LUIZ PAULO RODRIGUES DE FREITAS PARREIRAS

ARBITRAGEM ESTATÍSTICA E INTELIGÊNCIA ARTIFICIAL

Dissertação de Mestrado apresentada à Universidade de São Paulo para obtenção do título de Mestre em Modelagem Matemática em Finanças.

São Paulo
2007

LUIZ PAULO RODRIGUES DE FREITAS PARREIRAS

ARBITRAGEM ESTATÍSTICA E INTELIGÊNCIA ARTIFICIAL

Dissertação de Mestrado apresentada à Universidade de São Paulo para obtenção do título de Mestre em Modelagem Matemática em Finanças.

Orientador:
Prof. Dr. Renato Vicente

São Paulo
2007

Para minha irmã Mariana.

AGRADECIMENTOS

Devo agradecer a meus pais, Aurea e Alvaro, que infelizmente já não estão mais aqui – seu esforço instigou minha curiosidade intelectual e a vontade de superação. A minha irmã, Mariana, grande amiga, pela força nos momentos difíceis e pelo apoio incessante, mesmo quando eu era chato e dizia que precisava estudar. E ao resto de minha família, por todos os momentos felizes - vocês são parte daquilo que eu sou.

À Hedging-Griffo, empresa em que trabalho, pelos desafios, pelo incentivo, pela força, pelo ambiente. Toda a equipe de Gestão: Luís, Júlio, Dilton, Ettore e Ary, os economistas, os analistas de ações, os outros *traders*, o time internacional. Me orgulho de fazer parte dessa equipe, todos grandes colegas e amigos.

Devo ainda agradecer ao Prof. Renato Vicente, cujos conselhos fizeram com que este trabalho ficasse melhor do que jamais poderia ser. A todos os meus colegas e professores nestes dois anos de mestrado, pelas discussões, pelos ensinamentos, pelo excelente ambiente. Em especial ao André, Guilherme, João e Danilo, colegas já de uma outra viagem.

Por fim, a todos aqueles que, direta ou indiretamente, contribuíram com a elaboração deste trabalho.

RESUMO

O objetivo desta dissertação é o desenvolvimento de um modelo de arbitragem estatística, para identificar oportunidades no mercado de ações brasileiro, através do uso de técnicas econométricas e de inteligência artificial. O conceito de arbitragem estatística envolve a busca por anomalias momentâneas nas relações de preços entre diversos ativos, de modo que, quando tais distorções sejam corrigidas, seja possível obter lucros, com consistência e baixo risco.

O uso de técnicas do campo da econometria abre a possibilidade de determinar quando a relação entre dois (ou mais) ativos se desvia de um certo equilíbrio. O conceito de cointegração, aqui representado pela metodologia de Engle-Granger, permite testar a existência desse equilíbrio (mais precisamente, estacionariedade no resíduo), e determinar um modelo para aproveitar as oportunidades criadas pelos desvios. Na dissertação é apresentada uma variação da técnica de Engle-Granger que permite construir cestas de ações, cujos resíduos (ou "*mispricings*") são cointegrados. Contudo, tomar decisões de compra e venda apenas com base em idéias de reversão à média não necessariamente é lucrativo, como será mostrado através da simulação de estratégias de arbitragem estatística implícita.

As redes neurais aparecem então como uma ferramenta não-paramétrica de previsão, dada sua capacidade de se adaptar a dados com grande dose de ruído. A teoria relevante para o projeto e uso de uma rede neural é apresentada, e são discutidas aplicações dessa classe de modelos a problemas de previsão em finanças. Mais ainda, cada uma das características de uma boa rede é explorada, visando um modelo de alto desempenho. Este modelo é então combinado com o modelo de cointegração, e vai prever o comportamento futuro dos "*mispricings*", de maneira a identificar os pontos de compra e venda destas cestas de ações.

Ao fim, algumas técnicas de *trading* são implementadas em conjunto com os modelos, de maneira a melhorar os retornos sem aumentar os riscos, na presença de custos de transação. O modelo final é simulado fora da amostra ao longo de todo o ano de 2006, operando 29 estratégias de arbitragem simultaneamente, com retorno bastante satisfatório acima de 80%, índice de Sharpe de 3,5 e baixa correlação com o resto do mercado.

ABSTRACT

This thesis presents the development of a framework for statistical arbitrage, in which we try to identify opportunities in Brazilian stock prices, through the combined use of cointegration and artificial intelligence techniques. The concept of statistical arbitrage revolves around the search for small anomalies in the relationships between various asset prices, so that low-risk profits can be obtained once these distortions are corrected.

Econometric ideas allow the modeling of combinations of two (or more) asset prices, in order to assess deviation from any given equilibria. The concept of cointegration, here represented by the Engle-Granger methodology, develops ways of testing for the presence of this equilibrium (or more precisely, of stationarity in the residuals of a cointegrating regression), and to find an error-correcting model of these deviations. In the thesis a variation of the methodology is presented, as a way to build baskets of different stocks whose residuals (hereon called “mispricings”) are cointegrated. However, basing buy and sell decisions solely on the idea of mean-reversion is proved unprofitable once we test so-called implicit statistical arbitrage strategies.

Neural networks are a powerful non-parametric forecasting tool, especially with highly noisy data such as stock prices. The appropriate theory is presented, and applications of neural networks to forecasting in finance are discussed with the relevant literature. Furthermore, each of the building blocks of a network with good performance is explored. The “mispricing” construction model is then combined with the neural network predictive model, so that its forecasts are used as the base of trading decisions.

In the end, appropriate trading techniques are also employed, in order to enhance returns while keeping risks low, in the presence of transaction costs. The final model is tested out of sample throughout 2006, trading 29 different “mispricing” statistical arbitrage strategies, and reaching returns over 80%, with Sharpe ratio of 3.50 and very low correlation with the rest of the market.

SUMÁRIO

1. Introdução

2. Arbitragem Estatística e Cointegração

3. Redes Neurais e Previsão em Finanças

4. Modelos e Experimentos

5. Conclusões

6. Bibliografia

Apêndice A – Detalhamento dos Códigos

Índice de Figuras

Índice de Tabelas

1. INTRODUÇÃO

"A vida, e tudo que a envolve, é baseada em oportunidades de arbitragem, e como podemos explorá-las."
Paul Wilmott[1]

1.1. Motivação e Objetivos

O presente trabalho se insere na convergência de dois fenômenos absolutamente distintos, mas que têm marcado os anos recentes: o primeiro, e mais antigo, é o desenvolvimento da área de finanças quantitativas, onde uma confluência de diferentes disciplinas, somada aos avanços da disponibilidade de dados e do poder computacional, permitiu o estudo mais aprofundado da dinâmica dos mercados financeiros; o segundo fenômeno, este mais recente, é o crescimento do mercado brasileiro, o qual, após as diversas crises ocorridas entre 1997 e 2002, finalmente entrou numa fase de expansão mais acelerada, com novos participantes, maiores volumes e porque não, oportunidades.

Com o crescimento do mercado, claramente está ocorrendo uma competição mais acirrada por parte tanto dos participantes tradicionais quanto de novos entrantes, para obtenção de melhores retornos com menos risco. Obviamente, esta competição gera uma forte demanda por melhores modelos da dinâmica do mercado, e aí entra justamente o desenvolvimento da área de finanças quantitativas – a capacidade de trabalhar os dados de maneira a extrair sinais de boa qualidade vêm tendo cada vez mais importância.

Nesse contexto, o mercado de ações exerce um papel fundamental, pois permite a combinação das necessidades de capital por parte das empresas, com as demandas por investimento de prazos longos dos poupadores, seja pessoas físicas, fundos de investimento, fundos de pensão ou bancos. Assim, nos últimos anos vimos a explosão do número de novas empresas lançando suas ações na Bolsa de Valores de São Paulo (Bovespa), ao mesmo tempo em que os volumes de negócios cresceram fortemente, abrindo espaço para a aplicação de ferramentas sofisticadas de análise, na busca de oportunidades de lucro.

Nesta Dissertação são investigadas oportunidades criadas pela combinação de dois tipos específicos de ferramentas de modelagem, uma do campo da econometria e outra do campo de inteligência artificial / aprendizado de máquinas[2]. A primeira dessas ferramentas envolve o conceito de cointegração, e a segunda as chamadas redes neurais. A combinação dessas ferramentas visa explorar uma área específica de finanças quantitativas, chamada de "arbitragem estatística". A Dissertação desenvolverá uma metodologia para, utilizando estes

[1] *Apud* Poundstone, 2005. Tradução do autor.

[2] Uma tradução literal do termo *machine learning*, bastante presente na literatura.

conceitos, identificar e prever a dinâmica de preços de combinações de ações. Os modelos resultantes buscam identificar combinações de ativos que estejam (temporariamente) sub ou sobrevalorizadas, permitindo criar operações para se beneficiar do retorno dos preços aos níveis (supostamente) "justos". Este tipo de operação vai além da arbitragem tradicional, pois não se baseia apenas nas relações teóricas entre ativos mas também nas empíricas.

A combinação de metodologias de cointegração com a capacidade das redes neurais permite que se trabalhe em ambientes onde os dados contêm grandes quantidades de ruído e não-estacionariedade, o caso típico das séries temporais de ações As técnicas de inteligência artificial permitem que se relaxem algumas das hipóteses da técnica de cointegração, de maneira a gerar modelos que sejam mais facilmente operáveis, e permitem resultados melhores ao aprimorar a capacidade de prever as inovações futuras de séries temporais.

Usando dados de preços de ações negociadas na Bovespa como base, pretende-se desenvolver modelos consistentes para a geração e exploração de oportunidades de arbitragem, de maneira a avaliar a capacidade dessas técnicas de trazer retornos consistentes com baixos níveis de risco. Além disso, dada a potencial aplicabilidade da metodologia num ambiente real de investimentos[3], pretende-se trabalhar com parâmetros os mais próximos da realidade, especialmente custos de transação.

1.2. Organização da Dissertação

O trabalho se divide em quatro partes básicas. A proposta básica é solidificar uma base teórica, apresentar as aplicações dessa teoria presentes na literatura, e acrescentar os modelos e experimentos que constituem contribuição original.

A parte teórica começa no capítulo 2: após uma breve introdução, primeiro é discutido o conceito de "arbitragem estatística", dando uma perspectiva histórica e depois formalizando o conceito. Na segunda parte deste capítulo são apresentadas as técnicas econométricas que envolvem o conceito de cointegração. O início é uma discussão da representação de séries temporais, então são abordadas questões como estacionariedade e testes apropriados para caracterizar as séries de preços de ativos, para então chegar às técnicas criadas por Engle e Granger para identificar séries cointegradas. Por fim há uma breve discussão da metodologia de Johansen.

No capítulo 3 são introduzidas as Redes Neurais artificiais. Novamente inicia-se com uma introdução e um histórico, passando aos conceitos básicos de redes neurais, quais sejam, o neurônio artificial e suas características básicas, os tipos básicos de redes e os principais processos de aprendizagem. Restringindo o foco, é apresentado o algoritmo Backpropagation,

[3] O autor trabalha para uma empresa de gestão de fundos de investimentos.

base dos modelos discutidos adiante. Após isso, algumas variáveis importantes no projeto de redes de alta performance são discutidas, notadamente os algoritmos de otimização que serão aplicados na busca de soluções ótimas (ou quasi-ótimas). O capítulo termina com uma ampla discussão da literatura de aplicação de redes neurais a problemas de finanças, com ênfase na área de previsão de séries temporais de preços.

O capítulo 4 inicia com uma discussão acerca dos dados a serem utilizados nos experimentos. É importante caracterizar bem as séries de preços de ações a serem usadas, testá-las e conhecer seu comportamento antes de passar a utilizá-las nos modelos. A seguir serão apresentados os modelos de dinâmica de arbitragem estatística, onde serão construídas as séries temporais dos erros, a partir da aplicação da metodologia de *Engle-Granger*. Uma vez construídas essas séries, discutir-se-á brevemente aspectos de reversão à média, e serão apresentados os chamados modelos de "arbitragem estatística implícita", modelos de certa maneira "ingênuos", onde não serão aplicadas ainda as técnicas de previsão não-linear.

Na segunda parte do capítulo 4 são finalmente apresentados os modelos de Redes Neurais a serem utilizados na previsão da dinâmica das séries construídas na primeira parte do capítulo. Assim, serão obtidos os chamados modelos de "arbitragem estatística condicional", onde variáveis exógenas são usadas para prever o comportamento futuro das dinâmicas de preços, gerando as oportunidades de operar no mercado a partir da identificação de ativos sub ou sobrevalorizados. A partir daí, todos os conceitos do trabalho serão integrados em um grande modelo, e após uma discussão sobre diferentes regras de *trading*, poderá se avaliar os resultados finais das simulações, qual sejam, os retornos financeiros obtidos e qual o nível de risco incorrido.

Por fim, na conclusão são discutidos os resultados, quais as perspectivas de aplicação da metodologia, onde o trabalho pode ser aprimorado e quais linhas de pesquisa podem ser seguidas de maneira a incorporar melhorias nos modelos de modo a obter performances superiores.

2. ARBITRAGEM ESTATÍSTICA E COINTEGRAÇÃO

"A maioria das chamadas 'anomalias' do mercado não me parecem realmente anômalas.
Elas parecem pequenas pepitas de ouro, achadas por um de milhares
de mineiros em todo o mundo"
Fischer Black[1]

2.1. Introdução

Neste capítulo vamos iniciar a apresentação dos fundamentos teóricos por trás dos modelos a serem desenvolvidos mais adiante na Dissertação. Vamos iniciar a discussão pelo conceito de arbitragem estatística, que motivará o uso das ferramentas econométricas e de inteligência artificial, na busca de bons retornos com baixo risco. Primeiro aparecerá uma perspectiva histórica, depois os fundamentos do conceito de arbitragem, e por fim uma discussão teórica do que está por trás do termo "arbitragem estatística". Na seqüência, vamos apresentar o ferramental de análise de séries temporais a ser utilizado, inicialmente discutindo a questão da estacionariedade, em seguida apresentando as bases do conceito de cointegração, discutindo a metodologia de Engle e Granger, e por fim as técnicas de Johansen. Ao longo de todo o capítulo os conceitos serão ilustrados com exemplos do mercado financeiro, de modo a aproximar a teoria da prática.

2.2. Arbitragem Estatística

2.2.1. Histórico

Traçar a história do desenvolvimento das técnicas de arbitragem estatística não é tarefa simples. Primeiramente, porque técnicas de arbitragem estatística permitem (supostamente) a obtenção de lucros com baixo risco, e é possível imaginar que os detentores desse conhecimento gostariam de guardá-lo a sete chaves, e não divulgá-lo. Em segundo lugar, há quem defina arbitragem estatística como "qualquer tipo de *trading* com ações que use alguma forma de análise estatística"[2], e se torna um tanto quanto difícil precisar especificamente as raízes históricas desta disciplina.

[1] *Apud* Bass (1999), tradução do autor. Fischer Black é um dos autores do modelo Black & Scholes de opções.

[2] Kooi (2006). Tradução do autor.

De todo modo, boa parte da literatura segue Vidyamurthy em seu livro de 2004, datando as origens da arbitragem estatística no começo da década de 80, no banco americano Morgan Stanley. A história mais citada envolve um *trader* chamado Nunzio Tartaglia, que recrutou uma equipe de matemáticos, físicos e cientistas de computação para desenvolver modelos para operar no mercado. Essa equipe teria desenvolvido o conceito de arbitragem estatística a partir da idéia de *pairs trading*, que consiste em comprar uma ação e vender uma outra ação de maneira a explorar a performance relativa entre ambas, de maneira insulada da direção geral do mercado. Diz-se que o grupo teria obtido lucros em torno de US$ 50 milhões no ano de 1987, mas que nos anos subseqüentes a performance não teria sido tão brilhante, e por volta de 89 o grupo teria sido desmontado, e seus integrantes se espalhado pelo resto do mercado.

Um desses integrantes seria Gerry Bamberger, que teria ido trabalhar num fundo de investimentos chamado *Princeton-Newport Partners*, comandado pelo matemático Edward Thorp, um dos primeiros a aplicar sistematicamente técnicas quantitativas a operações no mercado financeiro[3]. Este teria começado a analisar conceitos de reversão à média em ações, evoluindo, com a ajuda de Bamberger, para operações de arbitragem estatística, que teriam sido utilizadas com bastante sucesso entre 1985 e 1988, quando o fundo foi encerrado.

A história segue com David Shaw[4], um cientista de computação que foi trabalhar com Tartaglia no Morgan Stanley em 1986, e em 1988 saiu para montar sua própria empresa, a *D.E. Shaw*, que gere fundos de investimento através de estratégias de arbitragem estatística, tendo iniciado com um capital de aproximadamente US$ 28 milhões[5] e hoje administrando em torno de US$ 23 bilhões. Pode-se dizer que por volta do fim dos anos 80 o uso de técnicas de arbitragem estatística se tornou comum nas mesas proprietárias dos grandes bancos e nos fundos de investimento mais agressivos (os chamados *hedge funds*).

Um desses fundos acabou por tornar-se um tanto quanto lendário. Chama-se Medallion Fund e é gerido por uma empresa chamada Renaissance Technologies. O fundador e principal nome da empresa é o matemático James Simons, um famoso matemático, que comanda uma equipe de mais de duzentas pessoas, aproximadamente um terço das quais têm Ph.Ds. O Medallion têm um histórico de retornos de aproximadamente 35% anualizado por mais de vinte anos, performance mais reluzente do que investidores famosos como George Soros, Julian Robertson e Steve Cohen[6]. O fundo opera 24 horas em todos os mercados líquidos, explorando

[3] A história de Thorp é contada em Poundstone (2006). Alguns dos detalhes aqui descritos estão em Thorp (2003).

[4] http://www.wired.com/wired/archive/5.01/ffshaw.html

[5] http://www.publicintegrity.org/report.aspx?aid=487

[6] www.olsen.ch/center/papers/hedgefuture.pdf

pequenas regularidades estatísticas nos preços dos ativos de modo a obter esse tipo de retorno, mostrando a capacidade desse tipo de estratégia de gerar retornos consistentes de longo prazo.

2.2.2. Uma Perspectiva sobre Arbitragem

Como discutido no Capítulo 1, este trabalho tem por filosofia fundamental a crença de que os avanços recentes nas técnicas de modelagem computacional abrem a possibilidade de buscar novas formas de previsão em finanças, e que isto se aplica particularmente bem ao tipo de operação conhecido como "arbitragem". Arbitragem aqui é visto num sentido amplo, como a identificação e aproveitamento das regularidades e padrões presentes na dinâmica de preços dos ativos.

Por trás dessa visão está a crença de que os preços dos ativos no mercado não têm que refletir automaticamente todas as informações disponíveis. Eles o fazem apenas na medida em que as informações são reconhecidas, interpretadas e quando finalmente os agentes tomam decisões baseadas nelas. Esta visão pode ser chamada de hipótese "eficiência relativa" do mercado[7]. Nos casos em que se pode identificar regularidades nas dinâmicas dos preços (ou dos retornos) dos ativos, isto atrairá a atenção de participantes do mercado, apropriadamente chamados de "arbitradores". Nesse sentido, arbitragem pode ser visto como um mecanismo de correção de erros, ou de *feedback* negativo, no sentido de que estes arbitradores, ao tomar decisões de compra e venda, devem eliminar (ou "arbitrar") estas regularidades (ou "oportunidades de arbitragem") que podem ser identificadas.

Há um caso clássico que ilustra este parágrafo anterior: vamos supor que seja sabido que o preço de um dado ativo sobe em um determinado dia da semana, talvez sexta-feira. Partindo desse conhecimento, os arbitradores mais capazes tenderiam a comprar esse ativo na quinta-feira, para vendê-lo ao fim do dia seguinte, dessa maneira obtendo um lucro sem risco. Ao fazer isso, estariam elevando o preço do ativo às quintas, e baixando-o nas sextas, fazendo com que parte da tradicional alta de preços deste dia passe a ocorrer na quinta-feira. Sendo assim, é possível que alguns arbitradores, mais capazes ainda, passem a fazer suas compras às quartas, novamente fazendo com que os preços passem a subir antes. Está claro que é possível estender este raciocínio de maneira a que os preços dos ativos não exibam mais nenhuma regularidade em relação ao dia da semana. Assim, a ação dos especuladores efetivamente foi responsável por eliminar o padrão observável no mercado.

Portanto, a essência competitiva dos mercados financeiros faz com que padrões presentes nas dinâmicas dos preços dos ativos sejam inerentemente difíceis de identificar, no que a

[7] Aqui não se está tomando posição em relação ao debate acerca da "Hipótese dos Mercados Eficientes" (vide Nota de Rodapé 20). Para detalhes dessa discussão, recomenda-se Malkiel, 2004, "*A Random Walk down Wall Street"* e Lo & MacKinlay, 2001, *"A Non-Random Walk Down Wall Street"*.

própria ação dos arbitradores tem boa dose de responsabilidade, fazendo com que os preços sejam, em larga medida, "eficientes", ou seja, reflitam toda a informação disponível em um determinado momento qualquer. Assim, num certo sentido, a eficiência do mercado pode ser visto como uma quasi-imprevisibilidade. A presença destes especuladores no mercado faz com que haja uma busca contínua por novas oportunidades de arbitragem.

Desse modo, e isso vai ser postulado formalmente mais adiante, podemos definir a arbitragem ideal como uma estratégia que gera lucros positivos, com risco zero, e não demandando nenhum financiamento. É possível afirmar que tais oportunidades existem nos mercados financeiros, embora raramente se apresentem nesta forma idealizada. A seguir, vamos discutir estratégias de arbitragem "sem risco", para em seguida analisarmos mais especificamente a chamada "arbitragem estatística".

2.2.3. Arbitragem "sem risco"

O conceito básico de arbitragem "sem risco" é bastante simples: se os fluxos de caixa futuros de um ativo puderem ser reproduzidos com uma combinação de outros ativos, então o preço de formar este "portfólio replicante" deve ser (aproximadamente) igual ao do ativo original. Mais especificamente, em um mercado eficiente não existirão oportunidades de arbitragem sem risco que permitam aos operadores obter lucros, através de compras e vendas de ativos equivalentes com preços diferentes, com esta diferença maior do que os custos de transação envolvidos nas operações. Assim, a condição de não-arbitragem pode ser generalizada da seguinte maneira:

$$\left| payoff\left(X_t - AS(X_t)\right)\right| < CT \tag{2.1}$$

onde X_t representa um ativo qualquer (ou combinação de ativos), $AS(X_t)$ representa um "ativo sintético" que é construído para replicar o retorno ("*payoff*") de X_t, e finalmente CT representa os "custos de transação", ou seja, o custo líquido incorrido na construção (compras e vendas) do ativo sintético $AS(X_t)$ e na correspondente compra ou venda do ativo-objeto X_t. Esta relação genérica representa a base para os modelos de apreçamento por não-arbitragem, que formam a base de boa parte da teoria de derivativos, permitindo o apreçamento de futuros, *swaps* e opções, simples ou exóticas. A idéia-força é que o preço de um derivativo pode ser obtido através do cálculo do custo de uma carteira replicante construída e gerenciada adequadamente (o que aqui chamamos de ativo sintético $AS(X_t)$). Vamos chamar a diferença $X_t - AS(X_t)$ de *erro de preço*[8].

[8] Burgess (2000) usa durante todo seu trabalho o termo em inglês "*mispricing*" para se referir a essa diferença. Adotamos a tradução "erro de preço" por se aproximar mais do espírito do termo original.

Apesar de, no detalhe, boa parte das estratégias de arbitragem "sem risco" serem bastante diferentes, existe uma estrutura comum entre todas elas. Este tipo de operação pode ser quebrado em três componentes básicos:

- Construção de relações de preços justos entre os ativos (através da derivação teórica dos portfólios replicantes).
- Identificação das oportunidades de arbitragem (quando os preços no mercado se desviam das relações de preço justo).
- Implementação das operações apropriadas para apropriação do lucro "sem risco" (ao comprar o ativo – ou conjunto de ativos – que está subvalorizado, e vender o correspondente portfólio replicante sobrevalorizado).

Um exemplo de arbitragem "sem risco" pode ser encontrado nas operações de "arbitragem de índice"[9]. Esse tipo de operação envolve as ações que compõem um particular índice do mercado, e um contrato futuro que o represente. Tipicamente o contrato futuro F_t é definido de modo que pague um valor igual ao nível do índice-objeto em uma determinada data de vencimento T. Se definirmos os preços à vista das ações hoje por S_t^i, a relação de não-arbitragem é dada por:

$$\left| F_t - \sum_i w_i S_t^i e^{(r-q_i)(T-t)} \right| < CT \tag{2.2}$$

onde w_i é o peso da ação *i* na composição do índice, r é a taxa de juros livre de risco[10] e q_i é a taxa de dividendos (contínuos) da ação *i*. Assim, no contexto da equação que define genericamente as condições de não-arbitragem, podemos enxergar a combinação ponderada das ações como o $AS(X_t)$, o ativo sintético que replica o índice futuro.

Assim, a estratégia de arbitragem baseada nesse modelo envolve o acompanhamento do chamado *"basis"*[11] $F_t - \sum_i w_i S_t^i e^{(r-q_i)(T-t)}$, que representa justamente os desvios em relação ao preço justo. Quando esse desvio excede os custos de transação de um determinado operador, este arbitrador pode obter um lucro sem risco, ao vender o índice futuro (sobrevalorizado) e comprar as ações que o compõem (que estão, ponderadas, subvalorizadas). No caso inverso,

[9] *"Index arbitrage"* no original. Existe uma boa literatura sobre esse tipo de operação. Vide por exemplo Neal (1996) para testes de modelos de arbitragem de índice, e Miller et al. (1994) para uma discussão sobre a real possibilidade de usar esse tipo de modelo para obtenção de lucros "sem risco". No mercado brasileiro esse tipo de operação é conhecido como *"cash and carry"*, pois relaciona o mercado de ações à vista (o *"cash market"*) e o de futuros (cuja precificação depende da taxa de juros, o *"carry"*).

[10] Aqui utilizamos juros exponenciais, embora o padrão no mercado brasileiro seja o juro composto de base 252, ie., por dias úteis.

[11] Aqui adotamos o termo original em inglês, como é praxe no mercado.

caso o valor negativo do desvio exceda os custos de transação, o arbitrador compraria o índice e venderia a combinação de ações. Note-se como a obtenção desse tipo de lucro de arbitragem sem risco é facilitada pelo uso de sistemas de operação automatizados, além da importância dos custos de transação. Por isso, via de regra este tipo de arbitragem é nicho de participantes de mercado com capacidade de operar rápida e eficientemente, com custos baixos.

O tópico de arbitragem "sem-risco" (ou quase "sem-risco") claramente é bastante importante, especialmente pela existência de diversas relações complexas de arbitragem, surgidas a partir do desenvolvimento do mercado de derivativos financeiros como futuros, *swaps* e opções. No entanto, a própria existência deste tipo de oportunidade é auto-limitadora, na medida em que os arbitradores estão constantemente à procura deste tipo de operação. Mais ainda, o crescimento do mercado e o aumento do grau de sofisticação da maioria dos participantes fazem com que a duração e a magnitude das oportunidades de arbitragem seja menor. Por isso, à medida que os lucros possíveis por operação diminuem, a quantidade de capital aplicada em cada uma delas deve aumentar – o que faz com que apenas os maiores participantes do mercado (tesourarias e *hedge funds*) acabem dominando este tipo de atividade.

Vale dizer ainda que na prática a teoria é outra: até as arbitragens que tecnicamente são "sem-risco" sempre envolvem uma certa dose de risco. Este é introduzido por uma variedade de fatores: a incerteza das taxas de dividendo q_i, a volatilidade do mercado durante o período em que as posições estão sendo montadas (o fenômeno do *slippage*, que comentamos em nota ao fim do Capítulo 1); e por vezes a dificuldade de realizar exatamente todas as operações requeridas pela fórmula de replicação[12], deixando algum risco que não está perfeitamente mitigado. Mais ainda, uma fonte importante de risco nesse tipo de operação de arbitragem é o chamado *"basis risk"*, ou seja, o risco de que a diferença entre o índice futuro e o índice à vista flutue ao longo do tempo, para um $t < T$. Por causa deste tipo de risco e da necessidade de marcar todas as posições a mercado, uma operação de arbitragem com um lucro "garantido", pode temporariamente mostrar prejuízo[13]. O outro lado da moeda é que esse risco pode apresentar oportunidades para o arbitrador, visto que o *"basis"* pode oscilar além do valor justo (de positivo para negativo), permitindo a reversão de operações com lucro além do esperado.

Concluindo, operações de arbitragem na prática são mais complexas do que parecem a primeira vista. De fato, a maioria das estratégias de arbitragem é, implicitamente, dependente das propriedades estatísticas dos desvios do preço justo. Dessa perspectiva, a atratividade de

[12] Um exemplo desse tipo de dificuldade é a existência de lotes-padrão de negociação no mercado de ações: digamos que o peso de uma determinada ação *i* no índice demande a compra de 8 ações, enquanto que o lote-padrão de negociação é de 10 ações.

[13] O caso do *hedge fund* americano *Long-Term Capital Management* (LTCM) é clássico nesse sentido. Este carregava diversas operações de arbitragem que momentaneamente, por causa do *default* da dívida russa, apresentaram, na marcação a mercado, enormes prejuízos, o que fez com que o fundo enfrentasse problemas (obviamente esta é uma visão simplista do caso). Mais detalhes em Dunbar (2000) ou Lowenstein (1999).

operações como esta “arbitragem de índice” vem do fato de que este desvio tende de reverter à média, ou seja, ele flutua em torno de um nível estável. A seguir vamos discutir como este fato, a existência de propriedades estatísticas interessantes nas dinâmicas dos desvios, permite a criação de uma classe mais geral de estratégias, agrupadas sob o termo “arbitragem estatística”.

2.2.4. Elementos de Arbitragem Estatística

A discussão anterior mostra como, de uma perspectiva estatística, a série temporal do desvio pode ser considerada como um ativo sintético que exibe fortes características de reversão à média, e portanto possui alto potencial preditivo. Assim, a premissa básica por trás do conceito de arbitragem estatística[14] é a existência, em situações de mercado, de regularidades estatísticas nos preços dos ativos que podem ser exploradas como base de estratégias de *trading* lucrativas, mesmo na ausência de uma relação teórica *a priori* de preço justo.

Embora esse tipo de estratégia seja sujeita a uma dose maior de risco do que aquela que examinamos no item anterior, as arbitragens “sem risco”, esse tipo de oportunidade de arbitragem estatística também deve ser mais persistente e mais presente nos mercados financeiros. Mais persistente porque arbitragens “sem risco” tendem a ser exploradas mais rapidamente pelos participantes do mercado. Mais presente porque não existe nenhuma restrição pré-existente em relação a que ativos (ou classe de ativos) exibem comportamentos que geram oportunidades de arbitragem estatística. Ao contrário, no tipo de operação que exemplificamos no item anterior depende da existência de uma relação de valor justo bem estabelecida. Um exemplo de dois ativos que dão origem a oportunidades de arbitragem estatística está na Figura 2.1, que mostra o *spread* entre Petrobrás ON e Petrobrás PN. O *spread* foi gerado de acordo com a metodologia que vamos desenvolver mais adiante no Capítulo 4.

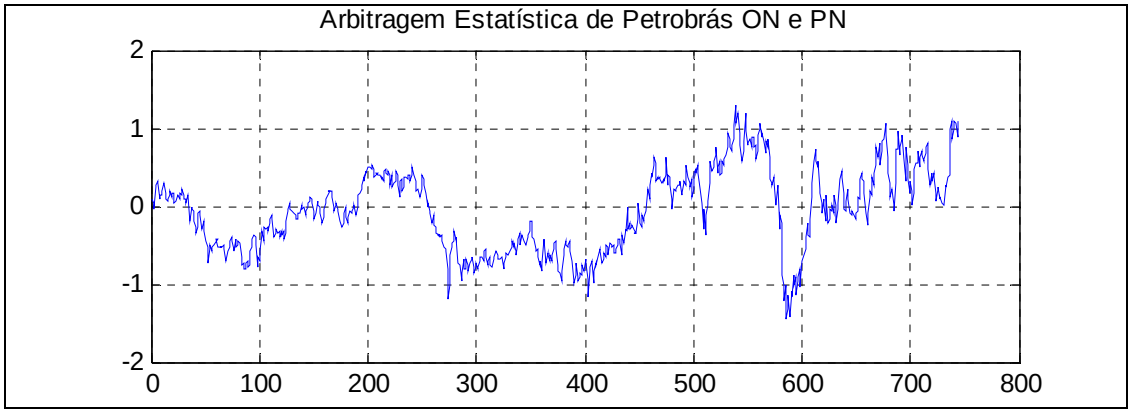

Figura 2.1: Arbitragem Estatística a partir do *spread* entre PETR3 e PETR4

[14] Para mais detalhes de estratégias de arbitragem, vide, entre outros, Nicholas (2000) ou Stefanini (2006).

Comparando este exemplo com o dado no item anterior, existe uma semelhança clara. Em ambos os casos os preços se movem "conjuntamente" no longo prazo, com desvios temporários dessa correlação de longo prazo, mas que exibem forte tendência a reverter à média. O importante aqui é notar como no caso da arbitragem estatística a magnitude dos desvios é significativamente maior do que no caso da arbitragem "sem risco" (em torno de +/- 10%, enquanto no anterior fica em torno de 0,50%), assim como o tempo para correção dos desvios também é bastante mais longo (estamos falando de dias ou semanas ao invés de minutos).

Nessa perspectiva, vamos caracterizar a arbitragem estatística como um caso geral da arbitragem "sem risco" que examinamos anteriormente, no qual os retornos relativos, dados por $payoff\left(X_t - AS(X_t)\right)$, não mais podem ser completamente mitigados com relação aos principais fatores de risco que guiam os preços dos ativos em geral. Vamos examinar com alguma calma esta afirmação: dado um conjunto de variáveis econômicas, ativos ou portfólios de ativos, que correspondam aos fatores de risco $F = \{F_1,...,F_{nF}\}$, podemos quantificar a quantidade de risco que cada fator introduz em nosso portfólio de "desvio estatístico", que é dado por $X_t - AS(X_t, w)$, através das sensibilidades $s = \{s_1,...,s_n\}$, que são definidas por:

$$s_i = \frac{dE\left[payoff\left(X_t - AS(X_t, w)\right)\right]}{dF_i} \tag{2.3}$$

onde w são os parâmetros que definem o ativo sintético AS. No caso da arbitragem "sem risco" que examinamos anteriormente, o retorno futuro $payoff\left(X_t - AS(X_t)\right)$ depende apenas da magnitude do desvio $X_t - AS(X_t, w)$ no início da operação, e todos os s_i são iguais a zero.

Num caso em que um conjunto adequado de fatores de risco pode ser pré-definido, talvez a maneira mais natural de desenvolver estratégias de arbitragem estatística seja introduzindo um conjunto especificado de sensibilidades s_i. Assim, poderíamos poder exemplo examinar qual o conjunto comum de fatores de risco de um grupo de ativos através de uma técnica como Análise de Componentes Principais (*PCA*[15]). Assim, o ativo sintético AS pode ser construído como uma particular combinação de ativos que exibe um perfil de fatores de risco desejado, em termos das sensibilidades s_i. É possível construir portfólios que tenham perfis de risco do tipo $s_j = 1, \forall_{i \neq j} : s_i = 0$, ou seja, com exposição a apenas um fator desejado – de uma maneira, esta é uma aposta apenas naquele determinado fator de risco.

Ao longo do resto do trabalho, notadamente no Capítulo 4, vamos explorar uma perspectiva diferente dessa modelagem por fatores descrita acima. No caso, ao invés de buscar explorar os fatores de risco $F = \{F_1,...,F_{nF}\}$, vamos tratar o problema de uma maneira

[15] Do inglês, *Principal Component Analysis.*

agregada, onde o que vai importar vai ser o desvio ou o *"tracking error"* entre a série temporal do ativo ou portfólio X_t e o ativo sintético $AS(X_t, w)$. O ativo sintético será construído de maneira a minimizar a variância entre as séries temporais, ou seja, os parâmetros w^* serão determinados através da relação:

$$w^* = \arg\min_w \text{var}(X_t - AS_t(X, w)) \tag{2.4}$$

No limite onde o desvio seja zero, ou seja, $\text{var}(X_t - AS_t(X, w)) = 0$, teríamos que as sensibilidades s_i aos fatores de risco (quaisquer fatores de risco), seriam todas zero. Já num caso mais geral, onde $\text{var}(X_t - AS_t(X, w)) > 0$, as sensibilidades s_i estarão implicitamente definidas, na medida em que ao minimizar a variância residual (com respeito aos parâmetros w^*), estaremos minimizando a exposição agregada aos fatores de risco, sem contudo determinar nenhuma condição em relação às exposições individuais a cada um dos fatores. Mais ainda vamos examinar as ferramentas econométricas para análise de cointegração, o que vai permitir a construção dos ativos sintéticos $AS(X_t, w)$ de maneira consistente.

Do mesmo modo como determinamos os três componentes básicos para estratégias de arbitragem "sem risco", podemos fazê-lo para estratégias de arbitragem estatística:

- Construção de relações estatísticas de preços justos entre os ativos, de maneira a que os desvios tenham uma componente potencialmente previsível (através da análise das séries temporais dos preços dos ativos).
- Identificação das oportunidades de arbitragem (através da previsão das variações nas combinações apropriadas dos ativos).
- Implementação das operações apropriadas (ao comprar o ativo – ou conjunto de ativos – que se prevê estar subvalorizado, e vender o ativo – ou conjunto – que se prevê estar sobrevalorizado).

Estas três componentes correspondem às três partes da metodologia que vamos aplicar nos modelos do Capítulo 4. A relação para arbitragem estatística equivalente àquela que desenvolvemos na Equação 2.1 pode ser dada por:

$$E\left[\left|payoff\left(X_t - AS(X_t)\right)\right|\right] < CT \tag{2.5}$$

onde usamos o operador de expectativa. Nesse contexto, o desafio a ser enfrentado no Capítulo 4 é claro: primeiro, dado um ativo X_t, identificar uma combinação de ativos $AS(X_t, w)$ que seja estatisticamente relacionada a X_t[16]. Em segundo lugar, devemos criar modelos capazes de prever o comportamento futuro da série temporal do desvio $X_t - AS(X_t)$, gerando dessa

[16] Mais adiante vamos definir o que quer dizer essa relação estatística.

forma uma previsão $E\left[\left|payoff\left(X_t - AS(X_t)\right)\right|\right]$, e terceiro, devemos construir uma estratégia de *trading* adequada para explorar as oportunidades geradas por essas previsões, de uma maneira a obter lucros acima dos custos de transação.

2.2.5. Algumas Considerações Teóricas

Parte da literatura busca definir teoricamente o que se quer dizer por uma estratégia de arbitragem estatística, indo além da definição que discutimos nos dois itens anteriores (que pode ser considerada um tanto *ad hoc*). Vamos aqui nos referir às discussões iniciais presentes tanto Hogan et al. (2004) e Teo et al. (2004), dois trabalhos cujos resultados vamos analisar no item seguinte.

Hogan et al. (2004) coloca a seguinte definição: um arbitragem estatística é uma estratégia de *trading* com custo inicial zero, auto-financiada $(x(t) : t \geq 0)$ com valor acumulado descontado $v(t)$ tal que:

1. $v(0) = 0$
2. $\lim_{t\to\infty} E^P[v(t)] > 0$
3. $\lim_{t\to\infty} P(v(t) < 0) = 0$
4. $\lim_{t\to\infty} \frac{Var^P[v(t)]}{t} = 0$ se $P(v(t) < 0) > 0, \forall t < \infty$

Assim, por definição, temos que uma estratégia de arbitragem estatística satisfaz quatro condições: (i) ela é autofinanciada, com custo inicial zero; (ii) ela tem expectativa de lucros positivos; (iii) ela tem uma probabilidade de perda que converge para zero e (iv) a média temporal da variância converge para zero se a probabilidade de perda não se tornar zero num tempo finito. Esta última condição, em termos econômicos, quer dizer que uma oportunidade de arbitragem estatística eventualmente produz lucros incrementais sem risco, ou seja, o Índice de Sharpe da estratégia cresceria monotonicamente ao longo do tempo. Esse fato é consistente com uma variância da estratégia de *trading* que tende ao infinito ao longo do tempo, mas com crescimento menor que linear (e portanto a média temporal tende a zero). A Condição 4 também implica que $P\left(v(t)/\sqrt{t} < 0\right) \to 0$, ou seja, o risco de se perder dinheiro (por unidade de tempo) vai para zero no limite[17]. Assim, a capacidade que uma estratégia tem de reduzir sua variância média ao longo do tempo, através de ajustes seja em sua composição quanto no tamanho das posições *long* e *short*, é essencial para termos uma arbitragem estatística.

[17] Hogan et al. (2004), pp. 9, na nota de rodapé 7, contém uma prova dessa afirmação.

Uma distinção importante presente em Hogan et al. (2004) é o fato de esta definição ser apropriada para períodos de tempos "longos", embora não seja necessário que um investidor espere até o infinito para se beneficiar de uma arbitragem estatística. O fato é que investidores com horizontes de tempo finitos podem observar oportunidades de arbitragem estatística como "boas demais para deixar passar", na medida em que oferecem retornos esperados positivos, risco decrescente de perda e variância (por unidade de tempo) que se torna arbitrariamente pequena. Desse modo, pode-se dizer que existe um ponto de tempo finito t^*, de maneira que a probabilidade de perda seja tão pequena quanto se deseje, ou seja, $P(v(t^*) < 0) = \varepsilon$. Comparando uma oportunidade de arbitragem sem risco (como discutimos no item 2.2.3) no tempo t^*, para a qual $P(v(t^*) < 0) = 0$, com uma oportunidade de arbitragem estatística (como visto no item 2.2.4), para a qual $P(v(t^*) < 0) = \varepsilon$, fica claro que ambas estão separadas por ε de probabilidade de perda.

Hogan et al. (2004) então define um processo estocástico que caracterizaria os lucros incrementais de uma estratégia de arbitragem estatística, e define uma série de testes para auferir o cumprimento dos quatro pré-requisitos definidos acima. Teo et al. (2004) estendem os testes, de maneira a obter resultados ainda mais robustos que os de Hogan et al. Dada a ênfase deste trabalho na aplicabilidade prática da estratégia, não vamos expor aqui os detalhes da formulação desses testes. A seguir vamos examinar os resultados da literatura sobre a performance de estratégias de arbitragem estatística.

2.2.6. Modelos de Arbitragem Estatística : Análise da Pesquisa Prévia

Após discutirmos a formulação teórica do conceito de arbitragem estatística, cabe analisar a literatura sobre o assunto, de maneira a buscarmos pontos fortes e fracos, caminhos a seguir e erros a evitar, e assim estar preparados para o desenvolvimento dos modelos que buscaremos no Capítulo 4. Vale aqui reforçar aquilo que dissemos anteriormente, na seção 2.2.1: "*técnicas de arbitragem estatística permitem (supostamente) a obtenção de lucros com baixo risco, e é possível imaginar que os detentores desse conhecimento gostariam de guardá-lo a sete chaves, e não divulgá-lo*". De qualquer maneira, vamos analisar brevemente os textos mais relevantes publicados sobre esse tema.

A literatura se divide em três tipos de abordagem ao problema de arbitragem estatística: o primeiro grupo trabalha com performance relativa de grupos de ações, explorando o chamado efeito *momentum*, ou seja, comprando o que subiu muito e vendendo o que subiu pouco (ou o contrário em mercados de baixa); o segundo grupo busca modelar o comportamento estatístico de grupos de ações, buscando previsibilidade e modelos de correção de erros, com técnicas como cointegração; e por fim o terceiro grupo busca uma equação paramétrica (normalmente

estocástica) que modele o comportamento conjunto de um grupo de ações. É possível argumentar que o segundo e o terceiro tipos são similares, e de algum modo eles têm conceitos em comum. Mas o modo como os autores atacam o problema nos leva a separar em dois grupos os artigos.

O primeiro grupo é exemplificado pelos artigos de Larsson et al. (2003), Hogan et al. (2004) e Teo et al. (2004). Em Larsson et al. (2003), os autores buscam explorar o chamado efeito "*momentum*" .Este efeito, já amplamente estudado na literatura de finanças, e pode ser definido como a "persistência de performance nos retornos de ações por períodos entre três meses e um ano" (Grinblatt & Han, 2002)[18]. Ele pode ser explicado pelo comportamento dos investidores, que agem de forma não plenamente racional e/ou não têm capacidade computacional ilimitada, e portanto tendem a observar novas informações de acordo com padrões já conhecidos no passado, gerando efeitos de sobre e sub-reação[19]. A existência deste efeito no mercado vai contra a *Hipótese do Mercados Eficientes*[20]. A proposta dos autores é construir um modelo para testar a existência deste efeito no mercado de ações da Suécia, usando para tanto séries de preços de 175 diferentes ações, e buscando também informações fundamentalistas das empresas (por exemplo, preço/valor patrimonial). A partir daí o modelo que classifica as ações de acordo com seu *momentum*, com base na performance das ações nos últimos 6 meses, e compra as 10 ações de melhor performance, enquanto vende aquelas 10 de pior performance. Por isso o modelo é chamado de *neutro ao mercado*, pois constrói um portfólio *long-short* que busca se isolar das oscilações do mercado e capturar em seus retornos apenas o efeito *momentum* das ações. Além disso, os autores propõem vários tipos de controles de risco para buscar melhorar a performance global do sistema, diminuindo sua volatilidade sem abrir mão de muito retorno. Para tanto, implementam regras de *stop-loss*, excluem ações de preços muito baixos e ações que estejam (supostamente) excessivamente valorizadas. A simulação do modelo dos autores mostra bons resultados, conseguindo retornos positivos (21,8% anualizado), mas com um Sharpe que não poderia ser chamado de espetacular (0,71). Além disso, a correlação com o mercado em geral é bastante baixa, cumprindo o objetivo de ser *market-neutral*. Os resultados são comparados com várias alternativas para testar a robustez do modelo, mostrando a consistência do efeito e a qualidade do sistema desenvolvido.

Já Hogan et al. (2004) e Teo et al. (2004) adotam uma perspectiva de mais longo prazo, mais teórica que a de Larsson et al. (2003), mas também buscando determinar se a existência de arbitragem estatística contradiz a *Hipótese dos Mercados Eficientes*. O trabalho de Hogan et al. define estratégias não só de *momentum* como também de *value*, esta relacionada a múltiplos

[18] Grinblatt, M.; Han, B., *The Disposition Effect and Momentum*, Working Paper 8734, NBER, janeiro 2002.

[19] Do inglês, *overreaction* e *underreaction*.

[20] *EMH* ou *Efficient Market Hypothesis*, para detalhes vide Fama, E., *Efficient Capital Markets: A Review of Theory and Empirical Work*, Journal of Finance, 48, pp. 383-417, 1970.

relacionados às ações, como preço/lucro, preço/valor patrimonial, e as testa com vários períodos de formação (período do passado em que se compilam os dados que servem de base para escolha das ações) e períodos de permanência dos portfólios, com uma mecânica operacional semelhante à de Larsson et al. (2003). Os autores também controlam para custos de transação e efeitos de ações de empresas pequenas (que normalmente têm baixa liquidez). Os resultados mostram que boa parte das estratégias de *momentum* e *value* geram arbitragens estatísticas (no primeiro caso, 6 de 16, enquanto que no segundo 5 de 12 estratégias testadas constituem arbitragens estatísticas, dentro dos critérios definidos pelos autores), contradizendo a existência dos mercados eficientes. O trabalho de Teo et al. (2004) estende o trabalho de Hogan et al. através de certas sofisticações e definições mais rigorosas dos testes estatísticos feitos nos resultados das estratégias de *momentum* e *value* que confirmam a existência de arbitragens.

De um modo geral, esse primeiro grupo de trabalhos pode ser chamado de estratégias simples de arbitragem estatística, com um viés comportamental. As estratégias simples do tipo "venda as que mais subiram" e "compre as que mais caíram" buscam explorar implicitamente a existência de reversão à média no comportamento dos preços dos ativos, mas sem definir exatamente o que é essa reversão e porque ela existe. Representam uma primeira aproximação do problema, por assim dizer.

O segundo grupo de trabalhos sofistica a questão, buscando ferramentas econométricas, principalmente o conceito de cointegração, que vamos discutir detalhadamente a partir do item 2.3 deste capítulo, para fundamentar a construção de estratégias de arbitragem estatística. Estão nesse grupo os trabalhos de Alexander & Dimitriu (2002), Alexander et al. (2003), Refenes et al. (1997), Burgess (1999), Burgess (2000) e Burgess (2003), Vidyamurthy (2004), além de Gatev et al. (1999), que embora não aplique cointegração, usa uma técnica de inspiração semelhante. O ponto comum de todos esses trabalhos é a busca por uma metodologia de construção de pares de ações (ou cestas de ações, como em Burgess (2000), por exemplo) através de técnicas econométricas.

Desse grupo, o trabalho mais simples é o de Gatev et al. (1999), que constrói pares de ações baseado na distância entre elas num espaço de estados normalizado, ou seja, minimizando o erro quadrático médio entre as duas ações. Os autores afirmam que "essa perspectiva é a que melhor aproxima a descrição de como operadores de mercado escolhem pares de ações"[21]. Definidos vários pares, é determinada uma regra de *trading*: compra-se o par quando ele vai dois desvios-padrão abaixo da média histórica, e vice-versa. Dadas essas regras, de seleção e operação, os autores montam um portfólio de arbitragens, composto dos vinte pares com a menor medida de distância, que é testado fora da amostra por seis meses. Os retornos em excesso (ou seja, além da taxa livre de risco), para períodos de seis meses, têm média de 6%,

[21] Gatev et al. (1999), pp. 7. Tradução do autor.

com desvio de aproximadamente 5%. Ou seja, estamos falando de retornos anualizados de 12% em excesso (ou seja, *alpha* puro), resultados nada desprezíveis.

Já Alexander et al. (2003) se busca a construção de carteiras *long-short* baseadas em cointegração, para replicar um determinado *benchmark* com uma parcela de *alpha*. O modelo parte de um índice específico de ações nos Estados Unidos, o *S&P100*, e a partir das ações que o compõem, forma portfólios com 75 ações que tem a maior cointegração com o *benchmark* mais *alpha* (ie, retorno do índice mais 5%aa.), com base na estatística de um teste de Dickey-Fuller aumentado (*ADF*)[22]. Após determinar esta carteira de 75 ações a partir das séries temporais para um dado período de treinamento, o modelo calcula a performance para períodos de 1, 2 e 3 meses fora da amostra, de maneira a determinar os retornos e a volatilidade das carteiras. Os autores constroem chamados "mapas de calor"[23] para determinar as variáveis (período de treinamento e *alpha*) que geram as melhores relações risco/retorno. A partir destes mapas, são determinadas a melhor carteira (posição a ser comprada – *long*) e a pior carteira (posição a ser vendida – *short*) de modo a ter um portfólio neutro com relação à direção do mercado. Dentro da metodologia acima, são construídos portfólios mês a mês, com rebalanceamento do modelo, e os resultados são bastante promissores: assumindo custos de transação fixos, os autores obtêm retorno anualizado de 27,2%, com Sharpe de 1,51 (no ano 2000, quando houve o *crash* das ações na Nasdaq, com o *S&P100* retornando -11,88%, o modelo obtêm retorno de 58,38%, com Sharpe de 3,93), mostrando a eficácia de técnicas de cointegração para a construção de carteiras que tenham não só retornos interessantes como com pouco risco, e imunes à tendência geral do mercado. Alexander & Dimitriu (2002) trabalham com metodologia semelhante e obtém resultados parecidos, com ênfase em encontrar qual o nível ótimo de *alpha* que deve ser adicionado ao *benchmark* na construção dos portfólios por cointegração.

Todos os trabalhos de Refenes e Burgess apresentam metodologia similar entre si, fazendo a construção de cestas de ações através de cointegração, e buscando técnicas para previsão da dinâmica do "*mispricing*", como discutimos no item 2.2.3, principalmente através do uso de redes neurais. A metodologia é construída passo a passo em Burgess (2000). Em resumo, ela começa com a discussão da construção de cestas de ações através de análise econométrica por cointegração, depois passa por um aprimoramento para tornar a metodologia dinâmica, ou seja, condizente com regimes de mercado variáveis ao longo do tempo. A partir daí são discutidas técnicas para construção de modelos de previsão com redes neurais, e depois a integração desses modelos em portfólios de arbitragem estatística, usando algoritmos genéticos. Em linhas gerais, tal metodologia inspira este trabalho. Os resultados são bastante interessantes: em Burgess

[22] Mais detalhes no item 2.3.2.

[23] *Heat maps* no original.

(2000), pp. 189-197, o autor apresenta a aplicação de um modelo de arbitragem entre o índice DAX da Bolsa de Frankfurt e o índice CAC da Bolsa de Paris, usando uma versão dinâmica do seu modelo (sem previsão com redes neurais), e mostra retornos acumulados num período de seis anos de 71,5%, com resultados positivos em todos os anos[24]. A performance dos modelos de arbitragem estatística com previsão neural também é promissora, apresentando retornos anualizados entre 19,4% e 28,2% como índices de *Sharpe* entre 3,26 e 5,14[25].

Em resumo, os trabalhos que definimos no segundo grupo, talvez o grupo mais ligado diretamente à prática de mercado, apresentam conclusões interessantes. Tanto Gatev et al., quanto Alexander e Burgess mostram resultados bastante sólidos, mostrando como a busca por técnicas de arbitragem estatística claramente pode adicionar retornos financeiros substanciais.

Por fim, temos ainda que brevemente falar do terceiro grupo de trabalhos, na verdade de um trabalho, o de Elliot et al. (2005). Esse trabalho se assemelha a Burgess (2000), na medida em que lida basicamente com o *"mispricing"* ou *"spread"*, de um par de ações. Mas a diferença é que enquanto Burgess (2000) não faz nenhuma hipótese acerca do comportamento desse valor, buscando uma técnica não-linear para tentar prevê-lo, Elliot et al. (2005) modela o "*spread*" com uma cadeia de Markov Gaussiana, um processo com reversão à média. A partir desse modelo, aplicam um filtro de Kalman para tentar calibrar o modelo a preços de mercado. Não vão muito além disso, sendo um tanto vagos. Em suma, o trabalho parte de uma definição *a priori* do comportamento de um par de ações, para tentar calibrá-lo ao mercado. É uma perspectiva inversa à do grupo anterior de trabalhos, onde os dados são primeiro analisados em buscas de comportamentos desejados (basicamente estacionariedade) para então montar estratégias de *trading* para lucrar com esses comportamentos. A seguir, vamos apresentar o ferramental econométrico que auxilia essa busca e permite encontrar arbitragens estatísticas.

2.3. Cointegração

2.3.1. Introdução

Alexander (2001) inicia sua discussão sobre cointegração com uma frase bastante interessante: "*Tentar modelas as interdependências complexas entre ativos financeiros usando apenas a ferramenta simples da correlação é como tentar surfar na internet usando um IBM AT*"[26]. Desde o trabalho seminal de Engle e Granger em 1987 que a cointegração se tornou uma

[24] Burgess (2000), pp. 194, Tabela 7.9.

[25] Burgess (2000), pp. 288, Tabela 11.10.

[26] Alexander (2001), pp. 347. Tradução do autor.

das ferramentas básicas da econometria. O fato é que esta é uma técnica poderosa para a análise de tendências comuns em séries temporais multivariadas, proporcionando uma teoria bastante sólida para exploração tanto das dinâmicas de longo prazo quanto das de horizonte mais curto de um sistema.

Embora modelos empíricos de séries temporais financeiras cointegradas sejam comuns na literatura[27], a implementação prática desses modelos em sistemas de avaliação de investimento e *trading* ainda está em seus estágios iniciais. A principal razão para tanto é o fato de que a análise financeira tradicional parte de uma análise da correlação dos retornos de dois ou mais ativos. Nos modelos tradicionais de análise de risco-retorno, as séries dos preços dos ativos são diferenciadas antes do início de qualquer análise, e essa diferenciação remove, *a priori*, qualquer tendência de longo prazo presente nos dados. Essas tendências (se houver alguma) continuam presentes implicitamente na série de retornos, mas quando se toma decisão utilizando a metodologia padrão na literatura de finanças, nada é baseado na (eventual) presença de tendências comuns de longo prazo entre os diferentes ativos. Já a análise por cointegração tem por objetivo fundamental a descoberta de tendências comuns estocásticas nas séries temporais de preços dos ativos e o uso dessas tendências para uma análise dinâmica do comportamento desses ativos, com a conseqüente possibilidade de tomada de decisões mais bem fundamentadas.

O cálculo de correlações (usualmente) se baseia nas séries temporais de retornos, enquanto a cointegração trabalha com séries de preços (ou taxas de câmbio, ou taxas de juros – em nosso caso, como vamos tratar de ações, apenas preços). Usualmente essas séries que desejamos analisar não são estacionárias, sendo comumente integradas de ordem 1 (denotados $I(1)$). Assim sendo vamos passar a seguir por uma breve discussão do que é estacionariedade, integração, e como testar esses conceitos.

2.3.2. Variáveis Estacionárias, Processos Integrados e Raiz Unitária

Séries temporais podem ter tanto componentes estocásticos quanto determinísticos. Por exemplo, uma série com uma tendência determinística e um componente estocástico de ruído branco pode ser denotada por:

$$y_t = \alpha + \beta t + \varepsilon_t \tag{2.6}$$

onde $\varepsilon_t \sim iid(0, \sigma^2)$. A maior parte dos modelos de séries temporais em finanças vai ter um componente estocástico, e assim podemos calcular tanto sua expectativa não-condicionada quanto a variância da *t*-ésima observação. Por exemplo, no modelo acima teríamos:

[27] Vimos alguns exemplos no item 2.2.6, e Alexander (2003), pp. 366-369 contém vários outros, com variadas aplicações que não só arbitragem estatística.

$$E[y_t] = \alpha + \beta t \tag{2.7}$$

$$VAR[y_t] = \sigma^2 \tag{2.8}$$

Mais ainda, a autocovariância de ordem *s* (ie, a covariância de y_t com respeito a y_{t-s} para o modelo acima seria:

$$\text{cov}(y_t, y_{t-s}) = E\left[(y_t - E[y_t])(y_{t-s} - E[y_{t-s}]\right] \tag{2.9}$$

E portanto teríamos que nessa formulação, a série temporal acima definida teria $\text{cov}(y_t, y_{t-s}) = E[\varepsilon_t \varepsilon_{t-s}] = 0$, para todo *t* e $s \neq 0$.

Uma série temporal $\{y_t\}$ é dita covariância-estacionária se a expectativa, variância e autocovariância são as mesmas a cada instante *t*, ou seja:

- $E[y_t]$ é uma constante finita;
- $VAR[y_t]$ é uma constante finita;
- $\text{cov}(y_t, y_{t-s})$ depende apenas do lag *s*.

Essa é uma forma fraca de estacionariedade, que é usualmente o que se quer dizer quando uma série temporal é dita "estacionária". Uma forma mais forte de estacionariedade, em que não só as autocovariâncias como toda a distribuição conjunta é independente da data em que é medida, mas apenas do *lag s*, é dita "estritamente estacionária".

A série temporal que definimos acima não é estacionária. Embora satisfaça a segunda e terceira condições, a média não-condicional de y_t não é independente do tempo. Qualquer série com uma tendência na média não será estacionária, e essa é uma razão pela qual preços de ativos financeiros (e seus logs) comumente não são estacionários. As tendências presentes nos mercados financeiros normalmente não seguem o modelo acima, e isso se deve principalmente ao fato de que as tendências presentes são estocásticas e não determinísticas. Mais adiante vamos analisar a diferença entre esses dois modelos.

A Figura 2.2 ilustra o comportamento bastante diferente entre preços de ativos e retornos de ativos. Enquanto preços (e seus logs) na maioria dos mercados são representados por modelos de séries temporais não-estacionárias, a primeira diferença dos preços (na verdade a primeira diferença do log dos preços, que é equivalente ao retorno), é representada por um processo estacionário (note como a série oscila em torno de valores bem definidos e limitados).

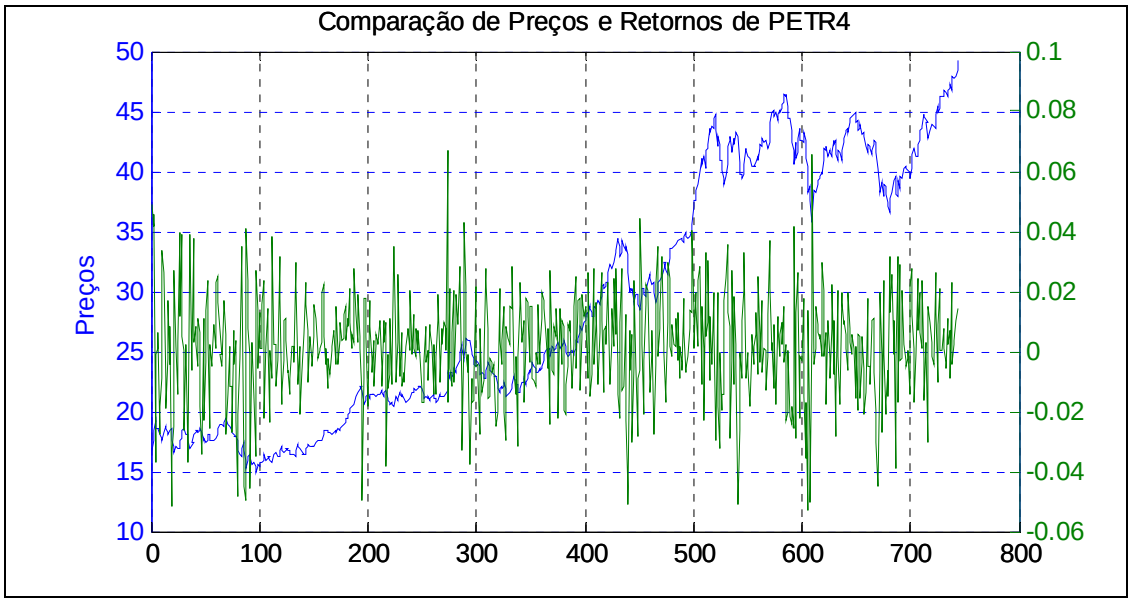

Figura 2.2: Comparação de séries temporais de preços e retornos de PETR4

Um exemplo simples de uma série temporal estacionária é dado por um modelo autoregressivo de ordem 1, ou um *AR(1)*. Vamos considerar uma versão de um modelo *AR(1)* sem constante:

$$y_t = \alpha y_{t-1} + \varepsilon_t \tag{2.10}$$

onde $\varepsilon_t \sim iid(0, \sigma^2)$. Sabe-se que o modelo geral *AR(1)* é estável somente se $|\alpha| < 1$, e nesse caso ele define um processo estacionário[28].

A propriedade de reversão à média de séries estacionárias é bem conhecida. Basicamente, um processo estacionário não pode se distanciar demais de sua média, por ter variância finita. A velocidade de reversão à média é determinada pela autocovariância: ela é rápida quando as autocovariâncias são pequenas, e lenta quando são grandes. Para um modelo *AR(1)* como o que definimos acima, a velocidade de reversão à média depende do parâmetro α. Se tivermos $\alpha = 0$, então $\{y_t\}$ é um ruído branco e a reversão é instantânea, pois $\text{cov}(y_t, y_{t-s}) = 0$. A medida que α aumenta em valor absoluto, a velocidade de reversão cai. No limite, quando $\alpha = 1$, $\{y_t\}$ é um passeio aleatório (sem tendência), um tipo de processo não-estacionário, e não há qualquer reversão à média.

Vamos discutir um pouco mais esta questão do passeio aleatório. Podemos defini-lo como:

$$\ln P_t = c + \ln P_{t-1} + \varepsilon_t \tag{2.11}$$

aqui aplicado aos logaritmos do preços de um ativo, e com $\varepsilon_t \sim iid(0, \sigma^2)$. Esse modelo de passeio aleatório está umbilicalmente ligado à chamada "Hipótese dos Mercados Eficientes"

[28] Alexander (2001), pp. 318-319 contém uma prova dessa afirmação. A prova envolve o fato de que somente com $|\alpha| < 1$ o modelo satisfaz as três condições que definimos anteriormente.

(vide Notas de Rodapé 7 e 20 neste Capítulo). Na verdade, a hipótese implica apenas que a distribuição dos retornos futuros condicionado ao conjunto de informações até o tempo *t* tenha $\varepsilon_t \sim iid(0,\sigma^2)$. Ou seja, $\varepsilon_t | I_t \sim iid(0,\sigma^2)$, onde I_t é o conjunto de informações disponíveis até o instante *t*.

O modelo do passeio aleatório permite que haja tendências nos preços dos ativos, ao incluir na sua formulação o termo constante *c*, correspondente ao retorno esperado. Assim, se tivermos *c>0* os (log-)preços estão tendendo para cima, e vice-versa. Mesmo no caso em que *c=0*, dizemos que há uma *tendência estocástica*. É provável que uma inspeção visual dos dados não indique nenhuma tendência clara dos preços, mas o termo "tendência estocástica" ainda se aplica porque os dados foram gerados por um processo *integrado*.

O que isso quer dizer? Uma série temporal é *integrada de ordem n*, ou seja $y_t \sim I(n)$, se sua componente estocástica for não-estacionária, mas se tornar estacionária após ser diferenciada no mínimo *n* vezes. Assim um processo que já é estacionário é denotado *I(0)*. Um passeio aleatório é um exemplo de um processo integrado de ordem 1. De modo geral, boa parte dos processos de preços nos mercados financeiros pode ser caracterizada pelo modelo:

$$\ln P_t = c + \ln P_{t-1} + \varepsilon_t \tag{2.12}$$

com $\varepsilon_t \sim I(0)$. A diferença entre este modelo e o que definimos na equação 2.11 é que lá as inovações não eram apenas estacionárias, elas eram um ruído branco. Em geral os mercados não são plenamente eficientes, e é possível que os log-preços não sejam passeios aleatórios puros, pois seus retornos podem ser autocorrelacionados, mesmo que ainda sejam processos *I(1)*.

Um aspecto relevante que devemos chamar atenção é que a tendência existente no modelo da equação acima não é uma tendência determinística. Ou seja, aquele modelo *I(1)* é diferente do modelo:

$$\ln P_t = c + \beta t + \varepsilon_t \tag{2.13}$$

que tem um componente estacionário ($\varepsilon_t \sim I(0)$) e uma tendência determinística (βt). Nem este nem o anterior produzem séries estacionárias, e uma inspeção visual de ambos pode confundir as coisas. Mas enquanto no modelo anterior a transformação necessária para deixá-lo estacionário é tirar a primeira diferença (e por isso aquele é um processo integrado de ordem 1), nesse modelo com tendência a transformação necessária para deixá-lo estacionário envolve os desvios em relação a uma tendência ajustada. De fato, se os dados forem gerados por um processo deste último tipo, se tirarmos a primeira diferença obteremos um processo com grande autocorrelação negativa[29].

[29] Alexander (2001), pp. 323 aponta esse fato, e pp. 324 mostra graficamente, usando o preço de uma ação, que tirar a primeira diferença e calcular os desvios em relação a uma tendência apresentam resultados distintos, sendo que o primeiro gera inovações estacionárias e o segundo não.

Tendo discutido a questão da estacionariedade, da integração e das tendências, a questão que resta é: como, defronte aos dados, como podemos testar todas essas características?

Testes estatísticos onde a hipótese nula é que uma série temporal é não-estacionária, contra a alternativa de que é estacionária, são chamados de testes de raiz unitária. Esse nome vem do fato de que processos *ARMA*[30] são não-estacionários quando o seu polinômio característico tem uma raiz que cai fora do círculo unitário. Por exemplo, o modelo *AR(1)* que definimos anteriormente na equação 2.10, é não-estacionário quando $\alpha = 1$. O polinômio característico desse processo *AR(1)* é $1 - \alpha x$, de maneira que sua raiz é $1/\alpha$, que ficará exatamente sobre o círculo unitário quando $\alpha = 1$. Justamente para $\alpha = 1$, esse processo *AR(1)* se torna o modelo do passeio aleatório que já discutimos:

$$y_t = c + y_{t-1} + \varepsilon_t \tag{2.14}$$

onde $\varepsilon_t \sim iid(0, \sigma^2)$. Para testar se $\alpha = 1$ não basta estimar α através de uma regressão e fazer um teste-*t* simples, pois o resultado será extremamente viesado caso o processo seja não-estacionário[31]. A alternativa é tirar a primeira diferença da equação 2.14, reescrevendo como:

$$\Delta y_t = c + (\alpha - 1) y_{t-1} + \varepsilon_t \tag{2.15}$$

e essa representação permite ver que como testar a hipótese nula de que $\alpha = 1$: fazendo uma regressão de Δy_t contra uma constante e y_{t-1}, e depois testando se o coeficiente de y_{t-1} é significativamente diferente de zero.

Esse tipo de teste é chamado de teste de Dickey-Fuller (DF). Dickey e Fuller mostraram em 1979 que os testes-*t* padrão baseados nessa equação acima são viesados, e que os valores críticos devem ser aumentados por um fator que depende do tamanho da amostra.

Após aplicar um teste de *Dickey-Fuller* , se não tivermos argumento para rejeitar a hipótese nula de não-estacionariedade, não podemos concluir imediatamente que o processo é *I(1)*. Isto porque ele pode ser *I(2)* ou até integrado de uma ordem superior. Assim, devemos mudar hipótese nula para $H_0 : y_t \sim I(2)$ e testá-la contra $H_1 : y_t \sim I(1)$ seguindo o mesmo procedimento anterior, apenas tirando mais uma diferença. Em termos práticos, esperamos que boa parte das séries temporais de preços de ações (o objeto de estudo no capítulo 4 deste trabalho) sejam não-estacionárias, integradas de ordem 1.

A evolução do teste *DF* deu origem ao teste de *Dickey-Fuller aumentado* (*ADF*)[32]. A diferença é que nesse teste se adicionam *lags* das variáveis dependentes à regressão *DF*. O

[30] Processos *ARMA* são um modelo geral de séries temporais univariadas, com termos tanto autoregressivos (*AR*) quanto de médias móveis (*MA*). Para detalhes, vide Alexander (2001), pp. 329-339, ou Kennedy (2003), pp. 320-323 e pp. 341-345.

[31] Como discute Kennedy (2003), pp. 319, essa questão provocou intensas discussões entre econometristas e estatísticos, quando estes apontaram que para variáveis não-estacionárias os resultados das regressões lineares tradicionais eram espúrios, levando a conclusões errôneas.

número de *lags* incluído deve ser o suficiente para remover a autocorrelação entre os erros[33], de maneira que uma regressão por mínimos quadrados ordinários (*OLS*) resulta em uma estimativa não-viesada do coeficiente de y_{t-1}. Os valores críticos para o teste *ADF* são marginalmente diferentes, mas de resto o princípio básico de testar a significância do coeficiente de y_{t-1} é similar ao teste *DF*: Δy_t é regredido contra uma constante, y_{t-1} e *m lags* de Δy_t, e as hipóteses $H_0 : y_t \sim I(1)$ contra $H_1 : y_t \sim I(0)$ são testadas, sendo equivalente a $H_0 : \beta = 0$ contra $H_1 : \beta < 0$ em um modelo do tipo:

$$\Delta y_t = c + \beta y_{t-1} + \alpha_1 \Delta y_{t-1} + \ldots + \alpha_m \Delta y_{t-m} + \varepsilon_t \tag{2.16}$$

e a estatística de teste é dada por:

$$ADF_t = \frac{\hat{\beta}}{SE(\beta)} \tag{2.17}$$

Além dos testes da família *Dickey-Fuller*, também há um conjunto de testes de raiz unitária desenvolvidos por Phillips e Perron[34], que diferem dos testes do tipo *DF* principalmente na maneira como tratam autocorrelação e heterocedasticidade nos erros. Ao invés de modificar a estrutura da regressão de teste, os testes *PP* modificam diretamente as estatísticas de teste para incluir autocorrelações e heteroscedasticidade. Por isso, há duas vantagens nesse tipo de teste, que vêm se tornando cada vez mais popular: a primeira é que os testes *PP* são robustos quanto à forma de heteroscedasticidade presente nos erros ε_t, o que é importante em séries temporais financeiras onde esse efeito está normalmente presente, e segundo, não é necessário especificar o número de *lags* na regressão de teste. No capítulo 4, quando testaremos para raiz unitária os dados de preços de ações, vamos aplicar tanto testes *ADF* quanto testes *PP*, o primeiro por ser mais conhecido e presente na literatura, o segundo por se adequar bem ao tipo de dado com o qual lidaremos e por ser bastante poderoso.

2.3.3. Equilíbrio de Longo Prazo e Tendências Comuns

Agora que discutimos extensamente a questão da estacionariedade de séries temporais, vamos retomar nosso foco, que é a discussão sobre cointegração. Uma primeira constatação importante a fazer, baseado em Alexander (2003) é que quando os preços dos ativos seguem passeios aleatórios, durante um período de tempo eles podem estar virtualmente em qualquer

---

[32] *Augmented Dickey-Fuller*.

[33] Essa questão da escolha do número de *lags* para o teste *ADF* está discutida em Zivot & Wang (2003), pp. 114-120, sendo apresentada a metodologia de Ng e Perron para resolvê-la.

[34] Esses testes estão apresentados em Zivot & Wang (2003), pp. 120-123.

ponto, porque um passeio aleatório tem variância infinita. Não faz muito sentido em modelar esses preços individualmente[35], visto que a melhor estimativa do preço amanhã é o preço hoje mais um eventual *drift*. Contudo, quando dois ou mais ativos são cointegrados, um modelo multivariado pode ser útil na medida em que revele equilíbrios de longo prazo no sistema, que talvez não sejam claros à primeira vista. Por exemplo, se um *spread* sabidamente reverte à média, podemos dizer que, em alguns anos à frente, onde a série temporal de um ativo estiver, a do outro ativo estará junto.

Os logaritmos de preços de ativos cointegrados seguem uma tendência estocástica comum[36]. Eles estão "ligados" no longo prazo, mesmo que no curto prazo possam se "separar", pois o *spread*, ou diferença entre eles, ou ainda algum outro tipo de combinação linear, reverte à média. Um exemplo simples que mostra porque séries cointegradas têm uma tendência estocástica comum é:

$$\begin{aligned} x_t &= w_t + \varepsilon_{xt} \\ y_t &= w_t + \varepsilon_{yt} \\ w_t &= w_{t-1} + \varepsilon_t \end{aligned} \tag{2.18}$$

onde todos os erros são *iid* e independentes uns dos outros. Nesse conjunto de equações, x_t e y_t são *I(1)*, mas a diferença (*spread*) entre elas, $x_t - y_t$, é *I(0)*. Essas duas variáveis têm uma tendência estocástica comum dada pelo passeio aleatório w_t. Note que a correlação entre Δx e Δy será menor que 1, e mais ainda, quando as variâncias de ε_{xt} e de ε_{yt} forem significativamente maiores que a variância de ε_t, essa correlação vai ser baixa[37], nos levando de volta ao ponto que afirmamos anteriormente: a correlação é uma ferramenta pouco poderosa quando trabalhamos com preços de ativos, podendo inclusive levar a conclusões enganosas. Claramente o modelo acima é estilizado, e é pouco provável que na prática esse tipo de situação ocorra, mas a ilustração é importante.

A combinação linear de variáveis *I(1)* que é estacionária é comumente denotada por z, e é chamada de *termo de desequilíbrio*, pois captura os desvios do equilíbrio de longo prazo entre as séries, no contexto de um modelo de correção de erros (*ECM*[38]), que vamos discutir mais adiante no item 2.3.7. A expectativa de z dá a relação de equilíbrio de longo prazo entre as variáveis x e y, e períodos de desequilíbrio temporário ocorrem no curto prazo, a medida que

[35] Embora o autor o faça em Parreiras (2003), a questão aqui envolve a distinção entre preços de ativos e preços de conjuntos de ativos e a aplicação de cointegração.

[36] Alexander (2003), pp. 350-353.

[37] A prova dessa afirmação é intuitiva e está feita em Alexander (2003), pp. 351.

[38] Do inglês *Error Correction Model*. Vamos adotar a sigla em inglês por simplicidade.

z varia em torno de seu valor esperado. O chamado *vetor de cointegração* é o vetor de pesos de z. Assim, no caso de uma relação de cointegração entre duas variáveis *I(1)* x e y, onde $x - \alpha y \sim I(0)$, o *vetor de cointegração* é dado por $(1, -\alpha)$. Quando apenas duas séries temporais estão incluídas no modelo, pode haver no máximo um vetor, pois se houvesse mais de um as séries originais deveriam ser estacionárias.

Genericamente, existe cointegração entre *n* séries temporais integradas se existe, no mínimo, um vetor de cointegração, ou seja, pelo menos uma combinação linear das séries *I(1)* que seja estacionária. Usando os termos de Alexander (2003), essa combinação linear atua como uma "cola" no sistema de séries temporais, e quanto mais vetores de cointegração existiram, mais forte é a co-dependência entre elas. A seguir, vamos analisar qual a metodologia utilizada para determinar a existência desse *vetor de cointegração*, e o que é possível fazer com ele.

2.3.4. Testes para Cointegração

O primeiro passo numa análise de cointegração envolve o uso de testes estatísticos para achar combinações lineares estacionárias entre as séries integradas, de modo a definir as relações de equilíbrio de longo prazo entre o conjunto de variáveis no sistema, se é que tais relações existem. Obviamente, se não existirem, então as variáveis não são cointegradas e não há muito sentido em seguir adiante.

Os *papers* clássicos sobre cointegração são de Robert Engle e Clive Granger[39], escritos entre 1986 e 87[40]. Neles, os autores propõem um teste para cointegração baseado em uma regressão linear ordinária, ou seja, na *metodologia Engle-Granger* simplesmente regredimos uma variável integrada contra as outras variáveis integradas, e testamos os resíduos para estacionariedade, usando um teste de raiz unitária como aquele que já discutimos nesse capítulo, com a ressalva de que os valores críticos são um pouco diferentes.

O leitor mais atento vai notar que dissemos anteriormente que não se devem usar regressões lineares ordinárias em dados não-estacionários. Se a variável dependente é não-estacionária, é bastante provável que os resíduos da regressão também o sejam, e as propriedades de estimadores por mínimos quadrados apenas estão bem estabelecidas para resíduos estacionários. No entanto, há uma circunstância em que uma regressão entre variáveis não-estacionárias resulta em resíduos estacionários: justamente quando estas variáveis são cointegradas. Em outras palavras: só é possível fazer regressões de preços de ativos (na verdade,

[39] Os dois dividiram o Prêmio Nobel de Economia de 2003. A contribuição pela qual Engle foi premiado foi o desenvolvimento dos métodos *ARCH*, enquanto Granger ganhou pelo desenvolvimento das técnicas de cointegração. Mais informações em http://nobelprize.org/nobel_prizes/economics/laureates/index.html.

[40] Entre outros, temos Engle, R.; Granger, C. *Co-integration and error correction: representation, estimation and testing*, In: Econometrica 55, pp. 251-276, 1987.

de seus logs) contra preços de outros ativos quando as variáveis são cointegradas e nesse caso a regressão define o equilíbrio de longo prazo entre as variáveis.

Além da metodologia de *Engle-Granger*, existe também um conjunto de técnicas desenvolvido pouco depois pelo dinamarquês Søren Johansen, conhecidas como *metodologia de Johansen*, que são mais abrangentes que Engle-Granger, pois utiliza uma função que tem propriedades mais interessantes, além de produzir menos viés quando o numero de variáveis envolvidas é maior que dois (vide Alexander (2003), pp.357-361). Os testes de Johansen são baseados nos autovalores de uma matriz estocástica, e buscam a combinação linear de variáveis que é mais estacionária, enquanto que os testes de Engle-Granger buscam a combinação que tem menor variância. A seguir vamos descrever com mais detalhes ambas as metodologias, buscando explicar os pontos fortes e fracos de cada uma.

2.3.5. A Metodologia de Engle-Granger

O teste de Engle-Granger é um processo com dois passos: primeiro se estima uma regressão linear ordinária nos dados das variáveis *I(1)*, e depois se aplica um teste de estacionariedade nos resíduos dessa regressão. Os valores críticos para esse teste são dados por MacKinnon (1991), e estão implementados na maioria dos pacotes computacionais para econometria. Para o caso de apenas duas variáveis, x e y, a regressão de Engle-Granger tem a seguinte equação:

$$x_t = c + \alpha y_t + \varepsilon_t \tag{2.19}$$

Temos que x e y serão cointegrados se, e somente se, ε for estacionário. Então teremos um vetor de cointegração dado por $(1,-\alpha)$, e o equilíbrio de longo prazo entre as duas variáveis é dado por $x = c + \alpha y$. È importante notar que testes de cointegração não produzirão bons resultados se for usada uma janela de tempo muito curta: o poder desta técnica vem justamente do fato de detectar tendências comuns no longo prazo.

Para o caso mais geral, uma regressão linear ordinária entre *n* variáveis *I(1)* cointegradas vai estimar uma combinação linear dessas séries que é estacionária. O vetor de cointegração será dado por $(1,-\beta_1,\ldots,-\beta_{n-1})$, onde $\beta_1,\ldots,\beta_{n-1}$ são os coeficientes para as *n-1* variáveis *I(1)* que são usadas como variáveis explicativas, com a outra variável *I(1)* restante sendo usada como variável dependente na regressão de Engle-Granger. O termo de desequilíbrio z contém os resíduos dessa regressão.

Quando *n=2*, não importa qual variável é tomada como dependente. Existe apenas um vetor de cointegração, que é o mesmo quando estimamos uma regressão de x em y e quando estimamos a regressão de y em x. O problema ocorre justamente quando temos mais de duas

séries *I(1)*, e aqui a metodologia de Engle-Granger pode sofrer um viés sério. Isso quer dizer que diferentes estimativas do vetor de cointegração aparecem conforme a variável dependente que se define, e apenas uma estimativa pode ser obtida, mesmo que existam até *n-1* possíveis vetores de cointegração. Por isso essa metodologia supostamente não pode ser usada para identificar todos os vetores de cointegração presentes em um sistema com mais de duas variáveis cointegradas[41].

Segundo Alexander (2003), a metodologia de Engle-Granger é aplicável a sistemas com mais de duas variáveis em algumas circunstâncias especiais, basicamente quando se sabe qual variável deve ser usada como dependente, e qual é o equilíbrio de longo prazo mais apropriado. No capítulo 4, antes de aplicarmos a metodologia aqui descrita, vamos examinar brevemente essa questão, seguindo a discussão feita por Burgess (2003). Antes disso contudo, vamos examinar a metodologia de Johansen para compreender de onde vem o maior poder dessa técnica.

2.3.6. A Metodologia de Johansen

A metodologia de Johansen para cointegração é baseada nos autovalores de uma matriz estocástica, na verdade se reduzindo a um problema de correlação canônica, semelhante ao que está presente em análise de componentes principais[42]. O teste de Johansen busca encontrar a combinação linear de variáveis que é mais estacionária, enquando que o teste de Engle-Granger, ao se basear em regressões lineares ordinárias, busca a combinação linear que minimiza a variância.

De fato, os testes de Johansen são uma generalização dos testes de raiz unitária que discutimos anteriormente. Ali a idéia era fazer uma regressão da primeira diferença Δy_t contra seu *lag* y_{t-1}. Assim, o teste se baseia no fato de que o coeficiente do *lag* deve ser zero se o processo tem raiz unitária. Generalizando esse argumento para um processo do tipo *VAR(1)*, temos a motivação para o teste de Johansen para uma tendência estocástica comum (ou seja, cointegração). O modelo *VAR(1)* pode ser escrito como:

$$\Delta y_t = \alpha_0 + (A - I) y_{t-1} + \varepsilon_t \tag{2.20}$$

[41] Alexander (2003), pp.356-361, aplica a metodologia de Engle-Granger e depois a compara com a metodologia de Johansen, para um exemplo com 12 séries de preços de diferentes contratos futuros na estrutura a termo do petróleo *WTI*, negociados na *NYMEX*.

[42] Análise de Componentes Principais, ou *PCA*, é uma importante ferramenta para lidar com sistemas multivariados onde há presença de colinearidade entre os retornos. Ela permite extrair as principais fontes não-correlacionadas de variação de um sistema. Para detalhes, vide Alexander (2003), pp. 143-178, Zivot & Wang (2003), pp. 571-584, ou ainda Haykin (2001), cap. 8, pp. 429-480.

Se cada uma das variáveis nesse vetor y for *I(1)*, então cada uma das equações no sistema vetorial acima terá uma variável estacionária do lado esquerdo. Os erros são estacionários, e portanto cada termo no vetor $(A-I)y_{t-1}$ deve ser estacionário, de modo que a equação seja balanceada. Na verdade, para cada um dos *r* termos em $(A-I)$ que for linearmente independente, teremos uma tendência estocástica comum entre as y variáveis. Sabemos que cada uma das *r* relações de independência linear corresponde a um autovalor da matriz $(A-I)$, e portanto o teste de Johansen vai consistir em testar o número de autovalores não-zero da matriz $(A-I)$. Nesse sentido é uma generalização vetorial do teste de raiz unitária, usando argumentos de álgebra linear.

Vale dizer que o modelo *VAR(1)* que definimos acima pode não ser o mais apropriado para o processo subjacente aos dados. Lembre-se do teste *ADF*, onde incluímos mais *lags* na regressão Dickey-Fuller padrão para dar conta da autocorrelação nos resíduos, ou ainda da discussão acerca de tendências determinísticas. A mesma idéia vale para a metodologia de Johansen: é possível aumentar o modelo básico para incluir esses termos. Assim, podemos ter um modelo de ordem superior *VAR(p)*, com equação:

$$\Delta y_t = \alpha_0 + (A_1 - I)y_{t-1} + (A_1 + A_2 - I)y_{t-2} + \ldots + (A_1 + \ldots + A_p - I)y_{t-p} + \varepsilon_t \quad (2.21)$$

e o teste de Johansen se torna um teste para o número de autovalores não-zero na matriz:

$$\Pi = A_1 + A_2 + \ldots + A_p - I \quad (2.22)$$

Alexander (2003), com base em Johansen e Juselius (1990)[43] recomenda o uso de um teste de "traço" para o número *r* de autovalores não-zero na matriz Π. A hipótese é dada por: $H_0 : r \leq R$ contra $H_1 : r > R$, e a estatística de teste por:

$$Tr = -T \sum_{i=R+1}^{n} \ln\left(1 - \hat{\lambda}_i\right) \quad (2.23)$$

onde T é o tamanho da amostra, *n* é o número de variáveis no sistema e λ são os autovalores de Π, com $0 \leq \lambda < 1$, ordenados de maneira que $\hat{\lambda}_1 > \hat{\lambda}_2 > \ldots > \hat{\lambda}_n$. Assim, a medida em que *R* cresce, a estatística do "traço" decresce. A metodologia de Johansen primeiro calcula os autovalores da matriz e depois calcula a estatística para todo *R* entre *0* e *n-1*. Os valores críticos estão dados em Johansen e Juselius (1990), e são dependentes da especificação do modelo *VAR*, do número de *lags*, e se inclui constante e tendência.

Em resumo, a metodologia de Johansen é mais informativa que Engle-Granger nos casos em que o sistema consiste de mais de duas variáveis e onde a variável dependente não é dada naturalmente (casos típicos, discutidos em Alexander (2003), são o da estrutura a termo de taxas

[43] Johansen, S.; Juselius, K. *Maximum likelihood estimation and inference on cointegration – with applications to the demand for money*, In: Oxford Bulletion of Economics and Statistics, 52(2), pp. 169-210, 1990.

de juros e da estrutura a termo de contratos futuros de petróleo). Johansen obtém todos as possíveis relações de cointegração, enquanto Engle-Granger obtém apenas uma. Ainda assim, em muitos problemas de finanças o uso de Engle-Granger pode ser justificado com base em:

- sua maior simplicidade (os resultados têm interpretações intuitivas);
- em termos de gestão de risco, muitas vezes é mais importante ter como critério a mínima variância (justamente o critério de Engle-Granger) do que a máxima estacionariedade (o critério de Johansen);
- para muitas aplicações de finanças, a variável dependente é dada naturalmente (vamos discutir essa questão adiante no Capítulo 4);

Além das duas metodologias que discutimos aqui, há alguns outros testes para cointegração, como o de Phillips e Ouliaris, que envolve um teste de duas etapas nos resíduos de uma regressão, e o teste Engle e Yoo, que envolve a significância dos termos do modelo de correção de erros. Apropriadamente, vamos a seguir apresentar justamente o *ECM*, ou modelo de correção de erros, justamente a técnica que vai permitir transformar os conceitos de cointegração em decisões de compra e venda nos mercados financeiros.

2.3.7. Correção de Erros

O mecanismo que "dá liga" a duas séries temporais cointegradas é a chamada causalidade. O termo não significa que mudanças estruturais em uma série levam a mudanças na outra. O que essa causalidade diz é que mudanças em uma série precedem mudanças na outra (mudanças no sentido da dinâmica temporal, não estrutural). Esse conceito na verdade é chamado de "causalidade de Granger"[44]. Quando séries temporais são cointegradas, existe no sistema algum fluxo causal de Granger, ou seja, uma relação de *leads* e *lags* entre as variáveis.

O chamado "teorema da representação de Granger" afirma que um modelo *VAR* nas diferenças de variáveis *I(1)* será mal-especificado se as variáveis forem cointegradas. Engle e Granger mostraram que uma especificação de equilíbrio estaria faltando do modelo *VAR(p)* genérico, mas que a inclusão de variáveis explicativas baseadas nos *lags* dos termos de desequilíbrio tornaria o modelo bem-especificado. Esse tipo de modelo é justamente chamado de um *modelo de correção de erros* (um *ECM*), porque inclui um mecanismo de *feedback negativo* (ou auto-regulação), onde os desvios em relação ao equilíbrio de longo prazo são automaticamente corrigidos. Automaticamente aqui quer dizer "sem a interferência variáveis exógenas" e não "rapidamente", como veremos adiante.

O *ECM* é um modelo dinâmico nas primeiras diferenças das variáveis *I(1)* que foram usadas na equação de cointegração. Portanto, se foi determinado que os *logs* dos preços são

[44] Para detalhes, vide Alexander (2003), pp. 344-346.

cointegrados, e o vetor de cointegração foi baseado neles, então o *ECM* será um modelo dinâmico das correlações dos retornos, e as estatíticas-*t* dos seus coeficientes estimados traram muita informação acerca das relações de *leads* e *lags* entre as variáveis (ou seja, qual será o sentido da causalidade de Granger). É importante diferenciar o papel do *ECM* das técnicas que discutimos anteriormente: ele analisa os desvios de curto prazo em relação ao equilíbrio, enquanto as metodologias de Engle-Granger e Johansen abrem as portas para determinação dos equilíbrios de longo prazo entre as variáveis. A relação entre as duas etapas é o z, o *termo de desequilíbrio*, que é determinado na primeira etapa e subseqüentemente entra no *ECM*.

O nome "correção de erros" vem do fato de o modelo ser construído de modo que desvios de curto prazo em relação ao equilíbrio de longo prazo sejam corrigidos. Vamos ilustrar essa dinâmica usando duas séries de *logs* de preços, x e y, cointegradas. O *ECM* terá a seguinte forma:

$$\begin{aligned}\Delta x_t &= \alpha_1 + \sum_{i=1}^{m_1} \beta_{1i} \Delta x_{t-i} + \sum_{i=1}^{m_2} \beta_{2i} \Delta y_{t-i} + \gamma_1 z_{t-1} + \varepsilon_{1t} \\ \Delta y_t &= \alpha_2 + \sum_{i=1}^{m_3} \beta_{3i} \Delta x_{t-i} + \sum_{i=1}^{m_4} \beta_{4i} \Delta y_{t-i} + \gamma_2 z_{t-1} + \varepsilon_{2t}\end{aligned} \tag{2.24}$$

onde Δ denota a primeira diferença, $z = x - \alpha y$ é o termo de desequilíbrio e os *lags* e os coeficientes são determinados por regressões lineares ordinárias.

Vamos supor um caso em que $\alpha > 0$. O modelo acima somente será um *ECM* se tivermos $\gamma_1 < 0$ e $\gamma_2 > 0$, pois nesse caso o último termo de cada uma das equações fará com que os desvios do equilíbrio de longo prazo sejam corrigidos. Exemplificando: suponha z grande e positivo. A única maneira de fazer com que x convirja pro equilíbrio de longo prazo é tendo γ_1 negativo, e vice-versa no caso de y. Os coeficientes γ_i determinam a velocidade do ajuste ao equilíbrio, após um choque de mercado. Quando esses coeficientes são altos, os ajustes são velozes e portanto z será altamente estacionário. De fato, um teste para cointegração desenvolvido por Engle e Yoo, que mencionamos anteriormente, é baseado na significância desses termos γ_i.

Quando os logs dos preços de dois ativos x e y são cointegrados, o *ECM* vai capturar as correlações dinâmicas e a causalidade entre os retornos de ambos. Se os coeficientes dos *lags* dos retornos de y na equação para x forem significativos, então pontos onde y muda de comportamento (por exemplo, o preço estava em tendência de alta e passou para uma tendência de baixa) precederão pontos onde o comportamento de x muda. Nesse caso, diz-se que y causa (em termos de Granger) x. Quando um *spread* reverte à média, deve existir uma relação de causalidade entre os preços, mas a direção dessa causalidade (quem causa quem) pode variar no tempo.

A generalização do *ECM* para mais de duas variáveis é intuitiva. O *ECM* contém uma equação para cada uma das variáveis de um sistema, onde a variável dependente está na primeira diferença, e cada equação contém as mesmas variáveis independentes: *lags* da primeira diferença até uma ordem *p*, e *r* termos de desequilíbrio correspondendo a *r* vetores de cointegração. Portanto, a especificação completa (e compacta) de um *VECM* (um *ECM* vetorial) fica:

$$\Delta y_t = \alpha_0 + B_1 \Delta y_{t-1} + B_2 \Delta y_{t-2} + \ldots + B_p \Delta y_{t-p} + \Pi y_{t-1} + \varepsilon_t \qquad (2.25)$$

Cada uma das *n* equações no modelo acima tem como regressores uma constante, *p lags* da primeira diferença de y, e todos os *lags* dos termos de desequilíbrio em Πy_{t-1}. Para *p* muito grande, há uma enormidade de regressores potenciais, e é bastante improvável que todos venham a ser significativos em uma regressão linear ordinária, por isso alguma parcimônia na especificação do modelo é recomendada.

Exemplos de construção de *(V)ECM*s na literatura são vários. Alexander (2003), pp.363-364 constrói um *ECM* para a relação entre índices de ações na Alemanha (o *DAX*), na França (o *CAC*) e na Holanda (*o AEX*), usando Johansen. A mesma autora também faz uma análise interessante entre os preços *spot* (à vista) e futuro no mercado de petróleo em Alexander (2003), pp. 365-367. Burgess (2000), também analisa *ECM*s aplicados a índices europeus de ações, inclusive desenvolvendo regras para a montagem de posições, em linha com o que discutimos anteriormente nesse capítulo sobre estratégias *long-short*. Zivot & Wang (2003) também discutem extensamente a construção de modelos de correção de erros, aplicando-os a preços de ações e à taxa de câmbio do dólar canadense.

3. REDES NEURAIS E PREVISÃO EM FINANÇAS

"É interessante notar que enquanto há relatos que alguns golfinhos aprenderam palavras – até cinqüenta palavras utilizadas no contexto correto – não há relatos de seres humanos terem aprendido golfinês."
Carl Sagan[1]

3.1. Redes Neurais – Aspectos Teóricos

3.1.1.Introdução

O trabalho em redes neurais artificiais, normalmente chamadas apenas de redes neurais, tem sido motivado desde o começo pelo reconhecimento de que o cérebro processa informações de um modo inteiramente diferente do computador digital convencional. O cérebro é um "computador" (sistema de processamento de informação) altamente complexo, não-linear e paralelo. Ele tem a capacidade de organizar seus constituintes estruturais, chamados de neurônios, de forma a realizar determinados processamentos (tarefas como reconhecimento de padrões, percepção, controle motor) muito mais rapidamente que o mais rápido computador digital existente. A visão humana é um exemplo interessante: a função do sistema visual é fornecer uma representação do ambiente à nossa volta, e fornecer a informação necessária para interagir com este ambiente. Mais especificamente, o cérebro realiza rotineiramente tarefas de reconhecimento perceptivo (por exemplo, reconhecer um rosto familiar em uma cena não-familiar) em aproximadamente 100-200 ms, enquanto tarefas de complexidade muito menor podem levar dias para serem executadas em um computador convencional.

Como é possível que o cérebro humano faça isso? No momento do nascimento, um cérebro tem uma grande estrutura e a habilidade de desenvolver suas próprias regras através daquilo que é comumente denominado de "experiência". Na verdade, a experiência vai sendo acumulada com o tempo, sendo que o mais dramático desenvolvimento (i.e., por ligações físicas) do cérebro humano acontece durante os dois primeiros anos de vida; de todo modo, o desenvolvimento continua por muito mais tempo além disso.

Um neurônio em desenvolvimento é sinônimo de um cérebro plástico: a plasticidade permite que o sistema nervoso em desenvolvimento se adapte ao seu meio ambiente. Assim como o processamento de informação do cérebro humano, também ela o é com relação às redes neurais construídas com neurônios artificiais. Na sua forma mais geral, uma rede neural é uma máquina projetada para modelar a maneira como o cérebro realiza uma tarefa particular ou função de interesse; a rede é normalmente implementada usando-se componentes eletrônicos ou

[1] *Apud* Rezende, 2003.

é simulada por programação em um computador digital. Para alcançarem bom desempenho, as redes neurais empregam uma interligação maciça de células computacionais simples denominadas neurônios, ou unidades de processamento. É possível então dar a seguinte definição de uma rede neural, vista como uma máquina adaptativa[2]: *Uma rede neural é um processador paralelo maciçamente distribuído, constituído de unidades de processamento simples. Assemelha-se ao cérebro em dois pontos: (i) o conhecimento do ambiente externo é incorporado à rede via um processo de aprendizagem e (ii) forças de conexão entre os neurônios são usadas para armazenar esse conhecimento.*

O método utilizado para realizar o processo de aprendizagem é chamado de algoritmo de aprendizagem, cuja função é modificar as forças de conexão (também chamadas de pesos sinápticos) da rede de forma ordenada para alcançar um determinado objetivo desejado.

A modificação dos pesos sinápticos é o procedimento tradicional para o projeto de redes neurais. Entretanto, é possível também para uma rede neural modificar sua própria topologia, o que é motivado pelo fato de os neurônios no cérebro humano poderem morrer e novas conexões poderem nascer[3].

É evidente que as redes neurais extraem seu poder computacional, primeiro, de sua estrutura paralela maciçamente distribuída e segundo, de sua habilidade de aprender e portanto de generalizar. A generalização se refere ao fato de a rede neural produzir saídas adequadas para entradas que não estavam presentes durante o treinamento (aprendizagem). Estas duas capacidades de processamento de informação tornam possível para as redes resolver problemas complexos de grande escala, que são normalmente intratáveis por métodos tradicionais.

O uso de redes neurais oferece as seguintes propriedades úteis:

- *Não-linearidade*: um neurônio artificial pode ser linear ou não linear. Uma rede neural, constituída por conexões de neurônios não-lineares, é ela mesma não-linear. Além disso, a não-linearidade está distribuída por toda a rede. Essa é uma propriedade muito importante, particularmente se o mecanismo responsável pela geração do sinal de entrada for inerentemente não-linear[4].
- *Mapeamento de Entrada-Saída*: um paradigma popular de aprendizagem chamado aprendizagem com professor ou aprendizagem supervisionada envolve a modificação dos pesos sinápticos de uma rede neural pela aplicação de um conjunto de amostras de treinamento rotuladas ou exemplos da tarefa. Cada exemplo consiste de um sinal de entrada único e de uma resposta desejada correspondente. Apresenta-se para a rede um exemplo escolhido ao acaso do conjunto, e os pesos sinápticos (parâmetros livres) da rede

[2] Definição adaptada de Haykin (2001), que por sua vez é adaptada de Aleksander e Morton (1990).

[3] Essa é a distinção básica entre treinamento supervisionado e não-supervisionado, conforme se verá adiante.

[4] É possível dizer que as séries temporais de retornos de ativos financeiros são inerentemente não-lineares. Para uma formulação interessante do problema, vide Sornette (2003).

são modificados para minimizar a diferença entre a resposta desejada e a resposta real da rede, produzida pelo sinal de entrada, de acordo com um critério estatístico apropriado. O treinamento da rede é reproduzido para muitos exemplos do conjunto, até que a rede alcance um estado estável onde não haja mais modificações significativas nos pesos sinápticos. Os exemplos de treinamento previamente aplicados podem ser reaplicados durante a sessão de treinamento, mas em ordem diferente. Assim, a rede aprende dos exemplos ao construir um mapeamento entrada-saída para o problema considerado. Há uma analogia próxima entre o processo de mapeamento entrada-saída e o conceito de inferência estatística não-paramétrica.

- *Adaptabilidade*: as redes neurais têm uma capacidade inata de adaptar seus pesos sinápticos a modificações do meio ambiente. Em particular, uma rede neural treinada para operar em um ambiente específico pode ser facilmente retreinada para lidar com pequenas modificações nas condições operativas do meio ambiente. Além disso, quando está em um ambiente não-estacionário (i.e.,onde as estatísticas mudam com o tempo), uma rede neural pode ser projetada para modificar os seus pesos sinápticos em tempo real. A arquitetura natural de uma rede neural para classificação de padrões, processamento de sinais e aplicações de controle, aliada à capacidade de adaptação da rede, a torna uma ferramenta muito útil para classificação adaptativa de padrões, processamento adaptativo de sinais e controle adaptativo. Como regra geral, pode-se dizer que quanto mais adaptativo se fizer um sistema, assegurando que ele permaneça estável, mais robusto será o seu desempenho quando o sistema for exigido a operar em um ambiente não-estacionário. Contudo, deve ser enfatizado, que adaptabilidade nem sempre resulta em robustez, e na verdade pode resultar no contrário. Um sistema adaptativo com constantes de tempo pequenas, por exemplo, pode se modificar rapidamente e assim tender a responder a perturbações espúrias, causando uma drástica degradação no desempenho do sistema. Para aproveitar todos os benefícios da adaptabilidade, as constantes de tempo principais do sistema devem ser grandes o suficiente par que o sistema ignore perturbações espúrias mas ainda assim pequenas o suficiente para responder a mudanças significativas no ambiente –esse trade-off normalmente é chamado de dilema estabilidade-plasticidade.
- *Resposta a Evidências*: no contexto de classificação de padrões, uma rede neural pode ser projetada para fornecer informação não somente sobre qual padrão particular selecionar, mas também sobre a confiança ou crença na decisão tomada.Esta última informação pode ser usada para rejeitar padrões ambíguos, caso eles estejam presentes, e com isso melhorar o desempenho de classificação da rede.
- *Informação Contextual*: o conhecimento é representado pela própria estrutura e estado de ativação da rede neural. Cada neurônio da rede é potencialmente afetado pela atividade de

todos os outros neurônios da rede. Conseqüentemente, a informação contextual é tratada naturalmente pela rede.

- *Tolerância a Falhas*: uma rede neural tem o potencial de ser inerentemente tolerante a falhas, ou capaz de realizar computação robusta, no sentido de que seu desempenho se degrada suavemente sob condições de operação adversas. Se um neurônio ou suas conexões é danificado, por exemplo, a recuperação de um padrão armazenado é prejudicada em qualidade. Contudo, devido à natureza distribuída da informação armazenada na rede, o dano deve ser extenso para que a resposta global da rede seja degradada seriamente. Assim, a principio, uma rede neural exibe uma degradação suave do desempenho em vez de apresentar uma falha catastrófica. Há algumas evidências para a computação robusta, mas geralmente ela não é controlada (Haykin, 2001). Para se assegurar que uma rede neural seja de fato tolerante é necessário adotar-se medidas corretivas no projeto do algoritmo utilizado para treinar a rede.

3.1.2.Histórico

A era moderna das redes neurais começou com o trabalho pioneiro de McCulloch e Pitts (1943). McCulloch era um psiquiatra e neuroanatomista por treinamento, que passou 20 anos refletindo sobre a representação de um evento no sistema nervoso. Pitts era um matemático, que se associou a McCulloch em 1942. No se clássico artigo *"A logical calculus of the ideas immanent in nervous activity"*, eles descrevem um cálculo das redes neurais que unificava os estudos de neurofisiologia e lógica matemática. Eles assumiam que seu modelo formal de um neurônio seguia uma lei "tudo ou nada". Com um número suficiente dessas unidades simples e com conexões sinápticas ajustadas apropriadamente e operando de forma síncrona, os autores mostraram que uma rede assim constituída realizaria, a principio, a computação de qualquer função (computável). Este era um resultado muito significativo e com ele é geralmente aceito o nascimento das disciplinas de redes neurais e inteligência artificial.

O próximo desenvolvimento significativo das redes neurais veio em 1949, com a publicação do livro de Hebb "*The Organization of Behavior"*, no qual foi apresentada pela primeira vez uma formulação explicita de uma regra de aprendizagem fisiológica para a modificação sináptica. Especificamente, Hebb propôs que a conectividade do cérebro é continuamente modificada conforme um organismo vai aprendendo tarefas funcionais diferentes e que agrupamentos neurais são criados por tais modificações. Além disso, também apresentou o "postulado da aprendizagem", que afirma que a eficiência de uma sinapse variável entre dois neurônios é aumentada pela ativação de um neurônio pelo outro, através daquela sinapse.

Ao longo dos anos seguintes muitos grandes nomes da computação deram contribuições ao estudo das redes neurais. Cientistas como Minsky[5], Gabor[6] e Von Neumann[7] trabalharam em aspectos do problema. No entanto o avanço mais significativo foi alcançado em 1958 por Rosenblatt, com o desenvolvimento do *perceptron*, hoje o modelo básico de neurônio artificial. Além disso, Rosenblatt também introduziu o famoso *teorema de convergência do perceptron*[8]. Em 1960 Widrow e Hoff introduziram o algoritmo *LMS* (*Least Mean-Square*, ou mínimos quadrados médios, um algoritmo de aprendizagem para redes neurais simples), e o estudo e desenvolvimento de redes neurais ganharam forte impulso.

Mas então veio o livro de Minsky e Papert[9], que utilizaram a matemática para demonstrar que existem limites fundamentais para aquilo que os *perceptrons* de camada única podem calcular. Em uma breve seção sobre *perceptrons* de múltiplas camadas, eles afirmavam que não havia razão para supor que qualquer uma das limitações do *perceptron* de camada única poderia ser superada na versão de múltiplas camadas. Foi um banho de água fria em um crescente campo de pesquisa. O problema básico encontrado no projeto de um *perceptron* de múltiplas camadas é o problema de *atribuição de crédito* (i.e., o problema de atribuir aos neurônios escondidos da rede o crédito pelo erro). A conjunção do livro de Minsky e Papert com a ausência de uma idéia clara sobre como resolver esse problema fez com que as redes neurais ficassem adormecidas por mais de uma década.

Alguns desenvolvimentos relevantes aconteceram no fim dos anos 70 e inicio dos 80, como a criação dos *mapas auto-organizáveis* de Kohonen[10], e das *máquinas de Boltzmann*[11], mas foi apenas em 1986, com a publicação, por Rumelhart, Hinton e Williams do artigo "*Learning representations of back-propagation erros*"[12], que o estudo das redes neurais voltou a ganhar força. O trio conseguiu resolver definitivamente o problema de atribuição de crédito, deixando as objeções de Minsky e Papert para trás e o algoritmo de retropropagação emergiu como o paradigma básico para aprendizagem, sendo o mais popular e conhecido algoritmo para treinamento de redes de múltiplas camadas.

---

[5] Seu artigo de 1961 *"Steps toward Artificial Intelligence"* contém uma grande seção sobre redes neurais.

[6] Mais conhecido como inventor da holografia, foi um dos pioneiros da teoria da comunicação e propôs a idéia do *filtro adaptativo não-linear*.

[7] Um dos pais da computação, realizou em 1957 as famosas Palestras Silliman, postumamente publicadas no livro *The Computer and the Brain* (1958).

[8] Esse teorema garante que um perceptron de camada única consegue, após um número finito de iterações de treinamento, classificar dois conjuntos de dados linearmente separáveis.

[9] *Perceptrons*, MIT Press, 1969, republicado em 1988.

[10] Um tipo de rede neural com treinamento não-supervisionado.

[11] Redes inspiradas pela mecânica estatística, onde há aprendizado estocástico.

[12] Revista *Nature*, vol. 323, pp.533-536.

Em 1988, Broomhead e Lowe descreveram um procedimento para o projeto de redes alimentadas adiante, em camadas utilizando *funções de base radial*, que fornecem uma alternativa aos modelos de *perceptrons* de múltiplas camadas. No início dos anos 90, Vapnik e co-autores desenvolveram uma classe de redes de aprendizagem supervisionada poderosa do ponto de vista computacional, chamada de *máquinas de vetor de suporte*, para ser utilizada em reconhecimento de padrões, regressão e problemas de estimação.

Este é apenas um breve histórico dos principais desenvolvimentos no contexto das redes neurais. Em resumo, pode-se dizer que o trabalho pioneiro de McCulloch-Pitts e Rosenblatt constitui o cerne de uma primeira era das redes neurais, e o trabalho de Rumelhart e parceiros trouxe o ressurgimento do campo, para uma – ainda corrente – segunda era das redes neurais.

3.1.3.O Neurônio Artificial

Um *neurônio* é uma unidade de processamento de informação que é fundamental para a operação de uma rede neural. O diagrama em blocos da Figura 3.1 mostra o modelo de um neurônio, que forma a base para o projeto de redes neurais artificiais.

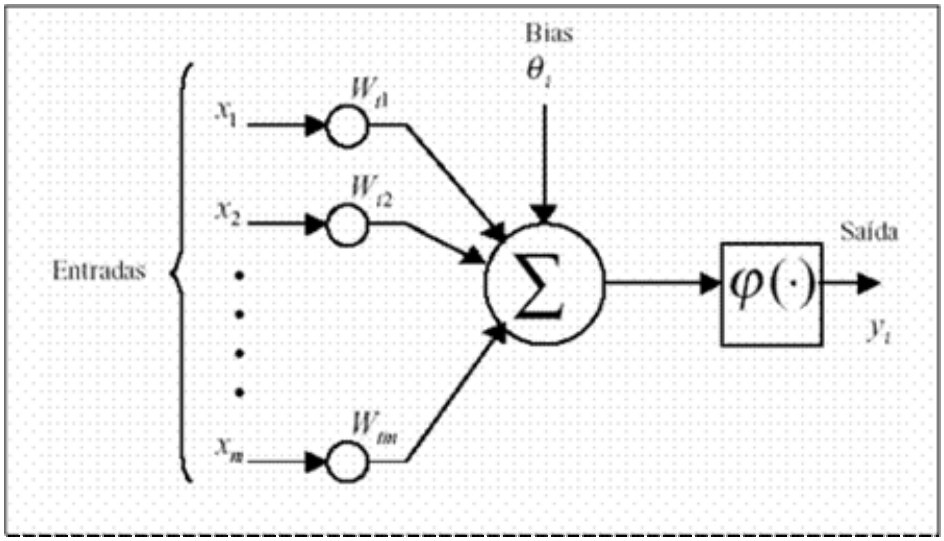

Figura 3.1: Modelo de um Neurônio Artificial (Haykin, 2001)

Há três elementos básicos no modelo neuronal:

- Um conjunto de sinapses ou elos de conexão, cada uma caracterizada por um peso próprio.Especificamente um sinal xj na entrada da sinapse j conectada ao neurônio k é multiplicado pelo peso sináptico wkj. É importante notar a maneira como são escritos os índices do peso sináptico wkj. O primeiro índice se refere ao neurônio em questão e o segundo se refere ao terminal de entrada da sinapse ao qual o peso se refere. Ao contrário

de uma sinapse do cérebro, o peso sináptico de um neurônio artificial pode estar em um intervalo que inclui valores negativos bem como positivos.

- Um somador para somar os sinais de entrada, ponderados pelas respectivas sinapses do neurônio; essas operações caracterizam um combinador linear.
- Uma função de ativação para restringir a amplitude da saída de um neurônio. A função de ativação é também referida como função restritiva já que restringe o intervalo permissível de amplitude do sinal de saída a um valor finito. Tipicamente o intervalo normalizado da amplitude da saída de um neurônio é escrito como o intervalo unitário fechado [0,1] ou alternativamente [-1,1].

O modelo neuronal da Figura 3.1 inclui também um *bias* (viés) aplicado externamente, representado por b_k. O viés tem o efeito de aumentar ou diminuir a entrada líquida da função de ativação, dependendo se ele é positivo ou negativo, respectivamente.

Em termos matemáticos, podemos descrever um neurônio *k* escrevendo o seguinte par de equações:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{3.1}$$

e,

$$y_k = \varphi(u_k + b_k) \tag{3.2}$$

onde, x_j são os sinais de entrada; w_{kj} são os pesos sinápticos do neurônio *k*; u_k é a saída do combinador linear devido aos sinais de entrada; b_k é o viés; φ é a função de ativação e y_k é o sinal de saída do neurônio. O uso do viés b_k tem o efeito de aplicar uma transformação afim à saída u_k do somador linear no modelo da Figura 3.1, como mostrado por:

$$v_k = u_k + b_k \tag{3.3}$$

Em particular, dependendo se o viés b_k é negativo ou positivo, a relação entre o *campo local induzido* ou *potencial de ativação* v_k do neurônio *k* e a saída do combinador linear u_k é modificada: a partir desta transformação afim, o gráfico de v_k em função de u_k não passa mais pela origem.

O viés b_k é um parâmetro externo do neurônio artificial *k*. É possível considerar sua presença como na equação 3.2. Equivalentemente se podem formular as equações 3.1 até 3.3 como segue:

$$v_k = \sum_{j=0}^{m} w_{kj} x_j \tag{3.4}$$

$$y_k = \varphi(v_k) \tag{3.5}$$

Na equação (3.4) se adicionou uma nova sinapse. A sua entrada e o seu peso são dados, respectivamente, por:

$$x_0 = +1 \tag{3.6}$$

$$w_{k0} = b_k \tag{3.7}$$

É possível, portanto, reformular o modelo do neurônio *k* (da Figura 3.1). onde o efeito do viés é levado em conta de duas maneiras: (i) adicionando-se um novo sinal de entrada fixo em +1 e (ii) adicionando-se um novo peso sináptico igual ao viés b_k.

3.1.3.1. Função de Ativação

A função de ativação, representada por $\varphi(v)$, define a saída de um neurônio em termos do campo local induzido *v*. Há três tipos básicos de função de ativação:

- *Função de Limiar*: para este tipo de função de ativação, tem-se:

$$\varphi(v) = \begin{cases} 1, se & v \geq 0 \\ 0, se & v < 0 \end{cases} \tag{3.8}$$

Na literatura de engenharia, esta forma de função de limiar é normalmente referida como *função de Heaviside* (Haykin, 2001). A saída do neurônio *k* que emprega esse tipo de função pode ser expressa como:

$$y_k = \begin{cases} 1 & se \quad v_k \geq 0 \\ 0 & se \quad v_k < 0 \end{cases} \tag{3.9}$$

onde v_k é o campo local induzido do neurônio. Tal neurônio é referido na literatura como *modelo de McCulloch-Pitts*, a partir do trabalho dos pioneiros de redes neurais (Haykin, 2001). Esse tipo de neurônio apresenta uma característica do tipo *tudo-ou-nada*.

- *Função Linear por Partes*: para este tipo de função de ativação, tem-se:

$$\varphi(v) = \begin{cases} 1, & v \geq +\frac{1}{2} \\ v, & +\frac{1}{2} > v > -\frac{1}{2} \\ 0, & v \leq -\frac{1}{2} \end{cases} \tag{3.10}$$

onde assume-se que o fator de amplificação dentro da região linear de operação é a unidade. Esta forma de função de ativação pode ser vista como uma *aproximação* de um amplificador não-linear (Lazo Lazo, 2000). A função de limiar pode ser considerada um caso especial desta, onde o amplificador é tomado infinitamente grande.

- *Função Sigmóide*: a função sigmóide, cujo gráfico tem a forma de *S*, é de longe a forma mais comum de função de ativação utilizada na construção de redes neurais artificiais. Ela é definida como uma função estritamente crescente que exibe um balanceamento adequado entre comportamento linear e não-linear (Haykin, 2001). Um exemplo de função com essas características é a *função logística*, dada por:

$$\varphi(v) = \frac{1}{1+\exp(-av)} \tag{3.11}$$

onde *a* é o parâmetro de inclinação da função sigmóide. Uma característica importante para o desenvolvimento da teoria subjacente às redes neurais é a diferenciabilidade da função de ativação. Nesse aspecto, funções sigmóides são melhores candidatas que funções de limiar ou lineares por partes (a necessidade da diferenciabilidade está ligada ao cálculo dos erros, que guia o processo de aprendizado de uma rede neural). Algumas vezes é desejável que a função de ativação se extenda de –1 a +1 (as anteriores vão de 0 a +1), ou seja, sejam anti-simétricas em relação a origem. Para a função sigmóide deste tipo, tem-se a *função tangente hiperbólica*:

$$\varphi(v) = \tanh(v) \tag{3.12}$$

De um modo geral, a função logística e a tangente hiperbólica são as mais usadas em aplicações reais de redes neurais artificiais.

3.1.3.2. Topologia de Rede

A maneira pela qual os neurônios de uma rede neural estão estruturados está intimamente ligada com o algoritmo de aprendizagem usado para treinar a rede. Assim, analisar a topologia de redes neurais sem estudar os algoritmos de aprendizagem pode não fazer muito sentido. Em seções subseqüentes, vai-se mostrar os principais tipos de aprendizagem, mas antes uma breve introdução aos tipos básicos de estruturas de rede é devida.

Em uma rede neural em *camadas*, os neurônios estão organizados na forma de camadas. Na forma mais simples de uma rede deste tipo, há uma camada de entrada de nós de fonte que se projeta sobre uma camada de saída de neurônios (nós computacionais), mas não vice-versa. Em outras palavras, é uma rede alimentada adiante (*feedforward*).

A segunda classe de rede neural alimentada adiante se distingue pela presença de uma ou mais *camadas ocultas*, cujos nós computacionais são chamados correspondentemente de *neurônios ocultos.* A função destes é intervir entre a entrada externa e a saída da rede de uma maneira útil. De um modo bastante geral, os neurônios da camada escondida servem para armazenar o conhecimento contido nos padrões de treinamento, de maneira a que a rede seja capaz de generalizar quando apresentada a padrões desconhecidos.

Já uma rede neural *recorrente* se distingue pelo fato de ter pelo menos um laço de realimentação. Isso se refere a uma situação em que pelo menos uma saída da rede é realimentada para sua entrada. De um modo geral, os laços de realimentação servem para introduzir um componente de memória na rede neural, de maneira a que ela considere os resultados passados que produziu no seu processo de aprendizagem. As Figuras 3.2 e 3.3, a seguir, ilustram os principais tipos de rede descritos.

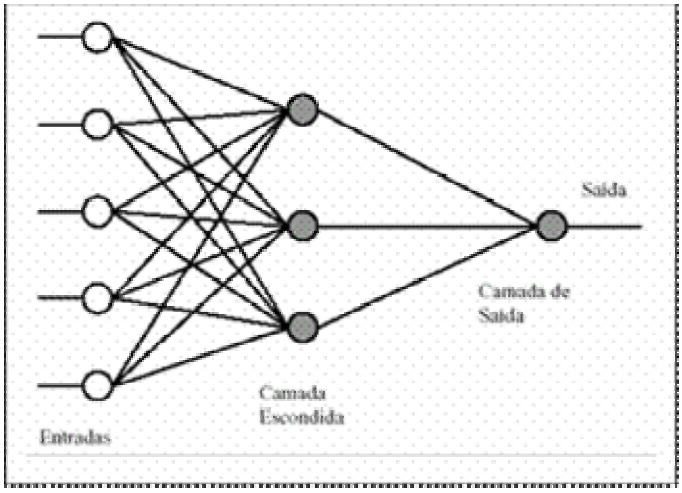

Figura 3.2: Rede Não-Recorrente com 1 camada escondida (Lazo Lazo, 2000)

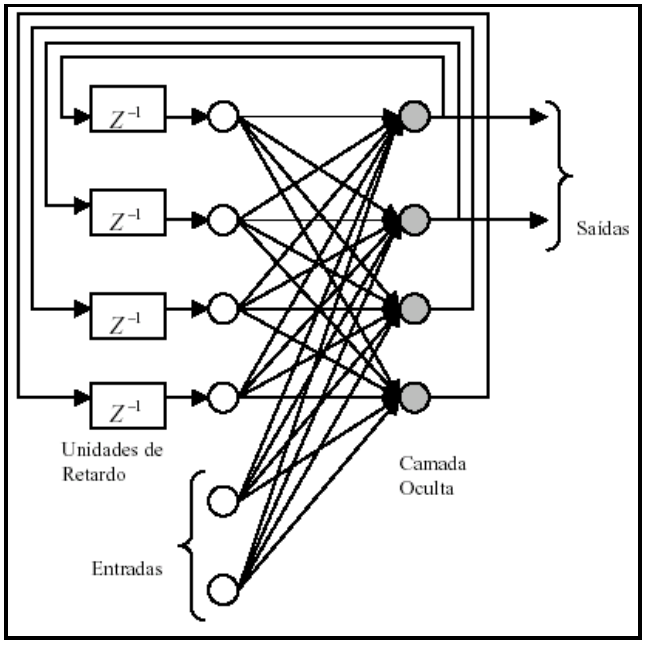

Figura 3.3: Exemplo de Rede Neural Recorrente (Haykin, 2000)

3.1.4.Processos de Aprendizagem

Aprendizagem, no sentido de redes neurais, é o processo de calcular os pesos sinápticos de uma rede. Os pesos, como já discutido, são um fator crucial, definindo o valor da saída de um neurônio, e portanto definindo qual o resultado que a rede obtém. Num sentido mais genérico, é possível dizer que os pesos "são" o conhecimento, já que todos os exemplos apresentados à rede são armazenados à medida que são apresentados durante o treinamento. Com a exceção do viés e de técnicas mais avançadas de poda de rede (explicadas mais adiante), os pesos são a característica básica da rede que é alterada ao longo do processo de aprendizagem. Há dois tipos

básicos de processos de aprendizagem: a supervisionada e a não-supervisionada, que vão ser descritas a seguir.

A aprendizagem supervisionada necessita de um par de vetores composto da entrada e do vetor alvo que se deseja como saída. Juntos, estes vetores são chamados de par de treinamento ou vetor de treinamento, sendo que geralmente a rede é treinada com vários vetores de treinamento. O processo de aprendizagem é feito da seguinte maneira: o vetor de entrada é aplicado, a saída da rede é calculada e comparada com o correspondente vetor alvo. O erro encontrado é então realimentado através da rede e os pesos são atualizados de acordo com um algoritmo determinado a fim de minimizar este erro. Este processo de treinamento é repetido até que o erro, para todos os vetores de treinamento, tenha alcançado o nível especificado.

A aprendizagem não-supervisionada não requer vetor alvo para as saídas. O conjunto de treinamento modifica os pesos da rede de forma a produzir saídas que sejam consistentes, isto é, tanto a apresentação de um dos vetores de treinamento, como a apresentação de um vetor que é suficientemente similar, produzirão o mesmo padrão de saídas. O processo de treinamento extrai as propriedades estatísticas do conjunto de treinamento e agrupa os vetores similares em classes.

3.1.5.O Algoritmo *Backpropagation*

O algoritmo de retropropagação do erro (Backpropagation) consiste, basicamente, em determinar as variações nos pesos sinápticos da rede neural, tendo como objetivo minimizar o erro obtido na saída através do aprendizado do vetor de treinamento (entrada-saída). A característica inovadora desse algoritmo é dada pela sua capacidade de atribuir os erros obtidos na saída da rede às camadas ocultas.

O algoritmo trabalha em duas etapas: um passo de computação “para frente” (*forward-propagation*), onde é calculada a saída da rede, e um passo de computação “para trás” (*backpropagation*), onde os erros são calculados e os pesos são recalculados, de maneira a minimizar o erro.

O passo para a frente funciona da seguinte maneira: dado um exemplo de treinamento representado por (*x*, *d*), como o vetor de entrada *x* aplicado à camada de entrada de nós da rede, e o vetor saída desejada apresentado à camada de nós de saída da rede. Primeiramente devem ser calculados os campos locais induzidos e os sinais funcionais, prosseguindo através da rede, camada por camada. O campo local induzido $v_j^{(l)}$ para o neurônio *j* na camada *l* é:

$$v_j^{(l)} = \sum_{i=0}^{m_0} w_{ji}^{(l)} \cdot y_i^{(l-1)} \tag{3.13}$$

onde $y_i^{(l-1)}$ é o sinal (função) de saída do neurônio *i* na camada anterior *l-1*, e $w_{ji}^{(l)}$ é o peso sináptico do neurônio *j* na camada *l*, que é alimentado pelo neurônio *i* da camada *l-1*.

Assumindo-se uma função de ativação sigmóide (vide item 3.3.1), o sinal de saída do neurônio *j* da camada *l* é dado por:

$$y_j^{(l)} = \varphi_j(v_j^{(l)}) \tag{3.14}$$

Se o neurônio *j* está na primeira camada da rede (ou seja, *l*=1), tem-se:

$$y_j^{(0)} = x_j \tag{3.15}$$

onde *xj* é o j-ésimo elemento do vetor de entrada *x*. Se o neurônio *j* está na camada de saída da rede (ou seja, *l*=*L*, onde *L* é chamado de *profundidade* da rede), tem-se:

$$y_j^{(L)} = o_j \tag{3.16}$$

O passo para a frente termina com o cálculo do erro na camada de saída:

$$e_j = d_j - o_j \tag{3.17}$$

O passo para trás se refere à atribuição de "responsabilidades" pelo erro na saída. Ele se inicia com o cálculo dos gradientes locais em cada camada da rede, definidos por:

$$\delta_j^{(l)} = e_j^{(L)} \cdot \varphi'_j(v_j^{(L)}) \text{ para o neurônio } j \text{ na camada de saída } L \tag{3.18}$$

$$\delta_j^{(l)} = \varphi'_j(v_j^{(l)}) \cdot \sum_k \delta_k^{(l+1)} \cdot w_{kj}^{(l+1)} \text{ para o neurônio } j \text{ da camada oculta } l \tag{3.19}$$

onde o apóstrofe em $\varphi'(\cdot)$ representa a diferenciação em relação ao argumento. Os pesos sinápticos da rede são ajustados então com base na seguinte regra:

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \eta \cdot \delta_j^{(l)}(n) \cdot y_i^{(l-1)}(n) \tag{3.20}$$

onde η é a taxa de aprendizagem.

Os passos de computação para frente e para trás são então iterados sucessivamente, até que algum critério de parada seja alcançado (normalmente número de iterações ou erro pré-estabelecido).

Embora o algoritmo pareça razoavelmente simples (talvez aí resida sua força), ele está por trás da maior parte das aplicações contemporâneas de redes neurais. Como é possível depreender do histórico (seção 3.2), seu desenvolvimento tornou o campo das redes neurais artificiais um dos mais estudados em inteligência artificial.

3.1.5.1. Taxas de Aprendizagem

O algoritmo de retropropagação fornece uma "aproximação" para a trajetória no espaço de pesos calculada pelo método da descida pelo gradiente. Quanto menor for o parâmetro da taxa de aprendizagem η, menor serão as variações dos pesos sinápticos da rede, de uma iteração para a outra, e mais suave será a trajetória no espaço de pesos. Esta melhoria, entretanto, é obtida à custa de uma taxa de aprendizagem lenta. Por outro lado, se o parâmetro da taxa de

aprendizagem for colocado muito grande, para acelerar o processo de treinamento, as grandes modificações nos pesos sinápticos resultantes podem tornar a rede instável (Haykin (2001) chama de “oscilatória”). Resolver este dilema não é tarefa simples, e requer uma de duas atitudes: (i) experimentação com o algoritmo, ou seja, alteração “manual” da taxa de aprendizagem de maneira a obter resultados melhores, ou (ii) o uso de técnicas mais avançadas, que contornam engenhosamente o problema, introduzindo alterações no algoritmo básico, de maneira a obter melhor performance. Tais técnicas serão discutidas na seção 3.6.2 adiante.

3.1.6.Desempenho de Redes Neurais

Apesar de serem uma ferramenta poderosa, as redes neurais artificiais são extremamente sensíveis. Assim, sua aplicação em problemas práticas requer toda uma carga de preparação e cuidados *a priori*, e a análise *a posteriori* também requer uma boa dose de cuidado, de modo a não tomar por certo aquilo que não necessariamente é. Neste item, vai-se discutir alguns fatores fundamentais para a boa performance de redes neurais na resolução de problemas práticos, e quais passos devem ser tomados para preparar os dados, treinar a rede, validar os resultados obtidos e analisá-los.

3.1.6.1. Generalização

Na aprendizagem por retropropagação, começa-se tipicamente com uma amostra de treinamento e se usa o algoritmo de retropropagação para calcular os pesos sinápticos de um *perceptron* de múltiplas camadas codificando tantos exemplos de treinamento quanto possível para dentro da rede. Espera-se que a rede neural assim projetada seja capaz de generalizar. Diz-se que uma rede generaliza bem quando o mapeamento de entrada-saída da rede for correto (ou aproximadamente correto) para dados de teste não-utilizados para a criação ou treinamento da rede; o termo “generalização” é tomado emprestado da psicologia. Aqui se assume que os dados de teste são tirados da mesma população usada para gerar os dados de treinamento.

O processo de aprendizagem (i.e., treinamento de uma rede neural) pode ser visto como um problema de “ajuste de curva”. A própria rede pode ser considerada simplesmente como um mapeamento não-linear de entrada-saída. Este ponto de vista nos permite considerar a generalização não como uma propriedade mística das redes neurais, mas simplesmente como o efeito de uma boa interpolação não-linear sobre os dados de entrada (Haykin, 2001). A rede realiza boa interpolação fundamentalmente porque *perceptrons* de múltiplas camadas com funções de ativação contínuas produzem funções de saída que também são contínuas.

Uma rede neural projetada para generalizar bem, produzirá um mapeamento de entrada-saída correto, mesmo quando a entrada for um pouco diferente dos exemplos usados para treinar

a rede. Entretanto, quando uma rede neural aprende um número excessivo de exemplos de entrada-saída, a rede pode acabar memorizando os dados de treinamento. Ela pode fazer isso encontrando uma característica (devido ao ruído, por exemplo) que está presente nos dados de treinamento, mas não na função subjacente que deve ser modelada. Este fenômeno é conhecido como *excesso de ajuste* ou *excesso de treinamento*[13]. Quando a rede é treinada em excesso, ela perde a habilidade de generalizar entre padrões de entrada-saída similares.

Normalmente, carregar dados desta forma em um *perceptron* de múltiplas camadas requer o uso de mais neurônios ocultos do que é necessário, resultando que contribuições indesejáveis no espaço de entrada devido a ruído sejam armazenadas nos pesos sinápticos da rede. A "memorização" é essencialmente uma "tabela de consulta", o que implica que o mapeamento de entrada-saída computado pela rede neural não é suave. A suavidade do mapeamento de entrada-saída está intimamente relacionada com critérios de seleção de modelos do tipo *Navalha de Occam*, cuja essência é selecionar a função "mais simples" na ausência de qualquer conhecimento prévio contrário. No contexto da generalização, a função mais simples significa a função mais suave que aproxima o mapeamento para um dado critério de erro, porque esta escolha geralmente demanda os menores recursos computacionais. É, portanto, importante procurar um mapeamento não-linear suave para relações de entrada-saída mal-formuladas, de modo que a rede seja capaz de classificar corretamente novos padrões em relação aos padrões de treinamento.

Além da discussão da generalização, é interessante mencionar um outro ponto: normalmente espera-se que uma rede neural se torne bem-treinada de modo que aprenda o suficiente do passado para generalizar no futuro. Desta perspectiva, o processo de aprendizagem se transforma em uma escolha de parametrização da rede para este conjunto de dados. Mais especificamente, é possível ver o problema de seleção da rede como a escolha, dentre um conjunto de estruturas de modelo candidatas (parametrizações), a "melhor" de acordo com um certo critério. Esta é uma maneira apenas semanticamente diferente de colocar a questão, em relação a abordagem de suavidade usada anteriormente.

Nesse sentido, uma ferramenta padrão da estatística conhecida como *validação cruzada* fornece um princípio orientador atraente (Haykin, 2001). Primeiramente, o conjunto de dados disponível é dividido aleatoriamente em um conjunto de treinamento e um conjunto de teste. O conjunto de treinamento é dividido adicionalmente em dois subconjuntos distintos: (i) *Subconjunto de estimação*, usado para selecionar o modelo; (ii) *Subconjunto de validação*, usado para testar ou validar o modelo.

A motivação é validar o modelo com um conjunto de dados diferente daquele usado para estimar os parâmetros. Desta forma, é possível usar o conjunto de treinamento para avaliar o

[13] Tradução do inglês *overfitting* ou *overtraining*.

desempenho de vários modelos candidatos e assim, escolher o "melhor". Há, entretanto, uma possibilidade considerável de que o modelo assim selecionado, com os valores de parâmetros com melhor desempenho, possa acabar ajustando excessivamente o subconjunto de validação. Para se resguardar dessa possibilidade, o desempenho de generalização do modelo selecionado é medido sobre o conjunto de teste, que é diferente do subconjunto de validação. O uso de validação cruzada é atrativo particularmente quanto é necessário projetar uma rede grande cujo objetivo seja uma boa generalização[14]. É possível, por exemplo, utilizar a validação cruzada para determinar o *perceptron* de múltiplas camadas com o melhor número de neurônios ocultos e quando é melhor parar o treinamento. Portanto, a validação cruzada é uma ferramenta útil no projeto de boas redes neurais e na busca por modelos com alta capacidade de generalização.

3.1.6.2. Algoritmos de Treinamento

O algoritmo de treinamento baseado em retropropagação trabalha normalmente com o método de descida por gradiente. Para a maioria dos problemas práticos, esse método é muito lento. Assim, ao longo do tempo foram sendo criados diversos algoritmos de alta performance para o treinamento de redes neurais com retropropagação, que convergem entre dez e cem vezes mais rapidamente que o algoritmo de descida por gradiente tradicional.

Estes algoritmos de alta performance podem ser divididos basicamente em duas categorias (Demuth & Beale, 2001): aqueles que usam técnicas heurísticas, desenvolvidas a partir de uma análise cuidadosa do algoritmo de descida por gradiente básico; e aqueles que utilizam métodos de otimização numérica, para acelerar a convergência dos pesos da rede.

Dentro do primeiro grupo, vale citar três diferentes métodos: (i) o uso de *momentum*, ou seja, a inclusão de um parâmetro adicional na equação de mudança de pesos da rede, de modo a acelerar a convergência em regiões onde o gradiente é pronunciado, e vice-versa em regiões onde a rede aprende pouco; (ii) o algoritmo com Taxa de Aprendizagem Variável, onde o parâmetro η da rede pode variar ao longo do processo de treinamento, evitando os problemas de instabilidade advindos de uma escolha *a priori* errônea deste parâmetro; e (iii) o algoritmo de Retropropagação Resiliente[15], que tenta evitar o problema de magnitude das derivadas parciais usadas para realizar a atualização dos pesos da rede (note-se nas equações do "passo para trás" do algoritmo *backpropagation*, que se usa constantemente as derivadas da função de ativação), e faz uso apenas do sinal destas derivadas, que indicam apenas a direção que se deve tomar na superfície de erro, deixando o problema da magnitude de variação dos pesos da rede para um

[14] No capítulo 4, vai-se tentar alcançar este objetivo.

[15] *Resilient Backpropagation.*

outro parâmetro. De um modo geral, estas três modificações no algoritmo básico tendem a melhorar consideravelmente a performance do processo de treinamento, sem incorrer em substancial aumento dos recursos computacionais utilizados.

Já no segundo grupo, há também três técnicas diferentes que permitem obter ganhos no processo de aprendizado: Gradiente Conjugado, quasi-Newton e Levenberg-Marquardt. O algoritmo do Gradiente Conjugado parte da premissa de que, embora uma função decresça mais rapidamente na direção do negativo de seu gradiente (o algoritmo básico de descida pelo gradiente), não necessariamente esta direção é que produz a melhor convergência global. Assim, o algoritmo faz uma busca numérica nas direções conjugadas à negativa do gradiente, tentando otimizar o resultado (numa analogia simples: é como o jogador de xadrez que pensa duas, três jogadas adiante). Há diversas maneiras de realizar tal busca, e daí surgem diversas variantes do algoritmo: Fletcher-Reeves, Polak-Ribiére, Powell-Beale, Golden Search, entre outros[16]. Os métodos quasi-Newton partem da mesma premissa básica das técnicas de gradiente conjugado, apenas utilizam métodos de otimização numérica distintos, baseados em variantes do algoritmo de Newton. Da mesma maneira, o algoritmo de Levenberg-Marquardt busca otimização numérica com uma aproximação do método de Newton (em linhas gerais, o método de Newton envolve o cálculo do Hessiano de uma função – os métodos quasi-Newton e Levenberg-Marquardt buscam aproximar o Hessiano). Segundo Demuth & Beale (2001), este último tem se mostrado ser o algoritmo mais rápido para o treinamento de redes alimentadas adiante (*feedforward*) de tamanho moderado (algumas centenas de pesos).

3.1.6.3. Treinamento Seqüencial e por Lote

Em uma aplicação prática do algoritmo de retropropagação, o aprendizado resulta das muitas apresentações de um determinado conjunto de exemplos de treinamento para o *perceptron* de múltiplas camadas. Como mencionado anteriormente, uma apresentação completa do conjunto de treinamento inteiro é denominada uma *época*. O processo de aprendizagem é mantido em uma base de época em época até os pesos sinápticos e os níveis de viés se estabilizarem e o erro médio quadrado sobre todo o conjunto de treinamento convergir para um valor mínimo. É uma boa prática tornar aleatória a ordem de apresentação dos exemplos de treinamento (Haykin, 2001), de uma época para a seguinte. Esta aleatoriedade tende a tornar a busca no espaço de pesos estocástica sobre os ciclos de aprendizagem, evitando assim a possibilidade de ciclos limitados, na evolução dos vetores de pesos sinápticos.

Para um dado conjunto de treinamento, a aprendizagem supervisionada pode então ocorrer de uma dentre duas formas básicas:

[16] Para detalhes de cada um destes algoritmos, ver Demuth & Beale (2001), ou Haykin (2001), pp. 262-271.

- *Modo Seqüencial*: o modo seqüencial da aprendizagem é também chamado de *modo online*, *modo padrão* ou *modo estocástico* (Haykin, 2001). Neste modo de operação, a atualização dos pesos é realizada após a apresentação de cada exemplo de treinamento.
- *Modo por Lote*: no modo por lote da aprendizagem supervisionada, o ajuste dos pesos é realizado após a apresentação de todos os exemplos de treinamento que constituem uma época. Para uma época particular, define-se a função de custo como:

$$\varepsilon_{médio} = \frac{1}{2N}\sum_{n=1}^{N}\sum_{j\in C} e_j^2(n) \tag{3.21}$$

onde o sinal de erro $e_j(n)$ é relativo ao neurônio de saída *j* do exemplo de treinamento *n*. O erro é igual à diferença entre a saída computada pela rede e a saída original do exemplo, ou seja, o *j-ésimo* elemento do vetor resposta desejada e o valor correspondente da saída da rede. Na equação anterior, o somatório interno em relação a *j* é realizado sobre todos os neurônios da camada de saída da rede, enquanto que o somatório externo em relação a *n* é realizado sobre todo o conjunto de treinamento da época considerada. Para um parâmetro de taxa de aprendizagem η, o ajuste aplicado ao peso sináptico w_{ji}, conectando o neurônio *i* ao neurônio *j*, é definido pela regra delta:

$$\Delta w_{ji} = -\eta\frac{\partial \varepsilon_{med}}{\partial w_{ji}} = -\frac{\eta}{N}\sum_{n=1}^{N} e_j(n)\frac{\partial e_j(n)}{\partial w_{ji}} \tag{3.22}$$

De acordo com essa equação, no modo por lote, o ajuste do peso w_{ji} é feito somente após o conjunto de treinamento inteiro ter sido apresentado à rede.

Do ponto de vista operacional *online*, o modo seqüencial de treinamento é preferível em relação ao modo por lote, porque requer menos armazenamento local para cada conexão sináptica (Haykin, 2001). Além disso, dado que os parâmetros são apresentados à rede de uma forma aleatória, o uso de ajuste de pesos de padrão torna a busca no espaço de pesos estocástica, o que torna menos provável que o algoritmo de treinamento fique preso em um mínimo local. Da mesma forma, a natureza estocástica do modo seqüencial torna mais difícil de estabelecer as condições teóricas para a convergência do algoritmo (Haykin, 2001).

Em suma, apesar de o modo seqüencial apresentar várias desvantagens, ele é bastante usado por duas razões práticas importantes: (i) é simples de implementar e (ii) fornece soluções efetivas a problemas grandes e difíceis.

3.1.6.4. Técnicas de Poda de Rede

Para resolver problemas do mundo real com redes neurais, normalmente é necessário o uso de redes de tamanho bastante grande, altamente estruturadas. Uma questão prática que surge neste contexto é a da minimização do tamanho da rede, mantendo bom desempenho. É menos

provável que uma rede neural com tamanho mínimo aprenda as idiossincrasias ou ruído dos dados de treinamento e, pode assim generalizar melhor sobre novos dados. É possível alcançar este objetivo de projeto de duas formas:

- Pelo *crescimento da rede*[17], começando com um *perceptron* de múltiplas camadas pequeno (para a tarefa em questão), e então se adiciona um novo neurônio ou uma nova camada de neurônios ocultos somente quando não se satisfizerem as especificações de projeto.
- Pela *poda da rede*[18], começando com um *perceptron* de múltiplas camadas grande, com um desempenho adequado para o problema em questão, e então o podando pela redução ou eliminação de certos pesos sinápticos de uma forma seletiva e ordenada.

As principais abordagens para a realização da poda da rede são: (i) baseada em uma forma de *regularização* e (ii) baseada em *eliminação* de certas conexões sinápticas da rede.

A regularização agrupa uma variedade de métodos de poda da rede[19], com um princípio motivador comum. No projeto de um *perceptron* de múltiplas camadas por qualquer método que seja, está-se de fato construindo um modelo não-linear, para um fenômeno responsável pela geração de exemplos de entrada-saída usados para treinar a rede. Na medida em que o projeto da rede é de natureza estatística, se deseja um compromisso adequado entre confiabilidade dos dados de treinamento e qualidade do modelo (ou seja, um método adequado para resolver o dilema bias-variância (Haykin, 2001)). No contexto de aprendizagem supervisionada (*backpropagation* ou outro algoritmo), é possível realizar esse compromisso minimizando o risco total, expresso como:

$$R(w) = \varepsilon_S(w) + \lambda \varepsilon_C(w) \tag{3.23}$$

onde o primeiro termo é a *medida de desempenho* da rede (ou também o erro da rede) e o segundo termo é a *punição por complexidade* da rede, multiplicada pelo *parâmetro de regularização*. A punição por complexidade define o quanto a rede pode aprender a partir dos exemplos de treinamento – em outras palavras, qual a confiabilidade dos dados disponíveis, no sentido de definir bem a rede. Os algoritmos de poda de rede baseados em regularização trabalham para diminuir este parâmetro, evitando redes (e pesos sinápticos) muito grandes e assim permitindo redes que aprendam melhor e portanto generalizem melhor.

A abordagem por eliminação parte da idéia básica de podar a rede a partir da informação sobre as derivadas de segunda ordem da superfície de erro, de forma a estabelecer um compromisso entre a complexidade da rede e o desempenho do erro de treinamento. Em particular, constrói-se um modelo local da superfície de erro para prever analiticamente o efeito

[17] Normalmente chamado de *cascading* ou *cascade-correlation* na literatura (Zeki-Susac, 1999).

[18] Normalmente chamado de *pruning* na literatura (Haykin, 2001 e Zeki-Susac, 1999).

[19] "Decaimento de Pesos", "Eliminação de Pesos" e "Suavizador Aproximativo" são exemplos (Haykin, 2001).

de perturbações sobre os pesos sinápticos, e a partir deste modelo tenta-se obter melhor performance de generalização da rede do que a normalmente seria obtida com uma técnica do tipo descida pelo gradiente. O objetivo é identificar um conjunto de parâmetros cuja eliminação do *perceptron* de múltiplas camadas causa o menor aumento do erro médio da rede, e a partir daí eliminar estes parâmetros.

3.1.6.5. Pré e Pós-Processamento

Um dos resultados mais importantes no trabalho com redes neurais é a prova de que redes neurais são aproximadores universais de funções[20]. Em outras palavras, dado um número suficientemente grande de parâmetros livres, garantidamente o processo de treinamento vai achar um mapeamento entre qualquer conjunto de variáveis independentes e dependentes. Este é um resultado poderoso que garante que redes neurais podem atacar uma enorme gama de problemas, mas há um porém: elas também vão achar relações onde elas não existem. Portanto, o processo de selecionar e tratar as variáveis de um problema tem de ser parte integrante do processo de projeto de uma rede neural.

Independentemente da eficiência do algoritmo de aprendizagem em termos de convergência, generalização e estabilidade, o indicador de performance último de um estimador neural vai depender na relevância das variáveis independentes escolhidas e na qualidade dos dados usados. Esta é uma maneira rebuscada de dizer "Entra lixo, sai lixo"[21]. Além disso, trabalhar com muito poucas variáveis independentes vai diminuir o espaço de busca excessivamente, e introduzir vieses no processo de modelagem que em geral levam a generalizações pobres. De outro lado, variáveis demais aumentam a dimensionalidade do espaço de busca e tornam qualquer algoritmo computacionalmente ineficiente e portanto inútil.

A seleção de variáveis é uma etapa importante do processo de modelagem antes de se iniciar a computação propriamente dita. Outra etapa crucial é o pré-processamento dos dados, de modo que a rede neural tenha mais facilidade para trabalhar. De um modo geral, esse pré-processamento envolve basicamente três etapas (Haykin, 2001 e Refenes, 1995):

- *Remoção da média*: cada variável deve ser pré-processada de modo que seu valor médio, calculado sobre todo o conjunto de treinamento ou seja próximo de zero, ou seja pequeno comparado com o desvio-padrão. Para avaliar o significado prático desta regra, considere-se o caso extremo, onde as variáveis de entrada são positivas de modo consistente. Nesta

[20] Esse resultado é o famoso *Teorema da Aproximação Universal*. Para uma formulação do mesmo, vide Haykin, 2001, pp. 234-235. Quem provou o teorema, no contexto de perceptrons de múltiplas camadas, foi Cybenko, em 1989, no artigo "Approximation by Superposition of a sigmoidal funcion", *in Mathematics of Control, Signals and Systems*, vol. 2, pp. 303-314.

[21] Uma tradução livre do famoso ditado "*Garbage in, garbage out*".

situação, os pesos sinápticos de um neurônio na primeira camada oculta podem apenas crescer juntos ou decrescer juntos. Conseqüentemente, se o vetor peso daquele neurônio deve mudar de direção, ele só pode fazer isso ziguezagueando seu caminho através da superfície de erro, o que é tipicamente lento e deve ser evitado (Haykin, 2001).

- *Descorrelação*: as variáveis de entrada contidas no conjunto de entrada não devem ser correlacionadas, o que pode ser feito através de uma técnica chamada Análise de Componentes Principais[22].
- *Equalização da Covariância*: as variáveis de entrada descorrelacionadas devem ser escaladas para que suas covariâncias sejam aproximadamente iguais, assegurando-se com isso que os diferentes pesos sinápticos da rede aprendam aproximadamente com a mesma velocidade.

Além do pré-processamento, é obviamente fundamental realizar um pós-processamento dos resultados, de maneira a ter resultados interpretáveis dada uma saída qualquer da rede neural. O pós-processamento segue as mesmas etapas que o pré-processamento, mantendo as características do processo gerador intactas.

3.1.7.Conclusões

Nesta breve introdução teórica ao vasto campo das redes neurais, buscaram-se fundamentalmente dois objetivos: (i) apresentar os principais conceitos envolvidos na concepção e projeto de uma rede neural artificial e (ii) dar uma medida dos fatores que podem fazer com que uma rede tenha um bom desempenho na resolução de problemas complexos. O uso desta ferramenta não é trivial, pelo contrário, logo a obtenção de bons resultados, como já mencionado, é profundamente dependente de um processo de projeto e modelagem cuidadoso. A teoria é um guia poderoso para o projetista, mas apenas a experimentação com os algoritmos e os diferentes parâmetros torna os resultados consistentes. Vamos analisar a seguir a literatura acerca da aplicação de redes neurais ao campo de finanças, com ênfase na questão da previsão de séries temporais, de maneira a aprofundar o detalhamento das potencialidades dessa ferramenta.

3.2. Redes Neurais e Previsão em Finanças : Análise da Pesquisa Prévia

Após apresentar a teoria relevante em finanças e redes neurais para o problema que nos pretendemos resolver, vamos passar a discutir agora o que a literatura apresenta em termos da união desses dois temas, ou seja, aplicações de redes neurais a problemas financeiros. Para

[22] Vide Nota de Rodapé 42, Capítulo 2. Vamos usar PCA mais adiante no Capítulo 4.

tanto, vamos primeiro dar uma visão geral do que já publicado, classificando os modelos e problemas que podem ser encontrados, e depois vamos focar naquilo que pode contribuir para nosso caso: redes neurais aplicadas a previsão de séries temporais em finanças.

Em um recente artigo, Liao & Wen (2007) levantam, através de pesquisas nas maiores bases de dados de *journals* internacionais, uma coleção de 10.120 artigos, publicados entre 1995 e 2005, que lidam com redes neurais artificiais. Destes, segundo os autores, aproximadamente 20% desses artigos são classificados como sendo de "ciências sociais", dos quais a maioria trata de problemas de negócios, o que inclui os problemas de previsão em finanças que são nosso foco de interesse. De partida temos então quase 2.000 artigos científicos nos quais navegar em busca de direções para melhor resolver nosso problema.

Alguns dos problemas mais significativos nos quais redes neurais têm sido aplicadas são: previsão de falências, avaliação e classificação de risco de crédito (*credit scoring*), previsão de mercado (tanto ações quanto títulos de renda fixa, além de preços de opções e taxas de câmbio), prognósticos de retornos de investimentos, gestão de portfólios, análise de hipotecas, entre outros[23]. Dito isso, há na literatura dois artigos cujos levantamentos de pesquisas com redes neurais em finanças são bastante completos, e podem nos ajudar na direção das melhores práticas. São eles Adya & Collopy (1998) e Zhang et al. (1998).

Antes de discutir estes trabalhos de maior extensão e portanto mais densos, há uma grande variedade de *papers* de *journals* internacionais para examinar, com aplicações variadas e metodologias idem. Boa parte destes trabalhos reporta bons resultados em termos de retornos obtidos a partir das previsões realizadas. A amplitude de problemas a que as redes neurais são aplicadas permite constatar a flexibilidade e robustez. A Tabela 3.1 e a Tabela 3.2 a seguir trazem um compêndio destes trabalhos. Estão descritos: o tipo de aplicação com que o artigo lida, o tipo de rede neural que usa para resolver o problema (incluídos detalhes de metodologia relevantes) e os tipos e quantidades de inputs aplicados às redes. Estes detalhes são importantes para poder traçar linhas comuns à maioria dos trabalhos, e tomar indicações para o projeto do sistema de previsão baseado em redes neurais do Capítulo 4.

Adya & Collopy (1998) se concentram na questão de quão efetivas são as redes neurais quando aplicadas a problemas de finanças (mas não apenas previsão de séries temporais). Partindo de uma amostra de 48 estudos publicados entre 1988 e 1994, os autores avaliam implementação das redes e validação dos resultados. Da amostra inicial, apenas 11 artigos passam pelo filtro dos dois critérios, mas os autores terminam por considerar um grupo de 22 estudos como relevantes. Destes, 19 (86%) concluem que as redes neurais têm performance superior a métodos alternativos. Um aspecto interessante do trabalho de Adya & Collopy é que

[23] Uma visão geral de aplicações de técnicas de inteligência artificial a finanças pode ser encontrada tanto em Deboeck (Ed.) (1994) quanto em Zhang & Zhou (2004).

eles levantam os detalhes das redes neurais de cada um dos artigos pesquisados. A seguir, no contexto do artigo de Zhang et al. (1998) vamos abordar essas questões.

Zhang et al. (1998) faz um levantamento bastante extenso de aplicações de redes neurais a problema de previsão, onde, dado o óbvio interesse financeiro, os problemas de previsão nos mercados têm destaque. Os autores buscaram determinar qual o estado da arte naquele momento, principalmente em aspectos importantes da modelagem de redes neurais. Para citar um trecho que resume bem o artigo: "*apesar das várias características satisfatórias das redes neurais, a construção de um modelo neural para um particular problema de previsão é uma tarefa não-trivial. Cada parte da modelagem de uma rede neural que afeta sua performance deve ser analisada cuidadosamente. Uma decisão crucial é a determinação da arquitetura apropriada, ou seja, o número de camadas, o número de nós em cada camada, e o número de arcos que interconecta os nós. Outras decisões de design incluem a seleção das funções de ativação nas camadas escondidas e de saída, o algoritmo de treinamento, os métodos de transformação ou normalização, os dados de treinamento e teste, e as medidas de performance*"[24].

Os autores analisam a literatura acerca de cada um dos tópicos acima, e chegam a algumas conclusões interessantes. Uma delas é que o "*desenho de um Rede Neural Artificial é mais arte que ciência*", visto que a maior parte dos artigos determina os parâmetros relevantes através de métodos de tentativa e erro, heurísticas ou simulação. Mas é possível tirar algumas informações importantes desse compêndio. Em termos de camadas escondidas, a maior parte dos autores usa uma, no máximo duas (as Tabelas 3.1 e 3.2 a seguir também mostram isso). Não nos parece claro que o uso de duas camadas escondidas proporciona qualquer benefício adicional (note-se que em Parreiras (2003) são testadas apenas configurações com uma camada), embora Zhang et al. (1998) cite alguns *papers* que fazem tão afirmação. Na Tabela 3.1, Buscema & Sacco (2000) e na Tabela 3.2, Madden e O'Connor (2006) usam duas camadas, e os resultados não parecem ser piores, ou melhores, por causa dessa camada extra. Parece um grau de complexidade adicional injustificado, contudo. Já acerca do número de nós na camada escondida, existem várias regras práticas[25], mas a maioria dos autores usa idéias intuitivas ou o empirismo de testar várias configurações. A decisão de número de nós de entrada e saída é bem pouco estruturada na literatura, como mostram Zhang et al. (1998), visto que essa questão depende muito do tipo de série temporal em mãos. Mais ainda, em seu artigo, Zhang et al. trabalham problemas de previsão com *inputs* apenas de *lags* dos dados, e não com modelos de fatores, que poderiam melhorar a performance das previsões, como pode ser visto em vários trabalhos nas Tabelas 3.1 e 3.2, como Chun & Kim (2004), Enke & Thawornwong (2005), Madden & O'Connor (2006) e

[24] Zhang et al. (1998), pp.42. Tradução do autor.

[25] Parreiras (2003), pp. 86, cita a regra (*inputs + outputs)/2*. Zhang et al. (1998), pp.44, cita quatro outras regras.

Zeki-Susac (1999), entre outros. Um fato é claro, a escolha dos dados de entrada é "*provavelmente a variável de decisão mais crítica em um problema de previsão de séries temporais*" (Zhang et al., 1998).

Um outro aspecto importante discutido por Zhang et al. é a função de ativação. Embora a maioria dos artigos citados ali use a função sigmóide (mesma conclusão do levantamento de Adya & Collopy (1998), onde todos os artigos usam a sigmóide), as Tabelas 3.1 e 3.2 mostram um quadro menos claro, com uma presença clara da tangente hiperbólica. Zhang et al. (1998) afirma que "*sugere-se usar a função logística para problemas de classificação e a tangente hiperbólica para problemas que envolvem aprendizado de desvios da média, como problemas de previsão"*, e seguindo Parreiras (2003), vamos usar esta última função para nossos experimentos no Capítulo 4.

O algoritmo de treinamento é um aspecto a nosso ver que define, em última instância, a boa performance de uma rede neural. Como bem afirmam Zhang et al. (1998): "*não há um algoritmo disponível atualmente que garanta uma solução ótima global para um problema de otimização não-linear geral em um tempo razoável de tempo. Sendo assim, todos os algoritmos de otimização na prática sofrem, inevitavelmente, de problemas de mínimos locais e o máximo que se pode fazer é usar o método de otimização disponível que dê o melhor ótimo local*"[26]. A maioria absoluta das pesquisas levantadas por Zhang et al. (1998) usam o algoritmo *Backpropagation* (ou descida pelo gradiente). Esta conclusão é semelhante à de Adya & Collopy (1998), em cuja pesquisa a maior parte (88% de um total de 48 artigos pesquisados) utilizam o algoritmo. A mesma constatação fica clara nas Tabelas 3.1 e 3.2, embora haja alguma diversidade maior, visto que a maior parte dos trabalhos ali condensados são mais recentes (quando o algoritmo *Backprop* já não parece tão atrativo – por ser já muito presente na literatura – e a diversidade de metodologias disponíveis e implementadas em pacotes computacionais disponíveis é bem maior). Um fato é claro, Zhang et al. (1998) conclui que o algoritmo *Backpropagation* padrão tem uma série de problemas, como falta de eficiência, dependência excessiva dos parâmetros iniciais, falta de robustez. Por isso é importante partir para aprimorar a metodologia. Ainda assim, são poucos os autores que trabalham com técnicas mais avançadas dentro do paradigma da retropropagação, como regularização, *early stopping*, algoritmos como *BFGS* e *Levenberg-Marquardt*, entre outras. Aqueles que o fazem normalmente constatam a melhora sensível dos resultados obtidos (Zhang et al. (1998) afirma que "*métodos de segunda-ordem têm convergência mais rápida, robustez e a habilidade de encontrar bons mínimos locais, o que os faz atrativos no treinamento de redes neurais*"). Em Parreiras (2003) o autor testa exaustivamente cinco algoritmos de treinamento, em um contexto de previsão de preços de ações: *Backpropagation* com *momentum*, *Backpropagation* resiliente, *Powell-Beale, BFGS*

[26] Zhang et al. (1998), pp. 48. Tradução do autor.

(Quasi-Newton) e *Levenberg-Marquardt*. As redes treinadas com *Levenberg-Marquardt* emergem superiores, seguidas de perto por *Powell-Beale*, *Backprop* resiliente, e mais longe *BFGS* e *Backprop* com *momentum*[27].

Há ainda três aspectos a considerar: normalização dos dados, amostras de treinamento e teste, e por fim medidas de performance. Segundo Refenes (Ed.) (1995), "*normalização é normalmente desejável de modo a remover a possibilidade de que os parâmetros da rede fiquem treinados apenas para um dado intervalo de entradas e saídas, e também para levar as entradas da função de ativação para o raio normal de operação dessa função*"[28]. Zhang et al. (1998) menciona o fato de que, em geral, a normalização dos dados é benéfica em termos de erros médios e de taxas de acerto. Há vários tipos de normalização possíveis (vide Refenes (Ed.) (1995), pp. 55-65 e Zhang et al. (1998), pp. 49-50 para detalhes), e novamente não há na literatura consenso acerca de melhores práticas. Sendo assim, no Capítulo 4 vamos utilizar alguns métodos de normalização (seguindo em grande parte Parreiras (2003), que se baseia no trabalho de Refenes), descrevendo em mais detalhes as técnicas.

A questão das amostras, de como dividir os dados disponíveis também não encontra um consenso na literatura disponível, segundo Zhang et al. (1998). Segundo os autores, "*em geral, como em qualquer perspectiva estatística, o tamanho da amostra é ligado de perto à precisão requerida no problema*". Ainda assim, não se sabe o que é uma população representativa, nem como dividi-la em dados de treinamento e teste, a não ser seguindo a recomendação geral de que ambas as parcelas devem ter comportamento representativo da população como um todo. O fato é que, em dados altamente estocásticos e não-lineares como é o caso de séries temporais em finanças, que ainda são bastante sujeitas a quebras de regime, determinar quando uma amostra é representativa não é tarefa trivial. Assim, vamos buscar trabalhar com a maior quantidade de dados disponível no Capítulo 4, e seguindo Parreiras (2003) em termos de divisão dos dados em amostras para treinamento, validação e teste fora da amostra.

As medidas de performance são amplamente discutidas em Refenes (Ed.) (1995), pp. 67-76. Zhang et al. (1998) afirma que "*uma medida de precisão apropriada para um determinado problema não é universalmente aceita pelos acadêmicos e praticantes*", e mostra como a medida mais comumente usada na literatura é o Erro Quadrático Médio (ou *MSE*, *Mean Square Error*), conclusão semelhante à de Adya & Collopy (1998). Já as Tabelas 3.1 e 3.2 mostram uma grande diversidade de medidas de performance, com presença relevante de medidas que medem a taxa de acerto do sinal gerado pela rede (vide por exemplo Jasic & Wood (2004) ou ainda Pérez-Rodríguez et al. (2005) para boas descrições), e também de medidas diretamente ligadas à lucratividade dos sinais gerados pela rede, na linha do trabalho de Refenes (Ed.)

[27] Parreiras (2003), pp. 77-105.

[28] Refenes (Ed.) (1995), pp. 56. Tradução do autor.

(1995). Dada a profusão de medidas disponíveis, vamos trabalhar com várias delas, notadamente a correlação previsto-realizado, na linha dos trabalhos de Motiwalla & Wahab (2000) e Parreiras (2003), e também medidas ligadas aos lucros gerados pela rede.

A metodologia de atacar o problema de modelagem de uma rede neural é presente em boa parte da literatura, como vimos acima, ao discutir cada elemento de um projeto de rede. Entretanto, alguns artigos, teses e livros propõem uma outra metodologia, mais integrada. Entre eles estão Refenes et al. (1997), Refenes & Zapranis (1999a), Refenes & Zapranis (1999b), Burgess (2000). Foram todos desenvolvidos a partir dos trabalhos de Apóstolos-Paul Refenes e seu time na *London Business School*, em meados da década de noventa. O que estes trabalhos buscam é uma metodologia que ataque o problema de previsão com redes neurais de uma maneira integrada, com técnicas estatísticas para seleção dos modelos, adequação das variáveis e determinação dos resultados. As técnicas de Refenes et al. têm uma fundamentação estatística sólida, tratando um modelo neural como um regressor não-linear, e desenvolvendo técnicas e testes para especificar corretamente este regressor. Os resultados reportados, tanto em Refenes & Zapranis (1999a) quanto em Burgess (2000) indicam que a metodologia mostra boa performance. No entanto, o grau de complexidade adicional e o seu custo de implementação computacional nos levam a crer que adotar uma metodologia modular, onde cada parâmetro de uma rede neural pode ser trabalhado, simulado e testado separadamente nos parece trazer benefícios de clareza e robustez, mesmo que a custa de alguma perda marginal de performance.

Uma tendência mais recente, que não aparece nos artigos de Adya & Collopy (1998) ou Zhang et al. (1998), mas que já aparece de maneira mais clara tanto na Tabela 3.1 quanto na Tabela 3.2, que apresentam vários trabalhos mais recentes, é a de combinar técnicas de redes neurais com outras metodologias do campo de inteligência artificial. Temos aqueles que usam Algoritmos Genéticos para otimizar a topologia da rede, como Harland (2000) e, Kalyvas (2001), em essência substituindo os algoritmos de otimização mais tradicionais, o que, dado a robustez das técnicas de Algoritmos Genéticos, não parece ser uma má idéia (embora o autor desconheça algum trabalho que compare a performance dessas técnicas num contexto de previsão em finanças). Outros autores também usam Algoritmos Genéticos, mas como uma outra parte de um modelo, onde a Rede Neural faz alguma previsão e o Algoritmo Genético executa alguma outra tarefa, que pode ser otimização de portfólio, como em Lazo Lazo (2000) e Parreiras (2003) ou seleção de *"experts"* como em Armano et al. (2005). Há ainda vários trabalhos que usam técnicas de lógica *fuzzy*[29], como Ang et al. (2006) que usam um modelo chamado *RSPOP* (*rough-set pseudo outer product*), que determina regras Se-Então a partir dos dados, ou ainda Kuo et al. (2001) que usam lógica *fuzzy* para determinar *inputs* qualitativos para um modelo de redes neurais aplicado à previsão dos retornos da Bolsa de Taiwan. Aparecem

[29] Para uma boa introdução sobre lógica *fuzzy*, vide Rezende (Org.) (2003), pp. 169-224.

ainda na literatura técnicas de reconhecimento de padrões, como o *"template matching"* usado em Leigh et al. (2002), ou outras metodologias de aprendizado de máquinas, como o *"Q-Learning"* de Casqueiro & Rodrigues (2006) e o *"Case-based reasoning"* de Chun & Kim (2004). Vale citar ainda vários autores que usam várias redes simultaneamente, seja através dos chamados *ensembles*, ou agrupamentos de redes, como os usados por Shadbolt & Taylor (2003) e Yu et al. (2005), ou ainda através de uma mistura de especialistas (*MoE,* ou *Mixture of Experts*), como os usados por Yümlü et al. (2005). Em suma, cada artigo sempre busca descobrir uma vantagem em relação aos trabalhos já existentes, qualquer vantagem. Por isso a adoção constante das mais recentes técnicas. O curioso é que em boa parte das vezes, essas técnicas mais avançadas são aplicadas junto de redes neurais bastante simples, com algoritmo *Backpropagation* primário, sem nenhuma espécie de refinamento. Parece claro que o conhecimento vai sendo gerado de maneira caótica, e não se acumulando de maneira incremental, onde um avanço é colocado sobre o avanço anterior. Assim, nos parece que há aqui várias técnicas que podem gerar excelentes modelos de previsão em finanças. Mas também nos parece que as possibilidades de um modelo neural bem construído e testado ainda são grandes, antes de ser esgotadas. Assim, vamos nos manter no caminho da robusta simplicidade na construção de nosso modelo de previsão, no Capítulo 4, e deixaremos para explorar as possibilidades abertas por estas técnicas mais recentes para o Capítulo 5, quando discutirmos trabalhos futuros.

A conclusão é clara, a partir da análise da literatura prévia de redes neurais aplicadas a problemas de previsão em finanças: nada substitui a cuidadosa experimentação com um modelo. Vários dos trabalhos que discutimos anteriormente permitem um levantamento das melhores práticas num projeto de rede neural, permitindo evitar algumas das armadilhas que pesquisadores anteriores podem ter caído. Ainda assim, seguindo o trabalho de Parreiras (2003), vamos apresentar no Capítulo 4 vários modelos para serem testados de maneira a obter a melhor rede possível.

Vale dizer que as redes neurais padecem, desde sua criação em meados dos anos cinqüenta, de um problema claro: seu aspecto de "caixa-preta" (Haykin, 2001). Essa é uma crítica constante na literatura de aplicações: embora muitas vezes as previsões tragam excelentes resultados, ninguém é capaz de apontar as razões inerentes para tanto[30]. O presente trabalho também não escapa desse dilema, e a literatura apresenta poucas saídas para ele. Aparentemente, é o preço a se pagar pela capacidade de explorar as não-linearidades inerentes aos mercados financeiros.

[30] Entre outros autores, apontam o problema Armano et al. (2005), Buscema & Sacco (2000), Harland (2000). A metodologia de Refenes & Zapranis (1999a) busca uma resposta para essa questão.

Trabalho	Ano	Objetivo	Rede Neural	*Inputs*
Ang et al.	2006	Prever preços de ações usando modelo fuzzy-neural inovador, chamado RSPOP (*rough-set pseudo outer product*), que determina regras *IF-THEN* a partir dos dados.	Vários modelos: Rede *Feedforward* com *Backpropagatio;* Rede *RBF*, e redes fuzzy-neurais que incorporam a geração de regras *IF-THEN* a partir dos dados.	Preços passados e suas médias móveis.
Armano et al.	2005	Prever retornos de 2 índices de ações: S&P500 (EUA) e Comit (Bolsa de Milão-Itália), a partir de um conjunto de *"experts"* neurais, otimizados com algoritmo genético.	Cada *"expert"* neural é uma rede *Backpropagation*, arquitetura 10-8-3-3, *gradient descent* com *momentum,* parâmetro *MSE.* São selecionados por um Algoritmo Genético.	Preços dos índices e indicadores técnicos derivados (médias móveis, RSI, etc.)
Azzini & Tettamanzi	2006	Prever preço do índice Dow Jones usando um modelo combinado de Redes Neurais e Algoritmos Genéticos, e realizar Arbitragem Estatística a partir desta previsão.	População de Redes *Backpropagation*. Um Algoritmo Genético seleciona a melhor rede com base em Erro Quadrático Médio. Rede escolhida: arquitetura 32-2-1.	32 séries: taxas de câmbio, taxas de juros, preços de commodities, setores da bolsa, *lags* dos preços.
Buscema & Sacco	2000	Previsão de taxas de câmbio e títulos de renda fixa de vários países.	Rede *Backpropagation*, arquitetura 80-12-12-1, incorpora propriedades como *Auto-momentum*, "Regra de Freud" e "Regra de Jung" para melhorar a função objetivo da rede.	80 Preços passados e Indicadores técnicos.
Cao et al.	2005	Comparar previsões dos modelos *CAPM, Fama-French* e de Redes Neurais para retornos de ações no mercado chinês.	Redes *Backpropagation* com descida pelo gradiente com *momentum*, arquiteturas 1-m-1 (4<m<10) e 3-n-1 (5<n<15).	Beta diário da ação, *Book-value* (valor patrimonial da ação) e Tamanho da empresa (*market-value*).
Casqueiro & Rodrigues	2006	Previsão dos preços do índice *PSI 20* (Bolsa de Portugal) e da ação *EDP*, usando redes neurais e técnica *Q-Learning*.	2 Redes: *Backpropagation* com descida pelo gradiente com *momentum*, e *RBF* Gaussiana.	Preços semanais do índice *PSI 20* e da ação *EDP*.
Chen et al.	2003	Prever direção dos movimentos da bolsa de Taiwan (*Taiwan Stock Exchange*), usando *inputs* econômicos, e comparar com um modelo *GMM* com filtro de Kalman.	Rede probabilística, arquitetura 4-68-2-1, treinamento com janela deslizante.	Valores de 3, 6 e 12 meses para retornos passados e dados econômicos como inflação, crescimento, produção industrial e taxas de juros.
Chun & Kim	2004	Comparar previsões de um modelo de Rede Neural com um modelo de *"Case-based Reasoning"* nos mercados da Coréia e Polônia.	Rede *Backpropagation com descida pelo gradiente com momentum*, arquitetura 6-4-1, função de ativação sigmóide, não explicita parâmetro.	Preços, volumes, retorno acumulado, taxas de juros, *price/earnings*, dividendos para os dois índices (Coréia e Polônia).
Enke & Thawornwong	2005	Previsão nível (preço) e Classificação de sinal (positivo / negativo) do índice S&P 500.	Previsão: Rede *Backpropagation* resiliente, 15-m-1 (11<m<27),validação cruzada e parâmetro*RMSE*. Classificação: 2 Redes, *Backprop,*arquitetura 15-27-2 e Probabílistica, arquitetura 15-27-2-1.	Dados mensais de preço, taxas de juros, *spreads* de risco, produção industrial e base monetária. Escolhidos pelo "*Information Gain"* que geram.
Harland	2000	Previsão do futuro de T-Bond (título de renda fixa dos EUA).	Redes *Backpropagation*, com Algoritmos Genéticos para treinamento e otimização da topologia da rede.	5 Indicadores técnicos.
Hassan et al.	2006	Previsão de preços de ações usando uma combinação de Redes Neurais, Algoritmos Genéticos e modelo *HMM* (*Hidden Markov Model*).	Rede *Backpropagation*, arquitetura 4-4-4, função de ativação *tanh.* A rede transforma a série temporal dos *inputs* e gera séries que alimentam o HMM otimizado com Algoritmo Genético.	Preços diários das ações: abertura, máximo, mínimo e fechamento.
Hung et al.	2003	Fazer previsão de retornos de 6 índices de ações (S&P 500, Hang Seng, Shangai Composite, Nikkei 225, CAC40 e AUS All Ordinaires) usando uma rede *RBF* e usar as previsões como *input* para modelos de otimização de portfólio.	Rede com função de base radial (*RBF*), com algoritmo "*cascade correlation"* para construir arquitetura.	Preços e *lags* dos 6 índices.
Jasic & Wood	2004	Previsão dos preços dos índices S&P500 (EUA), DAX (Alemanha), FTSE (Inglaterra) e Topix (Japão) usando Redes Neurais.	Rede *Backpropagation* com Gradiente Conjugado, arquiteturas 10-6-1 e 10-8-1, parâmetros *NMSE, RMSE, MAE*, e estatísticas de acerto de sinal.	10 *lags* dos preços de cada índice.
Kalyvas	2001	Previsão dos retornos diários do FTSE e S&P 500.	Redes *Backpropagation* resiliente, com Algoritmos Genéticos para treinamento e otimização da topologia da rede.	Preços Passados e Taxas de Juros.

Tabela 3.1: Pesquisa Prévia de Aplicações de Redes Neurais em Finanças (parte 1)

Trabalho	Ano	Objetivo	Rede Neural	*Inputs*
Kim	2006	Previsão do índice KOSPI (Bolsa da Coréia do Sul) usando um modelo que combina uma Rede Neural e um Algoritmo Genético que realiza *"Instance Selection"* nos dados.	Rede com arquitetura 12-12-1, pesos otimizados por Algoritmo Genético, função de ativação sigmóide. *Inputs* selecionados por Algoritmo Genético entre o total dos dados.	Preços do KOSPI e indicadores técnicos derivados (médias móveis, *RSI*, etc.)
Kuo et al.	2001	Prever retornos do índice da Bolsa de Taiwan usando um modelo combinando Rede Neural, Algoritmo Genético e Lógica *Fuzzy* com *inputs* quantitativos e qualitativos.	Redes *Backpropagation* com 1 ou 2 camadas escondidas, arquiteturas variando entre 42-30-1 até 54-75-1 ou 42-65-65-1. Parâmetro *MSE*.	Quantitativos (preços e indicadores técnicos) e Qualitativos (notícias de política, finanças, etc. classificadas por *experts*).
Lazo Lazo	2000	Prever os retornos de ações da Bovespa, e usá-los como *input* para um sistema de AGs para alocação da carteira.	Redes *Backpropagation,* com e sem Filtro de Kalman, 8~12 neurônios na camada escondida.	Preços passados dos 10 últimos dias..
Leigh et al.	2002	Prever retorno de 5 dias do índice *NYSE Composite Index* (Bolsa de Nova York), combinando rede neural e técnica de reconhecimento de padrões *"template matching"*.	Rede *Backpropagation*, arquitetura 22-8-2, com saída binária (sobe ou desce). Treinamento com janela deslizante.	*"Template matching"* identifica padrão técnico de alta ou baixa a partir de preços e volumes dos últimos 22 dias, formando uma matriz de correlações que serve de *input* à rede neural.
Madden & O'Connor	2006	Previsão dos retornos diários do índice Dow Jones usando redes neurais, com *inputs* de fatores externos.	Redes Backpropagation com descida pelo gradiente com *momentum*, testa várias arquiteturas, melhor é 31-37-20-1, critérios *RMSE* e sucesso direcional (retorno positivo ou negativo).	Preços do Dow Jones e suas médias móveis, preço do petróleo (WTI) e taxa de câmbio (JPY, CAD, GBP), e seus vários *lags*.
Motiwalla & Wahab	2000	Previsão dos preços de 11 índices de ações nos EUA, usando redes neurais, para determinação de sinais de compra e venda.	Rede *Backpropagation*, com descida pelo gradiente, arquitetura 20-9-1, função de ativação dupla: sigmóide e *tanh.* Construção da rede via *"cascade learning"*. Parâmetro: correlação prev./real fora da amostra.	Valores mensais de 11 índices de ações e seus *lags*, taxas de juros, *spreads* de risco.
Parreiras	2003	Previsão de retornos de ações brasileiras usando Redes Neurais, e otimização de portfólios usando Algoritmo Genético com *input* das previsões da rede neural.	Redes *Backpropagation*, vários algoritmos de treinamento (*Backpropagation* com *momentum*, Resiliente, *Powell-Beale*, *BFGS*, *Levenberg-Marquardt*), uma camada escondida, testa para número de nós ideal. Parâmetro correlação prev./real fora da amostra	Preços passados, e fatores de mercado (taxa de câmbio, taxas de juros, bolsas de valores internacionais), normalizados e tratados com PCA.
Pérez-Rodríguez et al.	2005	Comparar a performance de 3 tipos de Redes Neurais com um modelo *STAR* (*AR* com transição suave, para mudança de regime), na previsão dos retornos do índice *IBEX-35* (Bolsa de Madri - Espanha).	3 Redes: *Backpropagation*, *Jump Connection* (*input* direto no nó de saída) e Recorrente parcial. Arquitetura 2-4-1, com *cross-validation*. Parâmetros: *MAE, MAPE, RMSE, Theil-U* e estatística de sinal.	Preços do *IBEX-35* e um *lag*.
Refenes (Ed.)	1995	Previsão de diversos ativos financeiros (taxas de câmbio, títulos de renda fixa, ações).	Redes *Backpropagation* (diversas configurações). Parâmetros ligados á lucratividade do sinal gerado.	Conjunto de preços passados, indicadores técnicos e indicadores fundamentalistas.
Shadbolt & Taylor (Eds.)	2003	Construir um sistema de previsão de retornos de títulos de renda fixa.	Combinação de ferramentas estatísticas com modelos de redes neurais *Backpropagation* trabalhando em comitê.	Preços passados e dados macroeconômicos tratados com PCA..
Tsaih et al.	1998	Previsão do índice S&P 500 usando modelo combinado de Rede Neural e Geração de Regras (*rules-based system*) que seleciona os dados para treinamento.	4 Redes correspondentes a 4 estados (*Long*, *Short*, Espera e caso desconhecido). 3 tipos de Redes: *Perceptron*, *Backpropagation* (*MLP*) e Recorrente. Não detalha cada uma das redes, nem arquitetura nem especificações.	Preço do S&P 500 e indicadores técnicos derivados (médias móveis, RSI, osciladores).
Yu et al.	2005	Prever cotações de taxas de câmbio (DEM, GBP e JPY) usando comitês (*ensembles*) de redes neurais com PCA.	Rede *Backpropagation* com *Levenberg-Marquardt*, função de ativação *tanh*, arquitetura 4-4-1, parâmetro *MSE*.	Taxas de câmbio do DEM, GBP e JPY, e séries mensais de exportações de cada um dos 3 países.
Yümlü et al.	2005	Fazer previsão do XU100 (índice da Bolsa de Instanbul na Turquia), usando 3 modelos de Redes Neurais: *MLP*, Recorrente e um *Mixture of Experts* (*MoE*).	Rede *MLP Backpropagation*, função de ativação sigmóide. Rede Recorrente de Elman, arquitetura 7-15-1. Modelo *MoE* usa *clustering* para dividir os dados em grupos,corresponde aos nós na camada escondida. Treina com *Backpropagation* (detalhes pouco claros).	Preço do XU100, taxa de câmbio, taxas de juros de vários prazos, base monetária. Separa os dados em 2 blocos, devido à mudança de regime.
Zeki-Susac	1999	Prever os retornos das ações da IBM.	Vários tipos: *Backpropagation*, Função de Base Radial (RBF), Recorrente, Modular, *General Regression*, Probabílistica. Usa técnicas de poda de rede.	Indicadores fundamentalistas e técnicos, preços passados.

Tabela 3.2: Pesquisa Prévia de Aplicações de Redes Neurais em Finanças (parte 2)

4. MODELOS E EXPERIMENTOS

"O objetivo social do investimento profissional deve ser o de derrotar as forças negras do tempo e da ignorância que envolvem nosso futuro."
John Maynard Keynes

"Ao entendermos que o conhecimento imperfeito é inerente à condição humana, não há vergonha em errar, apenas em deixar de corrigir os erros"
George Soros

"Após atingido um certo nível de capacidade técnica, ciência e arte tendem a coalescer em estética, plasticidade e forma. Os maiores cientistas são também artistas"
Albert Einstein[1]

4.1. Introdução

Nos Capítulos 2 e 3 apresentamos o embasamento teórico que orienta este trabalho. Agora, no Capítulo 4, vamos projetar e simular os modelos de arbitragem estatística com cointegração e previsão por redes neurais, aplicando-os ao mercado de ações brasileiro.

O Capítulo está organizado da seguinte maneira: no item 4.2, vamos discutir os dados, como foram selecionados, olhando suas características estatísticas e realizando os primeiros testes de estacionariedade neles. A seguir, no item 4.3, vamos discutir a definição das cestas de ações[2], quais os critérios de construção, e aplicar a metodologia de cointegração de Engle-Granger para testar a dinâmica dos resíduos dessas cestas. Antes de continuar, vamos apresentar modelos de arbitragem estatística sem previsão, apenas com base em argumentos de reversão à média, chamados por Burgess (2000) de "arbitragem estatística implícita"[3].

No item 4.4 vamos apresentar os modelos de redes neurais, para realizar a previsão das dinâmicas. Seguindo a metodologia de Parreiras (2003) e com base na literatura discutida no item 3.2, vamos examinar técnicas de pré-processamento, as características constituintes dos modelos neurais, algoritmos de treinamento, divisão dos dados, a realização de previsões e seu pós-processamento. Ao fim deste item, em 4.4.4, vamos expandir os modelos de "arbitragem estatística implícita" para modelos de "arbitragem estatística condicional"[4], ou seja, condicionados às previsões.

---

[1] *Apud* Drobny (2006). Tradução do autor. As três citações estão na abertura do referido livro, exatamente na ordem em que aparecem aqui. Tal livro contém uma interessante série de entrevistas sobre o tema dos *hedge funds global macro*, fundos de investimento que investem ao redor do mundo em vários mercados tentando gerar lucros a partir de tendências macroeconômicas e como estas influenciam os preços dos ativos.

[2] *"Baskets"* na terminologia de Burgess (2000). Tradução do autor.

[3] Burgess (2000), pp. 177-197. Tradução do autor.

[4] Novamente seguindo a terminologia de Burgess (2000), pp. 268-296. Tradução do autor.

Por fim, no item 4.5., vamos agrupar todas as etapas do trabalho em modelos completos de arbitragem estatística, aplicando-os a toda a base de dados de preços de ações. A discussão sobre regras de *trading* e técnicas como *stop-loss* e alavancagem vai buscar adicionar valor aos modelos, na busca por lucros mais robustos. Por fim, vamos simular um modelo que incorpore todas as técnicas e aferir os resultados, apresentando as estatísticas de performance adequadas.

4.2. Dados

4.2.1. Seleção e Análise Preliminar

O mercado de ações brasileiro tem passado por uma revolução nos últimos anos. Com o advento da estabilidade econômica, combinado à excepcional liquidez internacional, e a implementação de reformas institucionais que deram mais segurança jurídica ao mercado, diversas empresas passaram a ver no mercado de ações uma fonte atrativa para financiar seus investimentos, passando a realizar operações de abertura de capital (os chamados "*IPOs*", ou *"Initial Public Offerings"*). Além disso, o aumento da poupança da população, o crescimento do mercado de fundos de investimentos e atração de investidores internacionais aumentaram consideravelmente o volume de transações na Bolsa de Valores de São Paulo.

O processo de modelagem que adotamos nesse trabalho é extremamente dependente da abundância e qualidade dos dados. Esse é um ponto fundamental: não é possível construir boas estratégias de arbitragem estatística sem muitos dados. Porquê? Pois estamos tentando explorar sistematicamente desvios de comportamento esperado. Agora, só é possível determinar qual o comportamento esperado se temos dados suficientes a partir dos quais inferi-lo. Só podemos determinar que um grupo de ações cointegra se tivermos uma série temporal suficientemente longa para poder testar. Mais ainda, essa série tem de ser relativamente representativa de vários comportamentos do mercado. Ou seja, temos de ser capazes de concluir que o modelo é robusto quanto à tendência geral do mercado, gerando retornos tanto em mercados de alta (os chamados "*bull markets*") quanto em tendência de baixa ("*bear markets*"). Assim, o primeiro critério de corte na seleção das ações para o trabalho é a existência de séries temporais longas de preços.

Um outro requisito fundamental é a liquidez das ações. Por duas razões: (i) ações com baixa liquidez não trazem, no sobe e desce de seus preços, variações ligadas a fatores externos, como preço do dólar e índices de ações internacionais, e (ii) não é possível usar uma estratégia que se fundamenta no *trading* de alta freqüência, em ações com baixa liquidez, ou seja, onde os custos de *slippage*, ou a diferença entre o preço de compra e o preço de venda, são extremamente elevados. O primeiro item está diretamente ligado à capacidade de treinar e usar modelos não-paramétricos de previsão, como as redes neurais que vamos usar adiante. O

segundo item está relacionado à capacidade de ter estratégias onde os lucros não sejam dominados pelos custos de transação. Assim, o segundo critério de corte vai ser a liquidez.

Definidos estes dois critérios, o próximo passo é examinar os principais índices de ações do mercado: o Ibovespa e o IBX-100. Ambos tem como critério de construção a liquidez das ações, facilitando nossa busca[5]. Uma vez levantados os componentes de ambos, restou um conjunto de 100 ações diferentes – note-se que não são 100 empresas diferentes, pois muitas estão presentes nos índices com suas ações ON (ordinárias, ações com direito a voto) e também PN (participativas, sem direito a voto). Temos 85 companhias diferentes, não excluindo casos de um mesmo grupo econômico, exemplificado pelo grupo siderúrgico Gerdau, que aparece como Gerdau SA (GGBR3 e 4) e Metalúrgica Gerdau (GOAU4).

Uma vez determinadas essas cem ações, o próximo passo era levantar seus dados. Para tanto, era necessário selecionar um período histórico. Optamos pelo período fechado de 3 anos, do **dia 02 de janeiro de 2004 até o dia 28 de dezembro de 2006** perfazendo um total de **744 observações**. O leitor pode perguntar: visto que quanto mais dados melhor, porque não estender o período para mais longe, quiçá 2003 ou 2002. A resposta está ligada aos fatos que ocorreram no mercado brasileiro nesses anos: 2002 foi um ano de volatilidade extrema, com fortes quedas causadas pela crise envolvendo as eleições presidenciais, enquanto 2003 foi um ano de recuperação da crise anterior, com o mercado subindo violentamente (o Ibovespa teve retorno de mais de 100%). Assim, os dois anos não são situações de equilíbrio de mercado, onde uma estratégia do tipo *market-neutral*, como a que desenvolvemos aqui, seria a mais adequada para gerar retornos. Além disso, quanto mais para longe formos, menor o grupo de ações disponível para trabalharmos[6]. A opção foi por usar apenas ações que tivessem suas séries temporais próximas da completude, ou seja, com as 744 observações. Há casos (Guararapes ON por exemplo), em que a ação tinha menos liquidez no passado e por isso não há registro do preço em algumas datas. Se o número de dados se aproximava dos 744, incluímos a ação em nossa base e substituímos o preço vazio pelo último disponível. Uma outra decisão importante foi o uso de **preços médios**, e não preços de fechamento. Isto se explica pela necessidade de aproximar da realidade do dia a dia de uma mesa de operações: ao tomarmos uma decisão de compra e venda, o preço médio é o que mais se aproxima do preço que um operador conseguirá executar, enquanto que o preço de fechamento não representa bem os negócios ao longo do dia, especialmente em dias de alta volatilidade. Além disso, dentro do conceito de marcação a mercado, ou "*mark-to-market"*, as cotas de fundos de investimento têm suas ações marcadas ao

[5] Vide www.bovespa.com.br/Mercado/RendaVariavel/Indices/FormConsultaApresentacaoP.asp?Indice=Ibovespa e www.bovespa.com.br/Mercado/RendaVariavel/Indices/FormConsultaApresentacaoP.asp?Indice=IBrX para maiores detalhes de cada um dos índices, seus componentes e metodologias de construção. Um ponto importante é que estes índices passam por rebalanceamentos periódicos. Para nossos propósitos, usamos os índices válidos para o trimestre setembro – dezembro de 2006.

[6] Ainda assim, não é uma série curta demais. Burgess (2000) trabalha com séries de 600 dados, divididos entre 400 "*in-sample"* e 200 *"out-of-sample"*, série menor que a utilizada aqui.

preço médio do dia, e não ao de fechamento. Por fim, todos os preços de ações serão **ajustados por proventos**, ou seja, eventos como dividendos, juros sobre capital próprio, agrupamentos ou *splits*, terão seus efeitos excluídos da base. Infelizmente, essa decisão afasta o trabalho marginalmente da realidade, principalmente na nossa última etapa de demonstrar o retorno dos modelos, mas foi tomada pela ausência de uma base de dados consolidada desses eventos.

Uma vez definidos os parâmetros relevantes, recorremos a uma base de dados de preços de ações com todas as características pré-definidas[7], e para as 100 ações levantamos o histórico. A partir daí, retiramos aquelas ações que não tinham séries longas o suficiente (por exemplo: UOL PN e Cosan ON, que abriram o capital em fins de 2005). Além disso, devido ao critério de liquidez, levantamos o volume negociado de cada uma das ações, e calculamos a mediana do volume nos últimos 120 dias, excluindo aquelas cujo volume diário era inferior a um milhão de reais (exemplo de Light ON que negocia por volta de 900 mil reais / dia). Uma vez aplicados estes filtros, restaram 55 ações, representando 17 setores econômicos distintos, o que traz boa dose de diversidade. As ações selecionadas são apresentadas na Tabela 4.1. Ali temos o código de cada ação (e doravante vamos se referir a elas sempre pelo código), o nome, classificação setorial, o número de dados disponíveis e a mediana de volume dos últimos 120 dias úteis (período entre 06 de julho de 2006 e 28 de dezembro de 2006). Optamos por não colocar a denominação do tipo (ON, PN, PNA, etc.) para não dificultar a consulta.

Levantados os dados, vamos fazer uma análise estatística preliminar. Embora nos modelos de cointegração usados para construção das cestas de ações o *input* básico seja o preço da ação (dentro da hipótese de que ela cointegra com as outras ações do modelo), aqui vamos analisar as estatísticas dos log-retornos. Vamos calcular os 4 primeiros momentos dos retornos (média, desvio-padrão, assimetria e curtose). Vamos rodar um teste Jarque-Bera de normalidade[8], e também um teste Dickey-Fuller aumentado[9] para a estacionariedade dos retornos[10]. No teste Jarque-Bera, o valor crítico para 5% é 6,63 e para o ADF o valor crítico para 5% é -2,87. Na Tabela 4.2 apresentamos os resultados da análise estatística (as células em cinza indicam rejeição da hipótese de normalidade dos retornos). Note-se que a maioria das ações tem retornos não-normais, e também como todas as séries temporais são estacionárias nos retornos (valores na coluna *ADF* maiores que o crítico).

---

[7] Foi utilizado o sistema *BDS*, disponível para o autor em seu local de trabalho. Detalhes, vide www.bdscorp.com.br .

[8] Para detalhes sobre testes de normalidade, vide Zivot & Wang (2003), pp. 61-63.

[9] Vide item 2.3.2 deste trabalho para detalhes sobre testes de estacionariedade.

[10] Todos os testes serão implementados em *MATLAB*, usando o "*Econometrics Toolbox*". Para mais informações, vide www.spatial-econometrics.com e LeSage (1999).

	Código	Nome	Setor	Nº Dados	Mediana Vol. 120d
1	ARCZ6	Aracruz	Papel e Celulose	744	9.491.796,50
2	BBAS3	Banco do Brasil	Finanças	744	26.850.963,00
3	BBDC4	Bradesco	Finanças	744	76.099.476,50
4	BRAP4	Bradespar	Mineração	744	24.544.520,00
5	BRKM5	Braskem	Petroquímica	744	19.561.779,50
6	BRTO4	Brasil Telecom (Oper.)	Telefonia	744	9.048.887,00
7	BRTP3	Brasil Telecom (Part.)	Telefonia	744	3.833.938,50
8	BRTP4	Brasil Telecom (Part.)	Telefonia	744	5.861.799,50
9	CCRO3	CCR	Concessões	744	12.260.162,50
10	CGAS5	Comgás	Gás	744	2.837.318,60
11	CLSC6	Celesc	Energia	744	5.057.285,00
12	CMIG3	Cemig	Energia	744	1.243.145,00
13	CMIG4	Cemig	Energia	744	32.340.805,00
14	CNFB4	Confab	Indústria	744	2.594.705,50
15	CPLE6	Copel	Energia	744	14.597.383,00
16	CRUZ3	Souza Cruz	Consumo	744	5.334.128,00
17	CSNA3	CSN	Siderurgia	744	35.794.169,50
18	DURA4	Duratex	Construção	744	6.111.751,50
19	EBTP4	Embratel	Telefonia	744	10.904.390,50
20	ELET3	Eletrobrás	Energia	744	20.118.952,50
21	ELET6	Eletrobrás	Energia	744	23.766.239,00
22	EMBR3	Embraer	Aviões	744	9.867.483,00
23	FFTL4	Fosfértil	Fertilizantes	744	1.190.048,00
24	GETI4	AES Tietê	Energia	678	2.673.624,00
25	GGBR4	Gerdau	Siderurgia	744	31.065.165,00
26	GOAU4	Gerdau Metalúrgica	Siderurgia	744	11.585.585,00
27	GUAR3	Guararapes	Varejo	641	3.446.807,00
28	ITAU4	Itaú	Finanças	744	58.574.087,50
29	ITSA4	Itaúsa	Finanças	744	29.659.070,00
30	KLBN4	Klabin	Papel e Celulose	744	5.433.390,00
31	LAME4	Lojas Americanas	Varejo	743	9.739.588,50
32	PCAR4	Pão de Açúcar	Varejo	744	8.987.491,00
33	PETR3	Petrobrás	Petróleo	744	47.902.646,50
34	PETR4	Petrobrás	Petróleo	744	269.905.529,50
35	POMO4	Marcopolo	Indústria	744	1.051.984,00
36	PTIP4	Petróleo Ipiranga	Consumo	744	3.562.626,00
37	RAPT4	Randon	Indústria	744	1.990.176,50
38	SBSP3	Sabesp	Saneamento	744	9.486.709,35
39	SDIA4	Sadia	Consumo	744	9.544.460,00
40	TBLE3	Tractebel	Energia	744	5.185.827,00
41	TCSL4	TIM Participações	Telefonia	744	10.296.224,00
42	TLPP4	Telesp Fixa	Telefonia	744	3.668.674,00
43	TMAR5	Tele Norte Leste	Telefonia	744	7.181.760,00
44	TMCP4	Telemig Celular	Telefonia	744	2.744.881,00
45	TNLP3	Telemar	Telefonia	744	30.366.212,50
46	TNLP4	Telemar	Telefonia	744	45.669.317,00
47	TRPL4	Transmissão Paulista	Energia	744	3.030.171,50
48	UBBR11	Unibanco	Finanças	744	27.268.176,00
49	UGPA4	Ultrapar	Petroquímica	743	1.731.594,00
50	UNIP6	Unipar	Petroquímica	744	2.158.307,00
51	USIM3	Usiminas	Siderurgia	711	1.122.148,50
52	USIM5	Usiminas	Siderurgia	744	73.763.242,00
53	VALE3	Vale do Rio Doce	Mineração	744	43.279.246,00
54	VALE5	Vale do Rio Doce	Mineração	744	149.859.034,50
55	VCPA4	Votorantim Celulose e Papel	Papel e Celulose	744	9.153.119,00

Tabela 4.1: Lista de Ações

	Código	Média	Desvio	Assimetria	Curtose	Jarque-Bera	ADF
1	ARCZ6	0,00056	0,01571	0,09696	3,25824	3,224	(30,5753)
2	BBAS3	0,00159	0,02289	0,20531	5,21285	156,602	(29,9927)
3	BBDC4	0,00182	0,01867	0,01665	3,41697	5,409	(29,1510)
4	BRAP4	0,00171	0,02089	(0,15252)	3,39969	7,816	(30,9054)
5	BRKM5	(0,00011)	0,02510	(0,05462)	4,99143	122,978	(27,4774)
6	BRTO4	(0,00016)	0,02272	0,43094	5,05379	153,373	(30,3397)
7	BRTP3	0,00107	0,02833	0,02003	8,71215	1.008,820	(32,0753)
8	BRTP4	0,00009	0,01901	0,18691	4,15914	45,860	(30,8435)
9	CCRO3	0,00218	0,02154	0,00184	4,01621	31,928	(32,5619)
10	CGAS5	0,00122	0,02200	(0,16544)	4,43856	67,365	(32,9306)
11	CLSC6	0,00117	0,02148	0,21262	4,27260	55,661	(30,2349)
12	CMIG3	0,00155	0,02140	0,01913	3,08906	0,290	(33,9917)
13	CMIG4	0,00122	0,02204	0,11775	3,18856	2,814	(29,1916)
14	CNFB4	0,00160	0,01974	0,10025	4,34345	57,043	(29,5832)
15	CPLE6	0,00090	0,02247	(0,00119)	3,81139	20,354	(31,1152)
16	CRUZ3	0,00058	0,01861	(0,03561)	3,72948	16,609	(32,3439)
17	CSNA3	0,00116	0,02295	(0,06739)	3,47693	7,594	(29,4795)
18	DURA4	0,00197	0,02248	0,34921	5,60633	225,097	(32,4691)
19	EBTP4	(0,00018)	0,02503	0,48798	8,21062	868,851	(29,7556)
20	ELET3	(0,00007)	0,02836	(0,73751)	5,95923	338,003	(29,4432)
21	ELET6	0,00020	0,02679	(0,77724)	7,01343	572,700	(29,5588)
22	EMBR3	0,00034	0,01823	0,16480	4,14606	43,966	(33,5885)
23	FFTL4	0,00075	0,01325	0,30352	5,26244	169,644	(33,3239)
24	GETI4	0,00259	0,02089	2,12401	23,58872	13.663,337	(35,5592)
25	GGBR4	0,00151	0,02193	0,00277	3,22748	1,601	(30,0079)
26	GOAU4	0,00192	0,02144	0,17340	3,63371	16,134	(27,5577)
27	GUAR3	0,00320	0,01817	0,37417	6,89495	486,338	(34,7179)
28	ITAU4	0,00146	0,01763	0,03711	3,46327	6,806	(30,2960)
29	ITSA4	0,00169	0,01766	0,15873	3,76124	21,032	(31,5012)
30	KLBN4	0,00067	0,02037	0,14244	3,72455	18,740	(32,1569)
31	LAME4	0,00192	0,02155	(0,05510)	3,87246	23,909	(32,3420)
32	PCAR4	(0,00006)	0,01977	(0,20520)	5,60675	215,290	(32,9385)
33	PETR3	0,00142	0,01745	(0,12161)	4,19828	46,221	(30,0606)
34	PETR4	0,00145	0,01699	(0,15567)	3,96706	31,910	(30,4632)
35	POMO4	0,00139	0,01759	0,11412	5,05800	132,553	(36,5592)
36	PTIP4	0,00164	0,02241	(0,00334)	4,35456	56,728	(31,3926)
37	RAPT4	0,00179	0,01894	0,15845	5,41787	183,846	(32,9276)
38	SBSP3	0,00105	0,02076	0,30245	5,32571	178,538	(32,5395)
39	SDIA4	0,00099	0,02159	(0,19745)	3,77396	23,341	(28,5687)
40	TBLE3	0,00133	0,02681	1,93740	23,81937	13.864,903	(28,5602)
41	TCSL4	0,00090	0,02284	0,07976	3,62395	12,823	(32,9063)
42	TLPP4	0,00080	0,01588	0,16001	3,88966	27,637	(32,9539)
43	TMAR5	0,00012	0,01989	0,03242	3,54723	9,388	(30,4313)
44	TMCP4	(0,00003)	0,02248	0,55852	5,63770	253,680	(29,7737)
45	TNLP3	0,00171	0,03694	7,59028	145,14532	631.804,98	(35,6733)
46	TNLP4	(0,00025)	0,01725	(0,06259)	3,76163	18,419	(31,1557)
47	TRPL4	0,00127	0,02471	0,03783	4,21410	45,750	(31,3869)
48	UBBR11	0,00160	0,01965	(0,00191)	3,38532	4,591	(30,7639)
49	UGPA4	0,00079	0,01624	(0,03466)	4,43855	64,128	(32,5076)
50	UNIP6	0,00035	0,01844	0,06352	3,90387	25,757	(29,0975)
51	USIM3	0,00209	0,02415	0,08284	4,47809	68,394	(33,3125)
52	USIM5	0,00145	0,02662	(0,25564)	3,55878	17,735	(26,9297)
53	VALE3	0,00120	0,01960	(0,02416)	3,38851	4,739	(31,3826)
54	VALE5	0,00119	0,01855	(0,12404)	3,29028	4,508	(29,1076)
55	VCPA4	0,00043	0,01638	0,17796	4,54349	77,571	(32,9236)

Tabela 4.2: Propriedades Estatísticas das Ações

4.2.2. Seleção das Variáveis Exógenas

A construção de um modelo de previsão multivariado não-paramétrico, como as redes neurais que desenvolveremos adiante, depende fundamentalmente da escolha de bons dados para serem usados como *inputs*. Aqui o velho truísmo "*garbage in, garbage out*" é mais verdadeiro do que nunca. Nossa capacidade de realizar boas previsões é diretamente dependente da qualidade dos dados, se estes contém informações úteis para a previsão das dinâmicas futuras das cestas de ações. Seguindo Parreiras (2003)[11] com algumas variações, escolhemos as seguintes variáveis:

- *Cotação do dólar*: após 1999, a cotação do dólar (ou seja, a relação entre o real e o dólar) flutua livremente no mercado, e dá indicações fortes dos rumos econômicos do país e portanto dos mercados, podendo indicar também as tendências para a Bolsa de Valores. Neste trabalho, será utilizada a cotação diária PTAX das 16:30, publicada pelo Banco Central do Brasil, representando o fechamento diário do mercado de câmbio.
- *Taxas de Juros de curto e longo prazo*: além da cotação do dólar, outro preço fundamental da economia é dado pela taxa de juros. Assim, é útil saber sua evolução, que pode indicar crescimento ou recessão, afetando diretamente as possibilidades de aumento dos lucros das empresas e portanto da valorização de suas ações. Para os fins deste trabalho, serão utilizadas duas *proxies* da taxa de juros da economia: uma para indicar a taxa de juros de curto prazo, outra para indicar a taxa de juros de longo prazo. São dadas pelas cotações de *swap pré x DI* para 90 dias e 360 dias. Essas cotações indicam efetivamente as taxas em que as empresas de grande porte conseguem tomar empréstimos para financiar seus investimentos e capital de giro, e são calculadas diariamente pela Bolsa de Mercadorias e Futuros, a BM&F.
- *Índices de Bolsa de Valores nacionais e internacionais:* além de dados específicos de cada ação, é útil ter informações sobre o mercado em geral, tanto o brasileiro quanto o americano, o maior mercado de ações do mundo. Para tanto, séries de dados do índice Bovespa (ou Ibovespa), do S&P 500[12] e do Nasdaq[13] serão utilizadas para agregar informação sobre a performance das bolsas de valores, que se supõe podem afetar os preços das cestas de ações.

Todas as informações foram levantadas em bases de dados como BDS e Bloomberg, para o mesmo período dos preços de ações. Devido a diferenças entre os calendários americano e brasileiro, as séries temporais foram normalizadas para os mesmos dias, com base nas datas onde havia preços de ações disponíveis. Quando um dos dados não estava disponível, foi

[11] Parreiras (2003), pp. 81-82.

[12] Índice calculado pela Standard & Poors, com as 500 principais empresas do mercado acionário americano. É geralmente tomado como a melhor proxy da performance do mercado. Vide www.standardandpoors.com .

[13] Índice das 100 principais empresas de tecnologia. Vide www.nasdaq.com .

utilizada sua cotação mais recente. Assim, cada uma dessas séries temporais também contém 744 dados. Na Tabela 4.3 resumimos essas informações.

Nome	Descrição	Fonte
S&P 500	Índice de Ações representativo do mercado americano.	Standard & Poors
Nasdaq 100	Índice das 100 principais empresas de tecnologia.	Nasdaq
Índice IBOVESPA	Índice da Bolsa de São Paulo.	Bovespa
Dólar Comercial	Cotação de fechamento do mercado de câmbio do Brasil	Banco Central
DIxPRE 0090	Taxa de juros (*swap*) de curto prazo - 90 dias.	BM&F
DIxPRE 0360	Taxa de juros (*swap*) de longo prazo - 360 dias.	BM&F

Tabela 4.3: Variáveis Exógenas

Os mesmos procedimentos estatísticos apresentados no item 4.2.1 foram aplicados às variáveis exógenas, com os resultados apresentados na Tabela 4.4.

Nome	Média	Desvio	Assimetria	Curtose	Jarque-Bera	ADF
S&P 500	(0,00040)	0,00835	0,69687	7,71727	748,03	(39,9205)
Nasdaq 100	(0,00028)	0,00376	1,95177	18,81440	8.203,21	(39,1725)
Índice IBOVESPA	(0,00033)	0,00854	1,68480	24,09749	14.112,17	(42,1389)
Dólar Comercial	0,00092	0,01633	(0,18303)	3,66406	17,78	(35,0017)
DIxPRE 0090	0,00034	0,00657	(0,04498)	3,23024	1,89	(37,8727)
DIxPRE 0360	0,00025	0,01011	0,01383	3,27477	2,36	(38,0844)

Tabela 4.4: Propriedades Estatísticas das Variáveis Exógenas

Note-se como novamente boa parte das variáveis tem retornos não-normais, com exceção das taxas de juros, e também como são todas estacionárias nos retornos.

Agora que escolhemos e analisamos previamente nossas variáveis de trabalho, e estando com os bancos de dados prontos, vamos a seguir detalhar o modelo de construção de cestas de ações para arbitragem estatística, aplicando as técnicas de cointegração.

4.3. Construção das Dinâmicas de Arbitragem Estatística

4.3.1. Metodologia de Construção de Cestas de Ações

O conceito de cointegração que discutimos amplamente no Capítulo 2 forma a base da nossa metodologia de construção de cestas de ações passíveis de arbitragem estatística, mas vamos expandir a metodologia para nos adequar a algumas necessidades práticas. Parte dessa metodologia se inspira em Burgess (2000), mas não vamos adotar aqui todas as técnicas desenvolvidas por ele, notadamente no que diz respeito ao uso da análise do perfil da razão de variância das dinâmicas de arbitragem estatística[14].

[14] Para detalhes sobre "análise do perfil da razão de variância", vide Burgess (2000), pp. 145-176. Essa técnica foi desenvolvida nos anos 80 para discussão da eficiência dos mercados, vide Campbell *et al*. (1997), pp. 48-55.

De um modo geral, a inspiração para a análise de valor relativo que propomos aqui é o fato de que os preços dos ativos, quando vistos de uma maneira relativa (ou seja, combinados entre si), e não absoluta, são mais aptos a serem previstos, e portanto arbitrados. De fato, vamos ver como em princípio, combinações de ativos bem construídas podem ser largamente imunizadas com relação aos fatores gerais de risco de mercado, destacando os fatores idiossincráticos na dinâmica dos preços dos ativos, que podem, potencialmente, ser mais previsíveis.

Em analogia com a discussão dos itens 2.2.3 e 2.2.4, o objetivo de nossa metodologia é identificar combinações de ativos que representem estatisticamente relações de "preço justo", nos quais podemos basear estratégias de arbitragem. Especificamente, dado um conjunto de ativos U_A, e um particular "ativo-objeto" $T \in U_A$, nosso objetivo é construir um "ativo-sintético" $SA(T)$, de modo que este possa fornecer, estatisticamente, um "preço justo" para o ativo-objeto:

$$E[T_t] = SA(T)_t \tag{4.1}$$

Mais ainda, relação de preço justo da equação 4.1 deve ser de tal modo que os desvios dessa relação possam ser considerados "erros de preço"[15]:

$$M_t = T_t - SA(T)_t \tag{4.2}$$

onde a dinâmica da série temporal dos erros de preço M_t contém um componente previsível, que pode ser explorado como base de uma estratégia de arbitragem estatística.

A partir desse ponto de partida, a metodologia se baseia no uso de técnicas de cointegração para estimar as relações justas entre os preços dos ativos. Uma regressão de cointegração, como discutimos no item 2.3.5, é usada para estimar uma combinação linear de ativos que exibe a correlação máxima possível com o ativo-objeto T. Os coeficientes da combinação linear são estimados regredindo os preços históricos de T contra os preços históricos de um conjunto de ativos constituintes $C \subset U_A - T$:

$$SA(T)_t = \sum_{C_i \in C} \beta_i \cdot C_{i,t} \qquad s.a. \qquad \{\beta_i\} = \arg\min \sum_{t=1\ldots n} \left(T_t - \sum_{C_i \in C} \beta_i \cdot C_{i,t} \right)^2 \tag{4.3}$$

e o vetor de cointegração $\beta = [\beta_1, \ldots, \beta_n]^T$ de pesos dos ativos constituintes é dado por:

$$\beta_{OLS} = \left(C^T C\right)^{-1} Ct \tag{4.4}$$

onde C é a matriz de preços históricos dos ativos constituintes e $t = [T_1, \ldots, T_n]^T$ é o vetor de preços do ativo-objeto.

Assim, o ativo-sintético pode ser considerado um *hedge* estatisticamente ótimo, condicionado ao conjunto C de ativos constituintes, na medida em que as propriedades padrão

[15] Vide nota de rodapé 8 no Capítulo 2 para origem do termo.

do método de mínimos quadrados ordinários garantem que o ativo-sintético será um estimador não-viesado para o ativo-objeto[16], ou seja, $E[T_t] = SA(T)_t$, e também que o desvio entre as duas séries será mínimo, num contexto de erros quadrados.

Dado esse modelo, podemos derivar a série temporal dos erros de preço como sendo:

$$M_t = T_t - \sum_{C_i \in C} \beta_i \cdot C_{i,t} \tag{4.5}$$

e é possível interpretar o erro de preço estatístico M_t como sendo um portfólio composto, consistindo dos ativos $\{T, C_1, C_2, \ldots, C_n\}$ com pesos $\{1, -\beta_1, -\beta_2, \ldots, \beta_n\}$ respectivamente. O preço deste portfólio representa o valor em excesso do ativo-objeto T em relação à combinação ponderada dos ativos C_i.

Num contexto de mercado, podemos ainda considerar que os ativos C_i representam aproximações para os fatores de risco de mercado que impactam os preços dos ativos. Essa interpretação aparece em alguns autores, como Burgess (2000) e Vidyamurthy (2004). Ao maximizar a correlação entre ativo-objeto e o ativo-sintético, por construção estamos minimizando a sensibilidade do portfólio a essas fontes comuns de risco, e por outro lado mantendo constante a exposição aos fatores idiossincráticos de risco[17]. Assim, o objetivo primordial dessa etapa da metodologia é construir combinações de séries temporais que sejam tanto descorrelacionadas das principais fontes de risco como contenham um componente determinístico (potencialmente previsível) na sua dinâmica. O primeiro objetivo é desejável pois aumenta a diversificação, tanto num conjunto de estratégias de arbitragem estatística, quanto num contexto mais amplo onde há outras estratégias de investimento envolvidas. O segundo objetivo simplesmente reconhece o fato de que o tamanho do componente determinístico representa um limite superior à performance possível de um modelo de arbitragem estatística.

Vale dizer que num contexto onde os ativos C_i representam fatores de risco de mercado, nossa metodologia poderia desenvolver um modelo multifatorial como no *APT*[18]. Uma técnica para encontrar os fatores é a Análise de Componente Principais. Contudo, nossa metodologia baseada em cointegração tem algumas vantagens práticas importantes. A maior delas é que os modelos podem ser condicionados à escolha de um ativo-objeto e de um conjunto (relativamente pequeno) de ativos-constituintes, enquanto que usando *PCA* teríamos todos os

[16] Desde que as séries sejam cointegradas, como visto na discussão do item 2.3.5. Para propriedades do estimador por mínimos quadrados ordinários, vide Kennedy (2003), pp. 47-56.

[17] Burgess (2000), pp. 122-126 faz um exemplo com três ativos, mostrando como a metodologia mantém na dinâmica de erros de preço majoritariamente os fatores específicos de risco, imunizando o portfólio dos fatores comuns.

[18] "*Asset Pricing Theory"*, desenvolvida por Stephen Ross. Vide Campbell *et al.* (1997), pp. 219-251.

ativos incluídos em todos os fatores, o que torna essa técnica pouco transparente e difícil de gerenciar quando se trabalha com cinqüenta e cinco ativos diferentes, como é nosso caso.

Uma complicação prática na metodologia é o fato de as relações de "preço justo" entre um ativo-objeto e seu ativo-sintético poderem ser instáveis ao longo do tempo. Esta pode advir tanto de uma variação temporal nos coeficientes do vetor de cointegração, quanto de erros nos estimadores. A significância dentro da amostra usada para estimação não é suficiente para garantir que o comportamento futuro do "*mispricing*" será semelhante ao passado, ou seja, mesmo modelos que funcionavam bem estão sujeitos a "quebras", ou degradação de performance devido a não-estacionariedades. A possível instabilidade advinda do procedimento de estimação pode ser causada pela presença de multi-colinearidade no conjunto de ativos constituintes. Esse perigo é aumentado pelo fato de o número de regressores ser grande, pois o número de correlações entre eles é $\mathrm{O}\left(N^2\right)$. Para atacar essa questão, vamos recorrer à metodologia de Burgess (2000), que envolve uma técnica com sucessivas regressões, introduzindo um ativo de cada vez no cálculo de $SA(T)$, o que facilita bastante as coisas quando se lida com ativos às dezenas.

A idéia aqui é que para reduzir a dimensionalidade do problema, e identificar relações entre pequenos subconjuntos dos dados. Para garantir que analisaremos o espaço inteiro, tomamos uma ação como ativo-objeto de cada vez. Para identificar o subespaço mais apropriado de constituintes, ao invés de rodar uma regressão do tipo "coloque todas as variáveis", vamos adotar uma técnica *"stepwise"*, onde um ativo entra na regressão de cada vez, com a complexidade do modelo aumentando seqüencialmente. A cada passo, a variável adicionada ao modelo vai ser aquela mais correlacionada com a atual série de desvios, e portanto, quando for colocada no modelo, levará à maior diminuição da variância residual. Voltando à interpretação econômica, podemos dizer que essa metodologia *"stepwise"* inclui como primeira variável aquela ação que tem a exposição aos fatores de risco de mercado mais similar à do ativo-objeto. Já a segunda ação será aquela que é mais correlacionada com a exposição residual, após o ativo-objeto ter sido "limpado" da exposição inicial, e assim sucessivamente.

A metodologia *"stepwise"* pode ser formalizada da seguinte maneira: começando do ativo-objeto apenas, com $C(0)=\varnothing$ e $M_t^{(0)}=T_t$, o conjunto de ativos-constituintes é construído passo a passo. Seja $M_t^{(k)}$ a série dos erros de preço na etapa *k*, e temos que:

$$M_t^{(k)} = T_t - \sum_{C_i \in C(k)} \beta_i^{(k)} \cdot C_{i,t} \tag{4.6}$$

então o modelo do passo *k+1* é determinado pela identificação da variável ainda não incluída no modelo que tem maior correlação com $M_t^{(k)}$, ou em outras palavras, que responde pela maior redução de variância no resíduo atual:

$$C(k+1) = C(k) \cup \underset{C_j \in U_a - C(k)}{\arg\max}\, E\left[M_t^{(k)^2} - \left(M_t^{(k)} - \beta_j \cdot C_j\right)^2\right] \tag{4.7}$$

e uma vez encontrada a ação que deve ser adicionada ao modelo, o vetor β é re-estimado, de maneira a levar em conta os efeitos de possíveis correlações cruzadas entre as variáveis.

Os passos continuam até que um critério de parada seja atingido. Pode ser um número máximo de ativos incluído no modelo, ou que não haja mais relações estatisticamente significantes entre o resíduo e os ativos restantes. Em nossos modelos, vamos utilizar o primeiro dos critérios. Vamos trabalhar com no máximo 4 (quatro) ativos dentro dos ativos-sintéticos[19], e usar testes *t* para atestar a significância de todos os coeficientes estimados.

Resumindo esquematicamente a metodologia para construção de cestas de ações passíveis de arbitragem estatística, temos a Figura 4.1, mostrando cada uma das etapas descritas acima.

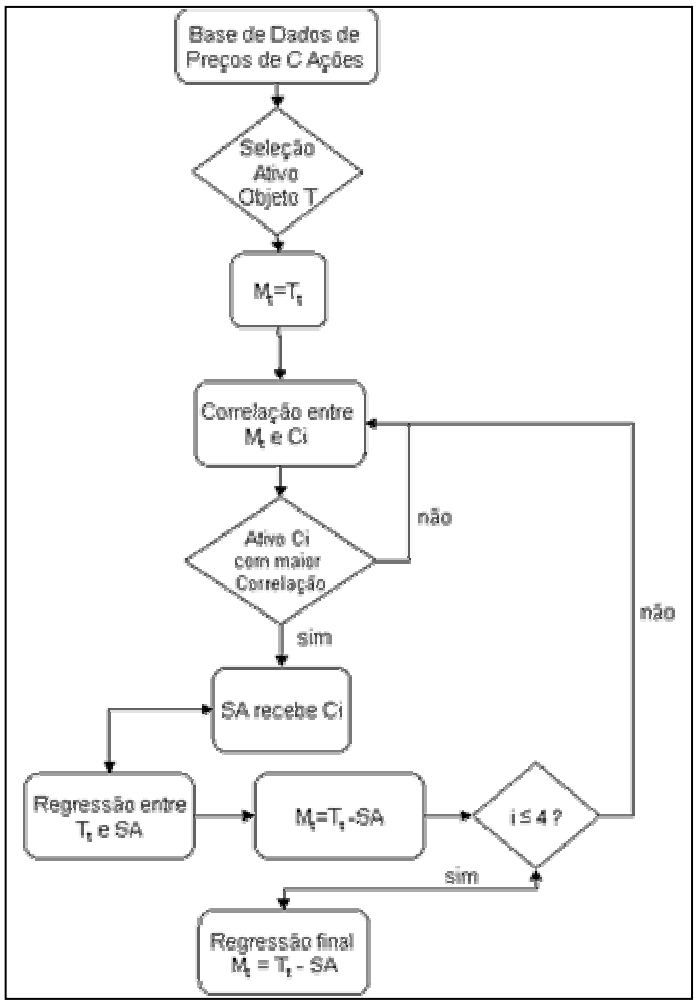

Figura 4.1: Esquema da Metodologia de Construção de Cestas de Ações

O modelo foi implementado em *MATLAB*, e os detalhes do código estão no Apêndice A1.

[19] Burgess (2000) trabalha com até cinco ativos. Como nosso universo é menor e menos diverso que o dele, decidimos reduzir o número de ativos sob consideração.

Vamos agora ilustrar a operação da metodologia de construção das cestas de ações através de um exemplo, usando os preços das 55 ações que selecionamos no item 4.2.1.

Como já dissemos, o objetivo da metodologia é construir ativos sintéticos que representem estatisticamente cada uma das ações tomadas individualmente. Como demonstrado na Figura 4.1, isto é feito tomando cada ação separadamente e aplicando a metodologia "*stepwise"* para seqüencialmente identificar as quatro ações mais apropriadas em cada caso[20]. Vamos trabalhar com a base de dados completa, ou seja, os 744 dados para cada uma das 55 ações, no período de 02 de janeiro de 2004 até 28 de dezembro de 2006. Nesse caso vamos usar as séries temporais completas, não separando os dados entre fora e dentro da amostra, visto que o propósito agora é apenas demonstrar o funcionamento da metodologia implementada.

A Tabela 4.5 mostra a especificação para os primeiros 10 ativos-sintéticos, listando primeiro o ativo-objeto e depois seqüencialmente cada um dos componentes e seus pesos.

	Ativo Objeto	Comp. 1	Peso 1	Comp. 2	Peso 2	Comp. 3	Peso 3	Comp. 4	Peso 4
1	ARCZ6	SBSP3	0,0235	UNIP6	0,2089	EBTP4	0,1431	VCPA4	0,1402
2	BBAS3	ITSA4	4,9561	PCAR4	0,0362	FFTL4	0,0201	KLBN4	0,6450
3	BBDC4	UBBR11	4,9244	TMAR5	(0,1711)	BRTP3	0,1231	RAPT4	0,1198
4	BRAP4	ITSA4	5,8795	VCPA4	(0,4241)	POMO4	7,2845	FFTL4	0,3709
5	BRKM5	UNIP6	10,3428	BRTO4	0,5275	FFTL4	(0,1619)	VCPA4	(0,0849)
6	BRTO4	BRTP4	0,5493	TRPL4	(0,0959)	PCAR4	0,0299	CRUZ3	0,0350
7	BRTP3	VALE3	0,3353	BRKM5	(0,0815)	BRTP4	(0,0055)	FFTL4	0,4635
8	BRTP4	BRTO4	1,1341	CRUZ3	0,0288	BRKM5	0,0398	TMCP4	0,6546
9	CCRO3	ITAU4	0,2777	POMO4	1,5516	TMAR5	0,0768	BRTP4	(0,4841)
10	CGAS5	PTIP4	5,0512	FFTL4	2,6783	TBLE3	4,8788	UGPA4	1,2101
11	CLSC6	ITSA4	2,2166	VCPA4	0,0608	TMAR5	(0,0243)	SDIA4	1,8028
12	CMIG3	CMIG4	0,6534	UGPA4	(0,2995)	PTIP4	1,4641	POMO4	1,3895
13	CMIG4	CMIG3	1,0608	TMCP4	2,5525	BRKM5	0,1621	ELET3	(0,0760)
14	CNFB4	GETI4	0,0661	CRUZ3	0,0113	KLBN4	0,0177	UGPA4	0,0036
15	CPLE6	ITSA4	1,8911	EBTP4	0,3029	TMAR5	(0,0022)	EMBR3	0,1643

Tabela 4.5: Exemplos de Ativos Sintéticos construídos com a metodologia

Vamos analisar apenas um caso, o do ativo-objeto BRAP4, ou Bradespar PN. Esta companhia é uma *holding* controlada pelo Banco Bradesco, cujos maiores investimentos são em ações da mineradora Vale do Rio do Doce (mais de 80% do ativo da Bradespar) e também em ações da companhia de energia CPFL[21]. O primeiro componente do ativo sintético é ITSA4, ou Itaúsa, também uma empresa *holding*, esta controlada pelo banco Itaú. Logo, vemos como a metodologia captou a relação de semelhança entre os dois ativos. Outros componentes do ativo sintético são VCPA4 e FFTL4, respectivamente, a empresa de papel e celulose VCP e a companhia de fertilizantes Fosfértil. Ou seja, temos aqui duas companhias de setores básicos e cíclicos, que lidam com *commodities*, cujos fatores de influência, supõe-se, sejam semelhantes

[20] Note que não necessariamente (e de fato apenas raramente) as quatro ações escolhidas serão as quatro ações mais correlacionadas com o ativo-objeto da vez. Depois do primeiro estágio, é a correlação com o *resíduo* que importa, e apenas no primeiro estágio está será equivalente àquela com o próprio ativo-objeto.

[21] Para detalhes, vide www.bradespar.com.br .

ao da Vale do Rio do Doce, principal ativo da Bradespar. A presença de POMO4 dentre os componentes, a empresa fabricante de ônibus Marcopolo, não parece ter uma razão de fácil explicação. Sendo ela apenas a terceira componente do ativo sintético contudo, podemos inferir que a correlação capturada por ela deve ser algum efeito de menor ordem.

Poderíamos realizar esse exercício para todos os outros ativos, buscando as razões da presença de cada uma das componentes do ativo sintético. Contudo, nem sempre as relações são óbvias, podendo estar relacionadas não ao setor de uma dada empresa, mas talvez à sua exposição a variações cambiais, ou ao tamanho do seu passivo por exemplo. Ainda assim, em todos os casos há implícita uma relação entre as duas ações, dados que suas ações se movem de maneira similar, tanto que fez com que o componente tivesse sido escolhido dentre as 54 possibilidades.

Para ilustrar ainda mais o funcionamento da metodologia, vamos explorar detalhes adicionais de dois modelos, mostrando cada etapa da construção *"stepwise"* do ativo sintético.

O primeiro modelo que vamos analisar é para PETR4, na Tabela 4.6. Aqui desejamos mostrar como com apenas um ativo componente já capturamos a maior parte da variância do ativo-objeto. Isso ocorre nesse caso porque o primeiro componente é justamente a PETR3, ou seja, a ação ON da mesma companhia.

Modelo	PETR3	ELET6	ELET3	CRUZ3	Variância	Δ Var.
PETR4	-	-	-	-	104,9681	
SA 1	(0,8931)	-	-	-	0,2714	99,74%
SA 2	(0,8987)	0,0052	-	-	0,2705	99,74%
SA 3	(0,8689)	(0,1757)	0,1403	-	0,2271	99,78%
SA 4	(0,8765)	(0,1788)	0,1262	0,0337	0,2193	99,79%

Tabela 4.6: Construção "*stepwise*" do Ativo-sintético para PETR4

Podemos ver também como os pesos vão mudam à medida que incluímos outras variáveis, embora no caso de PETR3 variem pouco, enquanto que ELET6 muda de sinal quando incluímos ELET3 no modelo. Na Figura 4.2 podemos observar o comportamento da série temporal deste *"mispricing"*. Note como claramente existe um comportamento de reversão à média.

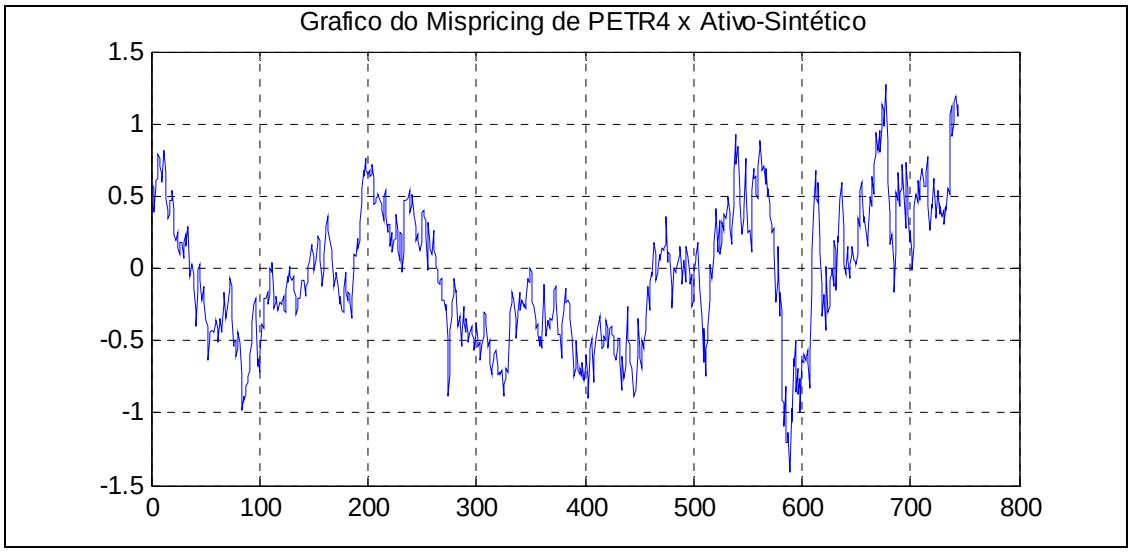

Figura 4.2: Série Temporal do *Mispricing* de PETR4

O segundo modelo que vamos analisar é para CSNA3, da empresa siderúrgica CSN, na Tabela 4.7. Aqui temos um comportamento semelhante, onde o primeiro componente, no caso empresa do mesmo setor, a Usiminas, captura a maior parte da variância do ativo-objeto. Ainda assim, a inclusão de novas variáveis reduz progressivamente a variância do mispricing, embora após o segundo componente a redução já seja marginal.

Modelo	USIM5	EBTP4	TBLE3	UNIP6	Variância	Δ Var.
CSNA3	-	-	-	-	196,9272	
SA 1	(0,9268)	-	-	-	10,7803	94,53%
SA 2	(0,8132)	(1,0736)	-	-	5,8775	97,02%
SA 3	(0,7541)	(0,9600)	(0,3041)	-	5,5833	97,16%
SA 4	(0,7371)	(0,8754)	(0,3665)	(0,3152)	5,5397	97,19%

Tabela 4.7: Construção *"stepwise"* do Ativo-sintético para CSNA3

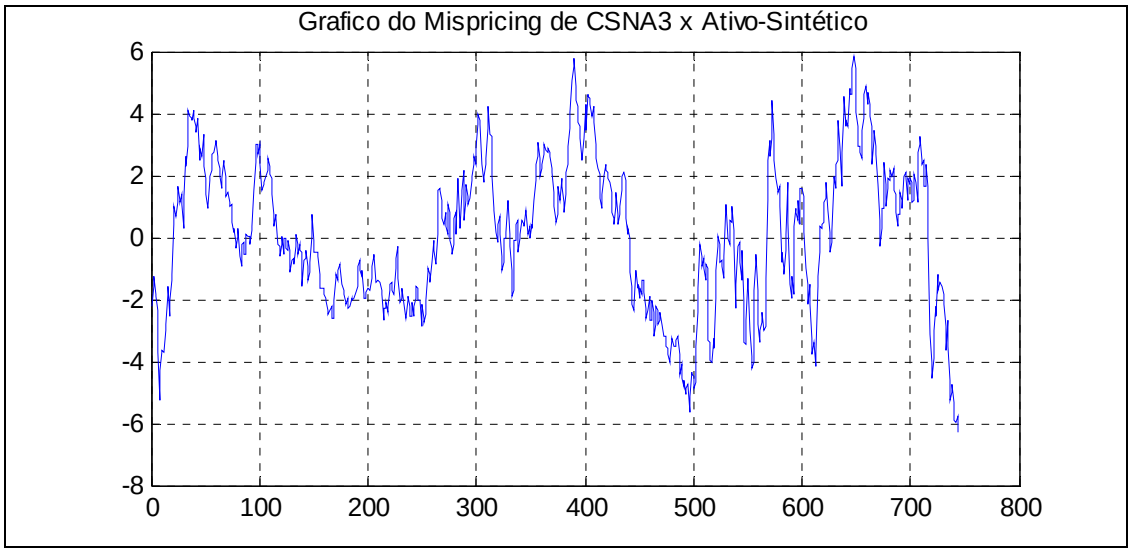

Figura 4.3: Série Temporal do *Mispricing* de CSNA3

Agora que nos familiarizamos com a metodologia de construção das cestas de ações, ou ativos sintéticos, temos que passar para a aplicação da técnica de cointegração, ou seja, testar se as séries temporais dos *"mispricings"* são estacionárias, ou em outras palavras, se os ativos-objeto cointegram com seus respectivos ativos-sintéticos.

4.3.2. Testes de Estacionariedade

A metodologia para construção de ativos-sintéticos não garante, *a priori*, que as dinâmicas dos erros de preços sejam estacionárias. Note que, como discutimos no item 2.3.5, o uso de regressão por mínimos quadrados em séries temporais não-estacionárias só é válido se as séries são cointegradas. Por isso, dentro do contexto da metodologia de Engle-Granger, após estimar (através da metodologia *"stepwise"*) o vetor de cointegração, o próximo passo é testar a estacionariedade dos resíduos. Em outras palavras, vamos testar a estacionariedade da série temporal dos desvios, ou o "*mispricing"*, afinal, a definição dele:

$$M_t = T_t - \sum_{C_i \in C} \beta_i \cdot C_{i,t} \tag{4.8}$$

nada mais é do que um rearranjo da equação de estimação.

Assim, precisamos escolher um teste de estacionariedade adequado e aplicá-lo às séries temporais de cada um dos "*mispricings*". Aqueles que forem considerados estacionários, segundo algum critério de confiança, vamos usar para arbitragem estatística em nossos modelos de *trading*. Aqueles que não forem estacionários vamos descartar, visto que o risco de operar com um modelo não-estacionário é o mesmo que realizar uma aposta aleatória na direção do mercado, e não fazer uso de uma técnica que agregue altos retornos com baixo risco, imune à tendência geral do mercado.

O teste que vamos utilizar em nossos modelos é o Dickey-Fuller aumentado, ou *ADF*, que foi apresentado na seção 2.3.2. A questão de qual nível crítico escolher é um pouco mais delicada: um nível muito estrito, como 1% (ou seja, 99% de confiança), pode excluir modelos demais, nos deixando com poucas oportunidades de arbitragem estatística, enquanto que um nível mais folgado, talvez 10%, pode nos deixar perigosamente próximos de modelos instáveis, que degenerem fora da amostra em que foram estimados. Dada a dúvida, a melhor solução era estimar os modelos e testar diversos níveis de confiança e testar quantos modelos eles selecionam.

O teste foi feito com as 55 ações, usando para estimação um período de dois anos entre 02 de janeiro de 2004 e 29 de dezembro de 2005, para um total de 498 dados. Uma vez estimado cada um dos modelos, aplicamos o teste *ADF* nos resíduos, e vimos para quais modelos podíamos rejeitar a hipótese nula da existência de uma raiz unitária, ou seja, quais modelos são

de fato estacionários. Na Tabela 4.8 podemos ver os resultados das estimativas e dos testes com os diferentes critérios.

Critério Confiança	Nº Modelos Estacionários	% Total
1%	14	25,5%
5%	29	52,7%
10%	34	61,8%

Tabela 4.8: Testes *ADF* com diversos critérios

A Tabela 4.8 mostra como o critério de 5% resulta em 29 modelos estacionários, mais da metade dos disponíveis, um ganho marginal bastante relevante em relação ao critério de 1% de confiança, que tem menos da metade dos modelos. Ou seja, a disponibilidade bem maior de modelos resulta em ganhos de diversificação e aumento de oportunidades aparentemente compensados pela perda de alguma segurança na estacionariedade das nossas estimativas. Já o ganho de 5 modelos adicionais não parece compensar o alargamento do critério para 10%. Assim, até o fim do trabalho, todos os modelos de arbitragem estatística usados serão testados para raiz unitária por Dickey-Fuller aumentado com critério de confiança de 5%.

É importante reforçar a diferença entre estacionário e um não-estacionário. Se soubermos que um *mispricing* é estacionário, uma vez que ele se afaste da sua média, por qualquer razão, podemos ter confiança de que ele retornará ao seu comportamento anterior, e podemos tomar uma posição para lucrar com isso. Já para um *mispricing* não-estacionário, não existe essa confiança, o que impede a tomada de posição. Mostramos as duas situações na Figura 4.4.

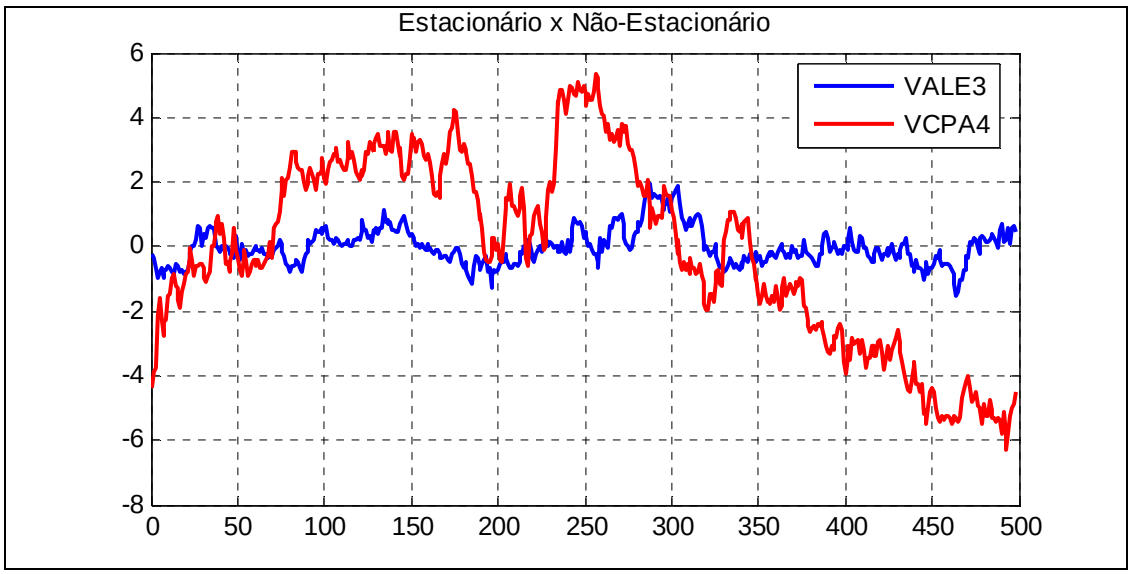

Figura 4.4: Comparação entre modelo estacionário (VALE3) e não-estacionário (VCPA4)

Fica claro como um resíduo não-estacionário tem muita volatilidade e não tende a reverter à média, fazendo com que seja pouco previsível. Logo, as técnicas de previsão que

desenvolvemos aqui não são aplicáveis a ele. Já o modelo de VALE3 claramente reverte à média, permitindo com que os desvios gerem oportunidades de lucro a serem exploradas.

Os testes de estacionariedade completam a parte econométrica da nossa metodologia de arbitragem estatística. Antes de desenvolvermos os modelos de previsão por redes neurais, vamos tratar daquilo que chamamos, inspirados por Burgess (2000), de "arbitragem estatística implícita", a aplicação desses modelos de cointegração diretamente à identificação de oportunidades de *trading*, baseados no conceito de reversão à média.

4.3.3. Arbitragem Estatística Implícita

Os modelos detalhados nos itens anteriores nos permitem estimar e construir séries temporais daquilo que chamamos *"mispricings'* ou seja, diferenças de preço entre um ativo e um *hedge* estatístico ótimo. Ao testar e concluir que uma dessas séries temporais é estacionária, abrimos a porta para explorar esse fato, ou seja, tentar lucrar com os desvios temporários da estacionariedade. A metodologia para explorar essas oportunidades chamamos de "Arbitragem Estatística Implícita".

A idéia é explorar o componente de reversão à média das dinâmicas dos *"mispricings"*, sem a necessidade de construir um modelo de previsão. O nome da estratégia vem justamente do fato de ela depender *implicitamente* dessa reversão. Assim, num contexto de cointegração, vamos explorar uma espécie de um modelo de correção de erros, como aquele que descrevemos no item 2.3.7. As estratégias serão lucrativas a medida em que esse efeito de correção de erros for suficientemente forte para superar os custos de transação de implementar a posição.

A estratégia de AEI vai ser implementada por regras paramétricas de trading, que definem o sinal e a magnitude da posição a ser tomada. Vamos discutir detalhadamente várias regras de *trading* mais adiante no item 4.5.2, mas por enquanto vamos nos reter às regras mais simples, para facilitar a compreensão do modelo e seus resultados. Assim, para um dado *"mispricing*", o modelo vai determinar o portfólio desejado, ou seja, o conjunto de ativos $\{T, C_1, \ldots, C_4\}$ nas proporções $\{1, -\beta_1, \ldots, -\beta_4\}$ respectivamente. A estratégia básica define a posição desejada de acordo com a regra[22]:

$$AEI(M_t, k) = -sign\left(M_{t-j}\right) \cdot \left|M_{t-j}\right|^k \tag{4.9}$$

onde o sinal negativo indica que o "*mispricing*" deve ser comprado quando for negativo (ou seja, quando o ativo-objeto valer menos que o ativo-sintético, compramos o ativo-objeto e vendemos o ativo-sintético), e vendido quando for positivo. O parâmetro *k* permite que a magnitude da posição varie de acordo com o desvio do "*mispricing*" em relação a seu equilíbrio.

[22] Burgess (2001), pp. 177-178.

Se tivermos *k=0*, temos uma função degrau, ou seja, a posição inteira é investida no portfólio (seja comprado ou vendido), apenas dependendo se o valor é positivo ou negativo. Com *k>0*, o tamanho do portfólio aumenta à medida que o *"mispricing*" vai se desviando de sua média, e diminui conforme ele retorna. Mais ainda vamos ilustrar o efeito da variação desse parâmetro nos retornos obtidos.

A partir dessa regra de *trading*, o retorno em um período entre *t* e *t+1* é dado por:

$$AEIRET(M_t, T_t, SA(T)_t, k)_t = AEI(M_t, k) \cdot \frac{\Delta M_t}{(T_t + SA(T)_t)} - c \cdot |\Delta AEI(M_t, k)| \quad (4.10)$$

ou seja, a posição atual do portfólio, multiplicada pela variação do valor desse portfólio, dividida pelo valor total das posições (e não o valor líquido), e ajustada pelos custos de transação incorridos devido a mudanças no sinal de *trading*. O primeiro termo no lado direito da equação corresponde à mudança proporcional do *"mispricing*" em relação ao valor total das posições de todos os ativos desse portfólio (a soma absoluta da posição no ativo-objeto e nos vários componentes do ativo-sintético). Já o custo de transação é aproximado por uma proporção *c* da mudança de posição realizada.

Os lucros acumulados em um período em que a estratégia é utilizada são dados por:

$$AEIPROF(M_t, T_t, SA(T)_t, k) = \sum_{t=1}^{n} AEIRET(M_t, T_t, SA(T)_t, k)_t \quad (4.11)$$

e note-se que não estamos compondo os lucros, estamos considerando que sempre trabalhando com uma base de capital fixa, onde os lucros não fazem com que as posições aumentem.

Uma vez definida a regra de *trading* e a forma de calcular os retornos dessa regra, o modelo foi implementado em *MATLAB*, com os detalhes do código estão apresentados no Apêndice A2, e assim podemos passar a experimentação. Vamos separar o conjunto de dados em dois: (i) dentro da amostra, onde vamos estimar os ativos-sintéticos, calcular as séries temporais dos *"mispricings*", e determinar se são ou não cointegrados, tudo de acordo com a nossa metodologia *"stepwise*" descrita nos item 4.3.1 e 4.3.2; e (ii) fora da amostra, onde vamos simular a aplicação da regra de *trading* e determinar se ela é capaz de explorar um componente de reversão à médias das dinâmicas dos "*mispricings*". O período dentro da amostra será entre 02 de janeiro de 2004 até 29 de dezembro de 2005, para um total de 498 observações, enquanto o período fora da amostra começa em 02 de janeiro de 2006 e vai até 28 de dezembro de 2006, para um total de 246 dados.

Estimando os modelos para as nossas 55 ações, com 498 dados em cada uma, obtemos 29 "*mispricings*" cointegrados. Para cada um desses vinte e nova vamos calcular os retornos obtidos através da estratégia de Arbitragem Estatística Implícita. Na Figura 4.5 podemos ver o retorno obtido pela estratégia no período de 246 dias úteis, para o *"mispricing*" de VALE5, usando *k=1*, ou seja posição linearmente dependente do tamanho do desvio, e custos de

transação de 0,25% (no item 4.5.3 temos uma discussão mais aprofundada sobre custos de transação e *slippage*, mas esta é uma boa aproximação dos custos incorridos).

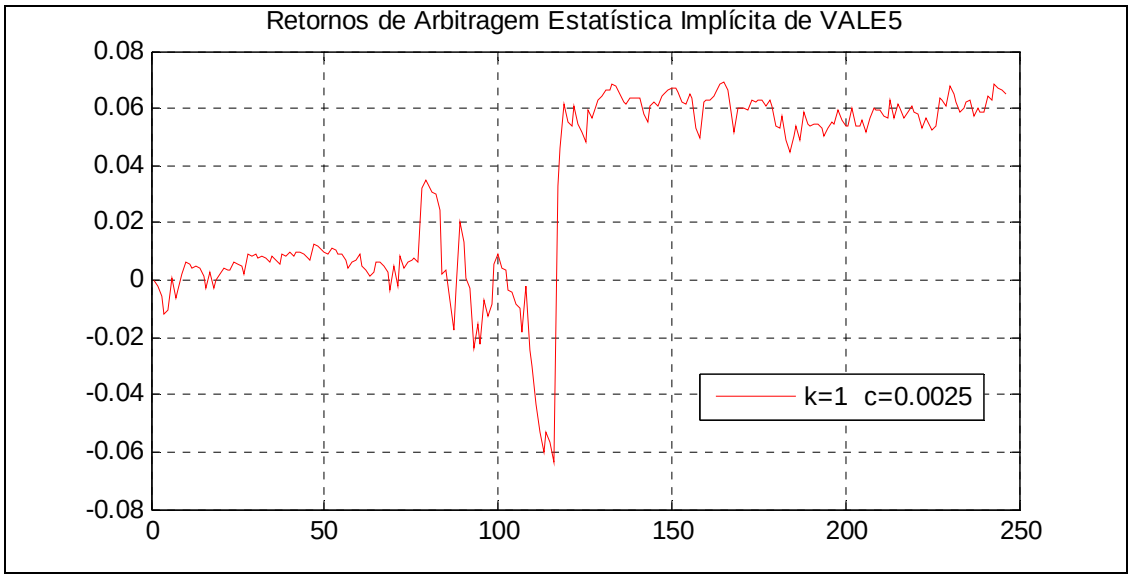

Figura 4.5: Retornos de arbitragem estatística implícita de VALE5

O retorno final de 6,49%, com volatilidade no período de 3,18% e índice de Sharpe de 2,04, se mostra interessante, pois foi obtido por uma estratégia sem nenhuma alavancagem, completamente neutra a direção geral do mercado. Contudo apenas um "*mispricing*" não conta toda a história. Temos que analisar a performance de todos os modelos estimados. A Tabela 4.9 mostra as estatísticas dos retornos da estratégia de arbitragem estatística implícita aplicada a cada um dos 29 conjuntos de ativos que identificamos como cointegrados. A Figura 4.6 mostra a evolução de um portfólio composto pela média dos retornos de todas as estratégias, uma carteira com montantes iguais em cada um dos 29 modelos.

qtde	média	mediana	mínimo	mispricing mínimo	máximo	mispricing máximo	retornos positivos	retornos negativos
29	-11,10%	9,18%	-2541,6%	SBSP3	1248,8%	TNLP3	22	7

Tabela 4.9: Resumo das estratégias de Arbitragem Estatística Implícita

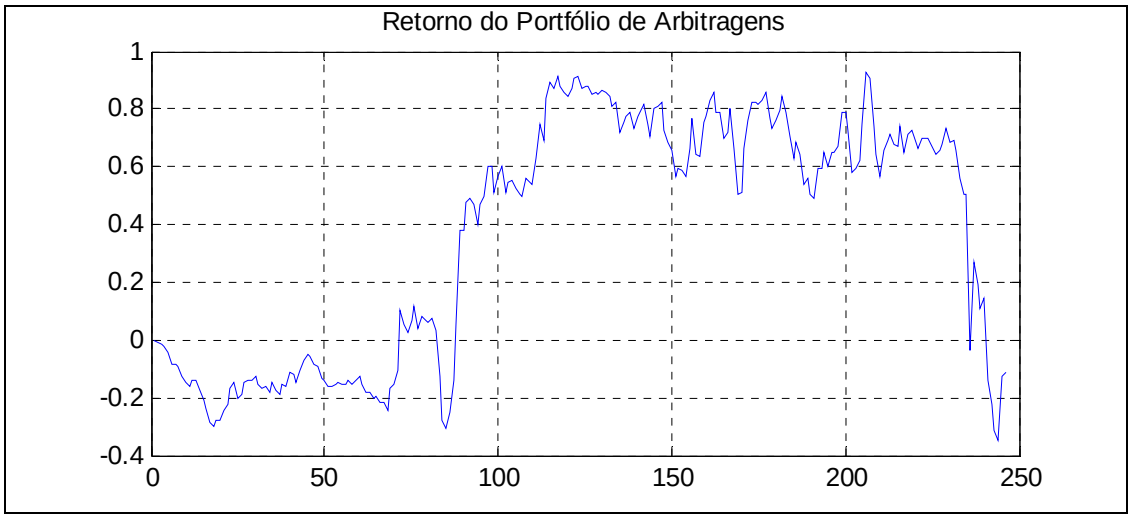

Figura 4.6: Retornos do portfólio de 29 modelos de arbitragem estatística implícita

Como visto na Tabela 4.9 esse portfólio tem retorno -11,10%, com volatilidade de 41,71%, que é bastante alta. Idealmente, teríamos 29 modelos que gerassem um portfólio com bom retorno médio, e que diversificassem o risco, reduzindo a volatilidade e conseqüentemente aumentando o índice de Sharpe. É justamente esse resultado encontrado por Burgess (2001)[23], embora o retorno obtido ali (média de 8%), tenha sido considerado pelo autor "pouco impressionante, principalmente quando ajustado ao risco". Ainda assim, o resultado obtido ao fim é interessante, pois mostra que a estratégia de arbitragem estatística implícita consegue gerar bons resultados na maioria das vezes, embora alguns casos se mostram bastante problemáticos, o modelo parece não funcionar, e acabam arruinando os retornos de todos os outros. Por exemplo, se retirarmos os 7 modelos que geram retornos negativos, o retorno médio passa a ser 111,26%, e o mediano passa a ser 16,03%.

Vamos estudar brevemente os dois casos extremos da nossa amostra para tentar entender o que aconteceu, e como poderíamos melhorar a performance geral do modelo.

O primeiro caso extremo é o modelo de arbitragem estatística implícita de SBSP3. Na Figura 4.7 temos a série temporal do "*mispricing*" de SBSP3.

[23] Burgess (2001), pp. 183, Tabela 7.3.

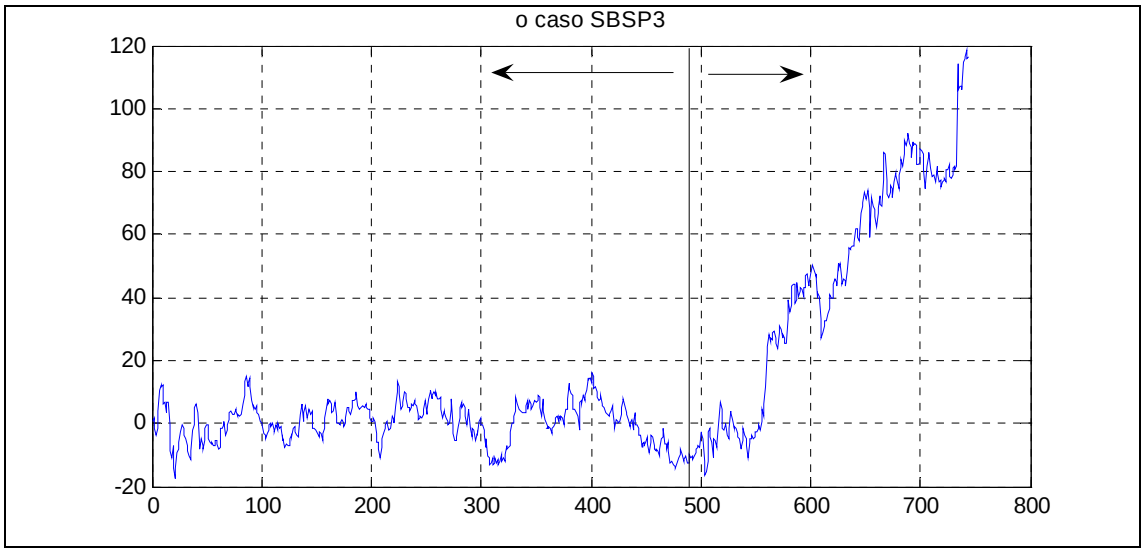

Figura 4.7: Série Temporal do "*Mispricing*" de SBSP3

Note como claramente a série sofre uma quebra estrutural pouco depois do ponto onde fizemos a estimação do modelo (em datas isso se dá em meados de março de 2006). Essa quebra estrutural é o tipo de evento a que estratégias de arbitragem estatística estão sujeitas, e ela ocorreu devido a uma possível mudança tributária que impactaria fortemente os lucros de empresas de saneamento, como é o caso da Sabesp[24]. Para tratar problemas desse tipo, há várias possibilidades. Uma delas é o uso de modelos do tipo "*Markov switching*", como o descrito em Roche e Rockinger (2003), que entendem os dados como sendo parte de várias distribuições gaussianas alternativas, sorteadas de maneira markoviana. Outra possibilidade é adicionar na metodologia de estimação uma técnica para detecção robusta de quebras, usando modelos *REGARIMA*, como os descritos em Zivot & Wang (2003)[25]. Uma terceira alternativa é o uso de modelos *"Hidden Markov*" (normalmente conhecidos como *HMM*), descritos em Carmona (2004)[26], que são uma versão não-linear (talvez mais adequada para séries temporais financeiras) dos modelos *"Markov switching*" já mencionados. Por fim, duas outras alternativas também são possíveis: (i) o conhecimento dos fundamentos da empresa permitiria evitar esse tipo de evento, mas requer analistas especializados e não pode ser sistematizado, e (ii) o uso de regras de *stop-loss* nos permitira evitar que as perdas fossem amplificadas de maneira explosiva como nesse caso. Note que a definição do nosso modelo de arbitragem faz com que quanto mais o *"mispricing*" se desloca para longe do equilíbrio, mais nossa aposta na reversão à média aumenta, e portanto, se o modelo não volta ao equilíbrio, temos um resultado catastrófico. A

[24] Vide jornal Valor Econômico de 13 de dezembro de 2006, "Lei do Saneamento vai à sanção presidencial", no site: http://www.valoronline.com.br/valoreconomico/285/primeirocaderno/politica/Lei+do+Saneamento++vai+a+sancao+presidencial,,,60,4051606.html

[25] Zivot & Wang (2003), cap. 17, pp. 609-625, principalmente pp. 619-623.

[26] Carmona (2004), pp. 391-403.

grande dúvida é: o uso de *stop-losses* evitaria que realizássemos lucros em modelos que eventualmente retornam à média, ou o fato de nos proteger de catástrofes é mais importante? Mais adiante vamos colocar essa questão em prática.

O segundo caso extremo é o de lucro máximo, que envolve o modelo de arbitragem de TNLP3. Na Figura 4.8 temos a série temporal, onde vemos que o modelo sofre três quebras estruturais, que por reverterem à média permitem a obtenção de lucros extraordinários.

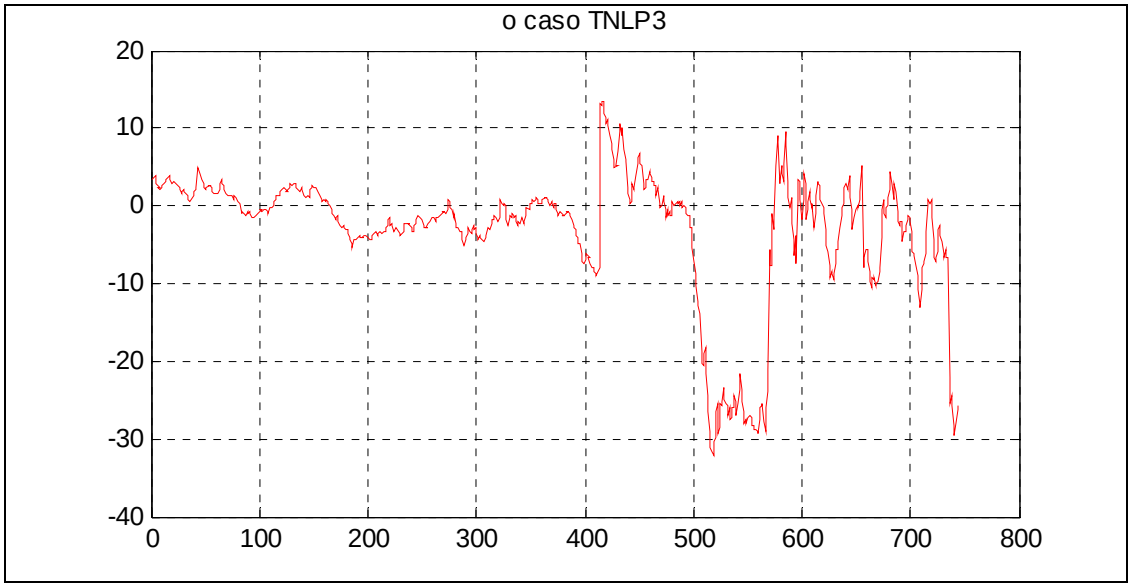

Figura 4.8: Série Temporal do "*Mispricing*" de TNLP3

No caso de TNLP3, as quebras estão relacionadas a um processo de reorganização societária da empresa, que beneficiaria os acionistas detentores de ações ordinárias da empresa (ou seja, a TNLP3) em detrimento dos donos de ações preferenciais (a TNLP4). Esse processo foi amplamente acompanhado pela mídia, e suas reviravoltas foram sendo refletidas nos preços das ações da empresa[27]. Nesse caso, embora o resultado final do modelo tenha sido bastante positivo, novamente o risco de um evento foi bastante relevante, e aqui, mais até do que no caso de SBSP3 que analisamos anteriormente, o conhecimento dos fundamentos do caso da empresa seriam o principal fator que permitiria a tomada de decisões adequadas sobre o que fazer com o modelo de arbitragem.

O resultado final dos modelos devidamente analisado, vamos nos deter brevemente na influência de dois parâmetros na lucratividade final. O primeiro é o parâmetro *k*, que calibra o tamanho da posição a ser tomada pelo modelo. E o segundo é o custo de transação *c*. A Figura 4.9 mostra a influência de diferentes parâmetros *k* no retorno final obtido pela estratégia de

[27] Vide jornal Valor Econômico de 15 de dezembro de 2006, "Acionista rejeita proposta de reestruturação da Telemar; ações caem", que detalha o desfecho do processo de reorganização societária da Telemar, no site: http://www.valoronline.com.br/valoronline/Geral/empresas/16/Acionista+rejeita+proposta+de+reestruturacao+da+Telemar+acoes+caem,,,16,4057942.html?highlight=&newsid=4057942&areaid=16&editionid=1681

arbitragem estatística implícita de VALE5 (com *c=0,25%*), e a Figura 4.10 mostra como o nível de custos *c* afeta os retornos (para um *k=1*) do mesmo "*mispricing*", que já vimos na Figura 4.5.

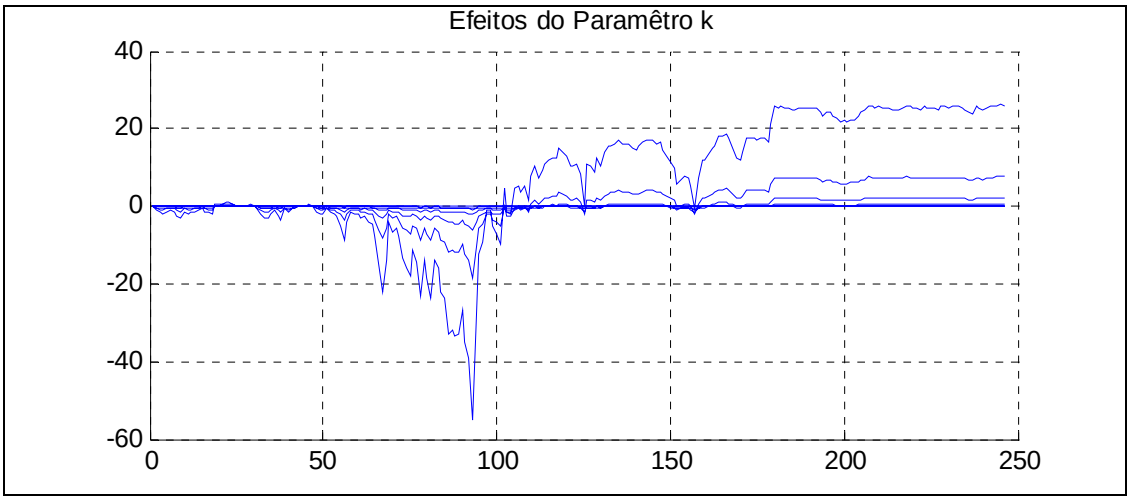

Figura 4.9: Efeitos da variação do parâmetro *k*

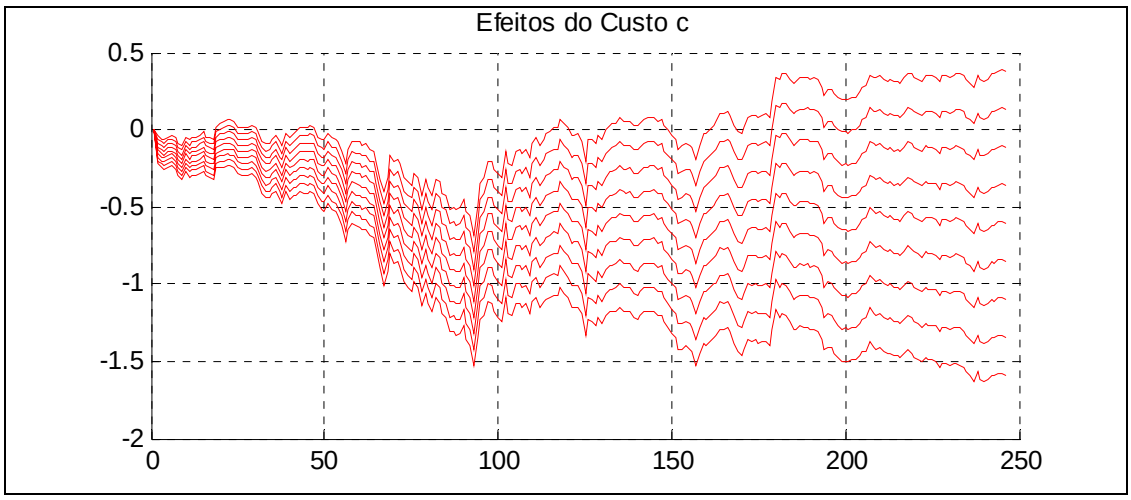

Figura 4.10: Efeitos da variação do custo *c*

Note-se como o aumento do parâmetro *k* alavanca os retornos, mas também piora muito a volatilidade do modelo, aumentando o risco de perdas catastróficas. E o custo *c* é uma variável extremamente relevante, pois pode fazer a diferença entre modelos de arbitragem estatística lucrativos e modelos que perdem dinheiro sistematicamente. A questão do custo de transação será discutida mais adiante no item 4.5.3.

Em resumo, pudemos observar até agora nossa metodologia de construção em ação, e ver como embora a maioria dos modelos gere bons lucros com baixo risco, há enormes possibilidades de melhorar. A seguir vamos discutir as técnicas de previsão com redes neurais que nos permitirão desenvolver modelos para prever o comportamento futuros dos "*mispricings*", de maneira a aumentar significativamente a performance da estratégia de arbitragem estatística.

4.4. Previsão das Dinâmicas dos Erros

4.4.1. Introdução

A construção de modelos de previsão de preços de ativos é um dos problemas mais complexos em finanças. Contudo, ao formular o problema de uma maneira apropriada, é possível reduzir a variância envolvida e assim conseguir melhores resultados, ou seja, obter modelos que prevêem a dinâmica futura e permitem explorar essas previsões como base de operações no mercado. Nossa proposta envolve a criação de ativos-sintéticos, através da metodologia construtiva descrita no item 4.3, que permitam explorar oportunidades de arbitragem estatística. Ao criar as séries temporais dos *"mispricings"*, o que fizemos foi, em essência, criar séries onde a variância fosse reduzida (vide Tabela 4.6 e Tabela 4.7) e portanto a tarefa de previsão fosse facilitada.

Problemas de previsão em finanças envolvem muito ruído, baixa quantidade de informação sobre relações passadas, pequenas amostras de dados e potencial não-estacionariedade. Ou seja, praticamente um pesadelo para o modelador. Nossa proposta envolve o uso de redes neurais artificiais como ferramenta para superar algumas dessas dificuldades e obter bons resultados. Essa ferramenta, onde de uma certa maneira deixamos "os dados falarem por si mesmos", ou seja, há uma estimação não-paramétrica, depende de algumas condições fundamentais para que funcione bem. Uma dessas condições fundamentais é que os *inputs* de treinamento sejam representativos. Discutimos essa questão anteriormente no item 4.2.2, enfatizando como é importante que os *inputs* representem bem as várias relações possíveis entre as diversas variáveis. Isso porque a rede neural vai, após o treinamento, absorver esse conhecimento em sua matriz de pesos. Agora, por mais poderosa que seja, ela não pode prever situações que ela nunca observou. Ainda assim, temos que tomar cuidado para não cair nos problemas de "*overfitting"* ou "*data snooping"*[28], ou seja, a rede não pode aprender os dados tão bem que sua capacidade de generalizar para o futuro fique comprometida. A preocupação é que a rede capture os sinais e não os ruídos, na linha do que apresentamos no item 3.1.6.1.

Nos itens que seguem, vamos descrever várias etapas da construção de uma boa rede neural para previsão do comportamento futuro das séries temporais dos *"mispricings"*. Boa parte das técnicas descritas segue Parreiras (2003). Vamos iniciar discutindo o pré-processamento dos dados, para facilitar o trabalho de treinamento da rede. Depois vamos discutir as características das redes que vamos usar, como algoritmo de treinamento, número de neurônios na camada escondida, e técnicas de parada antecipada. A seguir, vamos discutir o pós-processamento e explorar algumas previsões realizadas pelas redes em modelos de

[28] Vide Shadbolt & Taylor (Eds.) (2003), pp. 55-59.

arbitragem estatística. Como fizemos no item 4.3, vamos encerrar com uma aplicação prática da nossa metodologia, explorando os resultados obtidos através de modelos de "arbitragem estatística condicional", ou seja, condicionados às previsões.

4.4.2. Pré e Pós-Processamento

Antes de realizar a entrada dos dados na rede neural, é necessário realizar alguns pré-processamentos, de modo a obter uma boa performance preditiva do modelo (vide discussão no item 3.1.6.5, além de Demuth e Beale (2001) ou Refenes (Ed.) (1995), que são boas referências no assunto). Cada vez que uma rede neural for criada, todas as etapas de pré-processamento serão repetidas. São dois procedimentos básicos:

- *Normalização*: o processo de consiste na normalização de todos os dados de entrada (*inputs*) de modo a terem média zero e variância unitária. Isto é necessário devido a uma peculiaridade das redes neurais: elas não trabalham bem com valores elevados – já que as funções de transferência (a função sigmóide, por exemplo) realizam o chamado "*squashing"*, ou seja, comprimem os dados (Haykin, 2001). Assim, a normalização dos dados facilita o trabalho de cálculos das matrizes de pesos e viés, proporcionando melhor performance. Após o treinamento, todas as saídas da rede têm de ser renormalizadas, ou seja, colocadas na média e variância originais.
- *Análise de Componentes Principais (PCA)*: em várias situações (caso deste trabalho), a dimensão do vetor de entrada é razoavelmente grande, e há alguma correlação entre os componentes do vetor (ou seja, redundância) (Demuth e Beale, 2001). Assim, é útil nestas situações reduzir o tamanho desse vetor, e uma maneira eficiente de realizar este processo é através da Análise de Componentes Principais (ou "*Principal Component Analysis"*, PCA). Esta técnica traz três efeitos: ortogonaliza os componentes do vetor de entrada (de maneira a serem não-correlacionados), ordena os componentes ortogonais resultantes (componentes principais) de maneira que aqueles com maior variação venham antes, e elimina aqueles que contribuem menos para a variância total do conjunto. Neste trabalho, serão eliminados os componentes que explicam menos de 1% da variação total do conjunto de *inputs* (não se deseja reduzir excessivamente o conjunto de dados de entrada, por isso o limite baixo). Após o treinamento da rede, todos os novos dados que são introduzidos devem ser transformados usando a matriz de componentes principais, de maneira a garantir homogeneidade de resultados.

O uso das técnicas de normalização e análise de componentes principais auxilia no processo de obtenção de boa performance preditiva das redes a serem construídas, em linha com as melhores práticas de construção de modelos neurais, que delineamos no Capítulo 3.

Do mesmo modo que devemos pré-processar as entradas, após o treinamento se faz necessário analisar os resultados. Uma vez renormalizados para média e variância apropriados,

será realizada uma Análise de Regressão entre os dados simulados pela rede e os *targets*, ou seja, vão ser comparados os "*mispricings"* previstos pelo modelo de rede neural com os dados preços reais observados no mercado, de maneira a aferir a performance das redes testadas. O resultado desta análise, em termos de coeficiente de correlação, será de fundamental importância na avaliação da performance de cada um dos modelos de rede neural utilizados daqui por diante.

4.4.3. Características das Redes Neurais

Os dados de entrada devidamente prontos e formatados, vamos voltar nossas atenções para as características das redes neurais com que vamos trabalhar. A adequada definição destes parâmetros permitirá a obtenção de modelos parcimoniosos e de boa performance futura, ou seja, com boa capacidade de generalização.

O primeiro passo é a escolha do algoritmo de treinamento. No Capítulo 3, item 3.1.6.2, discutimos esse tema, introduzindo vários algoritmos e suas características. A dúvida é escolher o que melhor se adapta ao nosso problema. A partir dos resultados de Parreiras (2003), onde o autor testou extensivamente vários algoritmos em um contexto de previsão de preços de ações, a opção vai recair pelo algoritmo Levenberg-Marquardt. Portanto, vamos trabalhar com redes *feedforward*, algoritmo de treinamento *backpropagation* com otimização Levenberg-Marquardt[29]. Em Parreiras (2003), essa configuração de rede obteve os melhores resultados em termos de performance preditiva, quando cotejada com outros algoritmos.

Uma vez definido o algoritmo, temos de definir o número de neurônios na camada escondida da rede. O número de nós na camada de entrada é definido pelo número de *inputs*, o que está dado pelos dados pré-processados. A camada de saída tem um só nó, justamente o que fornece o resultado, a previsão adiante do valor do "*mispricing*". Então a camada escondida é justamente onde nossa decisão é fundamental: usar poucos neurônios pode fazer com que a rede não aprenda padrões suficientes, enquanto neurônios demais podem levar ao problema do *"overfitting"* que já mencionamos anteriormente.

Como a literatura traz uma variedade enorme de diferentes arquiteturas possíveis, e nenhum trabalho que analisou sistematicamente esse ponto em aplicações práticas de finanças, decidimos optar pela solução mais simples: vamos testar várias configurações diferentes e optar por aquela que gera melhor performance, num conjunto amplo de dados. Para isso, vamos implementar uma rotina que treine várias redes com diferentes números de neurônios na camada escondida, e ao fim vamos comparar os números de performance para escolher o melhor. Em Parreiras (2003) um procedimento similar foi utilizado, mas naquele trabalho se optou por usar

[29] Vide Demuth e Beale (2001), pp. 5.31–5.34.

uma rede específica para cada ação cujo preço se desejava prever. Aqui vamos escolher um caminho mais simples e usar uma só configuração de rede, com apenas um número determinado de neurônios na camada escondida. Acreditamos que as redes neurais são suficientemente robustas, e as séries temporais que vamos usar (os "*mispricings"*) são suficientemente semelhantes, de tal modo que a opção única não seja um problema.

Nosso experimento tem a seguinte forma: vamos rodar nossa metodologia de construção de "*mispricings"* com a base de dados completa, ou seja, todos os 744 dados para cada uma das 55 ações diferentes, e vamos escolher aqueles ativos-sintéticos que cointegram com seus ativos-base. Ao fim, vamos treinar, para cada uma das séries temporais de *"mispricing"*, várias redes neurais, com o número de neurônios na camada variando entre 2 e 15. Ao fim vamos comparar a performance de previsão de cada uma dessas 14 configurações, para um grupo de 50 dados que serão mantidos fora da amostra de treinamento, e determinar qual obtém o melhor resultado. Os detalhes do código estão dados no Apêndice A3.

O parâmetro de performance utilizado para avaliar a qualidade da rede é o R^2 de uma regressão entre as previsões e os dados realizados, para o conjunto de 50 dados que definimos como fora da amostra. Essa escolha é baseada em Burgess (2000), que explica[30] a importância de usar a variância explicada como métrica de performance. Um exemplo desta regressão pode ser visto na Figura 4.11, onde mostramos o resultado das previsões de uma rede com 8 neurônios na camada escondida, treinada para prever o "*mispricing"* de PETR4, cuja construção discutimos no item 4.3.1.

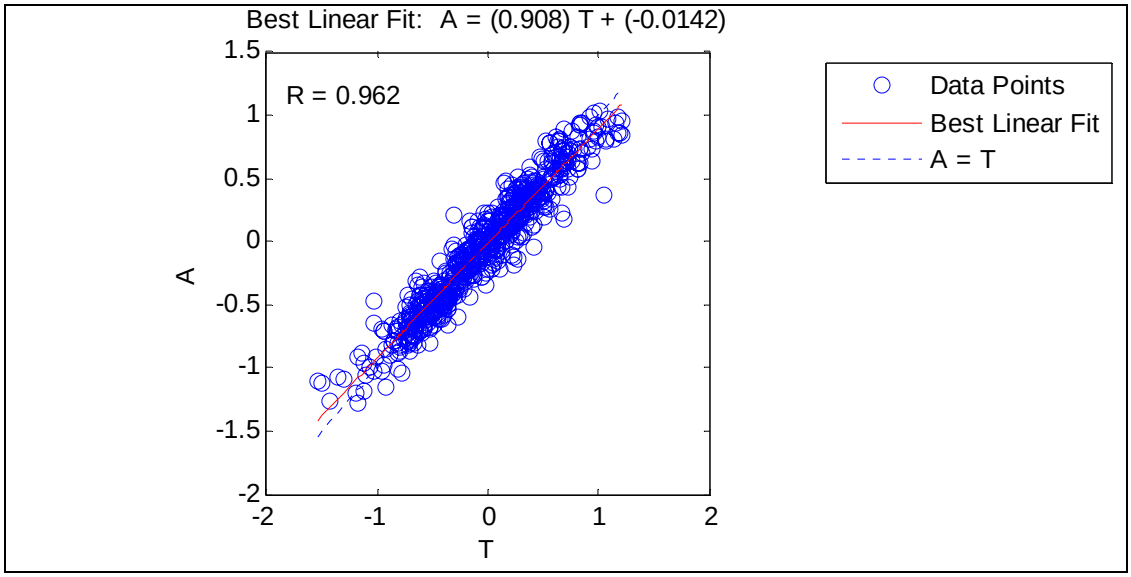

Figura 4.11: Regressão entre previsões (A) e realização para o "*mispricing*" de PETR4

[30] Burgess (2000), pp. 268-283.

Ao rodar os modelos, obtivemos 30 ações cujos *"mispricings*" são cointegrados. Assim, temos que treinar 14 redes para cada uma dessas ações, guardar o resultado da sua performance, e ao fim determinar qual número de neurônios na camada escondida gera a melhor performance preditiva. Na Figura 4.12 mostramos o resultado obtido, como uma figura de três dimensões.

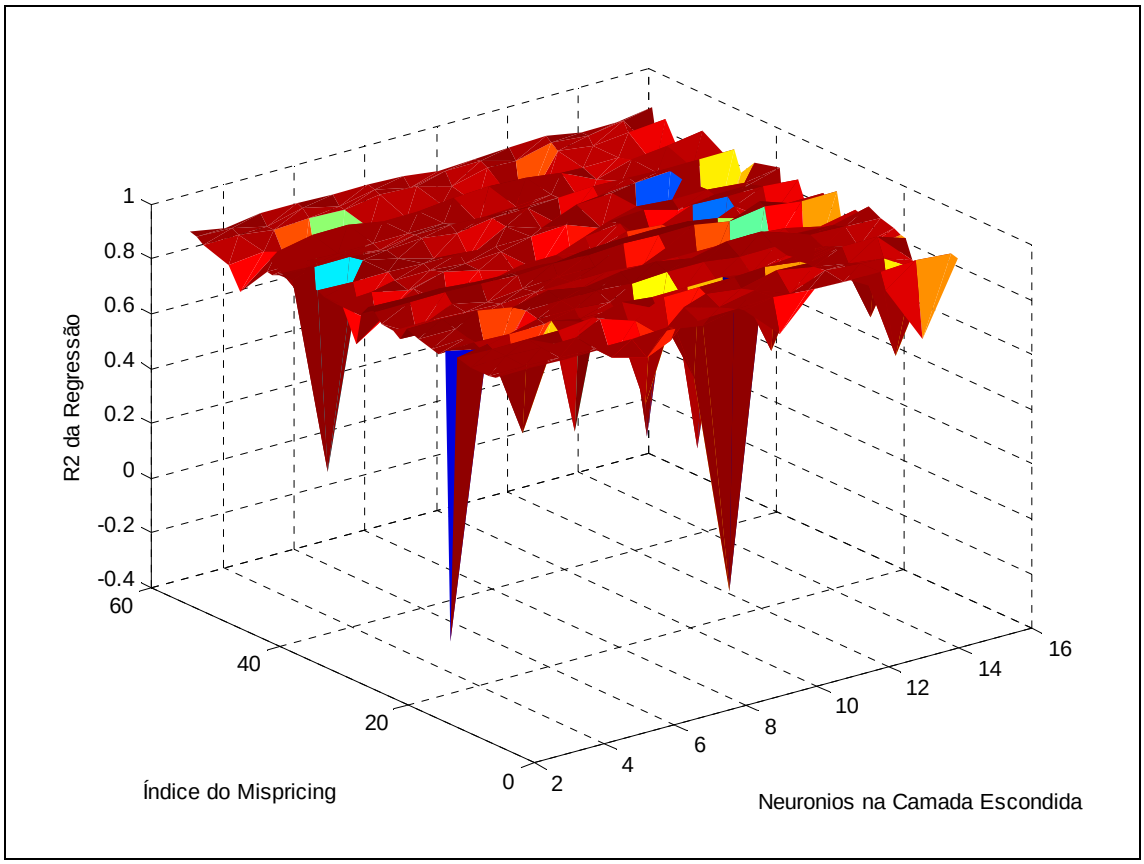

Figura 4.12: Resultado do experimento com número de neurônios na camada escondida

Da Figura 4.12 é relativamente difícil inferir o melhor número de neurônios. Para obtermos esse número, e passarmos adiante no processo de modelagem, rodamos 50 repetições do experimento, guardando em cada uma dela o número de neurônios de cada "*mispricing*" que gerava a melhor performance preditiva, e ao fim fizemos uma média de todos os resultados[31]. O resultado obtido foi: a configuração de rede neural que performa melhor, em média, é aquela com 8 neurônios na camada escondida[32]. Na Figura 4.13 temos um histograma dos resultados desse experimento, mostrando como cada uma das vezes o resultado podia ser ligeiramente diferente. O fato de o algoritmo de treinamento buscar uma solução quasi-ótima no espaço de pesos de uma rede neural é o responsável aqui – nem sempre a solução obtida será a mesma.

[31] Considerando que 30 *"mispricings"* são cointegrados, temos um total de 30 x 14 x 50 = 21.000 redes neurais sendo treinadas nesse experimento.

[32] O resultado do experimento foi 8.43 neurônios, em média. Arredondamos para 8.

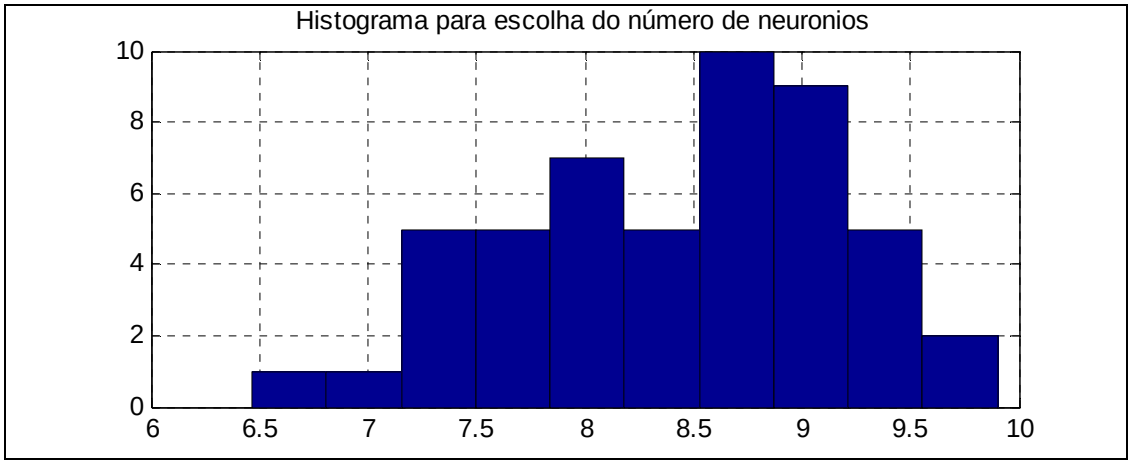

Figura 4.13: Histograma do experimento para escolha dos neurônios

Uma vez definido o número de neurônios da camada escondida, é importante definir um método para melhorar a capacidade de generalização dos nossos modelos. Nos itens 3.1.6.3 e 3.1.6.4 discutimos várias técnicas de treinamento e de construção de redes. Nossa opção recairá sobre uma técnica semelhante à utilizada por Parreiras (2003), chamada de "*early stopping*" ou *parada antecipada*[33]. Essa técnica consiste em dividir o conjunto de dados de treinamento em três partes: treinamento, validação e teste. De acordo com Demuth e Beale (2001), "o primeiro subconjunto é o de treinamento, que é usado para computar os gradientes e atualizar os pesos e vieses da rede, enquanto o segundo subconjunto, de validação, tem seu erro monitorado". Normalmente o erro no subconjunto de validação decresce nas primeiras etapas de treinamento, assim como o erro no subconjunto de treinamento. Contudo, em etapas posteriores, se a rede começa a se adaptar demais aos dados (o já citado "*overfitting*"), o erro de validação tipicamente começa a crescer, e se isso ocorrer por um número definido de iterações, o treinamento é parado e os pesos da rede são definidos como aqueles em que o erro foi mínimo. Em nossos modelos, tipicamente vamos utilizar 50 dados no subconjunto de validação. A Figura 4.14 mostra o exemplo de uma sessão de treinamento de uma rede neural com algoritmo Levenberg-Marquardt, 8 neurônios na camada escondida, parada antecipada com 643 dados para treinamento, 50 dados de validação e mais 50 dados no conjunto de teste. As redes foram treinadas para prever o "*mispricing*" de PETR4.

[33] Vide Demuth e Beale (2001), pp. 5.41–5.43.

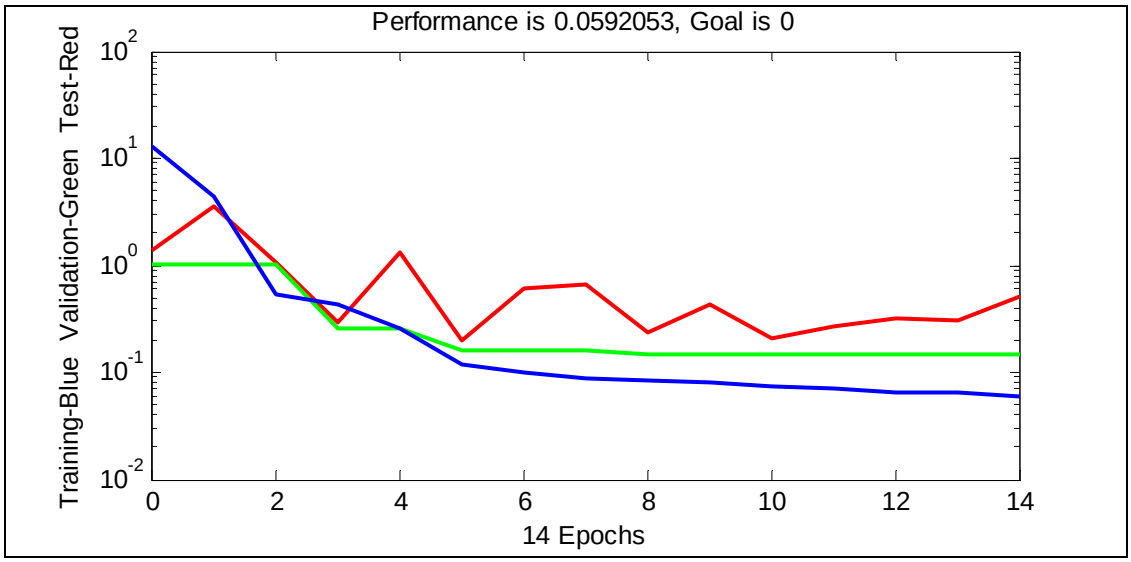

Figura 4.14: Exemplo de uma sessão de treinamento neural

Note como o erro na curva azul (treinamento) e na verde (validação) decrescem conjuntamente nas primeiras instâncias de treinamento (chamadas de *épocas*, vide item 3.1.6.3). A partir de um certo momento o erro de treinamento continua a cair, enquanto no conjunto de validação o erro fica estabilizado. A curva vermelha denota o erro no conjunto de teste – embora seu comportamento seja mais errático, o erro também decresce, mostrando alguma capacidade de generalização da rede.

Já discutimos vários aspectos do projeto de nossas redes neurais: algoritmo de treinamento, número de neurônios na camada escondida, técnica de parada antecipada. Faltam alguns detalhes menos importantes, que vamos definir a seguir. O número de épocas de treinamento máximo vai ser definido como 100, embora o algoritmo de parada antecipada normalmente pare o processo entre 10 e 20 épocas depois de iniciado. O parâmetro de erro de treinamento vai ser o chamado *MSE* ("*Mean Squared Error*" ou erro quadrático médio), que como vimos no item 3.2, a partir de Zhang et al. (1998) e Adya e Collopy (1998) é o mais comum na literatura. Note-se que para treinar as redes vamos usar o *MSE*, mas para aferir performance fora da amostra usamos o R^2, como discutimos anteriormente.

Concluindo, mostramos na Tabela 4.10 as principais características construtivas da rede neural. Os detalhes do algoritmo de treinamento das redes neurais estão dados no Apêndice A4.

Parâmetro	Valor
Plataforma de software	*MATLAB* com *Neural Network Toolbox*
Arquitetura	*Feedforward*
Número de Camadas	3 (1 camada escondida)
Número de *Inputs*	7 (vide Tabela 4.3)
Número de *Outputs*	1 (valor do *mispricing*)
Algoritmo de Treinamento	*Backpropagation* com otimização por algoritmo Levenberg-Marquardt
Função de Transferência	Tangente Hiperbólica (camada escondida) Linear (camada de saída)
Neurônios na Camada Escondida	8
Técnica de Generalização	Parada Antecipada (*early stopping*)
Parâmetro de Performance	*MSE* (erro quadrático médio)
Épocas de Treinamento	até 100

Tabela 4.10: Características dos Modelos Neurais

4.4.4. Arbitragem Estatística Condicional

Uma vez definidas as características construtivas das redes neurais que vamos utilizar para realizar previsões futuras das dinâmicas dos "*mispricings*", temos que partir para a experimentação. O escopo aqui é o mesmo do item 4.3.3, onde exploramos os modelos de arbitragem estatística implícita. Ali, nosso interesse era examinar a lucratividade que poderia ser obtida a partir da característica de reversão à média das dinâmicas. Aqui, vamos trabalhar com arbitragens condicionais, ou seja, que dependem das previsões realizadas. Como vimos nos casos de SBSP3 e TNLP3 do item 4.3.3, nem sempre as dinâmicas mantém o comportamento observado no período dentro da amostra. Assim, se conseguimos prever o comportamento dos "*mispricings*" podemos explorar não só a reversão à média como outros fenômenos, como *momentum* e correlação com o mercado.

A regra de *trading* para arbitragem estatística condicional é bastante semelhante àquela que utilizamos no item 4.3.3 para as estratégias implícitas. Vamos formular a seguinte função:

$$AEC\left(E\left[\Delta M_t\right],k\right)_t = sign\left(E\left[\Delta M_t\right]\right)\cdot\left|E\left[\Delta M_t\right]\right|^k \tag{4.12}$$

Da mesma maneira que para a regra da estratégia de arbitragem implícita, esta função define a posição que vamos tomar em relação ao portfólio de arbitragem, a partir de uma expectativa de variação, dada por $E\left[\Delta M_t\right]$. A diferença com a regra que vimos anteriormente está no fato de que o relacionamento negativo entre o nível do "*mispricing*" e a sua variação futura não é mais necessário. Aqui, a previsão pela rede neural faz esse papel – ela determina se o "*mispricing*" vai reverter à média, ou se vai continuar se movendo em uma direção divergente. Temos portanto:

$$E\left[\Delta M_t\right] = f\left(M_t, Z_t\right) \tag{4.13}$$

onde Z_t representa as variáveis exógenas que determinamos anteriormente no item 4.2.2.. Dado que vamos trabalhar com uma função de *trading* semelhante àquela que utilizamos para as

estratégias de arbitragem estatística implícita, o cálculo dos retornos também é bastante semelhante:

$$AECRET_{t+1} = AEC\left(E\left[\Delta M_t\right], k\right)_t \cdot \frac{\Delta M_{t+1}}{\left(T_t + SA(T)_t\right)} - c \cdot \left|\Delta AEC\left(E\left[\Delta M_t\right], k\right)_t\right| \quad (4.14)$$

e novamente o retorno acumulado vai ser a soma dos retornos ao longo do tempo.

Há dois detalhes importantes para o nosso processo de modelagem. O primeiro é o uso de uma janela deslizante de treinamento da rede neural. Essa janela funciona da seguinte maneira: como fizemos anteriormente, vamos estimar os *"mispricings*" usando uma amostra de 498 dados. A partir daí vamos treinar uma rede neural com 498 valores de cada uma das variáveis, e pedir pra ela estimar o valor do "*mispricing*" um dia adiante. A partir daí tomamos a decisão de *trading*, aferimos os retornos obtidos, e movemos a janela. A próxima rede que vamos treinar vai usar 499 dados *inputs*, e estimar um dia adiante, e assim sucessivamente. Assim, ao final do período, estaremos treinando as redes com praticamente todos os dados disponíveis. Essa técnica busca maximizar a quantidade de informação que mostramos para as redes neurais, de modo a obter as melhores previsões possíveis.

O segundo detalhe se refere ao caráter estocástico das redes neurais: quando vamos começar o treinamento de uma rede, os pesos e viés são inicializados randomicamente, e o algoritmo de treinamento faz uma busca no espaço de pesos para minimizar os erros. Assim, a solução obtida é, via de regra, quasi-ótima. Por isso, para cada previsão que devemos fazer, vamos treinar **5 redes neurais** com os mesmos *inputs*, e ao fim vamos usar como previsão a média das previsões. Embora aumente o custo computacional, essa técnica busca minimizar a possibilidade de previsões catastróficas, que poderiam levar a perdas na nossa simulação. De novo, os modelos foram implementados em *MATLAB*, e o código está no Apêndice A5.

Como fizemos com a simulação de arbitragem estatística implícita, vamos trabalhar com um período de estimação de "*mispricing*" dentro da amostra entre 02 de janeiro de 2004 até 29 de dezembro de 2005, para um total de 498 observações, enquanto o período fora da amostra começa em 02 de janeiro de 2006 e vai até 28 de dezembro de 2006, para um total de 246 dados. Após estimar modelos para as 55 ações, com 498 dados em cada, obtemos 29 "*mispricings*" cointegrados. Para cada um desses vinte e nove vamos simular retornos para a estratégia de Arbitragem Estatística Condicional. Na Figura 4.15 podemos ver o retorno obtido pela estratégia no período de 246 dias úteis, para o *"mispricing*" de VALE5, usando *k=1* e custos de transação zero.

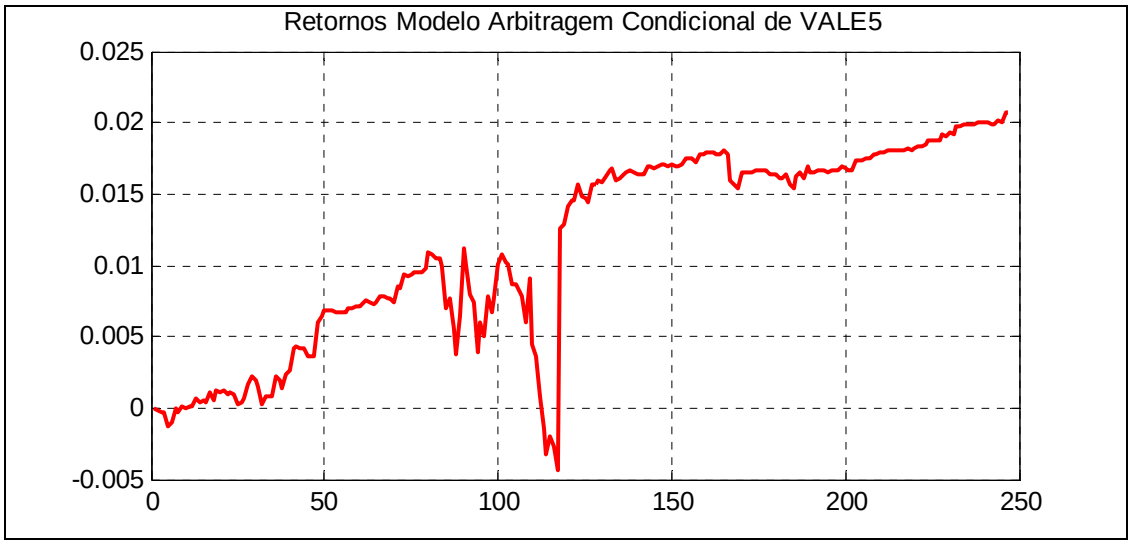

Figura 4.15: Retornos de arbitragem estatística condicional de VALE5

O retorno obtido foi de 2,08%, com volatilidade de 0,67%, para um Sharpe de 3,12. Embora a relação risco-retorno seja bastante boa, o retorno absoluto ainda não está satisfatório, mais sobre isso adiante. Na Figura 4.16, vamos comparar as previsões feitas pelo modelo neural com a evolução da dinâmica do "*mispricing*" de VALE5. Note-se que as previsões que aparecem aqui são a média, ponto a ponto, de cinco diferentes redes.

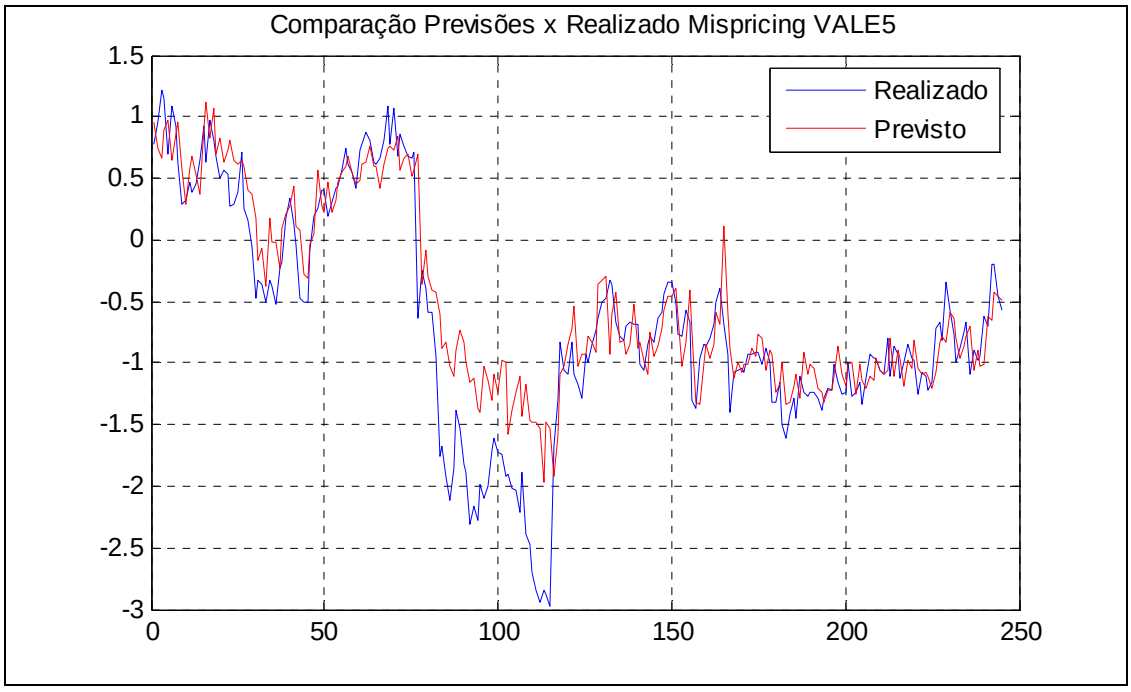

Figura 4.16: Qualidade das previsões do "*mispricing*" de VALE5

Numa inspeção visual aparentemente as redes neurais conseguem capturar bem a maior parte do comportamento do "*mispricing*", ou seja, *a priori*, deveriam ser capazes de gerar bons retornos.

Antes de examinarmos o conjunto geral das previsões, vamos nos deter momentaneamente no caso de SBSP3. No item 4.3.3, este foi o modelo que gerava o pior retorno, basicamente porque após o período de estimação, o "*mispricing*" se desviou sistematicamente da média, adotando uma tendência, que se mostrou catastrófica naquele caso. Como a proposta do modelo de previsão neural é justamente melhorar a performance das estratégias de arbitragem estatística nesses casos, vamos examinar a performance na previsão do comportamento do "*mispricing*" de SBSP3. Na Figura 4.17 comparamos as previsões com a realização, e na Figura 4.18 temos o retorno deste modelo de arbitragem.

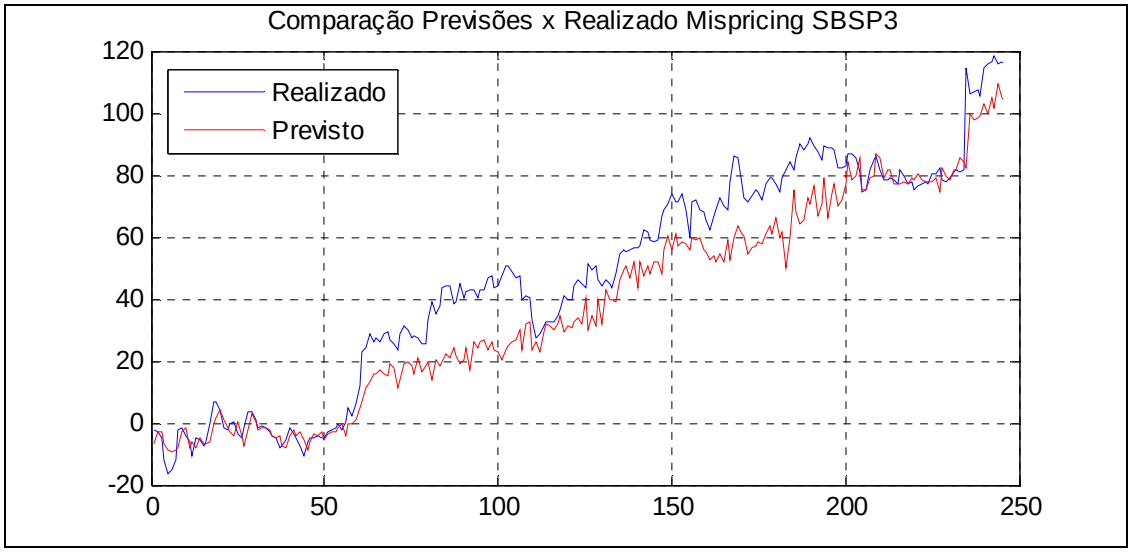

Figura 4.17: Qualidade das previsões do "*mispricing*" de SBSP3

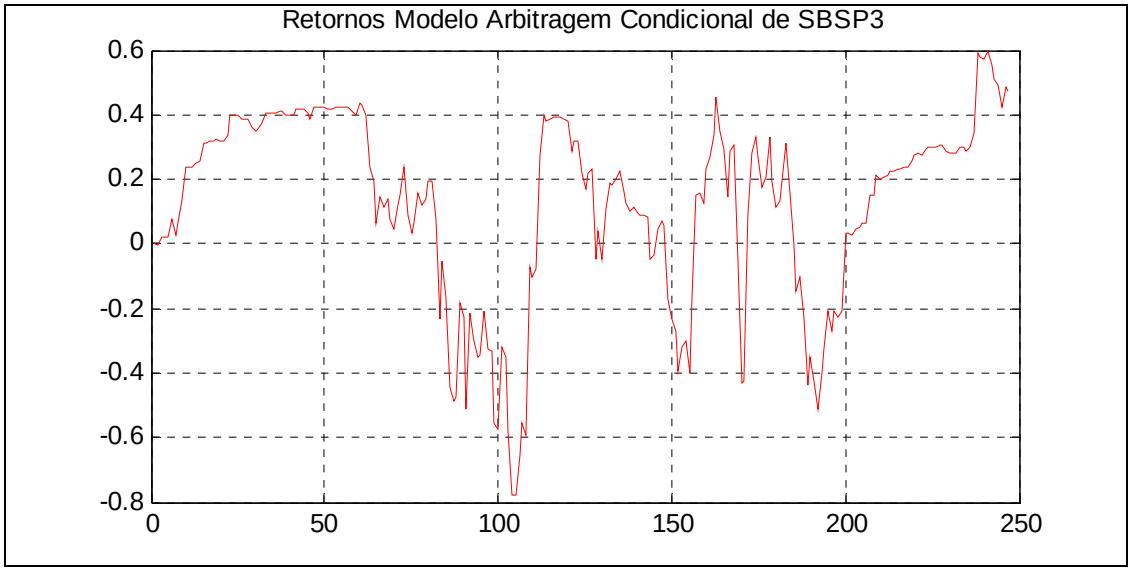

Figura 4.18: Retornos de arbitragem estatística condicional de SBSP3

O retorno do modelo de SBSP3 se mostra bastante positivo, embora bastante volátil também: 45,04%, com volatilidade de 15,23%, para um Sharpe de 2,96. Portanto o modelo

conseguiu melhorar definitivamente a performance do modelo de arbitragem estatística implícita, que discutimos no item 4.3.3.

Do mesmo modo que fizemos anteriormente, vamos examinar os resultados para todos os 29 modelos de arbitragem. A Tabela 4.11 resume os resultados:

qtde	média	mediana	mínimo	mispricing mínimo	máximo	mispricing máximo	retornos positivos	retornos negativos
29	2,08%	3,51%	-72,0%	TNLP3	30,1%	DURA4	23	6

Tabela 4.11: Resumo das estratégias de Arbitragem Estatística Condicional

O portfólio composto pelas 29 estratégias obtém retorno de 2,08%, com volatilidade de 1,15%, para um índice de Sharpe respeitável de 1,82. Novamente, a relação risco-retorno é bastante interessante, e o fato de utilizarmos 29 estratégias simultaneamente proporciona um benefício de diversificação bastante importante. Na Figura 4.19 vamos examinar a evolução dos retornos deste portfólio.

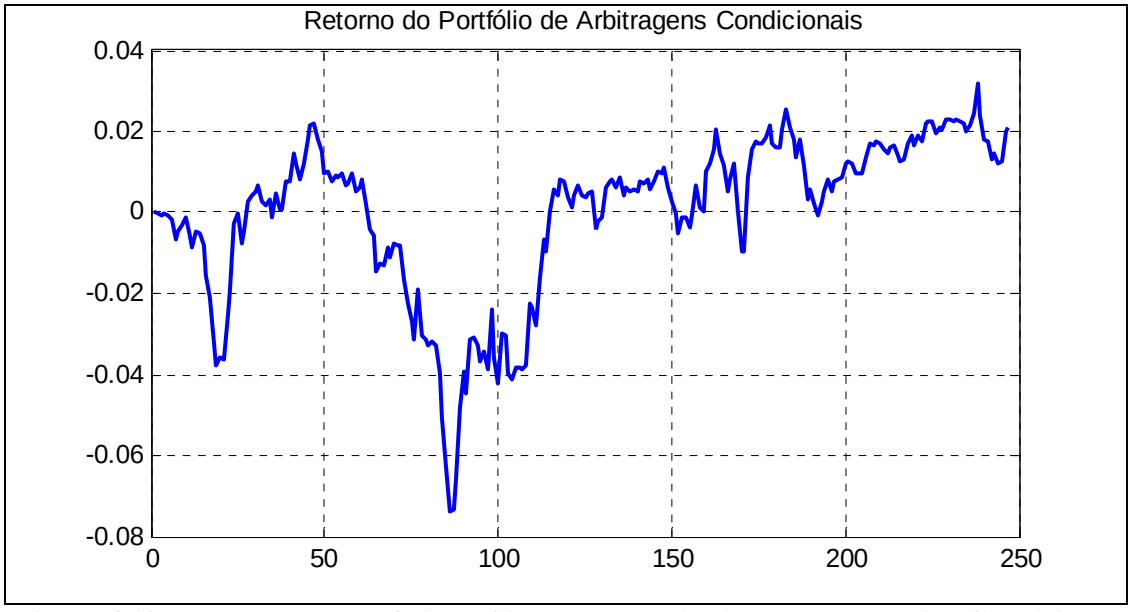

Figura 4.19: Retornos do portfólio de 29 modelos de Arbitragem Estatística Condicional

Note-se que apesar da baixa volatilidade os retornos finais não são particularmente fantásticos, ainda mais se considerarmos que utilizamos modelos com custo zero. Vamos examinar modelos com custos de 0,25%, semelhantes ao que usamos no item 4.3.3. Na Tabela 4.12 e na Figura 4.20 temos um resumo da performance desses modelos.

qtde	média	mediana	mínimo	mispricing mínimo	máximo	mispricing máximo	retornos positivos	retornos negativos
29	-24,29%	-9,90%	-190,3%	SBSP3	-0,1%	ITSA4	0	29

Tabela 4.12: Resumo das estratégias de Arbitragem Estatística Condicional (c=0,25%)

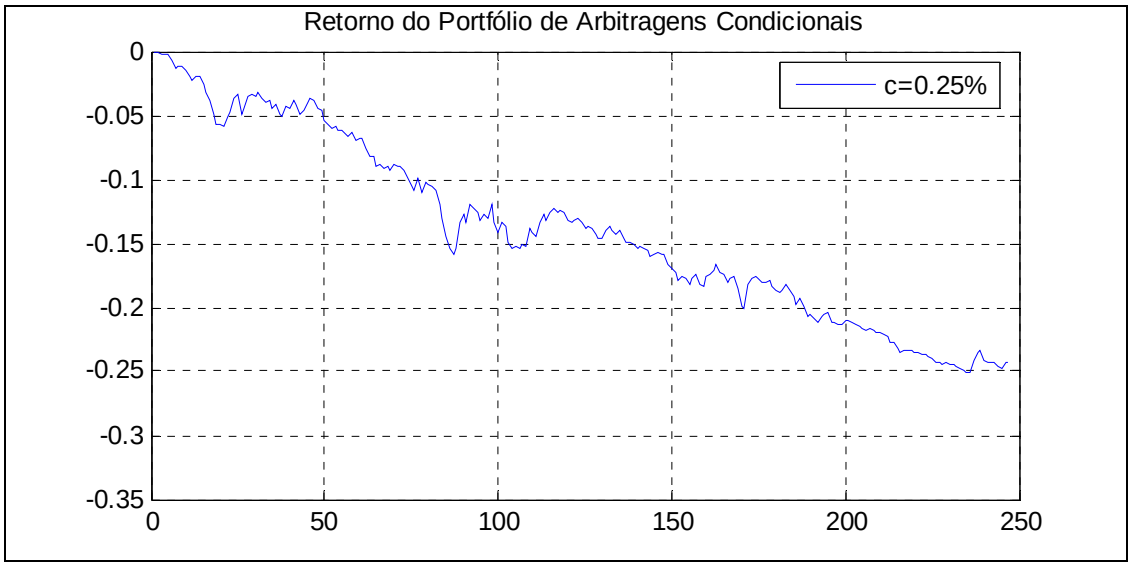

Figura 4.20: Retornos do portfólio de Arbitragem Estatística Condicional (c=0,25%)

Podemos ver pelos resultados que os modelos de arbitragem estatística condicional não lidam bem com a questão dos custos. É provável que o uso de previsões faça com que a posição do portfólio mude demais, e essas mudanças não gerem retornos, individualmente, para compensar os seus custos de transação.

Em resumo, pudemos ver como o modelo de previsões neurais permite a obtenção de uma melhor performance da maior parte das estratégias de arbitragem estatística, inclusive diminuindo consideravelmente os riscos e protegendo o portfólio de casos extremos de perdas, como vimos com SBSP3. Entretanto a regra de *trading* que utilizamos em conjunto com o modelo de previsões parece não ser a mais adequada, especialmente quando temos custos de transação. Assim, na busca por otimizar mais ainda os resultados financeiros obtidos pelo nosso conjunto de técnicas, vamos analisar a seguir outras alternativas de modelos de *trading*, e ao fim esperamos ter modelos robustos e (muito) lucrativos.

4.5. Modelos Completos

4.5.1. Estratégias de *Trading*

Segundo Towers & Burgess (2001), tão importante quanto a construção de um bom modelo de previsão é a sua implementação. Ou seja, devemos desenhar uma estratégia de *trading* que tome os sinais gerados pelo modelo de previsão por redes neurais e determine, a partir de regras paramétricas simples, qual a posição que deve ser tomada no mercado. Já vimos anteriormente como um modelo simples pode gerar bons resultados, mas o efeito dos custos de transação não é desprezível, podendo transformar um modelo de lucrativo em perdedor.

A literatura apresenta uma enormidade de possibilidades em termos de estratégias de *trading*. Via de regra, é possível agrupar os trabalhos em dois grupos: estratégias *"naive"*, onde uma previsão de retorno positivo gera uma compra e de retorno negativo gera uma venda, como por exemplo em Jasic & Wood (2004) ou Parreiras (2003); ou estratégias com algum tipo de filtro, onde o sinal gerado pelo modelo de previsão passa por um filtro que determina se o modelo compra, vende ou fica fora do mercado. Esse filtro pode ser algum retorno mínimo esperado, o valor dos custos de transação, o retorno de uma taxa livre de risco, ou qualquer outro limitador que o autor queira impor. Contém exemplos desse tipo de implementação os trabalhos de Casqueiro & Rodrigues (2006), Chen et al. (2003) ou ainda Kuo et al. (2001). Towers & Burgess (2001) expandem as possibilidades de estratégias de implementação, ao propor e testar cinco tipos diferentes de regras paramétricas de *trading*.

Temos primeiro uma formulação básica de uma regra de decisão paramétrica, que usa apenas a previsão do retorno de um ativo e ignora qualquer outra informação a respeito de decisões recentes, definida como:

$$D_t(\Delta\hat{y}_t, k, m) = m \cdot \left|\Delta\hat{y}_t\right|^k \cdot sign\left(\Delta\hat{y}_t\right) \tag{4.15}$$

onde k e m são dois parâmetros de decisão, $\Delta\hat{y}_t$ é o retorno previsto, e D_t é a decisão de *trading* a ser tomada no tempo *t*. Os parâmetros k e m controlam, respectivamente, o formato e a magnitude da função de decisão. Vamos trabalhar aqui com dois casos, para simplificar: $k = 0$, que nada mais é do que a estratégia *"naive"* que discutimos anteriormente, onde temos uma função degrau, onde uma previsão de retorno positivo gera uma decisão de compra, e um retorno negativo gera uma decisão de venda[34]; e $k = 1$, ou seja, tomamos uma posição proporcional ao retorno que nosso modelo prevê.

A partir deste modelo simples, Towers & Burgess (2001) determinam três outras regras de *trading* paramétricas que dependem das posições passadas[35]. São elas:

- *Média Móvel*: essa regra leva em conta as últimas *h* decisões para decidir o que fazer no instante *t*. Towers & Burgess (2001) sugerem o uso de $h = 5$. A regra pode ser escrita como:

$$D_t^*(h) = \frac{1}{h} \cdot \sum_{j=0}^{h-1} D_{t-j} \tag{4.16}$$

- *Média Móvel Exponencial*: semelhante à regra anterior, mas utilizando uma espécie de decaimento exponencial na tomada de decisão, controlado por um parâmetro θ. Towers & Burgess (2001) sugerem o uso de $\theta = 0{,}67$. A regra pode ser escrita como:

$$D_t^*(\theta) = \theta \cdot D_t + (1-\theta) \cdot D_{t-1}^* \tag{4.17}$$

[34] Note que não falamos nada da posição anterior, ou seja, admitimos fazer *short* de um ativo.

[35] *"Path dependent"* na terminologia de Towers & Burgess (2001).

➢ *Filtro com Degrau*: essa regra é semelhante àquelas que mencionamos anteriormente, incorporando algum tipo de filtro. Se espera que o sinal gerado pelo modelo de previsão seja mais potente do que um limite aceitável mínimo, controlado pelo parâmetro λ, que Towers & Burgess (2001) sugerem ser $\lambda = 0{,}67$. Formalmente, podemos definir essa regra como:

$$D_t^*(\lambda) = \begin{cases} D_t & se \quad \left| D_t - D_{t-1}^* \right| > \lambda \\ D_{t-1}^* & c.c. \end{cases} \tag{4.18}$$

Assim, temos cinco diferentes regras de *trading* passíveis de implementação. Note-se que Towers & Burgess (2001) simulam todas elas num problema de arbitragem estatística de índices europeus de ações[36], e concluem que as três regras mais sofisticadas geram tanto performance quanto índice de Sharpe melhores do que as duas regras simples. Entre todas, a melhor é a regra com média móvel exponencial, seguida da regra de filtro com degrau, da média móvel, da regra linear e por fim da regra "*naive"*.

Claramente a implementação de uma estratégia de *trading* mais sofisticada do que o simples compra e vende pode trazer benefícios adicionais em termos de performance. Vamos testar inicialmente a performance da regra de média móvel com $h = 5$[37] e da regra de média móvel exponencial com $\theta = 0{,}67$. O contexto é o mesmo que testamos no item 4.4.4, ou seja, o conjunto de 29 *"mispricings*" cointegrados, estimados no período de dois anos até o fim de 2005, e testado fora da amostra no ano de 2006. As estatísticas dos resultados podem ser vistas nas Tabelas 4.13 e 4.14, que podem ser comparados à Tabela 4.11 (inicialmente testamos os modelos sem custos).

qtde	média	vol.	Sharpe	mediana	mínimo	mispricing mínimo	máximo	mispricing máximo	retornos positivos	retornos negativos
29	7,84%	2,49%	3,14	5,08%	-46,1%	TNLP3	43,7%	SBSP3	25	4

Tabela 4.13: Estratégias de Arbitragem Estatística Condicional com $\theta = 0{,}67$

qtde	média	vol.	Sharpe	mediana	mínimo	mispricing mínimo	máximo	mispricing máximo	retornos positivos	retornos negativos
29	3,78%	1,75%	2,16	3,41%	-58,2%	TNLP3	36,6%	BBDC4	24	5

Tabela 4.14: Estratégias de Arbitragem Estatística Condicional com $h = 5$

Note-se como em ambos os casos tanto o retorno médio quanto a mediana melhoraram, além da proporção de retornos positivos versus negativos. Ou seja, ambas as regras de *trading* trazem ganhos em relação à regra que utilizamos no item 4.4.4. Mais ainda, vamos comparar a

[36] Vide Towers & Burgess (2001), pp. 323-325 para detalhes.

[37] Realizamos diversos testes com diferentes parâmetros *h*, e para modelos sem custos, *h=2* tende a performar melhor que *h=5*. Mas para modelos com custos como os que discutimos adiante, *h=5* performa melhor que todas as outras opções testadas. Por economia de espaço não vamos detalhar aqui estes resultados.

performance dessas regras quando estamos operando com custos, para podermos aferir se o fato de filtrarmos as operações traz algum ganho. Os resultados comparativos podem ser visto na Tabela 4.15.

Regra	qtde	média	vol.	Sharpe	mediana	mínimo	mispricing mínimo	máximo	mispricing máximo	retornos positivos	retornos negativos
naive	29	-24,29%	7,74%	-3,14	-9,90%	-190,3%	SBSP3	-0,1%	ITSA4	0	29
$\theta = 0.67$	29	-22,69%	7,30%	-3,11	-5,73%	-247,8%	TNLP3	13,0%	CRUZ3	6	23
$h = 5$	29	-6,50%	2,01%	-3,23	-0,58%	-127,0%	SBSP3	17,6%	BBDC4	14	15

Tabela 4.15: Comparação da Performance de três regras de *trading*

Claramente a regra de média móvel simples obtém os melhores retornos, quando estamos operando com custos. Ela performa melhor que a regra exponencial e também é bastante superior à regra simples que estávamos utilizando para examinar as estratégias de Arbitragem Estatística Condicional. Assim, podemos entender como uma regra de *trading* mais sofisticada pode agregar um retorno bastante importante a um modelo como o que aqui projetamos. Para visualizar os resultados da Tabela 4.15, basta observar a Figura 4.21. Fica patente que o melhor resultado é obtido pela regra de média móvel simples.

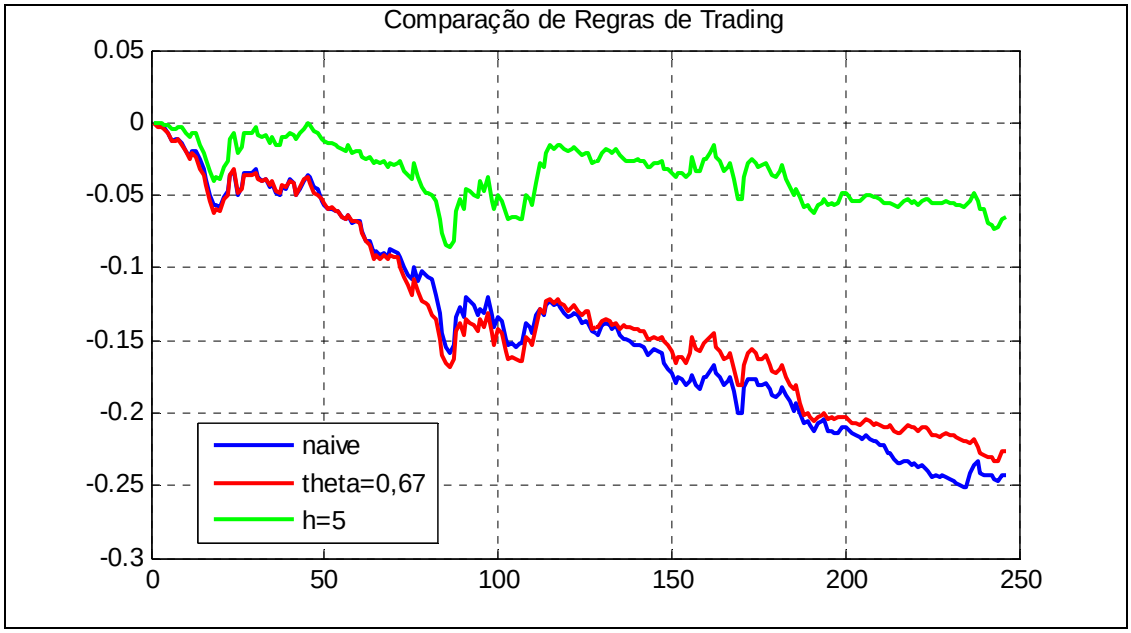

Figura 4.21: Comparação da Performance de três regras de *trading*

Uma vez que encontramos uma regra de *trading* que obtém melhor performance, a grande questão que resta é: como fazer agora para que nosso portfólio de arbitragens estatísticas obtenha um retorno final positivo? Afinal de nada adiantaria desenvolver todos esses modelos se eles não pudessem ser mostrar lucrativos. Para responder a esta pergunta, vamos introduzir mais dois conceitos no item a seguir, e mostrar enfim os resultados finais de nossos experimentos.

4.5.2. Políticas de *Stop-Loss*, Alavancagem e Re-investimento

A análise dos resultados obtidos anteriormente é importante em dois pontos: o primeiro é que um portfólio de modelos lucrativos depende de se evitar modelos catastróficos, e o segundo é que alguns modelos performam extremamente bem. Essas duas constatações podem ser resumidas num conhecido ditado do mercado financeiro: "*cut your losses and ride your winners*"[38]. A idéia aqui é determinar maneiras de reduzir as perdas, através do uso de políticas de *stop-loss*, e também modos de incrementar os ganhos, com o uso judicioso de alavancagem.

Harris (2003) define uma ordem de *stop-loss* como "uma ordem para comprar apenas quando o preço sobe além do preço de *stop*, ou para vender apenas após o preço cair além do *stop*, e mais comumente *traders* usam ordens de *stop* para estancar suas perdas, quando os preços se movem contra suas posições"[39]. A idéia básica é não deixar que um desenvolvimento adverso no mercado se torne catastrófico. Uma das grandes dificuldades que operadores de mercado encontram no dia a dia é manter a disciplina de executar a política de *stop-loss* – não é fácil aceitar que a decisão de comprar ou vender um ativo estava errada, e zerar a posição. Em nossos modelos, como estamos falando de *trading* automático, usando modelos, não encontramos essas dificuldades. Aqui, os problemas são de outra ordem.

A principal questão é: qual o tamanho do *stop-loss* ideal? Ou seja, qual tamanho de perda é digna de acionar a zeragem da posição, e qual perda é aceitável no contexto da volatilidade do mercado. Ao simular uma política de *stop-loss* em nossos modelos, vamos lidar com a seguinte situação: inicialmente, estaremos operando os 29 modelos de arbitragem simultaneamente. Quando um modelo atingir um ponto de *stop-loss*, vamos zerar as posições do modelo para aquele *"mispricing"* , e considerar que este modelo não teve sucesso. Assim, o retorno deste modelo ficará registrado como sendo o ponto onde ele foi *"stopado"*, para usar o jargão do mercado. Não reiniciaremos operações de um modelo que já atingiu um *"stop-loss*" porque introduziríamos vários graus de liberdade adicionais que complicariam sobremaneira a exposição dos resultados. Voltando à questão do início do parágrafo, o mais complicado é determinar qual o ponto de *stop-loss* ideal: se colocarmos ele em 10% de perda, podemos descartar modelos apenas temporariamente perdedores. Agora, se colocarmos ele em 25% de perdas, podemos permitir que modelos catastróficos permaneçam tempo demais no portfólio, causando estragos talvez irrecuperáveis. Assim, esse equilíbrio é um tanto fino e difícil de ser atingido, e mais ainda, nada garante que o *stop* ótimo no passado vai ser o *stop* ótimo no futuro.

---

[38] Na tradução livre do autor: "corte suas perdas e continue com as posições vencedores". Não é conciso como no inglês, mas transmite o significado. É difícil determinar um autor para este ditado. Uma fonte onde aparece recorrentemente é em Schwager (1993), por exemplo pp. 165 na entrevista com Ed Seykota.

[39] Harris (2003), pp. 78, tradução do autor.

Assim sendo, deixamos os dados falarem e mostrarem como se comportam os resultados quando determinamos diferentes pontos de *stop-loss*. Visto que determinamos no item 4.5.1 que o modelo com regra de *trading* por média móvel obtia os melhores resultados, vamos implementar e simular apenas modelos desse tipo daqui por diante. Novamente, estaremos trabalhando com o conjunto de 29 modelos de "*mispricings*" cointegrados, estimados dentro da amostra nos anos de 2004 e 2005, e testados fora da amostra no ano de 2006. Vamos simular os modelos com políticas de *stop-loss* para perdas de 10%, 15% e 25%.

A Figura 4.22 mostra dois exemplos de modelos que pararam de operar por atingirem seus *stops*. No caso em questão utilizamos *stop-loss* de 10%. Note-se como o modelo de CMIG3 vai acumulando perdas aos poucos ao longo do tempo, atingindo o ponto de *stop* apenas próximo ao fim do nosso período de simulação, com uma perda de 10,59%. Já o modelo de USIM5 sofre uma quebra estrutural e rapidamente passa de um modelo ganhador para um modelo perdedor, atingindo *stop* com uma perda de 14,68%. Note-se que determinamos o limite de 10%, mas apenas podemos saber que este limite foi ultrapassado *a posteriori*, e portanto muitas vezes a perda acaba sendo maior do que o limite.

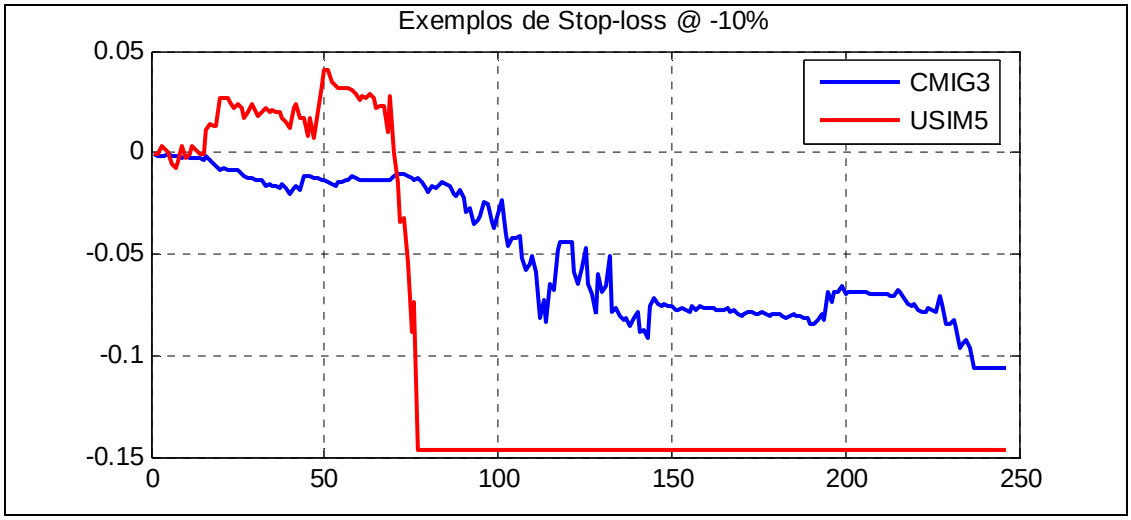

Figura 4.22: Exemplos do funcionamento do *stop-loss*

A Tabela 4.16 mostra os resultados das simulações com os três diferentes níveis de *stop-loss*: 10%, 15% e 25%. As estatísticas são semelhantes à que mostramos na Tabela 4.15.

Stop-Loss	qtde	média	vol.	Sharpe	mediana	mínimo	mispricing mínimo	máximo	mispricing máximo	retornos positivos	retornos negativos
-10%	29	-2,26%	0,80%	-2,82	-1,29%	-14,7%	USIM5	25,8%	BBDC4	10	19
-15%	29	-3,24%	1,20%	-2,69	-1,29%	-23,8%	TNLP3	29,7%	BBDC4	10	19
-25%	29	-2,77%	1,30%	-2,13	-0,58%	-31,3%	TNLP3	29,7%	BBDC4	13	16

Tabela 4.16: Comparação da Performance de três políticas de *stop-loss*

Note-se como o *stop-loss* em 25% traz os melhores resultados: embora sua média seja marginalmente pior que a do modelo com 10%, sua mediana é superior, a proporção de modelos

positivos e modelos negativos é bastante superior, a relação risco-retorno também é melhor, e o seu pior resultado é apenas marginalmente maior do que o dos outros modelos. A Figura 4.23 mostra graficamente os resultados da Tabela 4.16, mostrando a evolução dos retornos médios dos três diferentes portfólios.

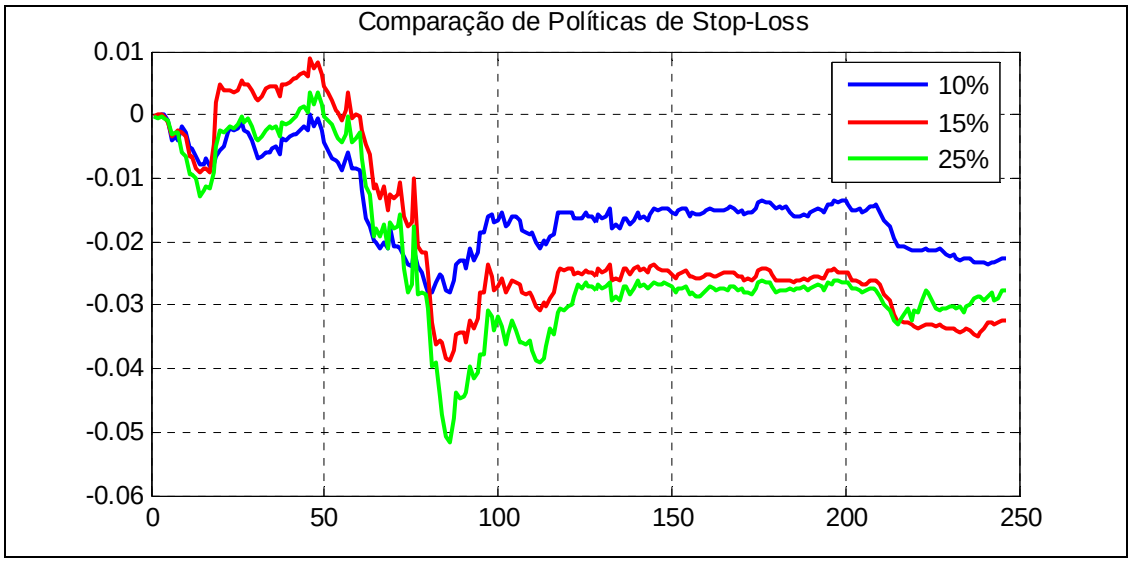

Figura 4.23: Comparação da Performance de três políticas de *stop-loss*

Os resultados obtidos nestas simulações nos permitem dizer que, para nossas circunstâncias, um *stop-loss* de 25% nos parece mais adequado do que um *stop-loss* menor. Essa conclusão vem do fato de que com uma política menos restritiva conseguimos obter uma proporção maior de modelos lucrativos, uma vez que não estaremos suspendendo as operações de modelos que apenas momentaneamente apresentam prejuízos.

A conclusão mais importante nesse ponto é: em todos os casos em que adotamos políticas de *stop-loss*, os resultados finais são superiores aos obtidos por modelos sem tal recurso. Para tanto, basta comparar as Tabelas 4.15 e 4.16: na primeira o modelo com média móvel simples e *h=5*, sem *stop-loss*, obtém retorno médio de **-6,15%**, com o modelo do *"mispricing"* de SBSP3 catastrófico perdendo -127,0%, enquanto que na segunda tabela, para qualquer das três políticas, o retorno médio do portfólio nunca é inferior a **-3,24%**, ou seja quase duas vezes melhor, e o pior modelo de todos perde -31,3%, ou seja, quatro vezes melhor. Claramente a implementação de políticas de *stop-loss* adiciona valor à nossa metodologia de arbitragem estatística.

A definição de uma política de *stop-loss* nos permite eliminar os modelos potencialmente catastróficos, mas ainda assim os resultados da Tabela 4.16 não permitem ainda responder a questão postulada ao fim do item 4.5.1, e os retornos positivos continuam elusivos. Para tentar mudar isso, vamos buscar mais uma ferramenta, esta talvez uma das mais perigosas disponíveis: a alavancagem. Nicholas (2000) define alavancagem como "o uso de fundos emprestados, ou de derivativos, para criar exposições que sejam em excesso em relação ao montante de ativos para

investimento. Por exemplo, alavancagem pode ser utilizada para criar uma exposição de um dólar e meio para cada dólar investido"[40]. Até aqui, nossa regra de *trading* está sendo definida por duas equações:

$$AEC\left(E\left[\Delta M_t\right],k\right)_t = sign\left(E\left[\Delta M_t\right]\right)\cdot\left|E\left[\Delta M_t\right]\right|^k \tag{4.19}$$

$$AEC_t^*(h) = \frac{1}{h}\cdot\sum_{j=0}^{h-1} AEC_{t-j} \tag{4.20}$$

A equação 4.19 é uma reprodução da equação 4.12 que definimos quando discutimos os modelos iniciais de arbitragem estatística condicional, no item 4.4.4, e a equação 4.20 é apenas uma variante da equação 4.16 que introduzimos ao explicar estratégias de *trading* no item 4.5.1. Agora, note a semelhança entre a equação 4.19 e a equação 4.21:

$$D_t(\Delta\hat{y}_t,k,m) = m\cdot\left|\Delta\hat{y}_t\right|^k\cdot sign\left(\Delta\hat{y}_t\right) \tag{4.21}$$

Até aqui estávamos usando, implicitamente, um parâmetro $m=1$ em todos nossos modelos, para controlar a magnitude da posição tomada. Entretanto, não há nada que nos obrigue a tanto, e por isso, na busca por melhorar os retornos, vamos passar a trabalhar com magnitudes de posição, na busca por maiores retornos. Em outras palavras, vamos **alavancar** os modelos, usando uma regra de *trading* redefinida por:

$$AEC\left(E\left[\Delta M_t\right],k\right)_t = m\cdot sign\left(E\left[\Delta M_t\right]\right)\cdot\left|E\left[\Delta M_t\right]\right|^k \tag{4.22}$$

O racional para tal decisão é simples: ao alavancar, aumentamos as chances de os modelos obterem maiores retornos. O risco vale para os dois lados, ou seja, teoricamente podemos também amplificar as perdas. Disso, contudo, já cuidamos ao definir uma política de *stop-loss*. Assim, estamos finalmente chegando perto de alcançar o objetivo daquele ditado inicial, "*cut your losses and ride your winners*".

Para determinar os efeitos da alavancagem em nosso portfólio, vamos novamente simular os modelos. As características básicas são as mesmas, e vamos determinar um *stop-loss* fixo de 25%. Os períodos também são iguais, e portanto teremos portfólios de 29 modelos para analisar. Vamos testar três níveis de alavancagem: $m=2$, $m=10$ e $m=20$. A Tabela 4.17 mostra os resultados médios para as três simulações:

m	qtde	média	vol.	Sharpe	mediana	mínimo	mispricing mínimo	máximo	mispricing máximo	retornos positivos	retornos negativos
2	29	-5,86%	2,64%	-2,22	-1,85%	-55,0%	USIM3	42,4%	BBDC4	12	17
10	29	-15,37%	6,71%	-2,29	-26,85%	-84,6%	SBSP3	73,8%	CPLE6	7	22
20	29	-10,29%	6,57%	-1,57	-28,36%	-104,8%	SBSP3	274,1%	CRUZ3	4	25

Tabela 4.17: Comparação da Performance de três níveis de alavancagem

A Figura 4.24 plota a evolução dos três portfólios médios com 29 modelos cada.

[40] Nicholas (2000), pp. 251, tradução do autor.

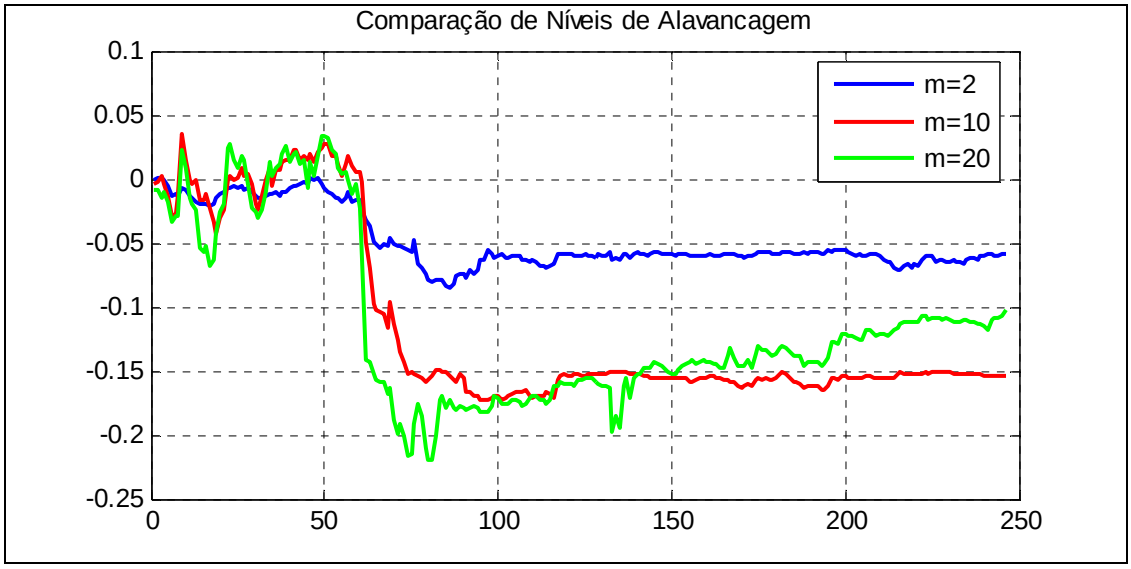

Figura 4.24: Comparação da Performance de três níveis de alavancagem

Os resultados, principalmente para $m = 20$, se mostram promissores. Note como a curva de retornos inicialmente cai, com a má performance de alguns modelos, que vão sendo progressivamente eliminados, e depois começa a subir, conforme restam bons modelos, cujos retornos são alavancados e vão compensando os modelos piores. Ainda assim, o retorno médio dos três modelos termina sendo negativo, o que vai contra nossas expectativas iniciais. Para contornar isto, e mostrar como é possível obter retornos positivos através da combinação dos conceitos de arbitragem estatística e inteligência artificial, vamos parar um momento e analisar friamente os resultados que computamos até aqui.

Toda vez que definimos um portfólio de estratégias de arbitragem estatística, partimos da seguinte premissa: tenho um capital ***x***, e vou dividir este capital igualmente entre o número ***n*** de estratégias disponíveis. Portanto, meu retorno final será a média simples dos retornos de cada modelo, que vai operar com capital ***x/n***. Agora, ao implementar uma política de *stop-loss*, teremos diversos modelos que, ao longo do caminho, deixarão de operar, por terem atingido seu limite de perdas. Assim, sua parcela ***x/n*** de capital ficará livre a partir de um certo momento no tempo (óbvio, subtraindo-se a perda realizada). A questão que se coloca é: porque não utilizar este capital livre e investi-lo nos modelos que até aqui se mostram lucrativos? Essa seria a epítome da aplicação do "*cut your losses and ride your winners*". Mais ainda, ao usarmos de alavancagem, estaremos multiplicando este efeito, ao cortar rapidamente os modelos com perdas, e dirigir todo o capital para os modelos potencialmente lucrativos.

Exemplificando simplisticamente esse conceito de reutilização do capital: temos R$ 2 milhões e duas estratégias, cada qual recebe R$1 milhão inicialmente para operar. A primeira rapidamente atinge seu *stop-loss* de 25%, parando de operar e deixando livre um capital de R$ 750.000. Assim, a partir deste momento, vamos investir este capital na segunda estratégia, que

passa a trabalhar com um capital de R$ 1.750.000 (estamos supondo que este segundo modelo não teve nem ganhos nem perdas até o momento). A expectativa é que os retornos dessa segunda estratégia mais do que compensem a perda de R$ 250.000 da primeira, e ao fim tenhamos um retorno geral positivo. Essa idéia de re-investimento do capital é simples e intuitiva, e no item a seguir vamos aplicá-la para mostrar como os modelos de arbitragem estatística com redes neurais obtêm excelentes retornos.

4.5.3. Modelo Final

Ao longo deste capítulo, progressivamente utilizamos os conceitos teóricos dos capítulos 2 e 3 para construir uma metodologia de arbitragem estatística com inteligência artificial. A essa combinação juntamos algumas técnicas padrão de finanças, refinando estratégias de *trading* e de uso do capital. Assim, chegamos aqui com um modelo final bem definido, e vamos analisar detidamente sua performance. O código de *MATLAB* para o modelo final está no Apêndice A6.

O modelo final é composto dos seguintes elementos: a combinação da metodologia de construção de dinâmicas de "*mispricing*" (item 4.3) com as técnicas de previsão não-paramétrica por redes neurais (item 4.4), aplicadas através de uma regra de *trading* com média móvel simples $h = 5$ (item 4.5.1), com política de *stop-loss* de 30%[41], alavancagem $m = 20$ e re-investimento sistemático do capital (item 4.5.2). É esse modelo cujos resultados mostramos a seguir. Como anteriormente, vamos trabalhar com um período dentro da amostra entre 02 de janeiro de 2004 e 29 de dezembro de 2005, para um total de 498 observações, enquanto o período fora da amostra vai de 02 de janeiro de 2006 até 28 de dezembro de 2006, com um total de 246 dados. Os resultados finais estão na Tabela 4.18.

qtde	retorno	vol.	Sharpe	mínimo	mispricing mínimo	máximo	mispricing máximo	retornos positivos	retornos negativos
29	80,24%	22,80%	3,52	-97,9%	SBSP3	252,1%	CRUZ3	5	24

Tabela 4.18: Performance do Modelo Final

Note-se que não mostramos a mediana dos retornos por não fazer sentido quando utilizamos a técnica de re-investir o capital nos melhores modelos. O retorno mínimo corresponde ao ponto onde o modelo atingiu o *stop-loss*, embora este fosse de apenas 30%. Claramente fazer arbitragem estatística de uma ação como SBSP3 não é simples, como já disséramos no item 4.3.3.

A Figura 4.25 mostra a curva de rentabilidade do modelo final.

[41] Em nossas simulações, o *stop-loss* de 30% obteve melhores resultados finais do que o de 25%. As linhas gerais dos resultados não mudam, contudo.

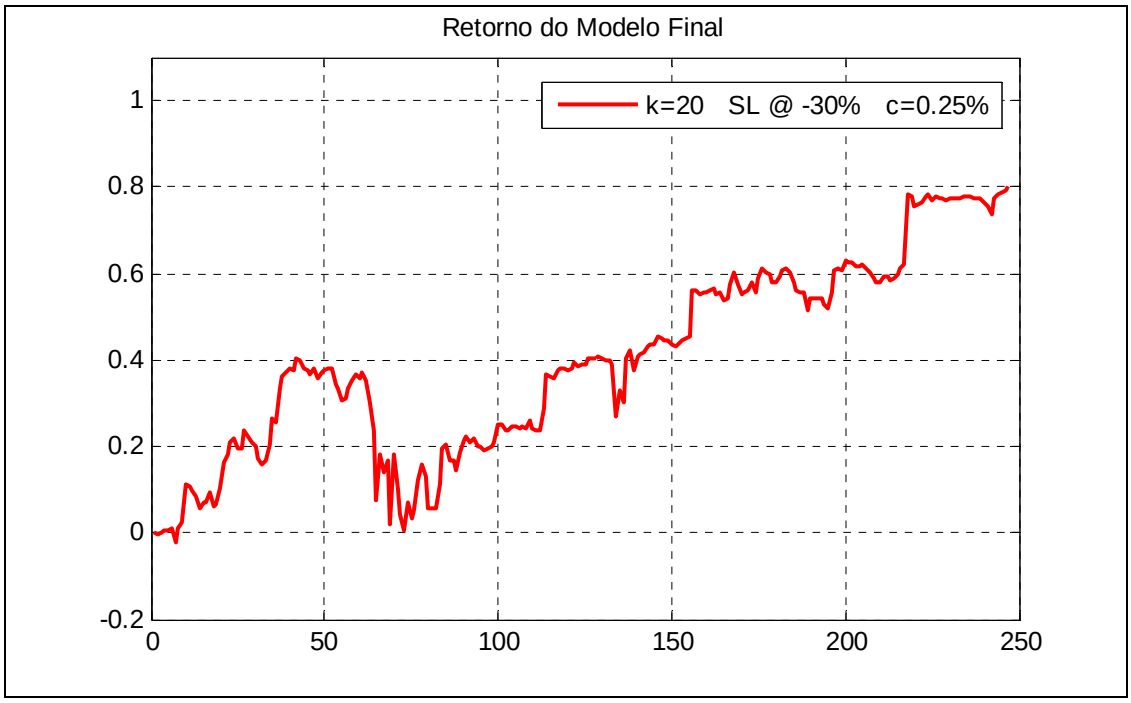

Figura 4.25: Evolução dos Retornos do Modelo Final

É importante dizer que os retornos sofrem uma perda considerável entre meio de março de 2006 e fins de abril de 2006, ou seja, os ganhos não são sempre positivos e é crucial manter a disciplina do modelo nesses momentos.

A Figura 4.26 e a Tabela 4.19 mostram o funcionamento da técnica de re-investimento em ação. A partir da esquerda, progressivamente os modelos para determinados "*mispricings*" vão sendo retirados, conforme vão atingindo seus pontos de *stop-loss*, e o capital vai sendo concentrado naqueles modelos que continuam mantendo boa performance.

t =	1	5	7	11	14	19	20	26	34	35	36
modelos retirados		TNLP3	DURA4 EMBR3	GOAU4	TNLP4	GGBR4	ARCZ6	ITAU4	GETI4	FFTL4	RAPT4

t =	37	55	65	69	73	80	88	112	113	155	217
modelos retirados	CMIG4	UGPA4	BBDC4	SBSP3	USIM3 USIM5	CNFB4	CMIG3	VALE5	VALE3	PETR4	ELET3

Tabela 4.19: Modelos Excluídos do Portfólio ao longo do tempo

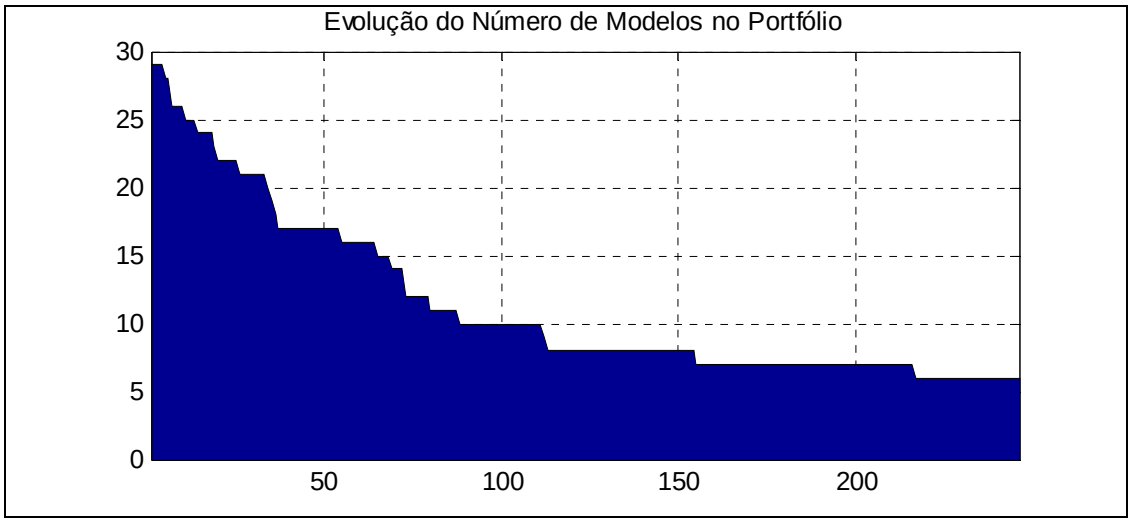

Figura 4.26: Evolução do número de modelos no Portfólio Final

Note como rapidamente vários dos modelos são retirados da amostra. Isso mostra que talvez os "*mispricings*" não estivessem bem estimados, ou que a rede neural não tenha sido capaz de aprender a prever suas dinâmicas. De todo modo, ao fim ainda restamos com 6 modelos de arbitragem, o que nos ajuda a obter ganhos de diversificação e não ficar dependente da performance de apenas um único modelo.

Um ponto importante a aferir é a correlação do modelo com os fatores de risco de mercado. Como repetimos diversas vezes nesse texto, buscamos aqui uma técnica que seja neutra ao mercado, ou seja, não depende diretamente da direção dos preços do mercado. O que desejamos é capturar retornos disponíveis devido a discrepâncias estatísticas no comportamento dos ativos. Assim, devemos calcular as correlações entre os retornos do nosso modelo e os retornos dos principais indicadores de mercado. Para tanto vamos olhar as seis variáveis que utilizamos nos nossos modelos de previsão, e calcular as correlações com os retornos do nosso modelo. Os resultados estão na Tabela 4.20.

correlação de retornos	**USD / BRL**	**Pré 90d**	**Pré 360d**	**Ibovespa**	**S&P 500**	**Nasdaq**
modelo	-3,38%	-7,22%	-7,66%	9,50%	10,80%	13,58%

Tabela 4.20: Correlações dos retornos do modelo com fatores de risco de mercado

As correlações são bastante baixas, e podemos dizer que alcançamos nosso objetivo de ter um modelo neutro ao mercado. Como comparação, temos que no período a correlação entre o Ibovespa e o S&P 500 foi de 75,04% e entre o Ibovespa e o dólar foi de -54,19%.

Por fim, é importante também observarmos brevemente o resultado de cada um dos 29 modelos que compõem, inicialmente, o portfólio. Assim, podemos ver quais performam bem e quais foram detratores. A Tabela 4.21 mostra os resultados: o grosso dos retornos vêm de operar

os modelos para CPLE6, CRUZ3, ELET6 e ITSA4. Embora estejam no portfólio até o fim do período, os modelos de PETR3 e UBBR11 não tem boa performance.

nº	mispricing	stop ?	quando	retorno	vol. período	Sharpe
1	ARCZ6	sim	30/jan/06	-37,3%	10,1%	(3,70)
2	BBDC4	sim	5/abr/06	-97,5%	61,1%	(1,60)
3	CMIG3	sim	11/mai/06	-31,7%	9,2%	(3,46)
4	CMIG4	sim	22/fev/06	-30,0%	9,2%	(3,25)
5	CNFB4	sim	28/abr/06	-33,3%	8,3%	(4,00)
6	CPLE6	não	-	146,6%	38,3%	3,83
7	CRUZ3	não	-	252,1%	85,7%	2,94
8	DURA4	sim	10/jan/06	-31,8%	15,0%	(2,13)
9	ELET3	sim	14/nov/06	-32,1%	8,3%	(3,85)
10	ELET6	não	-	45,3%	20,0%	2,26
11	EMBR3	sim	10/jan/06	-44,8%	18,0%	(2,49)
12	FFTL4	sim	20/fev/06	-38,9%	8,1%	(4,78)
13	GETI4	sim	17/fev/06	-44,5%	22,8%	(1,95)
14	GGBR4	sim	27/jan/06	-30,4%	8,1%	(3,73)
15	GOAU4	sim	16/jan/06	-39,9%	14,1%	(2,82)
16	ITAU4	sim	7/fev/06	-38,0%	12,5%	(3,04)
17	ITSA4	não	-	58,4%	24,2%	2,41
18	PETR3	não	-	-23,7%	7,9%	(3,01)
19	PETR4	sim	15/ago/06	-30,2%	10,4%	(2,89)
20	RAPT4	sim	21/fev/06	-36,6%	17,5%	(2,09)
21	SBSP3	sim	11/abr/06	-97,9%	90,8%	(1,08)
22	TNLP3	sim	6/jan/06	-30,4%	11,8%	(2,57)
23	TNLP4	sim	19/jan/06	-35,5%	11,9%	(2,97)
24	UBBR11	não	-	2,7%	3,8%	0,69
25	UGPA4	sim	22/mar/06	-39,5%	17,9%	(2,21)
26	USIM3	sim	18/abr/06	-78,1%	45,5%	(1,72)
27	USIM5	sim	18/abr/06	-42,5%	23,3%	(1,83)
28	VALE3	sim	16/jun/06	-31,1%	7,8%	(4,00)
29	VALE5	sim	14/jun/06	-30,4%	6,1%	(4,95)

Tabela 4.21: Performance dos 29 modelos de arbitragem

5. CONCLUSÕES

"Todos os modelos são errados, mas alguns podem ser úteis"
George Box

"Devemos usar modelos, não acreditar neles"
Henri Theil[1]

5.1. Conclusões

O desenvolvimento de um modelo de arbitragem estatística em combinação com técnicas de previsão por redes neurais se mostrou tarefa árdua e nem um pouco simples. Ao fim, obtivemos modelos bastante robustos, e nosso modelo final, que combinou todos os esforços teóricos e práticos, mostrou-se bastante promissor, com retornos simulados, na presença de custos, acima de 80% no período de um ano, com um respeitável índice de Sharpe de 3.52. A conclusão inicial é que atingimos os objetivos propostos no início deste trabalho, ao refinar um conjunto de técnicas que permitem identificar e explorar, sistematicamente, oportunidades de arbitragem estatística de ações no mercado brasileiro.

Uma conclusão importante é a importância do conhecimento de técnicas apropriadas de *trading* nos mercados. Boa parte dos trabalhos acadêmicos sobre modelos de previsão sofre da falta desse tipo de conhecimento, e seus resultados tendem a não ser reproduzíveis na prática de mercado. No nosso caso, tivemos que recorrer à caixa de ferramentas de *trading* para contornar o fato de que os modelos não são mostravam lucrativos na presença de custos. Basta comparar a Tabela 4.12 com a Tabela 4.18 para entender como essas técnicas foram fundamentais: saímos de um retorno negativo de 24% para um positivo de 80%, uma melhora de mais de cem por cento, apenas com o uso judicioso de boas práticas de operação no mercado. Este trabalho formaliza um conhecimento que existe na prática e contribui para que outros autores possam trazer resultados ainda mais expressivos.

Apesar desse foco inicial na parte de *trading*, devemos dar a devida importância às duas técnicas que formam a base de sustentação deste trabalho. A metodologia de cointegração, especialmente no contexto de Engle-Granger, se mostrou capaz de identificar grupos de ações cointegrados e passíveis de arbitragem estatística – não esqueçamos dos resultados promissores dos modelos de arbitragem estatística implícita. Tais modelos podem ser usados como base para aplicações reais, com alguns poucos aprimoramentos simples. E não só Engle e Granger merecem ser lembrados aqui (afinal já possuem um Nobel e o devido reconhecimento), mas

[1] Ambas citações a*pud* Kennedy (2003), com traduções do autor. George Box foi um dos mais famosos estatísticos de todos os tempos. Henri Theil foi um famoso econometrista.

principalmente Burgess, cujo trabalho[2] que serve como grande base para boa parte deste trabalho. A construção de "*mispricings*" e as análises de arbitragem estatística implícita e condicional, são todas baseadas no trabalho de Burgess. O fato de funcionarem no mercado brasileiro, instável, volátil e complexo, mostra a qualidade destas técnicas.

A segunda base fundamental de nossos esforços, o uso de redes neurais, também se mostrou, mais uma vez, extremamente importante e contribuiu decisivamente para os bons resultados aferidos por nossos modelos de arbitragem. Basta olhar as Figura 4.16 e 4.17: mesmo com o uso de poucas variáveis exógenas, conseguimos modelos de previsão bastante poderosos, tanto em casos de dinâmicas cuja característica é a reversão à média, quanto em casos de dinâmicas onde existe uma tendência clara. A plasticidade e a independência de parametrização das redes neurais mais uma vez se mostraram extremamente potentes, contribuindo decisivamente para nossos resultados. È importante contudo ressaltar o aspecto estocástico de um modelo de previsão com redes neurais: cada vez que treinamos uma rede, estamos fazendo uma busca em um espaço de estados, e o resultado desse busca nem sempre será o mesmo, para um conjunto de dados de entrada fixo. Buscamos contornar esse problema através da simulação repetida do treinamento das redes, mas fica o aviso. Convém também ressaltar a importância da boa escolha das características de uma rede neural: não é qualquer algoritmo que performa bem, nem qualquer rede que vai obter bons resultados. Parreiras (2003) mostrara a qualidade do algoritmo Levenberg-Marquardt, e sua escolha aqui contribuiu decisivamente para os altos retornos obtidos.

Antes de passar a uma breve discussão de potenciais avenidas para melhoria do trabalho, gostaríamos de deixar a seguinte conclusão, talvez um tanto controversa: os mercados são sim, passíveis de serem arbitrados. Mas fazê-lo está longe de ser simples, requer (muita) atenção aos detalhes e disciplina. Ainda assim, como bem diz Taleb (2001), "de maneira geral, nós subestimamos a parcela de aleatoriedade em praticamente tudo"[3], e portanto não nos enganemos: os resultados aqui apresentados podem ser fruto de sorte. O exemplo de James Simons e da Renaissance Technologies, citados no início do Capítulo 2, que há vinte e cinco anos usam técnicas estatísticas e computacionais para obter retornos extraordinários nos mercados, mostra que a sorte, embora seja um fator, não necessariamente é o principal.

5.2. Trabalhos Futuros

O número de maneiras em que podemos estender e melhorar os modelos aqui delineados é praticamente ilimitado. Assim, vamos dividir em três áreas os possíveis trabalhos futuros: (i)

[2] Burgess (2000) e vários outros artigos, muitos dos quais presentes na bibliografia deste trabalho.
[3] Taleb (2001), pp. 2.

novas formas de encontrar oportunidades de arbitragem estatística; (ii) mudanças nas técnicas de previsão e (iii) diferentes maneiras de operar os modelos.

O primeiro conjunto de melhorias está ligado às técnicas de cointegração e aos modelos dela resultantes. Lembremos primeiro que a metodologia de construção de "*mispricings*" faz uso intensivo de regressões lineares. É sabido que estas são bastante sujeitas a problemas, logo uma primeira melhoria testável seria o uso de técnicas de regressão robustas. Estas inclusive poderiam minimizar a influência de *outliers* nos dados, eventualmente melhorando a performance de modelos de *trading* baseados em reversão à média. Outra possível melhoria seria a incorporação de metodologias que permitissem identificar quais "*mispricings*" são mais previsíveis, através do uso de perfis de razão de variância. Essa técnica é amplamente discutida em Burgess (2000), com bons resultados. O uso desse tipo de modelo talvez evitasse os casos patológicos que encontramos ao longo do Capítulo 4, como os "*mispricings*" de TNLP3 e SBSP3. Como discutimos no item 4.3.3, técnicas para identificação de quebras estruturais também poderiam ser tentadas, modelos do tipo "*Markov switching*" ou HMM.

O segundo conjunto de possíveis aprimoramentos está ligado aos modelos de previsões. Como vimos no item 3.2, as variações de modelos de inteligência artificial usados para previsão em finanças são quase infinitas. Vale citar algumas que nos parecem promissoras, como o uso de comitês de redes, como aparece em Shadbolt & Taylor (2003), West et al. (2005) ou ainda Yu et al. (2005). Embora as possibilidades aqui sejam diversas, técnicas como *bagging*, *boosting* e vários outros esquemas de comitê se mostram promissoras, especialmente por diminuir o erro associado às previsões. West et al. (2005) é uma boa referência, ao comparar a performance de algumas dessas técnicas. Uma outra possibilidade de melhoria dos resultados é através do uso de metodologias paramétricas para a construção e seleção das redes neurais. Refenes & Zapranis (1999b) e Burgess (2000) desenvolvem uma elaborada metodologia para se apoiar em indicadores estatísticos durante o processo de escolha das variáveis e construção do modelo neural, que pode contribuir para nosso processo. Burgess (2000) vai além e combina o processo de criação de um portfólio de "*mispricings*" com a construção de um modelo neural, integrando todas as etapas com base apenas no risco e retorno obtido pelos modelos. A única ressalva é que eventualmente o modelo pode acabar sofrendo de "*overfitting*", ou seja, otimizado para um passado que não se repetirá mais. Por fim, uma técnica que recentemente tem sido objeto de grande número de artigos na literatura, especialmente de previsão em finanças, é a chamada máquina de vetor de suporte[4]. Um artigo que utiliza esta técnica é Huang et al. (2005), com aparentes bons resultados, e tanto Haykin (2001) quanto Shadbolt & Taylor (2003) são boas referências para mais detalhes. Suas vantagens estão ligadas à parcimônia do modelo e ganho computacional, além de lidarem melhor com espaços de muitas variáveis. No

[4] Do termo em inglês *support vector machine*.

contexto da nossa metodologia, podem oferecer boas possibilidades, tanto substituindo as redes neurais como formando comitês com elas.

Por fim, a terceira área passível de explorações futuras é a ligada diretamente ao mercado. Aqui as variáveis são muitas e as combinações possíveis idem. Podemos testar mais variáveis exógenas, o uso de mais ações (o crescimento do mercado brasileiro leva a isso), cestas de ações maiores ou menores, modelos de alta freqüência (que mudam de posição várias vezes ao longo do dia), a combinação com outros ativos, como moedas ou taxas de juros. A expansão geográfica também é uma possibilidade: mercados emergentes bem organizados, como no México, África do Sul ou no leste europeu, similares ao brasileiro, também podem oferecer boas possibilidades de aplicação das técnicas desenvolvidas nesse trabalho. Um portfólio global de arbitragem estatística parece uma idéia interessante e bastante promissora.

6. BIBLIOGRAFIA

ADYA, M.; COLLOPY, F. *How Effective are Neural Networks at Forecasting and Prediction? A Review and Evaluation*, In: Journal of Forecasting, Volume 17, Issue 6, pp. 481-495, Novembro 1998.

ALEXANDER, C. *Market Models: A Guide to Financial Data Analysis*, Chichester, West Sussex: John Wiley & Sons: 2001.

ALEXANDER, C.; DIMITRIU, A. *The Cointegration Alpha: Enhanced Index Tracking and Long-Short Equity Market Neutral Strategies*, Discussion Paper 2002-08, ISMA Centre Discussion Papers in Finance Series, Reading, Inglaterra, 2002.

ALEXANDER, C.; GIBLIN, I.; WEDDINGTON III, W. *Cointegration and Asset Allocation: A New Active Hedge Fund Strategy* , Discussion Paper 2001-03, ISMA Centre Discussion Papers in Finance Series, Reading, Inglaterra, 2003.

ANG, K.K.; QUEK, C. *Stock Trading Using RSPOP: A Novel Rough Set-Based Neuro-Fuzzy Approach*, In: IEEE Transactions On Neural Networks, Volume 17, Issue 5, pp. 1301-1315, Setembro 2006.

ARMANO G.; MARCHESI M.; MURRU A. *A hybrid genetic-neural architecture for stock indexes forecasting*, In: Information Sciences, Volume 170, Issue 1, pp. 3-33, Fevereiro 2005.

AZZINI, A.; TETTAMANZI, A. G. *A Neural Evolutionary Approach to Financial Modeling*, In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation. Seattle, Washington, USA, 2006. GECCO '06. ACM Press, New York, NY, pp. 1605-1612, 2006.

BASS, T. *Os Profetas de Wall Street*, Rio de Janeiro: Campus, 2000.

BURGESS, A.N. *A Computational Methodology for Modelling the Dynamics of Statistical Arbitrage*, Ph.D Thesis, Londres: Department of Decision Sciences, London Business School, 2000.

BURGESS, A.N. *Statistical Arbitrage Models of the FTSE100*, In: ABU-MUSTAFA, Y.S. *et al* (Eds.). *Computational Finance 1999*, pp. 297-312, Cambridge, Massachusetts: The MIT Press, 2001.

BURGESS, A.N. *Using Cointegration to Hedge and Trade International Equities*, In: DUNIS, C.; LAWS, J.; NAÏM, P. *Applied quantitative methods for trading and investment*, pp. 41-69, Chichester, West Sussex: John Wiley & Sons: 2003.

BUSCEMA, M.; SACCO, P.L. *Feedforward networks in financial predictions: the future that modifies the present*, In: Expert Systems, Volume 17, Number 3, pp. 149-170, Julho 2000.

CAMPBELL, J.Y; LO, A.W.; MACKINLAY, A.C *The Econometrics of Financial Markets*, Princeton, New Jersey: Princeton University Press, 1997.

CAO, Q.; LEGGIO, K. B.; SCHNIEDERJANS, M. J. *A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market*, In: Computers & Operations Research, Volume 32, Issue 10, pp. 2499-2512, Outubro 2005.

CARMONA, R. *Statistical Analysis of financial data in S-Plus*, New York: Springer, 2004.

CASQUEIRO, P. X.; RODRIGUES, A. J.L. *Neuro-dynamic trading methods*, In: European Journal of Operational Research, Volume 175, Issue 3, pp. 1400-1412, Dezembro 2006.

CHEN, A.-S.; LEUNG, M.T.; DAOUK, H. *Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index*, In: Computers & Operations Research, Volume 30, Issue 6, pp. 901-923, Maio 2003.

CHUN, S-H.; KIM, S. H., *Data mining for financial prediction and trading: application to single and multiple markets*, In: Expert Systems with Applications Volume 26, Issue 2, pp. 131-139, Fevereiro 2004.

DEBOECK, G. J. (Ed.) *Trading on the edge: neural, genetic, and fuzzy systems for chaotic and financial markets*, New York: John Wiley & Sons, 1994.

DEMUTH, H.; BEALE, M. *Neural Network Toolbox: for use with MATLAB – User's Guide Version 4*, Natick: The Mathworks, Inc., 2001.

DROBNY, S. *Inside the house of money: top hedge fund traders on profiting in the global markets*, Hoboken, New Jersey: John Wiley & Sons: 2006.

DUNBAR, N. *Inventing Money: The story of Long-Term Capital Management and the legends behind it*, Chichester, West Sussex: John Wiley & Sons: 2001.

ELLIOT, R. J.; VAN DER HOEK, J.; MALCOLM, W. P. *Pairs Trading*, In: Quantitative Finance, Volume 5, Number 3, pp. 271-276, Junho 2005.

ENKE, D.; THAWORNWONG, S. *The use of data mining and neural networks for forecasting stock market returns*, In: Expert Systems with Applications, Volume 29, Issue 4, pp. 927-940, Novembro 2005.

GATEV, E.; GOETZMANN, W. e ROUWENHORST, K. *Pairs Trading: performance of a relative value arbitrage rule*, Working Paper 7032, NBER, 1999.

HARLAND, Z. *Using Nonlinear Neurogenetic Models with Profit Related Objective Functions to Trade the US T-bond Future*, In: ABU-MUSTAFA, Y.S. *et al* (Eds.). *Computational Finance 1999*, pp. 327-342, Cambridge, Massachusetts: The MIT Press, 2001.

HARRIS, L. *Trading and Exchanges : Market Microstructure for Pratictioners*, New York: Oxford University Press, 2003.

HASSAN, M. R.; NATH, B.; KIRLEY, M. *A fusion model of HMM, ANN and GA for stock market forecasting*, In: Expert Systems with Applications, Volume 33, Issue 1, pp. 171-180, Julho 2007.

HAYKIN, S. *Redes Neurais: Princípios e prática*, Porto Alegre: Bookman, 2001.

HOGAN, S.; JARROW, R.; TEO, M.; WARACHKA, M. *Testing market efficiency using statistical arbitrage with applications to momentum and value strategies*, In: Journal of Financial Economics, Volume 73, Issue 3, pp. 525-565, Setembro 2004.

HUANG, W.; NAKAMORI, Y.; WANG, S-Y. *Forecasting stock market movement direction with support vector machine*, In: Computers & Operations Research, Volume 32, Issue 10, pp. 2513-2522, Outubro 2005.

HUNG, K.-K.; CHEUNG, Y.-M.; XU, L. *An Extended ASLD Trading System to Enhance Portfolio Management*, In: IEEE Transactions On Neural Networks, Volume 14, Issue 2, pp. 413-425, Março 2003.

JASIC, T.; WOOD, D. *The profitability of daily stock market indices trades based on neural network predictions: case study for the S&P 500, the DAX, the TOPIX and the FTSE in the period 1965–1999*, In: Applied Financial Economics, Volume 14, Issue 4, pp.285-297, Fevereiro 2004.

KALYVAS, E. *Using Neural Networks and Genetic Algorithms to Predict Stock Market Returns*, Dissertação de Mestrado, Manchester: Department of Computer Science, University of Manchester, 2001.

KENNEDY, P. *A Guide to Econometrics*, Cambridge, Massachusetts: The MIT Press, 2003.

KIM, K-J. *Artificial neural networks with evolutionary instance selection for financial forecasting*, In: Expert Systems with Applications, Volume 30, Issue 3, pp. 519-526, Abril 2006.

KUO, R.J.; CHEN, C.H.; HWANG, Y.C. *An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network*, In: Fuzzy Set and Systems, Volume 118, Issue 1, pp. 21-45, Fevereiro 2001.

LARSSON, E.; LARSSON, L.; ABERG, J. *A Market Neutral Statistical Arbitrage Trading Model*, Master Thesis, Estocolmo: Stockholm School of Economics, 2003.

LAZO LAZO, J. G. *Sistema Híbrido Genético-Neural para Montagem e Gerenciamento de Carteiras de Ações*, Dissertação de Mestrado, Rio de Janeiro: PUC-RJ, 2000.

LEIGH, W.; PAZ, M.; PURVIS, R. *An analysis of a hybrid neural network and pattern recognition technique for predicting short-term increases in the NYSE composite index*, In: Omega, Volume 30, Issue 2, pp. 69-76, Abril 2002.

LESAGE, J. P. *Applied Econometrics using MATLAB*, Toledo : Department of Economics, University of Toledo, Outubro 1999. Disponível em: www.spatial-econometrics.com/html/doc.html

LIAO, S.-H.; WEN, C.-H. *Artificial neural networks classification and clustering of methodologies and applications – literature analysis from 1995 to 2005*, In: Expert Systems with Applications. Volume 32, Issue 1, pp. 1-11, Janeiro 2007.

LOWENSTEIN, R. *When genius failed : the rise and fall of Long-Term Capital Management*, New York: Random House, 2000.

MACKINNON, J.G. *Critical values for cointegration tests*. In: Long-run Economic Relationships: Readings in Cointegration, Ch. 13, pp. 267-276, Oxford: Oxford University Press, 1991.

MADDEN, M.; O'CONNOR, N. *A neural network approach to predicting stock exchange movements using external factors*, In: Knowledge-Based Systems Volume 19, Issue 5, pp. 371-378, Setembro 2006.

MILLER, M.; MUTHUSWAMY, J.; WHALEY, R.E. *Mean Reversion of Standard & Poor's 500 Index Basis Changes: Arbitrage-Induced or Statistical Illusion?*, In: The Journal of Finance, Volume 49, No. 2, pp. 479-513, Junho 1994.

MOTIWALLA, L.; WAHAB, M. *Predictable variation and profitable trading of US equities: a trading simulation using neural networks*, In: Computers & Operations Research, Volume 27, Issues 11-12, pp. 1111-1129, Setembro 2000.

NEAL, R. *Direct Tests of Index Arbitrage Models*, In: The Journal of Financial and Quantitative Analysis, Volume 31, No. 4, pp. 541-562, Dezembro 1996.

NICHOLAS, J. G. *Market-Neutral Investing : Long/Short Hedge Fund Strategies*, Princeton, New Jersey: Bloomberg Press, 2000.

PARREIRAS, L. P. *Modelo Genético-Neural de Gestão de Carteiras de Ações*, Trabalho de Formatura, São Paulo: Escola Politécnica da Universidade de São Paulo, 2003.

PÉREZ-RODRÍGUEZ, J.V.; TORRA, S; ANDRADA-FÉLIX, J. *STAR and ANN models: forecasting performance on the Spanish "Ibex-35" stock index*, In: Journal of Empirical Finance, Volume 12, Issue 3, pp. 490-509, Junho 2005.

POUNDSTONE, W. *Fortune's Formula: the untold story of the scientific betting system that beat the casinos and Wall Street*, New York: Hill and Wang, 2005.

REFENES, A-P. N. (Ed.) *Neural Networks in the Capital Markets*, Chichester: John Wiley & Sons, 1995.

REFENES, A-P. N.; BURGESS, A. N.; BENTZ, Y. *Neural Networks in Financial Engineering: A Study in Methodology*, In: IEEE Transactions on Neural Networks, Volume 8, Issue 6, pp. 1222-1267, Novembro 1997.

REFENES, A-P. N.; HOLT, W. T. *Forecasting Volatility with Neural Regression: A Contribution to Model Adequacy*, In: IEEE Transactions on Neural Networks, Volume 12, Issue 4, pp. 850-864, Julho 2001.

REFENES, A-P. N.; ZAPRANIS, A.D. *Neural Model Identification, Variable Selection and Model Adequacy*, In: Journal of Forecasting, Volume 18, Issue 5, pp. 299-332, Setembro 1999.

REFENES, A-P. N.; ZAPRANIS, A.D. *Principles of Neural Model Identification, Selection and Adequacy: With Applications in Financial Econometrics*, Londres: Springer-Verlag, 1999.

REZENDE, S. O. (Org.) *Sistemas Inteligentes: fundamentos e aplicações*, Barueri, São Paulo: Manole, 2003.

ROCHE, B. B.; ROCKINGER, M. *Switching Regime Volatility : An Empirical Evaluation*, In: DUNIS, C.; LAWS, J.; NAÏM, P. *Applied quantitative methods for trading and investment*, pp. 193-211, Chichester, West Sussex: John Wiley & Sons: 2003.

SCHWAGER, J. D. *Market Wizards: Interviews with Top Traders*, New York: Harper Business, 1993.

SHADBOLT, J.; TAYLOR, J. G. (Eds.) *Neural Networks and the Financial Markets – Predicting, Combining and Portfolio Optimization*, Londres: Springer-Verlag, 2003.

SORNETTE, D. *Why Stock Markets Crash: Critical Events in Complex Financial Systems*, Princeton, New Jersey: Princeton University Press, 2003.

STEFANINI, F. *Investment strategies of hedge funds*, Chichester, West Sussex: John Wiley & Sons: 2006.

TALEB, N. N. *Fooled by Randomness : The hidden role of chance in the markets and in life*, New York: Texere, 2001.

TEO, M.; TSE, Y.K; WARACHKA, M. *Robust Tests of Market Efficiency using Statistical Arbitrage*, Paper No. 12-2004, SMU Economics & Statistics Working Paper Series, Cingapura: Singapore Management University, 2004.

TOWERS, N.; BURGESS, A.N. *Implementing Trading Strategies for Forecasting Models*, In: ABU-MUSTAFA, Y.S. *et al* (Eds.). *Computational Finance 1999*, pp. 313-326, Cambridge, Massachusetts: The MIT Press, 2001.

THORP, E.. *A perspective on quantitative finance: Models for beating the market*. In: The Best of Wilmott 1: Incorporating the Quantitative Finance Review, pp. 33–38. Chichester, West Sussex: John Wiley & Sons: 2005.

TSAIH, R.; HSU, Y.; LAI, C.C. *Forecasting S&P 500 stock index futures with a hybrid AI system*, In: Decision Support Systems, Volume 23, Issue 2, pp. 161-174, Junho 1998.

VIDYAMURTHY, G. *Pairs Trading – Quantitative Methods and Analysis*, New York: John Wiley & Sons, 2004.

WEST, D.; DELLANA, S.; QIAN, J. *Neural network ensemble strategies for financial decision applications*, In: Computers & Operations Research, Volume 32, Issue 10, pp. 2543-2559, Outubro 2005.

YU, L.; WANG, S.; LAI, K. K. *A novel nonlinear ensemble forecasting model incorporating GLAR and ANN for foreign exchange rates*, In: Computers & Operations Research, Volume 32, Issue 10, pp. 2523-2541, Outubro 2005.

YÜMLÜ, S.; GÜRGEN, F.S.; OKAY, N. *A comparison of global, recurrent and smoothed-piecewise neural models for Istanbul stock exchange (ISE) prediction*, In: Pattern Recognition Letters, Volume 26, Issue 13, pp. 2093-2103, Outubro 2005.

ZEKI-SUŠAC, M. *Neural Networks in Investment Profitability Predictions*, Dissertação de Doutorado, Zagreb: Faculty of Organization and Informatics Varaždin, University of Zagreb, 1999.

ZHANG, D.; ZHOU, L. *Discovering Golden Nuggets: Data Mining in Financial Application*, In: IEEE Transactions on Systems, Man, And Cybernetics - Part C: Applications and Reviews, Volume 34, Issue 4, pp. 513-522, Novembro 2004.

ZHANG, G.; PATUWO, B. E.; HU, M.Y. *Forecasting with artificial neural networks: The state of the art*, In: International Journal of Forecasting, Volume 14, Issue 1, pp. 35-62, Março 1998.

ZIVOT, E.; WANG, J. *Modelling Financial Times Series with S-Plus*, New York: Springer-Verlag, 2003.

APÊNDICE A - DETALHAMENTO DOS CÓDIGOS

A1. Metodologia de Construção de Cestas de Ações

Esta função implementa a metodologia de construção de "*mispricings*" do item 4.3.1 da dissertação. Toma como entrada uma base de dados de preços de ações, e para cada uma delas ela constrói o "*mispricing*", retornando o índice dos componentes, pesos, a série temporal, e quais são cointegrados. O código contém um *loop* onde são feitos testes para determinar qual o componente com maior correlação residual com o "*mispricing*", e uma vez encontrado, ele entra em uma regressão dentro da metodologia "*stepwise*". Ao fim todas as séries construídas passam por um teste *ADF* com nível de crítico de 5%.

```
function [indices, pesos, mispricingts, cointeg] = mispricing(basedados)

%metodologia para construção dos portfólios de arbitragens estatísticas
%loop de construção dos mispricings
for i = 1:length(basedados(1,:))
    portindex(i,1) = i;                     %guardo o índice do target da vez
    target = basedados(:,i);                %pego a serie temporal do target
    reg.resid = target - mean(target);      %inicializa os resíduos
    for j = 1:4                             %inicio o loop de construção do portfólio
        correl = corr(reg.resid, basedados);        %calcula correlação
        [scorrel, sidx] = sort(correl);             %classifica as correlações
        scorrel = scorrel(end:-1:1);                %ordem decrescente
        sidx = sidx(end:-1:1);                      %inverte os índices também
        h = 1;
        while h<=4                                  %rotina de teste
            ind = sidx(h);                          %coloco o primeiro índice
            if isempty(find(ind==portindex(i,:)))   %testo para ind no vetor
                portindex(i,j+1) = ind;
                regressor(:,j) = basedados(:,portindex(i,j+1));
                h=5;                        %garanto saída do teste
            else
                h = h+1;                    %se achar o ind, volta e recomeça
            end
        end
        reg = ols(target, regressor);    %faço regressão
    end
    reg = ols(target, regressor);        %calculo uma regressão final
    portweight(i,:) = [1 -reg.beta(:)']; %guardo os pesos finais
end

%saídas do programa
%os índices dos componentes, os pesos, e as series dos mispricings
indices = portindex;
pesos = portweight;
for aux = 1:length(basedados(1,:))
    mispricingts(:,aux) = basedados(:,portindex(aux,:))*portweight(aux,:)';
end

%precisamos testar quais mispricings são estacionários
for u = 1:length(mispricingts(1,:))
    res = adf(mispricingts(:,u),0,1);               %teste ADF com um lag
    testecoint(u) = res.adf;                        %guarda a estatística
end

%testa quais series cointegram @ 5%
cointeg = find(abs(testecoint)>=abs(res.crit(2)));
```

A2. Arbitragem Estatística Implícita

Este *script* implementa a estratégia de arbitragem estatística implícita, conforme descrito no item 4.3.3 da dissertação. Usando a função descrita em A1, inicialmente são construídos os "*mispricings*", depois construímos as séries temporais no período fora da amostra, e a partir daí temo um *loop* que simula as posições e os retornos da estratégia, uma para cada um dos modelos que são cointegrados. Esse *loop* utiliza uma função chamada "*isasimul*", cujo código aparece após o código do *script* (por questão de clareza não a colocamos antes). Esta função implementa as fórmulas 4.9, 4.10 e 4.11, onde determinamos o tamanho da posição a ser tomada a cada momento, e calculamos os retornos, tanto marginal quanto cumulativo, dessas posições, dentro de uma estratégia de arbitragem estatística implícita. Usando a saída da função, o *script* calcula todas as estatísticas relevantes das simulações, e termina construindo o portfólio médio com todas as estratégias, para ao fim termos um gráfico dos retornos deste portfólio (vide Figura 4.6).

```
%script para testar a estratégia de arbitragem estatística implícita
%ISA = implicit statistical arbitrage

%vou formar os mispricings com a parte insample dos dados
[indices, pesos, mispricingts, cointeg] = mispricing(insample);

%tenho que calcular o mispricing no periodo outsample
for col = 1:length(outsample(1,:))
    for line = 1:length(outsample(:,1))
        mispricingoutsample(line, col)=outsample(line, indices(col,:))*pesos(col,:)';
    end
end

%vamos calcular os retornos das estratégias de arb.estat.implícita
for varaux = 1:length(cointeg)
    w = cointeg(varaux);      %qual mispricing cointegrado vou simular
    k = 1;                    %parametro de calibragem do trading
    c = 0.0025;               %custo de transaçao
    ret(:,varaux) = isasimul(mispricingoutsample(:,w), insample(:,w), k, c);
    desvio(varaux) = std(ret(:,varaux));
    sharpe(varaux) = ret(length(ret),varaux)/desvio(varaux);
end

retornofinal = ret(length(ret),:);

%vamos calcular as estatisticas dos retornos
media = mean(retornofinal)
mediana = median(retornofinal)
[maximo, indmaximo] = max(retornofinal)
indmaximo = cointeg(indmaximo)
[minimo, indminimo] = min(retornofinal)
indminimo = cointeg(indminimo)

%retorno de um portfólio igualmente balanceado entre todas as estratégias
portfoliomedio = mean(ret,2);
volportfoliomedio = std(portfoliomedio);
plot(portfoliomedio, 'b');
grid on
title('Retorno do Portfólio de Arbitragens');
```

```
function retorno = isasimul(mispricingoutsample, targetoutsample, k, custo)
%outsample são os dados da serie temporal do mispricing cujo trading vamos simular
%k é o parametro que calibra o tamanho do investimento

mis = mispricingoutsample;
target = targetoutsample;

isa(1) = mis(1);
isaret(1) = 0;
isaprof(1) = 0;

for t = 2:length(mis)
    isa(t) = -sign(mis(t-1))*(abs(mis(t-1))^k);
    changeisa(t) = isa(t) - isa(t-1);
    delta(t) = mis(t) - mis(t-1);
    isaret(t) = isa(t)*delta(t)/(2*target(t-1) - mis(t-1)) - custo*abs(changeisa(t));
    isaprof(t) = isaprof(t-1) + isaret(t);
end

retorno = isaprof;
```

A3. Teste para escolha do número de neurônios na camada escondida

Este *script* é basicamente uma grande repetição de simulações de treinamentos de rede neural, para determinarmos qual é o número de neurônios que obtém o melhor resultado em termos de performance preditiva. Depois de construir os "*mispricings*", construímos as séries de *inputs* e *targets* para treinar a rede neural, e passamos para a função "*mispricing_fcast*", que se encarrega de treinar a rede neural, separando um pedaço dos dados de entrada para um teste fora da amostra, e retorna o R^2 de uma regressão entre as previsões e os dados reais para aquela rede. Essa função "*mispricing_fcast*" é praticamente igual à função "*neural_train*" que apresentamos no item A4, com a pequena diferença deste pedaço de código onde separamos um pedaço dos dados para um teste fora da amostra, e rodamos uma regressão para aferir a qualidade das previsões da rede neural. Não a colocamos aqui por restrições de espaço. Ao fim do *script*, calculamos o número de neurônios que gerou o melhor resultado para cada um dos "*mispricings*" e calculamos sua média. Para nos protegermos do caráter estocástico das redes neurais, ou seja, evitarmos que o resultado seja apenas sorte, repetimos o experimento cinqüenta vezes em um grande *loop* que atravessa todo o *script*, e ao fim calculamos o resultado médio para todos esses experimentos (vide Figura 4.13).

```
%script p/ testar previsão do mispricing com rede neural
%a função desse script é determinar um numero quasi-ótimo de
%neurônios na camada escondida das redes neurais p/ previsão de mispricings

%vou formar os mispricings com a parte insample dos dados
[indices, pesos, mispricingts, cointeg] = mispricing(insample);

%formacao dos inputs e targets a partir dos mispricings aptos

for bestone = 1:50
for k = 1:length(cointeg)
```

```
    aux = cointeg(k);               %pego o indice mispricing cointegrado
   %guardo a serie temporal do mispricing como input
    inputs = mispricingts(1:length(mispricingts)-1,aux);
    %concateno em 2D os outros fatores como input
    inputs = cat(2,inputs,fatores(1:length(inputs(:,1)),:));
    %desloco a serie temporal um dia para o target
    targets = mispricingts(2:length(mispricingts),aux);

    %passar para a funcao de train neural, tem que transpor as matrizes
    inputs = inputs';
    targets = targets';

    %loop para achar o melhor número de neurônios na camada escondida
    for i = 2:15
        escond = i; %numero de neurônios na camada escondida
        %treina a rede
        [result reg(i-1,k)] = mispricing_fcast(inputs, targets, escond);
        redes(k,i-1) = reg(i-1,k);          %guarda o R2 da rede treinada
    end
end

%mostra os resultados em termos de R2 para cada número de neurônios
[r2 indicemax] = max(redes');       %pega os R2 máximos de cada mispricing
final(bestone) = mean(indicemax+1); %guarda o numero de neurônios médio máximo

end

hist(final);
title('Histograma para escolha do número de neuronios');
mean(final)
```

A4. Algoritmo de Treinamento da Rede Neural

A função "*neural_train*" implementa o treinamento de uma rede neural, e ao final retorna a previsão um passo adiante. Toma como entrada *inputs* e a *targets*, além do número de neurônios da camada escondida. Os *targets* contêm a informação que desejamos que a rede neural aprenda. São realizados os passos de pré-processamento detalhados no item 4.4.2 da dissertação, e são separados os cinqüenta últimos dados da série de *inputs* para serem utilizados como validação, garantindo que a rede tenha boa capacidade de generalizar. Depois disso a rede é inicializada e o treinamento é realizado, utilizando um algoritmo Levenberg-Marquardt (mudar o algoritmo é trivial). Depois disso a rede é simulada para determinarmos todas as suas previsões, que são colocadas na média e variância originais, e ao fim a função retorna a última dessas previsões, que corresponde à previsão para o dia seguinte.

```
function previsao = neural_train(inputs, targets, hidden)

p = inputs;
t = targets;
h = hidden;

%Normalização das entradas e targets para media zero e variância unitária
[pn, meanp, stdp, tn, meant, stdt] = prestd(p,t);

%Realizar PCA e deixar apenas componentes
%responsáveis por mais de 0.1% da variação
[ptrans, transMat] = prepca(pn, 0.001);
```

```
%Dividir os dados em treinamento e validação
%A validação será feita com os 50 últimos dados
[R,Q] = size(ptrans);
iitr = 1:Q-50;
iival = Q-50:Q;
ptr = ptrans(:,iitr);
ttr = tn(:,iitr);
validation.P = ptrans(:,iival);
validation.T = tn(:,iival);

%Definir a Rede Neural
%Algoritmo de Treinamento Levenberg-Marquardt 'trainlm'
net = newff(minmax(ptr), [h 1], {'tansig' 'purelin'}, 'trainlm'); %cria a rede
net.trainParam.show = 100;
[net,tr] = train(net, ptr, ttr, [], [], validation);              %treinamento

%Simular a rede treinada.
%Converter os resultados para media e variância originais.
an = sim(net, ptrans);
a = poststd(an, meant, stdt);
previsao = a(Q);
```

A5. Arbitragem Estatística Condicional

Este *script* implementa a simulação de estratégias de arbitragem estatística condicional, como descrito no item 4.4.4 da dissertação. Procede de maneira similar ao *script* de arbitragem implícita, construindo os "*mispricings*", calculando-os fora da amostra, e fazendo um *loop* para simular sua operação. A diferença aqui é que no meio deste *loop* temos um outro *loop* que incorpora o treinamento neural, obtendo previsões um passo adiante, que vão alimentar a tomada de posição da estratégia. Note-se que para cada passo no tempo vamos treinar novamente as redes com todos os dados disponíveis até aquele momento, e além disso repetimos esse treinamento cinco vezes para garantir consistência nos resultados finais. Ao término das simulações, temos a construção do portfólio médio com todos os modelos. Por fim, obtemos várias figuras (estes códigos foram colocados para facilitar o exame de determinadas características dos resultados): (i) curva dos retornos acumulados de uma estratégia de arbitragem estatística condicional; (ii) comparação da evolução temporal do "*mispricing*" com a curva de retornos acumulados; (iii) comparação entre previsões e realizações para um "*mispricing*" e por fim (iv) a curva de retornos do portfólio médio.

```
% script simula estratégias de arbitragem estatística condicional
%CSA = conditional statistical arbitrage
%ele toma como dado as series fatores, insample e outsample

%parâmetros iniciais
k = 1;
custo = 0;
runs = 5;


%formar os mispricings com a parte insample dos dados
[indices, pesos, mispricingin, cointeg] = mispricing(insample);
```

```
%calcular o mispricing no período outsample a partir dos pesos
for col = 1:length(outsample(1,:))
    for line = 1:length(outsample(:,1))
        mispricingout(line, col) = outsample(line, indices(col,:)) * pesos(col,:)';
    end
end

mis = [mispricingin; mispricingout];

for contsimul=1:length(cointeg)

    aux = cointeg(contsimul);
    Tvar = [precos(:,aux)];
    [Lin, Win] = size(mispricingin);
    [Lout, Wout] = size(mispricingout);
    csa = zeros(Lin+Lout,1);
    csaret = zeros(Lin+Lout,1);
    csaprof = zeros(Lin+Lout,1);

    for tempo = Lin:(Lout+Lin-1)
        input = [mis(1:tempo, aux) fatores(1:tempo,:)]';
        target = mis(2:tempo, aux);
        target(length(target)+1,:) = target(length(target),:);
        target = target';

        for contador = 1:runs
            prev(contador) = neural_train(input, target, 8);
        end
        previsao(tempo) = mean(prev);

        expecdelta(tempo) = previsao(tempo) - mis(tempo, aux);
        csa(tempo)= sign(expecdelta(tempo))*(abs(expecdelta(tempo))^k);
        changecsa(tempo) = csa(tempo) - csa(tempo-1);
        valorport(tempo) = 2*Tvar(tempo) - mis(tempo, aux);
        delta(tempo+1) = mis(tempo+1, aux) - mis(tempo, aux);
        csaret(tempo+1) = csa(tempo)*delta(tempo+1)/(valorport(tempo)) -
custo*(abs(changecsa(tempo)));
        csaprof(tempo+1) = csaprof(tempo) + csaret(tempo+1);
    end

    retorno(:,contsimul) = csaprof(:,1);

end

retorno(length(retorno),:)

portfoliomedio = mean(retorno,2);
volportfoliomedio = std(portfoliomedio);

figure(1);
plot(csaprof(Lin:length(csaprof)),'r');
grid on;

figure(2);
plotyy(1:744,mis(1:744,aux),1:744,csaprof(1:744,1));
grid on;

figure(3);
plot(mis(500:744,aux),'b');
hold on;
plot(previsao(499:743),'r');
grid on;
hold off;

figure(4)
plot(portfoliomedio(498:744), 'b');
grid on
title('Retorno do Portfólio de Arbitragens Condicionais');
```

A6. Modelo Final

Este *script* incorpora todos os aprimoramentos discutidos ao longo Capítulo 4, como regra de *trading* por média móvel exponencial, política de *stop-loss*, alavancagem e re-investimento nos melhores modelos. Aqui temos o código que gerou os resultados finais apresentados no item 4.5.3. O que é diferente aqui é o cálculo do retorno do portfólio final: no *loop* de simulação da estratégia de arbitragem estatística, existe um teste *if* para ver se atingimos um ponto de *stop-loss*, e se sim o modelo é parado e vamos simular o próximo. Ao fim, pegamos os retornos efetivos (ou seja, o retorno após *stop-loss* passa a ser zero para um dado modelo), determinamos os pontos em que houve o *stop*, e calculamos o retorno do portfólio apenas considerando aqueles modelos que ainda estavam na amostra. Assim, garantimos automaticamente o re-investimento. Ao fim o código plota tanto o retorno médio quanto o retorno efetivo (com re-investimento) para os portfólios.

```
%esse script simula o modelo final
%ele toma como dado no as séries fatores, insample e outsample

%parâmetros iniciais
k = 1;
m = 20;
h = 5;
runs = 2;
stoploss = -0.25;
custo = 0.0025;

%formar os mispricings com a parte insample dos dados
[indices, pesos, mispricingin, cointeg] = mispricing(insample);

%calcular o mispricing no período outsample a partir dos pesos
for col = 1:length(outsample(1,:))
    for line = 1:length(outsample(:,1))
        mispricingout(line, col) = outsample(line, indices(col,:)) * pesos(col,:)';
    end
end

mis = [mispricingin; mispricingout];

for contsimul=1:length(cointeg)
    aux = cointeg(contsimul);
    Tvar = [precos(:,aux)];
    [Lin, Win] = size(mispricingin);
    [Lout, Wout] = size(mispricingout);
    previsao = zeros(Lin+Lout,1);
    csa = zeros(Lin+Lout,1);
    csaret = zeros(Lin+Lout,1);
    csaprof = zeros(Lin+Lout,1);

    for tempo = Lin:(Lout+Lin-1)
        input = [mis(1:tempo, aux) fatores(1:tempo,:)]';
        target = mis(2:tempo, aux);
        target(length(target)+1,:) = target(length(target),:);
        target = target';

        for contador = 1:runs
            prev(contador) = neural_train(input, target, 8);
        end
        previsao(tempo) = mean(prev);
```

```
        expecdelta(tempo) = previsao(tempo) - mis(tempo, aux);
        csa(tempo) = sign(expecdelta(tempo))*m*((abs(expecdelta(tempo)))^k);
        csama(tempo) = sum(csa(tempo-h+1:tempo))/h;
        changecsa(tempo) = csama(tempo) - csama(tempo-1);
        valorport(tempo) = 2*Tvar(tempo) - mis(tempo, aux);
        delta(tempo+1) = mis(tempo+1, aux) - mis(tempo, aux);
        csaret(tempo+1) = csama(tempo)*(delta(tempo+1)/(valorport(tempo))) -
custo*(abs(changecsa(tempo)));
        csaprof(tempo+1) = csaprof(tempo) + csaret(tempo+1);
        retornoefetivo(tempo+1,contsimul) = csaprof(tempo+1,1);
        if csaprof(tempo+1) < stoploss
            csaprof(tempo+1+1:Lout+Lin) = csaprof(tempo+1);
            retornoefetivo(tempo+1+1:Lout+Lin,contsimul) = 0;
            break
        end
    end
    retorno(:,contsimul) = csaprof(:,1);
end

retornofinal = retorno(length(retorno),:)

portfoliomedio = mean(retorno,2);
volportfoliomedio = std(portfoliomedio);
sharpemedio = portfoliomedio(length(portfoliomedio))/volportfoliomedio;

nz = (retornoefetivo(:,:)~=0);
for y=1:length(retornoefetivo(:,1))
    if (sum(nz(y,:))~=0)
        portfoliofinal(y,1) = sum(retornoefetivo(y,:))/sum(nz(y,:));
    else
        portfoliofinal(y,1) = 0;
    end
end
volportfoliofinal = std(portfoliofinal);
sharpefinal = portfoliofinal(length(portfoliofinal))/volportfoliofinal;

figure(1)
plot(portfoliomedio(498:744), 'b');
grid on;
title('Retorno Médio do Portfólio de Arbitragens Condicionais');

figure(2)
plot(portfoliofinal(498:744), 'r');
grid on;
title('Retorno Alavancado do Portfólio de Arbitragens Condicionais');
```