices de ações do mercado acionário chinês. Com a utilização das métricas de Erro Absoluto Médio (MAE), MSE, DS, R^2 os resultados empíricos demonstraram que os modelos híbridos com SVM têm melhor desempenho.

Segundo Geng e Liang (2011), para melhorar as previsões do modelo GARCH pode-se usar o modelo GM (1,1) para modificar a sequência do termo de erro do GARCH, levando ao surgimento do modelo híbrido GM-GARCH, que é superior ao GARCH. Devido às limitações na estimação dos parâmetros do modelo GM(1,1), os autores utilizaram o SVR para estimar seus parâmetros (SVRGM). Em seguida, esse modelo é integrado ao GARCH, formando o modelo SVRGM-GARCH. Com intuito de comparar a capacidade preditiva da volatilidade dos índices de preços dos de Shangai e Shenzen. Os resultados empíricos demonstraram que o SVRGM-GARCH é superior ao GM-GARCH e o GARCH na previsão da volatilidade.

Geng (2012) comparou o SVR com kernel gaussiano com três modelos de SVR com três tipos de *kernel* de ondaleta para previsão da volatilidade do Shanghai Composite Index (SHCI), usando *range volatility* como *proxy* da volatilidade. Com a utilização das métricas

de Erro Absoluto Médio (MAE), Raiz do Erro Quadrático Médio Ajustado pela Heteroscedasticidade (HRMSE), os resultados empíricos demonstraram que o SVR com *kernel* de ondaletas é superior ao *kernel* gaussiano. E dentre os três *kernels* de ondaletas utilizados, o Morlet possui melhor capacidade preditiva.

Ou e Wang (2013) construíram um modelo SVR combinado com um algoritmo genético caótico para modelar a média e a variância condicional dos retornos financeiros. Os resultados empíricos para os dados da NASDAQ de 2001 a 2010 mostram que o modelo proposto supera o SVR com algoritmo genético, SVR com *grid-search*, EGARCH, GJR e FIGARCH.

Li-yan *et al.* (2013) combinaram o modelo GM(1,1) com o LSSVM baseado na técnica de otimização de partícula de enxame para encontrar os parâmetros ótimos, originando o modelo GLLSVM-PSO. Os autores utilizaram dados de alta-frequência do preço de fechamento no intervalo de 1 minuto da bolsa de Shangai. Com a utilização das métricas de Erro Absoluto Médio (MAE), Raiz do Erro Quadrático Médio Ajustado pela Heteroscedasticidade (HRMSE) e logarithmic error statistic (LL) e Linear- Exponential (LINEX), os autores compararam a capacidade preditiva do GLSSVM-PSO, GLSSVM baseado em validação cruzada (GLSSVM-CV), LSSVM baseado no PSO (LSSVM-PSO) e GM(1,1) na previsão de volatilidade um período a frente. Os resultados empíricos demonstraram que o modelo GLSSVM-PSO é superior aos outros modelos. Além disso, o algoritmo de otimização de partícula de enxame (Particle Swarm Optimization, PSO) foi o mais rápido na busca dos parâmetros ótimos do GLSSVM.

Por ser um modelo paramétrico o GARCH não é capaz de realizar previsões da volatilidade de maneira adequada. Assim, Geng e Yu (2013) aprimoraram o GARCH com o uso do LSSVR com uso do (Particle Swarm Optimization, PSO) para encontrar os parâmetros originando o modelo LSSVR-GARCH-SIWPSO. Com a utilização das métricas de Erro Absoluto Médio (MAE), Raiz do Erro Quadrático Médio Ajustado pela Heteroscedasticidade (HRMSE) e *logarithmic error statistic* (LL) e Linear Exponential (LINEX), os autores compararam a capacidade preditiva do LSSVR-GARCH-SIWPSO com *kernel* gaussiano, LSSVR-GARCH baseado em validação cruzada (LSSVR-GARCH-CV) com *kernel* gaussiano e o GARCH na previsão de volatilidade um período a frente de quatro índices de preço com dados de alta frequência.

Ding *et al.* (1993) desenvolveram o modelo APARCH para capturar o efeito alavancagem (que não é modelado pelo GARCH). Além disso, considera uma estrutura autoregressiva flexiva dos retornos. Sabe-se que essa flexibilidade dificulta a estimação dos parâmetros, que habitualmente é feita por meio da estimação de máxima verossimilhança (ML) ou quasi-máxima verossimilhança (QML). Pesquisas já mostraram que o QML é ineficiente quando os dados não seguem uma distribuição normal. Assim, Li (2014) comparou a habilidade de estimação e previsão do QML em relação ao SVM em séries financeiras por três motivos. Primeiro, o SVM não faz nenhuma suposição sobre a distribuição dos dados. Segundo, por utilizar o princípio da minimização estrutural do risco, o SVM pode capturar melhor as características não-lineares dos dados como efeito alavancagem, caudas pesadas e agrupamentos de volatilidade. Por fim, o SVM torna o problema de estimação do APARCH menos complexo.

Assim, para comparar a estimação do QML e SVM para a família APARCH, Li (2014) usou uma simulação de Monte Carlo para o APARCH com as inovações seguindo uma distribuição t-Student assimétrica para modelar a assimetria e curtose. Com o uso das métricas de Erro Quadrático Médio Normalizado (NMSE), Erro Absoluto Médio Normalizado (NMAE) e Acurácia Direcional (DA) para mensurar a acurácia das previsões, os resultados mostraram que o SVM supera o QME tanto na estimação quanto na previsão.

É importante destacar que a escolha do *kernel* influencia o desempenho do SVM. Sendo

assim, para a estimação e previsão de volatilidade, Tang *et al.* (2009b) sugeriu que o *kernel* de ondaleta poderia teoricamente capturar melhor o *cluster* de volatilidade do que o *kernel* gaussiano, pois o *kernel* de ondaleta é construído sobre uma base ortonormal e poderia aproximar melhor curvas no espaço contínuo integral quadrático do que o *kernel* gaussiano. Assim, Li (2014) investigou se o núcleo de ondaleta apresenta melhores resultados que o gaussiano, que é o mais utilizado. A autora verificou que, para o APARCH com distribuição t-Student assimétrica, o *kernel* de ondaleta produz previsões mais acuradas que o gaussiano, conforme previsto teoricamente por Tang *et al.* (2009b). Além disso, menos suportes são necessários quando o núcleo de ondaleta é utilizado, o que simplifica a computação e melhora a acurácia da previsão.

Seethalakshmi *et al.* (2014) criaram um modelo híbrido denominado PCASVM com base na combinação uma análise de componente principal com o SVM para previsão da volatilidade. Com a utilização das métricas de Erro Absoluto Médio Normalizado (NMAE) e Erro Quadrático Médio Normalizado (NMSE) os autores mostraram que modelo PCASVM tem boa capacidade de previsão da volatilidade do índice S&P 500.

Para melhorar as previsões do modelo CARXX (*Conditional Autoregressive Range model with Exogenous Variables*), em vez de realizar a estimação dos parâmetros pelo método da quasi-verossimilhança Liyan e Zhanfu (2012) utilizaram o LSSVR (*Least Square-SVR*) com o uso do APSO (*Adaptive Particle Swarm Optimization*) na otimização dos parâmetros do SVR. Com a utilização das métricas de Erro Absoluto Médio (MAE), Raiz do Erro Quadrático Médio (RMSE) e *logarithmic error statistic* (LL) e *Linear-Exponential* (LINEX), os autores avaliaram a capacidade preditiva do LSSVR-APSO-CARRX, LSSVR-CARRX e CARRX na previsão da volatilidade de quatro índices de preço do mercado acionário chinês: Shanghai Composite Index (SHCI), Shenzhen Component Index (SZCI), HangSeng Index (HSI) e HuShen300 Index (HS300) no período de janeiro de 2010 a julho de 2011. Os resultados empíricos demonstraram que o modelo LSSVR-APSO-CARRX é superior aos outros dois modelos.

Em geral, os parâmetros do SVM são obtidos por meio de *grid search*, um método de força bruta. Mas tem os seguintes problemas: esbarra em ótimos locais, é demorado, requer informação *a priori* e não é capaz de otimizar concomitantemente o *kernel* e os parâmetros do SVR. Assim, Santamaría-Bonfil *et al.* (2015) rodaram um SVR com algoritmo genético (SVR_{gbc}) para previsão de volatilidade, que é capaz de selecionar ao mesmo tempo os parâmetros do *kernel* e do SVR. O modelo híbrido foi comparado com o GARCH(1,1) e o SVR com método de *grid-search* (SVR_{gs}). As previsões foram feitas em dois estágios: no ano de 2007 (treinamento e teste) e no ano de 2008 (treinamento e teste) para quatro índices de mercado - Malásia , Filipinas, México e Brasil. Com a utilização das métricas de erro absoluto médio percentual e acurácia direcional para mensurar a performance dos modelos, o SVR_{gbc} obteve resultados empíricos superiores ao GARCH(1,1) e ao SVR_{gs}. A Tabela 5.1 resume alguma das características dos 24 artigos descritos anteriormente:

Referência	Nome do Modelo	Kernel Utilizado
Fernando *et al.* (2003).	ν-SVR GARCH.	Não se aplica.
Gavrishchaka e Ganguli (2003).	SVM-based volatility.	Gaussiano.
Gavrishchaka e Banerjee (2006).	SVM-based volatility.	Gaussiano.
Chen *et al.* (2010).	SVM-GARCH.	Linear, polinomial e gaussiano.
Xu *et al.* (2011).	SVM-GARCH	Gaussiano.
Ou e Wang (2010b).	GARCH-LSSVM, EGARCH-LSSVM e GJR-LSSVM.	Gaussiano.
Ou e Wang (2010a).	RRVM	Gaussiano.
Tang *et al.* (2009b).	WSVM.	Gaussiano e Ondaleta.
Tang *et al.* (2009a).	SWSVM.	Ondaleta *spline*.
Hwang e Shin (2010).	GARCH estimado pelo SVR.	Polinomial e Gaussiano.
Shim e Lee (2010).	LSSVR.	Polinomial e Gaussiano.
Khan (2011b).	SVM-HAR.	Polinomial e Laplaciano.
Khan (2011a).	SVM-HAR-ARCH.	Laplaciano.
Wang *et al.* (2011).	SVM-MSM.	Gaussiano.
Hossain e Mohammed (2011).	RRVM.	Não se aplica.
Geng e Liang (2011).	SVRGM-GARCH.	Gaussiano.
Geng (2012).	WSVM.	Três tipos de Ondaleta.
Ou e Wang (2013).	SVRCGA.	Gaussiano.
Li-yan *et al.* (2013).	GLLSVM-PS.O	Gaussiano.
Geng e Yu (2013).	LSSVR-GARCH-SIWPSO.	Gaussiano.
Li (2014).	SVM para o APARCH.	Ondaleta.
Seethalakshmi *et al.* (2014).	PCASVM.	Gaussiano.
Liyan e Zhanfu (2012).	LSSVR-APSO-CARRX.	Gaussiano.
Santamaría-Bonfil *et al.* (2015).	SVR_{GBC}.	Linear, polinomial e gaussiano.

Tabela 5.1: *SVR na estimação e previsão da volatilidade*

Capítulo 6

Resultados empíricos

> "*The econometrician Robert Engel, an otherwise charming gentleman, invented a very complicated statistical method called GARCH and got a Nobel for it. No one tested it to see if it has any validity in real life. Simpler, less sexy methods fare exceedingly better, but they do not take you to Stockholm.*"
>
> (Taleb, 2010, p.171)

Para testar a habilidade preditiva do SVR-GARCH com misturas de *kernels* gaussianos na previsão da volatilidade dos retornos financeiros utiliza-se o índice Bovespa.

6.1 Ibovespa

O Ibovespa é o preço, em reais, das ações com maior negociabilidade e representatividade do mercado de ações do Brasil. Neste trabalho, utiliza-se a série de preços diários de fechamento do Ibovespa com início em 22 de dezembro de 2007 e fim em 04 de janeiro de 2016, totalizando 2000 observações, conforme Figura 6.1 :

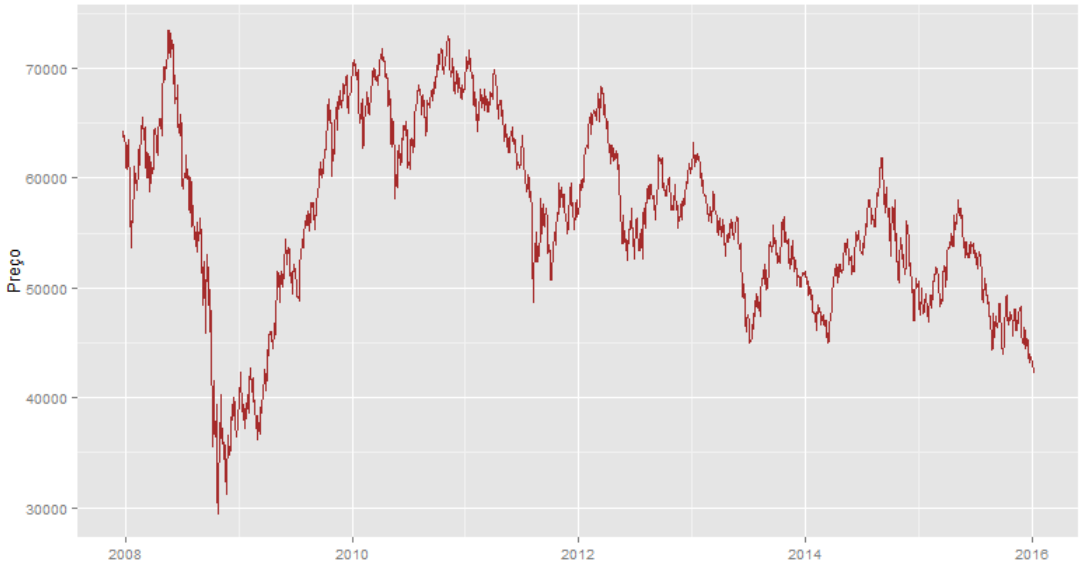

Figura 6.1: *Preço de fechamento diário Ibovespa de 22/12/2007 a 04/01/2016.*

Utiliza-se a série dos log-retornos diários dada pela seguinte transformação:

$$r_t = log\left(\frac{P_t}{P_{t-1}}\right) \tag{6.1}$$

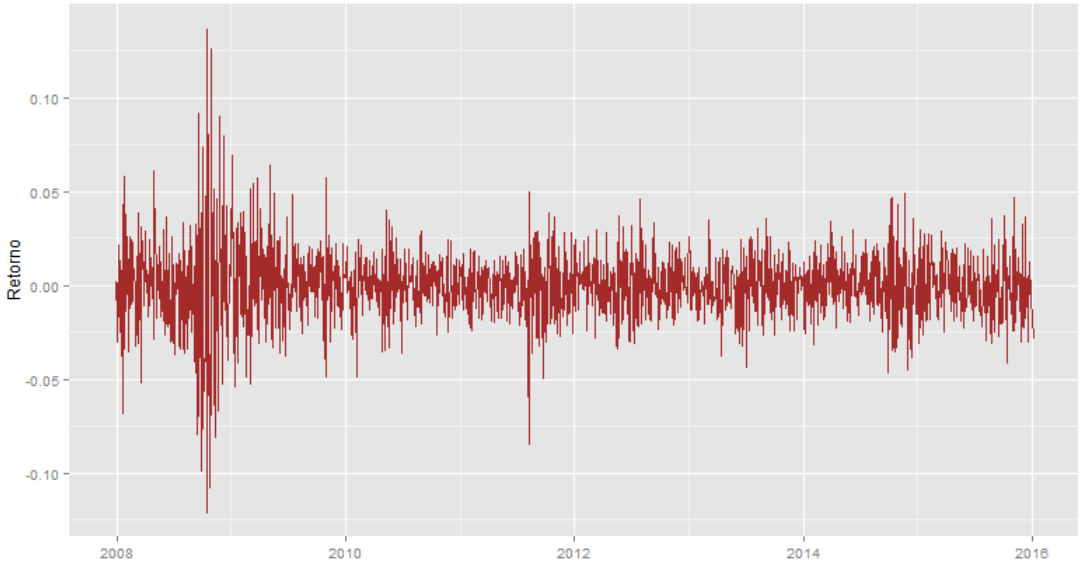

Figura 6.2: *Log-Retornos do Ibovespa de 22/12/2007 a 04/01/2016.*

Na Tabela 6.1 apresenta-se algumas estatísticas da série de log-retornos diários do Ibovespa:

Estatística	Valor
Observações	2000
Média	-0.00021
Mediana	0.0000
Assimetria	0.0825
Curtose	6.5769
Mínimo	-0.1210
Máximo	0.1368

Tabela 6.1: *Estatísticas descritivas da série dos retornos*

É evidente que a série de log-retornos é caracterizada pelo excesso de curtose em relação a distribuição normal, o que indica a presença de caudas pesadas.

6.2 Seleção dos parâmetros do SVR-GARCH

Para o ϵ-SVR é preciso selecionar os parâmetros C e ϵ. Além disso, é preciso determinar os parâmetros do *kernel* escolhido para a equação da média e da volatilidade. No caso do *kernel* gaussiano, é preciso encontrar o valor de γ em :

$$\exp\left(-\gamma \| \ x - x'\|^2\right), \quad \gamma > 0 \tag{6.2}$$

Para encontrar os parâmetros ótimos é feito o procedimento de validação descrito na seção 1.3.4. Das 2000 observações da série de retornos do Ibovespa, as primeiras 1000 observações são usadas para o treinamento, de 1001 a 1400 para validação e de 1401 à 2000 para o conjunto de teste. Assim, usa-se o conjunto de treinamento para estimar a função f da equação da média e da função g da equação da volatilidade do SVR-GARCH:

$$r_t = f\left(r_{t-1}\right) + a_t \tag{6.3}$$

$$\tilde{h}_t = g(\tilde{h}_{t-1}, a^2_{t-1}) \tag{6.4}$$

em que $\tilde{h}_t = (r_t - \bar{r})^2$ é a *proxy* da volatilidade. A análise de sensibilidade dos parâmetros é feita conforme a explicação da seção 1.3.2. A seguir apresenta-se os parâmetros ótimos do SVR-GARCH com dois *kernels* gaussianos. Os parâmetros ótimos do SVR-GARCH com um, três, quatro *kernels* e com ondaleta de Morlet podem ser vistos no Apêndice A.

6.2.1 Equação da média

Para a escolha dos melhores parâmetros do SVR para a equação da média, é feita uma análise de sensibilidade dos parâmetros C, ϵ e γ. Para a análise de sensibilidade de C, fixa-se $\epsilon = 0,0001$, $\gamma = 1,25$ e varia-se C de 0 a 10. Depois desse intervalo de *grid-search*, o erro absoluto médio continua a crescer, o que ocorre para os demais parâmetros do SVR. O menor EAM é atingido quando $C = 0.004$. Para a análise de sensibilidade de epsilon, fixa-se $\gamma = 1,25$, $C = 0.004$ e varia-se ϵ de 0 a 5. O menor EAM é atingido quando $\epsilon = 1.7405$. Para a análise de sensibilidade parâmetro gamma, fixa-se $C = 0.004$, $\epsilon = 1.7405$ e varia-se γ de 0 a 10. O menor EAM é atingido quando $\gamma = 0.576$. Assim, os parâmetros que obtiveram o

menor erro absoluto médio de previsão para a equação da média foram $C = 0.004$, $\epsilon = 1.7405$ e $\gamma = 0.576$.

Parâmetro	Variação	Parâmetro ótimo	EAM ótimo
C	[0,10]	0.004	0.01044
ϵ	[0,5]	1.7405	0.0103412
γ	[0,10]	0.576	0.0103407

Tabela 6.2: *Parâmetros ótimos da equação da média do SVR-GARCH com dois kernels gaussianos*

6.2.2 Equação da volatilidade

Do ajuste do SVR-GARCH à equação da média obtém-se o quadrado dos resíduos:

$$a_t = r_t - f(r_t) \quad \text{para} \quad i \in (2, \cdots, 1400) \tag{6.5}$$

Em seguida, realiza-se o ajuste do modelo à equação da volatilidade:

$$\tilde{h}_t = g(\tilde{h}_{t-1}, a_{t-1}^2) \quad \text{para} \quad i \in (2, \cdots, 1000) \tag{6.6}$$

em que a_t^2 é o quadrado do resíduo obtido do ajuste da equação da média. A *proxy* da volatilidade $\tilde{h}_t$ é calculada até a observação 1400 e a seleção dos parâmetros é feita com objetivo de minimizar:

$$EAM = \frac{1}{400} \sum_{t=1001}^{1400} |\tilde{h}_t - g(\tilde{h}_{t-1}, a_{t-1}^2)| \tag{6.7}$$

É importante ressaltar que para o SVR-GARCH com a combinação linear de dois *kernels* gaussianos, utiliza-se a seguinte parametrização:

$$K(x, x') = \rho \times \exp\left(-\gamma_1 \| x - x'\|^2\right) + (1 - \rho) \times \exp\left(-\gamma_2 \| x - x'\|^2\right) \tag{6.8}$$

em que ρ é a mistura ótima que deve ser determinada pelo SVR via análise de sensibilidade.

Para a análise de sensibilidade de C, fixa-se $\epsilon = 0,0001$, $\gamma_1 = 0,01$, $\gamma_2 = 0,07$, $\rho_1 = 0,5$ e varia-se C de 0 a 10. Após esse intervalo, o EAM continua a crescer. A mesma observação é válida para os outros parâmetros. O menor EAM é atingido quando $C = 0.196$. Para a análise de sensibilidade parâmetro epsilon, fixa-se $C = 0.196$, $\gamma_1 = 0.01$, $\gamma_2 = 0.07$, $\rho_1 = 0.5$ e varia-se ϵ de 0 a 0.1. O menor EAM é atingido quando $\epsilon = 0.00676$. Para a análise de sensibilidade do parâmetro γ_1, fixa-se $C = 0.196$, $\epsilon = 0.00676$, $\gamma_2 = 0.07$, $\rho = 0,8$ e varia-se γ_1 de 0 a 1. O menor EAM é atingido quando $\gamma_1 = 1$. Para a análise de sensibilidade parâmetro gamma 2, fixa-se $C = 0.196$, $\epsilon = 0.00676$, $\rho_1 = 0.5$, $\gamma_1 = 1$ e varia-se γ_2 de 0 a 1. O menor EAM é atingido quando $\gamma_2 = 1$. Para a análise de sensibilidade dos pesos da mistura, fixa-se $C = 0.196$, $\epsilon = 0.00676$,$\gamma_1 = 1$, $\gamma_2 = 1$ e varia-se ρ_1 de 0 a 1. O menor EAM é atingido quando $\rho = 0.06$.

Assim, os parâmetros que obtiveram o menor erro absoluto médio de previsão foram $C = 0.196$, $\epsilon = 0.00676$,$\gamma_1 = 1$, $\gamma_2 = 1$ e $p = 0.5$.

Parâmetro	Variação	Parâmetro ótimo	EAM ótimo
C	[0,10]	0.196	8.56×10^{-5}
ϵ	[0,0.1]	0.00676	8.55×10^{-5}
γ_1	[0,1]	1	6.72×10^{-5}
γ_2	[0,1]	1	6.65×10^{-5}
ρ	[0,1]	0.06	6.65×10^{-5}

Tabela 6.3: *Parâmetros ótimos da equação da volatilidade do SVR-GARCH com dois kernels gaussianos*

6.3 Estimação da volatilidade via GARCH

Para comparar a capacidade preditiva do modelo proposto, ajusta-se os seguintes modelos até a observação 1400 da série de retornos do Ibovespa: GARCH (1,1) com distribuição normal, t-Student, t-Student assimétrica e GED, EGARCH (1,1) com distribuição normal, t-Student, t-Student assimétrica e GED, GJR (1,1) com distribuição normal, t-Student, t-Student assimétrica e GED [1]. A estimação dos parâmetros de cada um dos modelos GARCH pode ser vista no Apêndice B.

O modelo que melhor se ajusta ao Ibovespa é o GJR com distribuição t-Student assimétrica, pois apresenta o maior valor de log-verossimilhança (log-vero) e menor valor de *Akaike Information Criteria* (AIC) e *Bayes Information Criteria*(BIC) em relação aos demais modelos estimados, conforme apresentado na Tabela 6.4.

Modelo	Log-vero	AIC	BIC
GARCH-N	3787	-5.4037	-5.3887
GARCH-t	3800	-5.421	-5.4023
GARCH-Skewed-t	3802	-5.4224	-5.3999
GARCH-GED	3800	-5.4213	-5.4026
EGARCH-N	3808	-5.4323	-5.4135
EGARCH-t	3816	-5.4427	-5.4202
EGARCH-Skewed-t	3819	-5.4454	-5.4192
EGARCH-GED	3816	-5.4425	-5.4201
GJR-N	3813	-5.4401	-5.4213
GJR-t	3819	-5.4478	-5.4253
GJR-Skewed-t	3822	-5.4503	-5.4261
GJR-GED	3820	-5.4483	-5.4258

Tabela 6.4: *Estatísticas de ajustamento.*

6.4 Avaliação das previsões

De posse das informações disponíveis até o conjunto de validação, realiza-se a previsão da volatilidade um passo a frente das 600 últimas observações. Após cada previsão, calcula-se o erro cometido pelo modelo e, posteriormente, atualiza-se as informações e repete-se o

[1] Utilizou-se o pacote rugarch de Ghalanos (2015) na versão 3.2.2 do R para a estimação dos modelos GARCH.

processo de previsão um passo à frente. As previsões da volatilidade via SVR-GARCH[2] com misturas de dois *kernels* Gaussianos [3] no conjunto de teste estão representadas na Figura 6.3:

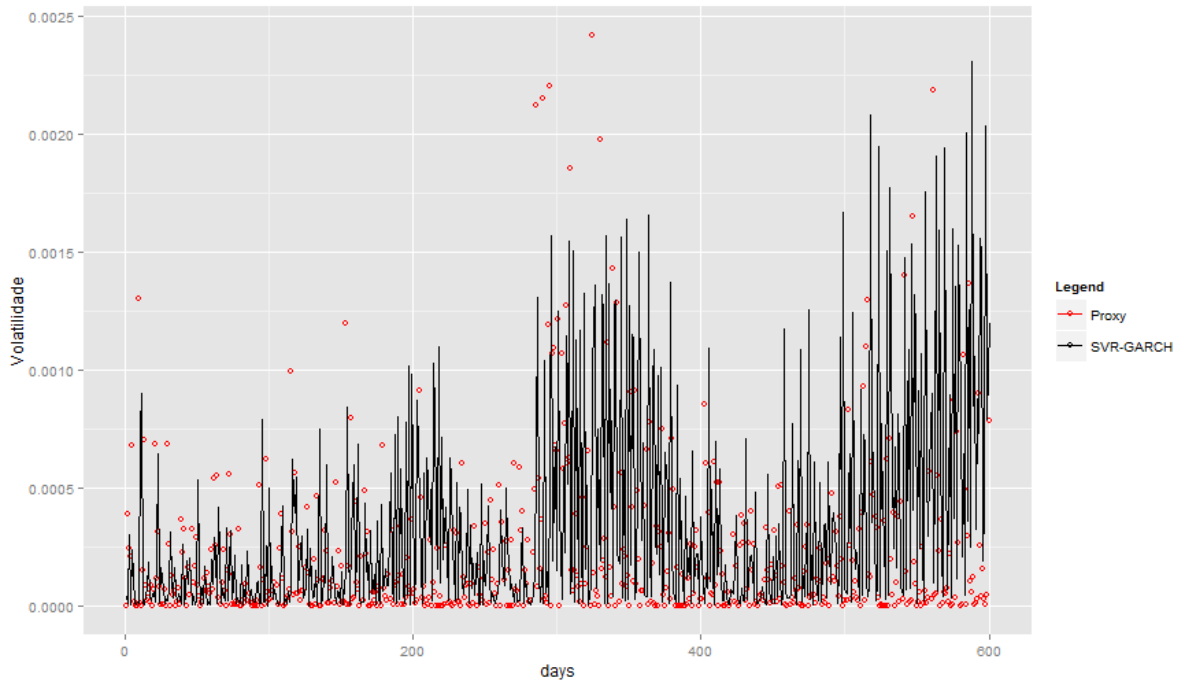

Figura 6.3: *Previsão da Volatilidade via SVR-GARCH com dois kernels gaussianos*

O Erro Quadrático Médio (MSE), Erro Quadrático Normalizado (NMSE) e a Raiz Quadrada do Erro Quadrático Médio (RMSE) de previsão um período a frente da volatilidade dos modelo testados para os retornos do Ibovespa estão na Tabela 6.5:

[2]Utilizou-se o pacote kernlab de Karatzoglou *et al.* (2004) na versão 3.2.2 do R.

[3]Os programas foram executados na versão 3.2.2 do R numa máquina com apenas 2 núcleos. O tempo de programação pode ser encontrado na Tabela A.11. Os gráficos das demais previsões via SVR-GARCH podem ser encontrados no Apêndice A.

	MSE		RMSE		NMSE	
	Absoluto	Relativo	Absoluto	Relativo	Absoluto	Relativo
Random walk	1.62×10^{-7}	1.025	4.025×10^{-4}	1.0125	-	-
SVR-GARCH-Mexican	1.58×10^{-7}	1.0000	3.97×10^{-4}	1.0000	0.04165	1.0000
SVR-GARCH [1]	4.44×10^{-7}	2.809	6.66×10^{-4}	1.6754	0.04705	1.1296
SVR-GARCH [2]	3.20×10^{-7}	1.976	5.66×10^{-4}	1.4251	0.05909	1.4187
SVR-GARCH [3]	3.12×10^{-7}	1.976	5.58×10^{-4}	1.4055	0.06087	1.4614
SVR-GARCH [4]	4.60×10^{-7}	2.914	6.78×10^{-4}	1.7071	0.04526	1.0866
SVR-GARCH-Morlet	4.27×10^{-7}	2.704	6.53×10^{-4}	1.6442	0.04533	1.0883
GARCH-N	2.24×10^{-4}	1422	0.01499	37.7106	0.08024	1.9265
GARCH-t	2.23×10^{-4}	1415	0.01495	37.6100	0.07943	1.9070
GARCH-Skewed-t	2.22×10^{-4}	1410	0.01493	37.5597	0.07961	1.9114
GARCH-GED	2.23×10^{-4}	1415	0.01496	37.6352	0.07969	1.9133
EGARCH-N	2.31×10^{-4}	1462	0.0152	38.2389	0.08964	2.1522
EGARCH-t	2.31×10^{-4}	1462	0.0152	38.2389	0.09212	2.2117
EGARCH-Skewed-t	2.31×10^{-4}	1467	0.01523	38.3144	0.09155	2.1980
EGARCH-GED	2.31×10^{-4}	1462	0.0152	38.2389	0.09144	2.1954
GJR-N	2.13×10^{-4}	1353	0.01463	36.8050	0.07674	1.8424
GJR-t	2.13×10^{-4}	1352	0.01461	36.7547	0.07785	1.8691
GJR-Skewed-t	2.14×10^{-4}	1355	0.01464	36.8301	0.07768	1.8650
GJR-GED	2.13×10^{-4}	1353	0.01463	36.8050	0.07767	1.8648

Nota: SVR-GARCH 1, 2, 3, 4 indicam o uso de um, dois, três e quatro *kernels* gaussianos na Equação da volatilidade, respectivamente. O erro relativo foi obtido pela razão do erro absoluto de cada modelo em relação ao erro do modelo com melhor desempenho para cada métrica.

Tabela 6.5: *Estatística de erro para previsão diária.*

De acordo com a métrica do Erro Quadrático Médio (MSE), Raiz Quadrada do Erro Quadrático Médio (RMSE) e Erro Quadrático Normalizado (NMSE), o modelo SVR-GARCH com *kernel* de ondaleta de Chapéu Mexicano obteve o melhor desempenho preditivo. Em segundo lugar, ficou o modelo *random walk*. O NMSE do *random walk* apresentou valor indeterminado. Nota-se ainda que as previsões desse modelo possuem erro 2,5% maior em relação ao melhor modelo segundo o MSE.

O SVR-GARCH com misturas de dois, três e quatro *kernels* gaussianos obteve resultados superiores ao SVR-GARCH com *kernel* de ondaleta de Morlet e um *kernel* gaussiano, o que mostra a relevância da proposta desta dissertação. Além disso, assim como neste trabalho, Li (2014) mostrou que o *kernel* de ondaleta de Morlet tem resultados preditivos superiores ao SVR com apenas um *kernel* gaussiano. Não obstante, a mistura de funções núcleos gaussianas também foi capaz de superar a ondaleta de Morlet.

Dentre os modelos GARCH, o GJR (1,1) com distribuição normal, t-Student, t-Student assimétrica e GED apresentaram os melhores desempenhos preditivos. É importante destacar a superioridade dos modelos SVR-GARCH em relação aos demais modelos GARCH, o que vai ao encontro dos achados da literatura (Chen *et al.*, 2010; Li, 2014; Santamaría-Bonfil *et al.* , 2015). Ressalta-se ainda que o SVR-GARCH com função núcleo de ondaleta de Chapéu Mexicano obteve resultados ligeiramente superiores ao *random walk*. Assim, com o aprimoramento do SVR-GARCH talvez seja possível melhorar ainda mais sua performance. Por exemplo, Lu *et al.* (2009a) mostraram que um modelo SVR combinado com análise de componentes independentes superou o *random walk* e o SVR simples na previsão de preços de índice de ações.

A Tabela 6.6 mostra que, em relação aos *kernels* utilizados, o SVR-GARCH com kernel de ondaleta de Chapéu Mexicano apresentou o menor número de suportes vetoriais na fase de treinamento para a Equação da Volatilidade, o que evidencia maior capacidade de generalização e eficiência computacional (Xia *et al.*, 2005). De fato, nota-se que o SVR-GARCH-Mexican obteve as melhores previsões de volatilidade em relação aos demais modelos.

Modelo	**Número de suportes vetoriais**
SVR-GARCH-Mexican	812
SVR-GARCH [1]	1399
SVR-GARCH [2]	1181
SVR-GARCH [3]	1274
SVR-GARCH [4]	1355
SVR-GARCH-Morlet	1381

Tabela 6.6: *Número de suportes vetoriais do SVR*

Para comparar a capacidade preditiva de dois modelos utiliza-se o teste bilateral Diebold-Mariano para a diferença da função de perda do Erro Quadrático Médio (MSE), que é dado pelas seguintes hipóteses nulas e alternativas:

$$H_0 : (\tilde{h}_t - \hat{h}_{0,t})^2 - (\tilde{h}_t - \hat{h}_{1,t})^2 = 0 \quad versus \quad H_1 : (\tilde{h}_t - \hat{h}_{0,t})^2 - (\tilde{h}_t - \hat{h}_{1,t}) \neq 0,$$

em que $\hat{h}_{0,t}$ é a volatilidade estimada pelo modelo SVR-GARCH-Mexican, $\hat{h}_{1,t}$ é a volatilidade do modelo testado e $\tilde{h}$ é a *proxy* da volatilidade dada por 1.10. Assim, se a hipótese nula for rejeitada, tem-se evidência de que o SVR-GARCH-Mexicané superior ao outro. Ademais, a estatística do teste Diebold-Mariano (DM) é dada por:

$$DM = \frac{1}{\sqrt{600}} \frac{1}{\sqrt{\hat{V}(d)}} \sum_{t=1401}^{2000} (\tilde{h}_t - \hat{h}_{0,t})^2 - (\tilde{h}_t - \hat{h}_{1,t})^2 \sim N(0,1) \tag{6.9}$$

em que $d = \sum_{t=1401}^{2000} (e_{t_0})^2 - (e_{t_1})^2$, $\tilde{h}$ é a *proxy* da volatilidade, $\hat{h}^2_{0,t}$ é a volatilidade estimada do modelo SVR-GARCH-Mexican, $\hat{h}^2_{1,t}$ é a volatilidade estimada do modelo testado e $\hat{V}(d)$ é uma estimativa da variância assintótica de d. A Tabela 6.7 reporta os valores da estatística do teste Diebold-Mariano para a diferença da função de perda do Erro Quadrático Médio (MSE).

Modelo	Estatística DM	P-valor
Random walk	-0.36	0.7
SVR-GARCH [1]	-6.8	2×10^{-11}
SVR-GARCH [2]	-6.8	2×10^{-11}
SVR-GARCH [3]	-8.1	4×10^{-15}
SVR-GARCH [4]	-6.6	1×10^{-10}
SVR-GARCH-Morlet	-6.7	5×10^{-11}
GARCH-N	-51	2×10^{-16}
GARCH-t	-52	2×10^{-16}
GARCH-Skewed-t	-51	2×10^{-16}
GARCH-GED	-51	2×10^{-16}
EGARCH-N	-42	2×10^{-16}
EGARCH-t	-42	2×10^{-16}
EGARCH-Skewed-t	-41	2×10^{-16}
EGARCH-GED	-42	2×10^{-16}
GJR-N	-43	2×10^{-16}
GJR-t	-44	2×10^{-16}
GJR-Skewed-t	-43	2×10^{-16}
GJR-GED	-44	2×10^{-16}

Tabela 6.7: *Teste Diebold-Mariano (Benchmark:SVR-GARCH-Mexican, previsão um período a frente).*

O teste indica que não há evidências de rejeição da hipótese nula de igualdade preditiva do SVR-GARCH com *kernel* de ondaleta de Chapéu Mexicano em relação ao *random walk*. No entanto, em relação aos demais modelos, o SVR-GARCH-Mexican apresenta p-valores[4] bem próximos a zero. Por conseguinte, rejeita-se a hipótese H_0 de igualdade dos erros quadráticos médios de previsão com nível de significância menor que 1%. Portanto, o SVR-GARCH-Mexican produz previsões mais acuradas em relação aos modelos: SVR-GARCH com um, dois, três e quatro *kernels* gaussianos, SVR-GARCH com kernel de ondaleta de Morlet, GARCH (1,1), EGARCH (1,1) e GJR (1,1) com distribuição normal, t-Student, t-Student assimétrica e GED.

[4]É importante ressaltar que, segundo (Taleb, 2016), a meta-distribuição do p-valor é extremamente assimétrica à direita, volátil e varia bastante entre repetições de um conjunto de cópias de processos estocásticos idênticos. Além disso, a interpretação desses resultados empíricos deve ser feita à luz dos seis princípios sobre a utilização e interpretação do p-valor e da significância estatística divulgados pela Associação Americana de Estatística (Wasserstein e Lazar, 2016).

Capítulo 7

Conclusão

> "*With the increasing interest in using complicated econometric techniques for volatility forecasting, our research strikes a warning bell. For those who are interested in forecasts with reasonable predictive accuracy, the best forecasting models may well be the simplest ones.*"
>
> (Dimson e Marsh, 1990, p.420)

A previsão de séries temporais é essencial na atividade financeira. Nos últimos 20 anos, a utilização do *Support Vector Regression* (SVR) na previsão de séries temporais obteve grande sucesso (Sankar *et al.*, 2009). Assim, o objetivo desse trabalho foi aprimorar as previsões da volatilidade do SVR com base no GARCH(1,1) (denominado SVR-GARCH), modelando os regimes de mercado por meio de misturas de kernels gaussianos. Considerando a existência de k regimes, optou-se por utilizar a combinação linear de um, dois, três e quatro *kernels* gaussianos, pois, em geral, a mistura de funções núcleos apresenta resultados preditivos superiores em relação ao SVR com apenas um *kernel* (Huang *et al.*, 2014). Além disso, a mistura de distribuições normais pode capturar as mudanças de regimes de mercado e características não-lineares dos retornos financeiros, como caudas pesadas, assimetria e os agrupamentos de volatilidade (Guidolin, 2011; Haas *et al.*, 2004).

Os resultados empíricos desta dissertação mostram evidências da superioridade do SVR-GARCH com kernel de ondaleta de Chapéu Mexicano e do *random walk* na previsão da volatilidade de um período a frente para dados diários do Ibovespa em relação ao SVR-GARCH com a combinação de um, dois, três e quatro *kernels* gaussianos, SVR-GARCH com kernel de ondaleta de Morlet, GARCH(1,1), EGARCH(1,1) e GJR(1,1) com distribuição normal, t-Student, t-Student assimétrica e distribuição de erro generalizada (GED), de acordo com as métricas do Erro Quadrático Médio (MSE), Erro Quadrático Normalizado (NMSE), Raiz Quadrada do Erro Quadrático Médio (RMSE) e o teste Diebold-Mariano de igualdade de acurácia preditiva. Além disso, o SVR-GARH com misturas de kernels gaussianos obteve resultado superior ao SVR-GARCH com ondaleta de Morlet e um kernel gaussiano, o que confirma o mérito da proposta deste trabalho.

É importante destacar que este trabalho têm as seguintes limitações: as previsões foram feitas apenas para um período a frente, utilizou-se apenas a estrutura GARCH, testou-se apenas uma *proxy* para a volatilidade, utilizou-se uma função de perda para o SVR. Para traba-

lhos futuros, sugere-se os seguintes pontos: comparar o SVR-GARCH com modelos da Física Estatística, Estatística Mecânica, outras técnicas de *machine learning* (por exemplo, processos gaussianos, *random forests*, *Deep Learning* (Heaton *et al.*, 2016; Längkvist *et al.*, 2014) ou *Deep Kernel Learning* (Wilson *et al.*, 2015)), com o Mixed Normal-GARCH, Markov-Switching GARCH e BetaSkew-t-EGARCH, usar a volatilidade realizada como *proxy* para a volatilidade, utilizar outras misturas de *kernels*, usar combinações não-lineares de *kernels*, usar um algoritmo genético caótico ou algoritmo de otimização de partícula de enxame para a escolha dos parâmetros do SVR, desenvolver um *kernel* para a volatilidade dos retornos financeiros.

Apêndice A

Parâmetros ótimos do SVR

As tabelas abaixo mostram os parâmetros ótimos da Equação da Média e da Volatilidade para o SVR-GARCH com um kernel gaussiano.

Parâmetro	**Variação**	**Parâmetro ótimo**	**EAM ótimo**
C	[0,10]	0.004	0.01044659
ϵ	[0,5]	1.7405	0.01034125
γ	[0,10]	0.576	0.01034074

Tabela A.1: *Parâmetros ótimos da equação da média um kernel Gaussiano.*

Parâmetro	**Variação**	**Parâmetro ótimo**	**EAM ótimo**
C	[0,10]	3.136	6.47×10^{-5}
ϵ	[0,0.1]	1×10^{-5}	6.39×10^{-5}
γ_1	[0,1]	1	6.39×10^{-5}

Tabela A.2: *Parâmetros ótimos da equação da volatilidade um kernel Gaussiano.*

A Figura A.1 mostra as previsões da volatilidade do SVR-GARCH com um *kernel* Gaussiano:

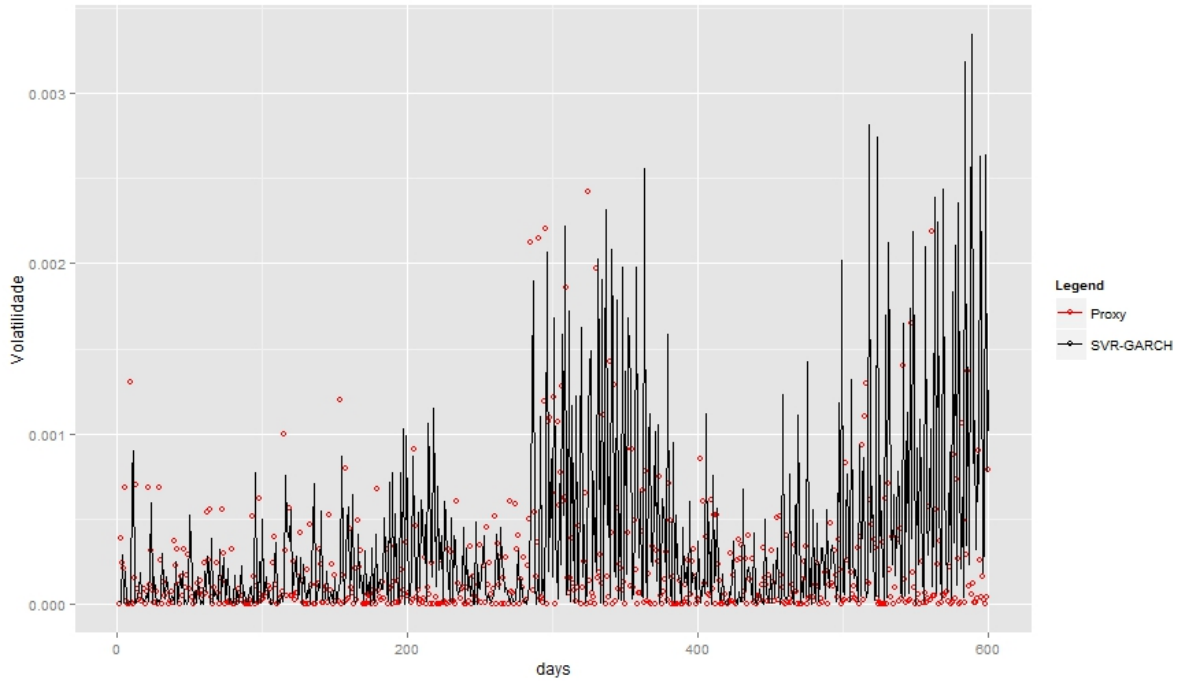

Figura A.1: *Previsão da Volatilidade via SVR-GARCH com um kernel Gaussiano.*

As tabelas abaixo mostram os parâmetros ótimos da Equação da Média e da Volatilidade para o SVR-GARCH com três kernels gaussianos.

Parâmetro	Variação	Parâmetro ótimo	EAM ótimo
C	[0,10]	0.004	0.01044659
ϵ	[0,5]	1.7405	0.01034125
γ	[0,10]	0.576	0.01034074

Tabela A.3: *Parâmetros ótimos da equação da média três kernels Gaussiano.*

Parâmetro	Variação	Parâmetro ótimo	EAM ótimo
C	[0,10]	0.169	8.56×10^{-5}
ϵ	[0,0.1]	0.00361	8.54×10^{-5}
γ_1	[0,1]	1	7.26×10^{-5}
γ_2	[0,1]	1	6.80×10^{-5}
γ_3	[0,1]	1	9.11×10^{-5}
ρ_1	[0,1]	0.93	6.68×10^{-5}
ρ_2	[0,1]	0.02	6.68×10^{-5}
ρ_3	[0,1]	0.05	-

Tabela A.4: *Parâmetros ótimos da equação da volatilidade três kernels Gaussiano.*

A Figura A.2 mostra as previsões da volatilidade do SVR-GARCH com três *kernels* Gaussiano:

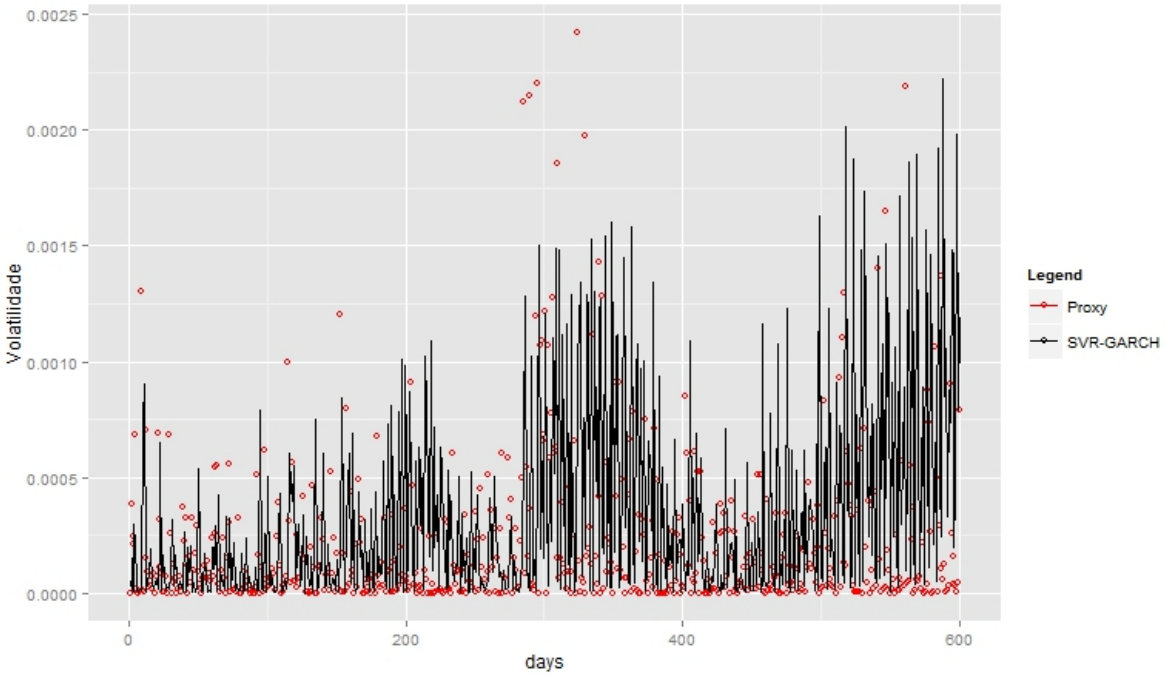

Figura A.2: *Previsão da Volatilidade via SVR-GARCH com três kernels gaussianos.*

As tabelas abaixo mostram os parâmetros ótimos da Equação da Média e da Volatilidade para o SVR-GARCH com quatro kernels gaussianos.

Parâmetro	Variação	Parâmetro ótimo	EAM ótimo
C	[0,10]	0.004	0.01044659
ϵ	[0,5]	1.7405	0.01034125
γ	[0,10]	0.576	0.01034074

Tabela A.5: *Parâmetros ótimos da equação da média com quatro kernels gaussianos.*

Parâmetro	Variação	Parâmetro ótimo	EAM ótimo
C	[0,10]	10	0.0001592653
ϵ	[0,0.1]	0.00144	0.000159134
γ_1	[0,1]	1	6.26×10^{-5}
γ_2	[0,1]	0.0064	6.22×10^{-5}
γ_3	[0,1]	0.0036	6.21×10^{-5}
γ_4	[0,1]	0.0036	6.21×10^{-4}
ρ_1	[0,1]	0.21	6.18×10^{-5}
ρ_2	[0,1]	0.35	6.18×10^{-5}
ρ_3	[0,1]	0.18	6.18×10^{-5}
ρ_4	[0,1]	0.26	-

Tabela A.6: *Parâmetros ótimos da equação da volatilidade com quatro kernels gaussianos.*

As tabelas abaixo mostram os parâmetros ótimos da Equação da Média e da Volatilidade para o SVR-GARCH com kernel de Morlet.

Parâmetro	Variação	Parâmetro ótimo	EAM ótimo
C	[0,10]	0.004	0.01044659
ϵ	[0,5]	1.7405	0.01034125
γ	[0,10]	0.576	0.01034074

Tabela A.7: *Parâmetros ótimos da equação da média do SVR-GARCH com kernel de Morlet.*

Parâmetro	Variação	Parâmetro ótimo	EAM ótimo
C	[0,10]	10	6.80×10^{-5}
ϵ	[0,0.1]	0.00064	6.74×10^{-5}

Tabela A.8: *Parâmetros ótimos da equação da volatilidade do SVR-GARCH com kernel de Morlet.*

A Figura A.3 mostra as previsões da volatilidade do SVR-GARCH SVR-GARCH-Morlet:

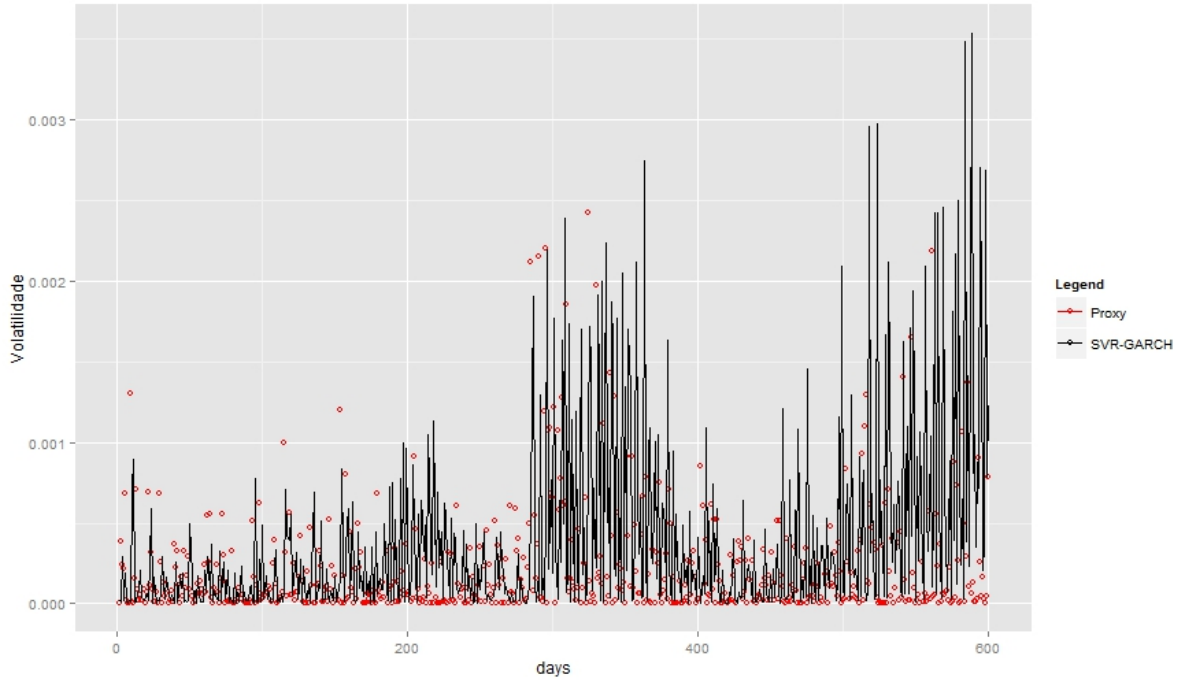

Figura A.3: *Previsão da Volatilidade via SVR-GARCH-Morlet.*

As tabelas abaixo mostram os parâmetros ótimos da Equação da Média e da Volatilidade para o SVR-GARCH com ondaleta de Chapéu Mexicano.

Parâmetro	Variação	Parâmetro ótimo	EAM ótimo
C	[0,10]	0.004	0.01044659
ϵ	[0,5]	1.7405	0.01034125

Tabela A.9: *Parâmetros ótimos da equação da média do SVR-GARCH com kernel de ondaleta de Chapéu Mexicano.*

Parâmetro	Variação	Parâmetro ótimo	EAM ótimo
C	[0,10]	0.001	0.000214537
ϵ	[0,0.1]	0.04225	0.0001231948
a	[0,0.1]	0.03969	8.96×10^{-5}

Tabela A.10: *Parâmetros ótimos da equação da volatilidade do SVR-GARCH com kernel de ondaleta de Chapéu Mexicano.*

A Figura A.4 mostra as previsões da volatilidade do SVR-GARCH *kernel* de ondaleta de Chapéu Mexicano:

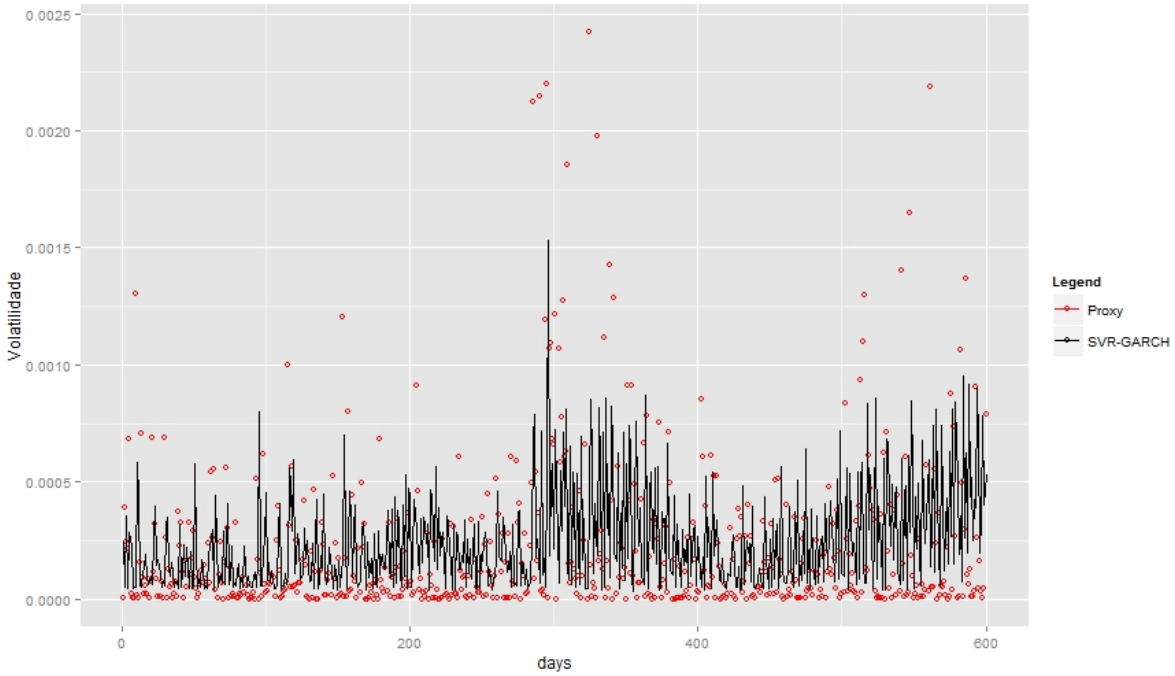

Figura A.4: *Previsão da Volatilidade via SVR-GARCH-Mexican.*

A Tabela A.11 apresenta o tempo de execução de cada modelo SVR-GARCH:

Modelo	Tempo
SVR-GARCH [1]	18.01 horas
SVR-GARCH [2]	1.58 dias
SVR-GARCH [3]	1.77 dias
SVR-GARCH [4]	3.17 dias
SVR-GARCH-Morlet	15.06 horas
SVR-GARCH-Mexican	1.37 dias

Tabela A.11: *Tempo de execução dos programas.*

Apêndice B

Estimação GARCH, EGARCH, GJR

As tabelas abaixo mostram a estimação dos parâmetros para GARCH(1,1), EGARCH(1,1), GJR(1,1) com distribuição Normal(N), t-Student (t), t-Student assimétrica (Skewed-t) e distribuição de erro generalizada (GED).

Parâmetro	**GARCH-N**	**GARCH-t**	**GARCH-Skewed-t**	**GARCH-GED**
μ	0.00016 [0.00038]	0.00013 [0.00037]	0.00003 [0.00038]	0.00023 [0.00036]
α_0	0.000006 [0.000004]	0.000005 [0.000003]	0.000005 [0.000003]	0.000006 [0.000004]
α_1	0.09170 [0.01444]	0.08215 [0.01552]	0.08173 [0.01611]	0.08648 [0.01571]
β_1	0.89011 [0.01658]	0.90105 [0.01742]	0.90276 [0.01842]	0.89522 [0.01781]
ν	-	9.44203 [2.20583]	9.5184 [2.2465]	1.51518 [0.08222]
ι	-	-	0.92965 [0.03431]	-
log-vero.	3787	3800	3802	3800
AIC	-5.4037	-5.421	-5.4224	-5.4213
BIC	-5.3887	-5.4023	-5.3999	-5.4026

Nota: cada modelo GARH foi estimado com uma Normal (N), t-Student (t), t-Student assimétrica (Skewed-t) e distribuição de erro generalizada (GED). O erro padrão está entre chaves.

Tabela B.1: *Estimação GARCH (1,1).*

Parâmetro	**EGARCH-N**	**EGARCH-t**	**EGARCH-Skewed-t**	**EGARCH-GED**
μ	-0.00063 [0.00043]	-0.00041 [0.00052]	-0.00061 [0.00034]	-0.00036 [0.00038]
α_0	-0.115930 [0.01371]	-0.10451 [0.02345]	-0.102699 [0.00884]	-0.113074 [0.00275]
α_1	-0.097849 [0.01285]	-0.09549 [0.01657]	-0.096679 [0.01346]	-0.096422 [0.014070]
β_1	0.98544 [0.00156]	0.98732 [0.00322]	0.98736 [0.001319]	0.98637 [0.00021]
γ	0.13980 [0.022547]	0.13277 [0.05286]	0.13409 [0.02581]	0.13535 [0.02360]
ν	-	11.75434 [6.08829]	11.53216 [1.73883]	1.59870 [0.00477]
ι	-	-	0.913778 [0.034140]	-
log-vero.	3808	3816	3819	3816
AIC	-5.4323	-5.4427	-5.4454	-5.4425
BIC	-5.4135	-5.4202	-5.4192	-5.4201

Nota: cada modelo EGARH foi estimado com uma Normal (N), t-Student (t), t-Student assimétrica (Skewed-t) e distribuição de erro generalizada (GED). O erro padrão está entre chaves.

Tabela B.2: *Estimação EGARCH (1,1).*

Parâmetro	GJR-N	GJR-t	GJR-Skewed-t	GJR-GED
μ	0.00049 [0.00036]	0.00035 [0.00048]	0.00050 [0.00046]	-0.00028 [0.00034]
ω	0.000005 [0.000002]	0.000005 [0.000009]	0.000005 [0.000008]	0.000005 [0.000004]
α_0	0.00638 [0.00641]	0.00626 [0.02017]	0.00649 [0.01941]	0.00620 [0.00520]
β_1	0.90617 [0.00998]	0.90880 [0.03111]	0.90910 [0.03034]	0.90746 [0.00298]
γ_1	0.1455 [0.01816]	0.13959 [0.05247]	0.14210 [0.05320]	0.14099 [0.00967]
ν	-	13.24738 [5.32473]	12.81156 [4.54425]	1.62755 [0.08796]
ι	-	-	0.91599 [0.03455]	-
log-vero.	3813	3819	3822	3820
AIC	-5.4401	-5.4478	-5.4503	-5.4483
BIC	-5.4213	-5.4253	-5.4241	-5.4258

Nota: cada modelo GJR foi estimado com uma Normal(N), t-Student(t), t-Student assimétrica(Skewed-t) e distribuição de erro generalizada (GED). O erro padrão está entre chaves.

Tabela B.3: *Estimação GJR (1,1).*

Referências

Alexander e Lazar (2006) Carol Alexander e Emese Lazar. Normal mixture GARCH(1,1): applications to exchange rate modelling. *Journal of Applied Econometrics*, 21(3):307–336. doi: 10.1002/jae.849. Citado na pág. 22

Amendola e Candila (2016) A Amendola e V Candila. Evaluation of volatility predictions in a VaR framework. *Quantitative Finance*, 16(5):695–709. ISSN 1469-7688. doi: 10.1080/14697688.2015.1062122. Citado na pág. 6

Andersen e Bollerslev (1998) Torben G. Andersen e Tim Bollerslev. Answering the Skeptics: Yes ARCH Models Do Provide Good Volatility Forecasts. *Internationl Economic Review*, 39(4):885–905. Citado na pág. 5

Ang e Timmermann (2012) Andrew Ang e Allan Timmermann. Regime Changes and Financial Markets. *Annual Review of Financial Economics*, 4(1):313–337. doi: 10.1146/annurev-financial-110311-101808. Citado na pág. 3, 22

Arlot e Celisse (2010) Sylvain Arlot e Alain Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79. doi: 10.1214/09-SS054. Citado na pág. 5

Bae *et al.* **(2014)** Geum Il Bae, Woo Chang Kim e John M. Mulvey. Dynamic asset allocation for varied financial markets under regime switching framework. *European Journal of Operational Research*, 234(2):450–458. doi: 10.1016/j.ejor.2013.03.032. Citado na pág. 3, 22

Bai *et al.* **(2003)** Xuezheng Bai, Jeffrey R. Russell e George C. Tiao. Kurtosis of GARCH and stochastic volatility models with non-normal innovations. *Journal of Econometrics*, 114(2):349–360. doi: 10.1016/S0304-4076(03)00088-5. Citado na pág. 22

BenSaïda (2015) Ahmed BenSaïda. The frequency of regime switching in financial market volatility. *Journal of Empirical Finance*, 32:63–79. doi: 10.1016/j.jempfin.2015.03.005. Citado na pág. i, ii, 3, 22

Bishop (2006) Christopher M. Bishop. *Pattern Recognition and Machine learning*. Springer Science+Business Media. ISBN 9780387310732. Citado na pág. 35

Black (1976) Fischer Black. Studies of Stock Price Volatility Changes. *Proceedings of the Business and Economics Section of the American Statistical Association*, páginas 177–181. Citado na pág. 10

Bollerslev (1986) Tim Bollerslev. Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, 31:307–327. Citado na pág. 10, 13, 14

Bollerslev (1987) Tim Bollerslev. A Conditionally Hetroskedastic Time Series Model for Speculative Prices and Rates of Return, 1987. Citado na pág. 10

Bollerslev (2008) Tim Bollerslev. Glossary to arch (garch). *CREATES Research Papers*, página 44. doi: http://dx.doi.org/10.1093/acprof:oso/9780199549498.003.0008. Citado na pág. 10, 17

Bollerslev *et al.* (1994) Tim Bollerslev, Robert Engle e Daniel B. Nelson. ARCH models. Em *Handbook of Econometrics*, volume 4, páginas 2959–3038. Elsevier. Citado na pág. 10

Boser *et al.* (1992) Bernhard E. Boser, Isabelle M. Guyon e Vladimir N. Vapnik. A Training Algorithm for Optimal Margin Classifiers. *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, páginas 144–152. doi: 10.1.1.21.3818. Citado na pág. 2, 39

Brailsford e Faff (1996) Timothy J Brailsford e Robert W Faff. An evaluation of volatility forecasting techniques. *Journal of Banking and Finance*, 20:419–438. Citado na pág. 1, 6, 17

Breiman (2001) Leo Breiman. Statistical Modeling: The Two Cultures. *Statistical Science*, 16(3):199–215. ISSN 2168-8745. doi: 10.2307/2676681. Citado na pág. 24

Brooks (2001) Chris Brooks. A Double-threshold GARCH Model for the French Franc/-Deutschmark exchange rate. *Journal of Forecasting*, 20(2):135–143. doi: 10.1002/1099-131X(200103)20:2<135::AID-FOR780>3.0.CO;2-R. Citado na pág. 5

Brooks e Persand (2003) Chris Brooks e Gita Persand. Volatility forecasting for risk management. *Journal of Forecasting*, 22(1):1–22. doi: 10.1002/for.841. Citado na pág. 5

Brownlees e Gallo (2009) C. T. Brownlees e G. M. Gallo. Comparison of Volatility Measures: a Risk Management Perspective. *Journal of Financial Econometrics*, 8(1): 29–56. doi: 10.1093/jjfinec/nbp009. Citado na pág. 9

Cao e Tay (2001) Lijuan Cao e Francis E.H Tay. Financial Forecasting Using Support Vector Machines. *Neural Computing & Applications*, 10(2):184–192. doi: 10.1007/s005210170010. Citado na pág. 2, 4, 45

Cao e Tay (2003) L.J. Cao e F.E.H. Tay. Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on Neural Networks*, 14(6):1506–1518. doi: 10.1109/TNN.2003.820556. Citado na pág. 2, 4, 6

Casella e Berger (2001) George Casella e Roger L. Berger. *Statistical Inference*. Duxbury Press, second ed. ISBN 978-0-534-24312-8. Citado na pág. 15

Cavalcante *et al.* (2016) Rodolfo C. Cavalcante, Rodrigo C. Brasileiro, Victor L.F. Souza, Jarley P. Nobrega e Adriano L.I. Oliveira. Computational Intelligence and Financial Markets: A Survey and Future Directions. *Expert Systems with Applications*, 55:194–211. doi: 10.1016/j.eswa.2016.02.006. Citado na pág. i, ii, 1, 39

Chen *et al.* (2010) Shiyi Chen, Wolfgang K Härdle e Kiho Jeong. Forecasting Volatility with Support Vector Machine-Based GARCH Model. *Journal of Forecasting*, 433(29): 406–433. doi: 10.1002/for.1134. Citado na pág. 1, 2, 4, 5, 47, 52, 59

Cherkassky e Ma (2004) Vladimir Cherkassky e Yunqian Ma. Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, 17(1):113–126. doi: 10.1016/S0893-6080(03)00169-2. Citado na pág. 43, 44

Cherkassky e Mulier (2007) Vladimir Cherkassky e Filip Mulier. *Learning from data*. John Wiley & Sons, Inc. Citado na pág. v, 33

Chernoff (1952) Herman Chernoff. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the sum of Observations. *The Annals of Mathematical Statistics*, 23 (4):493–507. doi: 10.1214/aoms/1177729330. Citado na pág. 28

Choudhry e Wu (2008) T. Choudhry e H. A. O. Wu. Forecasting Ability of GARCH vs Kalman Filter Method : Evidence from Daily UK Time-Varying Beta. *Journal of Forecasting*, 689:670–689. doi: 10.1002/for.1096. Citado na pág. 1

Cont (2001) R. Cont. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1(2):223–236. doi: 10.1088/1469-7688/1/2/304. Citado na pág. 10

Cortes *et al.* (2009) Corinna Cortes, M. Mohri e A. Rostamizadeh. Learning non-linear combinations of kernels. Em *Advances in Neural Information*, páginas 396–404. Citado na pág. 36

Daubechies (1992) Ingrid Daubechies. *Ten Lectures of Wavelets*. Springer-Verlag. Citado na pág. 36

Diebold (2004) Francis X. Diebold. The nobel memorial prize for Robert F. Engle. *Scandinavian Journal of Economics*, 106(2):165–185. doi: 10.1111/j.1467-9442.2004.00360.x. Citado na pág. 9

Diebold e Mariano (1995) Francis X. Diebold e Roberto S. Mariano. Comparing Predictive Accuracy. *Journal of Business & Economic Statistics*, 13(3):253–263. doi: 10.1080/07350015.1995.10524599. Citado na pág. 3, 6

Dimson e Marsh (1990) Elroy Dimson e Paul Marsh. Volatility forecasting without data-snooping. *Journal of Banking and Finance*, 14(2-3):399–421. ISSN 03784266. doi: 10.1016/0378-4266(90)90056-8. Citado na pág. 1, 17, 63

Ding *et al.* (2014) Shifei Ding, Fulin Wu e Zhongzhi Shi. Wavelet twin support vector machine. *Neural Computing and Applications*, 25(6):1241–1247. doi: 10.1007/s00521-014-1596-y. Citado na pág. 37, 38

Ding *et al.* (1993) Zhuanxin Ding, Clive W.J. Granger e Robert F. Engle. A long memory property of stock market returns and a new model. *Journal of Empirical Finance*, 1(1): 83–106. doi: 10.1016/0927-5398(93)90006-D. Citado na pág. 50

Domingos (2015) Pedro Domingos. *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. Basic Books. ISBN 978-0465065707. Citado na pág. 1, 23

Engle (1982) Robert F Engle. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, 50(4):987–1007. Citado na pág. 10, 12

Fan *et al.* (2014) Jianqing Fan, Lei Qi e Dacheng Xiu. Quasi-Maximum Likelihood Estimation of GARCH Models With Heavy-Tailed Likelihoods. *Journal of Business and Economic Statistics*, 32(2):178–191. doi: 10.1080/07350015.2013.840239. Citado na pág. 15

Fasshauer (2011) Gregory E Fasshauer. Positive definite kernels: past, present and future. *Dolomite Research Notes on Approximation*, páginas 1–48. Citado na pág. 35

Fender (2003) Thomas Fender. *Empirische Risiko-Minimierung*. Tese de Doutorado. Citado na pág. 46

Fernandez e Steel (1998) Carmen Fernandez e Mark F. J. Steel. On Bayesian Modeling of Fat Tails and Skewness. *Journal of the American Statistical Association*, 93(441):359. doi: 10.2307/2669632. Citado na pág. 16

Fernando *et al.* (2003) Pérez-Cruz Fernando, Julio A Afonso-Rodríguez e Javier Giner. Estimating GARCH models using support vector machines. *Quantitative Finance*, 3:1–10. Citado na pág. i, ii, 2, 46, 47, 52

Ferreira (2011) Tadeu Augusto Ferreira. *Previsão da volatilidade de séries financeiras via máquina de suporte vetorial*. Dissertação de mestrado, Universidade de São Paulo. Citado na pág. 46

Franses e van Dijk (2000) Philip Hans Franses e Dick van Dijk. *Nonlinear Time Series Models in Empirical Finance*. Cambridge University Press. Citado na pág. 9

Gavrishchaka e Banerjee (2006) Valeriy V. Gavrishchaka e Supriya Banerjee. Support vector machine as an efficient framework for stock market volatility forecasting. *Computational Management Science*, 3(2):147–160. doi: 10.1007/s10287-005-0005-5. Citado na pág. 46, 52

Gavrishchaka e Ganguli (2003) Valeriy V. Gavrishchaka e Supriya B. Ganguli. Volatility forecasting from multiscale and high-dimensional market data. *Neurocomputing*, 55(1-2): 285–305. doi: 10.1016/S0925-2312(03)00381-3. Citado na pág. 46, 52

Geng (2012) Li-Yan Geng. Forecast of Stock Index Volatility Using Wavelet Support Vector Machines. *Advanced Management Science*, páginas 19–22. Citado na pág. 49, 52

Geng e Liang (2011) Li Yan Geng e Yi Gang Liang. Prediction on Fund Volatility Based on SVRGM-GARCH Model. *Advanced Materials Research*, 403-408:3763–3768. doi: 10.4028/www.scientific.net/AMR.403-408.3763. Citado na pág. 49, 52

Geng e Yu (2013) Li-Yan Geng e Fei Yu. Forecasting Stock Volatility using LSSVR-based GARCH Model Optimized by Siwpso Algorithm. *Journal of Applied Sciences*, 13(22): 5132–5137. Citado na pág. 50, 52

Genton (2001) Marc G Genton. Classes of Kernels for Machine Learning: A Statistics Perspective. *Journal of Machine Learning Research*, 2:299–312. doi: 10.1162/15324430260185646. Citado na pág. 35

George e Rajeev (2008) Jose George e K. Rajeev. Hybrid wavelet support vector regression. Em *2008 7th IEEE International Conference on Cybernetic Intelligent Systems*, number 4. doi: 10.1109/UKRICIS.2008.4798920. Citado na pág. 36

Ghalanos (2015) Alexios Ghalanos. rugarch: Univariate GARCH models, 2015. URL https://cran.r-project.org/web/packages/rugarch/index.html. Citado na pág. 57

Glosten *et al.* **(1993)** Lawrence R. Glosten, Ravi Jagannthan e David E. Runkle. On the Relation between the Expected Value and the Volatility of the Nominal Excess Return on Stocks. *The Journal of Finance*, 48(5):1779–1801. doi: 10.1111/j.1540-6261.1993.tb05128.x. Citado na pág. 17

Goupillaud *et al.* **(1984)** P. Goupillaud, A. Grossmann e J. Morlet. Cycle-octave and related transforms in seismic signal analysis. *Geoexploration*, 23(1):85–102. ISSN 00167142. doi: 10.1016/0016-7142(84)90025-5. Citado na pág. 37

Guidolin (2011) Massimo Guidolin. Markov Switching Models in Empirical Finance. Em *Missing Data Methods: Time-Series Methods and Applications (Advances in Econometrics, Volume 27 Part 2)*, páginas 1–86. Emerald Group Publishing Limited. doi: 10.1108/S0731-9053(2011)000027B004. Citado na pág. 3, 21, 22, 63

Haas *et al.* **(2004)** M. Haas, S. Mittnik e M. S. Paolella. Mixed Normal Conditional Heteroskedasticity. *Journal of Financial Econometrics*, 2(2):211–250. doi: 10.1093/jjfinec/nbh009. Citado na pág. 22, 63

Hansen e Lunde (2005) Peter R. Hansen e Asger Lunde. A forecast comparison of volatility models: Does anything beat a GARCH(1,1)? *Journal of Applied Econometrics*, 20(February):873–889. doi: 10.1002/jae.800. Citado na pág. 4, 15

Hastie *et al.* **(2009)** Trevor Hastie, Robert Tibshirani e Jerome Friedman. *The Elements of Statistical Learning*. doi: 10.1007/b94608. Citado na pág. 5, 39

Haykin (1999) Simon Haykin. *Neural Networks-A Comprehensive Foundation*. Second ed. Citado na pág. 44, 47

Heaton *et al.* **(2016)** J. B. Heaton, N. G. Polson e J. H. Witte. Deep Learning in Finance. páginas 1–20. URL http://arxiv.org/abs/1602.06561. Citado na pág. 64

Herbrich (2001) Ralf Herbrich. *Learning Kernel Classifiers: Theory and Algorithms*. MIT Press. ISBN 026208306X. Citado na pág. 24

Hoeffding (1963) Wassily Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30. doi: 10.1080/01621459.1963.10500830. Citado na pág. 28

Hossain e Mohammed (2011) Altaf Hossain e Nasser Mohammed. Recurrent Support and Relevance Vector Machines Based Model with Application to Forecasting Volatility of Financial Returns. *Journal of Intelligent Learning Systems and Applications*, 3 (November):230–241. doi: 10.4236/jilsa.2011.34026. Citado na pág. 49, 52

Huang *et al.* **(2014)** Chao Huang, Fei Gao e Hongyan Jiang. Combination of Biorthogonal Wavelet Hybrid Kernel OCSVM with Feature Weighted Approach Based on EVA and GRA in Financial Distress Prediction. *Mathematical Problems in Engineering*, 2014. doi: http://dx.doi.org/10.1155/2014/538594. Citado na pág. 36, 63

Huerta *et al.* **(2013)** Ramon Huerta, Fernando Corbacho e Charles Elkan. Nonlinear support vector machines can systematically identify stocks with high and low future returns. *Algorithmic Finance*, 2:45–58. doi: 10.3233/AF-13016. Citado na pág. 46

Hwang e Shin (2010) Chang-Ha ; Hwang e Sa-Im ; Shin. Estimating GARCH models using kernel machine learning. *Journal of the Korean Data and Information Science Society*, 21 (3):419–425. Citado na pág. 48, 52

Jorion (1995) Philippe Jorion. Predicting Volatility in the Foreign Exchange Market. *The Journal of Finance*, 50(2):507–528. Citado na pág. 1

Karatzoglou *et al.* **(2004)** Alexandros Karatzoglou, Alex Smola, Kurt Hornik e Achim Zeileis. kernlab – An {S4} Package for Kernel Methods in {R}. *Journal of Statistical Software*, 11(9):1–20. Citado na pág. 58

Karush (1939) William Karush. *Minima of Functions of Several Variables with Inequalities as Side Constraints*. Tese de Doutorado, University of Chicago. Citado na pág. 41, 45

Khan (2011a) Ashraful Islam Khan. Modelling daily value-at-risk using realized volatility , non-linear support vector machine and ARCH type models. *Jounal of Economics and International Finance*, 3(May):305–321. Citado na pág. 49, 52

Khan (2011b) Md. Ashraful Islam Khan. Financial Volatility Forecasting by Nonlinear Support Vector Machine Heterogeneous Autoregressive Model: Evidence from Nikkei 225 Stock Index. *International Journal of Economics and Finance*, 3(4):138–150. doi: 10.5539/ijef.v3n4p138. Citado na pág. 49, 52

Kisinbay (2010) Turgut Kisinbay. The use of encompassing tests for forecast combinations. *Journal of Forecasting*, 29(8):715–727. doi: 10.1002/for.1170. Citado na pág. 7

Kohavi (1995) Ron Kohavi. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. Em *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, volume 14, páginas 1137–1143, Monreal. Morgan Kaufmann Publishers Inc. doi: 10.1067/mod.2000.109031. Citado na pág. 5

Kuhn e Tucker (1951) H. W. Kuhn e A.W Tucker. *Nonlinear Programming*. University of California Press. ISBN 1886529000. doi: 10.1007/BF01582292. Citado na pág. 41, 45

Längkvist *et al.* **(2014)** Martin Längkvist, Lars Karlsson e Amy Loutfi. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 42(1):11–24. ISSN 01678655. doi: 10.1016/j.patrec.2014.01.008. Citado na pág. 64

Levy e Kaplanski (2015) Moshe Levy e Guy Kaplanski. Portfolio selection in a two-regime world. *European Journal of Operational Research*, 242(2):514–524. doi: 10.1016/j.ejor.2014.10.012. Citado na pág. i, ii, v, 3, 21, 22

Li e Sun (2010) Jinbo Li e Shiliang Sun. Nonlinear combination of multiple kernels for support vector machines. Em *Pattern Recognition (ICPR), 2010 20th International Conference on*, páginas 2889–2892, Istanbul. IEEE. doi: 10.1109/ICPR.2010.708. Citado na pág. 36

Li (2014) Yushu Li. Estimating and Forecasting APARCH-Skew- t Model by Wavelet Support Vector Machines. *Journal of Forecasting*, 269(March):259–269. doi: 10.1002/for.2275. Citado na pág. i, ii, 2, 37, 50, 51, 52, 59

Li-yan *et al.* **(2013)** Geng Li-yan, Yu Fei e Zhou Xiao-ping. Grey Least Squares Support Vector Machines with Particle Swarm Optimization for Volatility Forecasting. *Advances in information Sciences and Service Sciences*, 5(8):580–588. doi: 10.4156/AISS.vol5.issue8.70. Citado na pág. 50, 52

Liyan e Zhanfu (2012) Geng Liyan e Zhang Zhanfu. CARRX Model Based on LSSVR Optimized by Adaptive PSO. Em *2012 Third International Conference on Digital Manufacturing & Automation*, páginas 268–271. ISBN 9780769547725. doi: 10.1109/ICDMA.2012.65. Citado na pág. 51, 52

Lu *et al.* **(2009a)** Chi-Jie Lu, Tian-Shyug Lee e Chih-Chou Chiu. Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems*, 47(2):115–125. ISSN 01679236. doi: 10.1016/j.dss.2009.02.001. URL http://linkinghub.elsevier.com/retrieve/pii/S0167923609000323. Citado na pág. 59

Lu *et al.* **(2009b)** Yan-Ling Lu Yan-Ling Lu, Lei Li Lei Li, Meng-Meng Zhou Meng-Meng Zhou e Guo-Liang Tian Guo-Liang Tian. A new fuzzy support vector machine based on mixed kernel function. Em *International Conference on Machine Learning and Cybernetics*, volume 1, páginas 526–531. IEEE. doi: 10.1109/ICMLC.2009.5212552. Citado na pág. 36

Luxburg e Schölkopf (2008) U. V. Luxburg e B. Schölkopf. Statistical Learning Theory : Models , Concepts , and Results. *ArXiv e-prints*, páginas 1–40. Citado na pág. 24, 25, 26, 27, 28, 29, 30, 31, 32, 33

Mangasarian (1994) Olvi L. Mangasarian. *Nonlinear Programming*. Society for Industrial and Applied Mathematics. ISBN 978-0898713411. doi: http://dx.doi.org/10.1137/1.9781611971255. Citado na pág. 44

Marcucci (2005) Juri Marcucci. Forecasting Stock Market Volatility with Regime-Switching GARCH models. *Studies in Nonlinear Dynamics & Econometrics*, 9(4). doi: 10.2202/1558-3708.1145. Citado na pág. 15, 22

Marron e Wand (1992) J. S Marron e M.P. Wand. Exact Mean Integrated Squared Error. *Annals of Statistics*, 20(2):712–736. Citado na pág. 19

McLachlan e Peel (2000) Geoffrey McLachlan e David Peel. *Finite Mixture Models*, volume 44. doi: 10.1198/tech.2002.s651. Citado na pág. 19

Mercer (1909) James Mercer. Functions of Positive and Negative Type and their connection with the Theory of Integral Equations. *Philosophical Transactions of the Royal Society of London*, 209(A):415–446. Citado na pág. 5, 35, 37

Mitchell (1997) Tom Mitchell. *Machine Learning*. McGraw Hill. Citado na pág. 23

Mohri *et al.* **(2012)** Mehryar Mohri, Afshin Rostamizadeh e Ameet Talwalkar. *Fundations of Machine Learning*. MIT Press. Citado na pág. v, 40, 41

Morettin (2011) Pedro A. Morettin. *Econometria Financeira: um curso em séries temporais financeiras*. Editora Edgard Blücher. ISBN 978-85-212-0597-5. Citado na pág. 15, 16

Morettin e Toloi (2006) Pedro A. Morettin e Clélia M. C. Toloi. *Análise de Séries Temporais*. Citado na pág. 15

Nason (2008) G.P. Nason. *Wavelet Methods in Statistics with R*. Springer Science+Business Media. doi: 10.1007/978-0-387-75961-6e-ISBN:. Citado na pág. 36

Nelson (1991) Daniel B Nelson. Conditional heteroskedasticity in asset returns: a new approach. *Econometrica*, 59(2):347–370. doi: 10.2307/2938260. Citado na pág. 16

Ning *et al.* (2015) Cathy Ning, Dinghai Xu e Tony S Wirjanto. Is volatility clustering of asset returns asymmetric ? *Journal of Banking and Finance*, 52:62–76. doi: 10.1016/j.jbankfin.2014.11.016. Citado na pág. 14

Ou e Wang (2010a) Phichhang Ou e Hengshan Wang. Predict GARCH Based Volatility of Shanghai Composite Index by Recurrent Relevant Vector Machines and Recurrent Least Square Support Vector Machines. *Journal of Mathematics Research*, 2(2):11–19. Citado na pág. 47, 52

Ou e Wang (2013) Phichhang Ou e Hengshan Wang. Volatility Modelling and Prediction by Hybrid Support Vector Regression with Chaotic Genetic Algorithms. *The International Arab Journal of Information Technology*, 11(3):287–292. Citado na pág. 50, 52

Ou e Wang (2010b) Phichhang Ou e Hengshan Wang. Financial Volatility Forecasting by Least Square Support Vector Machine Based on GARCH , EGARCH and GJR Models : Evidence from ASEAN Stock Markets. *International Journal of Economics and Finance*, 2(2):51–64. Citado na pág. 47, 52

Patton (2011) Andrew J. Patton. Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics*, 160(1):246–256. ISSN 03044076. doi: 10.1016/j.jeconom.2010.03.034. Citado na pág. 6

Poon, Huang. Clive (2003) Granger Poon, Huang. Clive. Forecasting Volatility in Financial Markets : A Review. *Journal of Economic Literature*, XLI(June):478–539. Citado na pág. 1, 4, 9

Ruping e Morik (2003) S. Ruping e K. Morik. Support vector machines and learning about time. Em *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003.*, volume 4, páginas 864–867. IEEE. doi: 10.1109/ICASSP.2003.1202780. Citado na pág. 46

Sangeetha e Kalpana (2010) R. Sangeetha e B. Kalpana. A Comparative Study and Choice of an Appropriate Kernel for Support Vector Machines. Em *Information and Communication Technologies*, páginas 549–553. doi: 10.1007/978-3-642-15766-0_93. Citado na pág. 35

Sankar *et al.* (2009) Ravi Sankar, South Florida, Nicholas I. Sapankevych e Ravi Sankar. Time Series Prediction using Support Vector Machines: A Survey. *Computational Intelligence Magazine*, (May):24–38. Citado na pág. 39, 46, 63

Santamaría-Bonfil *et al.* (2015) Guillermo Santamaría-Bonfil, Juan Frausto-Solís e Ignacio Vázquez-Rodarte. Volatility Forecasting Using Support Vector Regression and a Hybrid Genetic Algorithm. *Computational Economics*, 45:111–133. doi: 10.1007/s10614-013-9411-x. Citado na pág. i, ii, 2, 51, 52, 59

Schölkopf e Smola (2002) B. Schölkopf e A. J. Smola. *Learning with kernels- Support Vector Machines, Regularization, Optimization adn Beyond*, volume 1. The MIT Press, first ed. doi: 10.1198/jasa.2003.s269. Citado na pág. 24, 35

Seethalakshmi *et al.* (2014) R Seethalakshmi, V. Saavithri, C. Vijayabanu e V Badrinath. PCA based Support Vector Machine technique for volatility forecasting. *International Journal of Research in Engineering and Technology*, 3(8):389–395. Citado na pág. 51, 52

Sewell (2008) Martin Sewell. Structural risk minimization. 2008. Citado na pág. 32

Sewell (2011) Martin Sewell. Characterization of Financial Time Series. Relatório técnico, University College of London. Citado na pág. 10

Shalev-shwartz e Ben-david (2014) Shai Shalev-shwartz e Shai Ben-david. *Understanding Machine Learning: From Theory to Algorithms*. ISBN 9781107057135. doi: 10.1017/CBO9781107298019. Citado na pág. 5, 34

Shim e Lee (2010) Joo-Yong ; Shim e Jang-Taek Lee. Estimation of nonlinear GARCH-M model. *Journal of the Korean Data and Information Science Society*, 21(5):831–839. Citado na pág. 49, 52

Smits e Jordaan (2002) G.F. Smits e E.M. Jordaan. Improved SVM regression using mixtures of kernels. Em *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02*, volume 3, páginas 2785–2790. IEEE. ISBN 0-7803-7278-6. doi: 10.1109/IJCNN.2002.1007589. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1007589. Citado na pág. 36

Smola e Schölkopf (2004) A .J. Smola e B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14:199–222. doi: 10.1023/B:STCO.0000035301.49549.88. Citado na pág. 36, 43, 45

Song *et al.* (2014) Xin-Ping Song, Zhi-Hua Hu, Jian-Guo Du e Zhao-Han Sheng. Application of Machine Learning Methods to Risk Assessment of Financial Statement Fraud: Evidence from China. *Journal of Forecasting*, 33(8):611–626. doi: 10.1002/for.2294. Citado na pág. 39

Steinwart (2005) Ingo Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *IEEE Transactions on Information Theory*, 51(1):128–142. doi: 10.1109/TIT.2004.839514. Citado na pág. 32

Steinwart e Christmann (2008) Ingo Steinwart e Andreas Christmann. *Support Vector Machines*. Springer Science+Business Media. doi: 10.1007/978-0-387-77242-4. Citado na pág. 24, 34

Suykens (1999) J.a.K. Suykens. Least Squares SVM Classifiers, 1999. Citado na pág. 47

Taleb (2010) Nassim Nicholas Taleb. *The Black Swan: The Impact of the Highly Improbable*. Random House, 2ª ed. ISBN 9780375427534. Citado na pág. 53

Taleb (2016) Nassim Nicholas Taleb. The Meta-Distribution of Standard P-Values. páginas 1–4. URL http://arxiv.org/abs/1603.07532. Citado na pág. 61

Tang *et al.* (2009a) Ling-Bing Tang, Huan-Ye Sheng e Ling-Xiao Tang. GARCH prediction using spline wavelet support vector machine. *Neural Computing and Applications*, 18(8):913–917. doi: 10.1007/s00521-009-0241-7. Citado na pág. 37, 48, 52

Tang *et al.* (2009b) Ling-Bing Tang, Ling-Xiao Tang e Huan-Ye Sheng. Forecasting volatility based on wavelet support vector machine. *Expert Systems with Applications*, 36(2):2901–2909. doi: 10.1016/j.eswa.2008.01.047. Citado na pág. i, ii, 37, 48, 51, 52

Tipping (2001) Michael Tipping. Sparse Bayesian Learning and the Relevance Vector Mach. *Journal of Machine Learning Research*, 1:211–244. doi: 10.1162/15324430152748236. Citado na pág. 47

Tsay (2010) Ruey S Tsay. *Analysis of Financial Time Series*, volume 48. John Wiley & Sons, Inc., third ed. doi: 10.1198/tech.2006.s405. Citado na pág. 11, 12, 13, 14

Vapnik (1992) V Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, páginas 831–838. Citado na pág. 27, 28

Vapnik (1995) V N Vapnik. *The Nature of statistical Learning Theory*. Springer Science+Business Media. ISBN 9781475724424. Citado na pág. 25, 26, 39, 42, 43, 44, 45

Vapnik (1999) V N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999. doi: 10.1109/72.788640. Citado na pág. 24, 32

Vapnik (1982) Vladimir N. Vapnik. *Estimation of Dependences Based on Empirical Data*, volume 4. Springer-Verlag New York, Inc. Citado na pág. 2, 31, 39

Vapnik (1998) Vladimir N Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc., 1ª edição ed. ISBN 0471030031. Citado na pág. 23, 39

Vapnik (2006) Vladimir N. Vapnik. *Estimation of Dependences Based on Empirical Data*, volume 4. Springer-Verlag New York. doi: 10.1007/0-387-34239-7. Citado na pág. 25, 29

Varian (2014) Hal R Varian. Big Data : New Tricks for Econometrics. *Journal of Economic Perspectives*, 28(2):3–28. doi: 10.1257/jep.28.2.3. Citado na pág. 39

Wang *et al.* (2011) Baohua Wang, Hejiao Huang e Xiaolong Wang. A support vector machine based MSM model for financial short-term volatility forecasting. *Neural Computing and Applications*, 22(1):21–28. ISSN 0941-0643. doi: 10.1007/s00521-011-0742-z. Citado na pág. 49, 52

Wang e Taaffe (2015) Jin Wang e Michael R. Taaffe. Multivariate Mixtures of Normal Distributions: Properties, Random Vector Generation, Fitting, and as Models of Market Daily Changes. *INFORMS Journal on Computing*, 27(2):193–203. doi: 10.1287/ijoc.2014.0616. Citado na pág. 3, 19, 20

Wasserstein e Lazar (2016) Ronald L. Wasserstein e Nicole A. Lazar. The ASA's statement on p-values: context, process, and purpose. *The American Statistician*, páginas 1–17. doi: 10.1080/00031305.2016.1154108. Citado na pág. 61

Wilson *et al.* (2015) Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov e Eric P. Xing. Deep Kernel Learning. Em *Artificial Intelligence and Statistics (AISTATS)*, páginas 1–19. URL http://arxiv.org/abs/1511.02222. Citado na pág. 42, 64

Wirjanto e Xu (2009) Tony S Wirjanto e Dinghai Xu. The Applications of Mixtures of Normal Distributions in Empirical Finance : A Selected Survey. 2009. Citado na pág. 21, 22

Wong e Li (2001) Chun Shan Wong e Wai Keung Li. On a Mixture Autoregressive Conditional Heteroscedastic Model. *Journal of the American Statistical Association*, 96 (455):982–995. doi: 10.1198/016214501753208645. Citado na pág. 22

Xia *et al.* **(2005)** Xiao-Lei Xia, Michael R Lyu, Tat-Ming Lok e Guang-Bin Huang. Methods of Decreasing the Number of Support Vectors via k-Mean Clustering. Em *Lecture Notes in Computer Science*, volume 3644, páginas 717–726. doi: 10.1007/11538059_75. Citado na pág. 60

Xu *et al.* **(2011)** Jingfeng Xu, Jian Liu e Haijian Zhao. Financial Forecasting : Comparative Performance of Volatility Models in Chinese. Em *Fourth International Joint Conference on Computational Sciences and Optimization*. doi: 10.1109/CSO.2011.136. Citado na pág. 47, 52

Zakoian (1994) Jean Michel Zakoian. Threshold heteroskedastic models. *Journal of Economic Dynamics and Control*, 18(5):931–955. ISSN 01651889. doi: 10.1016/0165-1889(94)90039-6. Citado na pág. 17

Zhang *et al.* **(2004)** Li Zhang, Weida Zhou e Licheng Jiao. Wavelet support vector machine. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 34(1):34–39. doi: 10.1109/TSMCB.2003.811113. Citado na pág. 36, 37

Zimmermann (2015) Tom Zimmermann. *Inductive Learning and Theory Testing : Applications in Finance*. Tese de Doutorado, Harvard University. URL http://nrs.harvard.edu/urn-3:HUL.InstRepos:17467320. Citado na pág. 39