



Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Estatística

Yuri Medeiros de Andrade

Orientador: Prof. Dr. Raul Yukihiro Matsushita

A máquina de suporte vetorial aplicada
em análise de séries temporais

Brasília
2017

Yuri Medeiros de Andrade

**A máquina de suporte vetorial aplicada
em análise de séries temporais**

Dissertação submetida ao Departamento de Estatística do Instituto de Ciências Exatas da Universidade de Brasília como requisito parcial à obtenção do título de Mestre em Estatística

Orientador: Prof. Dr. Raul Yukihiro Matsushita

**Brasília
2017**

Yuri Medeiros de Andrade

**A máquina de suporte vetorial aplicada
em análise de séries temporais**

Dissertação submetida ao Departamento de Estatística do Instituto de Ciências Exatas da Universidade de Brasília como requisito parcial à obtenção do título de Mestre em Estatística.

Aprovada em: 30/6/2017

Banca examinadora

Prof. Dr. Raul Yukihiro Matsushita

Universidade de Brasília - UnB

Orientador

Prof. Dr. Pushpa Narayan Rathie

Universidade Federal do Ceará - UFC

Membro da banca examinadora

Prof. Dr. Peter Zörnig

Universidade de Brasília - UnB

Membro da banca examinadora

Agradecimentos

A Deus.

Aos meus pais, por desde sempre terem me ajudado e apoiado.

Ao meu orientador Raul Yukihiro Matsushita, sem o qual este trabalho não seria possível.

Aos demais professores e funcionários que conheci na UnB, pelos ensinamentos, ajuda e incentivos.

Aos meus amigos, por terem feito deste período muito mais agradável.

*“Don’t be afraid to give up the good
to go for the great.”*

(John Davison Rockefeller)

Resumo

Os modelos de aprendizado de máquina vêm tomando espaço na literatura moderna. Diversos artigos os comparam a métodos de series temporais, pois devido à sua maleabilidade e adaptabilidade, conseguem facilmente se ajustar a diversos tipos de dados, sejam provenientes de modelos lineares ou não (Tay & Cao, 2001; Zhao *et al.*, 2012). Este trabalho visou comparar os modelos tradicionais ARMA e GARCH à máquina de suporte vetorial para previsões um passo à frente na análise da volatilidade de séries temporais financeiras. Aqui foi analisado o desempenho de um híbrido AR-GARCH, aplicando diversas distribuições de probabilidade ao seu termo de ruído $\{e_t\}$, em relação ao desempenho de um novo método proposto por Chen *et al.* (2010) chamado de SVR recorrente. Modificamos o critério de parada desse novo e pouco conhecido método, para que se ajustasse melhor aos critérios estatísticos considerados por Morettin & Toloï (2006), no que diz respeito à estrutura de autocorrelação serial. Por fim, o aplicamos a dados gerados pelo modelo AR-GARCH e a dois bancos de dados reais, iguais aos utilizados por Chen *et al.* (2010), para compararmos o desempenho do SVR recorrente, com o novo critério de parada, aos modelos GARCH, ARMA e AR-GARCH.

Palavras-chave: SVM. Máquina de suporte vetorial. SVR recorrente. GARCH. AR-GARCH.

Abstract

Machine learning models have been taking space in modern literature. Several articles compare them to time series methods because, due to their malleability and adaptability, they can be easily adjusted to different types of data, whether linear or non-linear (Tay & Cao, 2001; Zhao *et al.*, 2012). This work aimed to compare the traditional ARMA and GARCH models to the support vector machine for predictions one step ahead in the analysis of the volatility of financial time series. Here we analyzed the performance of an AR-GARCH hybrid applying several probability distributions to its noise term $\{e_t\}$, in relation to the performance of a new method proposed by Chen *et al.* (2010) called recurrent SVR. We modified the stopping criterion of this new and little-known method to better fit the statistical criteria considered by Morettin & Toloï (2006) regarding the serial autocorrelation structure. Finally, we applied it to data generated by the AR-GARCH model and to two real databases, the same as those used by Chen *et al.* (2010), to compare the performance of the recurrent SVR with the new criterion of GARCH, ARMA and AR-GARCH models.

Keywords: SVM. Support vector machine. Recurrent SVR. GARCH. AR-GARCH.

Lista de Figuras

2.1	Na região à direita do hiperplano, em que $f(x) > 0$, temos os pontos do tipo +1 representados por triângulos vazios, enquanto que na região à esquerda do hiperplano, em que $f(x) < 0$, temos os pontos do tipo -1 representados por triângulos cheios.	28
2.2	Em (a), tem-se os dois vetores a e b , com um ângulo θ formado entre eles, e um hiperplano. Em (b), a projeção P_{ab} do vetor a sobre o vetor b está explicitada.	29
2.3	A figura mostra alguns dos infinitos hiperplanos que dividem o conjunto de dados.	30
2.4	Os pontos que estão nas envoltórias convexas, ou <i>convex hulls</i> (Barber <i>et al.</i> , 1996; Yao, 1981; Preparata & Hong, 1977; Brown, 1979; Chan, 1996; Crisp & Burges, 2000), são chamados de suportes vetoriais. A distância entre essas envoltórias é chamada de margem.	31
2.5	Máquinas de suporte vetorial: à esquerda temos o resultado proporcionado por um <i>kernel</i> polinomial de grau 3, e à direita, para o mesmo conjunto de dados, observa-se o resultado da aplicação de um <i>kernel</i> radial (James <i>et al.</i> , 2013).	38
2.6	Função de perda ε -insensível na linha mais espessa em vermelho. . . .	40
2.7	Curvas ajustadas por SVR aos dados (pontos pretos) da Tabela 2.1 utilizando o <i>kernel</i> linear no gráfico à esquerda, e o radial à direita. .	46
2.8	Evolução das últimas 50 observações da série $\{X_t\}$ (bolas e traços pretos) e suas previsões por meio do modelo AR(2) (em linha tracejada azul), e do ajuste obtido pelo SVR de <i>kernel</i> linear com $C = \varepsilon = 1$ (em linha pontilhada vermelha).	50

3.1	O esquema (a) representa o SVR tradicional, e o esquema (b) representa o SVR recorrente.	56
3.2	Esquema de obtenção passo a passo de $\{\hat{u}_t\}$	58
3.3	Os gráficos mostram 20 resultados dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) dos modelos GARCH e SVR recorrente, ambos supondo $\{e_t\}$ se comportando como uma normal padrão.	62
3.4	À esquerda, os gráficos mostram os EMA (gráfico de cima) e os EQM (gráfico de baixo) confrontando os resultados dos modelos GARCH e SVR recorrente. À direita, os gráficos mostram o logaritmo dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) também confrontando os resultados dos modelos GARCH e SVR recorrente referentes ao $\{e_t\}$ se comportando como uma normal padrão.	63
4.1	Evolução no tempo das séries temporais dos preços (à esquerda) e retornos (à direita) diários da taxa de câmbio Libra/Dólar.	66
4.2	60 últimas observações da série temporal de $\{u_t^2\}$ da taxa de câmbio Libra/Dólar (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma normal padrão (em linha vermelha com quadrados).	67
4.3	Evolução no tempo das séries temporais dos preços (à esquerda) e retornos (à direita) diários do índice NYSE.	67
4.4	60 últimas observações da série temporal de $\{u_t^2\}$ do índice NYSE (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma normal padrão (em linha vermelha com quadrados).	68
A.1	Os gráficos mostram 20 resultados dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) dos modelos GARCH e SVR recorrente, ambos supondo $\{e_t\}$ se comportando como uma normal assimétrica.	77

A.2	À esquerda, os gráficos mostram os EMA (gráfico de cima) e os EQM (gráfico de baixo) confrontando os resultados dos modelos GARCH e SVR recorrente. À direita, os gráficos mostram o logaritmo dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) também confrontando os resultados dos modelos GARCH e SVR recorrente referentes ao $\{e_t\}$ se comportando como uma normal padrão.	78
A.3	Os gráficos mostram 20 resultados dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) dos modelos GARCH e SVR recorrente, ambos supondo $\{e_t\}$ se comportando como uma t-Student padrão.	78
A.4	À esquerda, os gráficos mostram os EMA (gráfico de cima) e os EQM (gráfico de baixo) confrontando os resultados dos modelos GARCH e SVR recorrente. À direita, os gráficos mostram o logaritmo dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) também confrontando os resultados dos modelos GARCH e SVR recorrente referentes ao $\{e_t\}$ se comportando como uma t-Student padrão.	79
A.5	Os gráficos mostram 20 resultados dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) dos modelos GARCH e SVR recorrente, ambos supondo $\{e_t\}$ se comportando como uma t-Student assimétrica.	79
A.6	À esquerda, os gráficos mostram os EMA (gráfico de cima) e os EQM (gráfico de baixo) confrontando os resultados dos modelos GARCH e SVR recorrente. À direita, os gráficos mostram o logaritmo dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) também confrontando os resultados dos modelos GARCH e SVR recorrente referentes ao $\{e_t\}$ se comportando como uma t-Student assimétrica.	80
A.7	Os gráficos mostram 20 resultados dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) dos modelos GARCH e SVR recorrente, ambos supondo $\{e_t\}$ se comportando como uma GED.	80

A.8	À esquerda, os gráficos mostram os EMA (gráfico de cima) e os EQM (gráfico de baixo) confrontando os resultados dos modelos GARCH e SVR recorrente. À direita, os gráficos mostram o logaritmo dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) também confrontando os resultados dos modelos GARCH e SVR recorrente referentes ao $\{e_t\}$ se comportando como uma GED.	81
A.9	Os gráficos mostram 20 resultados dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) dos modelos GARCH e SVR recorrente, ambos supondo $\{e_t\}$ se comportando como uma GED assimétrica.	81
A.10	À esquerda, os gráficos mostram os EMA (gráfico de cima) e os EQM (gráfico de baixo) confrontando os resultados dos modelos GARCH e SVR recorrente. À direita, os gráficos mostram o logaritmo dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) também confrontando os resultados dos modelos GARCH e SVR recorrente referentes ao $\{e_t\}$ se comportando como uma GED assimétrica.	82
C.1	60 últimas observações da série temporal de $\{u_t^2\}$ da taxa de câmbio Libra/Dólar (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma normal assimétrica (em linha vermelha com quadrados).	90
C.2	60 últimas observações da série temporal de $\{u_t^2\}$ da taxa de câmbio Libra/Dólar (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma t-Student (em linha vermelha com quadrados).	91
C.3	60 últimas observações da série temporal de $\{u_t^2\}$ da taxa de câmbio Libra/Dólar (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma t-Student assimétrica (em linha vermelha com quadrados).	91

C.4	60 últimas observações da série temporal de $\{u_t^2\}$ da taxa de câmbio Libra/Dólar (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma GED (em linha vermelha com quadrados).	92
C.5	60 últimas observações da série temporal de $\{u_t^2\}$ da taxa de câmbio Libra/Dólar (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma GED padrão (em linha vermelha com quadrados).	92
D.1	60 últimas observações da série temporal de $\{u_t^2\}$ do índice NYSE (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma normal assimétrica (em linha vermelha com quadrados).	93
D.2	60 últimas observações da série temporal de $\{u_t^2\}$ do índice NYSE (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma t-Student (em linha vermelha com quadrados).	94
D.3	60 últimas observações da série temporal de $\{u_t^2\}$ do índice NYSE (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma t-Student assimétrica (em linha vermelha com quadrados).	94
D.4	60 últimas observações da série temporal de $\{u_t^2\}$ do índice NYSE (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma GED (em linha vermelha com quadrados).	95

D.5 60 últimas observações da série temporal de $\{u_t^2\}$ do índice NYSE (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma GED padrão (em linha vermelha com quadrados). 95

Lista de Tabelas

2.1	Dados utilizados no exemplo computacional da Subseção 2.5.1.	43
2.2	Valores de C e ε selecionados para cada <i>kernel</i> , e resultados do EQM e EMA dos modelos de SVR com <i>kernel</i> linear e polinomial de grau 3 ajustados aos dados da Tabela 2.1.	45
2.3	Valores de C e ε selecionados para cada <i>kernel</i> , e resultados do EQM e EMA dos modelos de SVR com <i>kernel</i> linear e polinomial de grau 3 ajustados aos dados provenientes do processo AR(2) definido na Equação 2.46.	48
3.1	Valores dos EMAs e EQMs referentes aos ajustes dos modelos de SVR recorrente e GARCH. A série temporal $\{e_t\}$ é uma sequência <i>iid</i> de ruídos e se comporta de acordo com as seis distribuições de probabilidade comentadas neste capítulo. Nesta tabela observam-se os resultados referentes à série temporal teórica gerada pelo processo AR-GARCH descrito nas Equações 3.16 e 3.17. Cada estimativa se refere ao valor médio das simulações obtidas por Monte Carlo e está acompanhada de seu erro-padrão em parênteres.	62
4.1	Valores dos EMAs e EQMs referentes aos ajustes dos modelos de SVR recorrente e GARCH. A sequência de erros $\{e_t\}$ é <i>iid</i> e se comporta de acordo com os 6 modelos GARCH testados. Nesta tabela observam-se os resultados referentes à série temporal da taxa de câmbio Libra/Dólar.	66

4.2	Valores dos EMAs e EQMs referentes aos ajustes dos modelos de SVR recorrente e GARCH. A sequência de erros $\{e_t\}$ é <i>iid</i> e se comporta de acordo com os 6 modelos GARCH testados. Nesta tabela observam-se os resultados referentes à série temporal do índice NYSE.	68
-----	---	----

Sumário

1	Introdução	17
1.1	Apresentação	17
1.2	Objetivos	19
1.3	Uma breve revisão de literatura	19
1.4	Estrutura da dissertação	24
2	A máquina de suporte vetorial	26
2.1	Para fins de classificação binária: caso linearmente separável	27
2.2	Para fins de classificação binária: caso não separável linearmente	32
2.3	Para fins de classificação binária: caso não linear	36
2.4	Para fins de regressão	38
2.5	Implementação computacional no <i>software</i> R	43
2.5.1	Exemplo: aplicação em regressão não-linear	43
2.5.2	Exemplo: aplicação em séries temporais autorregressivas	47
3	Previsão de volatilidade via SVR recorrente	51
3.1	Modelo GARCH	52
3.2	SVR recorrente	55
3.3	Exemplo	58
4	Aplicação em dados reais	64
4.1	Resultados para a taxa de câmbio Libra/Dólar	65
4.2	Resultados para o índice NYSE	67
5	Conclusões	69

Referências Bibliográficas	71
Apêndices	77
A	77
A.1 Gráficos da Seção 3.3	77
B	83
B.1 Código da Seção 3.3	83
C	90
C.1 Gráficos da Seção 4.1	90
D	93
D.1 Gráficos da Seção 4.2	93
E	96
E.1 Código das Seções 4.1 e 4.2	96

Capítulo 1

Introdução

1.1 Apresentação

Denomina-se série temporal um conjunto de observações ordenadas no tempo (Morettin & Toloi, 2006). As séries temporais representam uma forma particular de dados em que as observações passadas exercem um papel importante para a descrição de observações futuras. A relação entre o passado e o futuro depende tanto dos aspectos probabilísticos (como as tendências e os padrões cíclicos), quanto dos probabilísticos (quando a série temporal for representada por um processo estocástico). A formulação de previsões acerca do futuro constitui uma das principais finalidades em análise de séries temporais. Dentre as possíveis abordagens para esse fim, destacam-se as técnicas estatísticas e as de inteligência computacional.

Na abordagem probabilística, em geral, um modelo de séries temporais assume determinada forma dependendo do tipo de processo estocástico para o qual foi desenhado. Por exemplo, uma série gerada por filtro linear estacionário, em essência, é aquela que foi produzida por uma soma de erros aleatórios (Wei, 2006). Para essa situação, os modelos lineares da classe ARMA (autorregressivos e de médias móveis) são candidatos naturais que proporcionam boas previsões, em que cada previsão representa um valor esperado condicionado às observações passadas. Outro exemplo, desta vez para o caso não linear, refere-se ao interesse em descrever os movimentos da variância ao longo do tempo, ou seja, a heterocedasticidade. Nesse caso, o foco

da modelagem se volta para a evolução temporal do segundo momento, e os modelos da classe ARCH (autoregressivos condicionalmente heterocedásticos) e suas diversas variações são bastante úteis para a formulação de previsões da volatilidade (Nikolaev *et al.*, 2013).

Com respeito à abordagem via inteligência computacional, o rápido avanço tecnológico tem proporcionado ferramentas que se mostram fortemente competitivas com as técnicas estatísticas, como pôde ser visto na comparação feita por Hwang *et al.* (2012). Uma vantagem do método computacional é o seu aspecto não paramétrico, em que as previsões são elaboradas empiricamente com base, essencialmente, em simulações computacionais. Entre essas técnicas destacam-se as redes neurais artificiais e suas variações, e as máquinas de suporte vetorial (e.g., Niu & Wei, 2013).

A máquina de suporte vetorial (ou SVM, sigla de *support vector machine*) é uma técnica computacional para análise de dados que se fundamenta na teoria do aprendizado estatístico (Vapnik & Chervonenkis, 1971; Vapnik, 1982; Vapnik, 1995). Por causa da sua relativa simplicidade e flexibilidade, esse método tem sido extensivamente estudado e aplicado no contexto de análise de séries temporais para fins de elaboração de previsões (Tay & Cao, 2001).

Quanto à natural questão acerca da escolha da abordagem apropriada para a análise de determinada série temporal, resultados experimentais têm mostrado, dependendo da situação, tanto desempenho superior das previsões fornecidas por inteligência artificial em comparação com os modelos estatísticos (Xie & Hu, 2007) quanto o contrário (Foster *et al.*, 1992). Por isso, aqui prefere-se considerar as duas abordagens como complementares com vistas à melhoria do desempenho das previsões em séries temporais lineares.

Este trabalho concentra seu foco no estudo da máquina de suporte vetorial e suas aplicações em análises de séries temporais. Em particular, propõe-se investigar o desempenho da combinação entre os modelos ARMA e GARCH com o método SVM para fins de previsão da volatilidade de retornos financeiros. Esta proposta se baseia

no fato de que os modelos ARMA são candidatos naturais para a representação de processos lineares, enquanto o SVM tem se mostrado método eficiente e prático para a melhoria do desempenho das previsões (Hung *et al.*, 2014).

1.2 Objetivos

O objetivo geral do trabalho é estudar o método de máquina de suporte vetorial para fins de previsão em análise de séries temporais, mediante combinação com os clássicos modelos ARMA e GARCH. Como esses modelos contemplam variáveis não observáveis, o objeto desta dissertação contempla uma variação do método denominado SVM recorrente proposto por Chen *et al.* (2010).

O SVM recorrente permite descrever a parte das médias móveis no modelo ARMA, bem como modelar a volatilidade de uma série financeira. No entanto, Chen *et al.* (2010) não apresentaram uma implementação computacional que permita replicar seus resultados. Além disso, o SVM recorrente na forma proposta por esses autores possui pontos obscuros como o critério de parada que se baseia na autocorrelação residual de primeira ordem. Nesta dissertação, o estudo do método SVM recorrente contempla sua implementação computacional no *software* R. Depois, realiza-se um estudo de Monte Carlo em que seu desempenho é contrastado com aqueles proporcionados por algumas variações do modelo AR(1)-GARCH(1,1). E, finalmente, uma aplicação com dados reais é feita para mostrar sua competitividade em relação aos tradicionais modelos GARCH.

A seguir, será apresentada uma breve revisão de literatura acerca da máquina de suporte vetorial (SVM) para fins de classificação e de regressão, bem como suas aplicações para previsões em séries temporais.

1.3 Uma breve revisão de literatura

Para iniciarmos a discussão acerca do método a ser estudado neste trabalho, será apresentada nesta seção uma breve história dos métodos de classificação de dados. Na

estatística, um dos primeiros trabalhos nessa área foi apresentado por Fisher (1936), que sugeriu um algoritmo para reconhecimento de padrões. A partir daí, diversos outros autores se empenharam em tornar a ideia de classificação de dados cada vez mais rebuscada. Por exemplo, Aronszajn (1950) introduziu a teoria de reprodução de *kernels* e, posteriormente, Rosenblatt (1957) criou um classificador linear chamado de perceptron, até então o tipo mais simples de rede neural *feedforward*. Depois, Vapnik & Lerner (1963) introduziram o *generalized portrait algorithm*, sendo o algoritmo dos suportes vetoriais, sua generalização não linear.

A partir daí, Vapnik (1995) sugere que houve uma grande mudança na estatística, quando o paradigma de Fisher, introduzido em meados de 1930, foi substituído pelo que se deve saber *a priori* sobre uma dependência funcional, de modo que se possa estimá-la por meio de determinada base de dados. No paradigma de Fisher (1936), a resposta à linha de raciocínio exposta acima se dá de forma muito restrita, pois o interessado deveria saber diversos fatos sobre a dependência. O estudioso deve conhecer a dependência desejada até determinado número finito de parâmetros, e estimar esses valores é considerado o problema da estimação de dependência.

O novo paradigma, que é essencialmente não paramétrico, supera a restrição do anterior no sentido de se exigir conhecimento apenas de algumas das propriedades gerais do grupo de funções às quais pertence a dependência ainda desconhecida.

Se houver duas classes de dados a serem classificadas que possam ser separadas linearmente, qualquer hiperplano que não erre na classificação poderia separar esse conjunto de dados, por exemplo, classificando os de um tipo como -1 e os demais como 1 . Esse tipo de separação se chama separação linear. Entretanto, percebe-se facilmente que existem diversos hiperplanos capazes de satisfazer a proposição de classificar um conjunto de dados.

Como será detalhado no Capítulo 2, o método do SVM consiste em algoritmos computacionais que constroem um hiperplano, ou um conjunto de hiperplanos, em um espaço de muitas ou infinitas dimensões, de forma a encontrar o hiperplano que melhor

separe os dados que se deseja classificar. Geralmente esse procedimento garante que a separação seja feita de modo a se obter a maior distância entre os tipos distintos.

O SVM também pode ser utilizado como um método de regressão, nesse caso, o chamaremos de SVR (do inglês *Support Vector Regression*). No caso do SVR, a variável resposta não será categórica, mas sim um número real.

Os métodos de programação quadrática são utilizados na determinação do suporte vetorial, pois a ideia principal é construir uma função de Lagrange a partir da função objetivo, e suas restrições. Afinal, devemos minimizar a norma composta pelo produto interno dos pesos (comumente chamados de w), somada a uma espécie de média ponderada dos erros ξ_i que estão acima da margem aceita ε_i (dentro da qual consideramos quaisquer erros iguais a zero, e fora da qual, não se admite errar), que estará sujeita à equação de classificação do SVM. Uma explicação mais aprofundada será apresentada no próximo capítulo.

Uma vez que o método foi brevemente apresentado, será exposta a sua relevância no contexto de análise de séries temporais, visto que existe uma grande quantidade de trabalhos que fizeram essa utilização e obtiveram bons resultados empíricos nas mais diversas áreas, o que encoraja o aprofundamento na investigação desse tema.

O uso dos métodos de Máquinas de Suporte Vetorial, foi extensamente discutido em vários trabalhos, com aplicações nas mais diversas áreas. Se pôde verificar a eficácia desse método em estudos como o de Elbisy (2015) que utilizou a abordagem dos SVM testando várias funções de *kernel* para tentar prever determinados parâmetros que proporcionem a previsão do comportamento de ondas marítimas. O autor argumenta que o método dos SVM proporcionou uma facilitação na previsão desejada, pelo fato de na engenharia utilizada para estudos oceanográficos esse tipo de previsão ser considerada um problema de grande complexidade.

Os autores Tay & Cao (2001), utilizaram o SVM para previsões em séries financeiras, com o objetivo de analisar a factibilidade desse método em relação a um outro

de que utiliza Redes Neurais. O experimento mostrou um desempenho melhor do SVM em relação ao outro método estudado.

Outro trabalho que tratou de investigar a eficácia do SVM em previsões de séries temporais financeiras foi o de Zhao *et al.* (2012). Nesse artigo, se comparou o método do SVM com outro chamado de *locality preserving projection* (LPP). O método do SVM foi otimizado via *particle swarm optimization* (PSO), um método computacional que otimiza um problema tentando iterativamente melhorar uma solução candidata no que diz respeito a uma dada medida de qualidade. O SVM, mais uma vez, se mostrou mais eficiente. No caso desse artigo, o autor argumenta que com a melhoria feita por meio do PSO, a previsão conseguida por ele se mostrou competitiva em relação a outras utilizadas para fins práticos no mercado financeiro.

O SVM foi utilizado para prever séries climáticas, por exemplo, no artigo de Sharifi & Souri (2015). Nele os autores utilizaram o método para prever um parâmetro bastante importante na área de sistemas climáticos globais, que é o vapor de água precipitável (PWV), com ele se pode fazer previsões de formações de nuvens, e previsões no curto prazo de chuva. O método do SVM foi comparado com um híbrido de estimatórias harmônicas e mínimos quadrados, mais uma vez se mostrando mais eficiente.

No artigo de Hwang *et al.* (2012) foi feito um estudo do SVM aplicado à hidrologia, no artigo se investigou a factibilidade do método do SVM, e o comparou ao método de Regressões Lineares Múltiplas para ver qual dos dois seria mais eficiente em previsão de séries não-lineares de hidrologia. O resultado foi que o método do SVM se mostrou mais eficiente nas previsões desejadas.

Rubio *et al.* (2011) estudaram alguns modelos do método de SVM aplicados em modelagem e previsão de séries temporais. Após testar diversas funções *kernel*, argumentou que o método é bastante eficiente para o que se desejou fazer no estudo, e que a função utilizada no *kernel* do SVM é de grande importância e influência na eficiência dos modelos gerados para análise de séries temporais.

Foi utilizado no estudo de Köknar-Tezel *et al.* (2011) o método do SVM para análise de séries temporais com dados não balanceados. Os autores propuseram adicionar pontos sintéticos na classe minoritária como uma forma de tratar os dados não balanceados e, uma vez feito isso, o SVM se mostrou um excelente método de previsão e modelagem para as séries estudadas nesse artigo.

Weiqing *et al.* (2015) utilizaram o SVM na análise de séries temporais de temperatura para medições no fluxo de gás. Uma unidade de aquecimento foi instalada na parede externa de um gasoduto para gerar pulsos de calor, e sinais de temperatura dinâmicos foram recolhidos por meio da unidade de aquecimento. O SVM foi utilizado para estudar a relação entre a temperatura e o fluxo de gás, e seu desempenho foi considerado bastante eficaz.

Hung *et al.* (2014) propuseram em seu trabalho um modelo híbrido que integrou o método tradicional de médias móveis (MA) com SRV para previsões de uma série temporal do valor das retiradas nos caixas eletrônicos na Inglaterra. O modelo híbrido se propôs a prever não os valores exatos das retiradas de dinheiro, mas a diferença entre esse valor e a mediana de médias móveis semanais. Os autores chamaram essa técnica de SVR-*median*, e com isso foi feita uma comparação com um modelo MA simples. Nesse estudo, foi possível averiguar que o modelo híbrido se saiu melhor do que o MA simples.

A relevância do estudo da volatilidade foi resumidamente abordada por Chen *et al.* (2008) e Chen *et al.* (2010), e esses dois artigos foram de grande influência na confecção deste trabalho. Nesses dois trabalhos citados foi proposto o que se chamou de SVR recorrente, e este novo método foi comparado ao SVR tradicional e a outros modelos, podendo-se observar a sua superioridade nos exemplos abordados para fins de previsão. A partir dessa ideia foi possível a elaboração do código utilizado nos Capítulos 3 e 4 deste trabalho, que tratam de aprimorá-lo e de aplicá-lo em previsões de dados gerados pelo processo AR-GARCH e de dados reais.

Portanto, se pôde ver que o método do SVM é bastante utilizado em análise de séries temporais, inclusive em artigos recentes, como o de Weiqing *et al.* (2015). Os autores argumentam em seus estudos que o método do SVM é bastante eficaz na previsão e modelagem de séries temporais em diversas áreas, além disso, o método se mostra mais eficiente que diversos outros, como alguns de redes neurais ou outros mais tradicionais, como o de regressões lineares múltiplas.

1.4 Estrutura da dissertação

O Capítulo 2 introduz os conceitos metodológicos do SVM para classificação (Seções 2.1, 2.2 e 2.3) e regressão (Seção 2.4). A Seção 2.5 descreve a implementação computacional do SVM no *software* R segundo o pacote `e1071`, e em seguida é exposto um exemplo aplicado à regressão não linear (Subseção 2.5.1), e outro aplicado à análise de séries temporais (Subseção 2.5.2).

O Capítulo 3 expõe alguns conceitos de volatilidade seguido do modelo GARCH (Seção 3.1), e como este pode ser combinado ao modelo ARMA. Posteriormente, introduz-se o conceito de SVR recorrente (Seção 3.2), e como pode ser combinado à análise de séries temporais visando fazer previsões um passo à frente. Um estudo de Monte Carlo (Metropolis & Ulam, 1949) é apresentado na Seção 3.3, em que se comparam os desempenhos do SVM recorrente contra os do modelo GARCH e suas variações.

Inicialmente, no Capítulo 4, se discute o experimento feito com dados reais por Chen *et al.* (2010), e se propõe um novo critério de parada para o SVR recorrente. Em seguida são mostradas aplicações do SVM recorrente em dois conjuntos de dados reais: a série temporal da taxa de câmbio Libra/Dólar (Seção 4.1) e a série NYSE *Composite Index*, índice que mede o desempenho de todas as ações listadas na Bolsa de Valores de Nova Iorque (Seção 4.2), desta vez com o novo critério de parada. Os dados referentes à taxa de câmbio correspondem ao período de 5 de janeiro de 2004 a 31 de Dezembro de 2007. Já a outra série refere-se ao preço diário de fechamento da NYSE *Composite*, no período de 8 de janeiro de 2004 a 31 de Dezembro de 2007. As taxas

de câmbio foram obtidas por meio do *Policy Analysis Computing and Information Facility in Commerce* (PACIFIC) na *University of British Columbia*, e os dados da NYSE foram obtidos por meio do *Google Finance*.

Finalmente, o Capítulo 5 trata das conclusões deste trabalho.

Capítulo 2

A máquina de suporte vetorial

A máquina de suporte vetorial (SVM) é uma técnica para análise de dados que se baseia na teoria do aprendizado estatístico (Vapnik & Chervonenkis, 1971; Vapnik, 1982; Vapnik, 1995). Ela propicia aplicações, por exemplo, em análise multivariada, análise de regressão (Hill *et al.*, 2011) e análise de séries temporais (Morettin & Toloï, 2006).

Diversos trabalhos na área de estatística têm mostrado resultados interessantes no que se refere ao SVM, como foi discutido no Capítulo 1. O método se mostra bastante competitivo em relação aos métodos estatísticos em geral. Visto isso, neste capítulo detalharemos os aspectos essenciais sobre o SVM, abordando sua formulação para fins de classificação e de regressão. Além disso, a sua aplicabilidade em regressão permite sua utilização para a obtenção de previsões em análise de séries temporais.

Este capítulo está dividido em 5 seções que serão brevemente explicadas a seguir.

A Seção 2.1 introduz o SVM para fins de classificação binária, considerando o caso em que os dados sejam perfeitamente separáveis, em seguida, a Seção 2.2 trata do caso em que os dados não sejam perfeitamente separáveis. Nesse caso, consideram-se variáveis de folga que proporcionam certo grau de flexibilidade ao classificador, de modo que pequenos erros de classificação sejam tolerados.

A Seção 2.3 trata do caso “não linear”. Se x' e x são vetores de dados p -dimensionais, o método de SVM “linear” baseia-se, essencialmente, no produto interno $\langle x', x \rangle$. Dada uma transformação Φ definida em um espaço n -dimensional (espaço característico), o caso “não linear” baseia-se no produto $\langle \Phi(x'), \Phi(x) \rangle$. Assim, o método é, intrinsecamente linear, pois o produto interno é uma operação linear. Isso é chamado de kernel trick que, apesar da sua simplicidade, produz resultados bastante interessantes.

A Seção 2.4 aborda a máquina de suporte vetorial aplicada para fins de regressão. Aqui, a variável destinada para fins de classificação binária é substituída por uma variável resposta.

A Seção 2.5 apresenta brevemente uma implementação computacional no *software* R, com dois exemplos: um de regressão não linear e outro de séries temporais.

2.1 Para fins de classificação binária: caso linearmente separável

Seja \mathbf{X} uma matriz de tamanho $n \times p$ formada por n observações de treinamento num espaço de dimensão p , ou seja

$$x_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1p} \end{pmatrix}, \dots, x_n = \begin{pmatrix} x_{n1} \\ \vdots \\ x_{np} \end{pmatrix}.$$

Considere que a i -ésima linha da matriz \mathbf{X} possa ser classificada em dois tipos, conforme a variável $y_i \in \{-1, +1\}$. Considere ainda um vetor de observações de teste $x^* = (x_1^*, \dots, x_p^*)^\top$, cujos elementos são separados de acordo com os tipos -1 e $+1$ produzidos pelo classificador obtido com base nos dados de treinamento. Para a construção desse classificador, os dados são separados por meio de um hiperplano de forma que

$$\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip} > 0, \quad \text{se } y_i = 1,$$

e

$$\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip} < 0, \quad \text{se } y_i = -1.$$

Essas equações podem ser reescritas em uma só como

$$y_i(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) > 0.$$

Portanto, define-se o hiperplano

$$f(x_i) = \beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}, \quad (2.1)$$

como critério para classificar o tipo da observação de teste x^* com base no sinal da função $f(x^*)$.

Para ilustrar, a Figura 2.1 mostra um hiperplano de um espaço bidimensional, isto é, uma reta $f(x) = 0$ que separa o conjunto de dados em duas partes conforme o sinal de $f(x)$

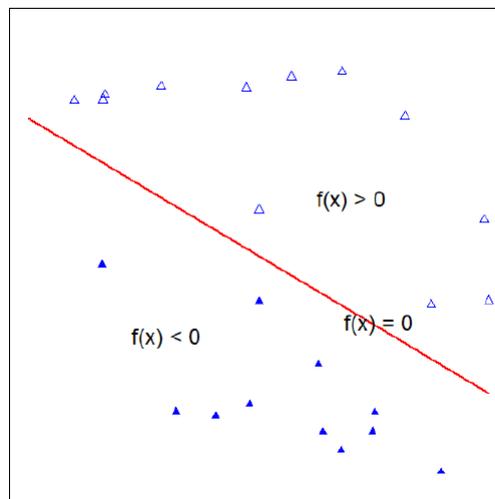


Figura 2.1: Na região à esquerda do hiperplano, em que $f(x) > 0$, temos os pontos do tipo +1 representados por triângulos vazios, enquanto que na região à direita do hiperplano, em que $f(x) < 0$, temos os pontos do tipo -1 representados por triângulos cheios.

A magnitude de $f(x')$ indica o quão distante a observação $x' = (x'_1, \dots, x'_p)^\top$ está do hiperplano. Para mostrarmos isso, vamos lembrar que se a e b forem dois vetores

em \mathfrak{R}^p , formando um ângulo θ entre eles conforme ilustra a Figura 2.2 (a), então temos que $\langle a, b \rangle = \|a\| \times \|b\| \times \cos\theta$.

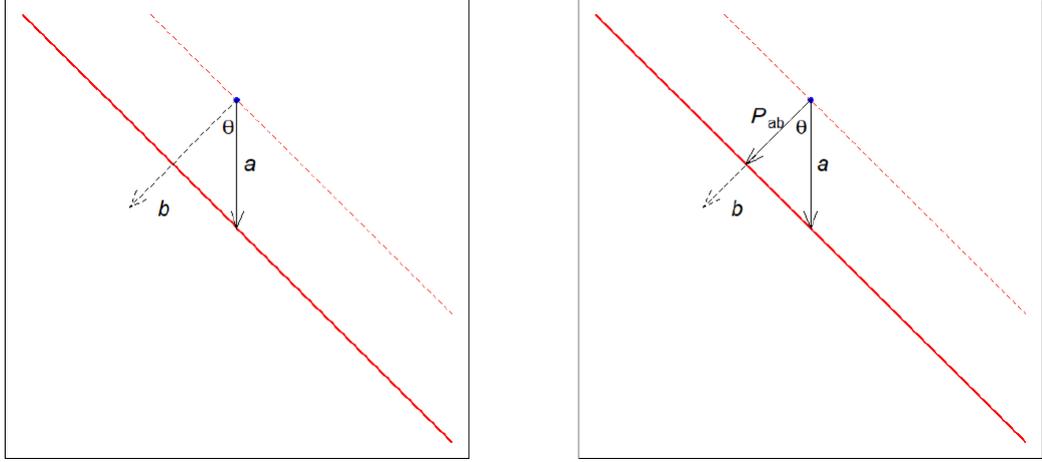


Figura 2.2: Em (a), tem-se os dois vetores a e b , com um ângulo θ formado entre eles, e um hiperplano. Em (b), a projeção P_{ab} do vetor a sobre o vetor b está explicitada.

O módulo da projeção do vetor a sobre o vetor b representado por P_{ab} na Figura 2.2 (b), é $\|P_{ab}\| = \|a\| \times \cos\theta$. Logo, podemos escrever

$$\|P_{ab}\| = \|a\| \times \cos\theta = \frac{\|a\| \times \langle a, b \rangle}{\|a\| \times \|b\|} = \frac{\langle a, b \rangle}{\|b\|}.$$

Em particular, considere a notação vetorial

$$\begin{aligned} f(x) &= \beta_0 + \sum_{j=1}^p \beta_j x_j \\ &= \beta_0 + \langle \beta, x \rangle, \end{aligned} \tag{2.2}$$

em que $\beta = (\beta_1, \dots, \beta_p)^\top$ e $x = (x_1, \dots, x_p)^\top$. Nesse caso, β é um vetor perpendicular ao hiperplano no ponto x , pois $\langle \beta, x \rangle = 0$, e o vetor do ponto x' ao ponto x é

$$w = \begin{pmatrix} x'_1 - x_1 \\ \vdots \\ x'_p - x_p \end{pmatrix}.$$

Logo, a distância $\|d\|$ do ponto x' ao hiperplano é a projeção escalar do vetor w sobre a normal β , ou seja

$$\begin{aligned}
\|d\| &= \|P_{w\beta}\| = \frac{\langle w, \beta \rangle}{\|\beta\|} \\
&= \frac{\langle \beta, x' \rangle - \langle \beta, x \rangle}{\|\beta\|} \\
&= \frac{\langle \beta, x' \rangle - \beta_0}{\|\beta\|} \\
&= \frac{f(x')}{\|\beta\|}.
\end{aligned} \tag{2.3}$$

A distância de um ponto qualquer x' ao hiperplano nos indica, de certa forma, uma medida de precisão na classificação feita. Isso porque à medida que x' se afasta do hiperplano, maior será a certeza acerca de sua classificação. Em contraste, a incerteza tende a aumentar à medida que x' se aproxima do hiperplano. Por desejarmos a maior separabilidade possível, tem-se como objetivo maximizar a magnitude de $f(x')$.

Na Figura 2.3, observe que é possível haver uma infinidade de hiperplanos que efetuem a separação do conjunto de dados de treinamento conforme seu tipo. Para que a ideia de maximização possa remeter a um único hiperplano, é preciso definir o conceito de margem.

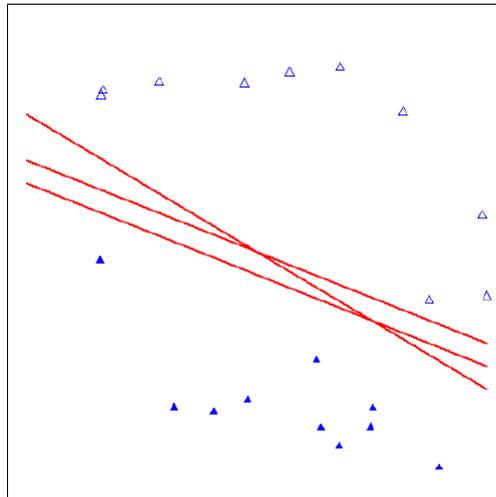


Figura 2.3: A figura mostra alguns dos infinitos hiperplanos que dividem o conjunto de dados.

Se os dados de treinamento forem linearmente separáveis, existem $M > 0$, β_0 e β tais que

$$y_i(\beta_0 + \langle \beta, x_i \rangle) \geq M,$$

para todo $i = 1, \dots, n$. Portanto, a margem que separa as duas classes corresponde à soma da distância do ponto x'_i (com $y'_i = +1$) e do ponto x''_i (com $y''_i = -1$) mais próximos do hiperplano $f(x) = 0$ que os separa (Figura 2.4). As observações x tais que $y_s(\beta_0 + \langle \beta, x_s \rangle) = M$ são chamadas de suporte vetorial.

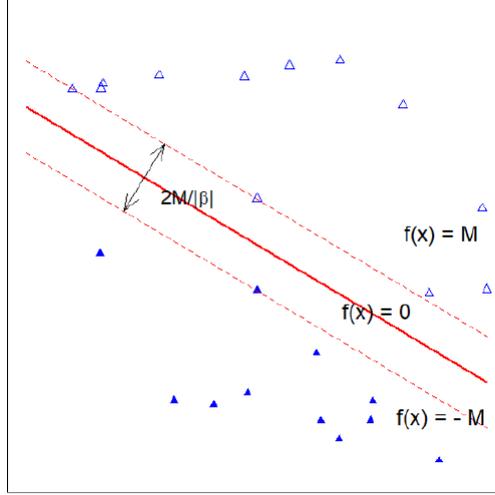


Figura 2.4: Os pontos que estão nas envoltórias convexas, ou *convex hulls* (Barber *et al.*, 1996; Yao, 1981; Preparata & Hong, 1977; Brown, 1979; Chan, 1996; Crisp & Burges, 2000), são chamados de suportes vetoriais. A distância entre essas envoltórias é chamada de margem.

Considerando a distância entre os dois pontos de suporte vetorial $x' \in \mathfrak{R}^p$ (do tipo $y' = +1$) e $x'' \in \mathfrak{R}^p$ (do tipo $y'' = -1$) mais próximos do hiperplano que separa o conjunto de dados, desenvolve-se um raciocínio parecido com o da Figura 2.2. Ou seja, calculamos a distância entre os dois pontos, a multiplicamos por um vetor unitário perpendicular ao hiperplano, e obtemos a projeção cuja medida representará a margem.

Se chamarmos essa distância de $\|D\|$, o melhor hiperplano será o que a maximize, proporcionando a maior separação possível entre os dados de tipos diferentes.

Lembrando que os pontos x de suporte vetorial são tais que $y_s(\beta_0 + \langle \beta, x_s \rangle) = M$, a distância $\|D\|$ (ou margem) que queremos maximizar é dada por

$$\begin{aligned}
\|D\| &= (x' - x'') \frac{\beta}{\|\beta\|} \\
&= (\langle \beta, x' \rangle - \langle \beta, x'' \rangle) \frac{1}{\|\beta\|} \\
&= [(M - \beta_0) - (-M - \beta_0)] \frac{1}{\|\beta\|} \\
&= \frac{2M}{\|\beta\|}.
\end{aligned}$$

Observe que maximizar $2M/\|\beta\|$ é equivalente a minimizar seu inverso multiplicativo $\|\beta\|/2M$. Além disso, Vapnik (1995) definiu que se considerarmos $M = 1$ (hiperplano canônico) existe outra forma de se encontrar o hiperplano ótimo, sem perda de generalidade, achando os β_i de modo a

$$\min_{\beta} \frac{\|\beta\|^2}{2},$$

em que $\beta = (\beta_1, \dots, \beta_p)^\top$, sujeito a

$$y_i(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) > 0 \quad \forall i = 1, \dots, n.$$

Uma vez que se obtém os β_i que otimizam a função anterior, a função classificadora do SVM é dada por

$$g(x) = \frac{f(x)}{|f(x)|} = \text{sign}(f(x)). \quad (2.4)$$

2.2 Para fins de classificação binária: caso não separável linearmente

No caso de não podermos separar os dados linearmente, o conceito de hiperplano que divide perfeitamente as duas classes será modificado para o que “quase separa” as classes. Assim é preciso considerar uma margem mais flexível (*soft margin*). A generalização do classificador de máxima margem para o caso não separável é conhecida como classificador de suporte vetorial.

Essa mudança consiste no seguinte problema de maximização

$$\max_{\beta_0, \dots, \beta_p, \xi_1, \dots, \xi_n} M,$$

sujeito a

$$\sum_{j=1}^p \beta_j^2 = 1, \quad (2.5)$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) \geq M(1 - \xi_i), \quad \forall i = 1, \dots, n, \quad (2.6)$$

e

$$\sum_{i=1}^n \xi_i \leq C, \quad \xi_i \geq 0, \quad (2.7)$$

em que C é um parâmetro de ajuste, e ξ_1, \dots, ξ_n são as variáveis de folga. A inclusão dessas variáveis de folga permite que alguns pontos sejam classificados incorretamente. Se $\xi_i = 0$ a observação i seria classificada corretamente, estando no lado correto da margem. Se $0 < \xi_i < 1$ a observação i também seria classificada corretamente, mas estando no lado errado da margem. E, se $\xi_i > 1$, ela seria classificada incorretamente, estando do lado errado do hiperplano. Se os valores dos parâmetros de folga forem demasiadamente elevados, haverá muitos pontos incorretamente classificados (ou seja, um ponto x_k incorretamente classificado será aquele tal que $y_k(\beta_0 + \langle \beta, x_k \rangle) \leq 0$).

Assim, a restrição dada pela Equação 2.7 proporciona um limite superior para o erro de classificação gerado pelo classificador da equação 2.4. Para minimizar a Equação 2.7 e, ao mesmo tempo, maximizar a margem, a formulação SVM para o caso de dados não separáveis consiste em

$$\min_{\beta_0, \dots, \beta_p, \xi_1, \dots, \xi_n} \frac{1}{n} \sum_{i=1}^n \xi_i + \frac{\lambda \|\beta\|^2}{2}, \quad (2.8)$$

sujeito a $\xi_i + y_i(\beta_0 + \langle \beta, x_i \rangle) \geq 1$, (considerando $M = 1$, sem perda de generalidade), em que $\lambda > 0$ é um parâmetro de ajuste que controla o balanceamento entre o erro e a margem.

Dada uma constante a , considere a notação $[a]_+ = \max\{a, 0\} = \min(\xi_i$ sujeito a

$\xi_i \geq 0$ e $\xi_i \geq a$) com base nela, a Equação 2.8 pode ser expressa como

$$\frac{1}{n} \sum_{i=1}^n [1 - y_i(\beta_0 + \langle \beta, x_i \rangle)]_+ + \frac{\lambda \|\beta\|^2}{2},$$

ou

$$\frac{1}{n} \sum_{i=1}^n [1 - y_i f(x_i)]_+ + \frac{\lambda \|\beta\|^2}{2}.$$

Nessa expressão, o termo

$$L(f(x_i), y_i) = [1 - y_i f(x_i)]_+$$

denomina-se função de perda, e $f(x_i) = y_i(\beta_0 + \langle \beta, x_i \rangle)$ é a margem do ponto (x_i, y_i) .

Considerando $M = 1$, note que a restrição dada pela Equação 2.6 pode ser reescrita como

$$\xi_i - 1 + y_i(\beta_0 + \langle \beta, x_i \rangle) \geq 0. \quad (2.9)$$

Assim, introduzindo dois conjuntos de multiplicadores de Lagrange, um para a restrição dada pela Equação 2.9 e outro para a restrição $\xi_i \geq 0$, temos o Lagrangeano

$$l_p(\beta_0, \beta, \xi, \alpha, \gamma) = \sum_{i=1}^n \xi_i + \frac{n\lambda \|\beta\|^2}{2} + \sum_{i=1}^n \alpha_i [1 - y_i(\beta_0 + \beta_1 x_{i1} \dots + \beta_p x_{ip}) - \xi_i] - \sum_{i=1}^n \gamma_i \xi_i. \quad (2.10)$$

Derivando a Equação 2.10 com respeito a β , β_0 e ξ_i , obtém-se

$$\frac{\partial l_p}{\partial \beta} = n\lambda\beta - \sum_{i=1}^n \alpha_i y_i x_i = 0 \iff \beta = \frac{1}{n\lambda} \sum_{i=1}^n \alpha_i y_i x_i, \quad (2.11)$$

$$\frac{\partial l_p}{\partial \beta_0} = - \sum_{i=1}^n \alpha_i y_i = 0 \iff \sum_{i=1}^n \alpha_i y_i = 0, \quad (2.12)$$

$$\frac{\partial l_p}{\partial \xi_i} = 1 - \alpha_i - \gamma_i = 0 \iff \gamma_i = 1 - \alpha_i, \quad (2.13)$$

para $\alpha_i \geq 0$, $\gamma_i \geq 0$ e $i = 1, \dots, n$. Simplificando l_p de acordo com esses resultados, obtemos o problema dual

$$\begin{aligned}
& \sum_{i=1}^n \xi_i + \frac{n\lambda\|\beta\|^2}{2} + \sum_{i=1}^n \alpha_i [1 - y_i(\beta_0 + \langle \beta, x_i \rangle) - \xi_i] - \sum_{i=1}^n \gamma_i \xi_i = \\
&= \sum_{i=1}^n (1 - \alpha_i - \gamma_i) \xi_i + \frac{n\lambda\|\beta\|^2}{2} + \sum_{i=1}^n \alpha_i - \beta_0 \sum_{i=1}^n \alpha_i y_i - \sum_{i=1}^n \alpha_i y_i \langle \beta, x_i \rangle \\
&= \sum_{i=1}^n \alpha_i + \frac{n\lambda\|\beta\|^2}{2} - \sum_{i=1}^n \alpha_i y_i \sum_{j=1}^n \beta_j x_{ij} \\
&= \sum_{i=1}^n \alpha_i + \frac{n\lambda}{2(n\lambda)^2} \left\langle \sum_{i=1}^n \alpha_i y_i x_i, \sum_{j=1}^n \alpha_j y_j x_j \right\rangle \\
&= \sum_{i=1}^n \alpha_i + \frac{1}{2n\lambda} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - \frac{1}{n\lambda} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \\
&= \sum_{i=1}^n \alpha_i - \frac{1}{2n\lambda} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle,
\end{aligned} \tag{2.14}$$

ou seja,

$$\min_{\alpha_1, \dots, \alpha_n} \sum_{i=1}^n \alpha_i - \frac{1}{2n\lambda} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle, \tag{2.15}$$

sujeito a $0 \leq \alpha_i \leq 1$ e $\sum_{i=1}^n \alpha_i y_i = 0$, para $i, j = 1, \dots, n$.

Computacionalmente se chega à conclusão de que a otimização que se deseja realizar para os classificadores de suporte vetorial se resumem a produtos internos entre as observações, e não com as próprias observações.

Também observa-se que o classificador de suporte vetorial linear pode ser escrito como

$$f(x) = \beta_0 + \langle \beta, x \rangle = \beta_0 + \left\langle \frac{1}{n\lambda} \sum_{i=1}^n \alpha_i y_i x_i, x \right\rangle = \beta_0 + \frac{1}{n\lambda} \sum_{i=1}^n \alpha_i y_i \langle x_i, x \rangle, \tag{2.16}$$

em que se tem n parâmetros α_i , um para cada observação de treinamento.

O problema dual revela algumas características interessantes do SVM. Em pri-

meio lugar, como ele requer n variáveis, o tamanho da amostra representa o fator dominante da complexidade do problema em vez do número p de preditores. Em segundo lugar, a solução $\hat{\alpha}$ depende apenas dos produtos internos dos dados de treinamento $\langle x_i, x_j \rangle$ da Equação 2.15. Em terceiro lugar, uma vez que $\hat{\alpha}$ satisfaz o critério $0 \leq \hat{\alpha} \leq 1$ e a condição de equilíbrio $\sum_{i=1}^n \hat{\alpha}_i y_i = 0$, tem-se o valor normal ao hiperplano ótimo, dado por $\hat{\beta} = \frac{1}{n\lambda} \sum_{i=1}^n \hat{\alpha}_i y_i x_i = 0$. E em quarto lugar, condicionando os pontos x'_i tais que $1 - y_i(\hat{\beta}_0 + \langle \hat{\beta}, x'_i \rangle) = 0$, tem-se $\hat{\beta}_0 = y'_i - \frac{1}{n\lambda} \sum_{i=1}^n \hat{\alpha}_i y_i \langle x_i, x'_i \rangle$.

Observe que, se $y_i(\hat{\beta}_0 + \langle \hat{\beta}, x_i \rangle) > 1$, então temos $\xi_i = 0$ (restrição da Equação 2.8) e, por isso, $\alpha_i = 0$. Por outro lado, se $\hat{\alpha}_i > 0$, o ponto constitui o suporte vetorial S , de modo que podemos reescrever

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i \langle x_i, x \rangle. \quad (2.17)$$

2.3 Para fins de classificação binária: caso não linear

Uma extensão direta do SVM para o caso não linear consiste em substituir o produto interno $\langle x_i, x_j \rangle$ do espaço Euclidiano por um outro espaço característico, em que se consideram transformações de $x \in \mathfrak{R}^p$ para um espaço de dimensão superior.

Seja $\Phi(x) = (\phi_1(x), \dots, \phi_n(x))^\top$ um vetor de transformações de x . Substituindo-se $\langle x_i, x_j \rangle$ por $\langle \phi(x_i), \phi(x_j) \rangle$, tem-se de imediato uma extensão do SVM linear.

Por exemplo, considere $x = (x_1, x_2)^\top \in \mathfrak{R}^2$, e defina $\Phi : \mathfrak{R}^2 \rightarrow \mathfrak{R}^3$ como $\Phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)^\top$. Essa transformação proporciona um novo produto interno no espaço característico dado por

$$\langle \Phi(x), \Phi(q) \rangle = (x_1^2, x_2^2, \sqrt{2}x_1x_2) \begin{pmatrix} q_1^2 \\ q_2^2 \\ \sqrt{2}q_1q_2 \end{pmatrix} = \langle x, q \rangle^2,$$

em que $q = (q_1, q_2)^\top$ e $x = (x_1, x_2)^\top$.

Para essa generalização não é necessário especificar a função Φ , sendo suficiente especificar uma função $K(x, q)$ que resulta do produto interno $\langle \Phi(x), \Phi(q) \rangle$. Tendo em mãos uma função K , a função de discriminação não linear pode ser escrita como

$$\hat{f}(x) = \frac{1}{n\lambda} \sum_{i=1}^n \hat{\alpha}_i y_i K(x_i, x) + \hat{\beta}_0. \quad (2.18)$$

Desse modo, a forma da curva que proporciona a classificação é determinada por K .

Essa função, que se denomina *kernel*, deve ser uma função simétrica e não negativa. Entre os *kernels* mais usuais, encontram-se o linear,

$$K(x_i, x_l) = \langle x_i, x_l \rangle, \quad (2.19)$$

o polinomial de grau d

$$K(x_i, x_l) = \gamma(C_0 + \langle x_i, x_l \rangle)^d, \quad (2.20)$$

o radial (ou gaussiano),

$$K(x_i, x_l) = \exp(-\gamma \|x_i - x_l\|^2), \quad (2.21)$$

e o sigmóide (ou sigmoidal),

$$K(x_i, x_l) = \tanh(C_0 + \gamma \langle x_i, x_l \rangle), \quad (2.22)$$

em que C_0 é uma constante e $\tanh(x)$ é a função tangente hiperbólica de x .

A Figura 2.5, retirada de James *et al.* (2013), ilustra a aplicação do *kernel* polinomial e do *kernel* radial para a classificação do conjunto de dados.

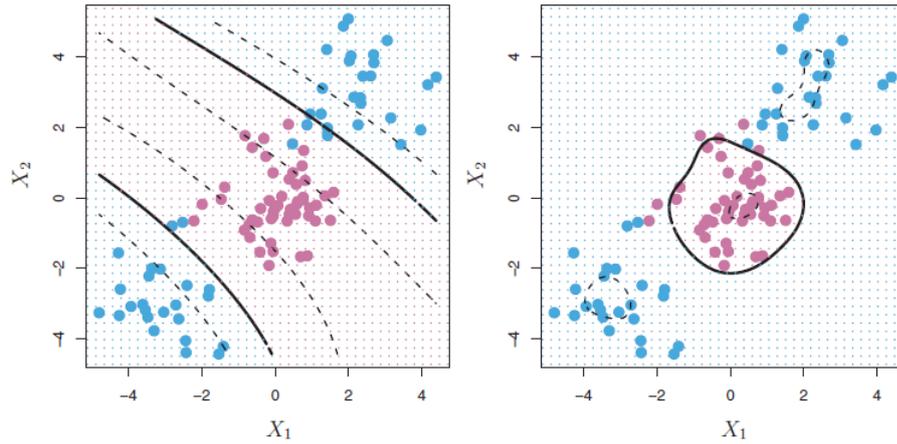


Figura 2.5: Máquinas de suporte vetorial: à esquerda temos o resultado proporcionado por um *kernel* polinomial de grau 3, e à direita, para o mesmo conjunto de dados, observa-se o resultado da aplicação de um *kernel* radial (James *et al.*, 2013).

2.4 Para fins de regressão

Nas seções anteriores, a máquina de suporte vetorial (SVM) foi apresentada como uma ferramenta para classificação. Esta seção trata da regressão por suporte vetorial (SVR), em que $y_i \in \mathfrak{R}$ é uma variável resposta e os elementos x_{i1}, \dots, x_{ip} são as variáveis regressoras. Do ponto de vista estatístico, a regressão do tipo SVR é uma técnica não paramétrica, em que se obtém um ajuste para a resposta y com base no princípio da máxima margem.

Considere inicialmente a função linear

$$f(x_i) = \beta_0 + \langle \beta, x \rangle, \quad (2.23)$$

em que $f(x_i)$ seja tal que minimize a norma

$$\frac{\|\beta\|^2}{2} = \frac{\langle \beta, \beta \rangle}{2}, \quad (2.24)$$

sujeita a uma condição do tipo

$$|y_i - f(x_i)| \leq \varepsilon, \quad \forall i = 1, \dots, n. \quad (2.25)$$

A Equação 2.25 é conhecida como L1 ou ε -insensível, e significa que o problema de otimização da Equação 2.24 se restringe aos resíduos absolutos inferiores a ε . No

entanto, é possível que nem todos os pontos possam atender a essa restrição. Para acomodar tais pontos, são introduzidas variáveis de folga ξ_i e ξ_i^* para cada ponto. Isso é similar à ideia de *soft margin* em SVM para fins de classificação. Agora, a função a ser maximizada é

$$\frac{\|\beta\|^2}{2} + C \sum_{i=1}^n (\xi_i + \xi_i^*), \quad (2.26)$$

sujeita às restrições

$$y_i - f(x_i) \leq \varepsilon + \xi_i, \quad (2.27)$$

e

$$f(x_i) - y_i \leq \varepsilon + \xi_i^*, \quad (2.28)$$

em que $\xi_i \geq 0$, $\xi_i^* \geq 0$ e $\forall i = 1, \dots, n$. A constante $C > 0$ permite controlar a penalidade imposta às observações que se encontram fora da margem ε .

A função de perda ε -insensível (Vapnik, 1995) é dada por

$$L_\varepsilon(y) = \begin{cases} 0, & \text{para } |f(x) - y| < \varepsilon, \\ |f(x) - y| - \varepsilon, & \text{caso contrário.} \end{cases} \quad (2.29)$$

Essa função de perda ignora os erros inferiores a ε , tratando-os como nulos. A Figura 2.6 mostra a forma gráfica da função de perda ε -insensível.

Com base na função objetivo primal $\|\beta\|^2/2$ e suas restrições, o Lagrangeano L é obtido mediante utilização de quatro multiplicadores de Lagrange: η_i , η_i^* , α_i , e α_i^* , de modo que

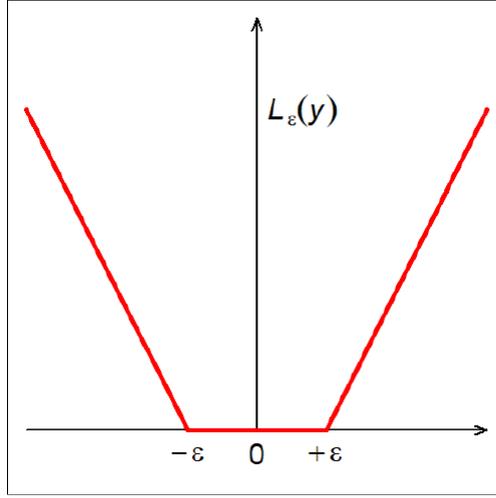


Figura 2.6: Função de perda ε -insensível na linha mais espessa em vermelho.

$$\begin{aligned}
L &= \frac{\|\beta\|^2}{2} + C \sum_{i=1}^n (\xi_i + \xi_i^*) - \sum_{i=1}^n (\eta_i \xi_i + \eta_i^* \xi_i^*) \\
&\quad - \sum_{i=1}^n \alpha_i (\varepsilon + \xi_i - y_i + \langle \beta, x_i \rangle + \beta_0) \\
&\quad - \sum_{i=1}^n \alpha_i^* (\varepsilon + \xi_i^* + y_i - \langle \beta, x_i \rangle - \beta_0),
\end{aligned} \tag{2.30}$$

em que $\eta_i \geq 0$, $\eta_i^* \geq 0$, $\alpha_i \geq 0$ e $\alpha_i^* \geq 0$.

A otimização de L requer

$$\frac{\partial L}{\partial \beta_0} = \sum_{i=1}^n (\alpha_i^* - \alpha_i) = 0, \tag{2.31}$$

$$\frac{\partial L}{\partial \beta} = \beta - \sum_{i=1}^n (\alpha_i - \alpha_i^*) x_i = 0, \tag{2.32}$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \eta_i = 0, \tag{2.33}$$

e

$$\frac{\partial L}{\partial \xi_i^*} = C - \alpha_i^* - \eta_i^* = 0. \tag{2.34}$$

Substituindo as Equações 2.31, 2.32, 2.33 e 2.34 em 2.30, obtém-se o problema dual. Da Equação 2.32, temos

$$\beta = \sum_{i=1}^n (\alpha_i - \alpha_i^*),$$

de modo que a função de regressão possa ser escrita como

$$\begin{aligned} f(x) &= \beta_0 + \left\langle \sum_{i=1}^n (\alpha_i - \alpha_i^*) x_i, x \right\rangle \\ &= \beta_0 + \sum_{i=1}^n (\alpha_i - \alpha_i^*) \langle x_i, x \rangle. \end{aligned} \quad (2.35)$$

Agora, a solução para o problema primal dos SVM será dada por

$$\max \left[-\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle + \sum_{i=1}^n \alpha_i (y_i - \xi) - \alpha_i^* (y_i + \xi) \right], \quad (2.36)$$

restrito a

$$0 \leq \alpha_i, \alpha_i^* \leq C \quad \forall i = 1, \dots, n,$$

e

$$\sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0,$$

em que C é um valor pré-especificado.

Para obter β_0 , consideram-se as condições

$$\alpha_i (\varepsilon + \xi_i - y_i + \langle \beta, x_i \rangle, \beta_0) = 0, \quad (2.37)$$

$$\alpha_i^* (\varepsilon + \xi_i^* - y_i + \langle \beta, x_i \rangle, \beta_0) = 0, \quad (2.38)$$

$$(C - \alpha_i) \xi_i = 0, \quad (2.39)$$

e

$$(C - \alpha_i^*) \xi_i^* = 0, \quad (2.40)$$

em que o produto entre as restrições e os seus respectivos multiplicadores de Lagrange

se anulam (Karush, 1939; Kuhn & Tucker, 1951). Se $0 < \alpha_i < C$, então $\xi_i = 0$ e $\varepsilon - y_i \langle \beta, x_i \rangle + \beta_0 = 0$. Analogamente, se $0 < \alpha_k^* < C$, temos $\xi_k^* = 0$ e $\varepsilon + y_k - \langle \beta, x_k \rangle - \beta_0 = 0$.

Vejamos agora o caso não linear

$$f(x_i) = \beta_0 + \langle \beta, \Phi(x_i) \rangle, \quad (2.41)$$

em que $\Phi(x) = [\phi_1(x), \dots, \phi_p(x)]^\top$, $\beta = (\beta_1, \dots, \beta_p)^\top$. A transformação Φ é uma função de transferência não linear que mapeia a entrada x do espaço \mathfrak{R}^p para o espaço característico desejado. Com base na forma expandida $f(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + \beta_0$, podemos escrever 2.40 como

$$\begin{aligned} f(x_i) &= \beta_0 + \sum_{i=1}^n (\alpha_i - \alpha_i^*) \langle \phi(x_i), \phi(x) \rangle \\ &= \beta_0 + \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(x_i, x), \end{aligned} \quad (2.42)$$

em que $K(x_i, x)$ é a função *kernel*. Assim, é suficiente especificar a função K em vez de explicitar a transformação Φ . Por analogia aos resultados anteriores, isso nos remete ao seguinte problema de otimização, maximizar

$$\max_{\alpha, \alpha^*} -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) K(x_i, x_j) - \varepsilon \sum_{i=1}^n (\alpha_i - \alpha_i^*) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*), \quad (2.43)$$

sujeito a $\sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0$, $0 \leq \alpha_i \leq C$ e $0 \leq \alpha_i^* \leq C$.

Na prática, os valores ε e C são determinados com base em experimentos computacionais. A próxima seção descreve algumas implementações computacionais no *software* R e exemplos serão desenvolvidos nas subseções 2.5.1 e 2.5.2.

2.5 Implementação computacional no *software* R

No *software* R, o pacote `e1071`, que é gratuito e de fácil utilização, permite utilizar a metodologia SVM para fins de classificação e de regressão. Em notação matricial, os modelos apresentados neste capítulo que se encontram implementados nele são:

- *C-classification* (Equação 2.18):

$$\min_{\alpha} \quad \frac{1}{2} \alpha^{\top} \mathbf{Q} \alpha - \mathbf{e}^{\top} \alpha,$$

sujeito a

$$0 \leq \alpha_i \leq C, \quad i = 1, \dots, n, \quad \mathbf{y}^{\top} \alpha = 0,$$

em que \mathbf{e} é o vetor unidade, C é o limite superior, \mathbf{Q} é uma matriz $n \times n$ semidefinida positiva, com $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, e $K(x_i, x_j) \equiv \langle \phi(x_i), \phi(x_j) \rangle$ é o *kernel*. Os *kernels* disponibilizados pelo pacote são o linear (Equação 2.19), o polinomial (Equação 2.20), o radial (Equação 2.21) e o sigmóide (Equação 2.22).

- ε -*regression* (Equação 2.26):

$$\min_{\alpha, \alpha^*} \quad \frac{1}{2} (\alpha - \alpha^*)^{\top} \mathbf{Q} (\alpha - \alpha^*) + \varepsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*)$$

sujeito a

$$0 \leq \alpha_i, \alpha_i^* \leq C, \quad i = 1, \dots, n, \quad \sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0.$$

Para fins ilustrativos, a seguir serão exemplificados casos simples feitos no *software* R por meio do pacote `e1071`.

2.5.1 Exemplo: aplicação em regressão não-linear

Considere o conjunto de dados

Tabela 2.1: Dados utilizados no exemplo computacional da Subseção 2.5.1.

$\{\mathbf{x}_i\}_{i=1, \dots, 10}$	1	2	3	4	5	6	7	8	9	10
$\{\mathbf{y}_i\}_{i=1, \dots, 10}$	1	4	8	13	18	22	24	25	26	27

Diversos modelos foram ajustados aos dados. Para cada *kernel*, foram testados os valores $\varepsilon_1 = 0, 1$, $\varepsilon_2 = 1$ e $\varepsilon_3 = 5$ para a variável ε ; $C_1 = 1$, $C_2 = 5$ e $C_3 = 10$ para a variável C . Quanto ao parâmetro γ do *kernel* radial (Equação 2.21), utilizou-se o valor padrão do *software* R, isto é, $\gamma = 0, 1$.

Para comparar o desempenho proporcionado por esses ajustes foram adotados duas medidas, o erro quadrático médio (EQM) e o erro médio absoluto (EMA), que se definem como

$$\text{EQM} = \sum_{i=1}^n \frac{(x_i - f(x_i))^2}{n}, \quad (2.44)$$

e

$$\text{EMA} = \sum_{i=1}^n \frac{|x_i - f(x_i)|}{n}, \quad (2.45)$$

em que $f(x_i)$ é a predição do modelo que se deseja testar e x_i o seu valor real, para $i = 1, \dots, n$.

Por meio da Tabela 2.2, chegamos à conclusão de que o modelo que conseguiu o melhor ajuste foi o de *kernel* radial com $\varepsilon = 0, 1$ e $C = 10$, resultando em EQM=1,07 e EMA=0,42. Como foi discutido na Equação 2.26, o parâmetro $C > 0$ penaliza as observações que se encontram fora da margem ε . Nesse caso, um valor de C alto associado a um ε baixo significa que a função ε -insensível (Equação 2.29) será mais rígida em relação aos possíveis erros de classificação, gerando mais pontos de suporte vetorial. Além disso, esse exemplo foi feito com um conjunto de dados pequeno o que facilitou o acerto de todos os pontos.

Para fins de previsão um passo à frente, os dados são separados em dois conjuntos, um de treinamento e outro de validação. Assim, utilizamos o conjunto de treinamento para ajustar os modelos de SVR e o de validação para testar as previsões passo a passo. Em vez disso, neste exemplo treinamos o SVR com todo o conjunto de dados para posteriormente prevê-los. Esses fatos reunidos podem explicar o excesso de

aderência do modelo que obteve EQM=1,07 e EMA=0,42. No caso em que se deseja fazer previsões, consideraremos que o modelo ajustado representará a generalização desse classificador, podendo assim prever satisfatoriamente um conjunto de novos dados (no caso, a parte separada para validação). Se o modelo estiver muito bem ajustado aos dados utilizados em seu treinamento, poderá apresentar uma baixa taxa de acerto quando confrontado com novos dados, e isso se denomina *overfitting* ou superajustamento (Lorena & Carvalho, 2007).

A Tabela 2.2 mostra os resultados do EQM e EMA para os C e ε testados. Em seguida o Programa 2.1 mostra o código utilizado para realizar os ajustes comentados nesta seção.

Tabela 2.2: Valores de C e ε selecionados para cada *kernel*, e resultados do EQM e EMA dos modelos de SVR com *kernel* linear e polinomial de grau 3 ajustados aos dados da Tabela 2.1.

Modelo	ε_i	C_i	EQM	EMA
SVR Linear	$\varepsilon_1 = 0,1$	$C_1 = 1$	6,03	1,92
	$\varepsilon_2 = 1$	$C_1 = 1$	5,5	2,1
	$\varepsilon_3 = 5$	$C_1 = 1$	12,7	3,22
	$\varepsilon_1 = 0,1$	$C_2 = 5$	6,03	1,92
	$\varepsilon_2 = 1$	$C_2 = 5$	5,5	2,1
	$\varepsilon_3 = 5$	$C_2 = 5$	12,7	3,22
	$\varepsilon_1 = 0,1$	$C_3 = 10$	6,03	1,92
	$\varepsilon_2 = 1$	$C_3 = 10$	5,5	2,10
	$\varepsilon_3 = 5$	$C_3 = 10$	12,7	3,22
SVR Radial	$\varepsilon_1 = 0,1$	$C_1 = 1$	71,26	6,47
	$\varepsilon_2 = 1$	$C_1 = 1$	71,26	6,47
	$\varepsilon_3 = 5$	$C_1 = 1$	67,64	6,98
	$\varepsilon_1 = 0,1$	$C_2 = 5$	15,06	2,08
	$\varepsilon_2 = 1$	$C_2 = 5$	14,58	2,53
	$\varepsilon_3 = 5$	$C_2 = 5$	25,56	4,78
	$\varepsilon_1 = 0,1$	$C_3 = 10$	1,07	0,42
	$\varepsilon_2 = 1$	$C_3 = 10$	1,98	1,19
	$\varepsilon_3 = 5$	$C_3 = 10$	20,9	4,42

Programa 2.1: Aplicação do SVR em regressão não-linear

```

1 library(e1071)
2
3 # Formulando os dados
4
5 x <- 1:10; x <- as.numeric(x)
6 y <- c(1,4,8,13,18,22,24,25,26,27)
7 dados <- data.frame(x,y)
8
9 # Testando valores de epsilon e C nos diferentes tipos de kernel
10
11 i<-0
12 EQM <- NULL
13 EMA <- NULL
14 svm.eps <- NULL
15 svm.cost <- NULL
16 #trocar kernel: "linear" e "radial"
17 kernel <- "linear"
18 for (svm_eps in c(0.1,1,5))
19 {
20   for (svm_cost in c(1,5,10))
21   {
22     set.seed(777)
23     svr.lin <- svm(y ~ x,dados,scale=FALSE,type="eps-regression",
24                  kernel=kernel,cost=svm_cost,epsilon=svm_eps)
25     EQM[1+i] <- round(mean((svr.lin$residuals)^2),2)
26     EMA[1+i] <- round(mean(abs(svr.lin$residuals)),2)
27     svm.eps[1+i] <- svm_eps
28     svm.cost[1+i] <- svm_cost
29     i <- i+1
30   }
31 }
32 resultados <- cbind(svm.eps,svm.cost,EQM,EMA);View(resultados)
33

```

A representação gráfica de alguns dos modelos é mostrada nos gráficos da Figura 2.7.

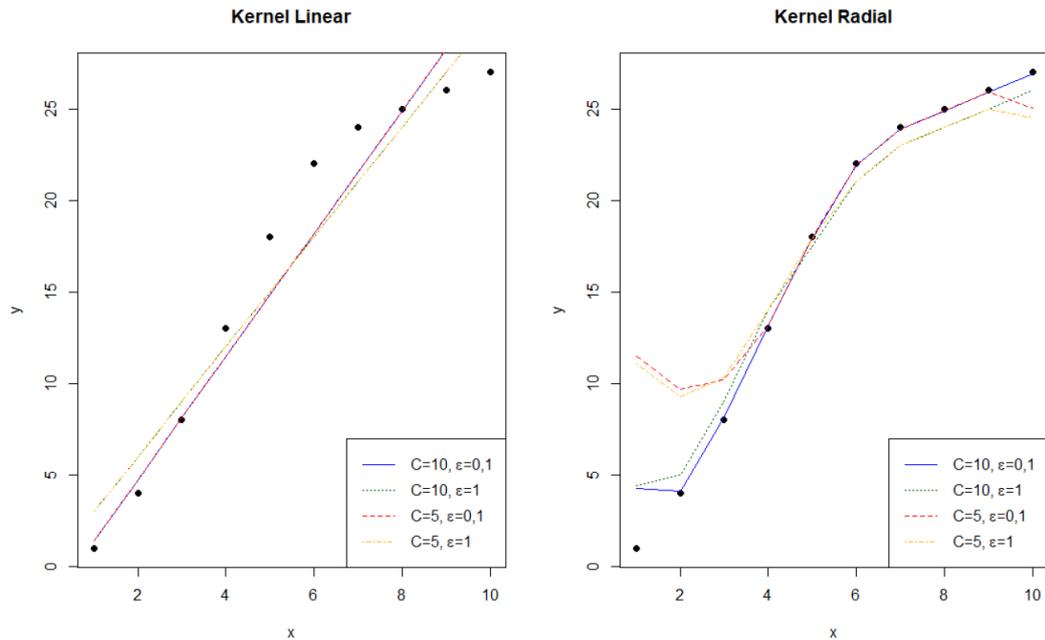


Figura 2.7: Curvas ajustadas por SVR aos dados (pontos pretos) da Tabela 2.1 utilizando o *kernel* linear no gráfico à esquerda, e o radial à direita.

2.5.2 Exemplo: aplicação em séries temporais autorregressivas

Considere uma série temporal $\{X_t\}_{t=1,\dots,1050}$ que segue um processo autorregressivo de ordem 2, AR(2), dado por

$$X_t = 0,7X_{t-1} - 0,2X_{t-2} + e_t, \quad (2.46)$$

em que $X_0 \equiv 0$ e $\{e_t\}$ representa uma sequência de erros aleatórios gerados por uma distribuição normal padrão, com média $\mu = 0$ e variância $\sigma^2 = 1$.

Considerando que o objetivo é se fazer previsões um passo à frente, o conjunto de dados $\{X_1, \dots, X_m, X_{m+1}, \dots, X_n\}$ é dividida em duas partes. A primeira constitui os dados de treinamento $\{X_1, \dots, X_m\}$, e a segunda os dados de validação $\{X_{m+1}, \dots, X_n\}$. Os dados de treinamento são utilizados para o ajuste inicial do modelo (e, na prática, é utilizado para a identificação do processo gerado que, neste caso, já se sabe que é o AR(2)).

A segunda parte dos dados é utilizada para a avaliação do desempenho do modelo. Com base nas m observações anteriores, obtém-se uma previsão para X_{m+1} , denotada por $\hat{X}_{m+1|m}$. Definindo-se erro de previsão um passo à frente como $X_m - \hat{X}_{m|m-1} = r_m$, podemos calcular medidas de desempenho do modelo ajustado como o EQM e o EMA (vistos na subseção anterior) na forma

$$\text{EQM} = \sum_{t=m+1}^n \frac{(\hat{X}_{t|t-1} - X_t)^2}{n - m}, \quad (2.47)$$

e

$$\text{EMA} = \sum_{t=m+1}^n \frac{|\hat{X}_{t|t-1} - X_t|}{n - m}. \quad (2.48)$$

Na etapa de validação, o modelo é reajustado sequencialmente à medida que uma nova observação do conjunto de dados de validação torna-se disponível. Para este exemplo, consideramos $m = 1000$. Usando, portanto, as 50 últimas observações como dados de validação, a Tabela 2.3 mostra os resultados proporcionados por diversos

ajustes.

Tabela 2.3: Valores de C e ε selecionados para cada *kernel*, e resultados do EQM e EMA dos modelos de SVR com *kernel* linear e polinomial de grau 3 ajustados aos dados provenientes do processo AR(2) definido na Equação 2.46.

Modelo	ε_i	C_i	EQM	EMA
AR(2)	-	-	0,88	0,81
SVR Linear	$\varepsilon_1 = 0,1$	$C_1 = 1$	0,92	0,82
	$\varepsilon_2 = 1$	$C_1 = 1$	0,91	0,81
	$\varepsilon_3 = 5$	$C_1 = 1$	1,37	1,17
	$\varepsilon_1 = 0,1$	$C_2 = 5$	0,92	0,82
	$\varepsilon_2 = 1$	$C_2 = 5$	0,92	0,82
	$\varepsilon_3 = 5$	$C_2 = 5$	1,37	1,17
	$\varepsilon_1 = 0,1$	$C_3 = 10$	0,92	0,82
	$\varepsilon_2 = 1$	$C_3 = 10$	0,92	0,82
	$\varepsilon_3 = 5$	$C_3 = 10$	1,38	1,17
SVR Radial	$\varepsilon_1 = 0,1$	$C_1 = 1$	0,94	0,83
	$\varepsilon_2 = 1$	$C_1 = 1$	0,97	0,81
	$\varepsilon_3 = 5$	$C_1 = 1$	1,17	1,08
	$\varepsilon_1 = 0,1$	$C_2 = 5$	0,96	0,84
	$\varepsilon_2 = 1$	$C_2 = 5$	0,99	0,83
	$\varepsilon_3 = 5$	$C_2 = 5$	1,16	1,08
	$\varepsilon_1 = 0,1$	$C_3 = 10$	0,97	0,84
	$\varepsilon_2 = 1$	$C_3 = 10$	1,00	0,84
	$\varepsilon_3 = 5$	$C_3 = 10$	1,36	1,17

Com base nesses resultados, o modelo que apresentou o melhor ajuste foi o AR(2), com EQM=0,88 e EMA=0,81. Entretanto, os ajustes do SVR se mostraram bons competidores em comparação ao modelo linear AR(2). O modelo SVR que apresentou melhor ajuste foi o de *kernel* linear, com $\varepsilon = C = 1$, EQM=0,91 e EMA=0,81.

A Figura 2.8 ilustra as previsões obtidas por dois desses modelos. Visualmente, observa-se que os resultados são muito semelhantes.

Por fim, o Programa 2.2 mostra o código utilizado para realizar os ajustes comentados nesta seção.

Programa 2.2: Aplicação do SVR em séries temporais auto-regressivas

```

1 library(e1071)
2 library(xts)
3 require(zoo)
4
5 # Formulando os dados
6
7 N.dados <- 1050
8 N.dados.1 <- 1000
9 set.seed(777)
10 y.t <- arima.sim(n=N.dados,list(order=c(2,0,0),ar=c(0.7,-0.2)),sd = 1)
11
12 # Modelo AR
13
14 x.t <- y.t[1:N.dados.1]
15 fit.ar <- arima(x.t,order=c(2,0,0))
16 res.x <- resid(fit.ar)
17 pred.x <- as.ts(x.t - res.x)
18
19 # Previsoes AR, k passos a frente
20
21 P.ar <- rep(0,50)
22 error.ar <- rep(0,50)
23 pred.y <- predict(fit.y,n.ahead = 1)
24 P.ar[1] <- pred.y$pred[1]
25 error.ar[1] <- y.t[1000+1] - pred.y$pred[1]
26
27 for (h in 1:49)
28 {
29   y = y.t[1:(1000+h)]
30
31   # reajusta o ARMA com base nas 1000+h observacoes disponiveis
32   fit.y <- arima(y,order=c(2,0,0))
33
34   # previsao para y.t[1000+h+1]
35   pred.y <- predict(fit.y,n.ahead = 1)
36   P.ar[h+1] = pred.y$pred[1]
37   error.ar[h+1] = y.t[1000+h+1] - pred.y$pred[1]
38 }
39
40 # Selecionando o melhor modelo SVR
41
42 x <- y.t[1:N.dados.1]
43 x.1 <- c(NA ,y.t[1:(N.dados.1-1) ])
44 x.2 <- c(NA ,NA ,y.t[1:(N.dados.1-2) ])
45 input.trein <- na.trim(cbind(x, x.1,x.2))
46
47 svm.eps <- 0.1
48 svm.cost <- 1
49 svm.fit <- svm(x=input.trein[,-1],y=input.trein[,1],
50               scale = FALSE,type = "eps-regression",
51               kernel = "radial", cost = svm.cost,
52               epsilon = svm.eps)
53
54 # Previsao com SVR
55
56 y.hat <- NULL # previsao 1 passo a frente
57 u.t <- NULL # erro de previsao 1 passo a frente
58 for (tau in 1:50)
59 {
60   x.new <- cbind(y.t[N.dados.1-1+tau],y.t[N.dados.1-2+tau])
61   y.hat[tau] <- predict(svm.fit, newdata = x.new,scale = FALSE)
62   u.t[tau] <- (y.hat[tau]-y.t[N.dados.1+tau])
63   new.line <- c(y.t[N.dados.1+tau],y.t[N.dados.1-2+tau])
64   input.updated <- rbind(input.trein,new.line)
65   rownames(input.updated) <- NULL
66   svm.fit <- svm(x=input.updated[,-1],y=input.updated[,1],scale = FALSE,
67                 type = "eps-regression", kernel = "radial",
68                 cost = svm.cost, epsilon = svm.eps)
69 }
70
71 # EQM
72
73 EQM.SVR <- round(mean(u.t^2),2);EQM.SVR
74 EQM.AR <- round(var(error.ar),2);EQM.AR
75
76 # EMA
77
78 EMA.SVR <- mean(abs(u.t));EMA.SVR
79 EMA.AR <- mean(abs(error.ar));EMA.AR

```

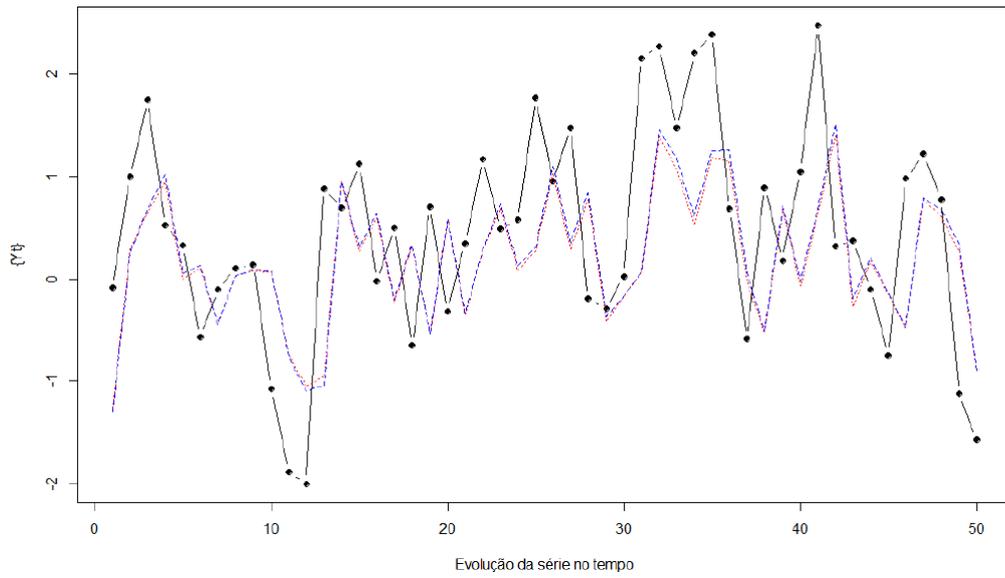


Figura 2.8: Evolução das últimas 50 observações da série $\{X_t\}$ (bolas e traços pretos) e suas previsões por meio do modelo AR(2) (em linha tracejada azul), e do ajuste obtido pelo SVR de *kernel* linear com $C = \varepsilon = 1$ (em linha pontilhada vermelha).

Capítulo 3

Previsão de volatilidade via SVR recorrente

Volatilidade é uma característica bastante importante em análise de séries temporais financeiras, pois ela remete a questões de avaliação e gestão de riscos. De forma resumida, dada uma série temporal $\{Y_t\}_{t=1,\dots,n}$, em um modelo de primeira ordem, define-se volatilidade como

$$h_t = \text{Var}(Y_t|Y_{t-1}). \quad (3.1)$$

Assim, para a análise de volatilidade definida pela Equação 3.1, encontram-se modelos como o GARCH (*Generalized Autoregressive Conditional Heterocedastic*) proposto por Engle (1982) e Bollerslev (1986). Esse modelo se popularizou devido a sua capacidade de descrever aspectos característicos em dados financeiros como a persistência e a aglomeração de volatilidades.

A definição de volatilidade da Equação 3.1, apesar de ser uma das mais conhecidas não é a única. Além dela, é comum encontrarmos

$$\bar{h}_t = (Y_t - \bar{Y})^2, \quad (3.2)$$

em que \bar{Y} representa a média histórica desses retornos.

Neste capítulo abordaremos a previsão da volatilidade por SVM, com base na idealização da volatilidade proporcionada pelo modelo GARCH, que é descrito brevemente na Seção 3.1. A Seção 3.2 introduz uma versão do SVM chamada “SVR recorrente”, em que as estimativas das variâncias condicionais são obtidas recursivamente com base em estimativas precedentes dos resíduos. E a Seção 3.3 mostra um exemplo obtido por meio de simulação de um processo AR(1)-GARCH(1,1), cuja implementação computacional em R se encontra no Apêndice A.

3.1 Modelo GARCH

A modelagem e previsão de volatilidade tem sido o foco de estudo de muitos pesquisadores, especialmente na área financeira, pois ela tem grande impacto na lucratividade de uma carteira de ativos. Um dos modelos mais conhecidos para modelar volatilidade é o GARCH, desenvolvido por Bollerslev (1986), que consiste em uma generalização do modelo ARCH proposto por Engle (1982). Uma das vantagens do modelo GARCH em relação ao ARCH é o fato de ele ser capaz de descrever a volatilidade de uma série temporal prescrevendo-se uma quantidade menor de parâmetros, o que o torna mais parcimonioso.

O modelo ARMA (*autoregressive moving average*), descrito na tese de doutorado de Whittle (1951) e popularizado por Box & Jenkins (1970), é um dos modelos mais famosos em análise de séries temporais. O fato de ele ser relativamente simples, o torna um dos primeiros modelos a serem estudados por quem se interessa por essa área da estatística. No entanto o modelo ARMA refere-se a filtros lineares, ou seja, diz respeito a modelos lineares de séries temporais. Por outro lado, o modelo GARCH permite descrever a variância condicional de uma série temporal, e por isso, trata-se de um modelo não linear de séries temporais. Assim, em séries temporais financeiras, é comum combinarmos o modelo ARMA com o GARCH, sendo que o primeiro se aplica a retornos, e o segundo a volatilidades.

Como foi discutido no Capítulo 2, os SVR também são bastante sensíveis à não linearidade de séries temporais a serem modeladas, devido às transformações Φ pelas

quais os dados passam, como se pode ver na Equação 2.41. Portanto, a princípio, é intuitivo comparar os dois modelos, GARCH e SVR, com o objetivo de se descrever a volatilidade de uma série temporal financeira.

O modelo GARCH(p,q) é definido como

$$y_t = \sqrt{h_t} e_t, \quad (3.3)$$

em que

$$h_t = \kappa + \sum_{j=1}^q \delta_j h_{t-j} + \sum_{i=1}^p \alpha_i y_i^2, \quad (3.4)$$

sendo $\{e_t\}$ uma sequência de erros *iid* (independente e identicamente distribuída), $\kappa > 0$, $\alpha_i \geq 0$ e $\delta_j \geq 0$. As constantes α_i e δ_j normalmente são calculadas por meio de regressão linear. Bollerslev (1986) mostra que a estacionariedade do modelo GARCH(p,q) é garantida se

$$\sum_{i=1}^m \delta_i + \alpha_i < 1, \quad (3.5)$$

em que $m = \max(p, q)$.

Diversos autores (Akgiray, 1989; Franses & Dijk, 1996; Brooks, 1998; Gokcan, 2000; Anderson, 2001; Brooks & Persaud, 2003; Poon & Granger, 2003; Gerlach & Tuyl, 2006) discutem que o modelo GARCH(1,1) é bom o suficiente para descrever aspectos típicos do comportamento da variância de retornos financeiros, como o acúmulo de volatilidades. Na prática, o modelo GARCH pode ser combinado com outros modelos de séries temporais. Um exemplo de modelo híbrido ARMA-GARCH é o modelo AR(1)-GARCH(1,1), que pode ser escrito como

$$y_t = c + \phi_1 y_{t-1} + u_t, \quad (3.6)$$

em que $u_t \sim N(0, h_t)$ e

$$h_t = \kappa + \delta_1 h_{t-1} + \alpha_1 u_{t-1}^2, \quad (3.7)$$

sendo $\kappa \geq 0$, $\delta_1 \geq 0$, $\alpha_1 \geq 0$, c e ϕ_1 parâmetros constantes, em que $\delta_1 + \alpha_1 < 1$ para evitar variâncias negativas. Essa restrição é equivalente à Equação 3.5 (Bollerslev, 1986).

A variância condicional de u_t é dada por $h_t = E_{t-1}(u_t^2) = \hat{u}_{t|t-1}^2$. Franses & Dijk (1996) mostram que as variâncias condicionais de um processo GARCH podem ser modeladas por um processo ARMA. Assim, a variância condicional de u_t proveniente de um processo GARCH pode ser escrita em termos do seguinte modelo ARMA

$$\begin{aligned} h_t &= \kappa + \delta_1 h_{t-1} + \alpha_1 u_{t-1}^2 + (u_t^2 - u_t^2) \\ u_t^2 &= \kappa + \delta_1 h_{t-1} + \alpha_1 u_{t-1}^2 + u_t^2 - h_t + (\delta_1 u_{t-1}^2 - \delta_1 u_{t-1}^2) \\ u_t^2 &= \kappa + (\delta_1 + \alpha_1) u_{t-1}^2 - \delta_1 (u_{t-1}^2 - h_{t-1}) + u_t^2 - h_t. \end{aligned} \quad (3.8)$$

Considerando $w_t \equiv u_t^2 - \hat{u}_{t|t-1}^2 = u_t^2 - h_t$, pode-se reescrever a Equação 3.8 como

$$u_t^2 = \kappa + (\delta_1 + \alpha_1) u_{t-1}^2 - \delta_1 w_{t-1} + w_t, \quad (3.9)$$

em que w_t é um ruído.

Como uma alternativa empírica de se descrever a volatilidade, alguns autores utilizam o método MA (*moving average*). Esse método utiliza médias móveis ponderadas do quadrado das inovações do passado para prever volatilidade (Niemira & Klein, 1994). Para fazer previsões um passo à frente por meio de um processo MA em dados simulados, utilizando-se as τ últimas observações temos

$$\hat{u}_{t+1} = \frac{1}{\tau} \sum_{i=t-\tau+1}^t u_i^2. \quad (3.10)$$

Para utilizarmos a Equação 3.10 em dados reais, fazemos

$$\hat{u}_{t+1} = \frac{1}{\tau} \sum_{i=t-\tau+1}^t (y_i - \bar{y}_{\tau,t})^2, \quad (3.11)$$

uma vez que os erros u_t não são observáveis (Engle *et al.*, 1993), sendo

$$\bar{y}_{\tau,t} = \frac{1}{\tau} \sum_{i=t-\tau+1}^t y_i.$$

3.2 SVR recorrente

Como foi discutido no Capítulo 2, o SVR geralmente é formado por uma rede *feedforward* simples alimentada por um conjunto de informações. Essas informações são projetadas por uma função *kernel* em um espaço de múltiplas dimensões onde, por meio de métodos computacionais, se obtém um classificador ou modelo de regressão. Entretanto, esse processo requer que as variáveis sejam todas observáveis. Observe nas Equações 3.6 e 3.7 que a variável y_t depende de uma componente não observável, h_t . Assim, é preciso modificar o algoritmo SVR para que seja possível contemplar tais variáveis.

Do modelo AR(1)-GARCH(1,1) apresentado na Equação 3.6, a parte autoregressiva pode ser descrita por um SVR na forma

$$y_t = f(y_{t-1}) + u_t. \quad (3.12)$$

Mas, com respeito à parte correspondente ao modelo GARCH(1,1), devemos considerar, com base na Equação 3.9,

$$u_t^2 = g(u_{t-1}^2, w_{t-1}) + w_t, \quad (3.13)$$

em que f e g são funções não lineares e não paramétricas para modelar a média e a variância condicional. Observe que o termo não observável w_t foi introduzido no modelo GARCH, e que o erro desse processo ARMA não se comporta como um ruído branco. Assim, o comportamento desses erros pode trazer consigo alguma informação relevante ao modelo. Para que sejam feitas as alterações necessárias, haverá um novo passo na estimação por SVR, que se dá pelo uso de ramos específicos compostos por um operador de atraso, z^{-1} .

Os resíduos do modelo SVR gerados em cada *loop* serão usados como estimativas

para a série de erros w_t de modo iterativo, semelhante à forma que o modelo ARMA linear é iterativamente estimado por mínimos quadrados (Box, 1994; Hamilton, 1997). Para inicializar o *loop*, considera-se que os resíduos sejam iguais a zero. Esse modelo será chamado de SVR recorrente.

A Figura 3.1 mostra a lógica do SVR comum, no esquema (a), e a do SVR recorrente, no esquema (b). O SVR recorrente tem maior habilidade de captar características dinâmicas da série temporal, quando comparado ao modelo SVR convencional. Isso resulta em um impacto considerável na capacidade de aprendizagem do SVR.

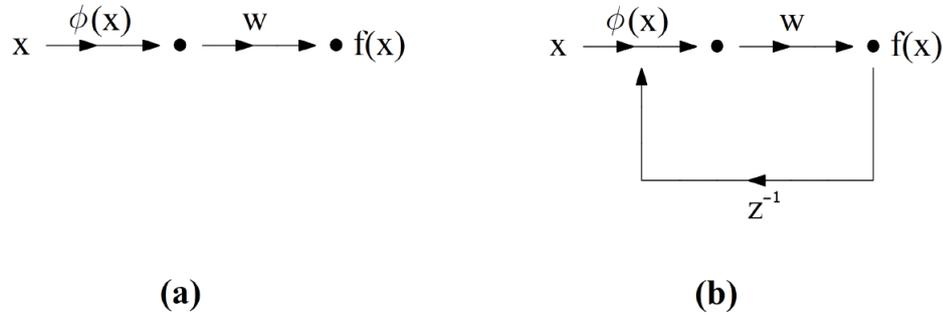


Figura 3.1: O esquema (a) representa o SVR tradicional, e o esquema (b) representa o SVR recorrente.

Para que seja feita a previsão da equação da volatilidade mostrada na Equação 3.2, que serviria de *proxy* para a volatilidade da Equação 3.1 do modelo GARCH, um novo algoritmo será descrito a seguir em quatro passos, em que será acrescido i para indicar a iteração.

- Passo 1: Faça $i = 1$ e comece com todos os resíduos zerados, ou seja $w_t^{(1)} = 0$.
- Passo 2: Execute o SVR para ter a função de decisão $f^{(i)}$, Equação 3.12, para os pontos $\{x_t, y_t\} = \{u_{t-1}^2, u_t^2\}$, com as entradas $x_t = \{u_{t-1}^2, w_{t-1}^{(i)}\}$.
- Passo 3: Compute os novos resíduos $w_t^{(i+1)} = u_t^2 - f^{(i)}$.
- Passo 4: Termine o processo computacional quando o critério de parada for satisfeito; caso contrário, faça $i = i + 1$ e volte ao Passo 2.

Nesse modelo, o tamanho do fator de atraso geralmente é desconhecido, ou seja, não se sabe quantas vezes a parte recorrente deve ser repetida. Assim, na prática define-se um critério de parada. Chen *et al.* (2010) consideram que o modelo SVR recorrente converge quando os resíduos correspondentes passam a se comportar como ruído, ou deixam de ser autocorrelacionados. No entanto essa avaliação é feita apenas na autocorrelação de primeira ordem, o que não garante que as demais autocorrelações não sejam significativas. Por isso, neste trabalho utilizamos como critério de parada o valor da correlação entre $w_t^{(i)}$ e $w_t^{(i+1)}$. Caso essa correlação seja alta, entende-se que não houve mudança significativa na estimativa do modelo do passo i para o passo $i + 1$.

Considere o conjunto de dados $\{x_1, \dots, x_{T1}, x_{T1+1}, \dots, x_T\}$, em que a série $\{x_1, \dots, x_{T1}\}$ representa o conjunto para treinamento, e $\{x_{T1}, \dots, x_T\}$ é o conjunto para validação do modelo.

Para fins de previsão, primeiramente estimamos a Equação 3.12 por meio do SVR convencional com base na série de treinamento, de onde se obtém os resíduos u_1, \dots, u_{T1} . No caso do modelo SVM-GARCH, estimaremos a Equação 3.7 da média condicional h_t por meio do SVR, e obteremos as estimativas $\sqrt{\hat{h}_t}$, uma por vez. Os resíduos quadrados obtidos servirão para prever a volatilidade um passo à frente.

O esquema de previsão recursiva será feito de modo que tanto as estimativas quanto as previsões serão atualizadas passo a passo, introduzindo-se sequencialmente uma nova observação estocada no conjunto de validação, como se observa no esquema da Figura 3.2. A cada passo, obtém-se uma estimativa atualizada do modelo. Aqui propõe-se estimar a volatilidade condicional em uma série temporal proveniente de dados simulados por meio do SVR recorrente. A primeira amostra para treinamento é u_1^2, \dots, u_{T1}^2 , com $T1 < T$. As observações de $T - T1$ serão utilizadas como amostra para testes de previsão ou de ajuste. Portanto, se pode estimar e prever a volatilidade condicional obtida por meio do SVR para $n = T - T1$ vezes.

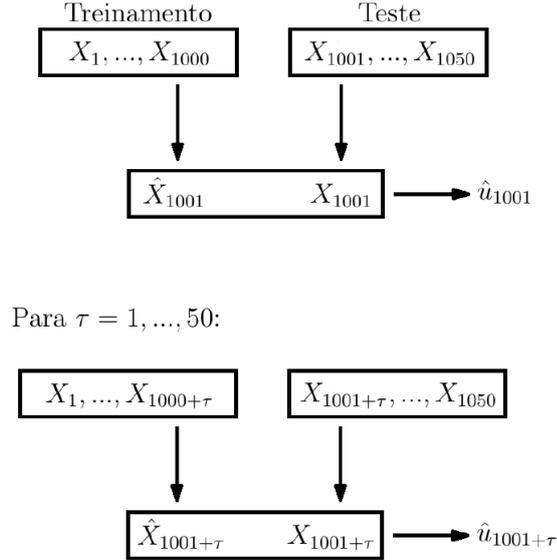


Figura 3.2: Esquema de obtenção passo a passo de $\{\hat{u}_t\}$.

3.3 Exemplo

Para exemplificar o que foi exposto anteriormente, nesta seção será mostrada uma implementação feita no *software* R.

Primeiramente, se deve lembrar que a convergência do procedimento de SVR recorrente considera-se alcançada quando o critério de parada for satisfeito. Tal critério pode ser qualquer um que indique que não há alterações substanciais no modelo de uma iteração i para a $i + 1$. Por exemplo, Chen *et al.* (2010), consideram como critério de parada a autocorrelação serial dos resíduos. Segundo esses autores, o processo iterativo para caso os resíduos se comportem como ruído. Para a formulação do critério de parada, Chen *et al.* (2010) supuseram que se os resíduos não apresentarem autocorrelação de primeira ordem, não deveriam apresentar autocorrelações de ordens superiores. Para testar essa autocorrelação de primeira ordem, eles utilizaram o teste *Q-test* de Ljung-Box-Pierce (Box & Pierce, 1970; Ljung & Box, 1978), que testa se existe autocorrelação entre os resíduos. A estatística do teste *Q-test* de Ljung-Box-Pierce consiste em

$$Q = n(n+2) \sum_{k=1}^h \frac{\hat{\rho}_k^2}{(n-k)}, \quad (3.14)$$

em que n é o tamanho da amostra, $\hat{\rho}_k^2$ é a autocorrelação da amostra na defasagem k , e h é o número de defasagens a serem testadas. A hipótese nula desse teste é que os dados não são autocorrelacionados, e a alternativa é que os dados apresentam correlação serial. Sob a hipótese nula, a estatística Q segue uma distribuição $\chi_{(h)}^2$, e para um nível de significância α , a região crítica para rejeição da hipótese nula é

$$Q > \chi_{1-\alpha, h}^2, \quad (3.15)$$

em que $\chi_{1-\alpha, h}^2$ é o quantil α da distribuição χ^2 com h graus de liberdade.

Neste trabalho será proposto um novo critério de parada, tendo em vista que o critério de parada de Chen *et al.* (2010) nem sempre é razoável, pois o fato de uma série de resíduos não apresentar autocorrelação de primeira ordem não implica que as autocorrelações de ordens superiores possam ser desprezíveis (Morettin & Toloi, 2006).

Propõe-se aqui um critério que consiste em um teste recorrente de correlação entre os resíduos obtidos na iteração $i+1$ e os obtidos na iteração anterior i , ou seja, $cor(w_t^{(i)}, w_t^{(i+1)})$. Enquanto, digamos, $cor(w_t^{(i)}, w_t^{(i+1)}) < 0,95$, o processo iterativo permanece, pois acredita-se que ainda possa haver modificação na estimativa do modelo, que se reflete nos resíduos. Por outro lado, digamos, se $cor(w_t^{(i)}, w_t^{(i+1)}) \geq 0,95$, é razoável supor que a próxima iteração não acrescenta modificações substanciais no modelo. Ou seja, considera-se que a partir daí os próximos $w_t^{(i+k)}$, com $k = 1, \dots, \infty$, se comportariam de forma muito parecida com $w_t^{(i)}$.

Para ilustrar o mecanismo SVR recorrente com o novo critério de parada, considere a série temporal $\{y_t\}_{t=1, \dots, 1000}$ gerada pelo seguinte processo AR-GARCH

$$y_t = 0,5y_{t-1} + u_t, \quad (3.16)$$

com $u_t = \sqrt{h_t}e_t$, e

$$h_t = 0,0005 + 0,1u_{t-1}^2 + 0,8h_{t-1}, \quad (3.17)$$

em que $\{e_t\}$ é uma sequência *iid* de erros aleatórios. Neste exemplo, consideramos cinco diferentes distribuições: normal (De Moivre, 1718), normal assimétrica (Az-zalini, 1985), t-Student (Student, 1908), t-Student assimétrica (Theodossiou, 1998), GED (Giller, 2005) e GED assimétrica (Theodossiou, 2015).

Neste exemplo, em vez de testarmos diversos valores para os parâmetros ε e C do modelo SVR recorrente, por se tratar da reprodução do experimento de Chen *et al.* (2010), foram utilizados os mesmos parâmetros para fins de comparação, ou seja, $\varepsilon = 0,0001$ e $C = 0,05$. Para encontrarem o melhor modelo preditivo, Chen *et al.* (2010) testaram diversos valores para os dois parâmetros na tentativa de seus resultados (no conjunto de dados de teste) fossem capazes de generalizar o modelo para a análise de dados futuros (um passo à frente), ou seja, utilizaram o método de validação cruzada. Chen *et al.* (2010) testaram também por validação cruzada diversos parâmetros nos *kernels* polinomial, radial e linear, e selecionaram, por meio do EMA, o radial como *kernel* de melhor desempenho, portanto neste exemplo utilizaremos apenas o *kernel* radial.

A Tabela 3.1 mostra o efeito da mudança na distribuição dos erros $\{e_t\}$ sobre os modelos GARCH em comparação ao modelo SVR recorrente. Para avaliar o desempenho dos modelos, foram utilizados como critério as medidas EMA e EQM, e para cada uma se obteve os valores referentes à série temporal $\{u_t^2\}$ (que funcionou como *proxy* da série $\{h_t\}$). As estimativas da série temporal $\{u_t^2\}$ foram geradas por meio do SVR tradicional, visto que o processo GARCH não pode gerar essas estimativas (Chen *et al.*, 2010).

A cada vez que mudamos a distribuição dos ruídos $\{e_t\}$, mudamos a base de dados a ser ajustada por meio dos modelos, portanto, para cada distribuição que os ruídos seguem, obteve-se resultados diferentes para o SVR recorrente e o modelo

GARCH apropriado para o banco de dados em questão. Assim, a Tabela 3.1 deve ser interpretada de modo a comparar os modelos dois a dois, pois teremos um SVR recorrente e um GARCH para cada distribuição dos ruídos. A Tabela 3.1 apresenta o valor médio dos EQM e EMA obtidos em 101 simulações para cada distribuição dos ruídos do processo GARCH gerador do banco de dados, seguida do seu erro padrão em parênteses. Para o tratamento dessas estimativas, foram consideradas *outliers* as simulações com $EQM > 0,0003$, pois foram muito poucas e distorceram as estimativas e gráficos dos modelos GARCH e SVR recorrente.

Quanto aos resultados, todos os modelos tiveram resultados parecidos, pois se compararmos as estimativas dois a dois, podemos observar que o erro-padrão sempre engloba o valor do modelo concorrente. Por exemplo, para ruídos normais, o modelo SVR recorrente obteve o resultado médio de $EMA=0,0023797$ com erro padrão de $0,0012696$, e quando o comparamos ao modelo GARCH, temos o resultado médio $EMA=0,0025055$ com erro padrão de $0,0011983$, isto é, estatisticamente os resultados são iguais. Entretanto, se observarmos apenas as médias, o SVR recorrente se saiu melhor em relação a todos os modelos GARCH quanto ao EMA, e pior em relação ao EQM (menos no caso dos ruídos seguindo uma t-Student padrão). Isso nos dá indícios de que o SVR recorrente erra menos vezes, mas quando erra tem erros bem maiores que o GARCH, assim, quando elevamos o erro ao quadrado no caso dos EQM, evidenciamos as rodadas em que o SVR recorrente se saiu pior.

Para ilustrar, a Figura 3.3 mostra o desempenho dos EQM e EMA em 20 das simulações que foram obtidas para cada um dos seis bancos de dados. A Figura 3.4 mostra os EQM e EMA dos modelos SVR recorrentes e de seus respectivos modelos AR-GARCH em um gráfico em que um é a imagem do outro. As Figuras 3.3 e 3.4 reforçam os resultados da Tabela 3.1 de que o comportamento dos dois tipos de modelos é muito parecido. O que leva à intuição de que o modelo SVR recorrente seria um bom concorrente ao tradicional modelo GARCH. As figuras referentes aos demais modelos se encontram no Apêndice A, e o código utilizado no *software* R se encontra no Apêndice B.

Tabela 3.1: Valores dos EMAs e EQMs referentes aos ajustes dos modelos de SVR recorrente e GARCH. A série temporal $\{e_t\}$ é uma sequência *iid* de ruídos e se comporta de acordo com as seis distribuições de probabilidade comentadas neste capítulo. Nesta tabela observam-se os resultados referentes à série temporal teórica gerada pelo processo AR-GARCH descrito nas Equações 3.16 e 3.17. Cada estimativa se refere ao valor médio das simulações obtidas por Monte Carlo e está acompanhada de seu erro-padrão em parênteres.

Ruído	Modelo	EMA	EQM
Normal padrão	SVR recorrente	0,0023797 ($\pm 0,0012696$)	0,0000405 ($\pm 0,0000567$)
	GARCH	0,0025055 ($\pm 0,0011983$)	0,0000360 ($\pm 0,0000480$)
Normal assimétrica	SVR recorrente	0,0025624 ($\pm 0,0014273$)	0,0000486 ($\pm 0,0000658$)
	GARCH	0,0026037 ($\pm 0,0013043$)	0,0000404 ($\pm 0,0000512$)
GED	SVR recorrente	0,0023764 ($\pm 0,0012558$)	0,0000405 ($\pm 0,0000582$)
	GARCH	0,0024943 ($\pm 0,0011943$)	0,0000365 ($\pm 0,0000510$)
GED assimétrica	SVR recorrente	0,0024565 ($\pm 0,0013387$)	0,0000453 ($\pm 0,0000597$)
	GARCH	0,0025153 ($\pm 0,0012333$)	0,0000378 ($\pm 0,0000525$)
t-Student	SVR recorrente	0,0015831 ($\pm 0,0009465$)	0,0000266 ($\pm 0,0000502$)
	GARCH	0,001863 ($\pm 0,0009592$)	0,0000266 ($\pm 0,0000494$)
t-Student assimétrica	SVR recorrente	0,0018541 ($\pm 0,0012767$)	0,0000362 ($\pm 0,0000593$)
	GARCH	0,0020018 ($\pm 0,0011712$)	0,0000323 ($\pm 0,0000526$)

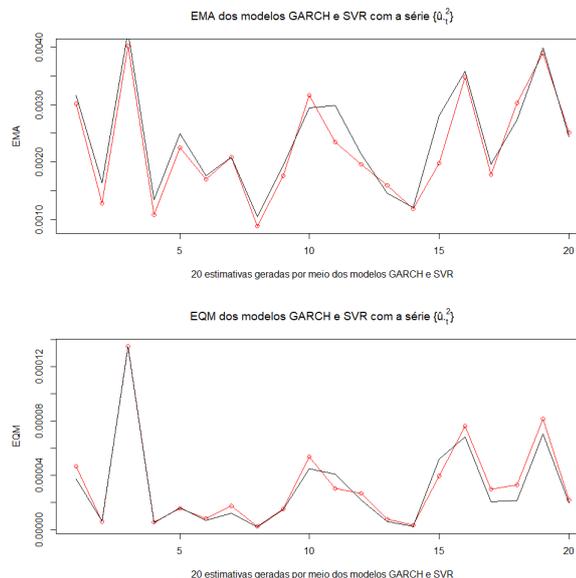


Figura 3.3: Os gráficos mostram 20 resultados dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) dos modelos GARCH e SVR recorrente, ambos supondo $\{e_t\}$ se comportando como uma normal padrão.

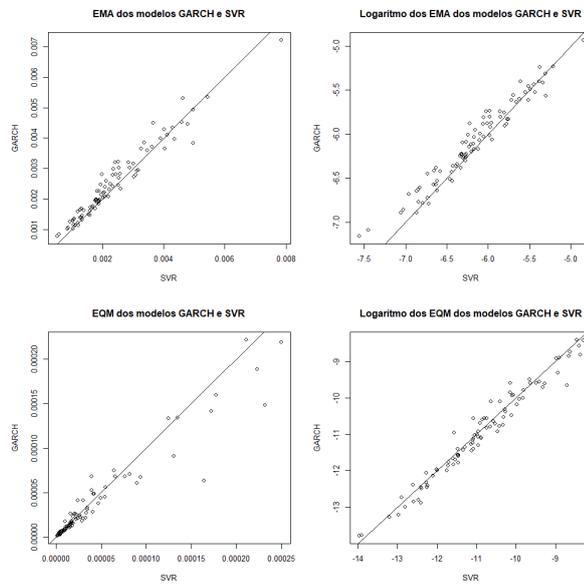


Figura 3.4: À esquerda, os gráficos mostram os EMA (gráfico de cima) e os EQM (gráfico de baixo) confrontando os resultados dos modelos GARCH e SVR recorrente. À direita, os gráficos mostram o logaritmo dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) também confrontando os resultados dos modelos GARCH e SVR recorrente referentes ao $\{e_t\}$ se comportando como uma normal padrão.

Capítulo 4

Aplicação em dados reais

Neste capítulo, será refeito o experimento da Seção 3.3, mas desta vez com dados reais. Os dados são compostos por dois tipos de variáveis financeiras, uma série temporal formada pela taxa de câmbio Libra/Dólar, e a outra série formada pelo índice médio da Bolsa de Valores de Nova Iorque (NYSE). Esta aplicação é inspirada na análise de dados reais feita por Chen *et al.* (2010), no entanto com o novo critério de parada discutido nas Seções 3.2 e 3.3.

Os dados referentes à taxa de câmbio consistem em uma série temporal do período de 5 de Janeiro de 2004 a 31 de Dezembro de 2007. O outro banco de dados é composto pelo preço diário de fechamento da NYSE, no período de 8 de Janeiro de 2004 a 31 de Dezembro de 2007. Esses períodos são iguais aos das séries utilizadas por Chen *et al.* (2010) na parte aplicada do artigo. Os dados da taxa de câmbio foram obtidos por meio da *Policy Analysis Computing and Information Facility in Commerce* (PACIFIC) na *University of British Columbia*. Já os dados da NYSE foram obtidos por meio do *Google Finance*.

É comum que séries temporais financeiras não apresentem estacionariedade, ou seja, que não se desenvolvam no tempo aleatoriamente ao redor de uma média constante (Morettin & Toloi, 2006). Para resolver esse problema, normalmente transforma-se a série em log-retornos, definidos por

$$r_t = \log\left(\frac{P_t}{P_{t-1}}\right) = \log(P_t) - \log(P_{t-1}), \quad (4.1)$$

em que $\{r_t\}_{t=1,\dots,T}$ é a série temporal de log-retornos, P_t e P_{t-1} são o preço do ativo financeiro no período t e no período $t - 1$, respectivamente, e $t = 1, \dots, T$ sendo T o tamanho da série dos log-retornos.

Uma vez feita a transformação nos dois bancos de dados, as séries temporais dos retornos $\{r_t\}$ ficaram com $T = 1001$ observações. Agora faz-se um ajuste por meio de um modelo GARCH para obtermos a média condicional dos resíduos e depois separamos $T1 = 940$ observações para serem o conjunto de dados de treinamento. Para obtermos a série temporal formada pelo quadrado dos resíduos das 1001 observações, as amostras de tamanho $T1$ são atualizadas passo a passo desde a observação no período $t = 940$ até chegarmos à observação do período $t = 1000$.

As Seções 4.1 e 4.2 apresentam os resultados das medidas de ajuste EQM e EMA e alguns dos gráficos dos ajustes encontrados a partir dos dados financeiros da taxa de câmbio Libra/Dólar, e do índice NYSE. O código utilizado nessas duas seções e os demais gráficos se encontram no Apêndice C, e o código em R utilizado nesta e na Seção 4.1, encontra-se no Apêndice E.

4.1 Resultados para a taxa de câmbio Libra/Dólar

No caso da série temporal de taxa de câmbio, encontramos resultados bastante parecidos com os de Chen *et al.* (2010), apesar de termos mudado o critério de parada. Neste caso utilizamos uma abordagem um pouco diferente da do capítulo anterior, pois todos os modelos GARCH foram testados para o mesmo banco de dados, ou seja, todos os modelos da Tabela 4.1 são comparáveis.

Em relação ao valor do EQM, os modelos GARCH tiveram um comportamento quase que igual, entretanto, o que melhor conseguiu prever os valores $\{u_t^2\}$ foi o GARCH em que $\{e_t\}$ se comporta como normal padrão. Já comparando os EMA,

o modelo SVR recorrente se saiu melhor, com $EMA=0,2353$ (Tabela 4.1). Percebe-se novamente o mesmo comportamento do capítulo anterior, em que o resultado do EMA foi ligeiramente favorável ao SVR recorrente, e o EQM aos modelos GARCH, entretanto todos os modelos tiveram desempenhos semelhantes.

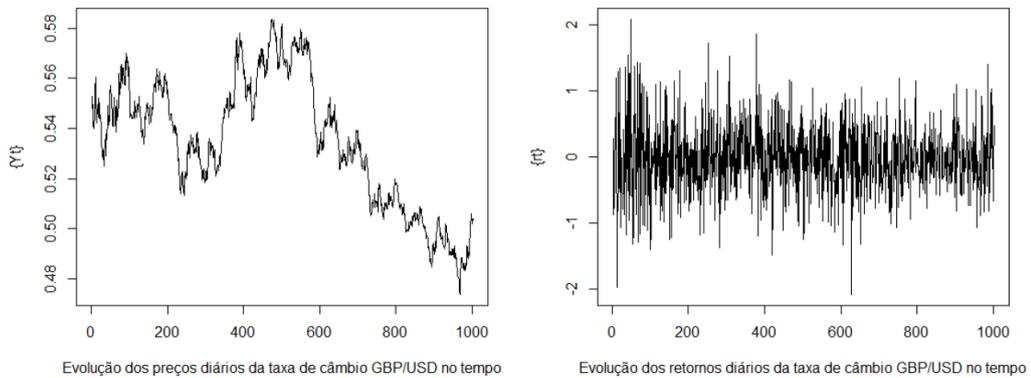


Figura 4.1: Evolução no tempo das séries temporais dos preços (à esquerda) e retornos (à direita) diários da taxa de câmbio Libra/Dólar.

Tabela 4.1: Valores dos EMAs e EQMs referentes aos ajustes dos modelos de SVR recorrente e GARCH. A sequência de erros $\{e_t\}$ é *iid* e se comporta de acordo com os 6 modelos GARCH testados. Nesta tabela observam-se os resultados referentes à série temporal da taxa de câmbio Libra/Dólar.

Modelo	EMA	EQM
SVR recorrente	0,2353	0,1710
GARCH normal	0,2496	0,1497
GARCH normal assimétrico	0,2494	0,1498
GARCH GED	0,2493	0,1499
GARCH GED assimétrico	0,2491	0,1499
GARCH t-Student	0,2496	0,1499
GARCH t-Student assimétrico	0,2494	0,1499

A Figura 4.1 ilustra a série temporal da taxa de câmbio à esquerda e a série de log-retornos à direita. E a Figura 4.2 nos mostra as previsões feitas pelos modelos SVR recorrente e pelo modelo GARCH com ruídos que se comportam segundo uma normal padrão. As figuras dos demais modelos se encontram no Apêndice C, e o código em R utilizado nesta e na Seção 4.1, encontra-se no Apêndice E.

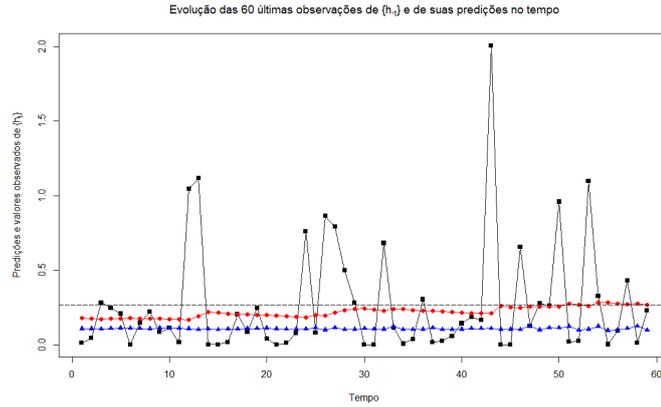


Figura 4.2: 60 últimas observações da série temporal de $\{u_t^2\}$ da taxa de câmbio Libra/Dólar (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma normal padrão (em linha vermelha com quadrados).

4.2 Resultados para o índice NYSE

No caso da série temporal do índice NYSE, encontramos algumas diferenças nos resultados das estimativas encontradas neste trabalho e no de Chen *et al.* (2010). Aqui, o modelo que melhor previu as observações da série temporal do índice NYSE em relação ao EMA foi o modelo GARCH com $\{e_t\}$ seguindo uma distribuição GED assimétrica. Em relação ao EQM, o melhor modelo foi o GARCH com $\{e_t\}$ seguindo uma distribuição t-Student. Portanto no caso do índice NYSE, o EMA não favoreceu o modelo SVR recorrente, mas ainda assim observamos bons resultados deste modelo.

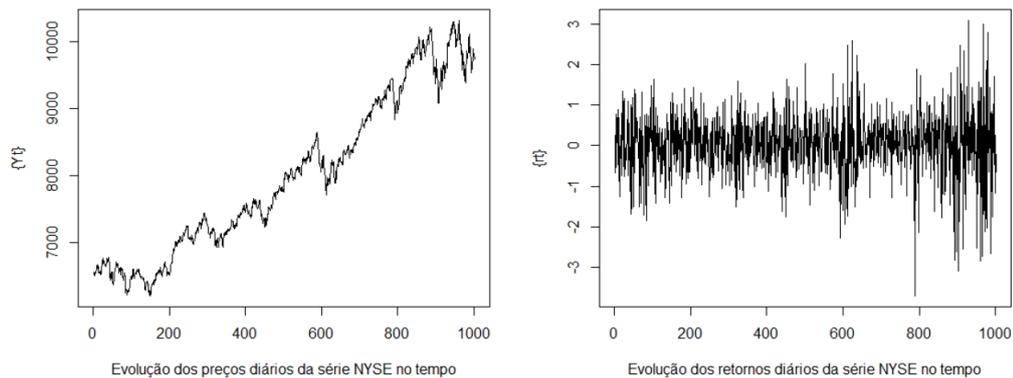


Figura 4.3: Evolução no tempo das séries temporais dos preços (à esquerda) e retornos (à direita) diários do índice NYSE.

A Figura 4.3 mostra à esquerda a série temporal do índice NYSE e à direita a série

Tabela 4.2: Valores dos EMAs e EQMs referentes aos ajustes dos modelos de SVR recorrente e GARCH. A sequência de erros $\{e_t\}$ é *iid* e se comporta de acordo com os 6 modelos GARCH testados. Nesta tabela observam-se os resultados referentes à série temporal do índice NYSE.

Modelo	EMA	EQM
SVR recorrente	1,6542	7,8689
GARCH normal	1,5843	5,6566
GARCH normal assimétrico	1,5873	5,6632
GARCH GED	1,5897	5,6337
GARCH GED assimétrico	1,5901	5,6519
GARCH t-Student	1,5968	5,6038
GARCH t-Student assimétrico	1,5959	5,6230

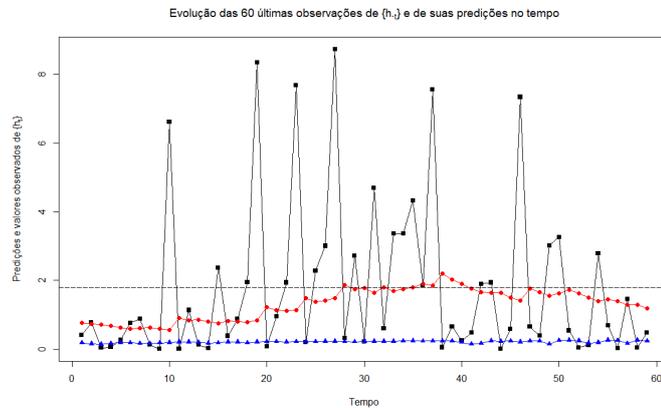


Figura 4.4: 60 últimas observações da série temporal de $\{u_t^2\}$ do índice NYSE (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma normal padrão (em linha vermelha com quadrados).

dos log-retornos desse índice. E a Figura 4.4 nos mostra as previsões feitas pelos modelos SVR recorrente e pelo modelo GARCH com ruídos que se comportam segundo uma normal padrão. As figuras dos demais modelos se encontram no Apêndice D.

Capítulo 5

Conclusões

Este trabalho propôs um estudo acerca do funcionamento das máquinas vetoriais tanto para fins de classificação (SVM) quanto para fins de regressão (SVR). Por causa do particular interesse em se considerar variáveis não observáveis para fins de previsões de séries temporais, Chen *et al.* (2010) inspiraram a formulação dos Capítulos 3 e 4.

O método proposto por esses autores se mostrou bastante adaptável e bom concorrente dos métodos tradicionais de análise de séries temporais. Como foi exemplificado na Seção 3.3 para dados simulados, e nas Seções 4.1 e 4.2 para dados reais de séries temporais financeiras. Entretanto se deve lembrar que esse método proporciona a previsão em dados reais da volatilidade definida pela Equação 3.2, enquanto os modelos GARCH são desenhados para prever a volatilidade $\{h_t\}$ definida pela Equação 3.4.

Apesar de os resultados dos modelos utilizados aqui terem sido satisfatórios, eles ainda podem ser melhorados, por exemplo, incorporando covariáveis não lineares ao modelo SVR recorrente, em vez de os deixarmos restritos à mesma estrutura do modelo autorregressivo. Além disso, novas funções *kernel* podem ser formuladas com vistas à melhoria da capacidade preditiva do SVM. Aqui o objetivo foi intencionalmente comparar um modelo AR-GARCH com seu correspondente não linear via SVM. Portanto, os resultados deste trabalho não esgotam a discussão entre o método SVM e os métodos estatísticos clássicos. O método SVM para classificação e regressão se

mostra muito eficaz e competitivo com diversos outros métodos, paramétricos ou não. Portanto, as pesquisas devem continuar para que sejam encontradas outras formas de comparação ou de mistura dos modelos GARCH com o SVM, ou deste com outras técnicas (computacionais e estatísticas), possibilitando melhorias na análise de volatilidade se visando obter maior capacidade preditiva.

Referências Bibliográficas

Akgiray, V. (1989). Conditional heteroscedasticity in time series of stock returns: Evidence and forecasts. *Journal of Business*, 55-80.

Anderson, J. L. (2001). An ensemble adjustment Kalman filter for data assimilation. *Monthly Weather Review*, **129**, 2884-2903.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, **68**, 337-404.

Azzalini, A. (1985). A class of distributions which includes the normal ones. *Scandinavian journal of statistics*, 171-178.

Barber, C. B., Dobkin, D. P. & Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, **22**, 469-483.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroscedasticity. *Journal of Econometrics*. **31**, 307-327.

Bollerslev, T. & Engle, R. F. (1993). Common persistence in conditional variances. *Econometrica*, 167-186.

Box, G. E. (1994). *Statistics for Experimenters. An Introduction to Design, Data Analysis and Model Building*. New York: Wiley.

Box, G. E. & Jenkins, G. M. (1970). *Time Series Models for Forecasting and Control*. San Francisco: Wiley.

Box, G. E. & Pierce, D. A. (1970). Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American Statistical Association*, **65**, 1509-1526.

- Brooks, C. (1998). Predicting stock index volatility: can market volume help?. *Journal of Forecasting*, **17**, 59-80.
- Brooks, C. & Persaud, G. (2003). Volatility forecasting for risk management. *Journal of Forecasting*, **22**, 1-22.
- Brown, K. Q. (1979). Voronoi diagrams from convex hulls. *Information Processing Letters*, **9**, 223-228.
- Chan, T. M. (1996). Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete & Computational Geometry*, **16**, 361–368.
- Chen, S., Jeong, K. & Härdle, W. K. (2008). Recurrent support vector regression for a nonlinear ARMA model with applications to forecasting financial returns. *Economic Risk, Berlin*, **649**, 1-29.
- Chen, S., Jeong, K. & Härdle, W. K. (2010). Forecasting volatility with support vector machine-based GARCH model. *Journal of Forecasting*, **29**, 406–433.
- Crisp, D. J. & Burges, C. J. C. (2000). A geometric interpretation of ν -SVM classifiers. *Advances in Neural Information Processing Systems*, 244-250.
- De Moivre, A. (1718). *The Doctrine of Chances: or, a Method for Calculating the Probabilities of Events in Play*. London: W. Pearson.
- Drucker, H., Wu, D. & Vapnik, V. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, **10**, 1048-1054.
- Elbisy, M. S. (2015). Sea Wave Parameters Prediction by Support Vector Machine Using a Genetic Algorithm. *Journal of Coastal Research*, **31**, 892-899.
- Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, **50**, 987-1007.
- Engle, R. F. & Ng, V. K. (1993). Measuring and testing the impact of news on volatility. *The Journal of Finance*, **48**, 1749-1778.

- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Human Genetics*, **7**, 87-193.
- Foster, W. R., Collopy, F. & Ungar, L. H. (1992). Neural network forecasting of short, noisy time series. *Computers and Chemical Engineering*, **16**, 293-297.
- Franses, P. H. & Van Dijk, D. (1996). Forecasting stock market volatility using (nonlinear) GARCH models. *Journal of Forecasting*, 229-235.
- Gale, D., Kuhn, H. W. & Tucker, A. W. (1951). Linear programming and the theory of games. *Activity Analysis of Production and Allocation*, **13**, 317-335.
- Gerlach, R. & Tuyl, F. (2006). MCMC methods for comparing stochastic volatility and GARCH models. *International Journal of Forecasting*, **22**, 91-107.
- Giller, G. L. (2005). A generalized error distribution. *Guiller Investments Research Note: 20031222*, **1**, 1-7.
- Gokcan, S. (2000). Forecasting volatility of emerging stock markets: linear versus non-linear GARCH models. *Journal of Forecasting*, **19**, 499-504.
- Gunn, S. *Support Vector Machines for Classification and Regression*. Southampton: University of Southampton, 1998. Accessed September 11, 2015.
- Hamilton, J. D. (1997). Measuring the liquidity effect. *The American Economic Review*, 80-97.
- Hill, R. C., Griffiths, W. E. & Lim, G. C. (2012). *Principles of Econometrics*. New York: Wiley.
- Houssain, A. & Nasser, M. (2011). Recurrent support and relevance vector machines based model with application to forecasting volatility of financial returns. *Journal of Intelligent Learning Systems and Applications*, **3**, 230-241.
- Hung, C., Hung, C. & Lin, S. (2014). Predicting time series using integration of moving average and support vector regression. *International Journal of Machine Learning and Computing*, **4**, 491-495.

- Hwang, H. S., Ham, H. D. & Kim, H. J. (2012). Forecasting performance of LS-SVM for nonlinear hydrological time series. *KSCE Journal of Civil Engineering*, **16**, 870-882.
- James, G., Witten, D., Hastie, T. & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. New York: Springer.
- Karush, W. (1939). *Minima of Functions of Several Variables With Inequalities as Side Conditions*. Master thesis, University of Chicago, Chicago.
- Klein, P. A. & Niemira, M. P. (1994). *Forecasting Financial and Economic Cycles. Finance Editions*. New York: Wiley.
- Kökner-Tezel, S. & Latecki, L. J. (2011). Improving SVM classification on imbalanced time series sets with ghost points. *Knowledge and Information Systems*, **28**, 1-23.
- Kuhn, H. W. & Tucker, A. W. (1951). *Nonlinear Programming*. Proceedings of 2nd Berkeley Symposium. Berkeley: University of California Press, 481-492.
- Ljung, G. M. & Box, G. E. (1978). On a measure of lack of fit in time series models. *Biometrika*, 297-303.
- Lorena, A. C. & de Carvalho, A. C. (2007). Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada*, **14**, 43-67.
- Metropolis, N. & Ulam, S. (1949). The monte carlo method. *Journal of the American Statistical Association*, **44**, 335-341.
- Morettin, P. A. & Toloi, C. (2006). *Análise de Séries Temporais*. São Paulo: Editora Edgard Blucher.
- Nikolaev, N. Y., Boshnakov, G. N. & Zimmer, R. (2013). Heavy-tailed mixture GARCH volatility modeling and value-at-risk estimation. *Expert Systems with Applications*, **40**, 2233-2243.
- Niu, D. & Wei, Y. (2013). An improved short-term power load combined forecasting with ARMA-GARCH-ANN-SVM based on FHNN similar-day clustering. *Journal of Software*, **8**, 716-723.

- Poon, S. H. & Granger, C. W. (2003). Forecasting volatility in financial markets: A review. *Journal of Economic Literature*, **41**, 478-539.
- Preparata, F. P. & Hong, S. J. (1977). Convex hulls of finite sets of points in two and three dimensions. *Communications of the ACM*, **20**, 87-93.
- Rosenblatt, F. *The Perceptron: A Perceiving and Recognizing Automaton*. Buffalo: Cornell Aeronautical Laboratory, INC., 1957. Accessed September 11, 2015.
- Rubio, G., Pomares, H., Rojas, I. & Herrera, L. (2011). A heuristic method for parameter selection in LS-SVM: application to time series prediction. *International Journal of Forecasting*, **27**, 725-739.
- Sharifi, M. A. & Souri, A. H. (2015). A hybrid LS-HE and LS-SVM to predict time series of precipitable water vapor derived from GPS measurements. *Arabian Journal of Geosciences*, **8**, 7257-7272.
- Smola, A. J. & Schölkopf, B. (2004). A tutorial on support vector regression. **Statistics and Computing**, **14**, 199-222.
- Student. (1908). The probable error of a mean. *Biometrika*, **6**, 1-25.
- Tay, F. E. H. & Cao, L. (2001). Application of support vector machines in financial time series forecasting. *Omega*, **29**, 309-317.
- Theodossiou, P. (1998). Financial data and the skewed generalized t distribution. **Management Science**. **44**, 1650–1661.
- Theodossiou, P. (2015). Skewed generalized error distribution of financial assets and option pricing. *Multinational Finance Journal*, **19**, 223-266.
- Vapnik, V. (1982). *Estimation of Dependences Based on Empirical Data*. New York: Springer-Verlag.
- Vapnik, V.(1995). *The Nature of Statistical Learning Theory*. New York: Springer-Verlag.

- Vapnik, V. & Chervonenkis, A. Y. A. (1964). On one class of perceptrons. *Avtomat. i Telemekh.*, **25**, 112-120.
- Vapnik, V. & Lerner, A. (1963). Pattern recognition using generalized portrait method. *Automaton and Remote Control*, **24**, 774-780.
- Wei, W. W. S. (2006). *Time Series Analysis: Univariate and Multivariate Methods*. Boston: Pearson Addison-Wesley.
- Weiqing, X., Fan, Z., Cai, M., Shi, Y., Tong, X. & Sun, J. (2015). Soft sensing method of LS-SVM using temperature time series for gas flow measurements. *Metrology and Measurement Systems*, **22**, 383-392.
- Whittle, P. (1951). *Hypothesis Testing in Times Series Analysis*. Uppsala: Uppsala University & Wiksells boktr.
- Xie, X. & Hu, G. (2007). A comparison of shanghai housing price index forecasting. *IEEE Computer Society, ICNC '07 Proceedings of the Third International Conference on Natural Computation - Volume 03*, 221-225.
- Yao, A. C. C. (1981). A lower bound to finding convex hulls. *Journal of the ACM*, **28**, 780-787.
- Zhao, X., Ma, Z. & Yin, M. (2012). Using support vector machine and evolutionary profiles to predict antifreeze protein sequences. *International Journal of Molecular Sciences*, **13**, 2196-2207.

Apêndice A

Este apêndice contém os gráficos da Seção 3.3.

A.1 Gráficos da Seção 3.3

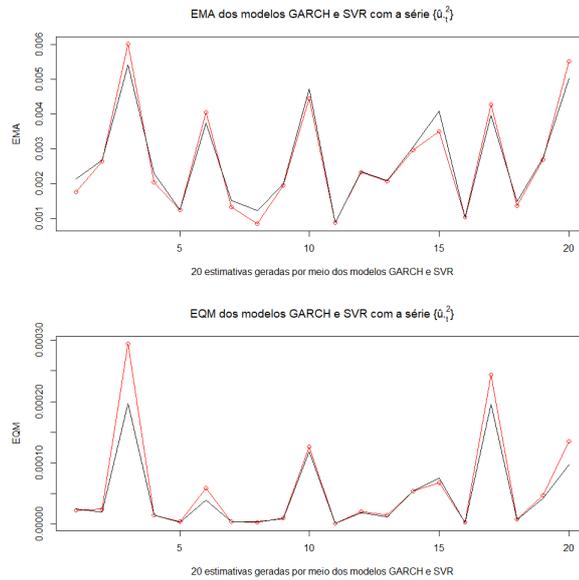


Figura A.1: Os gráficos mostram 20 resultados dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) dos modelos GARCH e SVR recorrente, ambos supondo $\{e_t\}$ se comportando como uma normal assimétrica.

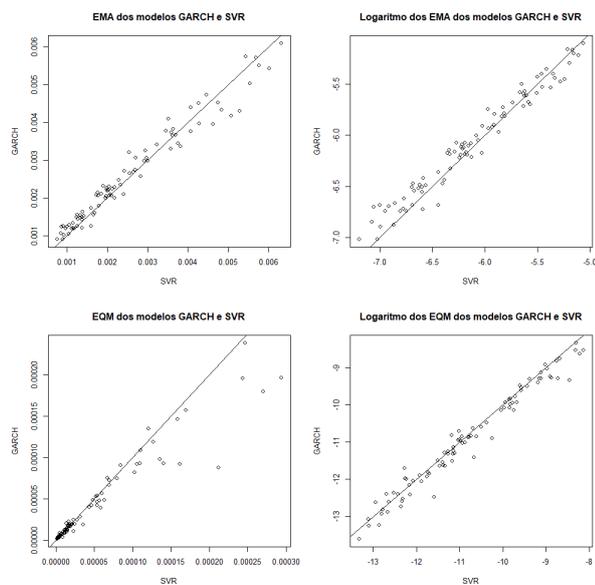


Figura A.2: À esquerda, os gráficos mostram os EMA (gráfico de cima) e os EQM (gráfico de baixo) confrontando os resultados dos modelos GARCH e SVR recorrente. À direita, os gráficos mostram o logaritmo dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) também confrontando os resultados dos modelos GARCH e SVR recorrente referentes ao $\{e_t\}$ se comportando como uma normal padrão.

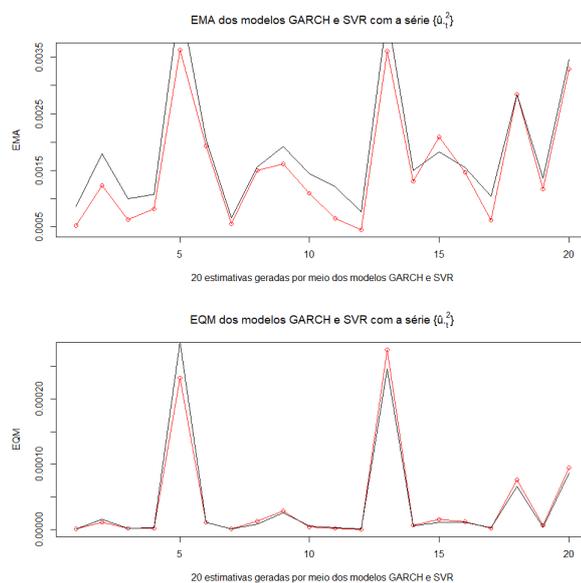


Figura A.3: Os gráficos mostram 20 resultados dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) dos modelos GARCH e SVR recorrente, ambos supondo $\{e_t\}$ se comportando como uma t-Student padrão.

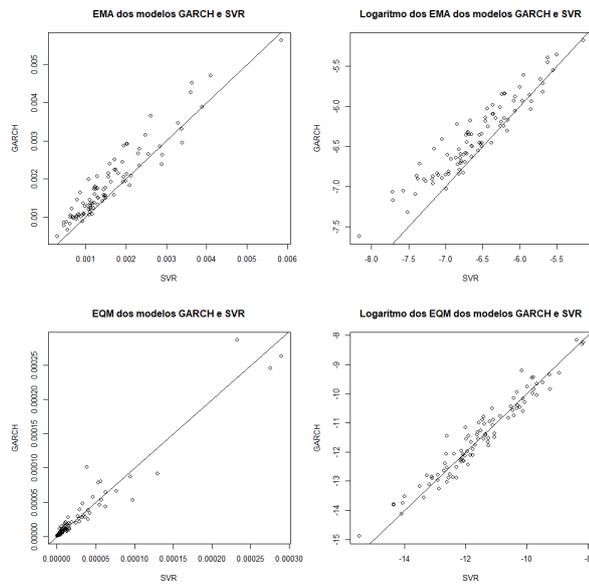


Figura A.4: À esquerda, os gráficos mostram os EMA (gráfico de cima) e os EQM (gráfico de baixo) confrontando os resultados dos modelos GARCH e SVR recorrente. À direita, os gráficos mostram o logaritmo dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) também confrontando os resultados dos modelos GARCH e SVR recorrente referentes ao $\{e_t\}$ se comportando como uma t-Student padrão.

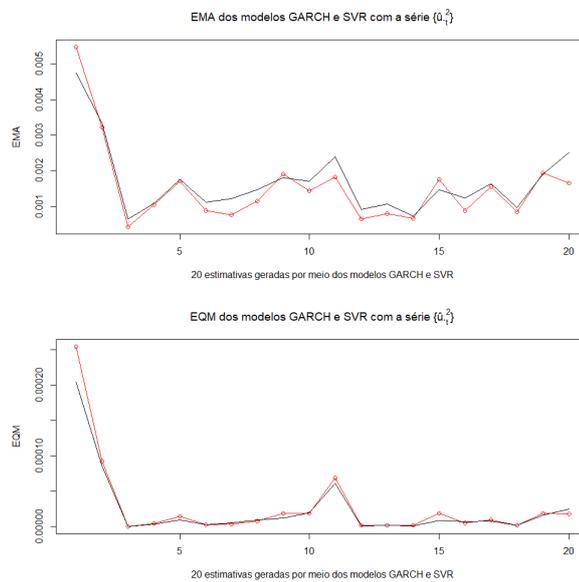


Figura A.5: Os gráficos mostram 20 resultados dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) dos modelos GARCH e SVR recorrente, ambos supondo $\{e_t\}$ se comportando como uma t-Student assimétrica.

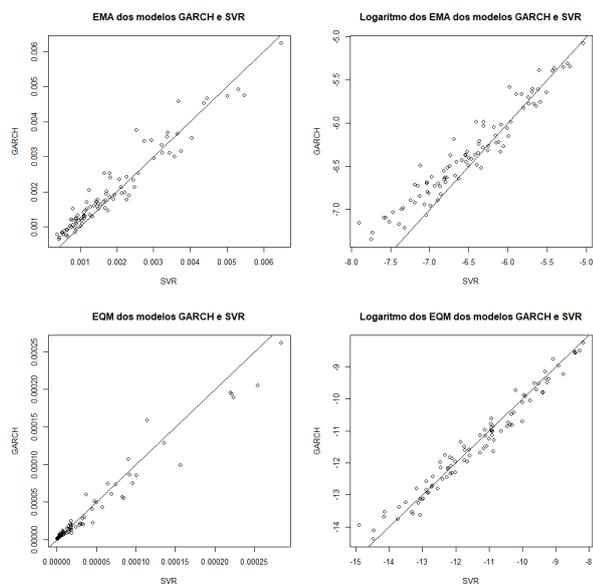


Figura A.6: À esquerda, os gráficos mostram os EMA (gráfico de cima) e os EQM (gráfico de baixo) confrontando os resultados dos modelos GARCH e SVR recorrente. À direita, os gráficos mostram o logaritmo dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) também confrontando os resultados dos modelos GARCH e SVR recorrente referentes ao $\{e_t\}$ se comportando como uma t-Student assimétrica.

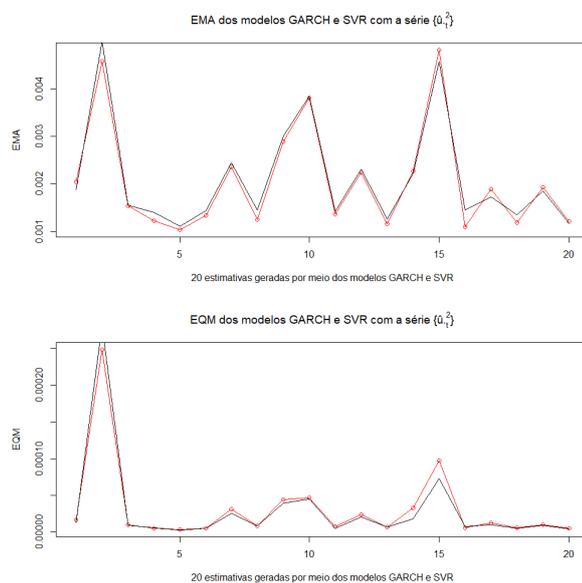


Figura A.7: Os gráficos mostram 20 resultados dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) dos modelos GARCH e SVR recorrente, ambos supondo $\{e_t\}$ se comportando como uma GED.

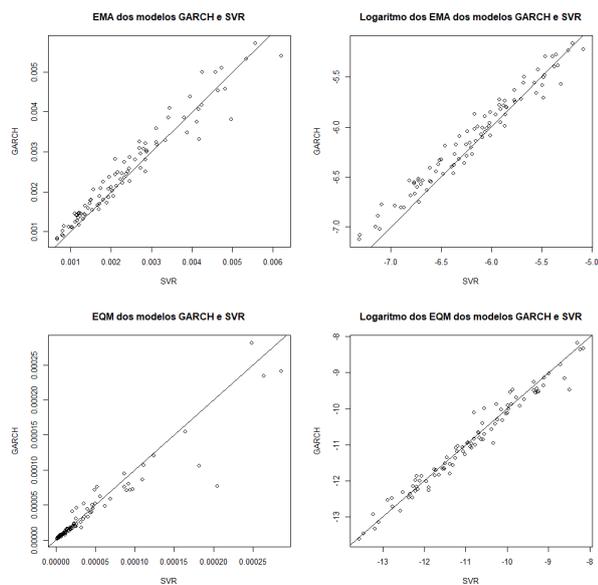


Figura A.8: À esquerda, os gráficos mostram os EMA (gráfico de cima) e os EQM (gráfico de baixo) confrontando os resultados dos modelos GARCH e SVR recorrente. À direita, os gráficos mostram o logaritmo dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) também confrontando os resultados dos modelos GARCH e SVR recorrente referentes ao $\{e_t\}$ se comportando como uma GED.

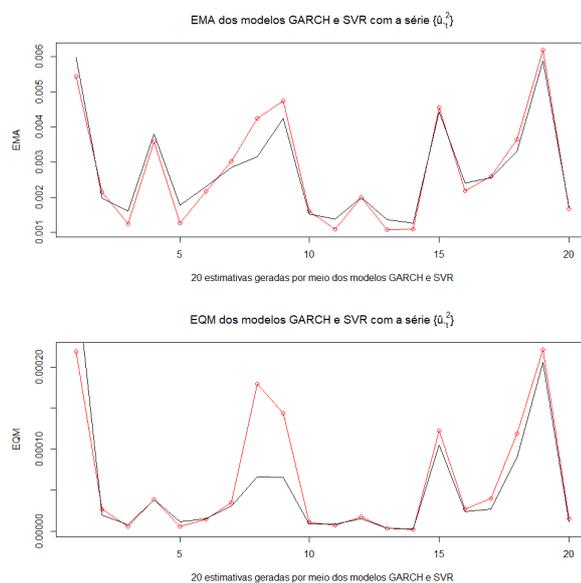


Figura A.9: Os gráficos mostram 20 resultados dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) dos modelos GARCH e SVR recorrente, ambos supondo $\{e_t\}$ se comportando como uma GED assimétrica.

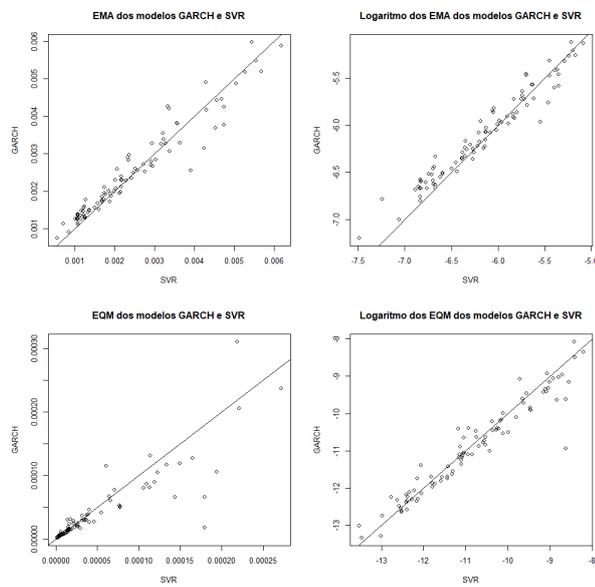


Figura A.10: À esquerda, os gráficos mostram os EMA (gráfico de cima) e os EQM (gráfico de baixo) confrontando os resultados dos modelos GARCH e SVR recorrente. À direita, os gráficos mostram o logaritmo dos EMA (gráfico de cima) e dos EQM (gráfico de baixo) também confrontando os resultados dos modelos GARCH e SVR recorrente referentes ao $\{e_t\}$ se comportando como uma GED assimétrica.

Apêndice B

Neste apêndice será apresentado o código do *software* R utilizado na Seção 3.3 para obter seus resultados.

B.1 Código da Seção 3.3

```
library(quantmod)
library(e1071)
library(fGarch)

# cond.dist = c("norm", "ged", "std", "snorm", "sged", "sstd")
# N.dados = tamanho da série
# N.teste = tamanho da série para teste
# N.trein = tamanho da série para treinamento

SimulSvmGarch <- function(N.dados,N.teste,CondDist="norm")
{
    # inicio da funcao SimulSvmGarch
    # SimulSvmGarch: escrito com base no Experimento 004.

    # =====
    # 1. Especificar o modelo
    # =====
    argarch <- garchSpec(model=list(ar=0.5,omega=0.0005,
                                   alpha=0.8,beta=0.1),cond.dist=CondDist)
    Y.t      <- garchSim(spec=argarch,n=N.dados,extended=TRUE)
    y.t      <- Y.t$garch
    h.t      <- (Y.t$sigma)^2
    e.t      <- Y.t$eps
    # =====
    # 2. Estimar os residuos
```

```

# =====
x.t  <- y.t[2:N.dados]
x.t1 <- y.t[1:(N.dados-1)]
input <- cbind(x.t, x.t1)
#-----
svm_eps  <- 0.0001
svm_cost <- 0.05
sigma2   <- 0.4
svm_gamma <- 1/(2*sigma2)
#-----
# =====
# 3. Reservar 940 primeiras observações para treinamento
#     e as 60 últimas para teste;
# =====
N.trein = N.dados - N.teste - 1

svm.0 <- svm(x=input[1:N.trein,-1],y=input[1:N.trein,1],
             scale=FALSE,type="eps-regression",kernel="radial",
             gamma=svm_gamma,cost=svm_cost,epsilon=svm_eps)
u.t  <- svm.0$residuals
u2.t <- u.t^2
# =====
# 4. Passo 1;
# =====
input.trein <- cbind(u2.t[-1], u2.t[-length(u2.t)])
#-----
svm_eps  <- 0.00005
svm_cost <- 10
sigma2   <- 0.01
svm_gamma <- 1/(2*sigma2)
#-----
epoch <- 1
svm.1 <- svm(x=input.trein[,-1],y=input.trein[,1],
             scale=FALSE,type="eps-regression",
             kernel="radial",gamma=svm_gamma,cost=svm_cost,
             epsilon=svm_eps)
w.t  <- svm.1$residuals
# =====

```

```

# 5. Passos 2 a 4;
# =====
crit2.values <- NULL
criterio2    <- 0

while(criterio2 < 0.95 & epoch < 100)
{
input.trein  <- cbind(input.trein[,1:2],c(0, w.t[-length(w.t)]))
svm.1       <- svm(x=input.trein[,-1],y=input.trein[,1],
                  scale=FALSE,type="eps-regression",
                  kernel="radial",gamma=svm_gamma,
                  cost=svm_cost,epsilon=svm_eps)

w.t_old     <- w.t
w.t         <- svm.1$residuals
criterio2   <- cor(w.t,w.t_old)
crit2.values[epoch] <- criterio2
epoch      <- epoch + 1
}

# gravar ++++++
epoch.trein <- epoch
# ++++++
input.trein <- cbind(input.trein[,1:2],c(0, w.t[-length(w.t)]))
# =====
# 6. predizer passo a passo
#   (com atualização por SVM recorrente);
# =====

w.teste     <- NULL
u2.pred     <- NULL
u2.est      <- NULL
j           <- 1
svm.2       <- svm.1
w.tt       <- w.t
input.teste <- input.trein
loops      <- NULL

for (tau in (N.trein+2):(N.dados-1))
{
y.new      <- predict(svm.0, newdata=y.t[tau-1],

```

```

                                scale=FALSE)
u2.t[tau-1]      <- (y.t[tau]-y.new)^2
# reestimar svm.0
#-----
svm_eps   <- 0.0001
svm_cost  <- 0.05
sigma2    <- 0.4
svm_gamma <- 1/(2*sigma2)
#-----
svm.0      <- svm(x=input[1:(tau-1),-1],
                  y=input[1:(tau-1),1],
                  scale = FALSE,type="eps-regression",
                  kernel = "radial",gamma = svm_gamma,
                  cost=svm_cost,epsilon=svm_eps)
input.pred <- cbind(u2.t[tau-1],last(w.tt))
u2.pred[j] <- abs(predict(svm.2,newdata=input.pred,
                          scale=FALSE))

u2.t[tau]   <- last(svm.0$residuals^2)
u2.est[j]   <- u2.t[tau]
w.teste[j]  <- u2.t[tau]-u2.pred[j]
input.new   <- cbind(u2.t[tau],input.pred)
input.teste <- rbind(input.teste,input.new)
w.t_old     <- c(w.tt,w.teste[j])
#-----
svm_eps   <- 0.00005
svm_cost  <- 10
sigma2    <- 0.01
svm_gamma <- 1/(2*sigma2)
#-----
criterio2 <- 0
loop <- 1
while(criterio2 < 0.95 & loop < 100)
{
rownames(input.teste)<-NULL
svm.2      <-svm(x=input.teste[,-1],y=input.teste[,1],
                 scale=FALSE,type="eps-regression",
                 kernel="radial",gamma=svm_gamma,
                 cost=svm_cost,epsilon = svm_eps)

```

```

w.tt      <- svm.2$residuals
criterio2 <- cor(w.tt,w.t_old)
loop      <- loop + 1
input.teste <- cbind(input.teste[,1:2],
                     c(0,w.tt[-length(w.tt)]))
w.t_old   <- w.tt
}
loops[j]  <-loop
j         <- j + 1
}
# gravar ++++++
epoch.tst.max <- max(loops)
# ++++++
# =====
# 7. comparações com o modelo ARMA-GARCH;
# =====
h.predict   <- NULL
h.estimated <- NULL
j          <- 1
k          <- 2
for (tau in (N.trein+3):(N.dados+1))
#           +3 porque refere-se a y.t
{
fit.garch = garchFit(~ arma(1,0)+garch(1,1),
                    data=y.t[1:(tau-1)],trace=FALSE,
                    cond.dist=CondDist)
u2.last    <- last((fit.garch$residuals)^2)
h.estimated[j]<- u2.last
h.last     <- last(fit.garch@h.t)
alpha_hat  <- coef(fit.garch)[4]
beta_hat   <- coef(fit.garch)[5]
omega_hat  <- coef(fit.garch)[3]
h.predict[j] <- omega_hat + beta_hat*h.last + alpha_hat*u2.last
j          <- j + 1
}
h.estimated <- h.estimated[-length(h.estimated)]
h.predict   <- h.predict[-length(h.predict)]
# =====

```

```

h.proxy =(y.t-mean(y.t))^2
Bias    <- mean((h.t[(N.dados-N.teste+2):(N.dados)]
               -h.proxy[(N.dados-N.teste+2):(N.dados)])^2)
Var.ht  <- var(h.t[(N.dados-N.teste+2):(N.dados)])
Var.hproxy <- var(h.proxy[(N.dados-N.teste+2):(N.dados)])
MAE.GARCH <- mean(abs(h.predict-h.t[(N.dados-N.teste+2):
                               (N.dados)]))
MAE.SVM   <- mean(abs(u2.pred-h.t[(N.dados-N.teste+2):
                               (N.dados)]))
MSE.GARCH <- mean((h.predict-h.t[(N.dados-N.teste+2):
                               (N.dados)])^2)
MSE.SVM   <- mean((u2.pred-h.t[(N.dados-N.teste+2):
                               (N.dados)])^2)
MAE.GARCH.proxy <- mean(abs(h.predict-h.proxy[(N.dados-N.teste+2):
                                             (N.dados)]))
MAE.SVM.proxy   <- mean(abs(u2.pred-h.proxy[(N.dados-N.teste+2):
                                             (N.dados)]))
MSE.GARCH.proxy <- mean((h.predict-h.proxy[(N.dados-N.teste+2):
                                             (N.dados)])^2)
MSE.SVM.proxy   <- mean((u2.pred-h.proxy[(N.dados-N.teste+2):
                                             (N.dados)])^2)

resultados <- cbind(epoch.trein,epoch.tst.max,Bias,Var.ht,
                    Var.hproxy,MAE.GARCH,MAE.SVM,MSE.GARCH,
                    MSE.SVM,MAE.GARCH.proxy,MAE.SVM.proxy,
                    MSE.GARCH.proxy,MSE.SVM.proxy)
return(resultados)
}
# fim da funcao SimulSvmGarch

# =====

t.ini <-Sys.time()
ResSVMGarchSim<-SimulSvmGarch(1000,60,"norm")
difftime(Sys.time(),t.ini)
for (replic in 1:100)
{
  SSG<-SimulSvmGarch(1000,60,"norm")
  ResSVMGarchSim <- rbind(ResSVMGarchSim,SSG)
}

```

```
    difftime(Sys.time(),t.ini)
}
print(paste("duracao total: "))
difftime(Sys.time(),t.ini)
Stats = data.frame(ResSVMGarchSim)
write.table(Stats,file="C:\\Users\\X\\Documents\\simulation.csv")
```

Apêndice C

Neste apêndice serão expostos os gráficos das Seção 4.1.

C.1 Gráficos da Seção 4.1

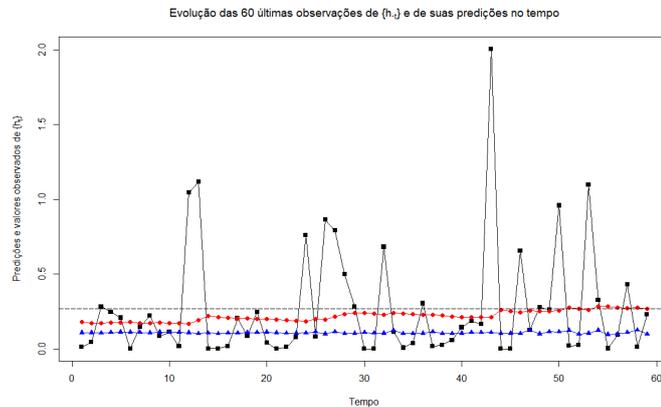


Figura C.1: 60 últimas observações da série temporal de $\{u_t^2\}$ da taxa de câmbio Libra/Dólar (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma normal assimétrica (em linha vermelha com quadrados).

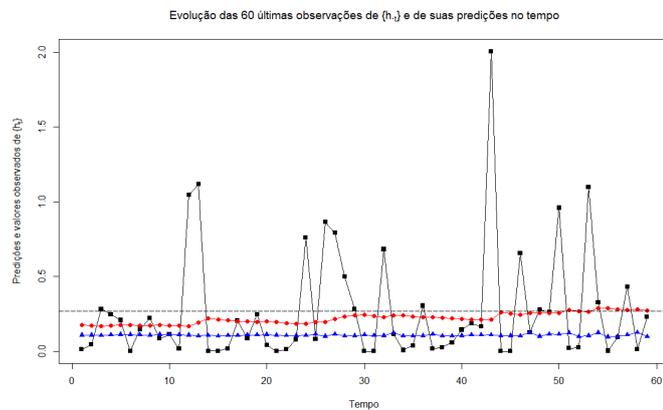


Figura C.2: 60 últimas observações da série temporal de $\{u_t^2\}$ da taxa de câmbio Libra/Dólar (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma t-Student (em linha vermelha com quadrados).

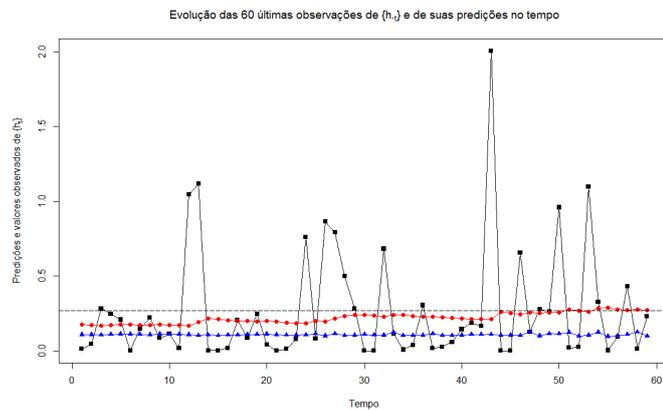


Figura C.3: 60 últimas observações da série temporal de $\{u_t^2\}$ da taxa de câmbio Libra/Dólar (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma t-Student assimétrica (em linha vermelha com quadrados).

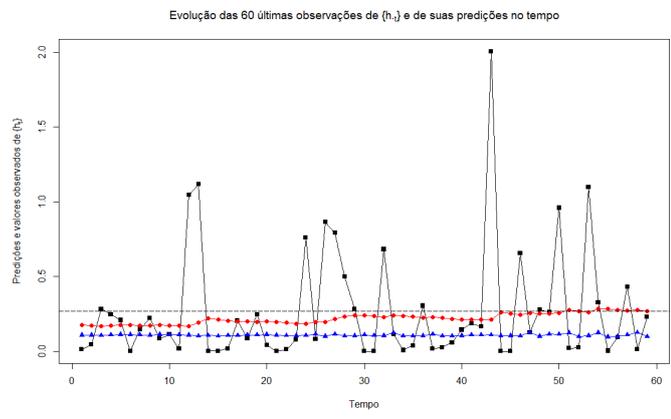


Figura C.4: 60 últimas observações da série temporal de $\{u_t^2\}$ da taxa de câmbio Libra/Dólar (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma GED (em linha vermelha com quadrados).

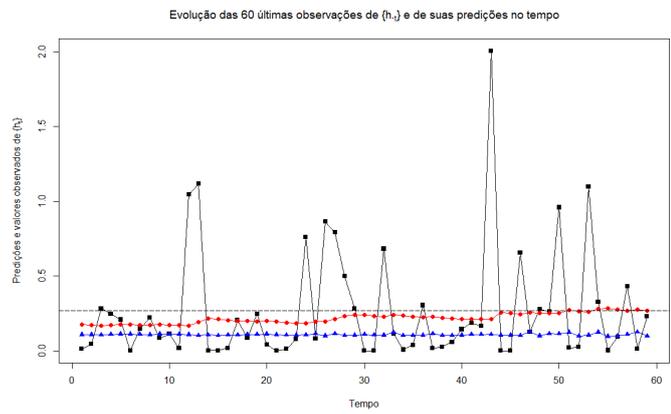


Figura C.5: 60 últimas observações da série temporal de $\{u_t^2\}$ da taxa de câmbio Libra/Dólar (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma GED padrão (em linha vermelha com quadrados).

Apêndice D

Neste apêndice serão expostos os gráficos das Seção 4.2.

D.1 Gráficos da Seção 4.2

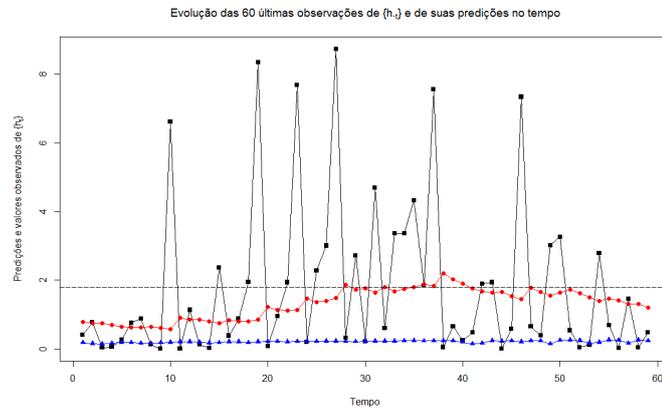


Figura D.1: 60 últimas observações da série temporal de $\{u_t^2\}$ do índice NYSE (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma normal assimétrica (em linha vermelha com quadrados).

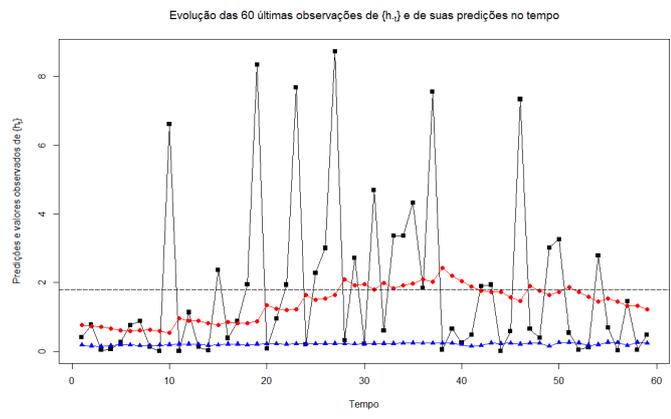


Figura D.2: 60 últimas observações da série temporal de $\{u_t^2\}$ do índice NYSE (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma t-Student (em linha vermelha com quadrados).

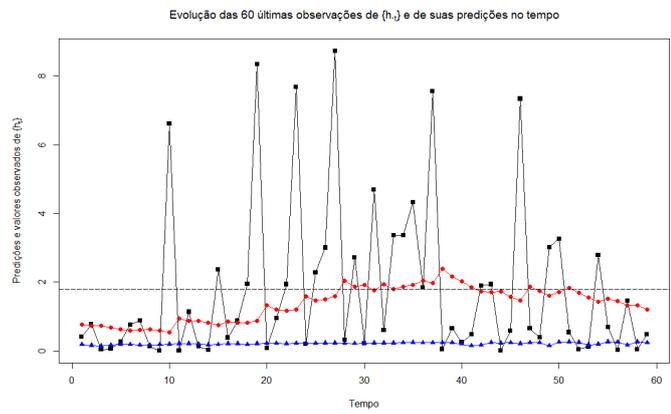


Figura D.3: 60 últimas observações da série temporal de $\{u_t^2\}$ do índice NYSE (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma t-Student assimétrica (em linha vermelha com quadrados).

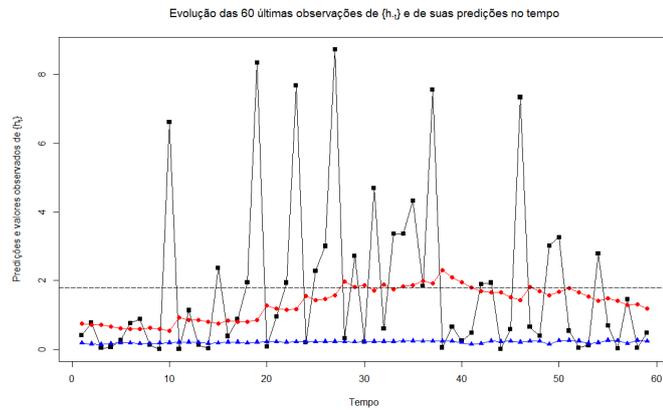


Figura D.4: 60 últimas observações da série temporal de $\{u_t^2\}$ do índice NYSE (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma GED (em linha vermelha com quadrados).

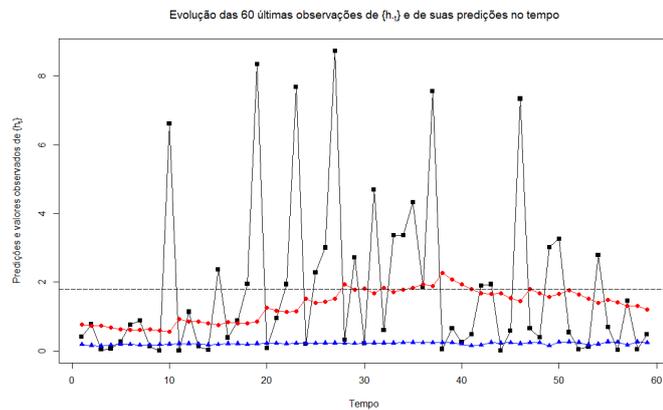


Figura D.5: 60 últimas observações da série temporal de $\{u_t^2\}$ do índice NYSE (em linha preta com pontos), e suas previsões passo a passo para o modelo SVR recorrente (em linha azul com triângulos) e o modelo GARCH com ruídos que se comportam segundo uma GED padrão (em linha vermelha com quadrados).

Apêndice E

Neste apêndice será exposto o código utilizado no *software* R para chegar aos resultados das Seções 4.1 e 4.2.

E.1 Código das Seções 4.1 e 4.2

```
library(quantmod)
library(e1071)
library(fGarch)
library(moments)

#cond.dist = c("norm", "ged", "std", "snorm", "sged", "sstd")

CondDist="norm"

# =====
# 1. Leitura dos dados;
# =====

dadosr <-read.table(file="C:\\Users\\X\\Documents\\DADOS_REAIS.txt",
                    header=TRUE,sep=",",dec=".")
Y.t    <- as.ts(dadosr)
y.t    <- diff(log(Y.t))*100

mean(y.t)
var(y.t)
skewness(y.t)
kurtosis(y.t)

# N.dados = tamanho da série
```

```

# N.teste = tamanho da série para teste
# N.trein = tamanho da série para treinamento

N.dados <- length(y.t)
N.teste <- 60
# =====
# 2. Estimar os resíduos
# =====
x.t <- y.t[2:N.dados]
x.t1 <- y.t[1:(N.dados-1)]
input <- cbind(x.t, x.t1)
#-----
svm_eps <- 0.05
svm_cost <- 0.005
sigma2 <- 0.2
svm_gamma <- 1/(2*sigma2)
#-----
# =====
# 3. Reservar 940 primeiras observações para treinamento
# e as 60 últimas para teste;
# =====
N.trein = N.dados - N.teste - 1
svm.0 <- svm(x=input[1:N.trein,-1],y=input[1:N.trein,1],
             scale=FALSE,type="eps-regression",kernel="radial",
             gamma=svm_gamma,cost=svm_cost,epsilon=svm_eps)
u.t <- svm.0$residuals
u2.t <- u.t^2
# =====
# 4. Passo 1;
# =====
input.trein <- cbind(u2.t[-1], u2.t[-length(u2.t)])
#-----
svm_eps <- 0.05
svm_cost <- 0.005
sigma2 <- 0.2
svm_gamma <- 1/(2*sigma2)
#-----
epoch <- 1

```

```

svm.1 <- svm(x=input.trein[,-1],y=input.trein[,1],scale=FALSE,
            type="eps-regression",kernel="radial",
            gamma=svm_gamma,cost=svm_cost,epsilon=svm_eps)
w.t    <- svm.1$residuals
# =====
# 5. Passos 2 a 4;
# =====
crit2.values <- NULL
criterio2    <- 0

while(criterio2 < 0.98 & epoch < 100)
{
input.trein <- cbind(input.trein[,1:2],c(0, w.t[-length(w.t)]))
svm.1      <- svm(x=input.trein[,-1],y=input.trein[,1],
                scale=FALSE,type="eps-regression",
                kernel="radial",gamma=svm_gamma,
                cost=svm_cost,epsilon=svm_eps)

w.t_old          <- w.t
w.t              <- svm.1$residuals
criterio2        <- cor(w.t,w.t_old)
crit2.values[epoch] <- criterio2
epoch            <- epoch + 1
}
# gravar ++++++
epoch.trein <- epoch
# ++++++

input.trein <- cbind(input.trein[,1:2],c(0, w.t[-length(w.t)]))
# =====
# 6. predizer passo a passo
# (com atualização por SVM recorrente);
# =====
w.teste    <- NULL
u2.pred    <- NULL
u2.est     <- NULL
j          <- 1
svm.2      <- svm.1
w.tt       <- w.t

```

```

input.teste <- input.trein
loops      <- NULL
for (tau in (N.trein+2):(N.dados-1))
{
y.new      <- predict(svm.0,newdata=y.t[tau-1],scale=FALSE)
u2.t[tau-1] <- (y.t[tau]-y.new)^2
# reestimar svm.0
#-----
svm_eps    <- 0.05
svm_cost   <- 0.005
sigma2     <- 0.2
svm_gamma  <- 1/(2*sigma2)
#-----
svm.0      <- svm(x=input[1:(tau-1),-1],y=input[1:(tau-1),1],
                 scale=FALSE,type="eps-regression",
                 kernel="radial",gamma=svm_gamma,
                 cost=svm_cost,epsilon=svm_eps)
input.pred <- cbind(u2.t[tau-1],last(w.tt))
u2.pred[j] <- abs(predict(svm.2,
                        newdata=input.pred,
                        scale=FALSE))
u2.t[tau]  <- last(svm.0$residuals^2)
u2.est[j]  <- u2.t[tau]
w.teste[j] <- u2.t[tau]-u2.pred[j]
input.new  <- cbind(u2.t[tau],input.pred)
input.teste <- rbind(input.teste,input.new)
w.t_old    <- c(w.tt,w.teste[j])
#-----
svm_eps    <- 0.05
svm_cost   <- 0.005
sigma2     <- 0.2
svm_gamma  <- 1/(2*sigma2)
#-----
criterio2 <- 0
loop <- 1
while(criterio2 < 0.98 & loop < 100)
{
rownames(input.teste)<-NULL

```

```

svm.2<-svm(x=input.teste[,-1],y=input.teste[,1],scale=FALSE,
           type="eps-regression",kernel="radial",
           gamma=svm_gamma,cost=svm_cost,epsilon=svm_eps)
w.tt      <- svm.2$residuals
criterio2 <- cor(w.tt,w.t_old)
loop <- loop + 1
input.teste <- cbind(input.teste[,1:2],c(0,w.tt[-length(w.tt)]))
w.t_old    <- w.tt
}
loops[j]   <-loop
j          <- j + 1
}
# gravar ++++++
epoch.tst.max <- max(loops)
# ++++++
# =====
# 7. comparações com o modelo ARMA-GARCH;
# =====
h.predict    <- NULL
j           <- 1
k           <- 2
for (tau in (N.trein+3):(N.dados+1))
#           +3 porque refere-se a y.t
{
fit.garch = garchFit(~ arma(1,0)+ garch(1,1),
                    data=y.t[1:(tau-1)],
                    trace=FALSE,
                    cond.dist=CondDist)
u2.last     <- last((fit.garch$residuals)^2)
h.last      <- last(fit.garch@h.t)
alpha_hat   <- coef(fit.garch)[4]
beta_hat    <- coef(fit.garch)[5]
omega_hat   <- coef(fit.garch)[3]
h.predict[j] <- omega_hat + beta_hat*h.last + alpha_hat*u2.last
j          <- j + 1
}
h.predict    <- h.predict[-length(h.predict)]
# =====

```

```
h.proxy =(y.t-mean(y.t))^2
MAE.SVM.proxy  <- mean(abs(u2.pred-h.proxy[(N.dados-N.teste+2):
                                           (N.dados)]))
MAE.GARCH.proxy <- mean(abs(h.predict-h.proxy[(N.dados-N.teste+2):
                                               (N.dados)]))
MSE.SVM.proxy  <- mean((u2.pred-h.proxy[(N.dados-N.teste+2):
                                         (N.dados)])^2)
MSE.GARCH.proxy <- mean((h.predict-h.proxy[(N.dados-N.teste+2):
                                           (N.dados)])^2)
```