Algoritmos de negociação com dados de alta frequência

Akira Aricê de Moura Galvão Uematsu

DISSERTAÇÃO APRESENTADA AO INSTITUTO DE MATEMÁTICA E ESTATÍSTICA DA UNIVERSIDADE DE SÃO PAULO PARA OBTENÇÃO DO GRAU DE MESTRE EM ESTATÍSTICA

Área de Concentração: Probabilidade Orientador: Prof. Dr. Anatoli Iambartsev

- Durante o desenvolvimento desse trabalho, o autor recebeu apoio financeiro da CAPES –

São Paulo 2012

Algoritmos de negociação com dados de alta frequência

Este exemplar corresponde à redação final da dissertação devidamente corrigida e definida por Akira A. de M. G. Uematsu e aprovada pela comissão julgadora.

São Paulo, 20 de março de 2012

Banca Examinadora:

- Prof. Dr. Anatoli Iambartsev (orientador)
- Prof. Dr. Yuri M. Suhov
- Prof. Dr. Vladimir Belitsky

Em nosso trabalho analisamos os dados provenientes da BM&F Bovespa, a bolsa de valores de São Paulo, no período de janeiro de 2011, referentes aos índices: BOVESPA (IND), o mini índice BOVESPA (WIN) e a taxa de câmbio (DOL). Estes dados são de alta frequência e representam vários aspectos da dinâmica das negociações. No conjunto de valores encontram-se horários e datas dos negócios, preços, volumes oferecidos e outras características da negociação. A primeira etapa da tese foi extrair as informações necessárias para análises a partir de um arquivo em protocolo FIX, foi desenvolvido um programa em R com essa finalidade. Em seguida, estudamos o carácter da dependência temporal nos dados, testando as propriedades de Markov de um comprimento de memória fixa e variável. Os resultados da aplicação mostram uma grande variabilidade no caráter de dependência, o que requer uma análise mais aprofundada.

Acreditamos que esse trabalho seja de muita importância em futuros estudos acadêmicos. Em particular, a parte do carácter específico do protocolo FIX utilizado pela Bovespa. Este era um obstáculo em uma série de estudos acadêmicos, o que era, obviamente, indesejável, pois a Bovespa é um dos maiores mercados comerciais do mundo financeiro moderno.

Palavras chave: bolsa de valores, mercado de alta frequência, protocolo FIX, processos de Markov e algoritmos de negociação.

Algorithmic Trading with high frequency data - Abstract

In our work we analyzed data from BM&F Bovespa, the stock exchange in São Paulo. The dataset refers to the month January 2011 and is related to BOVESPA index (IND), mini BOVESPA index (WIN) and the exchange tax (DOL). These, are high frequency data representing various aspects of the dynamic of negotiations. The array of values includes the dates/times of trades, prices, volumes offered for trade and others trades characteristics. The first stage of the thesis was to extract information to the analysis from an archive in FIX protocol, it was developed a program in R with this aim. Afterwards, we studied the character of temporal dependence in the data, testing Markov properties of a fixed and variable memory length. The results of this application show a great variability in the character of dependence, which requires further analysis.

We believe that our work is of great importance in future academic studies. In particular, the specific character of the FIX protocol used by Bovespa. This was an obstacle in a number of academic studies, which was, obviously, undesirable since Bovespa is one of the largest trading markets in the modern financial world.

Key Words: stock exchange, high frequency market, FIX protocol, Markov process and Algorithmic trading.

Sumário

Introdução	0 5
1 Algoritmos de negociação, dados e o mercado financeiro 1.1 Microestrutura do mercado financeiro 1.2 Liquidez	07 08 09 09 09
2 Protocolo FIX	11
2.1 O que é protocolo FIX?	11
2.2 Decodificando o FIX Protocol	13
2.3 Programa de extração de dados 1	17
3 Análise estatística dos dados	18
3.1 Análise descritiva dos dados	18
3.2 Teste para propriedade Markoviana 1	19
3.2.1 Geração da sequencia	22
3.2.2 Resultados	23
4 Algoritmos e suas aplicações	25
4.1 Índice DOL	28
4.2 Índice IND 3	32
4.3 Índice WIN	35
5 Conclusões	39
Apêndice	41
A.1 Extração dos dados	41
A.2 Marcação da série discreta	
A.3 Cálculo da estatística de teste	
A.4 Algoritmo de negociação	
Bibliografia	53

Introdução

Este trabalho tem como base principal o estudo de algoritmos de negociação com dados de alta frequência, suas aplicações e o desenvolvimento de técnicas na obtenção de dados à partir de arquivos no formato do protocolo FIX.

O problema dos Algoritmos de Negociação é de muita importância, já que são utilizados por diversos fundos de investimento nas bolsas de valores ao redor do mundo, como nos Estados Unidos, Inglaterra, e também no Brasil. Um exemplo deles é o Winton Capital Management, um grande fundo de investimento situado em Londres. Os dados em que baseamos nossos estudos foram obtidos na Bolsa de Valores de São Paulo a BOVESPA. Estudamos o desenvolvimento de um novo algoritmo que visa gerar lucros em uma eventual aplicação no mercado, baseado em análises estatísticas dos dados e averiguando se o conjunto dos mesmos segue a propriedade Markoviana. Tal conjunto contempla um período de tempo muito amplo, além de diversos índices que compõe a Bolsa de Valores de São Paulo. Eles também possuem todas as grandezas envolvidas nas negociações como, por exemplo, preços de venda e de compra, lotes comprados, etc.

Fundos de investimento ou bancos de médio e grande porte gerenciam portfólios com centenas de posições em qualquer tempo dado. O banco (ou o fundo) monitoram os dados do mercado, computadores atualizam os valores dos chamados "osciladores" indicando quais posições devem ser adquiridas. Valores de Risco (VAR) e outros valores são constantemente avaliados. No momento, na Winton Capital Management (WCM) os algoritmos de negociação de alta frequência são responsáveis por uma porcentagem relativamente pequena dos negócios. Eles possuem um protótipo em funcionamento que ainda está em teste, mas a economia atual pode não estar em um bom momento para mudanças drásticas no estilo dos negócios. Em geral, o que ocorre na prática é que os fundos de investimentos, tanto os pequenos quanto os grandes, antes de começarem a usar um novo modelo de algoritmo, fazem um teste de eficiência a partir de uma amostra de dados da seguinte forma – uma fração da amostra é utilizada para a estimação de parâmetros considerados adequados para o algoritmo, com a fração restante, o algoritmo é validado simulando seu desempenho. Sendo o algoritmo aprovado nessas etapas de teste, o mesmo é colocado no mercado, no entanto, o que pode ocorrer muitas vezes na prática é que mais de um fundo de investimento utilize um mesmo tipo de algoritmo, o que leva o modelo a ficar estagnado e não gerar mais lucros, fazendo assim com que os fundos busquem novas alternativas de automatização.

O líder mundial em negociação com algoritmos de alta-frequência é o Renaissance Hedge Fund em Nova Iorque (fundado por Jim Simons). Seus negócios são quase que exclusivamente de alta-frequência e algorítmico. Outros fundos, incluindo o WCM tentam replicar as aproximações adotadas pelo Renaissance.

Criar, testar e colocar em operação um sistema algorítmico de negociação é uma tarefa desafiadora. Será necessário um suprimento contínuo e grande com conjuntos completos de dados do mercado, os chamados Tick Data. A WCM recebe conjuntos de dados razoavelmente precisos do FTSE The Index Company – companhia independente de propriedade conjunta do The Financial Times e da London Stock Exchange. Objetiva o manejo dos índices e serviços correlatos de dados, em escala internacional, além do Reino Unido – mas esses dados necessitam ainda de muito trabalho para coloca-los no formato ideal para a manipulação. O designer de algoritmos requer habilidades especiais, particularmente para grandes portfolios, podem ser usadas para compor esses algoritmos, ou suas partes, ou uma grande variedade de idéias no domínio da pesquisa avançada, como algoritmos genéticos da biosciência, cadeias de Markov de reconhecimento de fala e assim por diante.

Mas não importa o quão bons sejam os algoritmos, eles sempre necessitarão de testes e atualizações. Os mercados estão em constante mudança e conquistas passadas não são indicativos de quaisquer sucessos futuros.

Um problema constante é lidar com as chamadas "slippage", incluindo o custo transacional. "Slippage" significa que o preço atual da transação difere do que foi gravado. Vários fatores contribuem aqui, um deles está relacionado às atividades dos predadores do mercado (investidores que visam lucros altíssimos). Até agora nenhuma solução satisfatória foi encontrada para esse problema, mas estudos vêm sendo realizados com esse intuito. Será possível ver que o processo de negociação automatizado é muito rápido e muito complexo, esse processo abre as portas para que os chamados predadores façam muito dinheiro, e geralmente é um investidor institucional quem paga o preço disso. Já ocorreu que a WCM, mencionada acima, estava perdendo dinheiro (não muito) ao aplicar uma estratégia de baixa-frequência em um mercado de alta frquência.

A tese será desenvolvida da seguinte forma, no primeiro capítulo do trabalho serão descritos alguns termos essenciais para a compreensão de como funciona o mercado financeiro, além de um panorama sobre algoritmos de negociação e bolsa de valores. No segundo capítulo apresentamos o protocolo FIX e o algoritmo de transformação dos dados em uma forma apropriada para as análises estatísticas. Em seguida começaremos a descrição dos dados, apresentando alguns gráficos como, por exemplo, o que representa a dinâmica dos preços de venda e de compra onde podemos identificar os negócios nos pontos onde as linhas, que representam a evolução dos dois preços, se encontram, além de investigar o tipo de comportamento que os dados possuem. No quarto capítulo é apresentado o algorítmo, desenvolvido por nós, e o resultado que ele gerou com os dados da BM&F e por último algumas conclusões que poderemos tomar.

Algoritmos de Negociação, dados e o Mercado Financeiro

O mercado financeiro nada mais é do que um mecanismo que possibilita que pessoas comprem e vendam determinados instrumentos financeiros a baixo custo e a preços que reflitam a hipótese de eficiência de mercado, que diz que o mercado financeiro é eficiente em termos de informação, ou seja, um negociante não pode alcançar um retorno acima do retorno médio do mercado, tendo em conta o ajuste da exposição ao risco, dada a informação pública disponível no momento em que o investimento é feito. Como exemplo de instrumentos financeiros temos ações, commodities (que são os bens como metais preciosos ou produtos da agricultura – milho, arroz, etc.), entre outros itens que tenham algum valor financeiro agregado.

São necessárias algumas definições, comuns no universo das finanças, importantes para o entendimento desse trabalho. Passemos adiante a elas.

1.1.Microestrutura do mercado financeiro

De acordo com [4] microestrutura de mercado é o estudo dos mecanismos de negociação usados para instrumentos financeiros tais como as commodities. O "National Bureau of Economic Research" – organização de pesquisa privada americana, sem fins lucrativos, que visa disseminar pesquisas econômicas entre o público, profissionais do mercado e a comunidade acadêmica e que ficou conhecida ao estimar as datas de início e fim das recessões dos Estados Unidos - tem um grupo de pesquisa sobre microestrutura de mercado, que realiza estudos teóricos, empíricos e experimentais na economia de mercados financeiros, incluindo o papel da informação no processo de descoberta dos preços das commodities. Também realizam estudos sobre liquidez e custos de transação com suas implicações na eficiência e regulagem de mecanismos de negociação alternativos e estruturas de mercados.

1.2.Liquidez

"Mercados financeiros podem ser caracterizados pela sua liquidez" [4]. Diferentemente do que ocorre em ciências exatas a definição desse termo pode variar de acordo com modelos particulares, mas, algumas qualidades relacionadas a esse termo são em geral aceitas e compreendidas. Por exemplo, em um mercado líquido, mudanças nos preços não são causadas por leves variações no fornecimento ou demanda de um determinado produto.

Muitas vezes associa-se liquidez aos corretores, negociantes e outros intermediários que trabalham nas bolsas, quando na verdade, sob esse aspecto, a liquidez está relacionada aos indivíduos e instituições investidoras que possuem necessidade de negociar.

No contexto do trabalho que iremos realizar, é fundamental que se possa entrar e sair de posições rapidamente (uma posição em termos de finanças é um comprometimento entre as duas partes do negócio em vender ou comprar alguma coisa). Para que isso ocorra é necessário que exista liquidez do mercado. Um ativo (bens, valores) será liquido se o mesmo possuir fornecimento prontamente disponível e demanda.

1.3.Bolsa de Valores

A primeira bolsa de valores de que se tem notícia foi criada por intermédio da Companhia Holandesa das Índias Orientais que instituiu e comercializou as primeiras ações. Isso ocorreu em Amsterdã por volta de 1600.

Muitas mudanças ocorreram desde então, porém, o princípio, permanece o mesmo, transações de mercadorias e valores. Ao contrário do que ocorria antigamente (negócios sendo realizados fisicamente no recinto da bolsa "pregão viva voz"), atualmente, as negociações, em geral, são feitas por meios eletrônicos em tempo real. Como exemplo disso temos a americana Nasdaq que somente opera através do pregão eletrônico.

É de responsabilidade das bolsas transmitir, aos seus investidores, através de diversos meios (como por exemplo, internet, jornais, revistas) informações sobre seus negócios diários, dados do mercado, etc. De maneira a manter a transparência de suas operações. No Brasil o responsável pela fiscalização da atividade das bolsas é a CVM ou Comissão de Valores Mobiliários.

Hoje em dia, na grande maioria, as bolsas de valores atuam como S/A visando lucro por meio de seus serviços. Seu patrimônio, quando falamos de associações civis, é composto por títulos pertencentes às sociedades corretoras que as compõe e no caso das Sociedades Anônimas esse patrimônio é composto por ações.

As operações executadas dentro de uma bolsa de valores devem sempre manter os padrões éticos de negociação, sendo que as mesmas devem sempre ser divulgadas com rapidez, amplitude e detalhes.

Temos a seguir exemplos de Bolsas de Valores ao redor do mundo:

- North American Securities Dealers Automated Quotation System (NASDAQ) e New York Stock Exchange (NYSE) ambas nos Estados Unidos, London Stock Exchange (LSE) no Reino Unido, Shangai Stock Exchange (SSE) na China, Bolsa de Valores de Tóquio (TSE) no Japão e Moscow Interbank Currency Exchange na Rússia.

Sendo que nesse trabalho iremos tratar dos dados provenientes da Bolsa de Valores de São Paulo (BM&F Bovespa).

Foi através de uma junção entre a Bolsa de Valores de São Paulo (BOVESPA) e a Bolsa de Mercadorias e Futuros, realizada em oito de maio de 2008, que surgiu a BM&F Bovespa. Sediada no centro de São Paulo é uma das Bolsas mais importantes do mundo em valor de mercado.

1.4.Dados de alta frequência

O mercado financeiro é a principal fonte de dados de alta frequência. Quando dizemos mercado financeiro, podemos pensar em alguns exemplos tais como a bolsa de valores, um mercado de taxa de câmbios, o mercado de comodities, etc. Os dados gerados nesse ambiente financeiro são obtidos em intervalos muito pequenos de tempo, em geral, em um intervalo irregularmente espaçado no tempo e em escala intra-diária. Os tempos decorridos entre cada observação são variáveis aleatórias (chamadas durações), entretanto, podem ocorrer também mais de uma observação em um mesmo instante de tempo. Os conjuntos de dados que possuem essas características são chamados "Tickby-tick", eles também trazem consigo outras informações que serão apresentadas e descritas no próximo capítulo, e esse será o tipo de dados utilizados no presente trabalho.

Um dos grandes desafios ao trabalhar com esse tipo de dados é a filtragem dos mesmos, já que podem ocorrer mais de uma negociação no mesmo instante de tempo além de eventuais erros de registros.

1.5.Protocolo FIX

Quando falamos da comunicação entre dois sistemas computacionais por intermédio de uma convenção ou padrão que possibilite uma conexão entre as mesmas, estamos falando de um protocolo. A BM&F Bovespa utiliza o FIX PROTOCOL para realizar suas negociações eletrônicas.

O Financial Information Exchange, protocolo FIX, nada mais é do que um conjunto de especificações técnicas para troca de mensagens eletrônicas relacionadas à negociação de valores na bolsa. Sendo esse um padrão aberto administrado por membros da FIX Protocol Ltd. Para esse trabalho, nos foi fornecido um conjunto de dados brutos dispostos no formato FIX. A partir dele nos foi necessário extrair os valores que nos seriam úteis para o trabalho, como o dia e horário das negociações, preço ASK, preço BID, volume negociado, preço final negociado, etc.

Não nos foi fornecido, para o trabalho, uma interface que permitisse a manipulação dos dados via FIX. Portanto, nos foi necessário desenvolver um programa para realizar a extração dos valores, como veremos no próximo capítulo.

1.6.Nomenclaturas

Citamos, no tópico anterior, algumas grandezas que serão o objeto principal de nosso estudo. Devemos entender o preço ASK, ou preço OFFER, como o preço que um vendedor irá aceitar por um bem, consequentemente o preço BID é o maior preço que um comprador está disposto a pagar por um bem. A diferença entre esses dois valores é chamada de SPREAD. E quando falarmos de volume estaremos nos referindo a quantidade de bens que estão sendo negociadas em um determinado momento.

1.7.Algoritmos

Mas o que são esses chamados algoritmos de negociação? Segundo [5] podemos vê-los como sistemas computacionais que possuem um conjunto preciso de regras que definem automaticamente, sem qualquer tipo de intervenção humana, a entrada e a saída do mercado (compra ou venda de determinado item e em qual quantidade). Como regras são precisas, não há nenhuma dúvida sobre quando e aonde aplica-las, isso faz com o que o sistema seja testável estatisticamente. Podemos então, sobre alguma análise estatística, fazer afirmações sobre como o algoritmo se comportou ou poderá se comportar no futuro com algum grau de confiança.

Dentro do universo dos algoritmos existem alguns que pertencem a uma classe especial, os chamados "high-frequency trading" que lidam com grandes quantidades de dados, onde computadores tomam decisões elaboradas, baseadas em informações que são recebidas eletronicamente, para seguirem determinados caminhos. Antes mesmo que negociadores humanos sejam capazes de processar a informação que observam. Esse tipo de sistemas gerou uma mudança drástica na microestrutura do mercado principalmente na maneira em que a liquidez é provida.

Protocolo FIX

O primeiro passo para o desenvolvimento do trabalho foi, como havia sido dito, a extração, a partir do arquivo bruto que nos foi entregue pela BM&F BOVESPA, dos dados apresentados no formato do protocolo FIX. Esta pode ser considerada a etapa de maior importância do trabalho dado que, sem os dados, nenhuma das análises que serão realizadas adiante seriam possíveis. Um espaço de tempo razoavelmente grande foi investido nessa etapa, já que, tivemos que trabalhar com um assunto de conhecimento restrito e que, portanto, não possuía muita literatura disponível a respeito.

Nesse capítulo iremos descrever alguns pontos importantes sobre esse protocolo para que, com isso, seja possível entender como o programa, desenvolvido por nós, faz a extração dos dados de uma maneira que se possa trabalhar com eles. Já que, o programa que nos foi disponibilizado pela BM&F (de nome FIX BOOK), só tornava possível a visualização dos dados e não sua manipulação.

2.1.0 que é Protocolo Fix

Segundo a própria organização criadora, o protocolo de troca de informações financeiras, FIX, é uma série de especificações para comunicação eletrônica de mensagens comerciais relacionadas. Foi desenvolvido com a colaboração de bancos, corretores de ações, mercados de intercâmbio, indústria de utilidades e associações, investidores institucionais e pessoas que trabalham com tecnologia da informação de todo o mundo. Esses participantes do mercado compartilham uma visão comum, uma linguagem global para o comércio automatizado de instrumentos financeiros.

FIX é aberto e grátis, mas não é um software, ao contrário, ele é um conjunto de especificações em cima da qual, programadores podem desenvolver programas de código aberto ou comerciais. Ninguém sabe ao certo quantas pessoas usam FIX ou quanto do volume global de transações comerciais é efetuado via esse protocolo. O que pode se dizer é que, virtualmente, a maioria das grandes bolsas de valores e bancos de investimento o utiliza. Então, desde o seu início em 1992, como um framework de comunicação bilateral para negociação de ações entre Fidelity Investments e Salomon Brothers, o protocolo FIX se tornou muito importante, entretanto, atualmente, a Bolsa de Valores de São Paulo possui planos de mudar o seu sistema eletrônico de negociação.

Em seguida podemos ver um pequeno esquema de seu funcionamento.

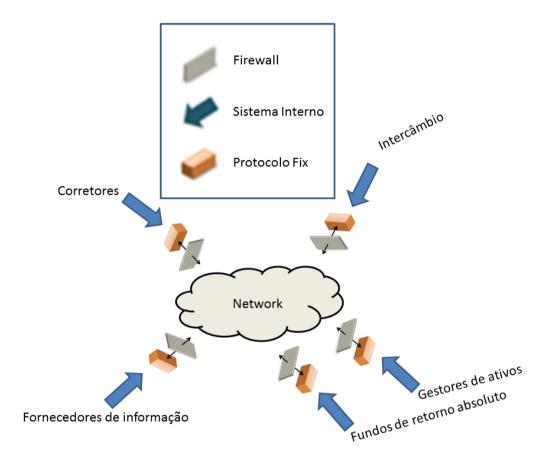


Figura 2.1.1 – mensagem no formato FIX

No exemplo acima, podemos ver a maioria dos componentes do mercado financeiro interagindo eletronicamente por meio do protocolo FIX em um dia de atividades. Cada uma das entradas, representando diversos ramos de atuação do mercado, utiliza o protocolo através de um firewall para chegar à rede.

2.2.Decodificando o FIX Protocol

Vejamos abaixo um exemplo de como as mensagens são trocadas entre os terminais eletrônicos. Basta uma chave de decodificação para que o conjunto numérico abaixo faça sentido pra quem o lê. Pois, cada número à esquerda representa o nome de um campo e cada número à direita seu respectivo valor.

[1] "8=FIX.4.4"	"9=463"	"35=X"
[4] "34=68797"		"52=20110124-12:07:11.878"
[7] "56=MD01A"	"10016=31480_INDG11_104	
[10] "268=3"	"279=0"	"269=0"
[13] "278=1275"	"55=INDG11"	"48=BMFBR3009171"
[16] "22=8"	"207=XBMF"	"270=69255"
[19] "271=5"	"272=20110124"	"273=12:07:11"
[22] "37=000469"	"288=BM000107"	"290=2"
[25] "279=0"	"269=Z"	"55=INDG11"
[28] "48=BMFBR3009171"	"22=8"	"207=XBMF"
[31] "270=69255"	"271=10"	"272=20110124"
[34] "273=12:07:11"	"346=2"	"290=1"
[37] "279=1"	"269=e"	"55=INDG11"
[40] "48=BMFBR3009171"	"22=8"	"207=XBMF"
[43] "270=69255"	"271=10"	"272=20110124"
[46] "273=12:07:11"	"346=2"	"290=1"
[49] "10=247"		

Figura 2.2.1 – uma mensagem típica no formato FIX

O arquivo cedido pela BM&F Bovespa era composto por conjuntos de dados, como esse acima, para cada tipo de produto negociado. Dentre os vários produtos negociados, escolhemos os seguintes para o desenvolvimento dos estudos:

- IND, WIN e DOL. O IND representa o índice BOVESPA que é um índice de mercado que mede o preço de uma carteira teórica composta pelas ações emitidas por companhias que correspondem por mais de 80% do número de negócios e do volume financeiro no mercado a vista. O WIN é o mini índice BOVESPA, mini índices são contratos negociados na BM&F exclusivamente via internet e representam uma fração dos contratos grandes (cheios) nas proporções determinadas pela BM&F, os tamanhos são sempre bem menores que os contratos futuros padrão de cada um dos produtos. O DOL é a taxa de câmbio de reais para dólar dos Estados Unidos, para entrega pronta, contratada nos termos da resolução do Conselho Monetário Nacional (CMN)

Dentro de uma mensagem FIX de um determinado produto, como, por exemplo, o índice WIN, podem haver várias outras sub mensagens. Vejamos um exemplo abaixo:

```
[1] "8=FIX.4.4"
                          "9=463"
                                                     "35=X"
                          "49=FIXGatewayDerivatives" "52=20110124-12:07:11.878"
[4] "34=68797"
[7] "56=MD01A"
                          "10016=31480 INDG11 1043 2" "75=20110124"
[10] "268=3"
                          "279=0"
                                                     "269=0"
                          "55=INDG11"
                                                     "48=BMFBR3009171"
[13] "278=1275"
[16] "22=8"
                          "207=XBMF"
                                                     "270=69255"
[19] "271=5"
                          "272=20110124"
                                                     "273=12:07:11"
[22] "37=000469"
                                                     "290=2"
                          "288=BM000107"
[25] "279=0"
                          "269=Z"
                                                     "55=INDG11"
[28] "48=BMFBR3009171"
                          "22=8"
                                                     "207=XBMF"
[31] "270=69255"
                          "271=10"
                                                     "272=20110124"
[34] "273=12:07:11"
                          "346=2"
                                                     "290=1"
                          "269=e"
[37] "279=1"
                                                     "55=INDG11"
                          "22=8"
                                                     "207=XBMF"
[40] "48=BMFBR3009171"
[43] "270=69255"
                          "271=10"
                                                     "272=20110124"
[46] "273=12:07:11"
                          "346=2"
                                                     "290=1"
[49] "10=247"
```

Figura 2.2.2 – identificação de sub mensagens no formato FIX

Destacamos com cores diferentes dinâmicas distintas e com isso também podemos explicar a função do campo número '268' que indica quantas negociações (entradas) seguem dentro da mensagem FIX. No exemplo acima vemos três (268=3). Passemos agora a detalhar cada chave utilizada no trabalho.

Primeiramente, temos o campo de número 55 que descreve qual tipo de produto está sendo negociado. No exemplo acima, podemos notar após o nome do produto (IND), uma letra "G" que representa, no caso, o mês de vencimento do contrato e, em seguida, vemos o ano de vencimento do mesmo (11). Na BM&F são utilizados os seguintes meses de vencimento para índices futuros: G (Fevereiro), J (Abril), M (Junho), Q (Agosto), V (Outubro) e Z (Dezembro).

Depois temos o campo número 8 que indica simplesmente a versão do FIX que está sendo utilizada. A versão em que os dados se encontram é a versão 4.4 que é também a mais recente.

O terceiro campo de importância para nos foi o de número 279 que indica o tipo de alteração que está sendo realizada. A seguinte classificação foi utilizada:



Figura 2.2.3 – valores assumidos pelo campo 279

Caso seja uma nova requisição ao sistema que estejamos fazendo, indicamos isso por meio do valor zero, para alterações nas ordens eletrônicas procedemos com o valor um e, para cancelamento de pedidos, entramos com o valor dois. No Brasil existe uma limitação da quantidade de entradas de pedidos e de cancelamentos que podemos fazer num determinado período de tempo, se ultrapassamos esse limite somos penalizados em um intervalo de tempo aonde não podemos acionar nenhuma ordem de compra no mercado eletrônico. Dizem os especialistas que isso garante a fluidez do mercado entre

outras coisas. Esse tipo de controle não é utilizado nas demais bolsas de valores do mundo.

O próximo campo é o de maior importância para nós, já que nos informa qual o tipo de entrada que a mensagem trata, os tipos podem ser os seguintes:

→ 269 MDEntryType	Y	Char	Type Market Data entry. Valid values: 0 = Bid 1 = Offer e = Aggregate bid f = Aggregate offer 2 = Trade 4 = Opening Price 5 = Closing Price 7 = Trading Session High Price 8 = Trading Session Low Price 9 = Trading Session VWAP Price B = Trade Volume X = Top of book – offer Z = Top of book – bid a = Referential prices b = Instrument trading phase c = Instrument state
-------------------	---	------	--

figura 2.2.4 – valores assumidos pelo campo 269

Quando estivermos diante de uma mensagem do tipo que contenha "269=0" sabemos que o que nos será informado é o preço BID do produto negociado. Por outro lado, quando temos "269=1" sabemos que o que segue é o preço ASK do mesmo. Esse dois últimos foram os tipos de dados que mais utilizamos no trabalho.

Em seguida temos o valor do campo 270. Esse campo é um complemento de valor do campo anterior, já que, ele somente existirá, quando o valor do campo 269 indicar um preço. Logo, em nosso exemplo acima, na primeira sub mensagem, temos uma nova entrada indicando que o preço BID do produto é 69255. Pois vimos, no campo anterior, que um preço seria descrito.

→	270	MDEntryPx	С	Price	Price of the Market Data Entry. Conditionally required when this market data entry involves a price. Represents the financial value for trade volume (B). Other entry types that do not involve price - such as trading phase (b) or instrument state (c) – do not require this tag.
----------	-----	-----------	---	-------	--

figura 2.2.5 – valores assumidos pelo campo 270

No campo 271 temos descrito a quantidade ou volume do que está sendo tratado na negociação. E, em seguida, nos campos 272 e 273 podemos extrair a data e hora em que a mensagem foi enviada, ou seja, quando o trade foi realizado.

→	271	MDEntrySize	С	Qty	Quantity or volume represented by the Market Data Entry. Conditionally required when MDEntryType = Bid (0), Offer (1), Aggregate bid (e), Aggregate offer (f), Trade (2), Top of Book Offer (X), Top of Book Bid (Z), Trade Volume (B) or Opening Price (4).
\rightarrow	272	MDEntryDate	Υ	UTCDateOnly	Date of Market Data Entry.
\rightarrow	273	MDEntryTime	Υ	UTCTimeOnly	Time of Market Data Entry.

figura 2.2.6 – valores assumidos pelos campos 271, 272 e 273

Por último, no campo 290 temos uma classificação, quando estamos falando de preços (BID e ASK), de competitividade do valor que está sendo ofertado. Por exemplo, podemos ter para uma mesma ação compradores querendo oferecer \$10 por ação enquanto outros possam estar ofertando \$20 criando assim um ranking das ofertas.

→	290	MDEntryPositionNo	N	Int	Display position of a bid or offer, numbered from most competitive to least competitive, per market side, beginning with 1. Conditionally required when MDEntryType=0,1, e or f (Bid, Offer, Aggregate bid or Aggregate offer).
----------	-----	-------------------	---	-----	---

figura 2.2.7 – valores assumidos pelo campo 290

2.3.Programa de extração dos dados

Com base nas informações apresentadas acima, foi possível montar um programa, desenvolvido em R, que realiza a extração dos dados à partir dos arquivos de dados brutos em protocolo FIX. Tal programa é apresentado no apêndice A.1.

O formato do arquivo de saída dos dados fica o seguinte:

data	time	bs5	bs4	bs3	bs2	bs1	bp5	bp4	bp3	bp2	bp1	op1	op2	ор3	op4	op5	os1	os2	os3	os4	os5	tp	ts
20110103	10:56:23,041	200	5	10	10	5	1650	1651	1652	1653	1655	1670	1675	1685	1700	NA	100	260	10	200	NA	NA	NA
20110103	10:56:23,861	5	10	10	5	500	1651	1652	1653	1655	1662	1670	1675	1685	1700	NA	100	260	10	200	NA	NA	NA
20110103	10:56:25,701	10	10	10	5	500	1652	1653	1654	1655	1662	1670	1675	1685	1700	NA	100	260	10	200	NA	NA	NA
20110103	10:56:27,981	10	10	10	5	500	1653	1654	1655	1655	1662	1670	1675	1685	1700	NA	100	260	10	200	NA	NA	NA
20110103	10:56:31,341	10	10	5	50	500	1654	1655	1655	1660	1662	1670	1675	1685	1700	NA	100	260	10	200	NA	NA	NA
20110103	10:56:39,941	10	5	5	50	500	1655	1655	1659,5	1660	1662	1670	1675	1685	1700	NA	100	260	10	200	NA	NA	NA
20110103	10:56:49,861	10	5	5	50	500	1655	1655	1659,5	1660	1662	1670	1675	1677	1685	1700	100	260	5	10	200	NA	NA
20110103	10:56:55,131	10	5	5	50	500	1655	1655	1659,5	1660	1662	1670	1675	1677	1685	1700	100	260	5	10	200	NA	NA
20110103	10:57:00,641	10	5	5	50	500	1655	1655	1659,5	1660	1662	1670	1675	1677	1685	1700	100	260	5	10	200	NA	NA

Figura 2.3.1 – exemplo arquivo de saída do programa de extração

Podemos observar os seguintes campos acima:

- (data), temos a data que indica o dia em que ocorreram as negociações na bolsa;
- (time), em seguida temos o instante em que é avaliado o negócio;
- (bs), essas próximas cinco colunas (bs) representam as cincos melhores quantidades à serem negociadas anunciadas pelos compradores (relativas ao produto em questão) que estão sendo negociadas;
- (bp), depois temos cinco colunas com os melhores preços BID que são os maiores preços que os compradores estão dispostos a pagar pelo produto negociado;
- (op), as cinco colunas seguintes contêm os menores preços OFFER ou ASK anunciados pelos vendedores, que indicam por quanto eles estão dispostos à vender cada produto;
- (os), indica as quantidades a serem negociadas, porém, anunciadas pelos vendedores:
- (tp), representa por quanto foi firmado o negócio;
- (ts), e por último qual quantidade negociada.

De posse dos dados passemos às análises estatísticas dos mesmos.

Análise Estatística dos dados

O principal ponto desse capítulo é o estudo do comportamento dos dados através do teste para a propriedade Markoviana dos mesmos. Iremos utilizar um teste não paramétrico com estimação das densidades utilizando o método de Kernel. Mas, primeiramente, apresentamos alguns gráficos que mostram como se dá a dinâmica da evolução dos preços para cada um dos índices considerados.

3.1. Análise gráfica dos dados

Abaixo temos a evolução ao longo do tempo dos preços para cada um dos índices, em um dia de atividade da BM&F.

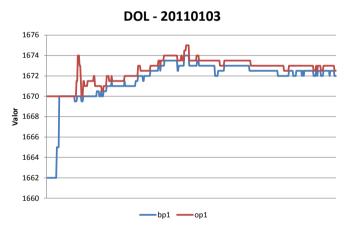


Figura 3.1.1 – evolução intradiária do DOL

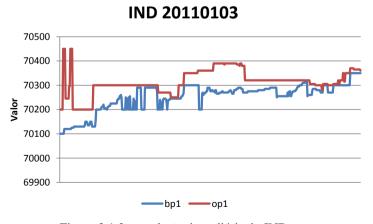


Figura 3.1.2 – evolução intradiária do IND

WIN 20110103 70610 70600 70590 70570 70560 70550 70540 70530 70520 bp1 op1

Figura 3.1.3 – evolução intradiária do WIN

Nas figuras acima, podemos ver uma das características presente nos preços das bolsas de todo o mundo. O valor do preço BID é sempre menor que o ASK, ou seja, sempre estão querendo pagar menos na hora de comprar e vender por um preço mais caro. Fica claro nesse gráfico que quanto maior for a diferença entre esses dois valores, o spread, mais distantes estaremos de ter um negócio realizado, no primeiro gráfico (fig.3.1.1) pelo fato dessa distância ser menor, na maior parte do tempo, os negócios ocorrem com mais frequência ao contrário da figura seguinte (fig.3.1.2) que, na maior parte da evolução, apresenta uma distância maior. Podemos ver, portanto, que o spread acaba tendo uma maior importância que os preços isoladamente. Devido à sua relevância, iremos voltar nossa atenção para ele no próximo tópico que irá tentar ajudar na previsão desse valor.

3.2. Teste para propriedade Markoviana

A propriedade fundamental Markoviana diz que, pensando em uma série de valores, a probabilidade da ocorrência de um determinado evento, depende somente da ocorrência do evento imediatamente anterior e não de todo o seu histórico. Segundo [7], existem na literatura somente dois testes que averiguam a propriedade Markoviana: Aït-Sahalia (2000) e Fernandes e Flores (2004). O primeiro teste usa o fato de que a equação de Chapman-Kolmogorov deve ser satisfeita para que exista a compatibilidade com a hipótese Markoviana, o segundo desenvolve caminhos alternativos para testar se observações discretas são consistentes com um processo de Markov subjacente.

No entanto, ambos os processos assumem que os dados são uniformemente espaçados no tempo. Dados de transações financeiras não satisfazem essa propriedade e, portanto, esses testes não são apropriados. Para construir um teste para propriedade Markoviana que seja adequado para dados de alta frequência, nos baseamos na teoria dos processos de Markov com mudanças estocásticas no tempo. Nós consideramos os dados que produzem realizações discretas no tempo à medida que mudanças acumulativas, nos valores da série, contribuem para um nível discreto. Portanto, acomodamos os espaçamentos irregulares que caracterizam dados de transição. Além disso, como um esquema de amostragem opcional sugere, incrementos consecutivos entre mudanças observadas na série são independentes condicionalmente, dada as

realizações discretas no tempo da série. Nós, então, desenvolvemos um simples teste não paramétrico, para a propriedade Markoviana, averiguando se a propriedade de independência condicional se mantem.

Existe uma grande literatura sobre como testar ou a independência incondicional ou independência em série, entretanto, existem somente alguns artigos discutindo testes de independência condicional: Linton e Gozalo (1999) e, mais especificamente, Su e White (2002, 2003). Linton e Gozalo (1999) testam a independência condicional, entre variáveis aleatórias independentes e identicamente distribuídas, olhando as restrições da função de distribuição acumulada sobre uma medida quadrática de distância. Su e White (2002, 2003) expandem essa aproximação considerando processos de dependência estocástica fraca assim como diferentes medidas. Em particular, Su e White (2002) verificam se a restrição da densidade implicada pela independência condicional se mantem usando a distância Hellinger, enquanto que Su e White (2003) verificam restrições na função característica e na função empírica de verossimilhança.

Nesse trabalho iremos testar, como em [7], combinando os ajustes de Linton e Gozalo (1999) e Su e White (2002). Como em Su e White (2002), [7] deriva testes sob condições mixadas, assim como trata a dependência da serie temporal dos dados. Entretanto, eles avaliam quão bem a restrição da densidade, implicada pela propriedade de independência condicional, se ajusta aos dados usando a medida quadrática de distância como em Lintoon e Gozalo (1999).

Uma aplicação relevante desse procedimento de teste é checar se a diferença bid-ask segue um processo de Markov. Modelos de informação assimétrica para microestrutura de mercado preveem que a diferença bid-ask depende de todo o histórico da negociação e, consequentemente, a propriedade Markoviana não é satisfeita. A aproximação não paramétrica de [7] é consistente com o objetivo em Hasbrouck's (1991), de descobrir a extensão dos custos de seleção adversa em um quadro que é robusto à desvios da suposição de modelos formais da microestrutura de mercado. Tendo isso em mente, nos examinamos dados da Bolsa de Valores de São Paulo (BM&F), com a ideia de que uma das possibilidades de melhoria dos algoritmos descritos em [1] seria a utilização da propriedade descrita acima, sobre a diferença bid-ask. Para tanto, devemos verificar se os dados satisfazem a propriedade fundamental dos processos de Markov. Como é explicado em [7], a estatística:

$$\hat{\lambda} = \frac{nb_n^{3/2}\Lambda_{\hat{f}} - b_n^{-3/2}\widehat{\delta_{\Lambda}}}{\widehat{\sigma_{\Lambda}}} \xrightarrow{d} N(0,1)$$

Tem distribuição Normal Padrão sob a hipótese nula:

$$H_0^*: f_{iX_j}(a_1, x, a_2) = f_{i|X}(a_1) f_{X_j}(x, a_2), para\ todo\ j < i$$

Onde

$$\hat{\delta}_{\Lambda} = \frac{e_k}{n} \sum_{k=1}^{n} \omega_{iX_j} (d_{k+j}, X_k, d_k) \hat{f}_{iX_j} (d_{k+j}, X_k, d_k)$$

$$\Lambda_{f} = \frac{1}{n} \sum_{k=1}^{n} \omega(d_{k+j}, X_{k}, d_{k}) \left[\hat{f}_{iX_{j}}(d_{k+j}, X_{k}, d_{k}) - \hat{g}_{iX_{j}}(d_{k+j}, X_{k}, d_{k}) \right]^{2}$$

$$\hat{f}_{iX_j}(a_1, x, a_2) = \frac{1}{nb_n^3} \sum_{k=1}^n K\left(\frac{a_1 - d_{k+j}}{b_n}\right) K\left(\frac{x - X_k}{b_n}\right) K\left(\frac{a_2 - d_k}{b_n}\right)$$

$$\hat{f}_{iX}(a_{1,i}x) = \frac{1}{nb_n^2} \sum_{k=1}^n K\left(\frac{a_1 - d_{k+j}}{b_n}\right) K(\frac{x - X_k}{b_n})$$

$$\hat{f}_{X_j}(x, a_2) = \frac{1}{nb_n^2} \sum_{k=1}^n K\left(\frac{x - X_k}{b_n}\right) K\left(\frac{a_2 - d_k}{b_n}\right)$$

$$\hat{f}_X(x) = \frac{1}{nb_n} \sum_{k=1}^n K(\frac{x - X_k}{b_n})$$

$$\hat{\sigma}_{\Lambda}^{2} = \frac{v_{K}}{n} \sum_{k=1}^{n} \omega_{iX_{j}}^{2} (d_{k+j}, X_{k}, d_{k}) \hat{f}_{iX_{j}}^{3} (d_{k+j}, X_{k}, d_{k})$$

$$e_K = \left[\frac{27}{32\sqrt{\pi}}\right]^3$$
, $v_K = \left[\frac{7871\sqrt{2}}{16384\sqrt{\pi}}\right]^3$, $b_n = b_0 n^{-1/9}$, $b_0 = \frac{2.04}{\log n}$

 $\omega(x,y,z)$, é uma função indicadora.

Portanto, caso a propriedade seja verificada pelo teste, podemos fatorar a densidade de probabilidade como o produto das densidades condicional e conjunta. Como é inviável testar tal restrição para todas as observações passadas iremos fixar j.

$$H_0: f_{iX_j}(a_1, x, a_2) = f_{i|X}(a_1) f_{X_j}(x, a_2), para todo o j fixo$$

Uma das transformações realizadas nos dados foi a tomada do logaritmo natural do spread para que, dessa forma, os dados não ficassem viesados por causa da utilização de núcleos simétricos no processo de estimação da distribuição, em seguida, fazemos a normalização dos dados (subtrair a média e dividir pelo desvio padrão cada dado), para que assim, tenhamos a mesma largura de banda para os resultados da duração e da série do spread.

Como a propriedade de independência condicional é invariante para as transformações monotônicas acima, podemos aplicar o teste não paramétrico descrito.

Para manter a natureza não paramétrica do teste, [7] aplica uma suavização de núcleo para estimar a funções de densidade. Funções de densidade nos fornecem uma descrição natural da distribuição de probabilidades de uma dada variável, permitindo que as mesmas sejam encontradas, a técnica mais básica para sua estimação consiste na análise do histograma que os dados possuam. No entanto, nesse trabalho, utilizaremos a estimação de densidades através do método de Kernel que pode ser pensado como uma generalização do histograma, ou seja, de uma maneira geral, a ideia é que iremos contar quantas observações estão contidas em cada intervalo, dividido em pedaços bem pequenos, e usar como estimativa da densidade uma média ponderada desses valores. Em particular, no método Kernel, utilizamos para esse ponderação uma função de peso não negativa, de integral no domínio igual à 1 e que na maioria das vezes é uma função simétrica. A função utilizada para o núcleo é:

$$K(u) = \frac{3}{\sqrt{8\pi}} \left(1 - \frac{u^2}{3} \right) \exp(-\frac{u^2}{2})$$

No apêndice A.3 apresentamos o código que realiza o teste descrito acima.

3.2.1.Geração da seguência

Primeiramente, vamos extrair a sequencia que desejamos analisar. Segundo a técnica descrita em [7], nós iremos marcar o processo contínuo no tempo somente quando a variação acumulada em \tilde{X}_t é no mínimo c. Ou seja, o tempo decorrido entre duas observações discretas no tempo e consecutivas é

$$\check{d}_{i+1} \equiv t_{i+1} - t_i = \inf_{\tau > 0} \{ |\widetilde{X}_{t_{i+\tau}} - \widetilde{X}_{t_i}| \ge c \}$$

para
$$i = 0,...,n-1$$
.

Os tempos, observados nos dados $\{t_i, i=1,2,...\}$, formam uma sequência crescente de tempos de parada do processo Markoviano contínuo no tempo. Por isso, o processo discreto no tempo satisfaz a propriedade de Markov também. Além disso, a duração \tilde{d}_{i+1} é uma função de medida do caminho de $\{\widetilde{X}_t, 0 < t_i \le t \le t_{i+1}\}$ e, portanto, depende da informação disponível no instante t_i somente por X_i . Logo, podemos concluir que a duração atual é independente da duração anterior condicionada na realização anterior discreta no tempo. É, portanto, natural testar a suposição Markoviana checando se a propriedade de independência condicional entre durações consecutivas se mantém.

No apêndice A.2 apresentamos o código desenvolvido para realizar o que foi descrito acima.

3.2.2.Resultados

DOL		INI	D	WI	N
20110103	106.24	20110103	4404.81	20110103	8045.13
20110104	482.51	20110104	5632.77	20110104	10824.99
20110105	10.30	20110105	6076.12	20110105	9393.75
20110106	12.64	20110106	4088.91	20110106	10837.94
20110107	68.69	20110107	5871.54	20110111	8325.06
20110110	9.45	20110111	2780.80	20110112	7395.37
20110111	4.91	20110112	2880.48	20110113	8038.57
20110112	346.47	20110113	1698.57	20110117	4806.87
20110113	24.66	20110117	1947.48	20110118	7638.11
20110117	19.63	20110118	3006.37	20110124	10987.17
20110118	7.89	20110119	2892.38	20110126	12426.31
20110119	44.60	20110124	4170.18	20110127	12198.93
20110120	0.69	20110126	4398.44	20110131	3604.34
20110124	26.67	20110127	4602.34		
20110126	3.93	20110131	5507.61		
20110127	138.04				
20110131	152.15				

Figura 3.2.2.1 - Tabela dos resultados das estatísticas de teste para Markov

Dos valores acima e comparando com uma tabela de valores da distribuição normal padrão vemos que nossos dados não possuem a propriedade Markoviana, ou seja, a maneira como esses preços se distanciam (spread) não depende somente da última mudança, mas, de todo o seu histórico e, portanto, não conseguimos antecipar, com um certo grau de chance, como se dará a evolução dos preços ASK e BID ao longo de um dia de negociação já que, na prática, fica inviável avaliar sempre todo o histórico das diferenças entre os preços, devido à dinâmica do mercado, para então tomar alguma decisão acerca do que fazer nele.

Portanto, utilizaremos no próximo capítulo o algoritmo descrito em [1], que leva em consideração uma janela de históricos dos dados e não somente a última observação, em nosso conjunto de valores e verificar se o mesmo consegue uma boa performance com os dados da bolsa brasileira.

Algumas discussões podem ser levantadas nesse ponto com relação à trabalhos futuros, a primeira delas é que a diferença entre os preços ASK e BID (Spread) pode perder a propriedade Markoviana enquanto que o par dos preços possa mantê-la. Uma possibilidade de novas pesquisas seria testar a propriedade para o processo bidimensional (Pask, Pbid). Outro ponto é considerar o conceito de cadeias de Markov com alcance variado, o algoritmo de árvores de contexto mostrou que o Spread se comporta como uma cadeia de Markov de alcance variado, mas, a altura da árvore de contexto (ou profundidade da memória) não é grande e é aproximadamente igual à nove ticks. Por último, poderíamos também considerar o processo dividido em instantes "ticks", ou seja, um processo com tempo discreto, e acompanhar as mudanças na evolução dos preços.

Algoritmos e suas aplicações

Nesse capítulo, iremos apresentar um método de aplicação do algoritmo distinto do descrito em [1]. Ambos os algoritmos descritos em [1] fazem uma comparação entre a média móvel exponencial dos últimos preços (Ask e Bid), em particular 5 minutos atrás e 15 minutos atrás (respectivamente p_{ask}^{EMA} , p_{bid}^{EMA} e p_{ask}^{EMA+} , p_{bid}^{EMA+}), sendo assim possível averiguar se uma possível tendência do preço é de aumento ou de queda e, então, entrar ou sair de uma posição visando ter um retorno positivo. A técnica de aplicação desses algoritmos, descrita em [1], consiste em, para um mesmo dia de negociação, testar diversos parâmetros de controle (constantes que delimitam quanto queremos investir no mercado, a quantidade de perda que estamos capacitados a enfrentar e a quantidade de ganhos que queremos ter) e então escolher qual o melhor dentre eles, ou seja, aquele que apresenta o maior e mais vezes positivo retorno. Tal método pode às vezes ser custoso, já que são necessárias muitas tentativas até o acerto dos parâmetros mais adequados.

Logo, para evitarmos tal processo, iremos apresentar uma metodologia de aplicação do algoritmo em duas etapas. Na primeira, aplicamos o algoritmo a uma parcela dos dados (10% do total) e armazenamos os valores assumidos pelos parâmetros para podermos, posteriormente, estimar valores para os mesmos através da média. A constante de ganho (chamada de rho) será a média dos valores observados assim como a de perda (chamada de sigma) também será. De posse desses valores, aplicamos os parâmetros estimados ao restante dos dados, sendo que, em ambos os casos, iremos nos expor a uma variação de 0.01% em relação à média (comparação entre a média exponencial e os preços Bid e Ask). Na prática é como se, por exemplo, na parte da manhã ficássemos fora das aplicações da bolsa somente armazenando valores, para que depois, feito os cálculos, passássemos a investir na parte da tarde.

O método é então aplicado dia à dia, visto que foi possível observar uma certa variação entre os dias de um mesmo índice na BM&F BOVESPA e, portanto, os parâmetros calculados em um dia já não servem para o seguinte. É possível também que, dentro desse período de desenvolvimento, o algoritmo não apresente os valores de controle de lucro. Nesse caso, iremos utilizar como constante a média dos dois últimos dias observados do índice tanto para a constante de ganho (rho) quanto a de perda (sigma).

Entrando em um pouco mais de detalhes sobre o funcionamento do algoritmo, ele atua fazendo comparações entre os preços ASK e BID e suas respectivas médias móveis exponenciais (janelas de 5 e 15 minutos para trás, como falamos) para decidir qual ação tomar. Abaixo temos a expressão para a média móvel utilizada no algoritmo.

$$X_1 = Y_1$$

$$t > 1, X_t = \alpha Y_t + (1 - \alpha) X_{t-1}$$

$$\alpha = \frac{2}{n+1}$$

Onde n é o tamanho da nossa amostra. Essas comparações entre os preços são feitas avaliando a diferença entre o produto das médias dos preços e o produto de seus próprios valores, se essa diferença é negativa indica que os preços atuais são maiores que seus valores passados e, portanto, o mercado está em um momento de alta, caso contrário, se a diferença é positiva, indica que os preços atuais são menores que suas médias e, portanto, o momento é de queda, como podemos ver na expressão abaixo.

$$D(t_i) = \frac{p_{ask}^{EMA} p_{bid}^{EMA} - p_{ask}(t_i) p_{bi}(t_i)}{p_{bid} p_{ask}^{EMA}(t_i)}$$

Aqui apresentamos a visão de 5 minutos para trás, a outra expressão utilizada nessas tomadas de decisão possui a mesma forma, porém, utiliza a média móvel exponencial 15 minutos para trás.

$$E(t_i) = \left[\frac{p_{bid}^{EMA+}(t_i)}{p_{bid}(t_i)} - 1 \right] - \left[\frac{p_{ask}(t_i)}{p_{ask}^{EMA+}(t_i)} - 1 \right]$$

Outra função utilizada é a que nos diz o quanto iremos nos expor no mercado, essa expressão atua de maneira recursiva e assume os seguintes valores.

$$g_{t_i}(D,E) = \begin{cases} g_{t_{i-1}}(D,E) + 1 \ se \ D(t_i) > 1.5\epsilon \ e \ E(t_i) > 1.5\epsilon \\ g_{t_{i-1}}(D,E) + \frac{1}{2} \ se \ D(t_i) > \epsilon \ e \ E(t_i) > \epsilon \\ g_{t_{i-1}}(D,E) - 1 \ se \ D(t_i) < -1.5\epsilon \ e \ E(t_i) < -1.5\epsilon \\ g_{t_{i-1}}(D,E) - \frac{1}{2} \ se \ D(t_i) < -\epsilon \ e \ E(t_i) < -\epsilon \end{cases}$$

Onde o valor D(t) e E(t) são as diferenças mencionadas acima.

No desenvolvimento do algoritmo, iniciamos a função recursiva de exposição com o valor zero.

No primeiro caso, momento de alta do mercado, a ação do algoritmo é vender a opção dentro do mercado, caso já tenha alguma, para conseguir assim um maior retorno devido à alta dos preços, no segundo caso, o algoritmo aproveita a queda dos preços para comprar opções e, portanto, fazer uma reserva da mesma na espera de um momento de alta para novamente vender e ficar com a diferença dos valores, de quanto comprou e por quanto vendeu. Em ambos os casos, iremos avaliar essa diferenca com uma precisão pré-determinada que será nosso parâmetro épsilon (como havíamos dito antes, escolhemos o valor 0.01%). Outra parte que o algoritmo apresenta são as funções de controle de perda e ganho (relacionadas com as quantidades de perda e ganho que estamos preparados para ter), cuja forma de estimação dos parâmetros, rho e sigma, foi explicada acima, elas fazem comparações entre os gastos das transações junto com a tendência que o mercado esteja seguindo (alta ou queda) e avalia se isso está de acordo com o esperado, ou seja, da forma como o algoritmo é aplicado, podemos avaliar na parte da manhã, onde os parâmetros são estimados, se vamos querer investir na parte da tarde daquele dia ou não, já que, podemos avaliar de ante mão quanto serão os ganhos e perdas médios do dia.

Passemos agora a analisar o comportamento de tal algoritmo, para os índices da BM&F BOVESPA, para os mesmos dias em que realizamos os testes. O código para aplicação do algoritmo encontra-se descrito no apêndice A.4, apresentaremos para cada índice a tabela com os parâmetros estimados nos primeiros 10% dos dados e, em seguida, os gráficos dos retornos obtidos com a aplicação desses parâmetros, suas tabelas com valores das estatísticas, o gráfico do retorno acumulado e quantas vezes os controles de ganho e perda foram acionados, sendo esse último um dos parâmetros de avaliação da qualidade do algoritmo descrita em [1].

Em geral, foi possível observar que o algoritmo é sensível às variações diárias que o índice possa sofrer, ou seja, nem sempre ele será lucrativo para o investidor. Sigamos, então, com os resultados para cada um dos índices, lembrando que o valor do retorno de aplicação é dado por.

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$$

4.1.Índice DOL

O primeiro índice onde aplicamos o algoritmo foi o do Dólar Comercial. Podemos ver o comportamento do mesmo abaixo. Para o período de desenvolvimento temos os seguintes parâmetros de controle estimados.

data	rho	sigma	rho=NaN(média dos 2 dias anteriores)
20110103	-0.0076	-24.9667	
20110104	-0.0165	-1.0901	
20110105	-0.0007	-43.1808	
20110106	-0.0213	-5.4071	
20110107	-0.0045	-2.7804	
20110110	-0.0121	0.0084	
20110111	-0.0013	-41.6498	
20110112	-0.0073	-4.7120	
20110113	NaN	-71.8505	-0.0043
20110117	-0.0078	-6.1249	
20110118	-0.0078	-0.0335	
20110119	-0.0010	-29.0735	
20110120	NaN	-71.6035	-0.0044
20110124	NaN	-62.3009	-0.0027
20110126	-0.0057	-5.7483	
20110127	-0.0004	-22.6940	
20110131	NaN	-29.1040	-0.0030

Figura 4.1.1 – Tabela dos parâmetros do algoritmo estimados para o índice DOL

De posse desses valores passamos a executar o algoritmo para cada um dos dias acima.

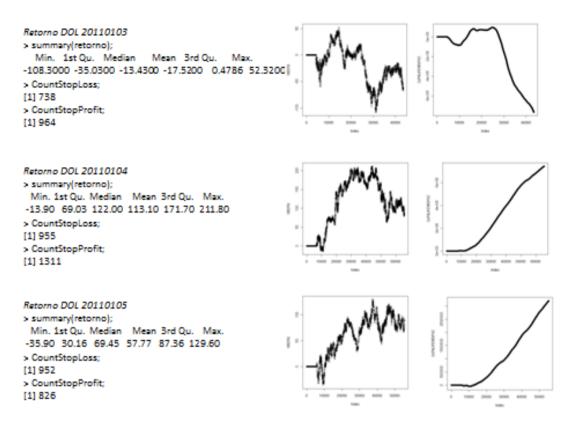


Figura 4.1.2 – Gráficos dos retornos e retornos acumulados DOL (3, 4 e 5 de jan de 2011)

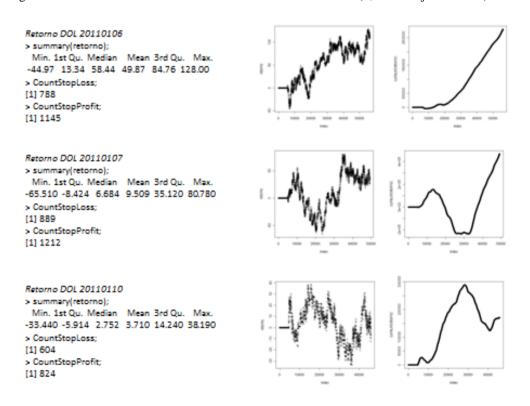


Figura 4.1.3 – Gráficos dos retornos e retornos acumulados DOL (6, 7 e 10 de jan de 2011)

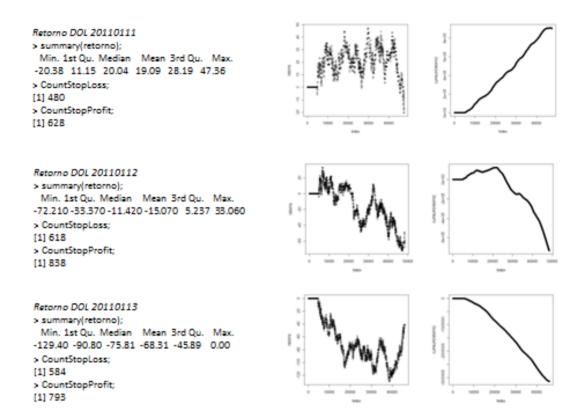


Figura 4.1.4 – Gráficos dos retornos e retornos acumulados DOL (11, 12 e 13 de jan de 2011)

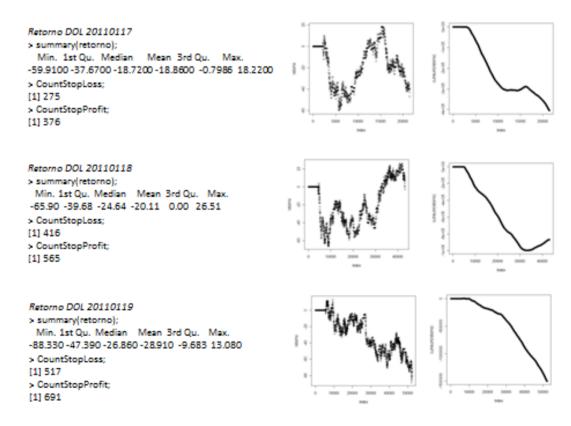
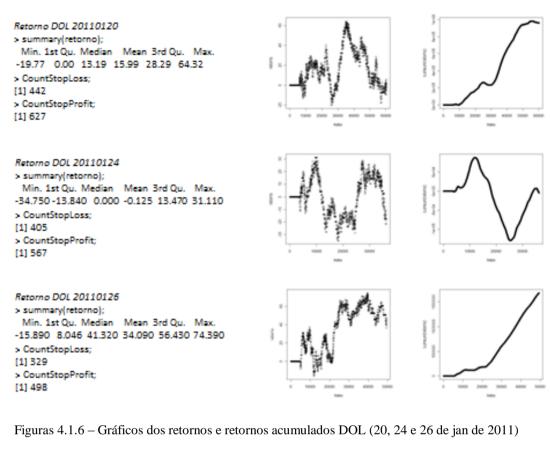


Figura 4.1.5 – Gráficos dos retornos e retornos acumulados DOL (17, 18 e 19 de jan de 2011)



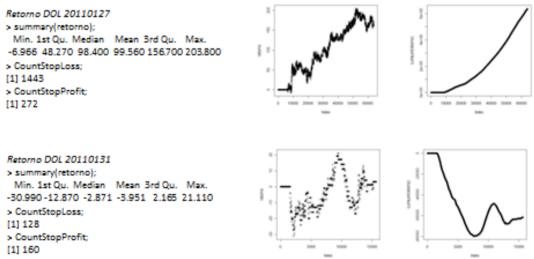


Figura 4.1.7 – Gráficos dos retornos e retornos acumulados DOL (27 e 31 de jan de 2011)

Fazendo simplesmente a aplicação do algoritmo, como se encontra descrito em [1], era muito difícil determinar quais os valores adequados para as constantes de controle. Por causa disso, muitas vezes não tínhamos controle sobre nossos ganhos e perdas e com isso podíamos lucrar bastante em um dia, mas perder muito mais em outro.

4.2.Índice IND

Temos para os parâmetros estimados.

data	rho	sigma
20110103	-0.1163	-1.4037
20110104	-0.0034	-2.1102
20110105	-0.0579	-0.0505
20110106	-0.0040	-1.5703
20110107	-0.0029	-27.1523
20110111	-0.0071	-4.4315
20110112	-0.0100	-0.0119
20110113	-0.0058	-0.5862
20110117	-0.0003	-49.6394
20110118	-0.0010	-17.0957
20110119	-0.0095	0.0075
20110124	-0.0164	-7.5581
20110126	-0.0009	-42.6819
20110127	-0.0445	0.0215
20110131	-0.0057	-24.0467

Figura 4.2.1 – Tabela dos parâmetros do algoritmo estimados para o índice IND

Os gráficos dos retornos ficam os seguintes.

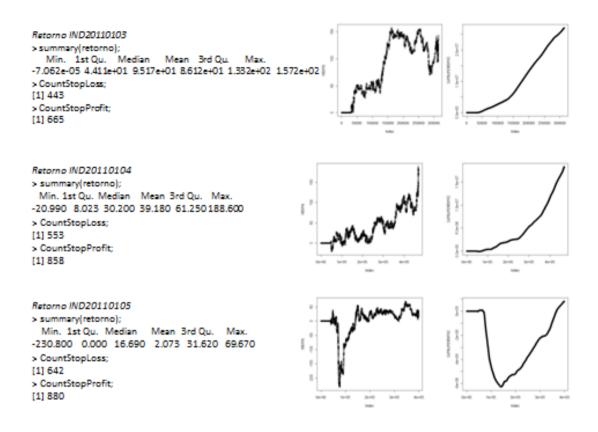


Figura 4.2.2 – Gráficos dos retornos e retornos acumulados IND (3, 4 e 5 de jan de 2011)

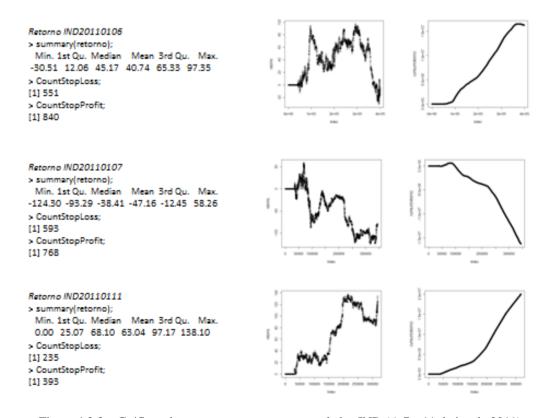


Figura 4.2.3 – Gráficos dos retornos e retornos acumulados IND (6, 7 e 11 de jan de 2011)

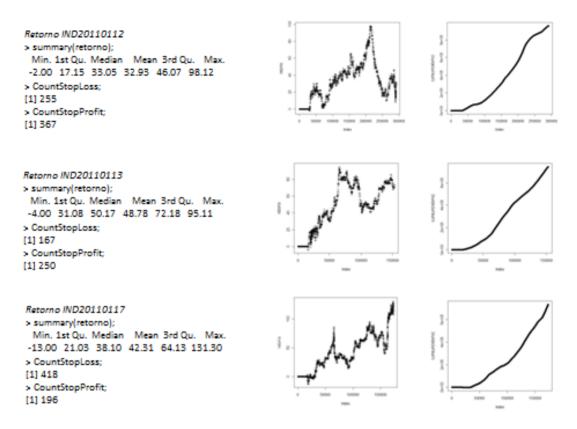


Figura 4.2.4 – Gráficos dos retornos e retornos acumulados IND (12, 13 e 17 de jan de 2011)

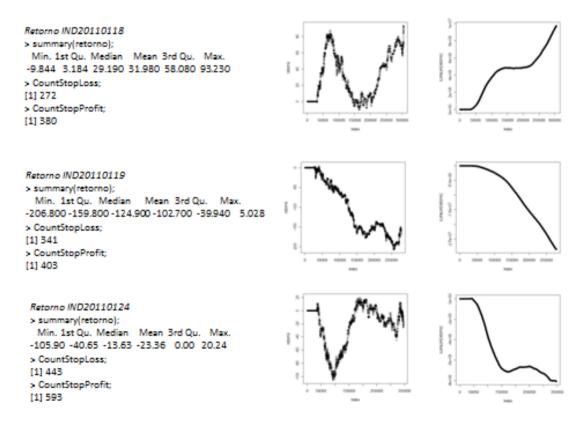


Figura 4.2.5 – Gráficos dos retornos e retornos acumulados IND (18, 19 e 24 de jan de 2011)

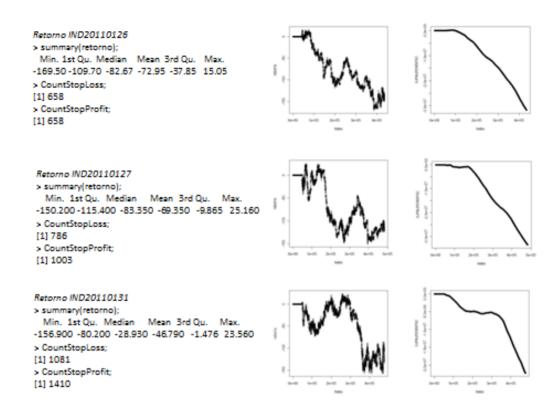


Figura 4.2.6 – Gráficos dos retornos e retornos acumulados IND (26, 77 e 31 de jan. de 2011)

Observando os gráficos podemos ver que a função de controle de perda tem um efeito substancial na redução das perdas do algoritmo. Uma das características desse algoritmo é que ele é um seguidor de tendências, por isso torna-se importante ter controles sobre nossa entrada no mercado.

4.3.Índice WIN

Por último temos os parâmetros estimados para esse índice.

data	rho	sigma	rho=NaN(média dos 2 dias anteriores)
20110103	-0.0782	-1.0754	
20110104	-0.0003	-18.0995	
20110105	-0.0118	0.0099	
20110106	-0.0030	-14.8567	
20110111	-0.0005	-13.6773	
20110112	-0.0048	0.0031	
20110113	-0.0077	-0.0015	
20110117	-0.0007	-40.2274	
20110118	-0.0011	-0.9247	
20110124	-0.0188	-0.0230	
20110126	NaN	-62.8336	-0.0099
20110127	-0.0879	0.0075	
20110131	-0.0041	-15.0268	

Figura 4.3.1 – Tabela dos parâmetros do algoritmo estimados para o índice WIN

E os retornos obtidos desses parâmetros.

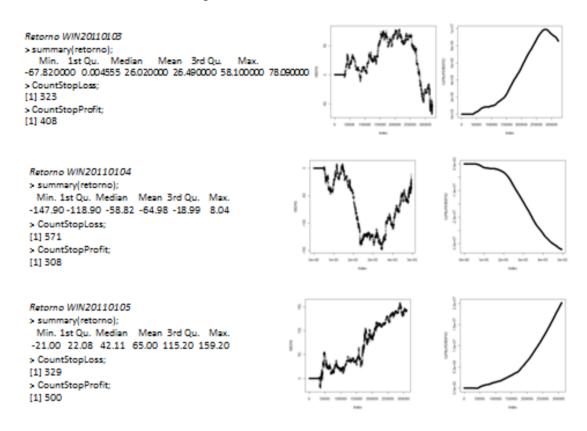


Figura 4.3.2 – Gráficos dos retornos e retornos acumulados WIN (3, 4 e5 de jan de 2011)

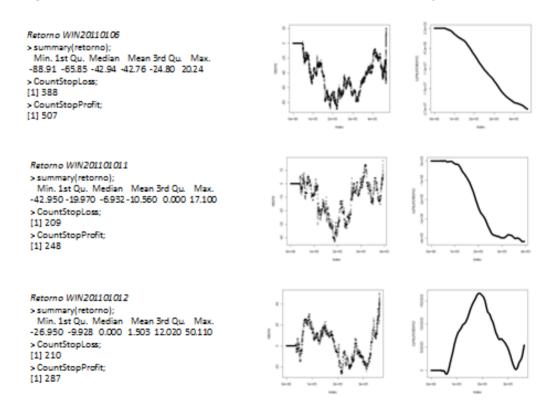
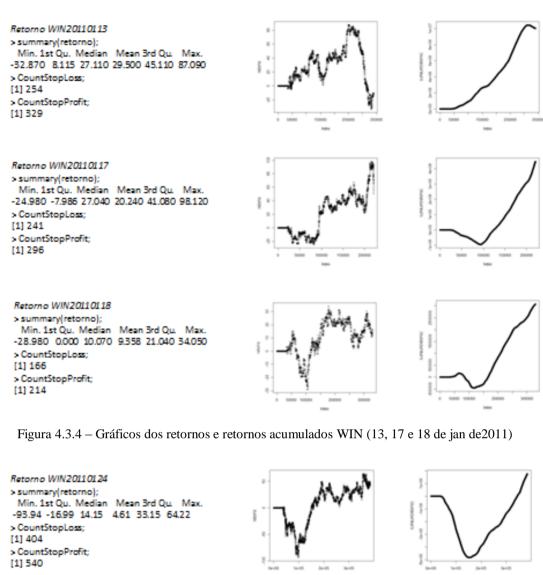


Figura 4.3.3 – Gráficos dos retornos e retornos acumulados WIN (6, 11 e 12de jan de2011)



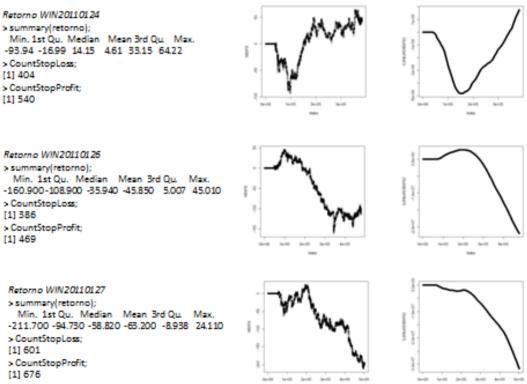
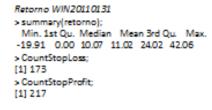
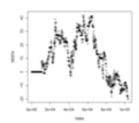


Figura 4.3.5 – Gráficos dos retornos e retornos acumulados WIN (24, 26 e 27de jan de2011)





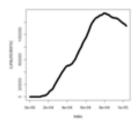


Figura 4.3.6 – Gráficos dos retornos e retornos acumulados WIN (31de jan de2011)

Conclusões

Com o grande conjunto de dados que nos foi fornecido, foi possível estudar melhor a aplicação desse algoritmo de negociação já que, em [1], uma das limitações encontradas pelos autores foi a pequena quantidade de dados disponíveis. Com essa maior quantidade de dias, foi possível observar um pouco melhor o desempenho desse algoritmo. E, pelo menos no mercado brasileiro, o mesmo se apresenta como um algoritmo seguidor de tendências. Aparentemente, os resultados obtidos em [1] foram similares aos nossos e, mais uma vez, a aplicação em tempo real do algoritmo não pode ser concretizada, mas, mais um passo foi dado com relação ao estudo desses algoritmos.

No entanto, ao contrário do que foi desenvolvido em [1], em nosso trabalho demos um maior enfoque matemático na tentativa de desenvolver um algoritmo com melhor aplicabilidade e desempenho. Caso a propriedade Markoviana tivesse sido satisfeita pelos dados da BM&F Bovespa poderíamos, eventualmente, ter elaborado um algoritmo, com tomada de decisão (entra ou sai da posição), baseado na probabilidade de transição de estados para o spread entre os preços Ask e Bid. Outros caminhos poderiam ter sido seguidos também, como por exemplo, modelagem através de Séries Temporais dos preços, para que depois através de projeções desses valores pudessemos optar pela maior exposição ou não.

O fato de nossos resultados indicarem que os dados não satisfazem a propriedade Markoviana também é importante, já que, trabalhos futuros, que tenham interesse em desenvolver algoritmos de negociação ou que tenham algum outro enfoque, poderão ter outro ponto de partida. Nesse ponto, fica a contribuição do código desenvolvido em R para a execução do teste.

Fica também a contribuição à respeito do protocolo FIX que, por ser um protocolo de uso exclusivo das bolsas, não possui nenhuma literatura à respeito. Sendo assim, nosso trabalho serve como um bom guia para trabalhos futuros que venham a ter que extrair determinados valores, como preços, datas, volumes para algum tratamento.

Os códigos que desenvolvemos em R, e que se encontram nos apêndices, podem ser também aproveitados posteriormente ou implementados para servir como base, já que, não existem no R, library's para o teste não paramétrico markoviano nem para extrair dados de um bloco de informação do protocolo FIX.

Atualmente existem grandes equipes, formadas por mestres e doutores, trabalhando no desenvolvimento de algoritmos de negociação em grandes fundos de investimentos, a quantidade de formas existentes desses algoritmos também é grande. A forma como nosso algoritmo foi desenvolvido muito provavelmente já foi abordada em um desses centros de pesquisa e desenvolvimento, no entanto, pensando em nossa base, que era o algoritmo descrito em [1], esse método que apresentamos foi uma melhora natural à ser tomada. E gerou ganhos satisfatórios já que, como havíamos dito antes, sem os parâmetros adequados, o algoritmo poderia ter retornos excelentes em um dia mas, no

seguinte, perder todos os ganhos e ainda mais. Também devemos levar em conta que o trabalho em [1] não possuía muitos dados a serem analisados, o que tornava possível a utilização de seu método. Caso fossemos aplica-lo aqui, teríamos um problema computacional já que, para cada dia de negociação, a quantidade de dados pode chegar, por exemplo, facilmente a 470000 entradas. É o tipo de grandeza que lidamos quando mexemos com dados de alta frequência, a quantidade de negociações na bolsa pode chegar a mais de 36 por segundo.

Outro ponto importante que devemos ressaltar é que o mercado financeiro, tanto aqui quanto na Inglaterra (origem dos dados do trabalho em [1]), é muito heterogêneo e, portanto, não existe estratégia de negociação absolutamente melhor que outras. A estratégia a ser escolhida dependerá do perfil de risco e de negócios do investidor. Sobre esse aspecto, o trabalho serve como uma base para o desenvolvimento de novas estratégias para o mercado brasileiro, que ainda se encontra em desenvolvimento quando comparado com outros países.

Esse algoritmo não se comporta da melhor maneira em pontos de mudança já que o mesmo é um seguidor de tendências do mercado, como podemos observar nos gráficos dos preços médios abaixo, para os primeiros dias de cada um dos índices, em comparação com os respectivos gráficos dos retornos apresentados anteriormente.

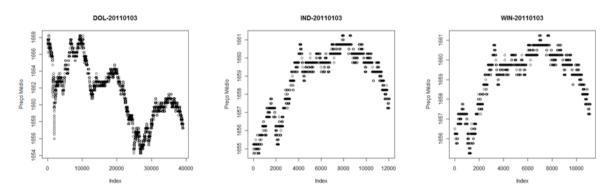


Figura 4.4.1 – Gráficos dos preços médios 3 de jan de 2011 (DOL, IND e WIN)

Ou seja, o que acontece é que, quando o mercado está em queda, o algoritmo compra (consequentemente por um preço menor) para depois vender quando o mercado estiver em alta (consequentemente por um preço maior).

Códigos

Os códigos apresentados a seguir foram desenvolvidos em R.

```
A.1.Extração dos dados
book=vector()
bid.price=vector()
bid.size=vector()
offer.price=vector()
offer.size=vector()
trade=c(NA,NA)
for (j in 1:length(a[,1]))
#for (j in 1:100)
{
v=strsplit(a[i,],"\001")[[1]]
lv=length(v) #dlina vectora s zapisiami
#ll=as.numeric(strsplit(v[grep("268=",v)],"=")[[1]][2]) #chislo
zapisej
ind=c(grep("279=",v),lv)
lind=length(ind)
#names (mm) = c ("x279", "x269", "x270", "x271", "x290")
xtime=strsplit(v[grep("52=",v)],"=")[[1]][2]
for (i in 1:(lind-1))
     #events=0;
     zapis=vector(length=5);
     vv=v[ind[i]:(ind[i+1]-1)];
      x279=strsplit(vv[1],"=")[[1]][2] #v ocherednoj zapisi pole
279 MDUpdateAction
      x269=strsplit(vv[grep("269=",vv)],"=")[[1]][2]
                                                                #∨
ocherednoj zapisi pole 269 MDEntryType
      xx=grep("270=", vv)[1];
                                                      (!is.na(xx))
x270=strsplit(vv[grep("270=",vv)],"=")[[1]][2] else x270=NA #v
ocherednoj zapisi pole 270 MDEntryPrice
      xx=grep("271=", vv)[1];
                                                      (!is.na(xx))
x271=strsplit(vv[grep("271=",vv)],"=")[[1]][2] else x271=NA #v
ocherednoj zapisi pole 271 MDEntrySize
      xx=grep("290=", vv)[1];
                                                      (!is.na(xx))
x290=as.numeric(strsplit(vv[qrep("290=",vv)],"=")[[1]][2]) else
x290=NA #v ocherednoj zapisi pole 290 MDEntryPositionNo
```

```
zapis=c(paste(x279, x269, sep=""), x270, x271, x290)
     # bid
     if
((zapis[1]=="00")&(!is.na(zapis[2]))&(!is.na(zapis[3]))&(!is.na(
zapis[4]))) #new bid price
           if (zapis[4]=="1")
           {
           bid.price=c(zapis[2],bid.price);
           bid.size=c(zapis[3],bid.size);
           #events=events+1
           else
                nomer=as.numeric(zapis[4]);
                if (nomer>length(bid.price))
                bid.price=c(bid.price, zapis[2]);
                bid.size=c(bid.size,zapis[3]);
                events=events+1
                else
                bid.price=c(bid.price[1:(nomer-
1)], zapis[2], bid.price[nomer:length(bid.price)]);
                bid.size=c(bid.size[1:(nomer-
1)],zapis[3],bid.size[nomer:length(bid.size)]);
                events=events+1
     }# end if new bid price
           ((zapis[1]=="20")&(!is.na(zapis[4]))) #delete
price/size
     nomer=as.numeric(zapis[4]);
     bid.price=bid.price[-nomer];
     bid.size=bid.size[-nomer]
     } # end if dete bid price
           ((zapis[1] == "10") & (!is.na(zapis[4]))) #change
                                                               bid
price/size
     nomer=as.numeric(zapis[4]);
     bid.price[nomer]=zapis[2];
     bid.size[nomer]=zapis[3]
     } # end if change bid price
     # offer
```

```
if
((zapis[1]=="01")&(!is.na(zapis[2]))&(!is.na(zapis[3]))&(!is.na(
zapis[4]))) #new bid price
           if (zapis[4]=="1")
           offer.price=c(zapis[2],offer.price);
           offer.size=c(zapis[3],offer.size);
           events=events+1
           }
           else
                nomer=as.numeric(zapis[4]);
                if (nomer>length(offer.price))
                offer.price=c(offer.price, zapis[2]);
                offer.size=c(offer.size,zapis[3]);
                events=events+1
                else
                offer.price=c(offer.price[1:(nomer-
1) ], zapis[2], offer.price[nomer:length(offer.price)]);
                offer.size=c(offer.size[1:(nomer-
1)], zapis[3], offer.size[nomer:length(offer.size)]);
                events=events+1
     }# end if new offer price
     if
          ((zapis[1]=="21")&(!is.na(zapis[4]))) #delete offer
price/size
     {
     nomer=as.numeric(zapis[4]);
     offer.price=offer.price[-nomer];
     offer.size=offer.size[-nomer]
     } # end if delete offer price
     if
          ((zapis[1] == "11") & (!is.na(zapis[4]))) #change
price/size
     nomer=as.numeric(zapis[4]);
     offer.price[nomer]=zapis[2];
     offer.size[nomer]=zapis[3]
     } # end if change offer price
     # trades
     if (zapis[1]=="02") # new trade note
     #nomer=as.numeric(zapis[4]);
     trade=c(zapis[2], zapis[3]);
     tt=1
```

```
} # end if new trade note
     if (tt==0) trade=c(NA,NA)
      }# end for
book=rbind(book,
c(xtime,bid.size[5:1],bid.price[5:1],offer.price[1:5],offer.size
[1:5], trade) )
tt=0;
#mat=rbind(mat,mm)
#mat=rbind(mat,c("x","x","x","x","x"))
}
book=data.frame(book)
names(book) = c("time", "bs5", "bs4", "bs3", "bs2", "bs1", "bp5", "bp4", "
bp3", "bp2", "bp1", "op1", "op2", "op3", "op4", "op5", "os1", "os2", "os3"
"os4", "os5", "tp", "ts")
write.table(book, file = "book-testR", append = FALSE, quote =
TRUE, sep="\t",
            eol = "\n", na = "NA", dec = ".", row.names = FALSE,
            col.names = TRUE)
```

A.2.Marcação da série discreta

```
dados<-
read.table("C:/.../xAAAMMDD.txt", header=T, stringsAsFactor=FALSE);
tempo x<-read.table("C:/.../</pre>
dAAAAMMDD.txt.", header=T, stringsAsFactor=FALSE);
dados<-dados[,1];</pre>
tempo x < -tempo x[,1];
x<-(dados-mean(dados))/sd(dados);
resp<-vector();</pre>
tempo resp<-vector();</pre>
c=1/16;
i=1;
j=1;
k=1;
lim inf=x[i]-c;
lim sup=x[i]+c;
while(i<=length(x)){</pre>
          if(x[i] < lim inf || x[i] > lim sup) {
          resp[j]=x[i];
          tempo resp[j]=tempo x[i]-tempo x[k];
          lim inf=x[i]-c;
          \lim \sup =x[i]+c;
          k=i;
          i=i+1;
          j=j+1;
      else i=i+1;
}
write.table(resp, "C:/.../smAAAMMDD.txt");
write.table(tempo resp, "C:/.../sm tempoAAAAMMDD.txt");
```

A.3.Cálculo da estatística de teste

```
#ENTRADA DE DADOS
dados x<-read.table("G:/.../smAAAMMDD.txt", header=T, dec=".");</pre>
dados d<-
read.table("G:/.../sm tempoAAAMMDD.txt", header=T, dec=".");
dados x < -dados x[,1];
dados d<-dados d[,1];
n<-length(dados x);</pre>
#LIMITES DA FUNÇÃO INDICADORA
\lim \inf =-2;
\lim \sup =2;
#CONTANTES
J=1;
bn=(2.04/log(n))*(n^{(-1/9)});
ek=(27/(32*sqrt(pi)))^3;
vk = ((7871*sqrt(2))/(16384*sqrt(pi)))^3;
indicadora <- function (x, y, z) {
                       if (x>=\lim \inf \&\& x<=\lim \sup) \{a=1;\}
                       else a=0;
                       if(y>=lim inf && y<=lim sup) {b=1;}
                       else b=0;
                       if(z \ge \lim \inf \&\& z \le \lim \sup \{d=1;\}
                       else d=0;
              a*b*d}
nucleo<-function(u) {</pre>
                  if( u>(-sqrt(3)) && u<sqrt(3)){
                      r=(3/sqrt(8*pi))*(1-((u^2)/3))*exp(-
((u^2)/2);
                  }
                  else{r=0;}
         r}
 fx<-function(x){</pre>
               i=1;
               soma=0;
               while(i<=length(dados x)){</pre>
                      soma=soma+nucleo((x-dados x[i])/bn);
                     i=i+1;
     (1/(n*bn))*soma}
fxj<-function(x,a){</pre>
               i=1;
               while(i<=length(dados x)){</pre>
```

```
soma=soma+(nucleo((x-
dados x[i])/bn)*nucleo((a-dados d[i])/bn));
                    i=i+1;
       (1/(n*(bn^2)))*soma}
 fiX<-function(x,a){</pre>
              i=1;
              soma=0;
              while(i<length(dados x)){</pre>
                    soma=soma+(nucleo((a-
dados d[i+J])/bn)*nucleo((x-dados x[i])/bn));
                    i=i+1;
      (1/(n*(bn^2)))*soma}
fixj<-function(a1,x,a2) {
              i=1;
              soma=0;
              while(i<length(dados x)){</pre>
                    soma=soma+(nucleo((a1-
dados d[i+J])/bn)*nucleo((x-dados x[i])/bn)*nucleo((a2-
dados d[i])/bn));
                    i=i+1;
      (1/(n*(bn^3)))*soma}
gixj < -function(x, y, z) {
                r = (fiX(x,y) * fxj(y,z)) / fx(y);
#CALCULO DO LAMBDA MAIUSCULO#
i=1;
soma=0;
while(i<length(dados x)){</pre>
soma=soma+(indicadora(dados d[i+J], dados x[i], dados d[i]) * ((fixj
(dados d[i+J], dados x[i], dados d[i]) -
gixj(dados d[i+J],dados x[i],dados d[i]))^2));
     i=i+1;
lambda f=(1/n)*soma;
#CALCULO DO DELTA#
i=1;
soma=0;
while(i<length(dados x)){</pre>
soma=soma+(indicadora(dados d[i+J],dados x[i],dados d[i])*fixj(d
ados d[i+J],dados x[i],dados d[i]));
     i=i+1;
delta=(ek/n)*soma;
```

```
#CALCULO DO SIGMA AO QUADRADO#
i=1;
soma=0;
while(i<length(dados_x)) {

soma=soma+((indicadora(dados_d[i+J],dados_x[i],dados_d[i])^2)*(
fixj(dados_d[i+J],dados_x[i],dados_d[i])^3 ));
    i=i+1;
}
sigma=(vk/n)*soma;

#ESTATISTICA DE TESTE#
estatistica=((n*(bn^1.5)*lambda_f)-((bn^(-1.5))*delta))/sqrt(sigma);</pre>
```

A.4.Algoritmo de negociação

```
#AJUSTE DO ALGORITMO MODELO 1 (VERSÃO2)
  library(TTR)
  #entrada dos dados
  a<-read.table("G:/.../aAAAMMDD.txt", header=TRUE, na.strings="NA")
  b<-read.table("G:/.../bAAAMMDD.txt", header=TRUE, na.strings="NA")
  a < -as.vector(a[,1])
  b<-as.vector(b[,1])
  #parâmetros do algoritmo
  M < -5;
  N < -15;
  epsilon<-0.0001;
  rho<-(valor1);
  sigma<-(valor2);
  #ajuste das médias e das razões a serem utilizadas
  pEMA ask<-NULL;
  pEMA P ask<-NULL;
  pEMA bid<-NULL;
  pEMA P bid<-NULL;
  A P<-NULL;
  A PP<-NULL;
  B P<-NULL;
  B PP<-NULL;
  pEMA ask<-EMA(a,M)
  pEMA P ask<-EMA(a,N)
  pEMA bid<-EMA(b, M)
  pEMA P bid<-EMA(b,N)
  A P \leftarrow (a[(N+1):length(a)]/pEMA ask[(N+1):length(a)]) -
rep(1, length(a) - N);
 B P<-(pEMA bid[(N+1):length(b)]/b[(N+1):length(b)])-</pre>
rep(1, length(b) - N);
  A PP \leftarrow (a[(N+1):length(a)]/pEMA P ask[(N+1):length(a)]) -
rep(1, length(a) - N);
 B PP<-(pEMA P bid[(N+1):length(b)]/b[(N+1):length(b)])-</pre>
rep(1,length(b)-N);
  D <-B P-A P;
  E <-B PP-A PP;
  # variáveis de monitoramento
  retorno<-rep(0,length(D)-1);
```

```
CountStopProfit<-0;
  CountStopLoss<-0;
  Gearing<-rep(0,2);
  AverPrice<-rep(0,2);
  TransacPrice<-0;
  i<-4848;
  while(i < length(E)){</pre>
  Gearing[1]<-Gearing[2];</pre>
  retorno[i] <-retorno[max(1,(i-1))];</pre>
  #condição de objetivos de lucros
  if((AverPrice[1]!=0&&Gearing[1]>0&&(Gearing[1]*((b[i+N]/AverPr
ice[1])-
1))>rho)||(AverPrice[1]!=0&&Gearing[1]<0&&Gearing[1]*((a[i+N]/Av
erPrice[1])-1)>rho)){
  cat("!! STOP PROFIT LEVEL REACHED !!");
  CountStopProfit<-CountStopProfit+1;</pre>
  # Dependendo da posição em que estamos nos voltamos para o
gearing neutro
  Gearing[2]<-0;
  if(Gearing[1]>0){TransacPrice<-b[i+N];}</pre>
  if (Gearing[1]<0) {TransacPrice<-(-a[i+N]);}</pre>
   AverPrice[1]<-AverPrice[2];</pre>
   AverPrice[2]<-0;
  retorno[i] <- retorno[max(1,(i-
1))]+Gearing[1]*((TransacPrice/AverPrice[1])-1);
         i < -i + 1;
         Gearing[1]<-Gearing[2];</pre>
         retorno[i]<-retorno[max(1,(i-1))];</pre>
       }
  #condição de controle de perda
  if((AverPrice[1]!=0&&Gearing[1]>0&&Gearing[1]*((b[i+N]/AverPri
ce[1])-1)< -sigma)||(AverPrice[1]!=0&&Gearing[1]<0&&
Gearing[1] * ((a[i+N]/AverPrice[1])-1) < -sigma)) {
         cat("!! STOP LOSS LEVEL REACHED !!");
         CountStopLoss<-CountStopLoss+1;</pre>
  #Dependendo da posição em que estamos nos voltamos para o
gearing neutro
         Gearing[2]<-0;
         if (Gearing[1]>0) {TransacPrice<-b[i+N];}</pre>
         if (Gearing[1]<0) {TransacPrice<-(-a[i+N]);}</pre>
         AverPrice[1]<-AverPrice[2];</pre>
```

```
AverPrice[2]<-0;
          retorno[i] <- retorno[max(1,(i-
1))]+Gearing[1]*((TransacPrice/AverPrice[1])-1);
          i<-i+1;
          Gearing[1]<-Gearing[2];</pre>
          retorno[i] <-retorno[max(1,(i-1))];</pre>
        if(D[i] < (-epsilon) \& E[i] < (-epsilon) \& (i <= length(E))){
               #Condição para decidir o quanto entramos na posição
               if(D[i] < (-1.5*epsilon) \& E[i] < (-1.5*epsilon) &
(i \le length(E))) {
                  Gearing[2]<-Gearing[1]-1;</pre>
                  TransacPrice<- b[i+N];</pre>
               }
              else{
                    Gearing[2] < -Gearing[1] - (1/2);
                    TransacPrice<- b[i+N];</pre>
               }
  #cálculo do preço médio
               if (abs (Gearing[2]) <abs (Gearing[1]) &&
Gearing[1] *Gearing[2] > 0) {
                 AverPrice[1]<-AverPrice[2];</pre>
               if (abs (Gearing[2]) > abs (Gearing[1]) &&
Gearing[1] *Gearing[2] > 0 & & AverPrice[2]!=0) {
                 AverPrice[1]<-AverPrice[2];</pre>
                 AverPrice[2] <- Gearing[2] / (((Gearing[2]-
Gearing[1])/TransacPrice)+(Gearing[1]/AverPrice[1]));
               if (Gearing[1] *Gearing[2] < 0 | | Gearing[1] == 0) {</pre>
                 AverPrice[1]<-AverPrice[2];</pre>
                 AverPrice[2]<-TransacPrice;</pre>
               if(Gearing[2]==0){
                 AverPrice[1]<-AverPrice[2];</pre>
                 AverPrice[2]<-0;
               }
  #cálculo do retorno
               if(Gearing[1]*Gearing[2]<0 && AverPrice[1]!=0){</pre>
                 retorno[i] <- retorno[max(1,(i-
1))]+Gearing[1]*((TransacPrice/AverPrice[1])-1);
              if (Gearing[1] *Gearing[2] >= 0 && AverPrice[1]!=0) {
                 retorno[i] < -retorno[max(1,(i-1))] + (Gearing[1] -
Gearing[2]) * ((TransacPrice/AverPrice[1]) -1)
        if(D[i] > epsilon \& E[i] > epsilon \& (i <= length(E))){
```

```
# Condição para decidir o quanto entramos na
posição
               if(D[i] > (1.5*epsilon) \& E[i] > (1.5*epsilon) \&
(i \le length(E)))
                  Gearing[2]<-Gearing[1]+1;</pre>
                  TransacPrice<- (-a[i+N]);</pre>
              else{
                   Gearing[2] < -Gearing[1] + (1/2);
                   TransacPrice<- (-a[i+N]);</pre>
  #cálculo do preço médio
if(abs(Gearing[2]) < abs(Gearing[1]) & & Gearing[1] * Gearing[2] > 0) {
                  AverPrice[1] <- AverPrice[2];</pre>
               }
if(abs(Gearing[2])>abs(Gearing[1])&&Gearing[1]*Gearing[2]>0&&
AverPrice[2]!=0){
                  AverPrice[1] <- AverPrice[2];</pre>
                  AverPrice[2] < - Gearing[2] / ( ((Gearing[2] -
Gearing[1])/TransacPrice)+(Gearing[1]/AverPrice[1]));
               if (Gearing[1] *Gearing[2] < 0 | | Gearing[1] == 0) {</pre>
                  AverPrice[1]<-AverPrice[2];</pre>
                  AverPrice[2]<-TransacPrice;</pre>
               if (Gearing[2] == 0) {
                  AverPrice[1] <- AverPrice[2];
                  AverPrice[2]<-0;
               }
               #cálculo do retorno
               if (Gearing[1] *Gearing[2] < 0 & & AverPrice[1]!=0) {</pre>
                 retorno[i] <- retorno[max(1,(i-
1))]+Gearing[1]*((TransacPrice/AverPrice[1])-1);
               if (Gearing[1] *Gearing[2] >= 0 && AverPrice[1]!=0) {
                 retorno[i] < -retorno[max(1,(i-1))] + (Gearing[1] - 
Gearing[2]) * ((TransacPrice/AverPrice[1]) -1)
  cat("Position",i,", Gearing ",Gearing[2], ", TransacPrice
",TransacPrice, ", AverPrice
", AverPrice[2], ", LOSS", CountStopLoss, ", PROFIT", CountStopProfit, f
ill=TRUE)
  TransacPrice<-0;
  i<-i+1;}
```

Bibliografia

[1] M. Cambou. Algorithmic Trading with High-Frequency Data. Dissertação para grau de Master of Philosophy, University of Cambridge, 2010.
[2] M. Dacorogna, R. Gencay, U. Müller, R. Olsen e O. Pictet. An introduction to High Frequency Finance. Academic Press, San Diego, CA, 2001.
[3] P. A. Morettin, Econometria Financeira. Editora Blucher, São Paulo, 2008.
[4] J. Hasbrouck. Empirical Market Microstructure. Oxford University Press, New York, 2007.
[5] U. Jaekle, E. Tomasini. Trading Systems. Harriman House, Great Britain, 2009.
[6] I. Aldridge. High-Frequency Trading. John Wiley & Sons, New Jersey, 2010.
[7] M. Fernandes, J.A. Matos. Testing the Markov property with high frequency data.