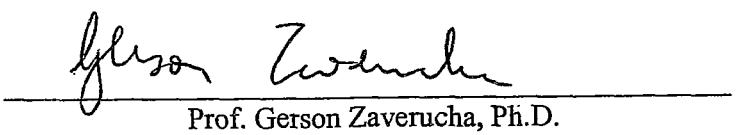
REDES BAYESIANAS DINÂMICAS PARA PREVISÃO DE SÉRIES TEMPORAIS: APLICAÇÃO AO SETOR ELÉTRICO

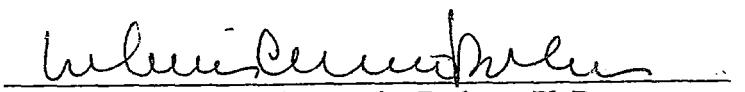
Marcelo Andrade Teixeira

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

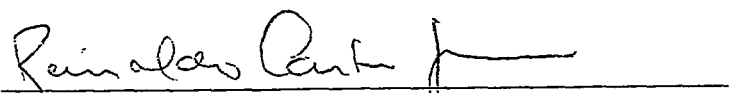
Aprovada por:



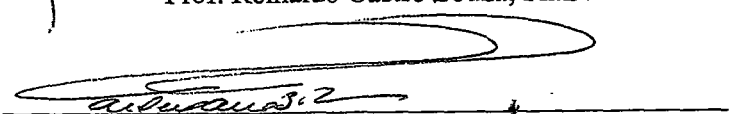
Prof. Gerson Zaverucha, Ph.D.



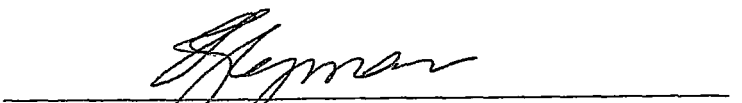
Prof. Valmir Carneiro Barbosa, Ph.D.



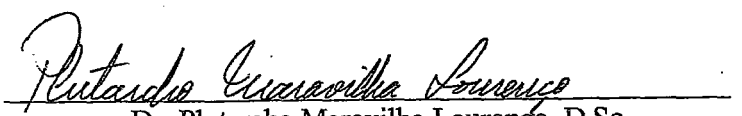
Prof. Reinaldo Castro Souza, Ph.D.



Prof. Marley Maria Bernardes Rebuzzi Vellasco, Ph.D.



Prof. Fábio Gagliardi Cozman, Ph.D.



Dr. Plutarcho Maravilha Lourenço, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

MAIO DE 2005

TEIXEIRA, MARCELO ANDRADE

Redes Bayesianas Dinâmicas para Previsão de Séries Temporais: Aplicação ao Setor Elétrico [Rio de Janeiro] 2005

VII, 119 p. 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia de Sistemas e Computação, 2005)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Arquiteturas de Redes Bayesianas e Redes Bayesianas Dinâmicas Aplicadas na Previsão de Séries Temporais

I. COPPE/UFRJ II. Título (série)

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

REDES BAYESIANAS DINÂMICAS PARA PREVISÃO DE SÉRIES TEMPORAIS: APLICAÇÃO AO SETOR ELÉTRICO

Marcelo Andrade Teixeira

Maio/2005

Orientador: Gerson Zaverucha

Programa: Engenharia de Sistemas e Computação

Este trabalho desenvolve uma abordagem de previsão de séries temporais para a qual não existem muitos trabalhos na literatura: a previsão de um valor contínuo através da estimação não-paramétrica da função densidade de probabilidade da variável aleatória contínua que se deseja prever. Para essa estimação não-paramétrica empregamos Redes Bayesianas e Redes Bayesianas Dinâmicas com variáveis aleatórias discretas através da discretização dos dados contínuos. Dessa forma criamos vários sistemas de previsão: *Markov Model for Regression*, *Hidden Markov Model for Regression* e *Multi-Hidden Markov Model for Regression*. A principal contribuição deste trabalho foi a generalização desses sistemas pelo uso da fuzzificação no lugar da discretização: *Fuzzy Bayes Predictor*, *Fuzzy Markov Predictor*, *Fuzzy Hidden Markov Predictor* e *Fuzzy Multi-Hidden Markov Predictor*. Também desenvolvemos métodos para efetuar o particionamento do espaço de dados contínuos a fim de serem usados por nossos sistemas que fazem fuzzificação. Nossos sistemas foram aplicados às tarefas de previsão *single-step* e *multi-step* de séries de carga elétrica mensal. As séries temporais empregadas apresentam um comportamento de mudança abrupta e significativa em seus últimos anos, como quando ocorre um racionamento de energia. Obtivemos resultados competitivos quando comparamos com várias técnicas estatísticas conhecidas.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

DYNAMIC BAYESIAN NETWORKS FOR TIME SERIES FORECASTING: APPLICATION TO THE ELECTRIC SECTOR

Marcelo Andrade Teixeira

May/2005

Advisor: Gerson Zaverucha

Department: Systems Engineering and Computer Science

This work presents an approach for time series prediction for which there are not many works in the literature: the prediction of a continuous value through the estimation of the probability density function of the continuous random variable meant to be predicted. For this nonparametric estimation we employed Bayesian Networks and Dynamic Bayesian Networks with discrete random variables by the discretization of the continuous data. This way we created several systems for prediction: Markov Model for Regression, Hidden Markov Model for Regression and Multi-Hidden Markov Model for Regression. The main contribution of this work was the generalization of these systems by the use of fuzzification instead of discretization: Fuzzy Bayes Predictor, Fuzzy Markov Predictor, Fuzzy Hidden Markov Predictor e Fuzzy Multi-Hidden Markov Predictor. We also developed some methods to make the partitioning of the space of continuous data in order to be used by our systems that make fuzzification. Our systems were applied to the tasks of single-step and multi-step forecasting of monthly electric load series. The employed time series present a sudden significant changing behavior at their last years, as it occurs in an energy rationing. We have obtained competitive results when compared with several known statistics techniques.

Sumário

1 - Definição do Problema de Previsão de Carga Elétrica

Este capítulo define precisamente o problema a ser tratado: a previsão de carga elétrica e, de forma mais geral, a previsão de séries temporais. No final é apresentada uma organização geral dos capítulos seguintes.

1.1 - Previsão de Séries Temporais

Uma série temporal consiste de uma seqüência de valores medidos (ou gerados) através do tempo. Na Figura 1 é mostrada uma série que representa o consumo trimestral de carvão por usuários finais no Reino Unido (dados obtidos de [17]).

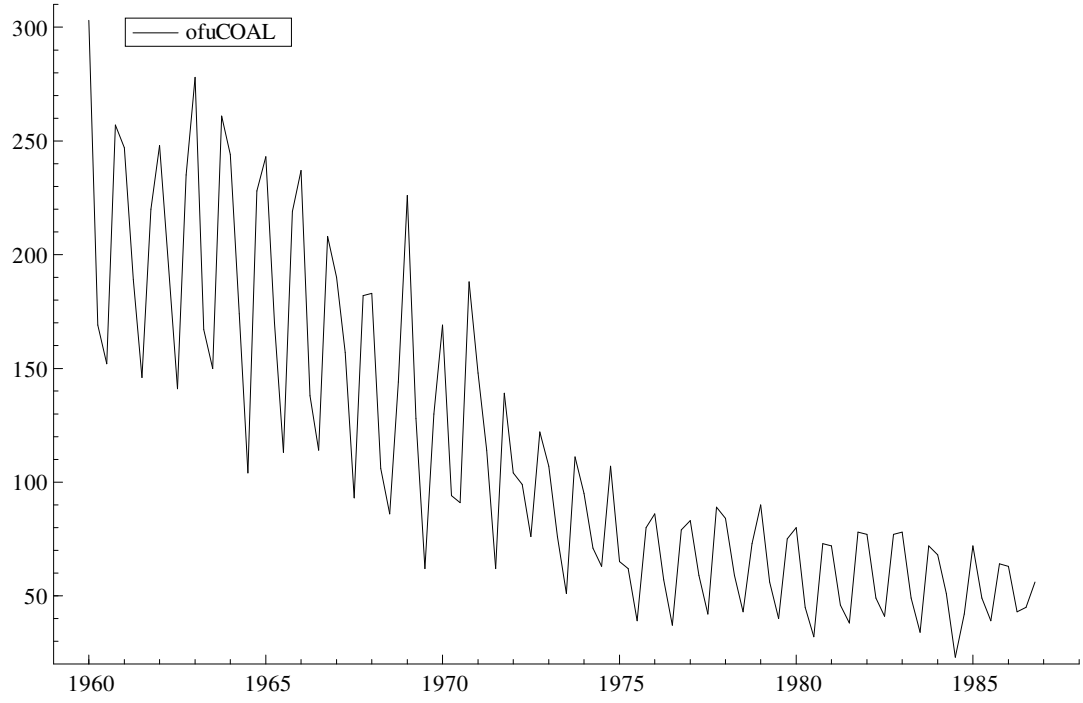


Figura 1: Consumo trimestral de carvão por usuários finais no Reino Unido.

O problema de previsão de uma série temporal consiste em conseguir prever valores futuros da série usando como informação valores passados dessa série. Esta informação passada deve ser empregada na construção de um modelo de previsão. Uma vez obtido

esse modelo, podemos utilizá-lo para fazer predições de valores da série que não foram usados em sua construção e, dessa forma, avaliar o desempenho do modelo. Assim, se temos Z_1, Z_2, ... Z_T como uma série temporal de tamanho T para a obtenção de um modelo de previsão, então valores $Z_{T+\tau}$ para $\tau \geq 1$ serão usados para avaliação deste modelo [51], [52]. Na série da Figura 1, poderíamos, por exemplo, utilizar os dados disponíveis até o ano de 1980 para construir um modelo, e os dados restantes seriam usados para julgar seu desempenho.

Existem várias técnicas, estatísticas e de inteligência computacional, que podem ser usadas para a construção de um modelo de previsão. Essas técnicas possuem seus pontos positivos e negativos. As técnicas estatísticas ([36], [3], [17], [74]) permitem não só uma previsão pontual da série (ou seja, a previsão de um valor Z_t da série para um instante t específico) mas também a obtenção de um intervalo de confiança para essa previsão. Entretanto, essas técnicas estatísticas se restringem a famílias específicas de distribuições paramétricas (por exemplo, Gaussianas) para modelar os dados. Isso implica em fazer fortes suposições sobre a natureza dos dados; se essas suposições não se concretizarem o modelo obtido pode não ser uma boa aproximação para a série temporal. Outra restrição geralmente colocada é a pressuposição de que os relacionamentos entre as variáveis envolvidas no modelo sejam lineares, o que pode não corresponder à realidade dos dados. No caso das técnicas de inteligência computacional ([18], [31], [77]) não existe nenhum comprometimento de que os dados possuam uma determinada distribuição paramétrica. Além disso, permitem o tratamento da não-linearidade presente nos dados. Todavia, as técnicas de inteligência computacional são limitadas apenas às previsões pontuais sem o conhecimento de intervalos de confiança.

Uma possível abordagem para se prever uma série temporal ou, de modo mais geral, qualquer valor contínuo (regressão) é através da estimação da função densidade de probabilidade da variável aleatória contínua que se deseja prever [11]. Uma vez de posse dessa função densidade então temos que a previsão é dada pelo valor esperado da variável aleatória contínua (o que corresponde a calcular a média da distribuição; outras alternativas seriam calcular a moda ou a mediana da distribuição). A estimação da densidade, ou seja, o ajuste de uma função densidade aos dados [49], pode ser paramétrica ou não-paramétrica. No caso do ajuste paramétrico, é necessária a especificação da forma da densidade (por exemplo, Gaussiana) e a estimação de seus parâmetros. Já o ajuste não-paramétrico (por exemplo, histogramas [49]) fornece um algoritmo consistente para quase qualquer densidade contínua e evita o passo de

especificação. Se considerarmos o histograma para o ajuste não-paramétrico, vemos que a estimação da densidade contínua é feita pela discretização dos dados: para cada valor contínuo existe um correspondente valor discreto representando o intervalo que contém o valor contínuo. A metodologia paramétrica foi amplamente usada para previsão [17], [74] enquanto que a não-paramétrica foi pouco desenvolvida [11] nesse campo.

Nosso objetivo foi explorar essa abordagem de previsão não-paramétrica usando fuzzificação [31], [72] como uma generalização da discretização. Fuzzificação é utilizada na técnica de inteligência computacional conhecida como sistemas *fuzzy* [31], [72] para efetuar o particionamento do espaço de dados contínuos (de forma similar à discretização). Além desse particionamento, a estimação da densidade contínua faz uso do cálculo de probabilidades discretas na forma de Redes Bayesianas (RB) [47] e Redes Bayesianas Dinâmicas (RBD) [15], [44], [47] (duas outras técnicas de inteligência computacional). Assim estendemos o trabalho desenvolvido nesse campo pelo uso de probabilidades *fuzzy* [69], [75] e não nos limitamos apenas às RB's (como em [11]) empregando também RBD's, tudo isso para previsão de séries temporais. Na literatura, RB's e RBD's [17], [74], [47] com apenas variáveis aleatórias contínuas representam a metodologia paramétrica para previsão através do valor esperado.

1.2 - Previsão de Carga Elétrica

O problema de previsão de carga elétrica é de grande importância na área de sistemas de potência. Através de sua resolução pode-se prover uma operação econômica e segura do sistema elétrico e fornecer informações para o planejamento e expansão do mesmo. Dois problemas comuns são a previsão de carga horária, na qual os valores de carga elétrica estão disponíveis em intervalos de uma hora, e a previsão de carga mensal, onde os valores de carga são apresentados em intervalos de um mês [19], [29], [27], [50], [76]. A Figura 2 mostra uma série de carga elétrica mensal (dados provenientes de uma empresa brasileira de energia elétrica).

Séries de carga elétrica mensal possuem uma tendência de crescimento pois refletem a evolução do consumo de carga elétrica ao longo de mais de uma década. Além disso, séries de carga mensal possuem sazonalidade, isto é, uma repetição periódica definida, visto que séries de consumo sofrem influências das mudanças de estação. Na Figura 3 é

apresentada uma série do consumo trimestral de eletricidade por usuários finais no Reino Unido (dados de [17]), onde o efeito da sazonalidade é bem evidente.

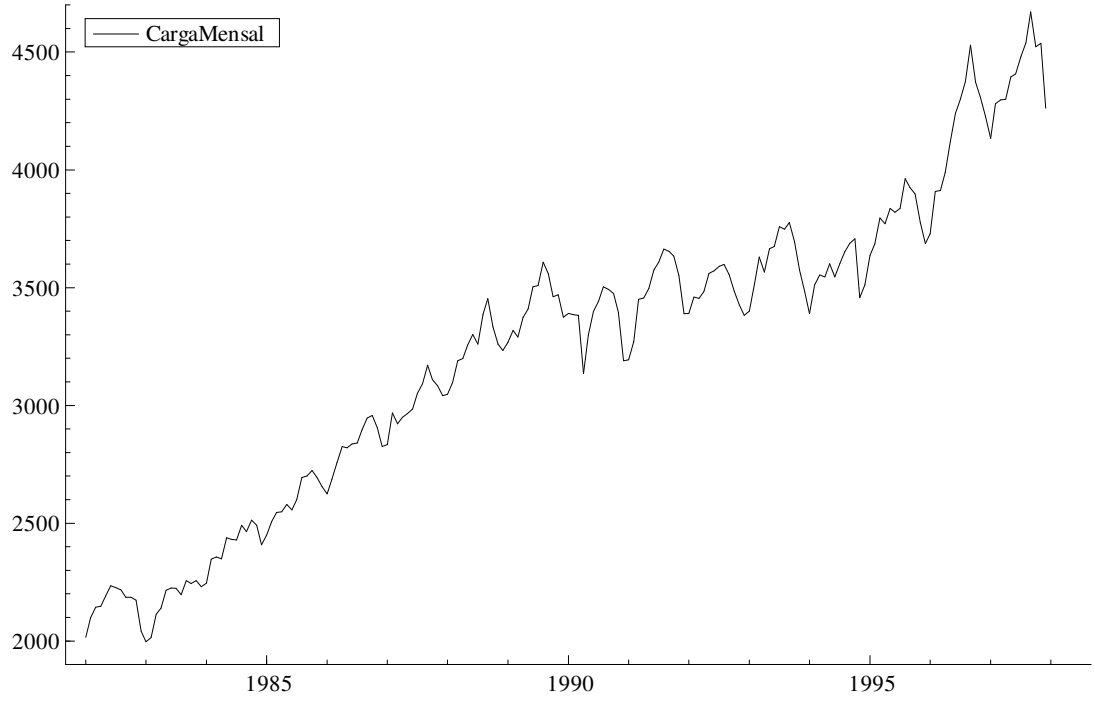


Figura 2: Uma série de carga elétrica mensal.

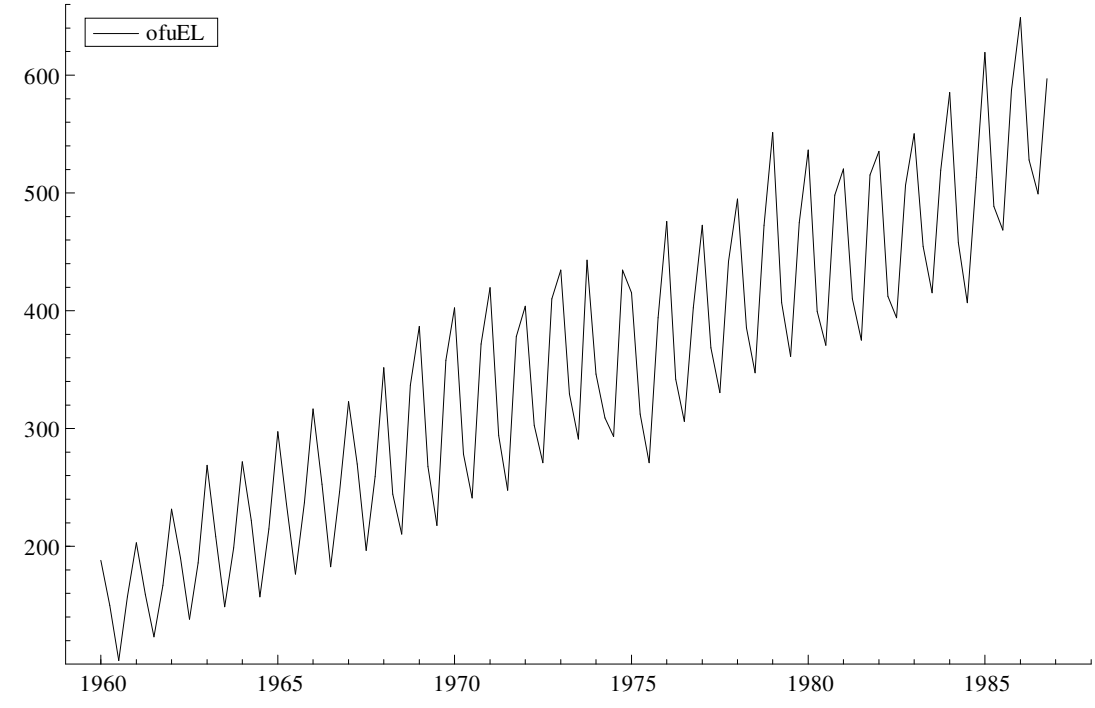


Figura 3: Consumo trimestral de eletricidade por usuários finais no Reino Unido.

1.3 - Organização da Dissertação

Como já foi mencionado, o objetivo de nosso trabalho foi explorar uma abordagem de previsão de séries temporais pouco investigada na literatura: a previsão de um valor contínuo através da estimação não-paramétrica da função densidade de probabilidade da variável aleatória contínua que se deseja prever. Essa abordagem já utilizou RB's com variáveis discretas para a estimação da densidade contínua mas a generalizamos para o uso de variáveis *fuzzy* além de também empregarmos RBD's. A expressão "previsão de um valor contínuo" significa prever um valor para uma variável aleatória contínua; do mesmo modo, "previsão de um valor discreto" significaria prever um valor para uma variável aleatória discreta.

O Capítulo 2 apresenta algumas das técnicas disponíveis, estatísticas e de inteligência computacional, para efetuar a previsão de séries temporais: métodos de amortecimento exponencial, modelos de Box & Jenkins, modelos de espaço de estados, redes neurais artificiais, sistemas *fuzzy* e sistemas inteligentes híbridos.

O Capítulo 3 descreve duas técnicas de inteligência computacional, RB's e RBD's, que são necessárias para os cálculos de probabilidades discretas na estimação não-paramétrica de uma densidade contínua. Aqui damos ênfase às descrições do *Naive Bayes Classifier* (NBC) [10] e do *Hidden Markov Model* (HMM) [40], dois modelos específicos de RB e RBD, respectivamente.

O Capítulo 4 mostra sistemas de previsão que fazem uso da estimação não-paramétrica da densidade contínua: *Naive Bayes for Regression* (NBR) [11], *Markov Model for Regression* (MMR) [62], [65], *Hidden Markov Model for Regression* (HMMR) [65] e *Multi-Hidden Markov Model for Regression* (MHMMR) [67]. O NBR é baseado no NBC enquanto que os demais foram desenvolvidos para essa tese e são baseados no HMM.

O Capítulo 5 apresenta os sistemas de previsão que desenvolvemos usando fuzzificação na estimação não-paramétrica da densidade contínua: *Fuzzy Bayes Predictor* (FBP) [60], [64], *Fuzzy Markov Predictor* (FMP) [61], [63], [64], *Fuzzy Hidden Markov Predictor* (FHMP) [65] e *Fuzzy Multi-Hidden Markov Predictor* (FMHMP) [67]. O FBP é baseado no NBC enquanto que os demais são baseados no HMM.

No Capítulo 6 são vistos vários métodos para efetuar o particionamento do espaço de dados contínuos. São discutidos tanto métodos para discretização como para fuzzificação.

O Capítulo 7 mostra os resultados obtidos pela aplicação dos modelos desenvolvidos a casos reais (séries de carga elétrica mensal), comparando-os com várias técnicas estatísticas.

O Capítulo 8 finaliza o trabalho, apontando suas contribuições. Também são discutidos alguns tópicos que estão sendo atualmente investigados ou que são importantes para uma futura pesquisa.

2 - Técnicas para Previsão de Carga Elétrica

Este capítulo faz um levantamento das técnicas disponíveis para efetuar a previsão de carga elétrica.

2.1 - Técnicas para Previsão de Séries Temporais

Existem várias técnicas, estatísticas e de inteligência computacional, que podem ser usadas para a construção de um modelo de previsão de uma série temporal e, particularmente, de uma série de carga elétrica [4], [73], [19], [29], [27], [50], [76]. Dentre as técnicas estatísticas destacam-se os métodos de amortecimento exponencial [36], os modelos de Box & Jenkins [3] e os modelos de espaço de estados (modelos estruturais [17] e modelos lineares dinâmicos [74]). No caso das técnicas de inteligência computacional destacam-se as redes neurais artificiais [18], [77], os sistemas *fuzzy* [31] e sistemas inteligentes híbridos.

Tanto os métodos de amortecimento exponencial quanto os modelos de espaço de estados pressupõem que a série temporal pode ser decomposta em componentes que possuem interpretação direta, como por exemplo a tendência e a sazonalidade. Uma série de carga elétrica mensal poderia ser representada da seguinte forma:

Série = Tendência + Sazonalidade + Irregular,

onde "Irregular" é a componente que resta ao retirarmos a tendência e a sazonalidade da série, ou seja, o comportamento da série que não é explicado pela tendência nem pela sazonalidade é considerado como uma componente irregular. Para se prever qual é o valor da série em um determinado instante, devemos saber quais os valores dessas componentes nesse instante. Entretanto, como essas componentes não são observáveis (ou seja, não existem de forma explícita nos dados) devemos estimá-las a partir dos dados disponíveis (se estamos prevendo o valor da série no instante t então nossos dados disponíveis são os valores da série até o instante t-1). Além disso, essas estimativas das componentes são atualizadas a cada instante empregando as estimativas calculadas no instante anterior e a nova observação (valor da série) disponível. O que difere os métodos de amortecimento exponencial dos modelos de espaço de estados é o modo pelo qual são feitas essas atualizações.

Nos modelos de Box & Jenkins a previsão do valor da série em um instante qualquer é uma função linear de valores da série em instantes anteriores, de previsões passadas e de erros de previsão (erro de previsão é o valor da série menos o valor previsto). Os coeficientes dessa função linear são estimados a partir dos dados. Ao contrário das componentes dos métodos de amortecimento exponencial e modelos de espaço de estados que são atualizados a cada instante, esses coeficientes são constantes. Por constantes queremos dizer que esses coeficientes não podem ser estimados usando estimativas passadas e a nova observação disponível. Para atualizar os coeficientes é necessário estimá-los novamente usando os dados anteriores mais a nova observação.

Redes neurais artificiais e sistemas *fuzzy* podem ser usados como técnicas gerais para aproximação de funções. Assim, queremos aproximar uma função desconhecida que tem como entradas x_1, x_2, ..., x_k e como saída y. Uma maneira simples de se prever uma série temporal empregando esses aproximadores de funções é definir a saída y como o valor da série a ser previsto em um instante qualquer e as entradas x_1, x_2, ..., x_k como valores da série em instantes anteriores. Redes neurais artificiais utilizam várias funções não-lineares intercaladas para aproximar a função $[x_1, x_2, ..., x_k] \rightarrow [y]$. De modo geral uma rede neural artificial representa uma complexa função não-linear cujos parâmetros são estimados a partir dos dados. Esses parâmetros, da mesma forma que os coeficientes nos modelos de Box & Jenkins, são constantes (ou seja, não são atualizados a cada instante). Nos sistemas *fuzzy* a função $[x_1, x_2, ..., x_k] \rightarrow [y]$ é aproximada pelo uso de várias regras que fornecem o mapeamento de entrada-saída. Essas regras são inferidas a partir dos dados.

2.2 - Amortecimento Exponencial

Seja Z_1, Z_2, ... Z_T uma série temporal de tamanho T. Os métodos de amortecimento exponencial (*exponential smoothing*) [36], [51] assumem que a série pode ser descrita pelo modelo

$$Z_t = f(t) + \varepsilon_t$$

onde ε_t é um ruído com média zero e variância constante, e f(t) é uma função que incorpora a informação sobre o nível médio da série (com ou sem tendência) e sua sazonalidade. Dois exemplos simples de nível médio podem ser vistos na Figura 4 (em ambos os casos não existe sazonalidade):

- nível médio constante (sem tendência): a série fica oscilando aleatoriamente ao redor de um valor constante a_1, ou seja $f(t) = a_1$
- nível médio linear (com tendência): a série fica oscilando aleatoriamente ao redor de uma reta, ou seja $f(t) = a_1 + a_2 t$ e, como a tendência é de crescimento temos $a_2 > 0$.

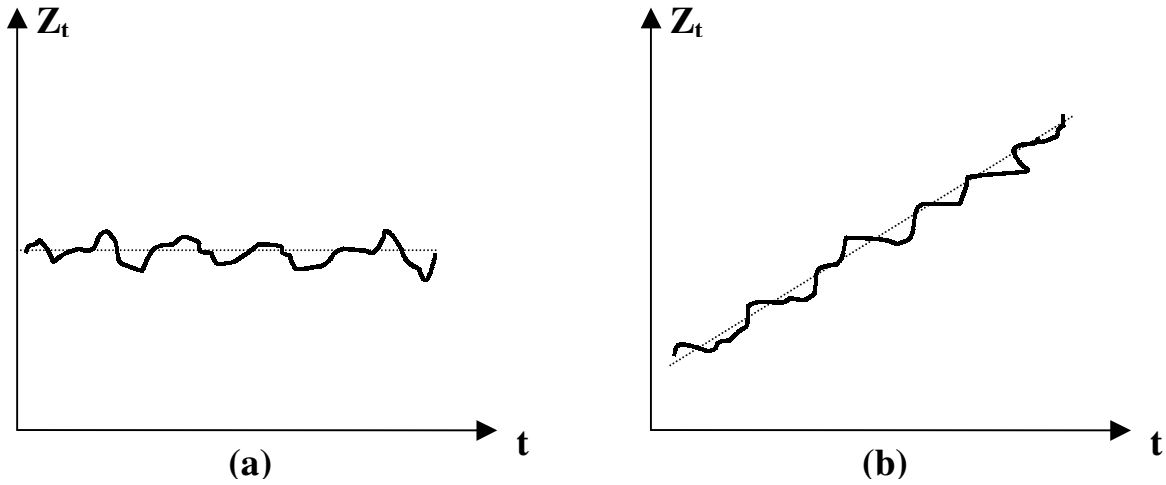


Figura 4: Exemplos de nível médio constante (a) e linear (b).

Nesses exemplos percebe-se que a função f(t) é composta de parâmetros (a_1 e a_2) que são desconhecidos. Assim eles devem ser estimados a partir dos dados.

A equação do modelo para um valor $Z_{T+\tau}$ ainda não observado da série (onde $\tau > 0$) é dada por

$$Z_{T+\tau} = f(T + \tau) + \varepsilon_{T+\tau}.$$

Como essa é uma equação estocástica (já que $\varepsilon_{T+\tau}$ é uma variável aleatória), a previsão para o valor de $Z_{T+\tau}$ pode ser dada pela média da variável aleatória $Z_{T+\tau}$. Por sua vez a média de uma variável aleatória X é igual ao valor esperado $E\{.\}$ dessa variável. Como existem parâmetros desconhecidos no modelo, emprega-se o valor esperado condicional aos valores da série já conhecidos Z_1, Z_2, ... Z_T. Logo, a previsão de $Z_{T+\tau}$ feita no instante T é dada por

$$\hat{Z}_T(\tau) = E\{Z_{T+\tau} \mid Z_1, Z_2, \ldots Z_T\}.$$

Essa previsão envolverá o cálculo de estimadores dos parâmetros que aparecem em f(t). As duas seções seguintes descrevem o modo pelo qual esses estimadores são obtidos para diferentes funções f(t) e, portanto, para modelar séries com diferentes características:

- modelar apenas o nível médio da série (para séries sem sazonalidade)
- modelar o nível médio da série e sua sazonalidade

2.2.1 - Séries Não-Sazonais

Assume-se que uma série não-sazonal (isto é, que não possui sazonalidade) pode ser descrita por

$Z_t = \mu(t) + \varepsilon_t$

onde $\mu(t)$ é uma função que representa o nível médio da série. Por exemplo, se a série não tem tendência (modelo constante) então $\mu(t) = a_1$, onde a_1 é o parâmetro que representa o nível constante da série. A equação de previsão para este modelo constante é dada por

$$\hat{Z}_T(\tau) = E\{Z_{T+\tau} | Z_1, \ldots Z_T\} = E\{a_1 + \varepsilon_{T+\tau} | Z_1, \ldots Z_T\} = \hat{a}_1(T)$$

onde $\hat{a}_1(T)$ é o estimador do parâmetro a_1 no instante T. Diz-se então que a previsão τ-passos-à-frente no instante T é igual a $\hat{a}_1(T)$.

Sabendo-se a equação de previsão sendo empregada, devemos obter os estimadores dos parâmetros envolvidos nessa equação. No caso do modelo constante, uma maneira simples seria pelo método de médias móveis:

$$\hat{a}_1(T) = M_T = (Z_{T-N+1} + \ldots + Z_{T-1} + Z_T)/N,$$

ou seja, a média aritmética dos últimos N valores da série. M_T é chamada de média móvel de tamanho N calculada no instante T, e sua expressão pode ser reescrita de forma recursiva:

$$M_T = M_{T-1} + (Z_T - Z_{T-N})/N,$$

onde M_{T-1} é a média móvel de tamanho N calculada em T-1. N é uma constante normalmente escolhida de forma a minimizar a soma dos quadrados dos erros de previsão 1-passo-à-frente. De modo geral, o erro de previsão τ-passos-à-frente é dado por $e_\tau(t) = Z_t - \hat{Z}_{t-\tau}(\tau)$

onde Z_t é o valor da série no instante t, $\hat{Z}_{t-\tau}(\tau)$ é o valor previsto para o instante t conhecendo-se apenas os valores Z_1, ..., $Z_{t-\tau}$.

A equação de previsão para o modelo constante empregando-se o método de médias móveis é dada por

$\hat{Z}_T(\tau) = M_T$, $\tau = 1, 2, \ldots$

Isso significa que as previsões feitas no instante T para os valores da série Z_{T+1}, Z_{T+2}, ..., serão todas iguais ao valor M_T.

O cálculo de $\hat{a}_1(T)$ pelo método de amortecimento exponencial pode ser obtido através de uma pequena modificação no cálculo feito pelo método de médias móveis. A equação de previsão feita no instante T-1 é dada por

$$\hat{Z}_{T-1}(\tau) = M_{T-1},\ \tau = 1, 2, ...$$

significando que as previsões para os valores da série Z_T, Z_{T+1}, ..., serão todas iguais ao valor M_{T-1} que é a média dos valores Z_{T-N}, ..., Z_{T-2}, Z_{T-1}. Logo, M_{T-1} é um estimador dos valores Z_T, Z_{T+1}, ..., e também poderia ser utilizado como estimador dos valores Z_{T-N}, ..., Z_{T-2}, Z_{T-1}. Considerando-se M_{T-1} o estimador de Z_{T-N}, fazemos a substituição na fórmula recursiva de médias móveis obtendo

$$M_T = M_{T-1} + (Z_T - M_{T-1})/N.$$

Colocando-se M_{T-1} em evidência:

$$M_T = \frac{1}{N} Z_T + \left(1 - \frac{1}{N}\right) M_{T-1}.$$

Renomeando $\frac{1}{N}$ como α, M_T e M_{T-1} como S_T e S_{T-1}, respectivamente:

$$S_T = \alpha . Z_T + (1-\alpha) S_{T-1},$$

onde S_T é o estimador amortecido da série temporal e α é a constante de amortecimento. Assim, nosso estimador do parâmetro passa a ser $\hat{a}_1(T) = S_T$. Normalmente o valor inicial do estimador amortecido S_0 é igual à média aritmética dos primeiros valores da série. A constante α pode ser escolhida de modo similar ao critério usado no método de médias móveis para N.

Fazendo substituições sucessivas na fórmula recursiva de S_T obtemos

$$S_T = \alpha . Z_T + \alpha (1-\alpha) Z_{T-1} + \alpha (1-\alpha)^2 Z_{T-2} + ... + \alpha (1-\alpha)^{T-1} Z_1 + (1-\alpha)^T S_0$$

onde percebe-se que os pesos $\alpha(1 - \alpha)^{T-t}$ associados a cada Z_t decrescem exponencialmente com a idade das observações. Por isso o método é chamado de amortecimento exponencial. No método de médias móveis todas as observações possuíam um mesmo peso (1/N) enquanto que no amortecimento exponencial quanto mais recente for a observação maior será o peso a ela associado.

Além do modelo constante, temos o modelo linear com $\mu(t) = a_1 + a_2 t$, onde a_1 é o parâmetro que representa o nível da série enquanto a_2 representa a inclinação. A equação de previsão para este modelo é dada por

$$\hat{Z}_T(\tau) = E\{Z_{T+\tau} | Z_1, \ldots Z_T\} = E\{a_1 + a_2.[T+\tau] + \varepsilon_{T+\tau} | Z_1, \ldots Z_T\} = \hat{a}_1(T) + \hat{a}_2(T).[T+\tau]$$

onde $\hat{a}_1(T)$ e $\hat{a}_2(T)$ são os estimadores dos parâmetros a_1 e a_2 no instante T, respectivamente. O cálculo desses estimadores empregam as seguintes fórmulas: para o nível temos

$$\hat{a}_1(T) = 2.S_T - S_T^{[2]} - \hat{a}_2(T).T,$$

e para a inclinação

$$\hat{a}_2(T) = \frac{\alpha}{1-\alpha}[S_T - S_T^{[2]}],$$

onde S_T e $S_T^{[2]}$ são obtidos por amortecimento exponencial usando as equações

$$S_T = \alpha.Z_T + (1-\alpha)S_{T-1}$$

$$S_T^{[2]} = \alpha.S_T + (1-\alpha)S_{T-1}^{[2]}.$$

Também existe o modelo quadrático com

$\mu(t) = a_1 + a_2 t + a_3 t^2$.

Aqui o cálculo dos estimadores desses três parâmetros envolverá o uso de mais um estimador amortecido exponencialmente:

$$S_T^{[3]} = \alpha.S_T^{[2]} + (1-\alpha)S_{T-1}^{[3]}.$$

O conjunto composto pelos modelos vistos anteriormente (constante, linear e quadrático) e seus respectivos estimadores de parâmetros amortecidos exponencialmente é conhecido como o método de amortecimento exponencial de Brown.

Além do método de Brown existe outro método de amortecimento exponencial no qual se emprega apenas o modelo linear. Esse método é chamado de Holt-2-parâmetros e os estimadores dos parâmetros de sua equação de previsão,

$$\hat{Z}_T(\tau) = \hat{a}_1(T) + \hat{a}_2(T).\tau$$

(aqui é feita uma translação da origem de tal forma que t = 0 coincida com o instante T, simplificando a equação de previsão e o cálculo dos estimadores de seus parâmetros), são obtidos por amortecimento exponencial da seguinte forma:

$$\hat{a}_1(T) = \alpha Z_T + (1-\alpha).[\hat{a}_1(T-1) + \hat{a}_2(T-1)]$$

$$\hat{a}_2(T) = \beta[\hat{a}_1(T) - \hat{a}_1(T-1)] + (1-\beta)\hat{a}_2(T-1)$$

onde α e β são duas constantes de amortecimento. A fórmula para estimação do nível a_1 representa a combinação linear de Z_T e $\hat{Z}_{T-1}(1) = \hat{a}_1(T-1) + \hat{a}_2(T-1)$, onde $\hat{Z}_{T-1}(1)$ é a previsão de Z_T feita no instante T-1. Empregando-se as estimativas do nível em dois

instantes consecutivos obtemos uma estimativa da inclinação pela diferença dessas estimativas. A fórmula para estimação da inclinação a_2 representa a combinação linear dessa diferença e da estimativa da inclinação feita no instante anterior.

Também pode-se fazer um amortecimento da previsão, conhecido por *damped trend*, através da seguinte equação de previsão:

$$\hat{Z}_T(\tau) = \hat{a}_1(T) + \left[\sum_{j=1}^{\tau} \varphi^{j-1}\right] \cdot \hat{a}_2(T)$$

onde $0 \leq \varphi \leq 1$ é a constante de amortecimento.

2.2.2 - Séries Sazonais

Assume-se que uma série sazonal (isto é, que possui sazonalidade) pode ser descrita por um modelo aditivo,

$Z_t = \mu(t) + \rho_t + \varepsilon_t$,

ou um modelo multiplicativo,

$Z_t = \mu(t).\rho_t + \varepsilon_t$,

onde $\mu(t)$ é uma função que representa o nível médio da série e ρ_t representa sua sazonalidade. Pela Figura 5 percebe-se que um modelo aditivo é adequado para uma série de variância constante enquanto que o modelo multiplicativo é adequado para uma série cuja variância está crescendo com o nível da série.

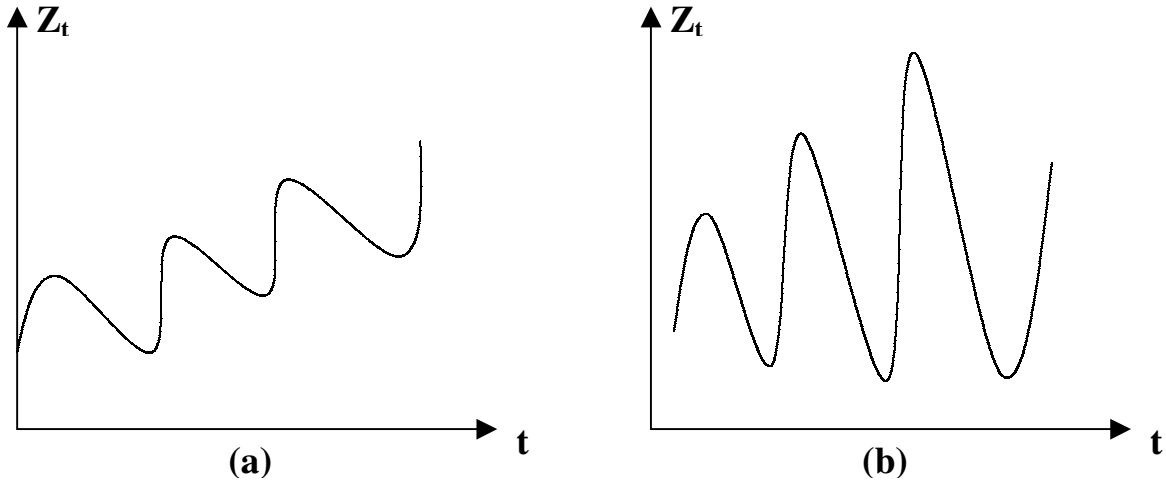


Figura 5: Exemplos dos modelos aditivo (a) e multiplicativo (b).

O comprimento do ciclo sazonal (isto é, o comprimento dessa repetição periódica chamada sazonalidade) é denotado por "S". Dessa forma, S = 12 para séries mensais, S = 4 para séries trimestrais, S = 52 para séries semanais, etc. Aqui a modelagem da sazonalidade será feita através de fatores sazonais (representados por ρ_1, ρ_2, ... ρ_S), que são valores que caracterizam cada mês, ou trimestre, ou semana dentro de um mesmo período sazonal.

O método de amortecimento exponencial que emprega um modelo linear sazonal (aditivo ou multiplicativo) é conhecido como método de Winters. Para o modelo linear sazonal multiplicativo

$Z_t = (a_1 + a_2 t).\rho_t + \varepsilon_t$

teremos a seguinte equação de previsão (feita uma translação da origem):

$$\hat{Z}_T(\tau) = [\hat{a}_1(T) + \hat{a}_2(T).\tau].\hat{\rho}_{m(T+\tau)}(T)$$

onde $m(t) \in \{1, 2, ... S\}$ é o mês ou trimestre ou semana correspondente ao instante t. Desta maneira $\rho_{m(t)}$ é o fator sazonal correspondente ao instante t. Os estimadores $\hat{a}_1(T), \hat{a}_2(T), \hat{\rho}_1(T), \hat{\rho}_2(T), ...\hat{\rho}_S(T)$ são calculados por amortecimento exponencial da seguinte forma:

$$\hat{a}_1(T) = \alpha\left[\frac{Z_T}{\hat{\rho}_{m(T)}(T-1)}\right] + (1-\alpha).[\hat{a}_1(T-1) + \hat{a}_2(T-1)]$$

$$\hat{a}_2(T) = \beta[\hat{a}_1(T) - \hat{a}_1(T-1)] + (1-\beta)\hat{a}_2(T-1)$$

$$\hat{\rho}_{m(T)}(T) = \gamma\left[\frac{Z_T}{\hat{a}_1(T)}\right] + (1-\gamma).\hat{\rho}_{m(T)}(T-1)$$

$$\hat{\rho}_j(T) = \hat{\rho}_j(T-1) \;;\; j = 1, 2, ... S \;;\; j \neq m(T)$$

onde α, β e γ são três constantes de amortecimento. Depois os estimadores dos fatores sazonais são normalizados de forma que $\sum_{j=1}^{S} \hat{\rho}_j(T) = S$. As fórmulas para as estimações do nível a_1 e da inclinação a_2 são as mesmas que as do método de Holt-2-parâmetros com uma pequena correção em Z_T por causa do efeito sazonal (Z_T dividido por seu fator sazonal correspondente é o valor da série dessazonalizado no instante T). A fórmula para estimação de cada fator sazonal representa a combinação linear da observação corrente sem a tendência (Z_T dividido pela estimativa corrente do nível) e da estimativa do respectivo fator sazonal feita a S instantes anteriores.

Para o modelo linear sazonal aditivo

$Z_t = a_1 + a_2 t + \rho_t + \varepsilon_t$

teremos a seguinte equação de previsão (feita uma translação da origem):

$$\hat{Z}_T(\tau) = \hat{a}_1(T) + \hat{a}_2(T).\tau + \hat{\rho}_{m(T+\tau)}(T)$$

onde os estimadores $\hat{a}_1(T), \hat{a}_2(T), \hat{\rho}_1(T), \hat{\rho}_2(T), \ldots \hat{\rho}_S(T)$ são calculados por amortecimento exponencial da seguinte forma:

$$\hat{a}_1(T) = \alpha[Z_T - \hat{\rho}_{m(T)}(T-1)] + (1-\alpha).[\hat{a}_1(T-1) + \hat{a}_2(T-1)]$$

$$\hat{a}_2(T) = \beta[\hat{a}_1(T) - \hat{a}_1(T-1)] + (1-\beta)\hat{a}_2(T-1)$$

$$\hat{\rho}_{m(T)}(T) = \gamma[Z_T - \hat{a}_1(T)] + (1-\gamma).\hat{\rho}_{m(T)}(T-1)$$

$$\hat{\rho}_j(T) = \hat{\rho}_j(T-1) \;;\; j = 1, 2, \ldots S \;;\; j \neq m(T)$$

Logo a seguir os estimadores dos fatores sazonais são normalizados de forma que $\sum_{j=1}^{S} \hat{\rho}_j(T) = 0$. As fórmulas para as estimações do nível a_1 e da inclinação a_2 são as mesmas que as do método de Holt-2-parâmetros com uma pequena correção em Z_T por causa do efeito sazonal (Z_T subtraído por seu fator sazonal correspondente é o valor da série dessazonalizado no instante T). A fórmula para estimação de cada fator sazonal representa a combinação linear da observação corrente sem a tendência (Z_T subtraído pela estimativa corrente do nível) e da estimativa do respectivo fator sazonal feita a S instantes anteriores.

Uma maneira alternativa de se modelar a sazonalidade de uma série é através do uso de uma combinação de funções trigonométricas (senos e cossenos). Neste caso utiliza-se o método de amortecimento direto (*direct smoothing*) para estimação dos parâmetros envolvidos. Esse método é aplicável a uma certa classe de modelos definidos como combinações lineares de funções do tempo (como funções polinomiais e trigonométricas). Maiores detalhes sobre esse método podem ser vistos em [36] e [51].

Não só as constantes de amortecimento devem ser estimadas a partir dos dados (séries não-sazonais e sazonais) mas também $\mathrm{Var}[\varepsilon_t]$, a variância de ε_t, se quisermos obter um intervalo de confiança para a previsão. Além disso é necessário supor que o erro de previsão τ-passos-à-frente $e_\tau(t) = Z_t - \hat{Z}_{t-\tau}(\tau)$ é normalmente distribuído com média zero e variância $\mathrm{Var}[e_\tau(t)]$ dada por

$$\mathrm{Var}[e_\tau(t)] = \mathrm{Var}[Z_t] + \mathrm{Var}[\hat{Z}_{t-\tau}(\tau)] = \mathrm{Var}[\varepsilon_t] + \mathrm{Var}[\hat{Z}_{t-\tau}(\tau)]$$

onde Var[Z_t] é a variância da série temporal (que é igual a Var[ε_t]) e Var[$\hat{Z}_{t-\tau}(\tau)$] é a variância da previsão pontual τ-passos-à-frente feita no instante t-τ. Maiores detalhes sobre a obtenção de um intervalo de confiança podem ser vistos em [36].

A escolha do modelo mais adequado (constante, linear, ...) pode ser feita através de algum critério de seleção de modelos como, por exemplo, o BIC (*Bayesian Information Criterion*) [48]. O modelo que minimiza o BIC provavelmente fornecerá as previsões mais acuradas. O BIC tenta balancear a recompensa pelo bom ajuste do modelo (aos dados usados na estimação das constantes do modelo) com a penalidade pela complexidade do modelo (quantidade de parâmetros).

2.3 - Box & Jenkins

Dado $Z_1, Z_2, ... Z_T$ uma série temporal de tamanho T. O método de Box & Jenkins [3] assume que a série pode ser descrita pelo modelo

$$Z_t = \mu + \psi_0\varepsilon_t + \psi_1\varepsilon_{t-1} + \psi_2\varepsilon_{t-2} + ...$$

onde $\varepsilon_t, \varepsilon_{t-1}, \varepsilon_{t-2}, ...$ são variáveis aleatórias independentes com média zero e variância constante, e $\mu, \psi_0, \psi_1, \psi_2, ...$ são constantes. Supondo que a distribuição de cada ε_i é normal, a seqüência de variáveis aleatórias $\varepsilon_t, \varepsilon_{t-1}, \varepsilon_{t-2}, ...$ é chamada de um processo ruído branco. Empregando-se o operador de retardo B definido por $B\varepsilon_t = \varepsilon_{t-1}$, e de modo geral $B^j\varepsilon_t = \varepsilon_{t-j}$, podemos reescrever o modelo como

$$Z_t = \mu + (\psi_0B^0 + \psi_1B^1 + \psi_2B^2 + ...)\varepsilon_t$$

ou então

$$Z_t = \mu + \Psi(B)\varepsilon_t, \text{ onde } \Psi(B) = \psi_0B^0 + \psi_1B^1 + \psi_2B^2 + ...$$

e normalmente consideramos $\psi_0 = 1$. A equação

$$Z_t = \mu + \sum_{j=0}^{\infty} \psi_j \varepsilon_{t-j}$$

é usualmente chamada de filtro linear. Dessa forma defini-se um modelo de série temporal como uma função que transforma um processo ruído branco em uma série temporal.

Infelizmente, o modelo descrito pelo filtro linear possui um número infinito de parâmetros. Em outras palavras, $\Psi(B)$ possui um número infinito de parâmetros. Para contornar isso, Box & Jenkins reescreveram o modelo da série fazendo

$$\Psi(B) = \Theta(B)/\Phi(B),$$

onde

$$\Theta(B) = 1 - \theta_1 B - ... - \theta_q B^q$$

$$\Phi(B) = 1 - \phi_1 B - ... - \phi_p B^p.$$

Dessa forma temos p+q parâmetros (θ_1, ..., θ_q, ϕ_1, ..., ϕ_p). Entretanto, essa modificação permite apenas que sejam modeladas séries estacionárias (ou seja, séries que flutuam aleatoriamente ao redor de uma média constante). Isso acontece pois foi definido que a série é o resultado da passagem de um ruído branco por um filtro linear. Como o ruído branco é um processo estacionário então sua passagem por um filtro linear produz um processo que também é estacionário [3]. Para o caso de séries não estacionárias (onde não existe uma média constante ao redor da qual as séries possam flutuar aleatoriamente) aplica-se o operador diferença (de ordem d) $\nabla^d = (1 - B)^d$ à variável Z_t. Um processo não estacionário homogêneo é um processo não estacionário que após diferenças sucessivas produz um processo estacionário [53]. Por exemplo, para d = 1 temos

$$\nabla Z_t = (1 - B)Z_t = Z_t - Z_{t-1},$$

para d = 2

$$\nabla^2 Z_t = (1 - B)^2 Z_t = (1 - B)(1 - B)Z_t = (1 - B)(Z_t - Z_{t-1}) = Z_t - 2.Z_{t-1} + Z_{t-2}.$$

Assim, os modelos Box-Jenkins, também conhecidos como modelos ARIMA (*Autoregressive Integrated Moving Average*), são descritos pela equação

$$\Phi(B).\nabla^d Z_t = \Theta(B).\varepsilon_t,$$

que representa a estrutura ARIMA (p, d, q) de um modelo.

Para o caso de séries sazonais com período S, Box & Jenkins definiram o modelo ARIMA (p, d, q) × $(P, D, Q)_S$ multiplicativo (ou SARIMA) que é expresso pela equação:

$$\Phi(B).\Lambda(B^S).\nabla_S^D \nabla^d Z_t = \Theta(B).\Gamma(B^S).\varepsilon_t$$

onde

$$\Lambda(B^S) = 1 - \gamma_1 B^S - \gamma_2 B^{2S} - ... - \gamma_P B^{PS}$$

$$\Gamma(B^S) = 1 - \lambda_1 B^S - \lambda_2 B^{2S} - ... - \lambda_Q B^{QS}$$

∇_S^D é o operador diferença sazonal definido por $\nabla_S^D = (1 - B^S)^D$.

Por exemplo, para D = 1 temos

$$\nabla_S Z_t = (1 - B^S)Z_t = Z_t - Z_{t-S},$$

para D = 2

$\nabla_S^2 Z_t = (1 - B^S)^2 Z_t = (1 - B^S)(1 - B^S)Z_t = (1 - B^S)(Z_t - Z_{t-S}) = Z_t - 2.Z_{t-S} + Z_{t-2S}$.

A modelagem de Box & Jenkins para séries temporais consiste de quatro etapas: identificação, estimação, testes de aderência e previsão. Na fase de identificação, o objetivo é determinar a estrutura do modelo ARIMA (ou SARIMA) a ser empregado, ou seja, descobrir os valores de d, p e q (adicionalmente D, P e Q para SARIMA). Isto é feito pela análise dos estimadores das funções de autocorrelação (FAC) e autocorrelação parcial (FACP). A FAC é definida por

$$\mathrm{Cor}(Z_t, Z_{t+k}) = \frac{E[(Z_t - E(Z_t))(Z_{t+k} - E(Z_{t+k}))]}{\sqrt{\mathrm{Var}(Z_t)\mathrm{Var}(Z_{t+k})}}$$

onde E(.) e Var(.) são o valor esperado e a variância, respectivamente, enquanto que a FACP é definida por

$\mathrm{Cor}(Z_t, Z_{t+k} \mid Z_{t+1}, ..., Z_{t+k-1})$.

Uma descrição detalhada sobre como é realizada a fase de identificação pode ser encontrada em [3].

Na fase de estimação, os parâmetros $(\theta_1, ..., \theta_q, \phi_1, ..., \phi_p)$, $(\lambda_1, ..., \lambda_Q, \gamma_1, ..., \gamma_P)$ são estimados minimizando a soma dos quadrados dos erros de previsão 1-passo-à-frente (também conhecidos como resíduos). Maiores detalhes são vistos em [3].

Os testes de aderência têm como propósito verificar se a seqüência de resíduos estimados pode ser considerada um processo ruído branco. Se isso for comprovado, então o modelo atual (estrutura e parâmetros) é considerado adequado. Caso contrário, volta-se à fase de identificação a fim de escolher um modelo alternativo. Esse ciclo (identificação, estimação e testes de aderência) é repetido até que se obtenha um modelo adequado. Maiores detalhes sobre os testes de aderência podem ser vistos em [3].

A fase final corresponde à previsão dos valores da série empregando o modelo obtido. A partir da equação deste modelo

$\Phi(B).\Lambda(B^S).\nabla_S^D \nabla^d Z_{t+\tau} = \Theta(B).\Gamma(B^S).\varepsilon_{t+\tau}$

obtém-se a equação de previsão por

$$\hat{Z}_t(\tau) = E\{Z_{t+\tau} \mid Z_1, Z_2, ... Z_t\}.$$

A aplicação do valor esperado condicional à equação do modelo requer o cálculo de vários componentes do tipo $E\{Z_{t+j} \mid Z_1, ... Z_t\}$ e do tipo $E\{\varepsilon_{t+j} \mid Z_1, ... Z_t\}$. Esses componentes são obtidos da seguinte forma:

$$E\{Z_{t+j} \mid Z_1, \ldots Z_t\} = \begin{cases} Z_{t+j} \text{ para } j \le 0 \\ \hat{Z}_t(j) \text{ para } j \ge 1 \end{cases}$$

$$E\{\varepsilon_{t+j} \mid Z_1, \ldots Z_t\} = \begin{cases} Z_{t+j} - \hat{Z}_{t+j-1}(1) \text{ para } j \le 0 \\ 0 \text{ para } j \ge 1 \end{cases}$$

O valor esperado condicional de um valor da série Z_{t+j} já observado é igual a esse mesmo valor Z_{t+j}. Se Z_{t+j} ainda não foi observado então seu valor esperado condicional será igual a sua previsão $\hat{Z}_t(j)$. O valor esperado condicional de ε_{t+j} é igual ao erro de previsão 1-passo-à-frente $e_1(t+j) = (Z_{t+j} - \hat{Z}_{t+j-1}(1))$ para um valor Z_{t+j} já observado. Se Z_{t+j} ainda não foi observado então o valor esperado condicional de ε_{t+j} será igual zero.

Diferentes horizontes de previsão (o valor de τ) podem produzir diferentes equações de previsão. Por exemplo, considere um modelo não-sazonal ARIMA(p, d, q) descrito pela equação

$\Phi(B).\nabla^d Z_t = \Theta(B).\varepsilon_t$

com p parâmetros em $\Phi(B)$ e q parâmetros em $\Theta(B)$. No caso específico de um modelo ARIMA(1, 1, 1) então temos a equação

$(1 - \phi_1 B)(1 - B)Z_t = (1 - \theta_1 B).\varepsilon_t$

que pode ser reescrita como

$(1 - \phi_1 B)(Z_t - Z_{t-1}) = (1 - \theta_1 B).\varepsilon_t \Rightarrow$

$Z_t - Z_{t-1} - \phi_1(Z_{t-1} - Z_{t-2}) = (1 - \theta_1 B).\varepsilon_t \Rightarrow$

$Z_t = Z_{t-1} + \phi_1(Z_{t-1} - Z_{t-2}) + (1 - \theta_1 B).\varepsilon_t \Rightarrow$

$Z_t = (1 + \phi_1)Z_{t-1} - \phi_1 Z_{t-2} + (1 - \theta_1 B).\varepsilon_t \Rightarrow$

$Z_t = (1 + \phi_1)Z_{t-1} - \phi_1 Z_{t-2} + \varepsilon_t - \theta_1 \varepsilon_{t-1}$

ou

$Z_{t+\tau} = (1 + \phi_1)Z_{t+\tau-1} - \phi_1 Z_{t+\tau-2} + \varepsilon_{t+\tau} - \theta_1 \varepsilon_{t+\tau-1}$

e a equação de previsão 1-passo-à-frente é dada por

$$\hat{Z}_t(1) = (1 + \hat{\phi}_1)Z_t - \hat{\phi}_1 Z_{t-1} - \hat{\theta}_1(Z_t - \hat{Z}_{t-1}(1)),$$

onde $\hat{\phi}_1$ e $\hat{\theta}_1$ são os estimadores dos parâmetros do modelo. O significado dessa equação é que nossa previsão de Z_{t+1} depende do conhecimento de Z_t, Z_{t-1} e do erro de previsão 1-passo-à-frente $e_1(t) = (Z_t - \hat{Z}_{t-1}(1))$ para o valor Z_t.

A equação de previsão 2-passos-à-frente é dada por

$$\hat{Z}_t(2) = (1+\hat{\phi}_1)\hat{Z}_t(1) - \hat{\phi}_1 Z_t,$$

que significa que nossa previsão de Z_{t+2} depende do conhecimento de Z_t e da previsão 1-passo-à-frente para o valor Z_{t+1}.

A equação de previsão 3-passos-à-frente é dada por

$$\hat{Z}_t(3) = (1+\hat{\phi}_1)\hat{Z}_t(2) - \hat{\phi}_1 \hat{Z}_t(1),$$

que significa que nossa previsão de Z_{t+3} depende do conhecimento da previsão 2-passos-à-frente para o valor Z_{t+2} e da previsão 1-passo-à-frente para o valor Z_{t+1}.

A equação de previsão τ-passos-à-frente para $\tau \geq 4$ é dada por

$$\hat{Z}_t(\tau) = (1+\hat{\phi}_1)\hat{Z}_t(\tau-1) - \hat{\phi}_1 \hat{Z}_t(\tau-2),$$

que significa que nossa previsão de $Z_{t+\tau}$ depende do conhecimento da previsão $(\tau-1)$-passos-à-frente para o valor $Z_{t+\tau-1}$ e da previsão $(\tau-2)$-passos-à-frente para o valor $Z_{t+\tau-2}$.

Além dos parâmetros (θ_1, ..., θ_q, ϕ_1, ..., ϕ_p), (λ_1, ..., λ_Q, γ_1, ..., γ_P), a variância de ε_t, Var[ε_t], também deve ser estimada a partir dos dados se quisermos obter um intervalo de confiança para a previsão. Da mesma forma que no método de amortecimento exponencial, supomos que o erro de previsão τ-passos-à-frente $e_\tau(t) = Z_t - \hat{Z}_{t-\tau}(\tau)$ é normalmente distribuído com média zero e variância dada por

$$\mathrm{Var}[e_\tau(t)] = \mathrm{Var}[Z_t] + \mathrm{Var}[\hat{Z}_{t-\tau}(\tau)] = \mathrm{Var}[\varepsilon_t] + \mathrm{Var}[\hat{Z}_{t-\tau}(\tau)].$$

Maiores detalhes sobre como um intervalo de confiança pode ser obtido são mostrados em [3].

O ciclo constituído pelas fases de identificação, estimação e testes de aderência tem por objetivo a seleção de um modelo adequado para uma determinada série temporal. Entretanto, é possível que na fase de identificação geremos mais de uma estrutura candidata. Se ambos os modelos candidatos forem aprovados pelos testes de aderência, podemos escolher o modelo mais adequado da mesma forma que no método de amortecimento exponencial, ou seja, através do critério de seleção de modelos conhecido como BIC (que leva em conta tanto o ajuste do modelo aos dados quanto a quantidade de parâmetros do modelo) [3].

2.4 - Modelos de Espaço de Estados

Na Seção 2.2 foi visto que os métodos de amortecimento exponencial pressupõem que uma série temporal pode ser modelada por várias componentes que não são observadas diretamente mas que possuem uma interpretação direta (como a tendência e a sazonalidade). Além disso, os estimadores dessas componentes são atualizados a cada instante em que um novo valor da série temporal (uma nova observação) se torna disponível. Um exemplo simples é o modelo constante

$Z_t = a_1 + \varepsilon_t$

onde a_1 é a componente que representa o nível médio da série. Essa componente pode ser estimada por médias móveis

$$\hat{a}_1(t) = M_t = M_{t-1} + (Z_t - Z_{t-N})/N$$

ou por amortecimento exponencial

$$\hat{a}_1(t) = S_t = \alpha.Z_t + (1-\alpha)S_{t-1}$$

onde Z_t é a observação disponível mais recente. Um outro exemplo é o modelo linear

$Z_t = a_1 + a_2 t + \varepsilon_t$

cujas componentes podem ser estimadas pelo método de Brown ou Holt-2-parâmetros. Uma maneira alternativa de se representar um modelo constituído por componentes não observadas cujos estimadores são atualizados a cada instante é através do uso de modelos de espaço de estados [1]. Um modelo colocado na forma de espaço de estados permite que os estimadores de suas componentes sejam atualizados através do procedimento conhecido como Filtro de Kalman.

O seguinte modelo (útil para modelar uma série de carga elétrica mensal)

$Z_t = \mu_t + \gamma_t + \varepsilon_t,$

pode ser colocado na forma de espaço de estados [52], [17], [74], onde μ_t representa a componente de tendência (de crescimento, ou decrescimento, ou sem tendência, isto é, nível constante), γ_t representa componente de sazonalidade e ε_t é uma componente irregular. Um modelo de espaço de estados é descrito pelas seguintes equações

- equação das observações: $Z_t = \mathbf{F}_t'\boldsymbol{\theta}_t + \nu_t$ $\quad\quad$ $\nu_t \sim N[0, V_t]$
- equação do estado: $\boldsymbol{\theta}_t = \mathbf{G}_t\boldsymbol{\theta}_{t-1} + \boldsymbol{\omega}_t$ $\quad\quad$ $\boldsymbol{\omega}_t \sim N[\mathbf{0}, \mathbf{W}_t]$

onde $\mathbf{F}_t$ é um vetor e $\mathbf{G}_t$ é uma matriz, ambos conhecidos para todo t, ν_t e $\boldsymbol{\omega}_t$ são ruídos brancos Gaussianos independentes com médias zero, variância V_t e matriz de

covariância $\mathbf{W}_t$, respectivamente, e $\boldsymbol{\theta}_t$ é o vetor de estado que contém cada uma das componentes não observáveis. Por exemplo, um modelo com tendência linear estocástica expresso pelas equações

- equação das observações: $Z_t = \mu_t + \varepsilon_t \qquad \varepsilon_t \sim N[0, \sigma^2_\varepsilon]$
- equações do estado: $\mu_t = \mu_{t-1} + \beta_{t-1} + \eta_t \qquad \eta_t \sim N[0, \sigma^2_\eta]$

$$\beta_t = \beta_{t-1} + \zeta_t \qquad \zeta_t \sim N[0, \sigma^2_\zeta]$$

onde μ_t é a tendência estocástica da série e β_t é a inclinação estocástica da série, pode ser colocado na forma de espaço de estados a seguir

- equação das observações: $Z_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \mu_t \\ \beta_t \end{bmatrix} + \varepsilon_t \qquad \varepsilon_t \sim N[0, \sigma^2_\varepsilon]$
- equação do estado: $\begin{bmatrix} \mu_t \\ \beta_t \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_{t-1} \\ \beta_{t-1} \end{bmatrix} + \begin{bmatrix} \eta_t \\ \zeta_t \end{bmatrix} \qquad \begin{bmatrix} \eta_t \\ \zeta_t \end{bmatrix} \sim N\left[\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma^2_\eta & 0 \\ 0 & \sigma^2_\zeta \end{bmatrix} \right]$

No caso específico em que as variâncias σ^2_η e σ^2_ζ possuem o valor zero, temos um modelo com tendência linear determinística. Quando essas variâncias são nulas as equações do estado passam a ser

$$\mu_t = \mu_{t-1} + \beta_{t-1}$$

$$\beta_t = \beta_{t-1}$$

que são agora duas equações determinísticas. Considerando μ_0 e β_0 os valores iniciais dessas componentes temos

$$\mu_t = \mu_{t-1} + \beta_0$$

na qual substituímos μ_{t-1} por

$$\mu_{t-1} = \mu_{t-2} + \beta_0$$

obtendo

$$\mu_t = \mu_{t-2} + \beta_0 + \beta_0$$

que por substituições sucessivas se torna

$$\mu_t = \mu_0 + t.\beta_0$$

e colocando-a na equação das observações temos

$$Z_t = \mu_0 + t.\beta_0 + \varepsilon_t \qquad \varepsilon_t \sim N[0, \sigma^2_\varepsilon]$$

que é a equação do modelo linear empregada pelo método de Brown e Holt-2-parâmetros. A única diferença é que μ_0 e β_0 são constantes enquanto que nos métodos de amortecimento exponencial elas seriam atualizadas a cada instante.

Seja D_t toda a informação disponível até o instante t, e I_t a informação adicional disponível no instante t. Dessa forma temos $D_t = \{I_t, D_{t-1}\}$. Se a única informação disponível é a série temporal $Z_1, Z_2, \ldots Z_t$ então $D_t = \{Z_t, D_{t-1}\}$ e $D_0 = \{\}$. Supondo que a cada instante t-1 conheçamos a distribuição do vetor de estado $\boldsymbol{\theta}_{t-1}$ condicionada à informação D_{t-1}:

$(\boldsymbol{\theta}_{t-1} \mid D_{t-1}) \sim N[\mathbf{m}_{t-1}, \mathbf{C}_{t-1}]$

onde $\mathbf{m}_{t-1}$ e $\mathbf{C}_{t-1}$ são respectivamente a média e a variância de uma distribuição normal ($\mathbf{m}_0$ e $\mathbf{C}_0$ são os valores iniciais para o instante 0 quando $D_0 = \{\}$). Aplicando-se a equação do estado a essa distribuição obtém-se a distribuição do vetor de estado $\boldsymbol{\theta}_t$ condicionada à informação D_{t-1}:

$(\boldsymbol{\theta}_t \mid D_{t-1}) \sim N[\mathbf{a}_t, \mathbf{R}_t]$

onde

$\mathbf{a}_t = \mathbf{G}_t\mathbf{m}_{t-1}$

$\mathbf{R}_t = \mathbf{G}_t\mathbf{C}_{t-1}\mathbf{G}_t' + \mathbf{W}_t.$

Aplicando-se a equação das observações a essa distribuição obtém-se a distribuição de Z_t condicionada à informação D_{t-1}, ou seja, a previsão 1-passo-à-frente de Z_t feita no instante t-1:

$(Z_t \mid D_{t-1}) \sim N[f_t, Q_t]$

onde

$f_t = \mathbf{F}_t'\mathbf{a}_t$

$Q_t = \mathbf{F}_t'\mathbf{R}_t\mathbf{F}_t + V_t.$

Para se efetuar a previsão 1-passo-à-frente no instante seguinte é realizada uma atualização seqüencial do vetor de estado, ou seja, é feita a transição

$(\boldsymbol{\theta}_{t-1} \mid D_{t-1}) \rightarrow (\boldsymbol{\theta}_t \mid D_t).$

Essa atualização é conhecida como Filtro de Kalman. A transição inicial

$(\boldsymbol{\theta}_{t-1} \mid D_{t-1}) \rightarrow (\boldsymbol{\theta}_t \mid D_{t-1})$

já foi definida pela aplicação da equação do estado, faltando apenas definir a transição final

$(\boldsymbol{\theta}_t \mid D_{t-1}) \rightarrow (\boldsymbol{\theta}_t \mid D_t).$

Como já conhecemos

$(\boldsymbol{\theta}_t \mid D_{t-1}) \sim N[\mathbf{a}_t, \mathbf{R}_t]$

$(Z_t \mid D_{t-1}) \sim N[f_t, Q_t]$

a distribuição do vetor de estado $\boldsymbol{\theta}_t$ condicionada à informação D_t é dada por:

$(\boldsymbol{\theta}_t \mid D_t) \sim N[\mathbf{m}_t, \mathbf{C}_t]$

onde

$\mathbf{m}_t = \mathbf{a}_t + \mathbf{A}_t e_t$

$\mathbf{C}_t = \mathbf{R}_t - \mathbf{A}_t \mathbf{A}_t' Q_t$

$e_t = Z_t - f_t$

$\mathbf{A}_t = \mathbf{R}_t \mathbf{F}_t / Q_t$.

Percebe-se que a média atualizada $\mathbf{m}_t$ do vetor de estado é sua média anterior $\mathbf{a}_t$ mais um termo proporcional ao erro de previsão $e_t = (Z_t - f_t)$ cometido no instante t. Além disso, existe uma diminuição da incerteza do vetor de estado visto que sua variância atualizada $\mathbf{C}_t$ é sua variância anterior $\mathbf{R}_t$ menos uma matriz de elementos não negativos. Detalhes de como esses resultados foram obtidos podem ser vistos em [17] e [74]. Como exemplo, considere um modelo de nível local expresso pelas equações

- equação das observações: $Z_t = \mu_t + \varepsilon_t$ $\quad \varepsilon_t \sim N[0, \sigma^2_\varepsilon]$
- equações do estado: $\mu_t = \mu_{t-1} + \eta_t$ $\quad \eta_t \sim N[0, \sigma^2_\eta]$

que já está na forma de espaço de estados com

$\mathbf{F}_t = 1$, $\boldsymbol{\theta}_t = \mu_t$, $\nu_t = \varepsilon_t$, $V_t = \sigma^2_\varepsilon$,

$\mathbf{G}_t = 1$, $\boldsymbol{\omega}_t = \eta_t$, $\mathbf{W}_t = \sigma^2_\eta$.

Assim, conhecendo-se

$(\mu_{t-1} \mid D_{t-1}) \sim N[m_{t-1}, C_{t-1}]$

aplica-se a equação do estado a essa distribuição obtendo-se:

$(\mu_t \mid D_{t-1}) \sim N[a_t, R_t]$

onde

$a_t = m_{t-1}$

$R_t = C_{t-1} + \sigma^2_\eta$.

Aplicando-se a equação das observações a essa distribuição obtém-se:

$(Z_t \mid D_{t-1}) \sim N[f_t, Q_t]$

onde

$f_t = a_t = m_{t-1}$

$Q_t = R_t + \sigma^2_\varepsilon = C_{t-1} + \sigma^2_\eta + \sigma^2_\varepsilon$.

Como já conhecemos

$(\mu_t \mid D_{t-1}) \sim N[a_t, R_t] = N[m_{t-1}, C_{t-1} + \sigma^2_\eta]$

$(Z_t \mid D_{t-1}) \sim N[f_t, Q_t] = N[m_{t-1}, C_{t-1} + \sigma^2_\eta + \sigma^2_\varepsilon]$

a distribuição do vetor de estado μ_t condicionada à informação D_t é dada por:

$(\mu_t \mid D_t) \sim N[m_t, C_t]$

onde

$e_t = Z_t - f_t = Z_t - m_{t-1}$

$A_t = R_t / Q_t = (C_{t-1} + \sigma^2_\eta) / (C_{t-1} + \sigma^2_\eta + \sigma^2_\varepsilon)$

$m_t = a_t + A_t e_t = m_{t-1} + A_t(Z_t - m_{t-1}) = A_t.Z_t + (1 - A_t).m_{t-1}$

$C_t = R_t - A_t A_t Q_t = R_t - (R_t / Q_t)R_t = R_t (1 - R_t / Q_t)$.

Além disso, passamos a conhecer (pela equação do estado) a distribuição do vetor de estado μ_{t+1} condicionada à informação D_t dada por

$(\mu_{t+1} \mid D_t) \sim N[a_{t+1}, R_{t+1}] = N[m_t, C_t + \sigma^2_\eta]$,

e também (pela equação das observações) a distribuição de Z_{t+1} condicionada à informação D_t dada por

$(Z_{t+1} \mid D_t) \sim N[f_{t+1}, Q_{t+1}] = N[m_t, C_t + \sigma^2_\eta + \sigma^2_\varepsilon]$.

Dessa forma, temos que a previsão (pontual) 1-passo-à-frente de Z_{t+1} feita no instante t é dada por

$$\hat{Z}_t(1) = m_t = A_t \cdot Z_t + (1 - A_t) \cdot m_{t-1},$$

ou seja, é a mesma equação de previsão para o modelo constante no amortecimento exponencial exceto pelo fato de que A_t não é constante.

A operacionalização do modelo na forma de espaço de estados depende do conhecimento da quádupla

$M_t = \{\mathbf{F}_t, \mathbf{G}_t, V_t, \mathbf{W}_t\}$

para todo instante t. Entretanto, V_t e $\mathbf{W}_t$ não são conhecidos. Dependendo da abordagem utilizada para a obtenção dessas quantidades desconhecidas, podemos considerar duas possíveis implementações:

- Modelos Estruturais de Harvey (abordagem clássica [17]): considera constantes todas as quantidades desconhecidas, as quais são estimadas através de métodos convencionais de maximização da função de verossimilhança.
- Modelos Lineares Dinâmicos de Harrison & Stevens (abordagem Bayesiana [74]): as quantidades desconhecidas são definidas subjetivamente pelo usuário ou seqüencialmente estimadas via inferência Bayesiana.

A abordagem clássica de Harvey permite que existam constantes desconhecidas na matriz $\mathbf{G}_t$, tornando essa abordagem mais abrangente quanto às componentes que podem ser modeladas. Todavia, isso traz como conseqüência a necessidade de série temporais de tamanho elevado a fim de que se possa garantir a consistência dos

estimadores das quantidades desconhecidas. Por exemplo, pode-se modelar a componente conhecida como ciclo, muito útil em séries econômicas e metereológicas (manchas solares, séries de precipitações, etc.) [52]. Um ciclo corresponde a flutuações no nível de uma variável (econômica, metereológica, etc.) que ocorrem de forma recorrente, com periodicidade aproximadamente regular e sempre superior a um ano. A abordagem Bayesiana de Harrison & Stevens não permite a modelagem de ciclos (a não ser que as constantes desconhecidas que integram o ciclo sejam pré-especificadas) mas, por outro lado, possibilita a monitoração da série temporal através do fator de Bayes [22], [74]. Pelo uso do fator de Bayes torna-se possível identificar de maneira automática quando o modelo atualmente empregado (ou seja, representando a série temporal) deve ser substituído por outro. Além de definir quantidades desconhecidas de forma subjetiva, a abordagem de Harrison & Stevens faz uso de "fatores de desconto" para permitir a atualização dessas quantidades (V_t e $\mathbf{W}_t$) a cada instante. A idéia por trás desses fatores de desconto (que são quantidades definidas no intervalo [0,1]) é que o conteúdo de informação de uma observação da série decai com a sua idade. Quanto menor for o fator de desconto menos importância é dada às informações antigas, ou seja, maior é a taxa de perda de informação. Dessa forma é razoável pensar que o fator de desconto para informação referente à componente sazonal seja maior que o fator de desconto para informação referente à componente de tendência. Maiores detalhes sobre a utilização de fatores de desconto podem ser vistos em [74].

Comparando-se os modelos de espaço de estados com os métodos de amortecimento exponencial e com os modelos de Box & Jenkins, percebe-se que os métodos de amortecimento exponencial são os que necessitam de menos conhecimento especializado por parte dos usuários. A seguir estão os modelos estruturais de Harvey, logo depois os modelos lineares dinâmicos de Harrison & Stevens e por último temos os modelos de Box & Jenkins. Quanto aos modelos disponíveis, os métodos de amortecimento exponencial possuem uma classe restrita de modelos com componentes representadas por funções do tempo que são ajustadas às observações. Nos modelos de espaço de estados existe uma classe bem mais ampla de modelos cujas componentes são representadas por processos estocásticos. A metodologia de Box & Jenkins também oferece uma classe ampla de modelos que, no entanto, sofrem do problema da carência de uma interpretação real de suas estruturas (ao contrário dos modelos representados por componentes não-observáveis). Além disso, os modelos de Box & Jenkins possuem a restrição de que seus parâmetros sejam constantes, sem a possibilidade de uma

atualização dinâmica dos mesmos no momento em que uma nova observação se torna disponível.

Como comentário final deve-se mencionar que existem maneiras de se introduzir não-linearidades nos modelos de espaço de estados. Uma possibilidade é o uso de modelos condicionalmente Gaussianos [17], onde $\mathbf{F}_t$, $\mathbf{G}_t$, V_t e $\mathbf{W}_t$ podem depender de observações passadas. Outra abordagem é definir Z_t como uma função não-linear de $\boldsymbol{\theta}_t$ na equação das observações, $\boldsymbol{\theta}_t$ como uma função não-linear de $\boldsymbol{\theta}_{t-1}$ na equação do estado, e então empregar um filtro aproximado como, por exemplo, o filtro de Kalman estendido [17], [74].

2.5 - Redes Neurais Artificiais

Uma rede neural artificial [18], [77] é constituída de elementos processadores muito simples denominados unidades ou neurônios os quais possuem conexões entre si denominadas pesos. A esses pesos estão associados valores numéricos. As unidades possuem ativações (também valores numéricos) que resultam de algum tipo de combinação das ativações das unidades as quais estão conectadas. Essa combinação leva em conta não só as ativações das unidades interconectadas como também os valores dos pesos que conectam essas unidades. O "comportamento" da rede neural pode ser descrito como os resultados finais das ativações de suas unidades dados valores iniciais de ativações e valores dos pesos existentes. Dependendo do tipo de rede neural empregado, os pesos podem já ter valores pré-definidos ou podem ser "aprendidos" através de um algoritmo capaz de fazer com que a rede tente se comportar de uma maneira já pré-definida. Essas redes com aprendizado são úteis para tarefas de previsão. Um exemplo desse tipo de rede neural é a rede *feedforward*. Ela possui várias camadas de unidades: uma camada de entrada, uma camada de saída e zero ou mais camadas intermediárias entre a camada de entrada e a de saída. Na Figura 6 é mostrado um exemplo de rede *feedforward* com 3 camadas. Os retângulos representam as camadas, os círculos são as unidades e as linhas entre as camadas são as conexões entre unidades. Também é indicado o sentido dessas conexões.

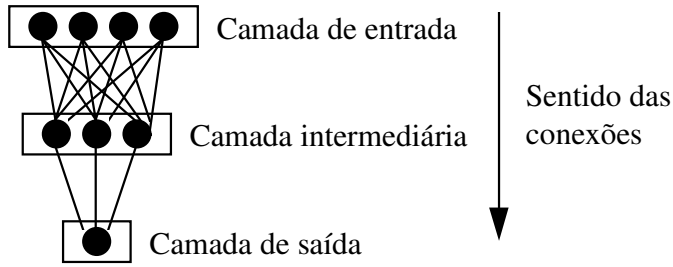


Figura 6: Rede *feedforward* com 3 camadas.

Com exceção das unidades da camada de entrada, que possuem ativações definidas *a priori*, qualquer outra unidade tem sua ativação calculada da seguinte forma: é feita a soma ponderada das ativações das unidades ligadas a ela pelos respectivos pesos (deve-se frisar que o sentido dessas conexões vai das outras unidades para a unidade cuja ativação deve ser calculada); a seguir o valor dessa soma ponderada passa por uma função (como exemplos, f(x) = x, ou f(x) = 1 / (1 + e^{-x})); o resultado dessa função (chamada de função de ativação) é a ativação da unidade. Assim, uma vez que as unidades da camada de entrada tenham alguma ativação, valores são propagados adiante de camada a camada, produzindo as ativações de suas unidades.

Se quiséssemos usar uma rede *feedforward* para a previsão de uma série $Z_1, Z_2, ... Z_T$ poderíamos fazer o seguinte [26], [55], [56], [57]: as unidades da camada de entrada representariam os valores $Z_{t-m+1}, ...,Z_{t-1}, Z_t$ (logo, temos m unidades de entrada) e na camada de saída teríamos uma única unidade que representaria o valor Z_{t+1} para um instante t qualquer. Isso corresponde a prever um valor futuro da série temporal após um determinado período de valores da série: deve-se prever Z_{t+1} após um período de valores $Z_{t-m+1}, ...,Z_{t-1}, Z_t$. A série temporal inteira poderia ser transformada em uma seqüência de padrões na forma

"entrada: $[Z_{t-m+1}, ...,Z_{t-1}, Z_t]$ - saída: $[Z_{t+1}]$"

que seriam empregados para o aprendizado da rede. Dessa forma teríamos o seguinte conjunto de treinamento:

"entrada: $[Z_1, ...,Z_{m-1}, Z_m]$ - saída: $[Z_{m+1}]$",

"entrada: $[Z_2, ...,Z_m, Z_{m+1}]$ - saída: $[Z_{m+2}]$",

...

"entrada: $[Z_{T-m-1}, ...,Z_{T-3}, Z_{T-2}]$ - saída: $[Z_{T-1}]$",

"entrada: $[Z_{T-m}, ...,Z_{T-2}, Z_{T-1}]$ - saída: $[Z_T]$".

Para avaliar o desempenho da rede empregaríamos um conjunto de teste construído a partir de valores da série que não foram usados no treinamento:

"entrada: $[Z_{T-m+1}, ..., Z_{T-1}, Z_T]$ - saída: $[Z_{T+1}]$",

"entrada: $[Z_{T-m+2}, ..., Z_T, Z_{T+1}]$ - saída: $[Z_{T+2}]$", ...

onde Z_{T+1}, Z_{T+2}, ... são valores da série não usados no treinamento.

Backpropagation [45] é um algoritmo de aprendizado usado para o treinamento de redes *feedforward*. Em [76] redes neurais são treinadas por esse algoritmo para a previsão de séries de carga elétrica mensal. Em [71] são usadas as mesmas séries de carga elétrica mensal mas emprega-se um algoritmo de aprendizado Bayesiano para as redes neurais.

Nas redes *feedforward* as ativações das unidades produzidas por um padrão de entrada passado não possuem qualquer influência no cálculo de novas ativações a partir de um padrão de entrada posterior. As unidades e conexões dessas redes formam grafos direcionados acíclicos. Podemos tornar os grafos representados por essas redes cíclicos através do acréscimo de conexões recorrentes. Obtemos assim redes neurais recorrentes [5], [18] nas quais os cálculos de novas ativações dependem de ativações passadas. Em [59] redes neurais recorrentes são usadas para a previsão de séries de carga elétrica horária.

2.6 - Sistemas *Fuzzy*

Na seção anterior construímos um modelo de previsão usando uma seqüência de valores da série $[Z_{t-m+1}, ..., Z_{t-1}, Z_t]$ para poder prever o valor imediatamente posterior Z_{t+1}. Sistemas *fuzzy* também podem ser usados para esta tarefa de previsão [72], [28]. Primeiro, os espaços de entrada ($[Z_{t-m+1}, ..., Z_{t-1}, Z_t]$) e saída ($Z_{t+1}$) para os dados numéricos são divididos em regiões *fuzzy*. Dividir um espaço numérico em regiões *fuzzy* é similar a uma discretização deste espaço, ou seja, o espaço é dividido em um conjunto fixo de intervalos. Entretanto, na discretização não existem interseções entre os intervalos (regiões) enquanto que existem interseções entre regiões *fuzzy*. Então regras *fuzzy* são geradas a partir dos dados (padrões na forma "entrada: $[Z_{t-m+1}, ..., Z_{t-1}, Z_t]$ - saída: $[Z_{t+1}]$" obtidos a partir da série temporal). Os antecedentes e o conseqüente de uma regra *fuzzy* são da forma "valor está na região *fuzzy*". Predições são feitas usando essas regras, um procedimento de inferência e um procedimento de defuzzificação

(*defuzzifying procedure*). Este procedimento transforma a informação *fuzzy*, que foi inferida pelas regras, em valores numéricos.

Em [72] e [31], regras *fuzzy* foram geradas pela fuzzificação (*fuzzification*) de valores numéricos provenientes de exemplos. Um exemplo é dado por:

$a_1^{(1)}$ e $a_2^{(1)}$ e ... e $a_m^{(1)} \rightarrow v^{(1)}$

onde cada "$a_k^{(1)}$" representa o valor numérico do k-ésimo atributo de entrada "a" para o exemplo 1 e "$v^{(1)}$", o valor numérico de saída. Considerando x como sendo a_k ou v, r(x) $\in$ {r1, r2, ..., rn} sendo a região *fuzzy* cuja função de pertinência, $m_r(x)$, tem o máximo grau para x. Na Figura 7, $v^{(1)}$ e $v^{(2)}$ são valores provenientes de dois exemplos, e [v^-, v^+] é o intervalo do domínio de v. O formato de cada função de pertinência é triangular. Por exemplo: m_{r2}("centro de r2") = 1, m_{r2}("centro de r1") = m_{r2}("centro de r3") = 0, r($v^{(1)}$) = r4, r($v^{(2)}$) = r2, $m_r(v^{(1)}) = m_{r4}(v^{(1)}) = 0.8$, $m_r(v^{(2)}) = m_{r2}(v^{(2)}) = 0.6$. O número de regiões *fuzzy* e os formatos das funções de pertinência são definidos pelo usuário.

Dado o exemplo anterior, nós obtemos a seguinte regra *fuzzy*:

Regra k:

Se [a_1 está em r($a_1^{(1)}$)] e [a_2 está em r($a_2^{(1)}$)] e ... e [a_m está em r($a_m^{(1)}$)]

Então [v está em r($v^{(1)}$)]

O grau D(.) dessa regra é dado por:

$D(\text{Regra k}) = m_r(a_1^{(1)}) \times m_r(a_2^{(1)}) \times ... \times m_r(a_m^{(1)}) \times m_r(v^{(1)})$

Se dois exemplos diferentes geram regras com as mesmas pré-condições então apenas é mantida a regra com maior grau. Repetindo esse processo para cada exemplo, geramos uma quantidade de regras que é menor ou igual ao número de exemplos.

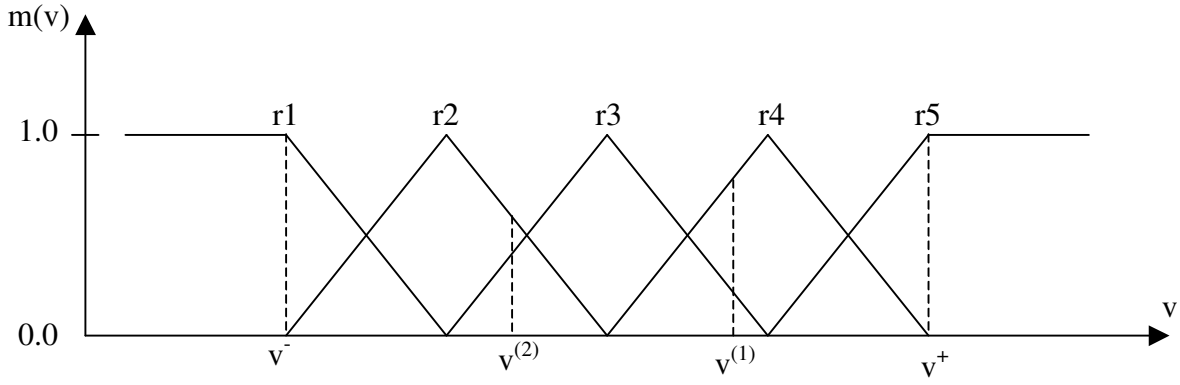


Figura 7**:** Regiões *fuzzy* e a função de pertinência.

Dado uma entrada (a_1, a_2, ..., a_m) pode-se obter um valor aproximado para a saída v através de um processo de inferência seguido por um processo denominado defuzzificação (*defuzzification*). Primeiro é executado o procedimento de inferência: computa-se o grau de saída de cada regra por

$$m_{o^k}^k = m_{i_1^k}(a_1) \times m_{i_2^k}(a_2) \times ... \times m_{i_m^k}(a_m)$$

onde o^k é a região de saída da Regra k (ou seja, é a região que aparece no conseqüente dessa regra), e i_j^k é a região de entrada da Regra k para a j-ésima componente (ou seja, é a região que aparece no j-ésimo antecedente dessa regra). Seja $\bar{v}^k$ o valor central da região o^k e u o número total de regras. A fórmula de defuzzificação mostrada a seguir [72] é usada para produzir a saída numérica

$$v = \frac{\sum_{k=1}^{u} m_{o^k}^k \cdot \bar{v}^k}{\sum_{k=1}^{u} m_{o^k}^k}.$$

Deve-se frisar que existem muitos outros métodos de defuzzificação [31] além desse.

Para previsão da série temporal cada exemplo é dado por

$[a_1^{(t)} = Z_t]$ e $[a_2^{(t)} = Z_{t-1}]$ e ... e $[a_m^{(t)} = Z_{t-m+1}] \rightarrow [v^{(t)} = Z_{t+1}]$

2.7 - Sistemas Inteligentes Híbridos

Sistemas inteligentes híbridos empregam duas ou mais técnicas de inteligência computacional a fim de se obter a solução para um determinado problema. Por exemplo, sistemas neuro-*fuzzy* [21], [38], [29] combinam características de redes neurais artificiais e de sistemas *fuzzy*. Um outro exemplo são modelos locais de redes neurais [42], [57], [58], [59], que fazem uso de dois tipos diferentes de redes neurais: uma rede de aprendizado não-supervisionado para clusterização (agrupamento dos dados segundo algum critério) e várias redes de aprendizado supervisionado (como a rede *feedforward*). A Seção 2.5 apresentou uma única rede de aprendizado supervisionado que foi treinada com todos os dados, obtendo-se um "modelo global" para previsão. Também é possível o aprendizado de "modelos locais" (Figura 8): uma rede neural de aprendizado não-supervisionado (como, por exemplo, *neural gas* [30]) divide os dados em grupos e cada um desses grupos é usado no treinamento de diferentes redes neurais de aprendizado supervisionado.

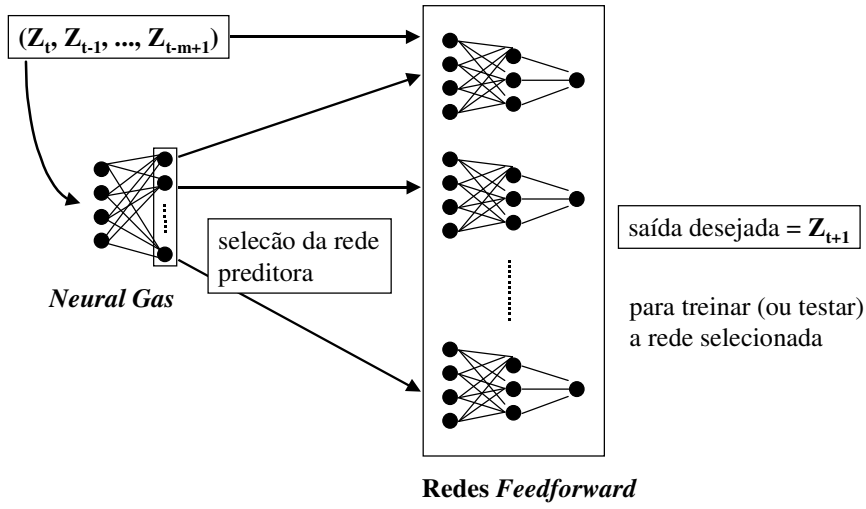


Figura 8: Modelos locais usando *neural gas* e redes *feedforward*.

Existem também sistemas inteligentes híbridos que empregam técnicas estatísticas além das técnicas de inteligência computacional. Em [27] e [50] são apresentados sistemas que utilizam redes neurais de aprendizado não-supervisionado para clusterização (neste caso, o *Self-Organizing Map* [25], também conhecido como rede de Kohonen), sistemas *fuzzy* e métodos estatísticos (médias móveis, amortecimento exponencial e modelo auto-regressivo) para previsão. Em [29], sistemas neuro-*fuzzy* são utilizados para determinar as estruturas de modelos de Box & Jenkins para séries não-sazonais.

3 - Redes Bayesianas e Redes Bayesianas Dinâmicas

Este capítulo descreve Redes Bayesianas e Redes Bayesianas Dinâmicas, duas técnicas de inteligência computacional necessárias na estimação não-paramétrica de uma densidade contínua.

3.1 - Outras Técnicas de Inteligência Computacional

Neste capítulo serão mostradas duas outras técnicas de inteligência computacional: Redes Bayesianas (RB's) e Redes Bayesianas Dinâmicas (RBD's) [47]. RB's são sistemas compostos de várias unidades interconectadas, generalizando redes neurais artificiais [23]. Nas RB's as unidades representam variáveis aleatórias e as conexões dependências condicionais. RBD's são RB's que representam um modelo probabilístico temporal que é adequado para dados que possuem uma dependência temporal entre si. Aqui é empregada a seguinte notação [47]:

- variáveis aleatórias são representadas por nomes que começam com letras maiúsculas (tais como Y, Z, Classe);
- valores específicos assumidos por variáveis aleatórias são representados por nomes que começam com letras minúsculas (tais como y, z, classe);
- conjuntos de variáveis são representados por nomes em negrito que começam com letras maiúsculas (tais como **Y**, **Z**, **Classe**);
- valores específicos assumidos por conjuntos de variáveis são representados por nomes em negrito que começam com letras minúsculas (tais como **y**, **z**, **classe**);
- a probabilidade de um valor possível de uma variável aleatória (ou um conjunto de variáveis) é denotada por P(.);
- a distribuição de probabilidade de uma variável aleatória (ou um conjunto de variáveis) é denotada por **P**(.).

3.2 - Redes Bayesianas

Uma Rede Bayesiana (RB) [47], [23], [39], [8] é um grafo direcionado acíclico cujos nós representam variáveis aleatórias (discretas ou contínuas) e arcos entre nós dependências condicionais. A cada nó X_i está associada uma distribuição de probabilidade

$\mathbf{P}(X_i \mid \text{Pais}(X_i))$,

onde $\text{Pais}(X_i)$ são todos os nós que possuem arcos que vão em direção ao nó X_i. A Figura 9 mostra um exemplo simples de uma RB que poderia ser usada para uma tarefa de classificação: a variável discreta Classe representaria as possíveis classes do problema enquanto que as variáveis (discretas ou contínuas) Atributo-k (k = 1, ...,m) seriam os atributos que caracterizam cada possível classe. Haveriam as seguintes distribuições associadas a essas variáveis:

$\mathbf{P}$(Classe) e

$\mathbf{P}$(Atributo-k | Classe) para k = 1, ...,m.

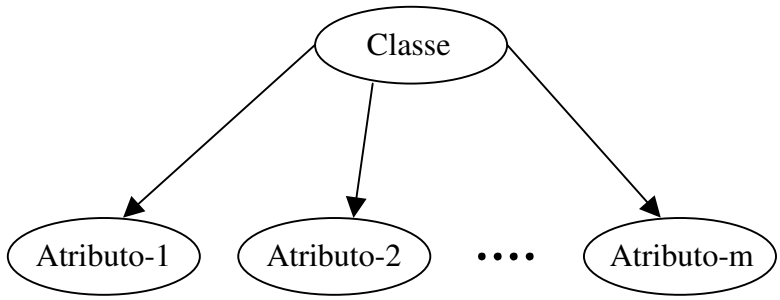


Figura 9**:** Um exemplo de RB.

A distribuição conjunta de todas as variáveis (X_1, X_2, ..., X_n) de uma RB é definida por

$$\mathbf{P}(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} \mathbf{P}(X_i \mid \text{Pais}(X_i)) .$$

Inferência probabilística é a computação da distribuição de probabilidade posterior para um conjunto de variáveis de consulta **C** dada alguma evidência observada. Essa evidência observada é uma atribuição de valores **e** feita a um conjunto de variáveis de evidência **E**. Seja $\mathbf{X} = \mathbf{C} \cup \mathbf{E} \cup \mathbf{Y}$ o conjunto de todas as variáveis aleatórias de uma RB, onde **Y** é o conjunto das demais variáveis não observadas que não são de consulta.

A inferência probabilística pode ser feita através do uso da distribuição conjunta de **X** da seguinte forma:

$$\mathbf{P}(\mathbf{C} \mid \mathbf{e}) = \frac{\mathbf{P}(\mathbf{C}, \mathbf{e})}{\mathrm{P}(\mathbf{e})} = \frac{\sum_{\mathbf{y}} \mathbf{P}(\mathbf{C}, \mathbf{e}, \mathbf{y})}{\sum_{\mathbf{c}} \sum_{\mathbf{y}} \mathrm{P}(\mathbf{c}, \mathbf{e}, \mathbf{y})}$$

onde $\mathbf{P}(\mathbf{C},\mathbf{e},\mathbf{y})$ e $\mathrm{P}(\mathbf{c},\mathbf{e},\mathbf{y})$ são provenientes da distribuição conjunta de $\mathbf{X} = (\mathbf{C} \cup \mathbf{E} \cup \mathbf{Y})$, e os somatórios são feitos sobre todas as possíveis combinações de valores **y** e **c** para as variáveis **Y** e **C**, respectivamente.

Infelizmente, inferência probabilística é um problema NP-*hard* [7], [6]. A não ser para alguns casos em que se fazem algumas restrições sobre o tipo da RB empregada (por exemplo, uma rede em que todas as variáveis são contínuas e representadas por Gaussianas), é necessária a utilização de algoritmos aproximados para inferência. Dois desses algoritmos são *likelihood weighting* e *Monte Carlo Markov Chain* (MCMC), ambos baseados na geração de amostras aleatórias a partir das distribuições de probabilidade $\mathbf{P}(X_i \mid \mathrm{Pais}(X_i))$ de cada variável aleatória X_i [47].

Normalmente as distribuições $\mathbf{P}(X_i \mid \mathrm{Pais}(X_i))$ de cada variável X_i de uma RB devem ser aprendidas a partir dos dados disponíveis. No caso em que se conhece a estrutura da rede (a topologia do grafo que a representa) e todas as variáveis são observáveis (ou seja, existem de forma explícita nos dados), o aprendizado é feito pela estimação por maximização da verossimilhança, *Maximum Likelihood* (ML) *estimation*. Se a estrutura é conhecida mas existem variáveis não observáveis emprega-se o algoritmo de gradiente ascendente [46] ou o algoritmo EM (*Expectation Maximization*) [32] para o aprendizado (internamente, ambos fazem uso do procedimento de inferência probabilística). O algoritmo EM é um método iterativo geral para obtenção de estimadores de máxima verossimilhança em situações onde temos dados incompletos. Quando todas as variáveis são observáveis mas a estrutura é desconhecida, deve-se realizar uma busca no espaço dos possíveis modelos de forma a otimizar uma função de avaliação (*scoring function*), como por exemplo, o BIC. Na situação em que existem variáveis não observáveis e a estrutura é desconhecida utiliza-se o algoritmo *Structural* EM [12] que combina a busca no espaço de modelos com o algoritmo EM.

3.2.1 - *Naive Bayes Classifier*

O *Naive Bayes Classifier* (NBC) [10] é um exemplo simples de uma RB que é aplicado a um problema de classificação (Figura 9). O NBC deve escolher uma classe s de um conjunto finito de valores discretos dom(S) dado uma conjunção de atributos [e_1, e_2, ..., e_m]. Vamos apenas considerar o caso mais simples em que todos os atributos são discretos. Primeiro é descoberta uma hipótese *maximum a posteriori* (MAP):

$$s_{MAP} = \arg\max_{s \in dom(S)} P(s \mid e_1, e_2, ..., e_m).$$

Pelo teorema de Bayes temos que

$$P(s \mid e_1, e_2, ..., e_m) = P(e_1, e_2, ..., e_m \mid s) \cdot P(s) / P(e_1, e_2, ..., e_m).$$

Assim a fórmula da hipótese MAP pode ser reescrita como

$$s_{MAP} = \arg\max_{s \in dom(S)} P(e_1, e_2, ..., e_m \mid s) \cdot P(s).$$

Como o NBC considera que os atributos são condicionalmente independentes dado a classe, temos que

$$P(e_1, e_2, ..., e_m \mid s) = \prod_{j=1}^{m} P(e_j \mid s).$$

Substituindo essa expressão na fórmula da hipótese MAP obtemos a fórmula da hipótese NB (*Naive Bayes*)

$$s_{NB} = \arg\max_{s \in dom(S)} P(s) \cdot \prod_{j=1}^{m} P(e_j \mid s).$$

Para usar essa expressão o NBC deve estimar as probabilidades referenciadas nela pela contagem dos valores discretos presentes em um conjunto de exemplos para treinamento. Para cada $s \in dom(S)$ e $e_j \in dom(E_j)$ (j = 1, 2, ..., m) nós computamos

$P(s) = N(s) / N$

$P(e_j|s) = P(e_j,s) / P(s) = N(e_j,s) / N(s)$

onde dom(S) e $dom(E_j)$ são conjuntos finitos de valores discretos, N é o número total de exemplos para treinamento, N(s) é o número de exemplos para treinamento com a classe "s", e $N(e_j,s)$ é o número de exemplos para treinamento com o atributo "e_j" e a classe "s". Para evitar uma contagem nula podemos adicionar M exemplos virtuais [33] ao conjunto de treinamento:

$P(s) = (N(s) + M.Q) / (N + M)$

$P(e_j|s) = (N(e_j,s) + M.Q.Q_j) / (N(s) + M.Q)$

onde $Q = 1/K$ se S tem K valores possíveis, e $Q_j = 1/K_j$ se E_j tem K_j valores possíveis.

3.3 - Redes Bayesianas Dinâmicas

Uma Rede Bayesiana Dinâmica (RBD) [15], [44], [47] é uma RB que representa um modelo probabilístico temporal como aquele que pode ser visto na Figura 10: em cada instante t, $\mathbf{S}_t$ é um conjunto de variáveis de estado (não observadas, podendo ser discretas ou contínuas) e $\mathbf{E}_t$ é um conjunto de variáveis de evidência (observadas, podendo ser discretas ou contínuas).

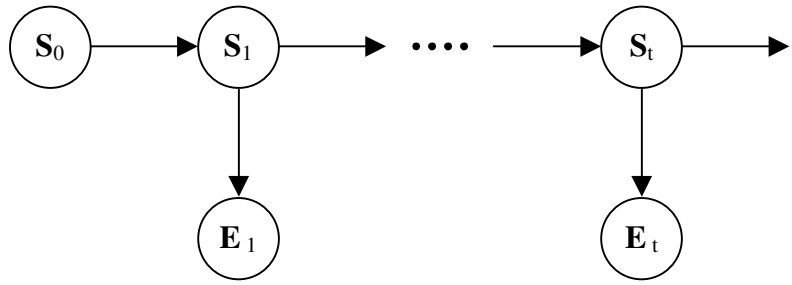


Figura 10**:** RBD de primeira ordem.

Três tipos importantes de inferência em uma RBD são a filtragem (*filtering*), a predição (*prediction*) e a suavização (*smoothing*). Na filtragem computa-se $\mathbf{P}(\mathbf{S}_t|\mathbf{e}_{1:t})$, na predição $\mathbf{P}(\mathbf{S}_{t+k}|\mathbf{e}_{1:t})$ para $k > 0$, e na suavização $\mathbf{P}(\mathbf{S}_k|\mathbf{e}_{1:t})$ para $1 \leq k < t$, onde $\mathbf{e}_{1:t}$ denota os conjuntos de valores $\mathbf{e}_1$, $\mathbf{e}_2$, …, $\mathbf{e}_t$ para os conjuntos de variáveis $\mathbf{E}_1$, $\mathbf{E}_2$, …, $\mathbf{E}_t$. Normalmente assume-se que o modelo é invariante no tempo (estacionário), ou seja, $\mathbf{P}(\mathbf{S}_t|\mathbf{S}_{t-1})$ e $\mathbf{P}(\mathbf{E}_t|\mathbf{S}_t)$ são os mesmos para todo t.

Como RBD's são RB's, os mesmos algoritmos para inferência probabilística em RB's podem ser empregados. Por questões de eficiência, geralmente emprega-se o algoritmo aproximado para inferência conhecido como *particle filtering* [47], que é uma modificação do algoritmo *likelihood weighting*.

Normalmente as distribuições $\mathbf{P}(\mathbf{S}_t|\mathbf{S}_{t-1})$ e $\mathbf{P}(\mathbf{E}_t|\mathbf{S}_t)$ de uma RBD devem ser aprendidas a partir dos dados disponíveis [37]. Da mesma forma que em uma rede Bayesiana, conhecendo-se a estrutura da rede e sendo todas as variáveis observáveis, o aprendizado é feito pela estimação ML. Se a estrutura é conhecida mas existem variáveis não observáveis, utilizam-se métodos de gradiente ou EM para o aprendizado. Quando a estrutura é desconhecida empregam-se extensões do algoritmo de busca no espaço de

modelos (todas as variáveis são observáveis) ou do *Structural* EM (existem variáveis não observáveis) [13].

3.3.1 - *Hidden Markov Models*

Um *Hidden Markov Model* (HMM) [40] é um caso particular de uma RBD onde cada $\mathbf{S}_t$ é constituído por uma única variável aleatória discreta. Como S_t não é observado nos dados de treinamento, a estimação das probabilidades $\mathbf{P}(S_{t+1}|S_t)$, $\mathbf{P}(\mathbf{E}_t|S_t)$ e $\mathbf{P}(S_0)$ geralmente é feita pelo algoritmo EM [15]. Se tanto S_t como $\mathbf{E}_t = (E_{t,1}, E_{t,2}, ..., E_{t,m})$ fossem observados nos dados de treinamento, essas probabilidades seriam calculadas por simples contagem:

$\mathbf{P}(S_0) = N(S_0) / N$

$\mathbf{P}(S_{t+1} \mid S_t) = N(S_{t+1}, S_t) / N(S_t) = N(S_{t+1}, S_t) / (\Sigma_{s_{t+1}} N(s_{t+1}, S_t))$

$\mathbf{P}(E_{t,j} \mid S_t) = N(E_{t,j}, S_t) / N(S_t) = N(E_{t,j}, S_t) / (\Sigma_{e_{t,j}} N(e_{t,j}, S_t))$, $1 \le j \le m$

onde N é o número total de exemplos para treinamento, N(.) é o número de exemplos para treinamento com valores distintos para uma variável, e N(., .) é o número de exemplos para treinamento com valores distintos para uma conjunção de variáveis. Para evitar uma contagem nula podemos adicionar M exemplos virtuais ao conjunto de treinamento:

$\mathbf{P}(S_0) = (N(S_0) + M.Q) / (N + M)$

$\mathbf{P}(S_{t+1} \mid S_t) = (N(S_{t+1}, S_t) + M.Q.Q) / (N(S_t) + M.Q)$

$\mathbf{P}(E_{t,j} \mid S_t) = (N(E_{t,j}, S_t) + M.Q.Q_j) / (N(S_t) + M.Q)$, $1 \le j \le m$

onde $Q = 1/K$ se S_t tem K valores possíveis, e $Q_j = 1/K_j$ se $E_{t,j}$ tem K_j valores possíveis.

Primeiramente vamos analisar um caso mais simples para aplicação do HMM. Considere um problema de classificação onde os exemplos de treinamento possuem uma dependência temporal tal que, para cada t, S_t é observado nos dados de treinamento e desconhecido (e portanto predito) nos dados de teste. Neste caso onde temos a estrutura do modelo conhecida e completa observabilidade, podemos usar a estimação ML para o aprendizado (sem a necessidade de um método iterativo como o EM).

Para esse problema cada exemplo é dado por uma classe s_t e uma conjunção de atributos $(e_{t,1}, e_{t,2}, ..., e_{t,m}) = \mathbf{e}_t$. Adicionalmente, assumimos que os atributos são condicionalmente independentes dado a classe. A qualquer instante t o HMM pode escolher uma classe $s_{HMM\,t}$ usando a evidência $\mathbf{e}_{1:t}$ por

$s_{HMM\,t} = \text{argmax}_{\,s_t \in dom(S_t)} P(s_t|\mathbf{e}_{1:t})$

onde a filtragem é feita da seguinte forma:

se t = 0 então $\mathbf{P}(S_t \mid \mathbf{e}_{1:t}) = \mathbf{P}(S_t)$

se t > 0 então $\mathbf{P}(S_t \mid \mathbf{e}_{1:t}) = \alpha.P(\mathbf{e}_t \mid S_t).(\Sigma_{s_{t-1}} \mathbf{P}(S_t \mid s_{t-1}).P(s_{t-1} \mid \mathbf{e}_{1:t-1})) =$

$$= \alpha.\{\prod_j P(e_{t,j}|S_t)\}.(\Sigma_{s_{t-1}} \mathbf{P}(S_t \mid s_{t-1}).P(s_{t-1} \mid \mathbf{e}_{1:t-1}))$$

e α é uma constante de normalização. No instante zero a filtragem corresponde à distribuição inicial $\mathbf{P}(S_0)$ pois não temos nenhum dado disponível, enquanto que nos demais instantes a filtragem depende da nova evidência disponível $\mathbf{e}_t$ e da filtragem realizada no instante anterior correspondente à distribuição $\mathbf{P}(S_{t-1} \mid \mathbf{e}_{1:t-1})$. O fator referente à nova evidência disponível é obtido da suposição de que os atributos são condicionalmente independentes dado a classe:

$$\mathbf{P}(\mathbf{e}_t \mid S_t) = \mathbf{P}(e_{t,1}, e_{t,2}, ..., e_{t,m} \mid S_t) = \prod_{j=1}^{m} \mathbf{P}(e_{t,j} \mid S_t) .$$

A fórmula recursiva da filtragem é obtida pela utilização das seguintes suposições (*Markov assumptions*):

$\mathbf{P}(S_t \mid s_{0:t-1}, \mathbf{e}_{1:t-1}) = \mathbf{P}(S_t \mid s_{t-1})$, (MA1)

$\mathbf{P}(\mathbf{E}_t \mid s_{0:t}, \mathbf{e}_{1:t-1}) = \mathbf{P}(\mathbf{E}_t \mid s_t)$. (MA2)

Portanto:

$\mathbf{P}(S_t|\mathbf{e}_{1:t}) = \mathbf{P}(S_t|\mathbf{e}_{1:t-1}, \mathbf{e}_t)$ [dividindo a evidência]

$= \alpha.P(\mathbf{e}_t|S_t,\mathbf{e}_{1:t-1}).\mathbf{P}(S_t|\mathbf{e}_{1:t-1})$ [pelo teorema de Bayes]

$= \alpha.P(\mathbf{e}_t|S_t).\mathbf{P}(S_t|\mathbf{e}_{1:t-1})$ [por (MA2)]

$= \alpha.P(\mathbf{e}_t|S_t).\sum_{st-1}\{\mathbf{P}(S_t|s_{t-1},\mathbf{e}_{1:t-1})p(s_{t-1}|\mathbf{e}_{1:t-1})\}$ [condicionando]

$= \alpha.P(\mathbf{e}_t|S_t).\sum_{st-1}\{\mathbf{P}(S_t|s_{t-1})P(s_{t-1}|\mathbf{e}_{1:t-1})\}$ [por (MA1)]

$= \alpha\{\prod_j P(e_{t,j}|S_t)\}\sum_{st-1}\{\mathbf{P}(S_t|s_{t-1})P(s_{t-1}|\mathbf{e}_{1:t-1})\}$ [pela independência]

No caso geral, S_t não é observado nos dados de treinamento e temos de empregar o algoritmo EM. Esse algoritmo usa o mecanismo de inferência do HMM com o propósito de computar as contagens N(.) e N(., .). O tipo de inferência que usamos aqui é a filtragem (cálculo de $\mathbf{P}(S_t \mid \mathbf{e}_{1:t})$) ao invés da suavização (que é a abordagem mais comum):

$N(S_t) = (\Sigma^T_{t=1} \mathbf{P}(S_t \mid \mathbf{e}_{1:t}))$

$N(S_{t+1}, S_t) = (\Sigma^T_{t=1} \mathbf{P}(S_{t+1}, S_t \mid \mathbf{e}_{1:t}))$

$N(E_{t,j}, S_t) = (\Sigma^T_{t=1} \mathbf{P}(E_{t,j}, S_t \mid \mathbf{e}_{1:t}))$, $1 \le j \le m$

onde T é o último instante disponível no conjunto de treinamento. As probabilidades $\mathbf{P}(S_{t+1},S_t \mid \mathbf{e}_{1:t})$ e $\mathbf{P}(E_{t,j},S_t \mid \mathbf{e}_{1:t})$ são inferidas por:

$\mathbf{P}(S_{t+1}, S_t \mid \mathbf{e}_{1:t}) = \mathbf{P}(S_{t+1} \mid S_t, \mathbf{e}_{1:t}).\mathbf{P}(S_t \mid \mathbf{e}_{1:t})$ [pelo teorema de Bayes]

$= \mathbf{P}(S_{t+1} \mid S_t).\mathbf{P}(S_t \mid \mathbf{e}_{1:t})$ [por (MA1)]

$\mathbf{P}(E_{t,j}, S_t \mid \mathbf{e}_{1:t}) = \mathbf{P}(E_{t,j} \mid S_t, \mathbf{e}_{1:t}).\mathbf{P}(S_t \mid \mathbf{e}_{1:t})$ [pelo teorema de Bayes]

$= 1.\mathbf{P}(S_t \mid \mathbf{e}_{1:t})$ se $E_{t,j} = e_{t,j}$, senão 0

EM é um procedimento iterativo: a fim de computar os parâmetros (probabilidades $\mathbf{P}(S_{t+1}|S_t)$ e $\mathbf{P}(\mathbf{E}_t|S_t)$) temos de calcular contagens pelo uso de inferência; e a inferência é feita através do uso dos parâmetros correntes. Esse processo se repete até alcançar uma condição de parada (por exemplo, um número máximo de iterações).

A previsão de uma observação futura $E_{t+1,j}$ $(1 \leq j \leq m)$ é feita pela computação de $\mathbf{P}(E_{t+1,j}|\mathbf{e}_{1:t})$ que também faz uso da filtragem:

$\mathbf{P}(E_{t+1,j} \mid \mathbf{e}_{1:t}) = (\Sigma_{s_{t+1}} \mathbf{P}(E_{t+1,j}, s_{t+1} \mid \mathbf{e}_{1:t}))$

onde

$\mathbf{P}(E_{t+1,j}, S_{t+1} \mid \mathbf{e}_{1:t}) = \mathbf{P}(E_{t+1,j} \mid S_{t+1}, \mathbf{e}_{1:t}).\ \mathbf{P}(S_{t+1} \mid \mathbf{e}_{1:t})$ [pelo teorema de Bayes]

$= \mathbf{P}(E_{t+1,j} \mid S_{t+1}).\ \mathbf{P}(S_{t+1} \mid \mathbf{e}_{1:t})$ [por (MA2)]

$= \mathbf{P}(E_{t+1,j} \mid S_{t+1}).(\Sigma_{s_t} \mathbf{P}(S_{t+1}, s_t \mid \mathbf{e}_{1:t}))$ [marginalizando]

$= \mathbf{P}(E_{t+1,j} \mid S_{t+1}).(\Sigma_{s_t} \mathbf{P}(S_{t+1} \mid s_t, \mathbf{e}_{1:t}).P(s_t \mid \mathbf{e}_{1:t}))$ [pelo teorema de Bayes]

$= \mathbf{P}(E_{t+1,j} \mid S_{t+1}).(\Sigma_{s_t} \mathbf{P}(S_{t+1} \mid s_t).P(s_t \mid \mathbf{e}_{1:t}))$ [por (MA1)]

3.3.2 - Modelos de Filtro de Kalman

Um outro caso particular de uma RBD é aquele em que cada um dos conjuntos $\mathbf{S}_t$ e $\mathbf{E}_t$ é constituído por variáveis aleatórias contínuas. RBD's com essas características são chamadas de modelos de filtro de Kalman ou modelos de espaço de estados [47], [17], [74] (vistos no Capítulo 2, embora tenha sido mostrado apenas o caso em que $\mathbf{E}_t$ é constituído por uma única variável Z_t). Esses modelos empregam distribuições Gaussianas lineares (ou seja, variáveis são funções lineares de outras variáveis que possuem distribuições Gaussianas) e o procedimento de filtragem é realizado pelo filtro de Kalman. Assim temos um modelo descrito pelas equações

- equação das observações: $\mathbf{E}_t = \mathbf{F}_t\mathbf{S}_t + \boldsymbol{\nu}_t$ $\boldsymbol{\nu}_t \sim N[\mathbf{0}, \mathbf{V}_t]$

- equação do estado: $\mathbf{S}_t = \mathbf{G}_t\mathbf{S}_{t-1} + \boldsymbol{\omega}_t$ $\boldsymbol{\omega}_t \sim N[\mathbf{0}, \mathbf{W}_t]$

ou de forma equivalente

$P(\mathbf{e}_t \mid \mathbf{s}_t) = N[\mathbf{F}_t\mathbf{s}_t, \sum_{\mathbf{e}_t}](\mathbf{e}_t)$

$P(\mathbf{s}_t \mid \mathbf{s}_{t-1}) = N[\mathbf{G}_t\mathbf{s}_{t-1}, \sum_{\mathbf{s}_t}](\mathbf{s}_t)$

onde

$\sum_{\mathbf{e}_t} = \mathbf{F}_t.\mathrm{Var}(\mathbf{S}_t).\mathbf{F}_t' + \mathbf{V}_t$

$\sum_{\mathbf{s}_t} = \mathbf{G}_t.\mathrm{Var}(\mathbf{S}_{t-1}).\mathbf{G}_t' + \mathbf{W}_t$

com a distribuição Gaussiana multivariada dada por

$$N[\boldsymbol{\mu}, \boldsymbol{\Sigma}](\mathbf{x}) = \alpha \cdot e^{-\frac{1}{2}((\mathbf{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}))}$$

sendo α uma constante de normalização.

Como visto na Seção 2.4 sobre modelos de espaço de estados, a transição

$(\mathbf{S}_{t-1} \mid \mathbf{e}_{1:t-1}) \rightarrow (\mathbf{S}_t \mid \mathbf{e}_{1:t})$

representa a atualização das variáveis de estado pelo procedimento de filtragem (neste caso, o filtro de Kalman). Sendo conhecida a distribuição de $(\mathbf{S}_{t-1} \mid \mathbf{e}_{1:t-1})$, a transição inicial

$(\mathbf{S}_{t-1} \mid \mathbf{e}_{1:t-1}) \rightarrow (\mathbf{S}_t \mid \mathbf{e}_{1:t-1})$

é realizada por

$$\mathbf{P}(\mathbf{S}_t \mid \mathbf{e}_{1:t-1}) = \int_{\mathbf{s}_{t-1}} \mathbf{P}(\mathbf{S}_t \mid \mathbf{s}_{t-1}) \cdot P(\mathbf{s}_{t-1} \mid \mathbf{e}_{1:t-1})\, d\mathbf{s}_{t-1}$$

e a transição final

$(\mathbf{S}_t \mid \mathbf{e}_{1:t-1}) \rightarrow (\mathbf{S}_t \mid \mathbf{e}_{1:t})$

é feita por

$$\mathbf{P}(\mathbf{S}_t \mid \mathbf{e}_{1:t}) = \alpha \cdot \mathbf{P}(\mathbf{e}_t \mid \mathbf{S}_t) \cdot \mathbf{P}(\mathbf{S}_t \mid \mathbf{e}_{1:t-1})$$

sendo α uma constante de normalização. Como estamos trabalhando com distribuições Gaussianas lineares, os resultados dessas transições também são Gaussianas. No caso de outras distribuições, não existe uma solução analítica para a equação que representa a filtragem (equação da transição inicial inserida na equação da transição final)

$$\mathbf{P}(\mathbf{S}_t \mid \mathbf{e}_{1:t}) = \alpha \cdot \mathbf{P}(\mathbf{e}_t \mid \mathbf{S}_t) \cdot \int_{\mathbf{s}_{t-1}} \mathbf{P}(\mathbf{S}_t \mid \mathbf{s}_{t-1}) \cdot P(\mathbf{s}_{t-1} \mid \mathbf{e}_{1:t-1})\, d\mathbf{s}_{t-1}$$

e, assim, um método como o MCMC deve ser empregado para aproximar o resultado dessa equação.

4 - Previsão através da Estimação Não-Paramétrica de uma Densidade Contínua

Este capítulo descreve Preditores Probabilísticos Discretos: sistemas de previsão que fazem uso da estimação não-paramétrica de uma densidade contínua.

4.1 - Objetivo

O objetivo deste trabalho foi explorar uma abordagem de previsão de séries temporais pouco investigada na literatura: a previsão de um valor contínuo através da estimação não-paramétrica da função densidade de probabilidade da variável aleatória contínua que se deseja prever. A expressão "previsão de um valor contínuo" significa prever um valor para uma variável aleatória contínua; de forma similar, "previsão de um valor discreto" significa prever um valor para uma variável aleatória discreta. O ajuste não-paramétrico da função densidade empregado aqui é o histograma, que corresponde a uma discretização dos dados. Uma vez feita essa discretização, Redes Bayesianas (RB's) ou Redes Bayesianas Dinâmicas (RBD's) utilizando variáveis aleatórias discretas são responsáveis pela inferência probabilística necessária para a estimação da densidade. Tendo em mãos a estimação não-paramétrica da densidade, o cálculo do valor esperado produz a previsão de um valor contínuo.

Este capítulo consiste na apresentação de vários sistemas de previsão que estimam uma densidade de forma não-paramétrica: *Naive Bayes for Regression* (NBR) [11], *Markov Model for Regression* (MMR) [62], [65], *Hidden Markov Model for Regression* (HMMR) [65] e *Multi-Hidden Markov Model for Regression* (MHMMR) [67]. O NBR é baseado no NBC enquanto que os demais (frutos do trabalho desta tese) são baseados no HMM.

4.2 - Preditores Probabilísticos Discretos

Preditores Probabilísticos Discretos (PPD's) fazem a previsão de um valor contínuo através da estimação não-paramétrica de uma densidade contínua. Em PPD's, o ajuste

não-paramétrico é feito por RB's ou RBD's utilizando apenas variáveis discretas. Tudo funciona com se uma RB ou RBD com apenas variáveis contínuas fosse aproximada por uma RB ou RBD equivalente com apenas variáveis discretas.

Para se prever uma série temporal (Z_1, Z_2, ... Z_T), existem três possíveis abordagens a serem seguidas na definição do espaço dos dados contínuos que dependem do PPD empregado. A primeira abordagem (empregada pelo NBR) define os espaços de entrada $\mathbf{A} = (A_1, A_2, ..., A_m) = (Z_{t-m+1}, ...,Z_{t-1}, Z_t)$ e saída $V = Z_{t+1}$ dos dados contínuos, que é bem similar ao que é feito nos sistemas *fuzzy*: dada uma entrada contínua $\mathbf{a} = (a_1, a_2, ..., a_m)$ o PPD prevê o valor contínuo $v = z_{t+1}$. A segunda abordagem (empregada pelo MMR) introduz um índice de tempo t nas variáveis do PPD, definindo os espaços de entrada $\mathbf{A}_t = (A_{t,1}, A_{t,2}, ..., A_{t,m}) = (Z_{t-m+1}, ...,Z_{t-1}, Z_t)$ e saída $V_t = Z_{t+1}$: dada uma entrada contínua $\mathbf{a}_t = (a_{t,1}, a_{t,2}, ..., a_{t,m})$ o PPD prevê o valor contínuo $v_t = z_{t+1}$. Na terceira abordagem (empregada pelo HMMR e MHMMR) é definido apenas o espaço das observações contínuas $\mathbf{A}_t = (A_{t,1}, A_{t,2}, ..., A_{t,m}) = (Z_{t-m+1}, ...,Z_{t-1}, Z_t)$: dada uma observação contínua $\mathbf{a}_t = (a_{t,1}, a_{t,2}, ..., a_{t,m})$ o PPD prevê o valor contínuo $a_{t+1,m} = z_{t+1}$.

4.2.1 - *Naive Bayes for Regression*

O *Naive Bayes for Regression* (NBR) [11] é o NBC aplicado à regressão pela discretização dos espaços de entrada e saída dos dados contínuos: para cada valor desejado (v) / atributo (a_j) contínuos existe um valor discreto correspondente (pseudo-classe s / atributo e_j) representando o intervalo que contém esse valor contínuo. O processo de aprendizado pode ser visto como a transformação de um conjunto de treinamento constituído por dados contínuos em um conjunto de dados discretos que são usados para o treinamento de um NBC.

O processo para se prever um valor contínuo é apresentado na Figura 11. Primeiro ocorre a discretização dos atributos contínuos $\mathbf{a} = (a_1, a_2, ..., a_m)$ produzindo atributos discretos $\mathbf{e} = (e_1, e_2, ..., e_m)$. A inferência feita pelo NBC emprega a distribuição a priori $\mathbf{P}(S)$ para produzir a distribuição $\mathbf{P}(S|\mathbf{e})$ que incorpora os novos atributos discretos. Essas distribuições são representadas por histogramas: r1, r2, ..., rn são os possíveis valores de cada variável aleatória discreta (S e E_j, $1 \leq j \leq m$); as colunas são os valores das probabilidades para r1, r2, ..., rn. A previsão contínua é feita pela seguinte fórmula:

$$v_{NBR} = \Sigma_{s \in dom(S)} \{m(s).P(s|\mathbf{e})\}$$

$$\mathbf{P}(S|\mathbf{e}) = \mathbf{P}(S).\prod_j P(e_j|S) / (\sum_s P(s).\prod_j P(e_j|s)) = \alpha.\mathbf{P}(S).\prod_j P(e_j|S)$$

onde m(s) é o valor central do intervalo s e α é uma constante de normalização.

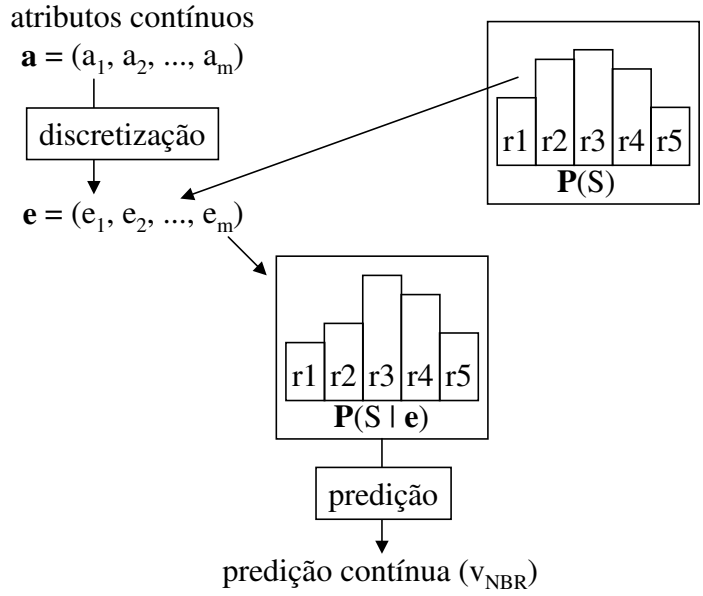


Figura 11: NBR - previsão de um valor contínuo.

De fato, o NBR está estimando a seguinte função densidade de probabilidade:

$$f(v|\mathbf{a}) = f(v \mid a_1, a_2, \ldots, a_m) = f(v).\prod_j f(a_j|v) / \int(f(v).\prod_j f(a_j|v))dv$$

através do uso de estimativas de histograma [49]:

$$f(v) \approx P(s) / h_s \text{ , } v \in s$$

$$f(a_j, v) \approx P(e_j, s) / (h_{e_j} . h_s) \text{ , } a_j \in e_j \text{ e } v \in s$$

$$f(a_j|v) = f(a_j, v) / f(v) \approx P(e_j|s) / h_{e_j} \text{ , } a_j \in e_j \text{ e } v \in s$$

onde h_{e_j} e h_s são os tamanhos dos intervalos e_j e s, respectivamente.

Substituindo as estimativas de f(v) e $f(a_j|v)$ em $f(v|\mathbf{a})$ obtém-se:

$$f(v|\mathbf{a}) \approx P(s|\mathbf{e}) / h_s$$

para $v \in s$, $a_1 \in e_1$, $a_2 \in e_2$, ... e $a_m \in e_m$.

Tendo em mãos a estimativa da função densidade de probabilidade $f(v|\mathbf{a})$, o próximo passo é calcular o valor esperado da variável contínua V (conhecidos os valores contínuos $a_1, a_2, \ldots, a_m$). Esse valor esperado é a previsão do NBR (v_{NBR}):

$$v_{NBR} = \int v.f(v|\mathbf{a})dv \approx \sum_s m(s).P(s|\mathbf{e})$$

que é exatamente o resultado da previsão contínua do NBR apresentado inicialmente. Detalhes sobre esta demonstração são apresentados no Apêndice A.

4.2.2 - *Hidden Markov Model for Regression*

O *Hidden Markov Model for Regression* (HMMR) [65] é o HMM aplicado a um problema de regressão pela discretização do espaço das observações contínuas: para cada observação contínua ($a_{t,j}$) existe um valor discreto correspondente ($e_{t,j}$) representando o intervalo que contém o valor contínuo. De forma similar ao NBR, o aprendizado pode ser visto como a transformação de um conjunto de treinamento formado por dados contínuos em um conjunto de dados discretos usados para o treinamento de um HMM.

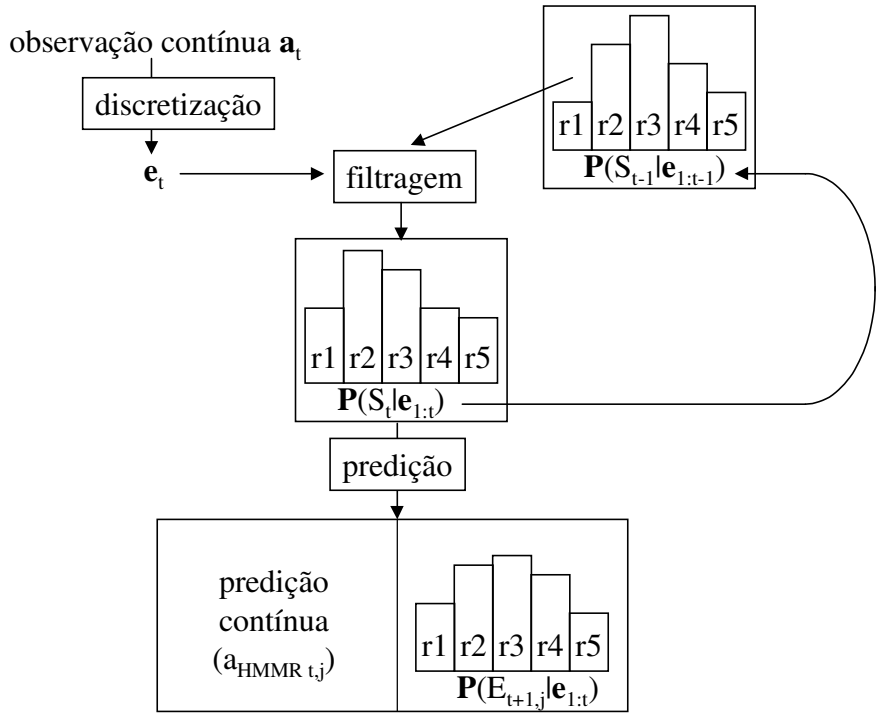


Figura 12: HMMR - previsão de um valor contínuo.

A Figura 12 descreve de forma geral o funcionamento do processo para previsão de um valor contínuo. Em primeiro lugar acontece a discretização da observação contínua $\mathbf{a}_t = (a_{t,1}, a_{t,2}, ..., a_{t,m})$ produzindo uma observação discreta $\mathbf{e}_t = (e_{t,1}, e_{t,2}, ..., e_{t,m})$. A filtragem realizada pelo HMM utiliza a distribuição a priori $\mathbf{P}(S_{t-1}|\mathbf{e}_{1:t-1})$ para gerar a distribuição $\mathbf{P}(S_t|\mathbf{e}_{1:t})$ que incorpora a nova observação discreta. A seta que conecta essas duas distribuições representa o passo de atualização feito para o processamento da próxima observação. Após a filtragem, a distribuição $\mathbf{P}(E_{t+1,j}|\mathbf{e}_{1:t})$ de uma observação

discreta futura (na verdade, uma componente j dessa observação) é calculada. A previsão contínua é obtida pela seguinte fórmula:

$$a_{HMMR\ t,j} = \sum_{e_{t+1,j} \in\ dom(E_{t+1,j})} \{m(e_{t+1,j}).P(e_{t+1,j} \mid \mathbf{e}_{1:t})\}$$

$$\mathbf{P}(E_{t+1,j} \mid \mathbf{e}_{1:t}) = \Sigma_{s_{t+1}} \mathbf{P}(E_{t+1,j} \mid s_{t+1}).(\Sigma_{s_t} \mathbf{P}(s_{t+1} \mid s_t).P(s_t \mid \mathbf{e}_{1:t}))$$

$$= \Sigma_{s_{t+1}} \mathbf{P}(E_{t+1,j} \mid s_{t+1}).P(s_{t+1} \mid \mathbf{e}_{1:t})$$

onde j ($1 \leq j \leq m$) é uma componente selecionada da evidência e $m(e_{t+1,j})$ é o valor central do intervalo $e_{t+1,j}$.

Da mesma forma que o NBR, o HMMR também está estimando uma função densidade de probabilidade, neste caso a densidade da previsão da variável de evidência:

$$f(a_{t+1,j} \mid \mathbf{a}_{1:t}) = \int \{f(a_{t+1,j} \mid v_{t+1}).f(v_{t+1} \mid \mathbf{a}_{1:t})\} dv_{t+1}$$

onde a densidade da previsão da variável de estado é dada por

$$f(v_{t+1} \mid \mathbf{a}_{1:t}) = \int \{f(v_{t+1} \mid v_t).f(v_t \mid \mathbf{a}_{1:t})\} dv_t$$

e a densidade da filtragem da variável de estado é

se $t = 0$ então $f(v_t \mid \mathbf{a}_{1:t}) = f(v_t)$

se $t > 0$ então $f(v_t \mid \mathbf{a}_{1:t}) = \beta.f(\mathbf{a}_t \mid v_t).f(v_t \mid \mathbf{a}_{1:t-1}) = \beta.\{\prod_j f(a_{t,j}|v_t)\}.f(v_t \mid \mathbf{a}_{1:t-1})$

com a constante de normalização $\beta = 1 / \int \{f(\mathbf{a}_t \mid v_t).f(v_t \mid \mathbf{a}_{1:t-1})\} dv_t$.

A estimativa de $f(a_{t+1,j} \mid \mathbf{a}_{1:t})$ faz uso das seguintes estimativas de histograma:

$f(v_t) \approx P(s_t) / h_{s_t}$, $v_t \in s_t$

$f(a_{t,j}, v_t) \approx P(e_{t,j} , s_t) / (h_{e_{t,j}} . h_{s_t})$, $a_{t,j} \in e_{t,j}$ e $v_t \in s_t$

$f(a_{t,j}|v_t) = f(a_{t,j}, v_t) / f(v_t) \approx P(e_{t,j}|s_t) / h_{e_{t,j}}$, $a_{t,j} \in e_{t,j}$ e $v_t \in s_t$

$f(v_{t+1}, v_t) \approx P(s_{t+1} , s_t) / (h_{s_{t+1}} . h_{s_t})$, $v_{t+1} \in s_{t+1}$ e $v_t \in s_t$

$f(v_{t+1}|v_t) = f(v_{t+1}, v_t) / f(v_t) \approx P(s_{t+1}|s_t) / h_{s_{t+1}}$, $v_{t+1} \in s_{t+1}$ e $v_t \in s_t$

onde $h_{e_{t,j}}$ e h_{s_t} são os tamanhos dos intervalos $e_{t,j}$ e s_t, respectivamente. É importante lembrar que os tamanhos dos intervalos não mudam com o tempo, isto é, $h_{e_{t,j}} = h_{e_j}$ e $h_{s_t} = h_s$ para todo instante t.

Substituindo as estimativas de $f(v_t)$, $f(v_{t+1}|v_t)$ e $f(a_{t,j}|v_t)$ em $f(a_{t+1,j} \mid \mathbf{a}_{1:t})$ obtém-se:

$$f(a_{t+1,j} \mid \mathbf{a}_{1:t}) \approx P(e_{t+1,j} \mid \mathbf{e}_{1:t}) / h_{e_j}$$

para $a_{t+1,j} \in e_{t+1,j}$, ($a_{1,1} \in e_{1,1}$, ... e $a_{1,m} \in e_{1,m}$) ... e ($a_{t,1} \in e_{t,1}$, ... e $a_{t,m} \in e_{t,m}$).

O valor esperado da variável contínua $A_{t+1,j}$ (condicionada pelas observações conhecidas até o instante t) é a previsão do HMMR ($a_{HMMR\ t,j}$):

$a_{HMMR\ t,j} = \int a_{t+1,j}.f(a_{t+1,j}|\mathbf{a}_{1:t})da_{t+1,j} \approx \sum_{e_{t+1,j}}\{m(e_{t+1,j}).P(e_{t+1,j}|\mathbf{e}_{1:t})\}$

que corresponde ao resultado da previsão contínua do HMMR apresentado antes. Detalhes sobre esta demonstração são apresentados no Apêndice A.

4.2.3 - *Markov Model for Regression*

O *Markov Model for Regression* (MMR) [65] é uma versão simplificada do HMMR assumindo a variável de estado do modelo sendo observável nos dados de treinamento. Aqui o HMM é aplicado a um problema de regressão pela discretização dos espaços de entrada e saída dos dados contínuos: para cada valor desejado (v_t) / atributo ($a_{t,j}$) contínuos existe um valor discreto correspondente (pseudo-classe s_t / atributo $e_{t,j}$) representando o intervalo que contém esse valor contínuo. Como todas as variáveis do HMM são observadas no conjunto de treinamento (obtido da discretização dos dados contínuos), não é utilizado o algoritmo EM para o cálculos dos parâmetros e sim simples contagens (estimação ML). O MMR é bem parecido com o NBR mas agora existe uma dependência temporal entre os exemplos.

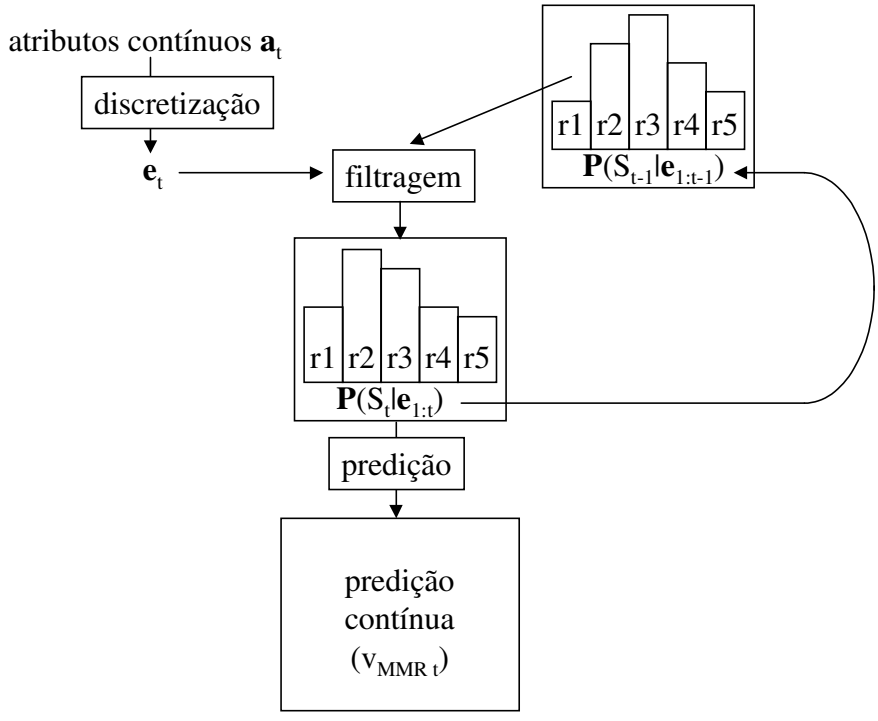


Figura 13: MMR - previsão de um valor contínuo.

Na Figura 13 é mostrado o processo de previsão de um valor contínuo no MMR. Esse processo é similar ao do HMMR e a única diferença está na não computação da distribuição $\mathbf{P}(E_{t+1,j}|\mathbf{e}_{1:t})$. Depois que a distribuição $\mathbf{P}(S_t|\mathbf{e}_{1:t})$ é obtida, a previsão contínua é executada pela seguinte fórmula:

$$v_{MMR\,t} = \sum_{s_t \in dom(S_t)} \{m(s_t).P(s_t \mid \mathbf{e}_{1:t})\}$$

onde $m(s_t)$ é o valor central do intervalo s_t.

A função densidade de probabilidade sendo estimada pelo MMR é a densidade da filtragem da variável de estado:

se $t = 0$ então $f(v_t \mid \mathbf{a}_{1:t}) = f(v_t)$

se $t > 0$ então $f(v_t \mid \mathbf{a}_{1:t}) = \beta.f(\mathbf{a}_t \mid v_t).f(v_t \mid \mathbf{a}_{1:t-1}) = \beta.\{\prod_j f(a_{t,j}|v_t)\}.f(v_t \mid \mathbf{a}_{1:t-1})$

com a constante de normalização $\beta = 1 / \int\{f(\mathbf{a}_t \mid v_t).f(v_t \mid \mathbf{a}_{1:t-1})\}dv_t$.

Através das mesmas estimativas de histograma do HMMR, obtém-se a seguinte estimativa de $f(v_t \mid \mathbf{a}_{1:t})$ para o MMR:

$$f(v_t|\mathbf{a}_{1:t}) \approx P(s_t \mid \mathbf{e}_{1:t}) / h_s$$

para $v_t \in s_t$, $(a_{1,1} \in e_{1,1}$, ... e $a_{1,m} \in e_{1,m})$ ... e $(a_{t,1} \in e_{t,1}$, ... e $a_{t,m} \in e_{t,m})$.

O valor esperado da variável contínua V_t (condicionada pelos valores contínuos $\mathbf{a}_{1:t}$) é a previsão do MMR ($v_{MMR\,t}$):

$$v_{MMR\,t} = \int v_t.f(v_t|\mathbf{a}_{1:t})dv_t \approx \sum_{s_t}\{m(s_t).P(s_t|\mathbf{e}_{1:t})\}$$

que é o mesmo resultado inicialmente apresentado da previsão contínua do MMR. Detalhes sobre esta demonstração são apresentados no Apêndice A.

4.2.4 - *Multi-Hidden Markov Model for Regression*

O *Multi-Hidden Markov Model for Regression* (MHMMR) [67] é uma versão mais geral do HMMR assumindo que não existe apenas uma única variável de estado. Neste caso o ajuste não-paramétrico da densidade contínua do modelo é realizado por uma RBD similar ao HMM mas com várias ($w > 1$) variáveis de estado $\mathbf{S}_t = (S_{t,1}, S_{t,2}, ..., S_{t,w})$. Neste *Multi-Hidden Markov Model* (MHMM) [67], conhecido na literatura por HMM estruturado [23], cada variável de estado no instante t é condicionalmente dependente de todas as variáveis de estado no instante t-1, e cada variável de evidência no instante t é condicionalmente dependente de todas as variáveis de estado no instante t. A Figura 14 mostra um MHMM com 2 variáveis de estado (as variáveis de evidência em um mesmo instante não são colocadas explicitamente). O MHMM é aplicado a um

problema de regressão pela discretização do espaço das observações contínuas: para cada observação contínua ($a_{t,j}$) há um valor discreto correspondente ($e_{t,j}$) que representa o intervalo contendo esse valor contínuo. O aprendizado pode ser visto como a transformação de um conjunto de treinamento constituído por dados contínuos em um conjunto de dados discretos que são empregados para o treinamento de um MHMM.

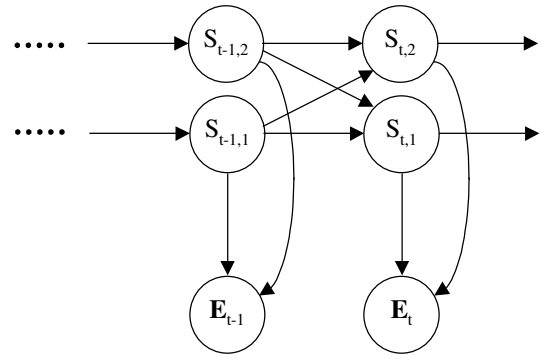


Figura 14: MHMM com 2 variáveis de estado.

O processo para previsão de um valor contínuo no MHMMR pode ser visto na Figura 15. É praticamente igual ao do HMMR só que, ao invés de termos uma única variável de estado S_t, temos um conjunto de variáveis de estado $\mathbf{S}_t = (S_{t,1}, S_{t,2}, ..., S_{t,w})$. Primeiro ocorre a discretização da observação contínua $\mathbf{a}_t = (a_{t,1}, a_{t,2}, ..., a_{t,m})$ gerando uma observação discreta $\mathbf{e}_t = (e_{t,1}, e_{t,2}, ..., e_{t,m})$. Através da filtragem no MHMM, faz-se uso da distribuição a priori $\mathbf{P}(\mathbf{S}_{t-1}|\mathbf{e}_{1:t-1})$ para gerar a distribuição $\mathbf{P}(\mathbf{S}_t|\mathbf{e}_{1:t})$ incorporando a nova observação discreta. A distribuição $\mathbf{P}(E_{t+1,j}|\mathbf{e}_{1:t})$ de uma observação discreta futura é calculada após essa filtragem e a previsão contínua é dada por:

$$a_{MHMMR\ t,j} = \sum_{e_{t+1,j} \in\ dom(E_{t+1,j})} \{m(e_{t+1,j}).P(e_{t+1,j} \mid \mathbf{e}_{1:t})\}$$

$$\mathbf{P}(E_{t+1,j} \mid \mathbf{e}_{1:t}) = \Sigma_{s_{t+1}} \mathbf{P}(E_{t+1,j} \mid \mathbf{s}_{t+1}).(\Sigma_{\mathbf{s}_t} \mathbf{P}(\mathbf{s}_{t+1} \mid \mathbf{s}_t).P(\mathbf{s}_t \mid \mathbf{e}_{1:t}))$$

$$= \Sigma_{s_{t+1}} \mathbf{P}(E_{t+1,j} \mid \mathbf{s}_{t+1}).P(\mathbf{s}_{t+1} \mid \mathbf{e}_{1:t})$$

onde j ($1 \leq j \leq m$) é uma componente selecionada da evidência, $m(e_{t+1,j})$ é o valor central do intervalo $e_{t+1,j}$, e a filtragem é feita por:

se t = 0 então $\mathbf{P}(\mathbf{S}_t \mid \mathbf{e}_{1:t}) = \mathbf{P}(\mathbf{S}_t)$

se t > 0 então $\mathbf{P}(\mathbf{S}_t \mid \mathbf{e}_{1:t}) = \alpha.\{\prod_j P(e_{t,j}|\mathbf{S}_t)\}.(\Sigma_{\mathbf{s}_{t-1}} \mathbf{P}(\mathbf{S}_t \mid \mathbf{s}_{t-1}).P(\mathbf{s}_{t-1} \mid \mathbf{e}_{1:t-1}))$

com α como uma constante de normalização. Por uma questão de simplicidade, as variáveis de estado em um mesmo instante não são colocadas explicitamente nas fórmulas. Todas essas distribuições são representadas por histogramas:

- r1, r2, ..., rk são os possíveis valores para a variável aleatória discreta $E_{t,j}$, enquanto que as colunas são os valores das probabilidades para r1, r2, ..., rk;
- **r1**, **r2**, ..., **rn** são os possíveis conjuntos de valores para o conjunto de variáveis aleatórias discretas $\mathbf{S}_t$, enquanto que as colunas são os valores das probabilidades para **r1**, **r2**, ..., **rn**.

O MHMMR está estimando a mesma função densidade de probabilidade que o HMMR, ou seja, a densidade da previsão da variável de evidência. A diferença é que agora o cálculo dessa densidade pressupõe que os dados contínuos são modelados por uma RBD com w variáveis contínuas de estado $\mathbf{V}_t = (V_{t,1}, V_{t,2}, ..., V_{t,w})$ mas mantendo as mesmas m variáveis contínuas de evidência $\mathbf{A}_t = (A_{t,1}, A_{t,2}, ..., A_{t,m})$:

$$f(a_{t+1,j} \mid \mathbf{a}_{1:t}) = \int\{f(a_{t+1,j} \mid \mathbf{v}_{t+1}).f(\mathbf{v}_{t+1} \mid \mathbf{a}_{1:t})\}d\mathbf{v}_{t+1}$$

onde a densidade da previsão do conjunto das variáveis de estado é dada por

$$f(\mathbf{v}_{t+1} \mid \mathbf{a}_{1:t}) = \int\{f(\mathbf{v}_{t+1} \mid \mathbf{v}_t).f(\mathbf{v}_t \mid \mathbf{a}_{1:t})\}d\mathbf{v}_t$$

e a densidade da filtragem do conjunto das variáveis de estado é

se $t = 0$ então $f(\mathbf{v}_t \mid \mathbf{a}_{1:t}) = f(\mathbf{v}_t)$

se $t > 0$ então $f(\mathbf{v}_t \mid \mathbf{a}_{1:t}) = \beta.f(\mathbf{a}_t \mid \mathbf{v}_t).f(\mathbf{v}_t \mid \mathbf{a}_{1:t-1}) = \beta.\{\prod_j f(a_{t,j}|\mathbf{v}_t)\}.f(\mathbf{v}_t \mid \mathbf{a}_{1:t-1})$

com a constante de normalização $\beta = 1 / \int\{f(\mathbf{a}_t \mid \mathbf{v}_t).f(\mathbf{v}_t \mid \mathbf{a}_{1:t-1})\}d\mathbf{v}_t$.

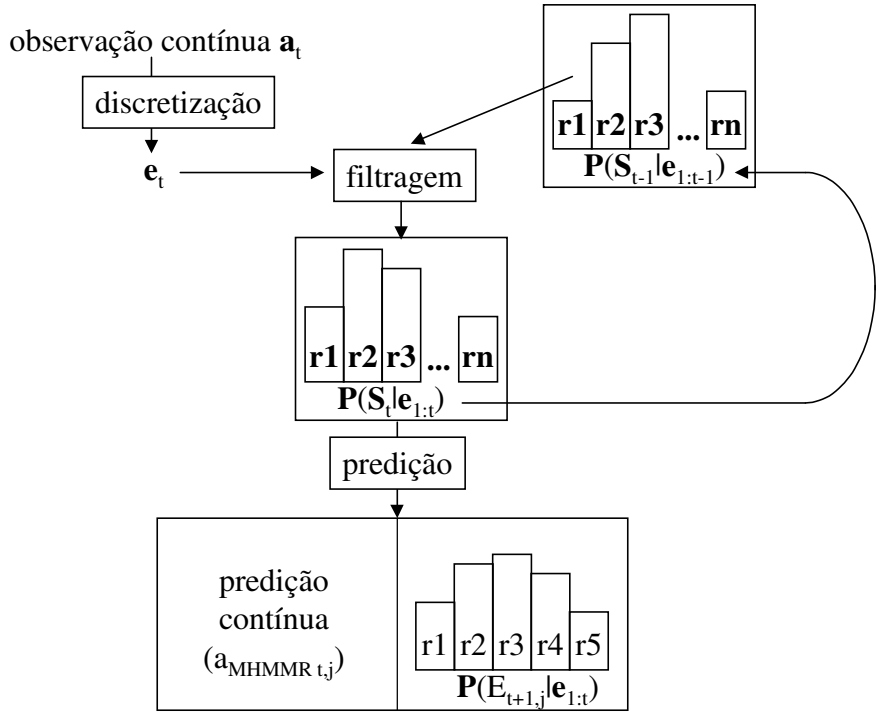


Figura 15: MHMMR - previsão de um valor contínuo.

A estimativa de $f(a_{t+1,j} \mid \mathbf{a}_{1:t})$ faz uso das seguintes estimativas de histograma:

$f(\mathbf{v}_t) = f(v_{t,1}, ..., v_{t,w}) \approx P(s_{t,1}, ..., s_{t,w}) / \prod_k h_{s_k} = P(\mathbf{s}_t) / \prod_k h_{s_k}$, para $\mathbf{v}_t \in \mathbf{s}_t$

(ou seja, para $v_{1,1} \in s_{1,1}$, ... e $v_{1,m} \in s_{1,w}$)

$f(a_{t,j}, \mathbf{v}_t) = f(a_{t,j}, v_{t,1}, ..., v_{t,w}) \approx P(e_{t,j}, s_{t,1}, ..., s_{t,w}) / (h_{e_j} . \prod_k h_{s_k}) =$

$= P(e_{t,j}, \mathbf{s}_t) / (h_{e_j} . \prod_k h_{s_k})$, $a_{t,j} \in e_{t,j}$ e $\mathbf{v}_t \in \mathbf{s}_t$

$f(a_{t,j}|\mathbf{v}_t) = f(a_{t,j}, \mathbf{v}_t) / f(\mathbf{v}_t) \approx P(e_{t,j}|\mathbf{s}_t) / h_{e_j}$, $a_{t,j} \in e_{t,j}$ e $\mathbf{v}_t \in \mathbf{s}_t$

$f(\mathbf{v}_{t+1}, \mathbf{v}_t) = f(v_{t+1,1}, ..., v_{t+1,w}, v_{t,1}, ..., v_{t,w}) \approx P(s_{t+1,1}, ..., s_{t+1,w}, s_{t,1}, ..., s_{t,w})/(\prod_k h_{s_k})^2 =$

$= P(\mathbf{s}_{t+1}, \mathbf{s}_t) / (\prod_k h_{s_k})^2$, $\mathbf{v}_{t+1} \in \mathbf{s}_{t+1}$ e $\mathbf{v}_t \in \mathbf{s}_t$

$f(\mathbf{v}_{t+1}|\mathbf{v}_t) = f(\mathbf{v}_{t+1}, \mathbf{v}_t) / f(\mathbf{v}_t) \approx P(\mathbf{s}_{t+1}|\mathbf{s}_t) / \prod_k h_{s_k}$, $\mathbf{v}_{t+1} \in \mathbf{s}_{t+1}$ and $\mathbf{v}_t \in \mathbf{s}_t$

Através dessas estimativas de histograma obtém-se a estimativa para $f(a_{t+1,j} \mid \mathbf{a}_{1:t})$:

$f(a_{t+1,j}|\mathbf{a}_{1:t}) \approx P(e_{t+1,j}|\mathbf{e}_{1:t}) / h_{e_j}$

para $a_{t+1,j} \in e_{t+1,j}$, ($a_{1,1} \in e_{1,1}$, ... e $a_{1,m} \in e_{1,m}$) ... e ($a_{t,1} \in e_{t,1}$, ... e $a_{t,m} \in e_{t,m}$).

A previsão do MHMMR ($a_{MHMMR\ t,j}$) é o valor esperado da variável contínua $A_{t+1,j}$ condicionada pelas observações conhecidas até o instante t:

$a_{MHMMR\ t,j} = \int a_{t+1,j}.f(a_{t+1,j}|\mathbf{a}_{1:t})da_{t+1,j} \approx \sum_{e_{t+1,j}}\{m(e_{t+1,j}).P(e_{t+1,j}|\mathbf{e}_{1:t})\}$

que é exatamente igual à previsão contínua do MHMMR vista anteriormente. Detalhes sobre esta demonstração são apresentados no Apêndice A.

5 - Previsão pela Estimação Não-Paramétrica *Fuzzy* de uma Densidade Contínua

Este capítulo descreve Preditores Probabilísticos *Fuzzy*: sistemas de previsão que fazem uso de uma abordagem *fuzzy* para a estimação não-paramétrica de uma densidade contínua.

5.1 - Uma Abordagem *Fuzzy* da Previsão pela Estimação Não-Paramétrica de uma Densidade Contínua

A principal contribuição deste trabalho foi a generalização dos PPD's vistos no capítulo anterior pelo uso da fuzzificação no lugar da discretização. Desenvolvemos assim uma abordagem *fuzzy* da previsão através da estimação não-paramétrica da função densidade da variável contínua que se deseja prever. Uma vez feita a fuzzificação dos dados contínuos, RB's ou RBD's utilizando variáveis aleatórias *fuzzy* (e probabilidades *fuzzy* [69], [75]) são responsáveis pela inferência probabilística necessária para a estimação da densidade. A partir dessa estimação, o cálculo do valor esperado produz a previsão de um valor contínuo. Serão apresentados os seguintes sistemas probabilísticos *fuzzy* para previsão: *Fuzzy Bayes Predictor* (FBP) [60], [64], *Fuzzy Markov Predictor* (FMP) [61], [63], [64], *Fuzzy Hidden Markov Predictor* (FHMP) [65] e *Fuzzy Multi-Hidden Markov Predictor* (FMHMP) [67]. O FBP é baseado no NBC enquanto que os demais são baseados no HMM.

5.2 - Preditores Probabilísticos *Fuzzy*

Preditores Probabilísticos *Fuzzy* (PPF's) fazem a previsão de um valor contínuo através da estimação não-paramétrica *fuzzy* de uma densidade contínua. Em PPF's, o ajuste não-paramétrico é feito por RB's ou RBD's utilizando variáveis e probabilidades *fuzzy*.

Probabilidades *fuzzy* [69], [75] surgem quando variáveis aleatórias são consideradas *fuzzy*. A probabilidade de uma variável *fuzzy* B ser igual a região *fuzzy* rk é:

$P(B = rk) = E(m_{rk})$

onde $E(m_{rk})$ é o valor esperado da função de pertinência da região *fuzzy* rk. Esse valor esperado pode ser estimado a partir de uma amostra constituída por vários valores contínuos:

$E(m_{rk}) \approx (\sum_{x \in amostra} m_{rk}(x)) / |amostra|$

onde |amostra| é o tamanho da amostra, ou seja, a quantidade de valores contínuos que a amostra contém.

A probabilidade de uma conjunção de variáveis *fuzzy* A e B, tal que A seja igual a rj e B igual a rk, é dada por:

$P(A = rj, B = rk) = E(m_{rj} \times m_{rk})$

onde $E(m_{rj} \times m_{rk})$ é o valor esperado do produto das funções de pertinência das regiões *fuzzy* rj e rk. Esse valor esperado também pode ser estimado a partir de uma amostra:

$E(m_{rj} \times m_{rk}) \approx (\sum_{(x, y) \in amostra} m_{rj}(x) \times m_{rk}(y)) / |amostra|$

Utilizando essas probabilidades *fuzzy*, outras podem ser obtidas como, por exemplo, a probabilidade *fuzzy* condicional:

$P(A = rj \mid B = rk) = P(A = rj, B = rk) / P(B = rk)$

Na previsão de uma série temporal (Z_1, Z_2, ... Z_T), o espaço dos dados contínuos de um PPF é definido da mesma forma que nos PPD's:

- definir os espaços de entrada $\mathbf{A} = (A_1, ..., A_m) = (Z_{t-m+1}, ..., Z_t)$ e saída $V = Z_{t+1}$ (empregada pelo FBP); ou
- definir os espaços de entrada $\mathbf{A}_t = (A_{t,1}, ..., A_{t,m}) = (Z_{t-m+1}, ..., Z_t)$ e saída $V_t = Z_{t+1}$ (empregada pelo FMP); ou
- definir apenas o espaço das observações contínuas $\mathbf{A}_t = (A_{t,1}, ..., A_{t,m}) = (Z_{t-m+1}, ..., Z_t)$ (empregada pelo FHMP e FMHMP).

Em um sistema *fuzzy* convencional, várias regras *fuzzy* são ativadas quando é apresentada uma entrada contínua $\mathbf{a} = (a_1, ..., a_m)$. Cada regra possui um grau de ativação (dado pelo grau de saída da regra) e seu conseqüente informa a qual região *fuzzy* deve pertencer a saída contínua. Os graus de ativação e as regiões *fuzzy* de saída fornecidos pelas regras são combinados para a obtenção da saída contínua v do sistema *fuzzy*. Em um PPF, várias distribuições de probabilidade para regiões *fuzzy* são ativadas quando é apresentada uma entrada (ou uma observação) contínua. Cada uma dessas

distribuições possui um grau de ativação. As distribuições *fuzzy* e seus graus de ativação são combinados para a obtenção de uma única distribuição *fuzzy* que é usada para o cálculo da saída contínua do PPF.

5.2.1 - *Fuzzy Bayes Predictor*

O *Fuzzy Bayes Predictor* (FBP) é similar ao NBR mas utiliza fuzzificação ao invés de discretização: no NBR, os espaços de entrada e saída dos dados contínuos são divididos em intervalos; no FBP, esses espaços são divididos em regiões *fuzzy*. Para cada valor desejado (v) / atributo (a_j) contínuos existe um conjunto de valores *fuzzy* correspondente $\{s \mid m_s(v) > 0\}$ (pseudo-classe) / $\{e_j \mid m_{e_j}(a_j) > 0\}$ (atributo). O FBP possui a mesma topologia que o NBR, mas as variáveis aleatórias S e E_j ($1 \leq j \leq m$) não são mais discretas e sim *fuzzy*. As probabilidades *fuzzy* $\mathbf{P}(S)$ e $\mathbf{P}(E_j|S)$ são calculadas não por uma simples contagem de valores discretos e suas conjunções mas pelo somatório de pertinências *fuzzy* e seus produtos:

$\mathbf{P}(S) = N(S) / N$

$\mathbf{P}(E_j \mid S) = N(E_j , S) / N(S) , 1 \leq j \leq m$

onde N é o número total de exemplos para treinamento, N(.) e N(., .) são definidos por:

$N(S) = (\sum_v m_S(v))$

$N(E_j , S) = (\sum_{(a_j, v)} m_{E_j}(a_j) \times m_S(v))$

onde $\sum_v$ é o somatório para cada valor contínuo v proveniente do conjunto de treinamento e, de forma similar, $\sum_{(a_j, v)}$ é o somatório para cada par de valores contínuos (a_j, v). Para evitar uma contagem nula podemos adicionar M exemplos virtuais ao conjunto de treinamento:

$\mathbf{P}(S) = (N(S) + M.Q) / (N + M)$

$\mathbf{P}(E_j \mid S) = (N(E_j , S) + M.Q.Q_j) / (N(S) + M.Q) , 1 \leq j \leq m$

onde $Q = 1/K$ se S tem K valores (regiões *fuzzy*) possíveis, e $Q_j = 1/K_j$ se E_j tem K_j valores possíveis.

A Figura 16 apresenta o processo de previsão de uma valor contínuo feito pelo FBP. Em primeiro lugar acontece a fuzzificação da tupla de atributos contínuos $\mathbf{a} = (a_1, a_2, ..., a_m)$ produzindo um conjunto de tuplas de atributos *fuzzy* $d = \{\mathbf{e} \mid m_{\mathbf{e}}(\mathbf{a}) > 0\}$ onde $\mathbf{e} = (e_1, e_2, ..., e_m)$ e $m_{\mathbf{e}}(\mathbf{a}) = m_{e_1}(a_1) \times m_{e_2}(a_2) \times ... \times m_{e_m}(a_m)$. Partindo-se da distribuição a priori

P(S), várias distribuições condicionais **P**(S|**e**) são produzidas de forma a incorporar cada uma das possíveis tuplas de atributos *fuzzy* **e** $\in$ d:

$$\mathbf{P}(S|\mathbf{e}) = \mathbf{P}(S).\prod_j P(e_j|S) \,/\, (\sum_s P(s).\prod_j P(e_j|s)) = \alpha.\mathbf{P}(S).\prod_j P(e_j|S)$$

onde α é uma constante de normalização. Essas distribuições condicionais são integradas em uma única distribuição condicional **P**(S|d) através de um mecanismo de defuzzificação:

$$\mathbf{P}(S|d) = (\sum_{\mathbf{e} \in d} m_{\mathbf{e}}(\mathbf{a}).\mathbf{P}(S|\mathbf{e})) \,/\, (\sum_{\mathbf{e} \in d} m_{\mathbf{e}}(\mathbf{a}))$$

Supondo que $\sum_{r \in dom(R)} m_r(x) = 1$ para todo valor contínuo x e toda variável *fuzzy* R, ou seja, considerando um sistema de informação *fuzzy* ortogonal [69], a fórmula de defuzzificação é simplificada para:

$$\mathbf{P}(S|d) = \sum_{\mathbf{e} \in d} m_{\mathbf{e}}(\mathbf{a}).\mathbf{P}(S|\mathbf{e})$$

Para cada **e** $\in$ d, $m_{\mathbf{e}}(\mathbf{a})$ poder ser vista como o grau de ativação da distribuição **P**(S|**e**). Todas essas distribuições *fuzzy* são representadas por histogramas: r1, r2, ..., rn são os possíveis valores (regiões *fuzzy*) de cada variável aleatória *fuzzy* (S e E_j, $1 \le j \le m$); as colunas são os valores das probabilidades para r1, r2, ..., rn. A previsão contínua é feita pela seguinte fórmula:

$$v_{FBP} = \Sigma_s \ \overline{s}.P(s|d)$$

onde $\overline{s}$ é o centro de gravidade da região *fuzzy* s, ou seja, $\overline{s} = \int v.m_s(v)dv \,/\, \int m_s(v)dv$.

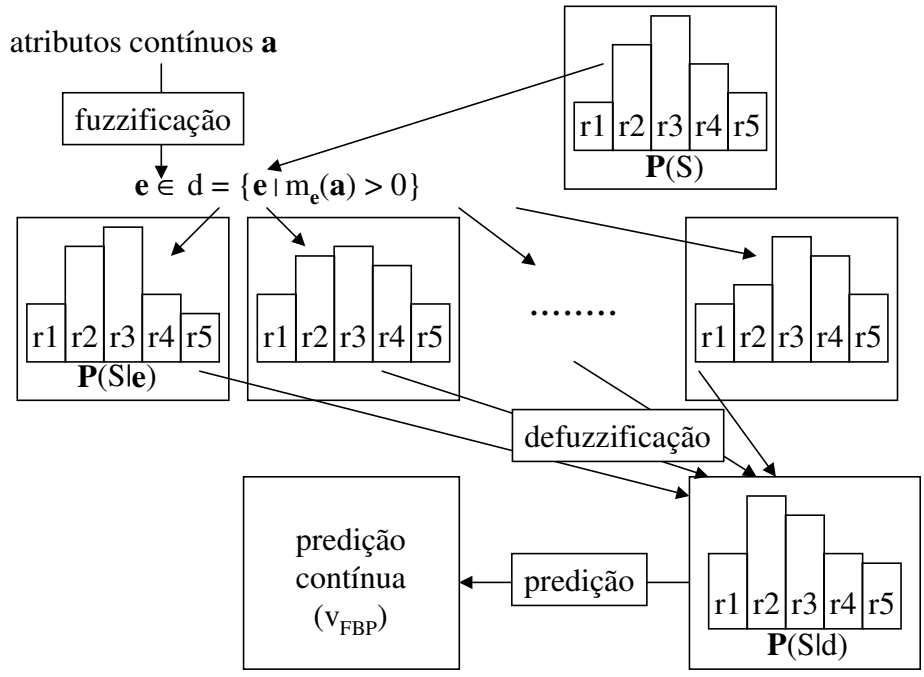


Figura 16: FBP - previsão de um valor contínuo.

O FBP está aproximando a mesma função densidade de probabilidade estimada pelo NBR:

$$f(v|\mathbf{a}) = f(v).\prod_j f(a_j|v) \,/\, \int(f(v).\prod_j f(a_j|v))dv$$

através da seguinte fórmula de defuzzificação:

$$f(v|\mathbf{a}) \approx (\sum_s \sum_{\mathbf{e} \in d} m_s(v).m_{\mathbf{e}}(\mathbf{a}).f_{s,\mathbf{e}}(v|\mathbf{a})) \,/\, (\sum_s \sum_{\mathbf{e} \in d} m_s(v).m_{\mathbf{e}}(\mathbf{a}))$$

que é simplificada para (supondo um sistema de informação *fuzzy* ortogonal):

$$f(v|\mathbf{a}) \approx \sum_s \sum_{\mathbf{e} \in d} m_s(v).m_{\mathbf{e}}(\mathbf{a}).f_{s,\mathbf{e}}(v|\mathbf{a})$$

e cada $f_{s,\mathbf{e}}(v|\mathbf{a})$ sendo combinado pela defuzzificação é dado por:

$$f_{s,\mathbf{e}}(v|\mathbf{a}) = P(s|\mathbf{e}) \,/\, h_s, \text{ para } m_s(v).m_{\mathbf{e}}(\mathbf{a}) > 0$$

onde P(.) e P(.|.) são probabilidades *fuzzy*, e h_s é a área da região *fuzzy* s ($h_s = \int m_s(v)dv$). Se as funções de pertinência do FBP forem escolhidas de forma que a fuzzificação corresponda a uma discretização dos dados contínuos então a aproximação de $f(v|\mathbf{a})$ feita pelo FBP torna-se igual àquela feita pelo NBR.

O valor esperado da variável contínua V condicionada pelos valores contínuos a_1, a_2, ..., a_m corresponde à previsão do FBP (v_{FBP}):

$$v_{FBP} = \int v.f(v|\mathbf{a})dv \approx \sum_s \overline{s}\,.\, P(s|d)$$

que é o mesmo resultado da previsão contínua do FBP mostrado anteriormente. Detalhes sobre esta demonstração são apresentados no Apêndice A. Se as funções de pertinência do FBP forem selecionadas de forma que a fuzzificação corresponda a uma discretização então a previsão contínua do FBP torna-se igual a do NBR.

5.2.2 - *Fuzzy Hidden Markov Predictor*

O *Fuzzy Hidden Markov Predictor* (FHMP) é semelhante ao HMMR, embora use fuzzificação no lugar da discretização: no HMMR, o espaço das observações contínuas é dividido em intervalos; no FHMP, esse espaço é dividido em regiões *fuzzy*. Para cada observação contínua ($a_{t,j}$) existe um conjunto de valores *fuzzy* correspondente $\{e_{t,j} \mid m_{e_{t,j}}(a_{t,j}) > 0\}$. O FHMP possui a mesma topologia que o HMMR, só que agora S_t e $E_{t,j}$ ($1 \le j \le m$) são variáveis aleatórias *fuzzy*.

A estimação das probabilidades *fuzzy* $\mathbf{P}(S_{t+1}|S_t)$ e $\mathbf{P}(E_{t,j}|S_t)$ é realizada pelo algoritmo EM. Depois do aprendizado dessas probabilidades (que será detalhado posteriormente), a previsão de um valor contínuo pode ser efetuada (Figura 17). Para a previsão, primeiro ocorre a fuzzificação da tupla de observações contínuas $\mathbf{a}_t = (a_{t,1}, a_{t,2}, ..., a_{t,m})$

produzindo um conjunto de tuplas de observações *fuzzy* $d_t = \{\mathbf{e}_t \mid m_{\mathbf{e}_t}(\mathbf{a}_t) > 0\}$ onde $\mathbf{e}_t = (e_{t,1}, e_{t,2}, ..., e_{t,m})$ e $m_{\mathbf{e}_t}(\mathbf{a}_t) = m_{e_{t,1}}(a_{t,1}) \times m_{e_{t,2}}(a_{t,2}) \times ... \times m_{e_{t,m}}(a_{t,m})$. A filtragem efetuada pelo FHMP faz uso da distribuição a priori $\mathbf{P}(S_{t-1}|d_{1:t-1})$ a fim de produzir várias distribuições $\mathbf{P}(S_t|d_{1:t-1},\mathbf{e}_t)$ condicionadas a cada nova tupla de observações *fuzzy* $\mathbf{e}_t \in d_t$:

$$\mathbf{P}(S_t|d_{1:t-1},\mathbf{e}_t) = \alpha.P(\mathbf{e}_t|S_t).(\Sigma_{s_{t-1}} \mathbf{P}(S_t|s_{t-1}).P(s_{t-1}|d_{1:t-1})) =$$

$$= \alpha.\{\prod_j P(e_{t,j}|S_t)\}.(\Sigma_{s_{t-1}} \mathbf{P}(S_t|s_{t-1}).P(s_{t-1}|d_{1:t-1}))$$

onde $d_{1:t-1} = (d_1, d_2, \ldots, d_{t-1})$, e α é uma constante de normalização. Por defuzzificação, essas distribuições condicionais são integradas em uma única distribuição condicional $\mathbf{P}(S_t|d_{1:t})$:

$$\mathbf{P}(S_t|d_{1:t}) = (\sum_{\mathbf{e}_t \in d_t} m_{\mathbf{e}_t}(\mathbf{a}_t).\mathbf{P}(S_t|d_{1:t-1},\mathbf{e}_t)) / (\sum_{\mathbf{e}_t \in d_t} m_{\mathbf{e}}(\mathbf{a}_t))$$

ou (considerando um sistema de informação *fuzzy* ortogonal)

$$\mathbf{P}(S_t|d_{1:t}) = \sum_{\mathbf{e}_t \in d_t} m_{\mathbf{e}_t}(\mathbf{a}_t).\mathbf{P}(S_t|d_{1:t-1},\mathbf{e}_t)$$

A seta ligando $\mathbf{P}(S_t|d_{1:t})$ a $\mathbf{P}(S_{t-1}|d_{1:t-1})$ representa o passo de atualização feito para o processamento da próxima tupla de observações *fuzzy*. Para cada $\mathbf{e}_t \in d_t$, $m_{\mathbf{e}_t}(\mathbf{a}_t)$ pode ser vista como o grau de ativação de cada uma das distribuições $\mathbf{P}(S_t|d_{1:t-1},\mathbf{e}_t)$. Após a filtragem (cálculo de $\mathbf{P}(S_t|d_{1:t})$), a distribuição $\mathbf{P}(E_{t+1,j}|d_{1:t})$ $(1 \leq j \leq m)$ de uma observação *fuzzy* futura é computada:

$$\mathbf{P}(E_{t+1,j}|d_{1:t}) = (\Sigma_{s_{t+1}} \mathbf{P}(E_{t+1,j}, s_{t+1}|d_{1:t}))$$

onde

$$\mathbf{P}(E_{t+1,j}, S_{t+1}|d_{1:t}) = \mathbf{P}(E_{t+1,j}|S_{t+1},d_{1:t}).\ \mathbf{P}(S_{t+1}|d_{1:t}) \quad \text{[pelo teorema de Bayes]}$$

$$= \mathbf{P}(E_{t+1,j}|S_{t+1}).\ \mathbf{P}(S_{t+1}|d_{1:t}) \quad \text{[por (FMA2)]}$$

$$= \mathbf{P}(E_{t+1,j}|S_{t+1}).(\Sigma_{s_t} \mathbf{P}(S_{t+1},s_t|d_{1:t})) \quad \text{[marginalizando]}$$

$$= \mathbf{P}(E_{t+1,j}|S_{t+1}).(\Sigma_{s_t} \mathbf{P}(S_{t+1}|s_t,d_{1:t}).P(s_t|d_{1:t})) \quad \text{[pelo teorema de Bayes]}$$

$$= \mathbf{P}(E_{t+1,j}|S_{t+1}).(\Sigma_{s_t} \mathbf{P}(S_{t+1}|s_t).P(s_t|d_{1:t})) \quad \text{[por (FMA1)]}$$

considerando as *fuzzy Markov assumptions*:

$$\mathbf{P}(S_t|s_{0:t-1},d_{1:t-1}) = \mathbf{P}(S_t|s_{t-1}) \quad \text{(FMA1)}$$

$$\mathbf{P}(\mathbf{E}_t|s_{0:t},d_{1:t-1}) = \mathbf{P}(\mathbf{E}_t|s_t) \quad \text{(FMA2)}$$

que são generalizações das *Markov assumptions* (MA1) e (MA2) feitas com o propósito de permitir conjuntos de tuplas de observações *fuzzy* $(d_{1:t-1})$ nessas probabilidades condicionais. A previsão contínua é obtida pela seguinte fórmula:

$$a_{\text{FHMP } t,j} = \sum_{e_{t+1,j}} \overline{e}_{t+1,j}.P(e_{t+1,j}|d_{1:t})$$

onde j ($1 \leq j \leq m$) é uma componente selecionada da tupla de observações e $\overline{e}_{t+1,j}$ é o centro de gravidade da região *fuzzy* $e_{t+1,j}$, ou seja, $\overline{e}_{t+1,j} = \int a.m_{e_{t+1,j}}(a)da \,/\, \int m_{e_{t+1,j}}(a)da$.

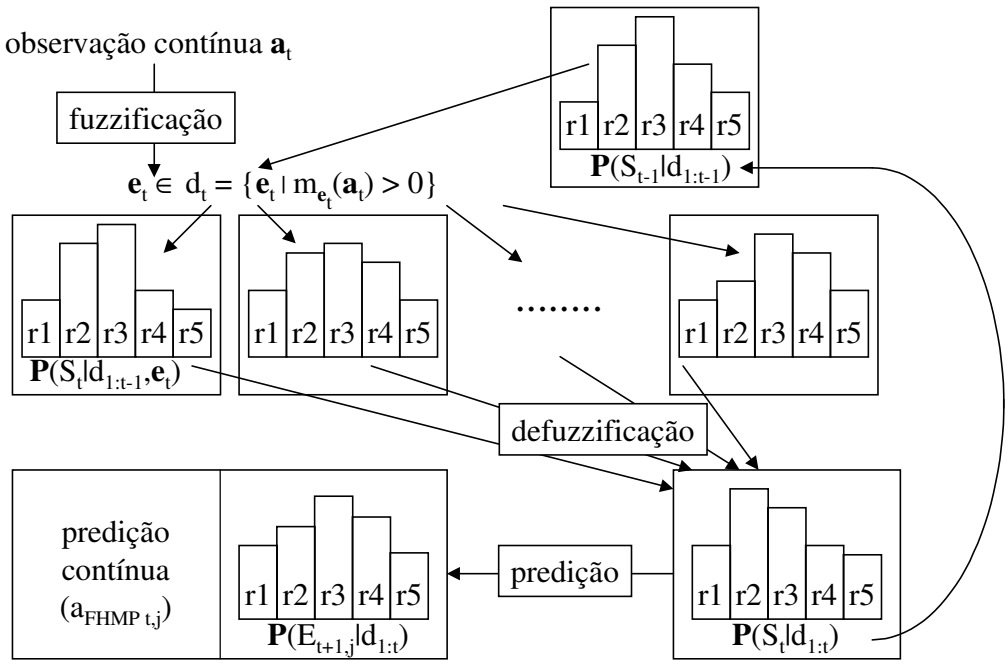


Figura 17: FHMP - previsão de um valor contínuo.

O algoritmo EM estima as probabilidades $\mathbf{P}(S_{t+1}|S_t)$ e $\mathbf{P}(\mathbf{E}_t|S_t)$ de forma iterativa. Partindo-se de uma estimativa inicial dessas probabilidades, o algoritmo EM usa o mecanismo de inferência do FHMP para obter uma nova estimativa delas. Esse processo é repetido até que se alcance uma condição de parada:

$\mathbf{P}(S_0) = N(S_0) / N$

$\mathbf{P}(S_{t+1}|S_t) = N(S_{t+1},S_t) / N(S_t)$

$\mathbf{P}(E_{t,j}|S_t) = N(E_{t,j},S_t) / N(S_t)$, $1 \leq j \leq m$

onde N é o número total de exemplos para treinamento, N(.) e N(., .) são computados por:

$N(S_t) = (\Sigma^T_{t=1}\ \mathbf{P}(S_t|d_{1:t}))$

$N(S_{t+1},S_t) = (\Sigma^T_{t=1}\ \mathbf{P}(S_{t+1},S_t|d_{1:t}))$

$N(E_{t,j},S_t) = (\Sigma^T_{t=1}\ \mathbf{P}(E_{t,j},S_t|d_{1:t}))$, $1 \leq j \leq m$

onde T é o último instante disponível no conjunto de treinamento. As probabilidades $\mathbf{P}(S_{t+1},S_t|d_{1:t})$ e $\mathbf{P}(E_{t,j},S_t|d_{1:t})$ são inferidas por:

$\mathbf{P}(S_{t+1},S_t|d_{1:t}) = \mathbf{P}(S_{t+1}|S_t,d_{1:t}).\mathbf{P}(S_t|d_{1:t})$ [pelo teorema de Bayes]

$= \mathbf{P}(S_{t+1}|S_t).\mathbf{P}(S_t|d_{1:t})$ [por (FMA1)]

$\mathbf{P}(E_{t,j},S_t|d_{1:t}) = \mathbf{P}(E_{t,j}|S_t,d_{1:t}).\mathbf{P}(S_t|d_{1:t})$ [pelo teorema de Bayes]

$= m_{E_{t,j}}(a_{t,j}).\mathbf{P}(S_t|d_{1:t})$

onde $\mathbf{P}(S_t|d_{1:t})$, $\mathbf{P}(S_{t+1},S_t|d_{1:t})$ e $\mathbf{P}(E_{t,j},S_t|d_{1:t})$ utilizam as probabilidades $\mathbf{P}(S_{t+1}|S_t)$ e $\mathbf{P}(\mathbf{E}_t|S_t)$ estimadas na iteração anterior. Se ao invés de filtragem fosse empregada suavização, N(.) e N(., .) seriam definidos por:

$N(S_t) = (\Sigma^T_{t=1}\ \mathbf{P}(S_t|d_{1:T}))$

$N(S_{t+1},S_t) = (\Sigma^T_{t=1}\ \mathbf{P}(S_{t+1},S_t|d_{1:T}))$

$N(E_{t,j},S_t) = (\Sigma^T_{t=1}\ \mathbf{P}(E_{t,j},S_t|d_{1:T}))\ ,\ 1 \le j \le m$

É importante frisar que essas equações para estimar as probabilidades *fuzzy* $\mathbf{P}(S_{t+1}|S_t)$ e $\mathbf{P}(\mathbf{E}_t|S_t)$ são generalizações das equações vistas no capítulo anterior para a estimação das probabilidades discretas utilizadas pelo HMMR. Embora seja possível demonstrar que o procedimento iterativo do EM converge para um máximo local de uma função de verossimilhança no caso discreto, o mesmo ainda não pode ser afirmado no caso *fuzzy*. Uma demonstração formal da convergência para o caso *fuzzy* deve ser investigada futuramente.

A função densidade de probabilidade sendo aproximada pelo FHMP é igual àquela estimada pelo HMMR:

$f(a_{t+1,j}|\mathbf{a}_{1:t}) = \int\{f(a_{t+1,j}|v_{t+1}).f(v_{t+1}|\mathbf{a}_{1:t})\}dv_{t+1}$

O FHMP utiliza a seguinte fórmula de defuzzificação para aproximar a densidade da previsão da variável de evidência:

$f(a_{t+1,j}|\mathbf{a}_{1:t}) \approx (\sum_{e_{t+1,j}} m_{e_{t+1,j}}(a_{t+1,j}).f_{e_{t+1,j}}(a_{t+1,j}|\mathbf{a}_{1:t})) / (\sum_{e_{t+1,j}} m_{e_{t+1,j}}(a_{t+1,j}))$

que é simplificada para (supondo um sistema de informação *fuzzy* ortogonal):

$f(a_{t+1,j}|\mathbf{a}_{1:t}) \approx \sum_{e_{t+1,j}} m_{e_{t+1,j}}(a_{t+1,j}).f_{e_{t+1,j}}(a_{t+1,j}|\mathbf{a}_{1:t})$

e cada $f_{e_{t+1,j}}(a_{t+1,j}|\mathbf{a}_{1:t})$ sendo combinado pela defuzzificação é dado por:

$f_{e_{t+1,j}}(a_{t+1,j}|\mathbf{a}_{1:t}) = P(e_{t+1,j}|d_{1:t}) / h_{e_j}$, para $m_{e_{t+1,j}}(a_{t+1,j}) > 0$

onde $P(e_{t+1,j}|d_{1:t})$ é a probabilidade de uma observação *fuzzy* futura $e_{t+1,j}$, e h_{e_j} é a área da região *fuzzy* $e_{t+1,j}$ ($h_{e_j} = \int m_{e_{t+1,j}}(a_{t+1,j})da_{t+1,j}$). Se for feita a escolha de funções de pertinência a fim de que a fuzzificação corresponda a uma discretização dos dados contínuos então a aproximação de $f(a_{t+1,j}|\mathbf{a}_{1:t})$ realizada pelo FHMP torna-se igual àquela efetuada pelo HMMR.

O valor esperado da variável contínua $A_{t+1,j}$ condicionada pelos valores contínuos $\mathbf{a}_{1:t}$ equivale à previsão do FHMP ($a_{FHMP\ t,j}$):

$a_{FHMP\ t,j} = \int a_{t+1,j}.f(a_{t+1,j}|\mathbf{a}_{1:t})da_{t+1,j} \approx \sum_{e_{t+1,j}} \overline{e}_{t+1,j} \,.\, P(e_{t+1,j}|d_{1:t})$

que corresponde à previsão contínua do FHMP apresentada inicialmente. Detalhes sobre esta demonstração são apresentados no Apêndice A. Essa previsão torna-se igual a do HMMR quando as funções de pertinência são selecionadas de forma que a fuzzificação se iguale a uma discretização dos dados contínuos.

5.2.3 - *Fuzzy Markov Predictor*

O *Fuzzy Markov Predictor* (FMP) é uma versão simplificada do FHMP que assume a variável de estado do modelo observável nos dados de treinamento. O FMP é uma generalização do MMR através do uso da fuzzificação como substituta da discretização: no MMR, os espaços de entrada e saída dos dados contínuos são divididos em intervalos; no FMP, esses espaços são divididos em regiões *fuzzy*. Para cada valor desejado (v_t) / atributo ($a_{t,j}$) contínuos existe um conjunto de valores *fuzzy* correspondente $\{s_t \mid m_{s_t}(v_t) > 0\}$ (pseudo-classe) / $\{e_{t,j} \mid m_{e_{t,j}}(a_{t,j}) > 0\}$ (atributo). A topologia do FMP é igual a do MMR, mas agora S_t e $E_{t,j}$ ($1 \le j \le m$) são variáveis aleatórias *fuzzy*. De forma similar ao MMR, o FMP não utiliza o algoritmo EM para a estimação das probabilidades *fuzzy* $\mathbf{P}(S_{t+1}|S_t)$ e $\mathbf{P}(E_{t,j}|S_t)$. Essas probabilidades *fuzzy* são calculadas pelo somatório de pertinências *fuzzy* e seus produtos:

$\mathbf{P}(S_0) = N(S_0) / N$

$\mathbf{P}(S_{t+1}|S_t) = N(S_{t+1},S_t) / N(S_t)$

$\mathbf{P}(E_{t,j}|S_t) = N(E_{t,j},S_t) / N(S_t)$, $1 \le j \le m$

com

$N(S_t) = (\sum_{v_t} m_{S_t}(v_t))$

$N(S_{t+1},S_t) = (\sum_{(v_{t+1},v_t)} m_{S_{t+1}}(v_{t+1}) \times m_{S_t}(v_t))$

$N(E_{t,j},S_t) = (\sum_{(a_{t,j},v_t)} m_{E_{t,j}}(a_{t,j}) \times m_{S_t}(v_t))$, $1 \le j \le m$

onde N é o número total de exemplos para treinamento, $\sum_{v_t}$ é o somatório para cada valor contínuo v_t proveniente do conjunto de treinamento, $\sum_{(a_{t,j},v_t)}$ é o somatório para cada par de valores contínuos $(a_{t,j},v_t)$, e $\sum_{(v_{t+1},v_t)}$ é o somatório para cada par de valores contínuos (v_{t+1},v_t). M exemplos virtuais podem ser adicionados ao conjunto de treinamento para evitar uma contagem nula:

$\mathbf{P}(S_0) = (N(S_0) + M.Q) / (N + M)$

$\mathbf{P}(S_{t+1}|S_t) = (N(S_{t+1},S_t) + M.Q.Q) / (N(S_t) + M.Q)$

$\mathbf{P}(E_{t,j}|S_t) = (N(E_{t,j},S_t) + M.Q.Q_j) / (N(S_t)+ M.Q)$, $1 \le j \le m$

onde $Q = 1/K$ se S_t tem K valores (regiões *fuzzy*) possíveis, e $Q_j = 1/K_j$ se $E_{t,j}$ tem K_j valores possíveis.

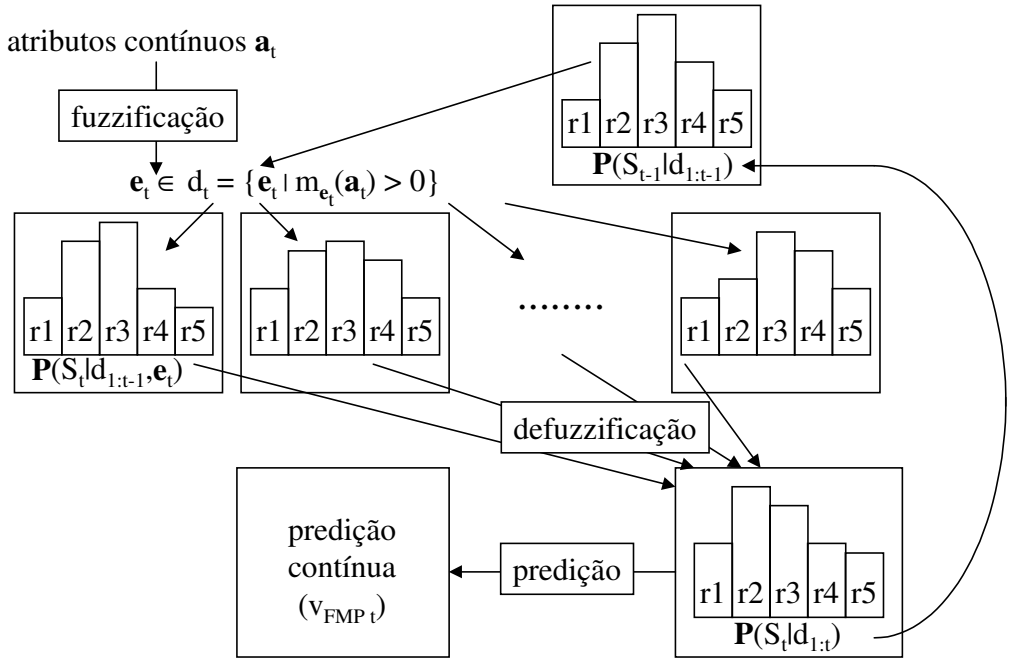


Figura 18: FMP - previsão de um valor contínuo.

A Figura 18 apresenta a previsão de um valor contínuo no FMP. Primeiro ocorre a fuzzificação da tupla de atributos contínuos $\mathbf{a}_t$ produzindo um conjunto de tuplas de atributos *fuzzy* d_t. A filtragem no FMP faz uso da distribuição a priori $\mathbf{P}(S_{t-1}|d_{1:t-1})$ para produzir várias distribuições $\mathbf{P}(S_t|d_{1:t-1},\mathbf{e}_t)$ condicionadas a cada tupla de atributos *fuzzy* $\mathbf{e}_t \in d_t$. Por defuzzificação, essas distribuições são integradas em uma única distribuição $\mathbf{P}(S_t|d_{1:t})$. Essa filtragem é exatamente igual a do FHMP, mas $\mathbf{P}(E_{t+1,j}|d_{1:t})$ não precisa ser calculada. Após a filtragem, a previsão contínua é imediatamente computada pela seguinte fórmula:

$v_{FMP\,t} = \sum_{s_t} \overline{s_t}.P(s_t|d_{1:t})$

onde $\overline{s_t}$ é o centro de gravidade da região *fuzzy* s_t, ou seja, $\overline{s_t} = \int v.m_{s_t}(v)dv \,/\, \int m_{s_t}(v)dv$.

A função densidade de probabilidade sendo aproximada pelo FMP é igual àquela estimada pelo MMR, ou seja, a densidade da filtragem da variável de estado:

se $t = 0$ então $f(v_t|\mathbf{a}_{1:t}) = f(v_t)$

se $t > 0$ então $f(v_t|\mathbf{a}_{1:t}) = \beta.f(\mathbf{a}_t|v_t).f(v_t|\mathbf{a}_{1:t-1}) = \beta.\{\prod_j f(a_{t,j}|v_t)\}.f(v_t|\mathbf{a}_{1:t-1})$

com a constante de normalização $\beta = 1 / \int\{f(\mathbf{a}_t|v_t).f(v_t|\mathbf{a}_{1:t-1})\}dv_t$.

Essa densidade é aproximada pela seguinte fórmula de defuzzificação:

$f(v_t|\mathbf{a}_{1:t}) \approx (\sum_{s_t} m_{s_t}(v_t).f_{s_t}(v_t|\mathbf{a}_{1:t})) / (\sum_{s_t} m_{s_t}(v_t))$

que é simplificada para (em um sistema de informação *fuzzy* ortogonal):

$f(v_t|\mathbf{a}_{1:t}) \approx \sum_{s_t} m_{s_t}(v_t).f_{s_t}(v_t|\mathbf{a}_{1:t})$

e cada $f_{s_t}(v_t|\mathbf{a}_{1:t})$ sendo combinado é dado por:

$f_{s_t}(v_t|\mathbf{a}_{1:t}) = P(s_t|d_{1:t}) / h_s$, para $m_{s_t}(v_t) > 0$

onde $P(s_t|d_{1:t})$ é a probabilidade da filtragem do estado *fuzzy* s_t, e h_s é a área da região *fuzzy* s_t ($h_s = \int m_{s_t}(v_t)dv_t$). Através de uma escolha adequada das funções de pertinência, a fuzzificação pode corresponder a uma discretização dos dados contínuos e, assim, a aproximação de $f(v_t|\mathbf{a}_{1:t})$ feita pelo FMP torna-se igual àquela do MMR.

A previsão do FMP ($v_{FMP\ t}$) é o valor esperado da variável contínua V_t condicionada pelos valores contínuos $\mathbf{a}_{1:t}$:

$v_{FMP\ t} = \int v_t.f(v_t|\mathbf{a}_{1:t})dv_t \approx \sum_{s_t} \overline{s_t}\ .\ P(s_t|d_{1:t})$

que é o mesmo resultado da previsão contínua do FMP já vista. Detalhes sobre esta demonstração são apresentados no Apêndice A. Essa previsão e a do MMR são iguais quando as funções de pertinência são ajustadas para que a fuzzificação seja uma discretização dos dados contínuos.

5.2.4 - *Fuzzy Multi-Hidden Markov Predictor*

O *Fuzzy Multi-Hidden Markov Predictor* (FMHMP) [67] é uma versão mais geral do FHMP assumindo que existem várias variáveis de estado. O FMHMP também pode ser visto como uma generalização do MHMMR na qual a fuzzificação substitui a discretização. Dessa forma, o espaço das observações contínuas passa por um processo de fuzzificação: para cada observação contínua ($a_{t,j}$) há um conjunto de valores *fuzzy* correspondente $\{e_{t,j} \mid m_{e_{t,j}}(a_{t,j}) > 0\}$. O FMHMP possui a mesma topologia que o MHMMR, só que agora $S_{t,i}$ ($1 \le i \le w$) e $E_{t,j}$ ($1 \le j \le m$) são variáveis aleatórias *fuzzy*.

A estimação das probabilidades *fuzzy* $\mathbf{P}(\mathbf{S}_{t+1}|\mathbf{S}_t)$ e $\mathbf{P}(E_{t,j}|\mathbf{S}_t)$ no FMHMP é efetuada pelo algoritmo EM da mesma forma que no FHMP, só que agora é considerado um conjunto de variáveis de estado $\mathbf{S}_t = (S_{t,1}, S_{t,2}, ..., S_{t,w})$:

$\mathbf{P}(\mathbf{S}_0) = N(\mathbf{S}_0) / N$

$\mathbf{P}(\mathbf{S}_{t+1}|\mathbf{S}_t) = N(\mathbf{S}_{t+1},\mathbf{S}_t) / N(\mathbf{S}_t)$

$\mathbf{P}(E_{t,j}|\mathbf{S}_t) = N(E_{t,j},\mathbf{S}_t) / N(\mathbf{S}_t)$, $1 \leq j \leq m$

com

$N(\mathbf{S}_t) = (\Sigma^T_{t=1}\ \mathbf{P}(\mathbf{S}_t|d_{1:t}))$

$N(\mathbf{S}_{t+1},\mathbf{S}_t) = (\Sigma^T_{t=1}\ \mathbf{P}(\mathbf{S}_{t+1},\mathbf{S}_t|d_{1:t}))$

$N(E_{t,j},\mathbf{S}_t) = (\Sigma^T_{t=1}\ \mathbf{P}(E_{t,j},\mathbf{S}_t|d_{1:t}))$, $1 \leq j \leq m$

onde $\mathbf{P}(\mathbf{S}_t|d_{1:t})$ é a filtragem do conjunto das variáveis de estado *fuzzy*, e as probabilidades $\mathbf{P}(\mathbf{S}_{t+1},\mathbf{S}_t|d_{1:t})$ e $\mathbf{P}(E_{t,j},\mathbf{S}_t|d_{1:t})$ são inferidas por:

$\mathbf{P}(\mathbf{S}_{t+1},\mathbf{S}_t|d_{1:t}) = \mathbf{P}(\mathbf{S}_{t+1}|\mathbf{S}_t).\mathbf{P}(\mathbf{S}_t|d_{1:t})$

$\mathbf{P}(E_{t,j},\mathbf{S}_t|d_{1:t}) = m_{E_{t,j}}(a_{t,j}).\mathbf{P}(\mathbf{S}_t|d_{1:t})$

onde $\mathbf{P}(\mathbf{S}_t|d_{1:t})$, $\mathbf{P}(\mathbf{S}_{t+1},\mathbf{S}_t|d_{1:t})$ e $\mathbf{P}(E_{t,j},\mathbf{S}_t|d_{1:t})$ usam as probabilidades $\mathbf{P}(\mathbf{S}_{t+1}|\mathbf{S}_t)$ e $\mathbf{P}(\mathbf{E}_t|\mathbf{S}_t)$ obtidas da iteração anterior.

A filtragem é definida a seguir:

$\mathbf{P}(\mathbf{S}_t|d_{1:t}) = (\sum_{\mathbf{e}_t \in d_t} m_{\mathbf{e}_t}(\mathbf{a}_t).\mathbf{P}(\mathbf{S}_t|d_{1:t-1},\mathbf{e}_t)) / (\sum_{\mathbf{e}_t \in d_t} m_{\mathbf{e}}(\mathbf{a}_t))$

ou (para um sistema de informação *fuzzy* ortogonal)

$\mathbf{P}(\mathbf{S}_t|d_{1:t}) = \sum_{\mathbf{e}_t \in d_t} m_{\mathbf{e}_t}(\mathbf{a}_t).\mathbf{P}(\mathbf{S}_t|d_{1:t-1},\mathbf{e}_t)$

com

$\mathbf{P}(\mathbf{S}_t|d_{1:t-1},\mathbf{e}_t) = \alpha.P(\mathbf{e}_t \mid \mathbf{S}_t).(\Sigma_{s_{t-1}}\ \mathbf{P}(\mathbf{S}_t|\mathbf{s}_{t-1}).P(\mathbf{s}_{t-1}|d_{1:t-1})) =$

$= \alpha.\{\prod_j P(e_{t,j}|\mathbf{S}_t)\}.(\Sigma_{s_{t-1}}\ \mathbf{P}(\mathbf{S}_t|\mathbf{s}_{t-1}).P(\mathbf{s}_{t-1}|d_{1:t-1}))$

O processo para previsão de um valor contínuo no FMHMP (Figura 19) é igual ao do FHMP com exceção do uso de um conjunto de variáveis de estado. O primeiro passo consiste na fuzzificação da observação contínua $\mathbf{a}_t$ gerando sua correspondente *fuzzy* $d_t = \{\mathbf{e}_t \mid m_{\mathbf{e}_t}(\mathbf{a}_t) > 0\}$. Na filtragem, partindo-se da distribuição a priori $\mathbf{P}(\mathbf{S}_{t-1}|d_{1:t-1})$ várias distribuições $\mathbf{P}(\mathbf{S}_t|d_{1:t-1},\mathbf{e}_t)$ são geradas de forma a incorporar a nova observação *fuzzy* $\mathbf{e}_t \in d_t$. Por defuzzificação, essas distribuições são integradas em uma única distribuição $\mathbf{P}(\mathbf{S}_t|d_{1:t})$. A distribuição $\mathbf{P}(E_{t+1,j}|d_{1:t})$ é calculada a partir de $\mathbf{P}(\mathbf{S}_t|d_{1:t})$ e a previsão contínua é dada por:

$a_{FMHMP\ t,j} = \sum_{e_{t+1,j}}\ \overline{e}_{t+1,j}.P(e_{t+1,j}|d_{1:t})$

com

$\mathbf{P}(E_{t+1,j}|d_{1:t}) = \Sigma_{\mathbf{s}_{t+1}}\ \mathbf{P}(E_{t+1,j}\ ,\mathbf{s}_{t+1}|d_{1:t}) = \Sigma_{\mathbf{s}_{t+1}}\ \mathbf{P}(E_{t+1,j}|\mathbf{S}_{t+1}).(\Sigma_{\mathbf{s}_t}\ \mathbf{P}(\mathbf{S}_{t+1}|\mathbf{s}_t).P(\mathbf{s}_t|d_{1:t}))$

onde j ($1 \leq j \leq m$) é uma componente selecionada da observação e $\overline{e}_{t+1,j}$ é o centro de gravidade da região *fuzzy* $e_{t+1,j}$. Todas essas distribuições *fuzzy* são representadas por histogramas:

- r1, r2, ..., rk são os possíveis valores (regiões *fuzzy*) para a variável aleatória *fuzzy* $E_{t,j}$, enquanto que as colunas são os valores das probabilidades para r1, r2, ..., rk;
- **r1**, **r2**, ..., **rn** são os possíveis conjuntos de valores (conjuntos de regiões *fuzzy*) para o conjunto de variáveis aleatórias *fuzzy* $\mathbf{S}_t$, enquanto que as colunas são os valores das probabilidades para **r1**, **r2**, ..., **rn**.

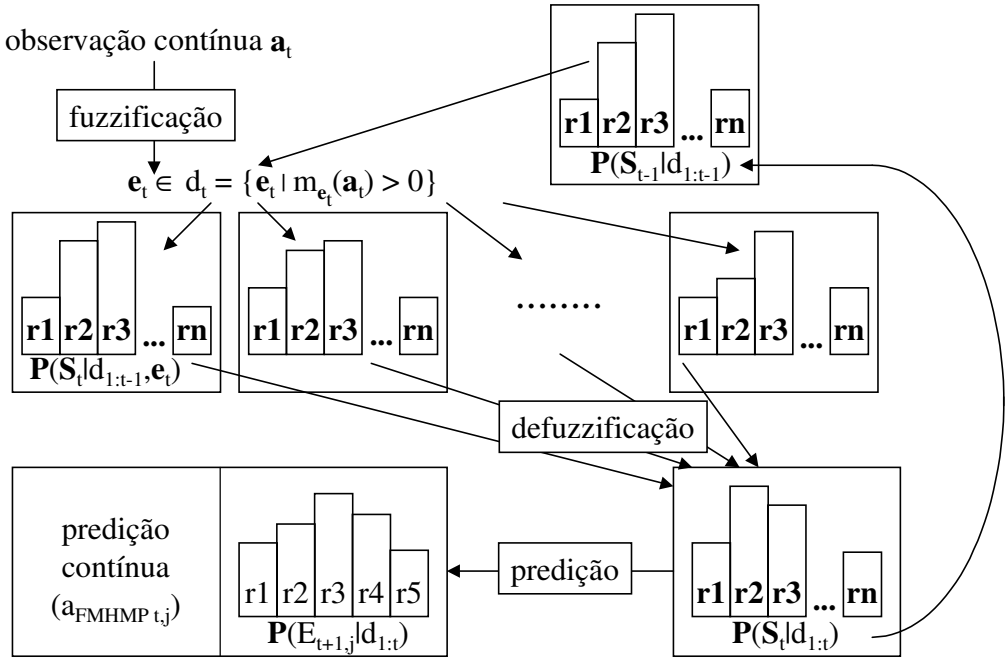


Figura 19: FMHMP - previsão de um valor contínuo.

O FMHMP está aproximando a mesma função densidade de probabilidade que o FHMP, ou seja, a densidade da previsão da variável de evidência:

$$f(a_{t+1,j} \mid \mathbf{a}_{1:t}) = \int \{f(a_{t+1,j} \mid \mathbf{v}_{t+1}).f(\mathbf{v}_{t+1} \mid \mathbf{a}_{1:t})\} d\mathbf{v}_{t+1}$$

e usa a mesma fórmula de defuzzificação para essa aproximação (considerando um sistema de informação *fuzzy* ortogonal):

$$f(a_{t+1,j}|\mathbf{a}_{1:t}) \approx \sum_{e_{t+1,j}} m_{e_{t+1,j}}(a_{t+1,j}).f_{e_{t+1,j}}(a_{t+1,j}|\mathbf{a}_{1:t})$$

com

$$f_{e_{t+1,j}}(a_{t+1,j}|\mathbf{a}_{1:t}) = P(e_{t+1,j}|d_{1:t}) / h_{e_j}, \text{ para } m_{e_{t+1,j}}(a_{t+1,j}) > 0.$$

Através dessa aproximação pode-se provar que a previsão do FMHMP ($a_{FMHMP\ t,j}$) dada pelo valor esperado da variável $A_{t+1,j}$ condicionada por $\mathbf{a}_{1:t}$ é igual à previsão contínua do FMHMP já apresentada:

$$a_{FMHMP\ t,j} = \int a_{t+1,j}.f(a_{t+1,j}|\mathbf{a}_{1:t})da_{t+1,j} \approx \sum_{e_{t+1,j}} \overline{e}_{t+1,j} \ . \ P(e_{t+1,j}|d_{1:t})$$

Detalhes sobre esta demonstração são apresentados no Apêndice A. Tanto essa previsão quanto a aproximação de $f(a_{t+1,j}|\mathbf{a}_{1:t})$ realizadas pelo FMHMP tornam-se idênticas àquelas do MHMMR quando as funções de pertinência são escolhidas de forma que a fuzzificação se iguale a uma discretização.

6 - Métodos para o Particionamento do Espaço de Dados Contínuos

Este capítulo descreve vários métodos para efetuar o particionamento do espaço de dados contínuos. São discutidos tanto métodos para discretização como para fuzzificação.

6.1 - Métodos de Particionamento na Obtenção de Intervalos e Regiões *Fuzzy*

Tanto os PPD's quanto os PPF's partem do princípio de que o espaço dos dados contínuos foi previamente particionado em intervalos ou regiões *fuzzy*. Em ambos os casos, a abordagem mais simples é efetuar um particionamento uniforme do espaço contínuo: na discretização, o espaço é dividido em intervalos de mesmo tamanho; na fuzzificação, regiões *fuzzy* com o mesmo formato são distribuídas uniformemente no espaço (Figura 7). *Density Trees* (DT's) [70], *K-Means* (KM) [16], [41] e suas respectivas versões *fuzzy* [66], [68] são métodos de particionamento que não possuem essa restrição de uniformidade.

6.2 - Particionamento Discreto e *Fuzzy* através de *Density Trees*

DT's fazem um particionamento recursivo do espaço contínuo com o propósito de estimar (de forma não-paramétrica) a função densidade de probabilidade "f" de uma variável aleatória contínua. Em [70], uma DT é utilizada para dividir (discretizar) o espaço contínuo em intervalos da seguinte maneira:

1. comece com N valores (exemplos) contínuos em um intervalo (o nó raiz) que cobre o domínio inteiro de "f";
2. divida o intervalo corrente em dois intervalos (filhos) de mesmo tamanho se o corrente possui ao menos $\sqrt{N}$ valores e sua distância ao nó raiz não excede

$\lfloor(\log_2 N)/4\rfloor$ (a distância entre um nó/intervalo e seus nós/intervalos filhos é igual a 1);

3. repita este processo para cada novo intervalo enquanto as condições forem satisfeitas.

A Figura 20 mostra um exemplo do particionamento feito por uma DT. Neste exemplo, os intervalos finais obtidos são [0.0; 0.25), [0.25; 0.5) e [0.5; 1.0].

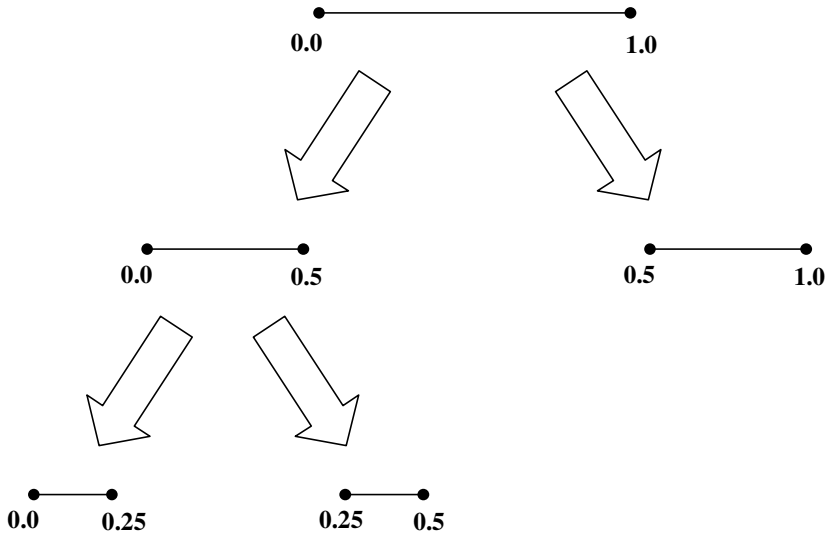


Figura 20: Particionamento feito por uma *Density Tree*.

A versão *fuzzy* desse algoritmo para a obtenção de uma DT funciona da mesma maneira que o original na divisão de intervalos. A única diferença é que são colocadas regiões *fuzzy* nesses intervalos:

1. o intervalo/nó raiz contém duas regiões *fuzzy* triangulares cujos máximos e mínimos (valores cujas pertinências são iguais a 1 e 0, respectivamente) são os limites do intervalo, e a interseção das regiões fuzzy é o centro do intervalo;
2. cada intervalo corrente contém parte de duas regiões *fuzzy* triangulares cujos máximos e mínimos são os limites do intervalo, e a interseção dessas regiões fuzzy é o centro do intervalo;
3. quando o intervalo corrente é dividido em dois intervalos de mesmo tamanho (se as condições do passo 2 da DT são satisfeitas), os mínimos das duas regiões *fuzzy* contidos no intervalo corrente são mudados para o centro desse intervalo, e uma nova região *fuzzy* triangular é inserida com seu máximo igual ao centro do intervalo corrente e mínimos iguais aos limites desse intervalo;

4. este processo é repetido para cada novo intervalo enquanto as condições forem satisfeitas.

A colocação de partes de regiões *fuzzy* dentro de um intervalo é similar àquela do *Binary Space Partitioning* (BSP) em sistemas *neuro-fuzzy* [54]. Um exemplo da criação de regiões *fuzzy* através desse método de particionamento é mostrado na Figura 21. Partindo de duas regiões *fuzzy*, r1 e r2, o algoritmo adiciona as novas regiões *fuzzy* r3, r4 e r5 nesta ordem.

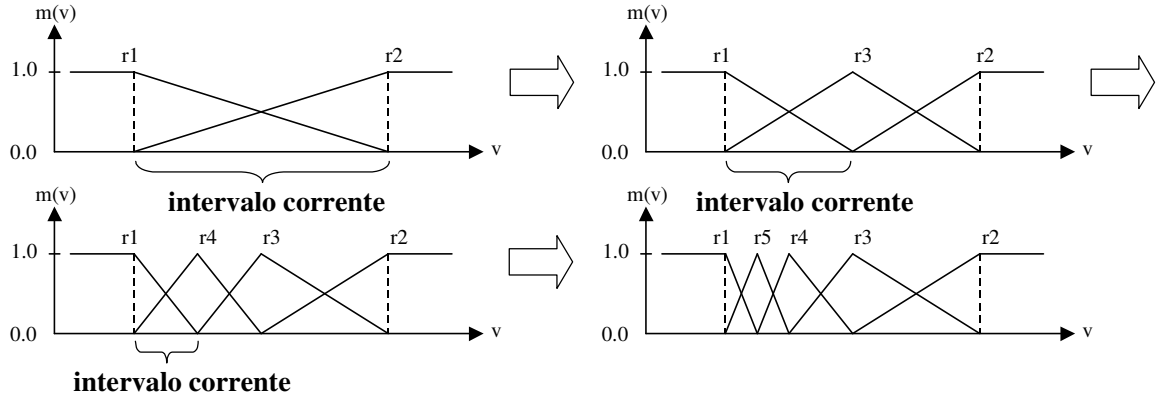


Figura 21: Particionamento feito pela versão *fuzzy* do algoritmo para DT.

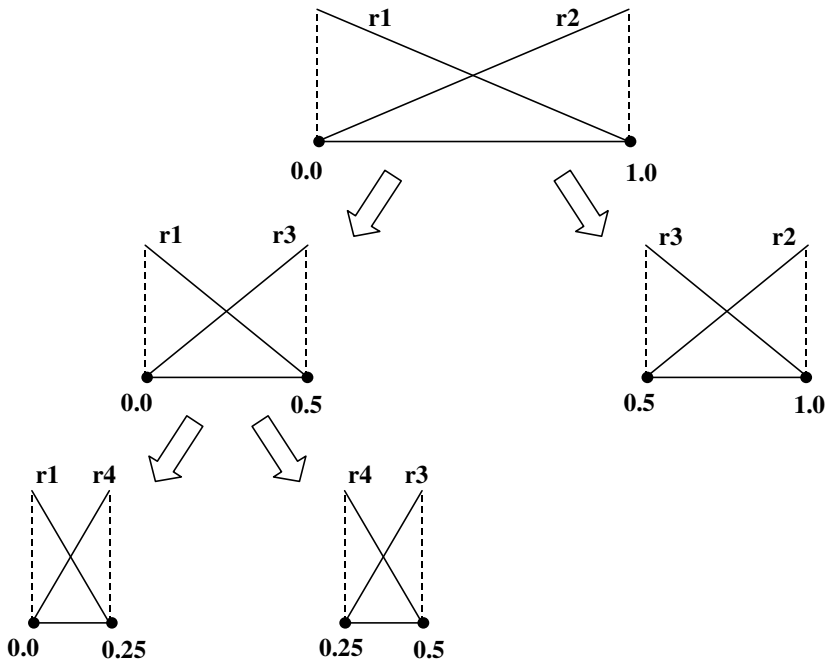


Figura 22: Particionamento *fuzzy* usando uma DT.

Na Figura 22 é visto o mesmo exemplo anteriormente apresentado para uma DT, mas agora usando sua versão *fuzzy*. As regiões *fuzzy* finais obtidas são r1, r2, r3 e r4.

6.3 - *K-Means* e sua Versão *Fuzzy*

O método de particionamento conhecido como *K-Means* (KM) tenta construir k intervalos de forma a minimizar a soma das distâncias $d_{i,j}$ ($d_{i,j} = |x_i - c_j|$) de cada valor (exemplo) contínuo x_i contido em um intervalo para o centro de gravidade c_j do intervalo:

1. inicialize o centro de gravidade c_j de cada intervalo j (por exemplo, pode-se escolher os intervalos de forma que cada um tenha a mesma quantidade de exemplos);
2. calcule (para i = 1, ..., N; j = 1, ..., k) $m_{i,j} = \{1$ se $d_{i,j} < d_{i,r}$ para todo $r \neq j$; 0 caso contrário$\}$, ou seja, a pertinência $m_{i,j}$ de um valor x_i em um intervalo j;
3. calcule o centro de gravidade c_j de cada intervalo j: $c_j = (\sum_{i=1}^{N} m_{i,j}.x_i) / (\sum_{i=1}^{N} m_{i,j})$;
4. repita os passos 2-3 até não haver alterações nos centros de gravidade (ou atingir um número máximo de iterações).

O ponto de interseção entre intervalos adjacentes j e j+1 é dado por $(c_j + c_{j+1}) / 2$. Supondo que a e b sejam os limites inferior e superior do domínio contínuo, respectivamente, os intervalos obtidos são $[a; (c_1 + c_2) / 2)$, $[(c_1 + c_2) / 2; (c_2 + c_3) / 2)$, ... e $[(c_{k-1} + c_k) / 2; b]$.

Uma outra possibilidade para inicialização dos centros de gravidade dos intervalos é usar o particionamento feito por uma DT. Se os k intervalos obtidos pela DT forem $[l_1; l_2)$, $[l_2; l_3)$, ... e $[l_k; l_{k+1}]$ então os centros de gravidade iniciais podem ser $c_1 = (l_1 + l_2) / 2$, $c_2 = (2.l_2 - c_1)$, $c_3 = (2.l_3 - c_2)$, ... e $c_k = (2.l_k - c_{k-1})$. Com esses centros consegue-se manter os mesmos pontos de interseção entre intervalos adjacentes produzidos pela DT pois $l_{j+1} = (c_j + c_{j+1}) / 2$.

A versão *fuzzy* do KM apresentada aqui utiliza k regiões *fuzzy* triangulares ao invés de intervalos. O máximo (valor cuja pertinência é igual a 1) de uma região *fuzzy* j é representado por c_j. Dado duas regiões *fuzzy* adjacentes j e j+1, um mínimo (valor cuja pertinência é igual a 0) da região *fuzzy* j é igual a c_{j+1}, um mínimo da região *fuzzy* j+1 é

igual a c_j, e o ponto com valor de pertinência igual a 0.5 (para ambas as regiões) é igual a $(c_j + c_{j+1}) / 2$. O algoritmo consiste nos seguintes passos:

1. inicialize o máximo c_j de cada região *fuzzy* j (por exemplo, pode-se escolher esses máximos de forma que sejam iguais aos centros de gravidade da inicialização do KM onde cada intervalo tem a mesma quantidade de exemplos);
2. calcule (para i = 1, ..., N; j = 1, ..., k) $m_{i,j} = m_j(x_i)$, onde $m_j(.)$ é a função de pertinência da região *fuzzy* j (que depende dos máximos de j e das regiões adjacentes a j);
3. calcule o máximo c_j de cada região *fuzzy* j: $c_j = (\sum_{i=1}^{N} m_{i,j}.x_i) / (\sum_{i=1}^{N} m_{i,j})$;
4. repita os passos 2-3 até não haver alterações nos máximos (ou atingir um número máximo de iterações).

Alguns aspectos devem ser ressaltados sobre esse algoritmo:

- embora não mencionado explicitamente, os máximos das regiões *fuzzy* cujos valores são os limites inferior e superior do domínio contínuo não devem ser alterados pelo algoritmo a fim de preservar a informação sobre os limites do domínio;
- de forma semelhante ao que acontece no KM ao utilizar um DT no passo de inicialização, o particionamento feito pela versão *fuzzy* do algoritmo para DT pode ser usado para inicializar os máximos das regiões *fuzzy*.

A versão *fuzzy* do algoritmo KM desenvolvida neste trabalho é diferente do *Fuzzy K-Means* (FKM) que aparece na literatura [2], [9]. No FKM (também conhecido como *Fuzzy C-Means*), a pertinência $m_{i,j}$ presente no algoritmo do KM é dada por

$$m_{i,j} = d_{i,j}^{2/(\phi-1)} / (\sum_{r=1}^{k} d_{i,r}^{2/(\phi-1)})$$

onde ϕ ($1 < \phi < \infty$) é o grau de sobreposição entre os k grupos de dados. Como o FKM não usa regiões *fuzzy* triangulares, ele não poderia utilizar a versão *fuzzy* do algoritmo para DT no passo de inicialização. Por isso optamos pelo uso de uma versão *fuzzy* do KM diferente do FKM.

7 - Aplicações

Este capítulo apresenta os resultados obtidos pela aplicação dos modelos desenvolvidos a casos reais, comparando-os com várias técnicas estatísticas.

7.1 - Aplicações: Predições *Single-Step* e *Multi-Step*

Existem duas maneiras de se fazer a previsão de uma série temporal: previsão *single-step* e *multi-step*. No primeiro caso sempre é feita a previsão de um valor da série localizado no instante imediatamente posterior ao instante do último valor da série sendo observado. Ou seja, estamos sempre fazendo a previsão 1-passo-à-frente do último valor da série observado (para se prever um valor z_t da série então todos o valores nos instantes anteriores a t devem ser conhecidos). Já no segundo caso, considera-se que a série é observável até um instante t específico. Se z_t é o último valor observado da série então a previsão 1-passo-à-frente obtém o valor não observado z_{t+1}, a previsão 2-passos-à-frente obtém o valor não observado z_{t+2} (sem o conhecimento do valor z_{t+1} da série), e assim por diante (dessa forma, fazemos a previsão em múltiplos passos).

Podem ser consideradas duas abordagens distintas para a previsão *multi-step* dependendo do preditor probabilístico utilizado. Na primeira abordagem, normalmente usada em redes neurais [5], [55], predições passadas são utilizadas como substitutas dos valores não observados da série. Por exemplo, considerando z_t como o último valor observado da série então a previsão 3-passos-à-frente para a obtenção de z_{t+3} deve empregar as previsões passadas de z_{t+1} (1-passo-à-frente) e de z_{t+2}. (2-passos-à-frente). Essa abordagem pode ser utilizada por todos os PPD's e PPF's. Já a segunda abordagem só pode ser empregada pelos preditores probabilísticos baseados em RBD's. Isso porque ela consiste na inferência de predição de uma RBD, ou seja, a computação de $\mathbf{P}(\mathbf{S}_{t+k}|\mathbf{e}_{1:t})$ para $k > 0$:

$$\mathbf{P}(\mathbf{S}_{t+k} \mid \mathbf{e}_{1:t}) = (\Sigma_{\mathbf{s}_{t+k-1}} \mathbf{P}(\mathbf{S}_{t+k}, \mathbf{s}_{t+k-1} \mid \mathbf{e}_{1:t})) \qquad \text{[marginalizando]}$$

$$= (\Sigma_{\mathbf{s}_{t+k-1}} \mathbf{P}(\mathbf{S}_{t+k} \mid \mathbf{s}_{t+k-1}, \mathbf{e}_{1:t}).\mathrm{P}(\mathbf{s}_{t+k-1} \mid \mathbf{e}_{1:t})) \qquad \text{[pelo teorema de Bayes]}$$

$$= (\Sigma_{\mathbf{s}_{t+k-1}} \mathbf{P}(\mathbf{S}_{t+k} \mid \mathbf{s}_{t+k-1}).\mathrm{P}(\mathbf{s}_{t+k-1} \mid \mathbf{e}_{1:t})) \qquad \text{[por (MA1)]}$$

e a seguir, supondo que $\mathbf{E}_t$ seja uma tupla de variáveis aleatórias, a inferência de $\mathbf{P}(E_{t+k,j}|\mathbf{e}_{1:t})$ a partir da distribuição $\mathbf{P}(\mathbf{S}_{t+k}|\mathbf{e}_{1:t})$:

$$\mathbf{P}(E_{t+k,j} \mid \mathbf{e}_{1:t}) = (\Sigma_{\mathbf{s}_{t+k}} \mathbf{P}(E_{t+k,j}, \mathbf{s}_{t+k} \mid \mathbf{e}_{1:t}))$$

onde

$$\mathbf{P}(E_{t+k,j}, \mathbf{S}_{t+k} \mid \mathbf{e}_{1:t}) = \mathbf{P}(E_{t+k,j} \mid \mathbf{S}_{t+k}, \mathbf{e}_{1:t}).\ \mathbf{P}(\mathbf{S}_{t+k} \mid \mathbf{e}_{1:t}) \quad \text{[pelo teorema de Bayes]}$$
$$= \mathbf{P}(E_{t+k,j} \mid \mathbf{S}_{t+k}).\ \mathbf{P}(\mathbf{S}_{t+k} \mid \mathbf{e}_{1:t}) \quad \text{[por (MA2)]}$$

Através de $\mathbf{P}(E_{t+k,j}|\mathbf{e}_{1:t})$ calcula-se a previsão k-passos-à-frente de um PPD baseado em RBD:

$$a_{PPD\ t,k,j} = \sum_{e_{t+k,j} \in\ dom(E_{t+k,j})} \{m(e_{t+k,j}).P(e_{t+k,j} \mid \mathbf{e}_{1:t})\}$$

onde j ($1 \leq j \leq m$) é uma componente selecionada da evidência, $m(e_{t+k,j})$ é o valor central do intervalo $e_{t+k,j}$. No caso da previsão k-passos-à-frente de um PPF baseado em RBD:

$$a_{PPF\ t,k,j} = \sum_{e_{t+k,j}} \overline{e}_{t+k,j}.P(e_{t+k,j}|d_{1:t})$$

onde $\overline{e}_{t+k,j}$ é o centro de gravidade da região *fuzzy* $e_{t+k,j}$. A inferência de predição de um PPF é dada por $\mathbf{P}(\mathbf{S}_{t+k}|d_{1:t})$ para $k > 0$:

$$\mathbf{P}(\mathbf{S}_{t+k} \mid d_{1:t}) = (\Sigma_{\mathbf{s}_{t+k-1}} \mathbf{P}(\mathbf{S}_{t+k}, \mathbf{s}_{t+k-1} \mid d_{1:t})) \quad \text{[marginalizando]}$$
$$= (\Sigma_{\mathbf{s}_{t+k-1}} \mathbf{P}(\mathbf{S}_{t+k} \mid \mathbf{s}_{t+k-1}, d_{1:t}).P(\mathbf{s}_{t+k-1} \mid d_{1:t})) \quad \text{[pelo teorema de Bayes]}$$
$$= (\Sigma_{\mathbf{s}_{t+k-1}} \mathbf{P}(\mathbf{S}_{t+k} \mid \mathbf{s}_{t+k-1}).P(\mathbf{s}_{t+k-1} \mid d_{1:t})) \quad \text{[por (FMA1)]}$$

e a distribuição $\mathbf{P}(E_{t+k,j}|d_{1:t})$ é obtida por:

$$\mathbf{P}(E_{t+k,j} \mid d_{1:t}) = (\Sigma_{\mathbf{s}_{t+k}} \mathbf{P}(E_{t+k,j}, \mathbf{s}_{t+k} \mid d_{1:t}))$$

onde

$$\mathbf{P}(E_{t+k,j}, \mathbf{S}_{t+k} \mid d_{1:t}) = \mathbf{P}(E_{t+k,j} \mid \mathbf{S}_{t+k}, d_{1:t}).\ \mathbf{P}(\mathbf{S}_{t+k} \mid d_{1:t}) \quad \text{[pelo teorema de Bayes]}$$
$$= \mathbf{P}(E_{t+k,j} \mid \mathbf{S}_{t+k}).\ \mathbf{P}(\mathbf{S}_{t+k} \mid d_{1:t}) \quad \text{[por (FMA2)]}$$

7.1.1 - Resultados Experimentais para Previsão *Single-Step*

Os PPD's e PPF's descritos nos capítulos anteriores foram aplicados à tarefa de previsão *single-step* de séries de carga elétrica mensal e foram comparados a dois modelos de filtro de Kalman, STAMP (abordagem clássica de Harvey) e BATS (abordagem Bayesiana de Harrison & Stevens), e dois métodos de previsão tradicionais, Box-Jenkins e amortecimento exponencial de Winters.

Foram empregadas três séries (Figuras 23, 24 e 25) de carga elétrica mensal (5×12 meses de dados para treinamento e 3×12 meses para teste). Essas séries foram obtidas

de empresas brasileiras de energia elétrica e apresentam um comportamento de mudança abrupta e significativa em seus últimos anos, como quando ocorre um racionamento de energia.

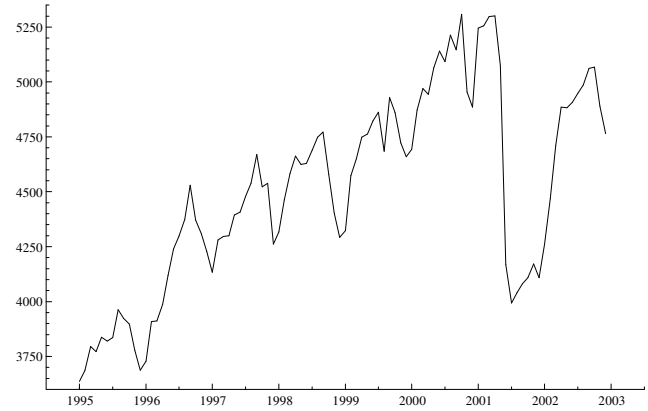


Figura 23: Série 1 de carga elétrica mensal.

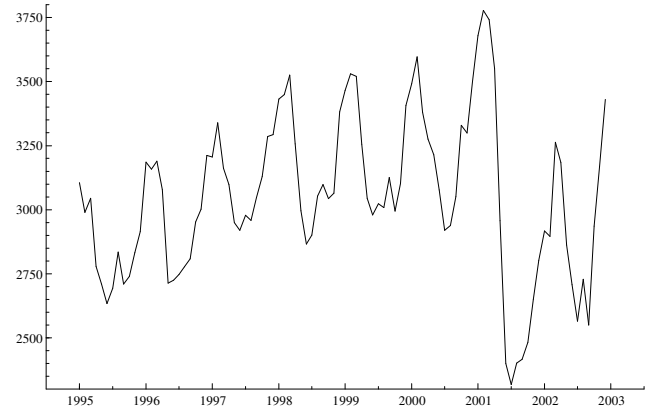


Figura 24: Série 2 de carga elétrica mensal.

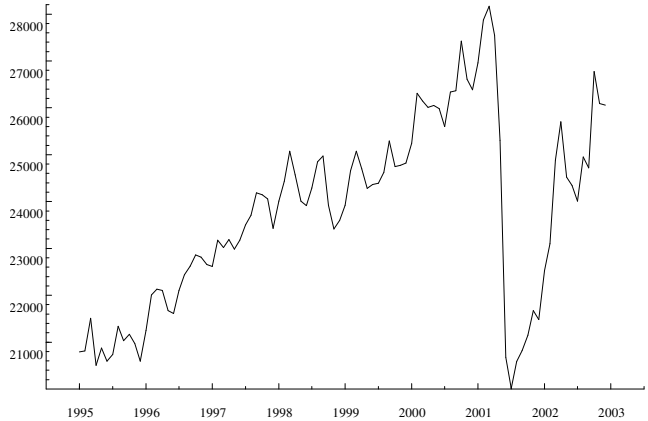


Figura 25: Série 3 de carga elétrica mensal.

A métrica de erro utilizada foi MAPE (*Mean Absolute Percentage Deviation*):

MAPE = ($\sum_{i=1}^{n} | e_i |$) / n

onde

e_i = *((desejado$_i$ - previsto$_i$)/desejado$_i$)* * 100%,

n = número de exemplos de teste.

Todos os sistemas (PPD's, PPF's, modelos de filtro de Kalman e métodos tradicionais) fizeram uso dos últimos 3 anos das séries como o conjunto de teste, e os 5 anos anteriores como o conjunto de treinamento para efetuar a previsão *single-step* do próximo mês que corresponde ao primeiro mês do conjunto de teste. Para se prever cada mês do conjunto de teste os sistemas são retreinados com os 5 anos que precedem o mês sendo previsto.

Quatro casos foram considerados para a distribuição dos intervalos ou regiões *fuzzy* (triangulares) no espaço contínuo de um PPD ou PPF: distribuição uniforme das partições (intervalos ou regiões *fuzzy*); partições distribuídas de acordo com o algoritmo para *Density Tree* (DT); partições obtidas pelo *K-Means* (KM); inicialização das partições através do algoritmo para DT seguido pelo ajuste das mesmas pelo KM. Os erros de previsão para cada uma dessas possibilidades são mostrados nas Tabelas 1, 2, 3, 4, 6, 7, 8, 9, 11, 12 13 e 14. Nessas tabelas, "F#HMP" ou "#HMMR" indicam um FMHMP ou um MHMMR com o número de variáveis de estado igual a "#". Já as Tabelas 5, 10 e 15 apresentam os erros de previsão para os dois modelos de filtro de Kalman e os dois métodos tradicionais. Nessas tabelas, cada coluna contém os erros (MAPE) para um ano específico do conjunto de teste, com exceção da última coluna que apresenta uma média dos erros para os três anos do conjunto de teste. Todas essas tabelas estão contidas no Apêndice B.

Forward validation (FV) [20] foi utilizado para a seleção do número de atributos (ou número de componentes da observação), onde $\mathbf{A} = (A_1, ..., A_m)$ ou $\mathbf{A}_t = (A_{t,1}, ..., A_{t,m})$ são os atributos (ou componentes da observação), e o número de partições nos PPD's e PPF's, exceto quando a escolha do número de partições é feita por um algoritmo para DT. Como os modelos de filtro de Kalman testados consideram cada observação constituída por uma única componente, também foram testados PPD's e PPF's com essa mesma característica (sem a necessidade do *forward validation* para a escolha do número de componentes da observação). Esses PPD's e PPF's que possuem apenas uma única variável servindo como entrada (ou uma única variável de evidência) são referenciados nas tabelas pelo sufixo "1e". As Tabelas 16, 17, 18 e 19 (Apêndice B)

apresentam os números de atributos e partições selecionados para os quatro casos de distribuição das partições de um PPD ou PPF no espaço contínuo.

M(p) representa um modelo de PPD ou PPF cujos parâmetros p são o número de atributos (componentes da observação) e o número de partições. O FV efetua a seguinte computação para cada possível modelo M(p):

- comece com r-1 exemplos de treinamento, como conjunto de validação o exemplo $desejado_r$ e compute Erro(p,r);
- inclua $desejado_r$ no conjunto de treinamento, faça o conjunto de validação conter apenas o exemplo $desejado_{r+1}$ e compute Erro(p,r+1);
- repita até que o conjunto de validação contenha apenas $desejado_N$ e compute Erro(p,N);

onde Erro(p,s) = | ($desejado_s$ - $previsto_s$) / $desejado_s$ |, N é o número de exemplos de treinamento original (5 × 12 meses), e r-1 é a menor quantidade de exemplos usada para treinamento (3 × 12 meses) durante a validação. O modelo M(p) escolhido será aquele que minimiza a expressão:

$$C(p) = \sum_{s=r}^{N} \gamma_s.Erro(p,s)$$

onde $\gamma_s = (1 / (1 + N_p / (s - 1))) / (\sum_{i=r}^{N} 1 / (1 + N_p / (i - 1)))$

e N_p é o número de parâmetros do modelo M(p) (neste caso, é o número de parâmetros de uma RB ou RBD com variáveis discretas [13] ou *fuzzy*).

As três séries foram diferenciadas para os PPD's e PPF's, ou seja, cada série temporal diferenciada é obtida por $Y_t = Z_t - Z_{t-1}$, onde Z_t corresponde a um valor de cada uma das séries originais. Para os PPD's e PPF's baseados em RBD's, os tempos t-1 e t representam o mesmo mês em anos consecutivos: isso é equivalente a fazer o treinamento com 12 séries distintas onde cada uma delas contém apenas os valores de um determinado mês (isto é, temos uma série para o mês de janeiro, outra para o de fevereiro e assim por diante). No caso das variáveis de estado não serem observadas no conjunto de treinamento, a estimativa inicial (para ser usada pelo algoritmo EM) das probabilidades $\mathbf{P}(\mathbf{S}_{t+1}|\mathbf{S}_t)$ e $\mathbf{P}(\mathbf{E}_t|\mathbf{S}_t)$ é dada pela suposição de que as variáveis de estado são observadas no conjunto de treinamento e iguais a $E_{t+1,j}$ no instante t, onde j é a componente da observação que se deseja prever. Para um HMMR ou FHMP, isso equivale a usar como estimativa inicial as probabilidades de um MMR ou FMP, respectivamente.

Para Box-Jenkins, a identificação da diferenciação usada no modelo ARIMA é feita por um algoritmo baseado no método *Augmented* Dickey-Fuller [14]. A identificação do modelo para os dados diferenciados é realizada pela minimização do BIC (*Bayesian Information Criterion*) sobre um número de modelos ARIMA estimados aproximadamente.

Nas tabelas de erros (previsão *single-step*) foram colocados em negrito os menores erros obtidos para cada série em cada uma das colunas (onde cada coluna representa os erros para um ano específico ou as médias dos erros para os três anos). Tanto para a série 1 quanto para a série 3, houve 3 colunas onde um PPD ou PPF foi o melhor enquanto que houve 1 coluna onde um modelo de filtro de Kalman ou método tradicional foi o melhor. Para a série 2, houve 4 colunas onde um PPD ou PPF foi o melhor enquanto que não houve nenhuma coluna onde um modelo de filtro de Kalman ou método tradicional foi o melhor.

Nas tabelas dos números de atributos e partições selecionados (previsão *single-step*), percebe-se que a utilização de DT's obteve sempre 3 ou 4 intervalos e 4 ou 5 regiões *fuzzy*. Também é interessante notar que houve situações onde o FV decidiu que era mais adequado usar um único atributo.

7.1.2 - Resultados Experimentais para Previsão *Multi-Step*

As séries de carga elétrica mensal apresentadas anteriormente para a tarefa de previsão *single-step* também foram utilizadas para a previsão *multi-step*. Apenas alguns dos sistemas citados na seção anterior foram selecionados para essa tarefa: PPD's e PPF's baseados em RBD's (com exceção do MMR e FMP), um modelo de filtro de Kalman (BATS), Box-Jenkins e amortecimento exponencial de Winters.

Todos os sistemas fizeram uso dos últimos 3 anos das séries como o conjunto de teste, e os 5 anos anteriores como o conjunto de treinamento para efetuar a previsão *multi-step* do conjunto de teste.

Dois casos foram considerados para a distribuição das partições (intervalos ou regiões *fuzzy* triangulares) no espaço contínuo de um PPD ou PPF: partições distribuídas de acordo com o algoritmo para DT; e inicialização das partições através do algoritmo para DT seguido pelo ajuste das mesmas pelo KM. Os erros (MAPE) de previsão para cada uma dessas possibilidades são mostrados nas Tabelas 20, 21, 23, 24, 26 e 27. Foram

usados apenas PPD's e PPF's que possuem uma única variável de evidência (por isso todos são identificados pelo sufixo "1e" nas tabelas). Como a escolha do número de partições é sempre feita por um algoritmo para DT e existe apenas uma única variável de evidência, o FV não é necessário. As Tabelas 22, 25 e 28 apresentam os erros de previsão para um modelo de filtro de Kalman e os dois métodos tradicionais. As Tabelas 29 e 30 apresentam os números de atributos (sempre 1) e partições selecionados. Todas essas tabelas estão contidas no Apêndice B.

As três séries foram diferenciadas para os PPD's e PPF's baseados em RBD's, e nesses modelos os tempos t-1 e t representam o mesmo mês em anos consecutivos. A estimativa inicial das probabilidades $\mathbf{P}(\mathbf{S}_{t+1}|\mathbf{S}_t)$ e $\mathbf{P}(\mathbf{E}_t|\mathbf{S}_t)$ é feita da mesma forma discutida na seção passada. Também são testados MHMMR's e FMHMP's nos quais são adicionados ruídos aleatórios a essas estimativas iniciais de probabilidades (eles são referenciados nas tabelas pelo prefixo "a") para averiguar se tal procedimento melhora ou não as previsões realizadas.

Para Box-Jenkins, a identificação da diferenciação usada no modelo ARIMA e a identificação do modelo para os dados diferenciados são realizadas exatamente como vistas na seção anterior.

Nas tabelas de erros (previsão *multi-step*) foram colocados em negrito os menores erros obtidos para cada série em cada uma das colunas. Tanto para a série 1 quanto para a série 2, houve 3 colunas onde um PPD ou PPF foi o melhor enquanto que houve 1 coluna onde um modelo de filtro de Kalman ou método tradicional foi o melhor. Para a série 3, houve 1 coluna onde um PPD ou PPF foi o melhor enquanto que houve 3 colunas onde um modelo de filtro de Kalman ou método tradicional foi o melhor.

Nas tabelas dos números de atributos e partições selecionados (previsão *multi-step*), percebe-se que a utilização de DT's obteve sempre 4 intervalos e 5 regiões *fuzzy*.

7.2 - Conclusões

De forma geral, os PPD's e PPF's obtiveram resultados competitivos na tarefa de previsão *single-step* quando comparados com os dois modelos de filtro de Kalman, STAMP e BATS, e os dois métodos tradicionais para previsão, Box-Jenkins e amortecimento exponencial de Winters. Isso é mais evidente para os PPD's e PPF's com variáveis de estado não observáveis no conjunto de treinamento. Para a previsão

multi-step foi considerado apenas um subconjunto dos modelos anteriores para análise, consistindo apenas de PPD's e PPF's com variáveis de estado não observáveis, e os resultados foram razoáveis. Nas duas seções seguintes são mostradas várias figuras resumindo as tabelas de erros anteriores. Em cada uma dessas figuras os erros são referentes às predições dos três anos do conjunto de teste (para as três séries) e sempre são comparados PPD's e PPF's com modelos de filtro de Kalman e métodos tradicionais.

7.2.1 - Conclusões sobre a Previsão *Single-Step*

As Figuras 26, 27, 28 e 29 apresentam os erros de previsão *single-step* para PPD's e PPF's com a escolha do número de entradas feita pelo FV. Comparando-se cada preditor que usa a abordagem uniforme com o mesmo preditor que usa DT (totalizando 8 preditores), a abordagem uniforme obteve 13 vitórias (de 24 possibilidades = 8 preditores x 3 séries), onde cada uma dessas vitórias representa a constatação de que um preditor específico usando a abordagem uniforme obteve um erro de previsão menor que o mesmo preditor usando DT para alguma série. Nessa mesma comparação, a abordagem por DT obteve 7 vitórias e ocorreram 4 empates. A princípio isso pode sugerir que a bordagem uniforme seja mais adequada que a de DT, mas também é preciso considerar que o uso de DT é computacionalmente mais rápido que o processo de escolha do número de partições uniformes. Por exemplo, o tempo de computação para o uso de PPD's e PPF's com três variáveis de estado para a abordagem uniforme é extremamente maior do que a abordagem por DT (pois a escolha do número de partições feita por DT não realiza previsões nos dados de treinamento como ocorre na escolha do número de partições uniformes feitas com o FV).

Já na comparação da abordagem uniforme com a que usa KM (totalizando 8 preditores) evidencia-se os seguintes resultados: para a uniforme 12 vitórias e a do KM 12 vitórias, o que nos leva a crer que o uso de KM não necessariamente melhora nossas predições. Comparando-se a abordagem que usa DT com aquela que combina DT e KM (totalizando 10 preditores), ocorrem 9 vitórias para a de DT e 21 vitórias para a de DT-KM (de 30 possibilidades = 10 preditores x 3 séries), o que sugere que o KM pode melhorar o particionamento feito pela DT. Comparando-se DT-KM com uniforme temos 14 vitórias para o primeiro e 10 vitórias para o segundo. Considerando DT-KM

versus KM (total de 8 preditores) temos 14 vitórias para o primeiro, 9 vitórias para o segundo e 1 empate. Pode-se concluir que o uso conjunto de DT com KM é uma abordagem eficiente e rápida em PPD's e PPF's de estruturas complexas: por exemplo, o F3HMP e o 3HMMR obtiveram resultados satisfatórios quando comparados aos modelos de filtro de Kalman e métodos tradicionais.

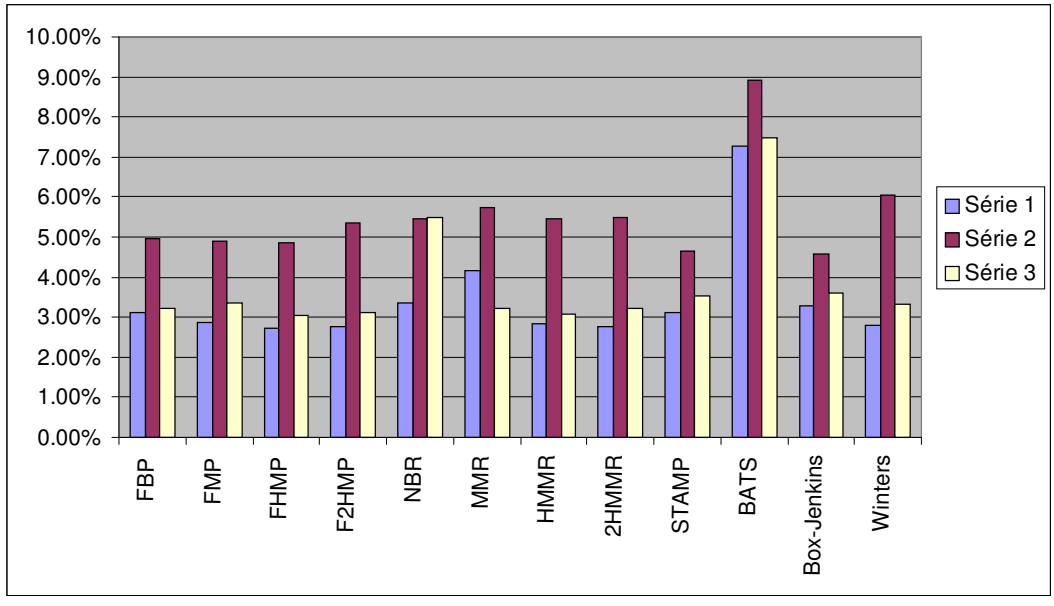


Figura 26: MAPE de PPD's e PPF's com distribuição uniforme das partições.

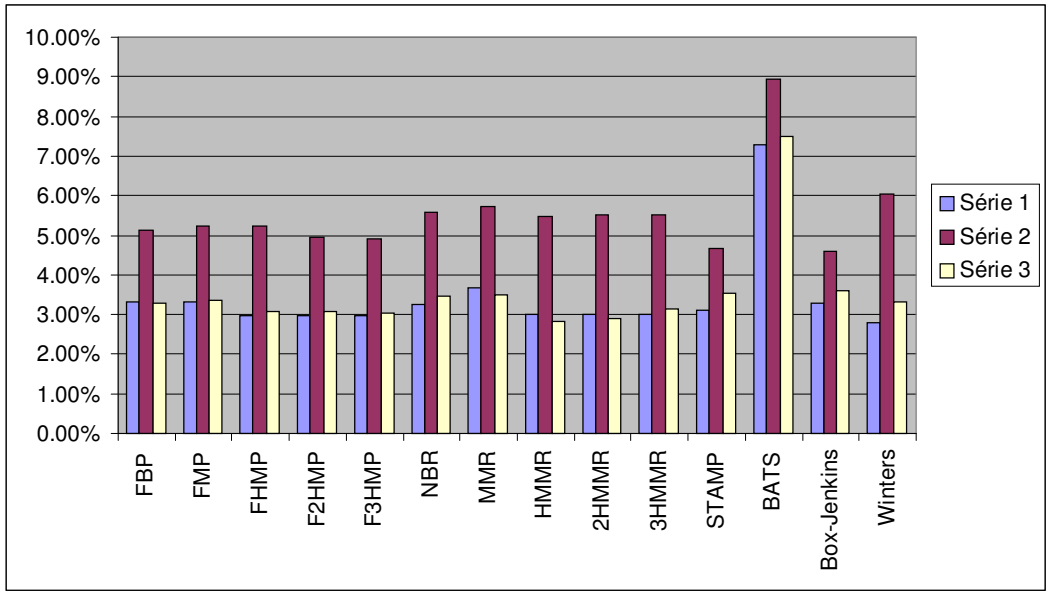


Figura 27: MAPE de PPD's e PPF's com partições distribuídas através de DT.

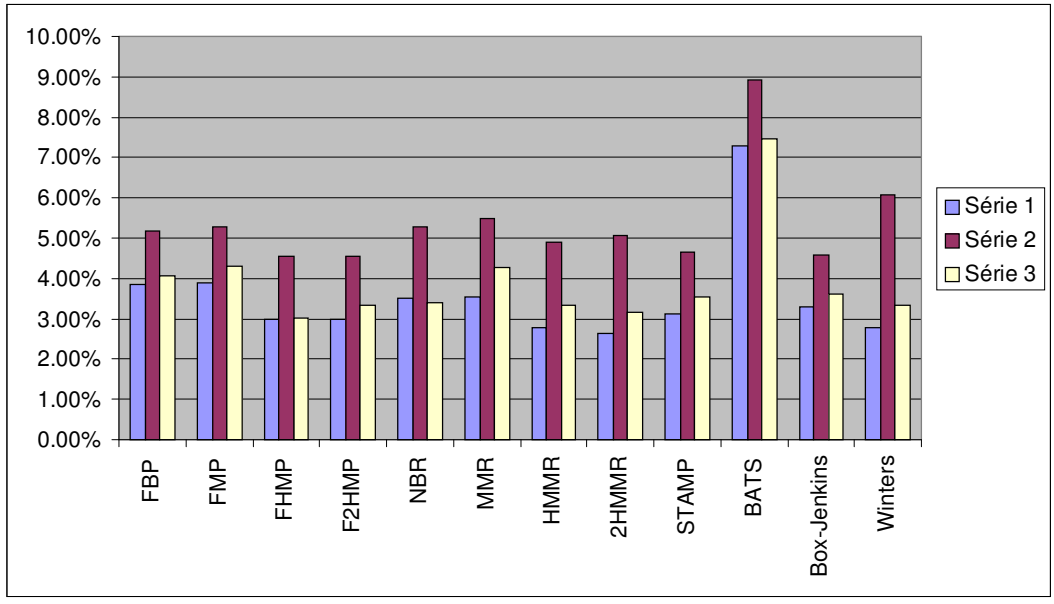


Figura 28: MAPE de PPD's e PPF's com partições obtidas pelo KM.

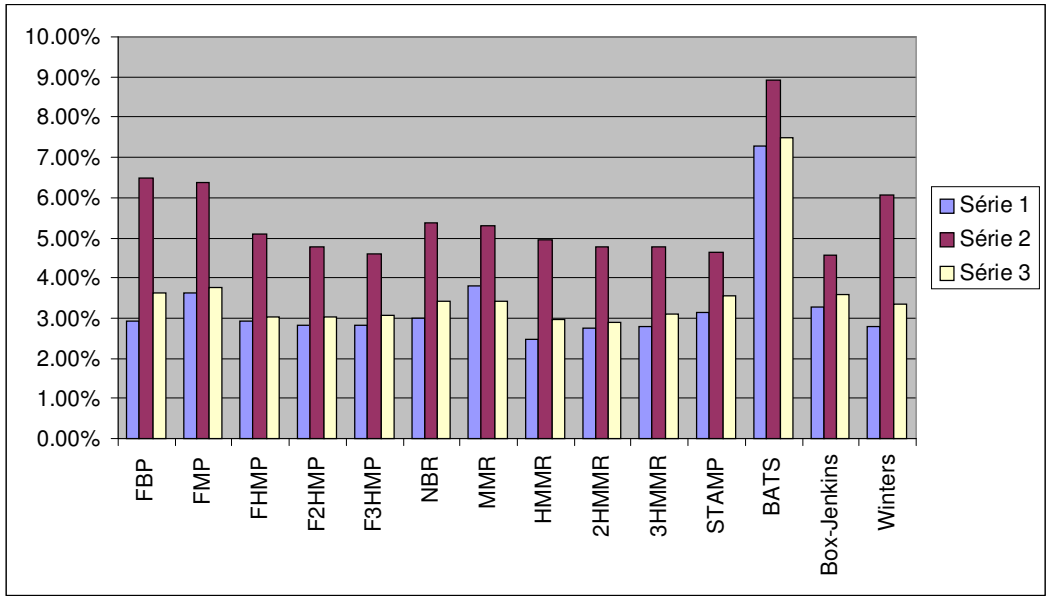


Figura 29: MAPE de PPD's e PPF's com partições obtidas por DT e KM.

As Figuras 30, 31, 32 e 33 apresentam os erros de previsão *single-step* para PPD's e PPF's com apenas uma única entrada. Aqui as comparações entre abordagens para particionamento mostram resultados similares ao caso anterior:

- uniforme (22 vitórias) versus DT (4 vitórias e 1 empate);
- uniforme (16 vitórias) versus KM (14 vitórias);
- DT (4 vitórias) versus DT-KM (26 vitórias);
- DT-KM (17 vitórias) versus uniforme (12 vitórias e 1 empate);

- DT-KM (22 vitórias) versus KM (8 vitórias).

Mesmo com a restrição da entrada única, PPD's e PPF's de estruturas complexas usando DT-KM se mostraram promissores: por exemplo, o F3HMP1e e o 3HMMR1e conseguiram resultados razoáveis se comparados aos modelos de filtro de Kalman e métodos tradicionais.

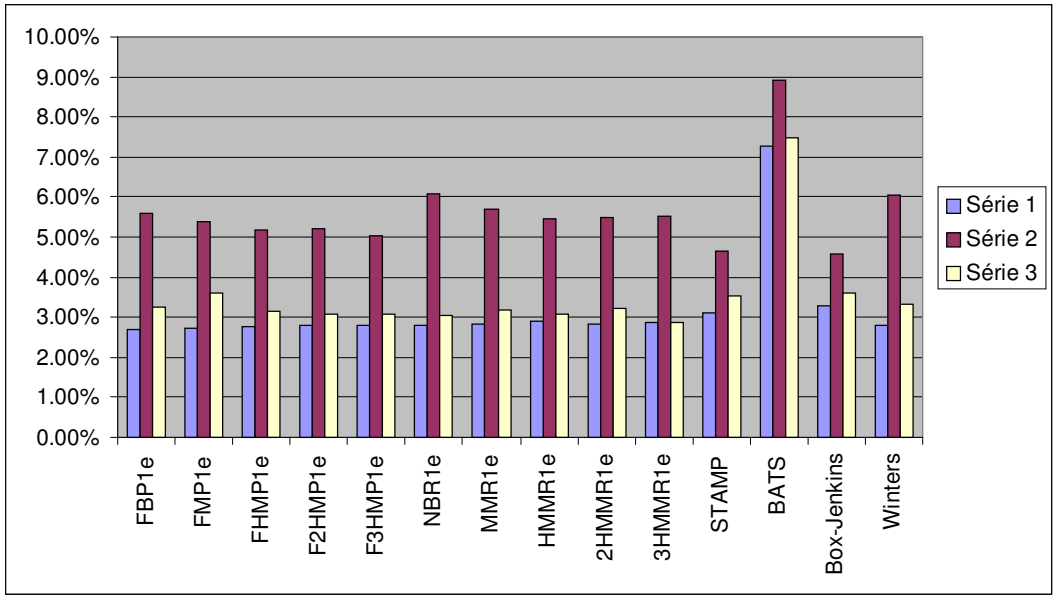


Figura 30: MAPE de PPD's e PPF's com 1 entrada e partições uniformes.

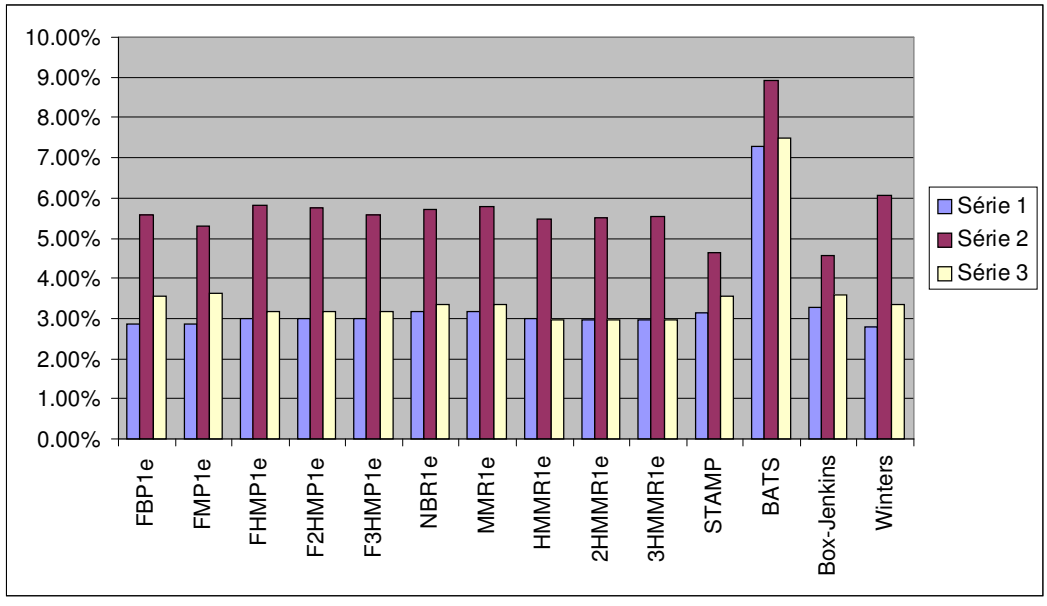


Figura 31: MAPE de PPD's e PPF's com 1 entrada e partições obtidas por DT.

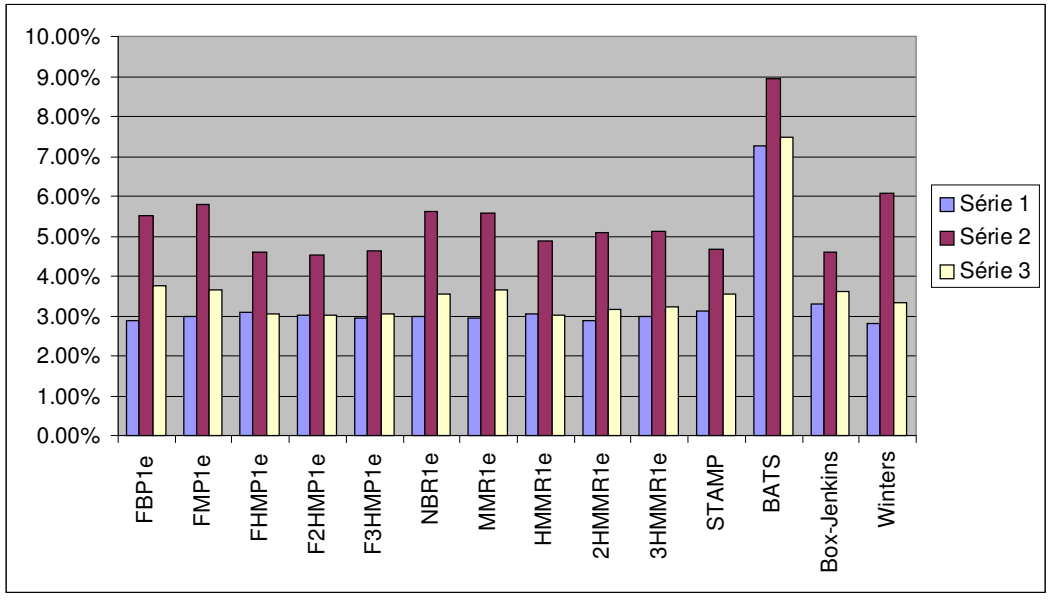


Figura 32: MAPE de PPD's e PPF's com 1 entrada e partições obtidas pelo KM.

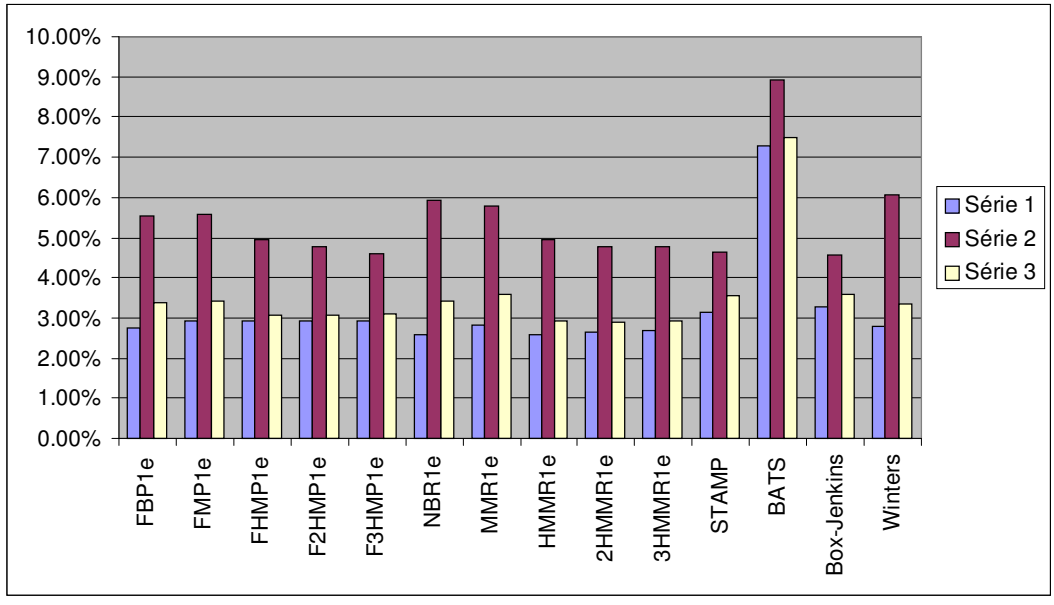


Figura 33: MAPE de PPD's e PPF's com 1 entrada e partições obtidas por DT e KM.

7.2.2 - Conclusões sobre a Previsão *Multi-Step*

As Figuras 34 e 35 apresentam os erros de previsão *multi-step* para PPD's e PPF's considerando-se os três anos do conjunto de teste. Foram examinados apenas preditores probabilísticos com variáveis de estado não observáveis, uma única entrada e a escolha do número de partições feita através de DT's. Essas escolhas são fundamentadas no desejo de otimizar ao máximo a velocidade de execução para os PPD's e PPF's de

estruturas complexas. Comparando-se o uso ou não do KM nesses preditores, foram obtidos os seguintes resultados: 16 vitórias para DT-KM e 14 vitórias para DT.

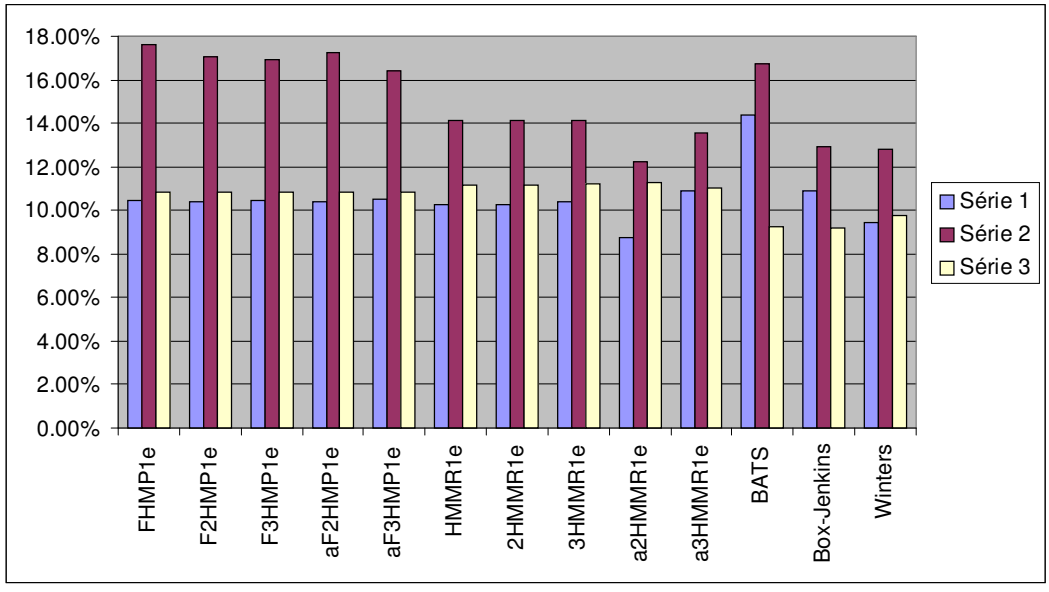


Figura 34: MAPE de PPD's e PPF's com 1 entrada e partições obtidas por DT.

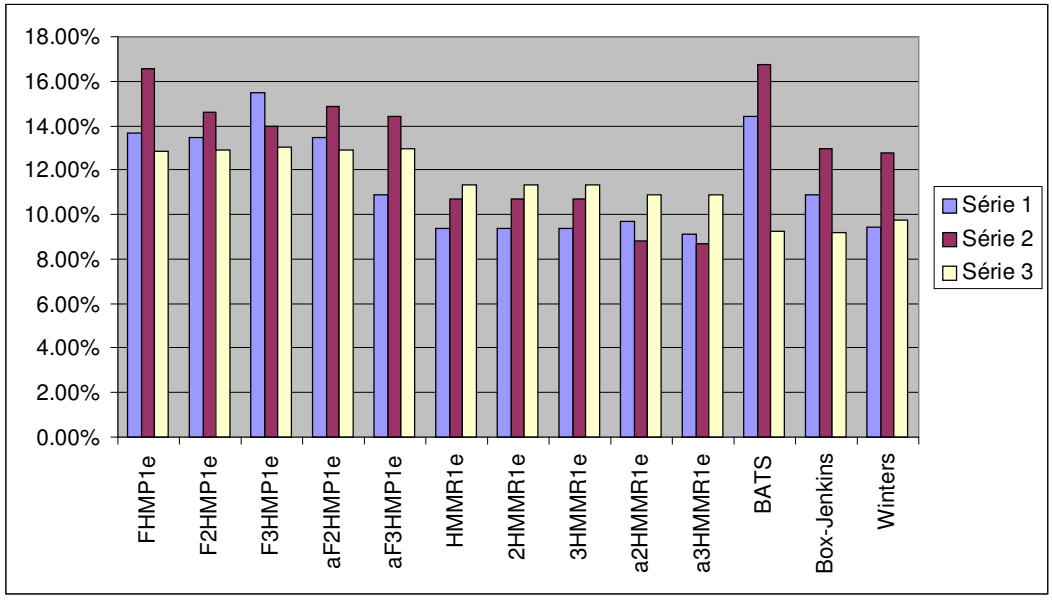


Figura 35: MAPE de PPD's e PPF's com 1 entrada e partições obtidas por DT e KM.

O problema aqui é que no segundo ano dos conjuntos de teste ocorre uma mudança brusca no comportamento das séries que não é incorporada pelos modelos. Como os erros dos dois últimos anos não auxiliam muito na comparação dos modelos, as Figuras 36 e 37 apresentam os erros de previsão *multi-step* para PPD's e PPF's considerando-se apenas o primeiro ano do conjunto de teste. Para esse conjunto mais restrito de teste

temos 26 vitórias para DT-KM e 4 vitórias para DT, que é similar aos resultados obtidos na previsão *single-step*. Quanto à aplicação de ruídos aleatórios nas estimativas iniciais de probabilidades dos MHMMR's e FMHMP's, ela conseguiu diminuir os erros de previsão em apenas metade dos casos (na outra metade houve um aumento dos erros). PPD's e PPF's de estruturas complexas com entrada única usando DT-KM conseguiram resultados razoáveis (principalmente os preditores *fuzzy*) se comparados aos outros métodos (embora não tanto para a série 2).

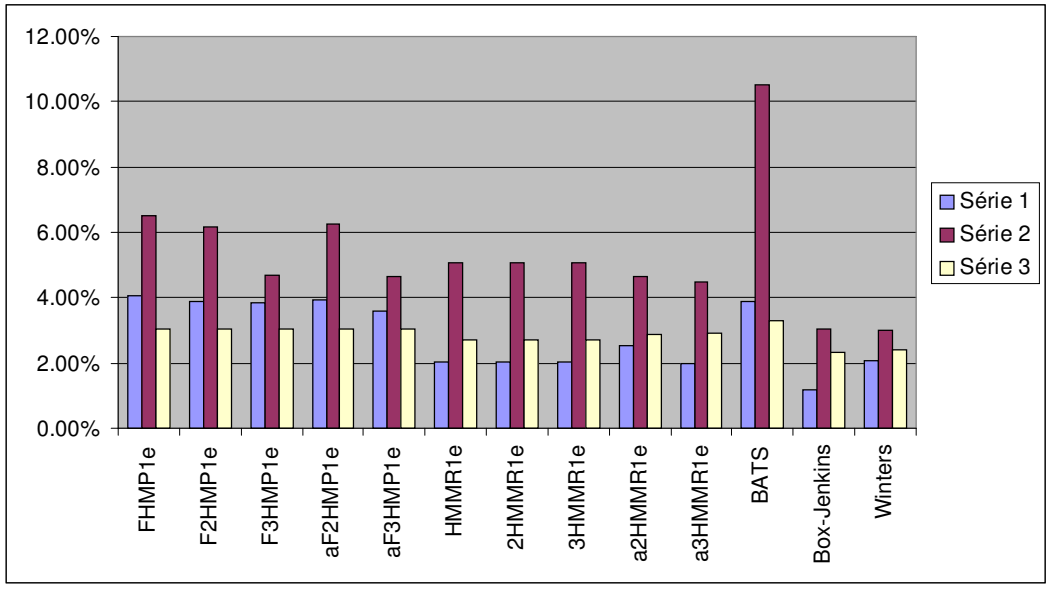


Figura 36: MAPE de PPD's e PPF's com 1 entrada e DT para 1. º ano de teste.

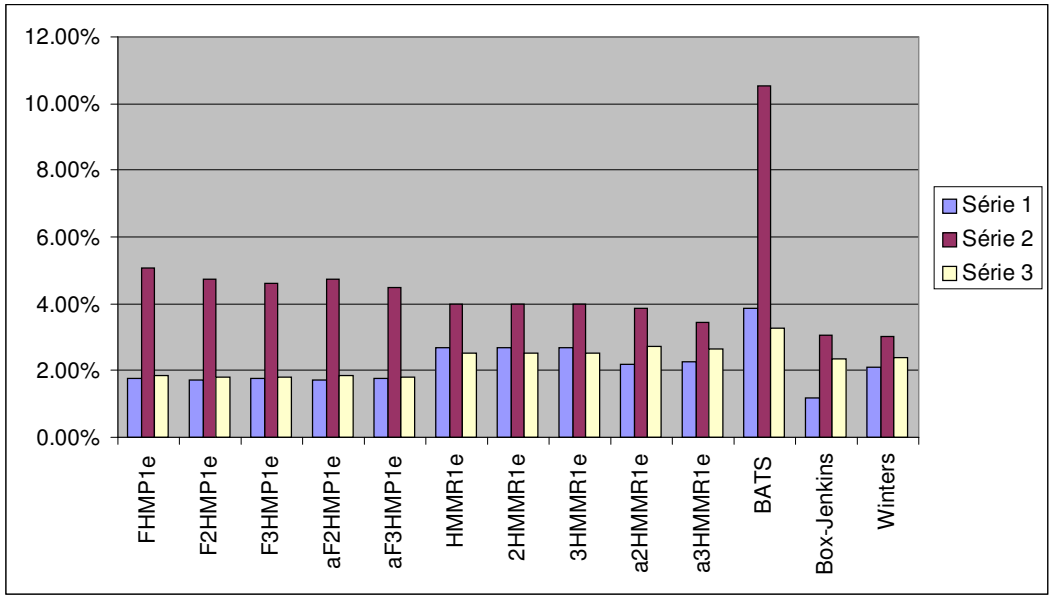


Figura 37: MAPE de PPD's e PPF's com 1 entrada, DT e KM para 1. º ano de teste.

8 - Conclusões e Trabalhos Futuros

Este capítulo apresenta algumas conclusões sobre o trabalho desenvolvido nesta tese e também são discutidos alguns tópicos que estão sendo atualmente investigados ou que são importantes para uma futura pesquisa.

8.1 - Conclusões

Desenvolvemos uma abordagem de previsão de séries temporais para a qual não existem muitos trabalhos na literatura: a previsão de um valor contínuo através da estimação não-paramétrica da função densidade de probabilidade da variável aleatória contínua que se deseja prever. Uma vez que a função densidade tenha sido estimada então temos que a previsão é dada pelo valor esperado da variável aleatória contínua.

Para essa estimação não-paramétrica empregamos Redes Bayesianas (RB) e Redes Bayesianas Dinâmicas (RBD) com variáveis aleatórias discretas. RB's são sistemas compostos de várias unidades interconectadas, generalizando redes neurais artificiais. Nas RB's as unidades representam variáveis aleatórias e as conexões dependências condicionais. RBD's são RB's que representam um modelo probabilístico temporal que é adequado para dados que possuem uma dependência temporal entre si.

Criamos vários sistemas que fazem previsão através do valor esperado usando uma função densidade estimada por RB's ou RBD's com variáveis discretas: *Markov Model for Regression*, *Hidden Markov Model for Regression* e *Multi-Hidden Markov Model for Regression*. A principal contribuição deste trabalho foi a generalização desses sistemas pelo uso da fuzzificação no lugar da discretização. Assim obtivemos vários preditores com variáveis aleatórias *fuzzy*: *Fuzzy Bayes Predictor*, *Fuzzy Markov Predictor*, *Fuzzy Hidden Markov Predictor* e *Fuzzy Multi-Hidden Markov Predictor*. Esses preditores probabilísticos que usam discretização foram denominados Preditores Probabilísticos Discretos (PPD) enquanto que aqueles que usam fuzzificação foram denominados Preditores Probabilísticos *Fuzzy* (PPF). É interessante notar que RB's e RBD's com apenas variáveis aleatórias contínuas representam a metodologia paramétrica para previsão através do valor esperado.

Outra contribuição deste trabalho foi o desenvolvimento de alguns métodos para efetuar o particionamento do espaço de dados contínuos com a finalidade de serem usados por nossos PPF's. Esses novos métodos são modificações dos métodos conhecidos por *Density Trees* e *K-Means* para funcionar com variáveis *fuzzy*.

Finalmente, aplicamos nossos sistemas de previsão às tarefas de previsão *single-step* e *multi-step* de séries de carga elétrica mensal. As séries temporais utilizadas possuem um comportamento de mudança abrupta e significativa em seus últimos anos, como quando ocorre um racionamento de energia. Obtivemos resultados competitivos quando comparamos nossos sistemas de previsão com várias técnicas estatísticas conhecidas. Embora tenhamos utilizado apenas séries de carga elétrica mensal em nossos experimentos, os sistemas de previsão que desenvolvemos são aplicáveis a qualquer outra série temporal. Esses resultados nos levam a acreditar que sistemas não-paramétricos para previsão através do valor esperado podem ser promissores na previsão de séries temporais.

8.2 - Tópicos para Futura Pesquisa

Existem vários pontos que ainda podem ser investigados no âmbito de nosso trabalho. Em primeiro lugar, como PPD's e PPF's estão estimando a função densidade de probabilidade de uma variável aleatória contínua, além de se fazer uma previsão pontual da série eles poderiam também obter um intervalo de confiança para essa previsão (de forma similar ao que é feito nos modelos de filtro de Kalman).

Os dois modelos de filtro de Kalman, STAMP e BATS, que utilizamos em nossos experimentos assumem que Gaussianas são empregadas como suas distribuições. Como esses modelos obtiveram resultados satisfatórios nas séries de carga elétrica mensal (ou seja, essas séries são modeladas de forma adequada por Gaussianas), seria interessante testar esses modelos e nossos preditores probabilísticos em séries não-Gaussianas, ou seja, séries cujos dados não são adequadamente modelados por Gaussianas. Assim seria possível averiguar qual abordagem, paramétrica com Gaussianas ou não-paramétrica com discretização/fuzzificação, é a mais eficiente aproximação para prever séries não-Gaussianas.

Em nossos testes, *Forward Validation* (FV) foi utilizado para efetuar a escolha do número de entradas e/ou partições nos PPD's e PPF's. O modelo selecionado pelo FV é

aquele que minimiza uma expressão que leva em conta tanto os erros de previsão cometidos em um conjunto de validação como também o número de parâmetros do modelo. Como o custo computacional do FV é bem grande (principalmente para modelos de estruturas mais complexas), seria bom o uso de métodos alternativos mais rápidos para a escolha do número de entradas e/ou partições. Uma possibilidade é escolher um modelo que maximiza (ou minimiza) uma função de avaliação como o *Bayesian Information Criterion* (BIC) [13], ou o *Minimum Description Length* (MDL) [13], ou o *Akaike's Criterion* (AIC) [43]. Essas funções de avaliação combinam a verossimilhança (*likelihood*) dos dados com uma penalidade estrutural que desencoraja modelos muito complexos. Para o caso de uma Redes Bayesiana B com variáveis $\mathbf{X} = (X_1, ..., X_n)$ e distribuição conjunta $\mathbf{P}_B(\mathbf{X})$, se considerarmos um conjunto de treinamento $D = \{\mathbf{x}^1, ..., \mathbf{x}^N\}$ onde cada $\mathbf{x}^i$ representa uma tupla de valores para as variáveis em $\mathbf{X}$, então a verossimilhança de B dado D é igual a $\prod_{i=1}^{N} P_B(\mathbf{x}^i)$. A verossimilhança possui uma interpretação estatística: quanto maior a verossimilhança, mais próximo B está de modelar a distribuição de probabilidade nos dados D. Se a RB contém variáveis não-observáveis no conjunto de treinamento, utiliza-se o valor esperado da verossimilhança nas funções de avaliação [12], [13]. O uso dessas funções de avaliação pode permitir a escolha de modelos com diferentes dependências entre suas variáveis aleatórias além de diferentes números de variáveis de estado e de evidência.

Em PPD's, as distribuições $\mathbf{P}(\mathbf{S}_t|\mathbf{S}_{t-1})$ e $\mathbf{P}(\mathbf{E}_t|\mathbf{S}_t)$ (ou $\mathbf{P}(S)$ e $\mathbf{P}(\mathbf{E}|S)$) para preditores baseados no *Naive Bayes Classifier*) representam estimativas de densidade contínuas feitas por histogramas. O histograma é o mais simples de todos os estimadores de densidade. Existem vários outros estimadores mais robustos [49]: *frequency polygon*, *averaged shifted histogram* e *kernel estimator*. O estimador de densidade de *kernel*, por exemplo, já foi utilizado no *Naive Bayes for Regression* (NBR) [11] como uma alternativa ao histograma. Logo, é importante descobrir se a substituição dos histogramas nos PPD's por outros estimadores de densidade pode produzir resultados mais satisfatórios em termos da realização da previsão de séries temporais.

Sistemas *fuzzy* como aquele visto no Capítulo 2 são conhecidos como *Type-1 Fuzzy Logic Systems* (FLS's). Neles os valores retornados pelas funções de pertinência são números reais no intervalo [0; 1]. Também existem os *Type-2* FLS's [24] onde os valores retornados pelas funções de pertinência são representados por regiões *fuzzy*, ou seja, o contradomínio dessas funções de pertinência também é dividido em regiões

fuzzy. Como os PPF's utilizam a mesma fuzzificação de um *Type*-1 FLS então seria interessante desenvolver PPF's nos quais o mecanismo de fuzzificação fosse o mesmo de um *Type*-2 FLS.

Por fim, na literatura existem sistemas que conseguiram generalizar o HMM em algum aspecto, como por exemplo o *Monte Carlo Hidden Markov Model* (MCHMM) [70] e o *Generalized Hidden Markov Model* (GHMM) [34], [35]. O MCHMM apresenta uma abordagem não-paramétrica para um HMM cujas variáveis de estado e de evidência são contínuas. Neste HMM contínuo as densidades de probabilidade necessárias são aproximadas por amostras e DT's geradas a partir dessas amostras. Percebe-se que o MCHMM e o HMMR são bem semelhantes (diferem apenas quanto a forma de aproximar as densidades de probabilidade), o que sugere a possibilidade de estender o MCHMM para trabalhar com várias variáveis de estado como no MHMMR. A principal característica da generalização feita pelo GHMM em relação ao HMM convencional é o relaxamento da propriedade aditiva existente nas medidas de probabilidade. Essa propriedade é substituída por uma bem menos restritiva: a monotonicidade com respeito à inclusão de conjuntos. Isso traz uma oportunidade para a pesquisa de preditores contínuos que aproximam medidas mais gerais que as da probabilidade.

Bibliografia

[1] H. Akaike. Canonical Correlations Analysis of Time Series and the Use of an Information Criterion. In R. Mehra & D.G. Lainiotis (eds.), *Advances and Case Studies in System Identification*, Academic Press, 1976.

[2] J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[3] G.E.P. Box, G.M. Jenkins & G.C. Reinsel. *Time Series Analysis: Forecasting & Control*. Prentice Hall, 1994.

[4] D.W. Bunn & E.D. Farmer. *Comparative Models for Electrical Load Forecasting*. John Wiley & Sons Ltd., 1985.

[5] L. Chan & F. Young. Using recurrent network for time series prediction. *Proceedings of World Congress on Neural Networks 1993*, Portland, volume 4, pp. 332-336, 1993.

[6] D.M. Chickering. Learning Bayesian networks is NP-complete. In D. Fisher & H. -J. Lenz (eds.), *Learning from Data: Artificial Intelligence and Statistics V*, pp. 121-130, Spring-Verlag, 1996.

[7] G. Cooper. Computational complexity of probabilistic inference using Bayesian belief networks (research note). *Artificial Intelligence*, v.42, pp.393-405, 1990.

[8] R.G. Cowell, A.P. Dawid, S.L. Lauritzen & D.J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Spring-Verlag, 1999.

[9] J.J De Gruijter & A.B. McBratney. A modified fuzzy k means for predictive classification. In: Bock,H.H.(ed) *Classification and Related Methods of Data Analysis*. pp. 97-104. Elsevier Science, Amsterdam, 1988.

[10] P. Domingos & M. Pazzani. On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. *Machine Learning* 29(2/3), pp. 103-130, 1997.

[11] E. Frank, L. Trigg, G. Holmes & I.H. Witten. Naive Bayes for Regression. *Machine Learning*, Vol.41, No.1, pp. 5-25, 1999.

[12] N. Friedman. Learning Bayesian networks in the presence of missing values and hidden variables. *Proceedings of the Fourteenth International Conference on Machine Learning*, pp. 125-133, 1997.

[13] N. Friedman, K. Murphy & S.J. Russell. Learning the structure of dynamic probabilistic networks. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pp. 139-147, 1998.

[14] W.A. Fuller. *Introduction to statistical time series*. John Wiley, 1996.

[15] Z. Ghahramani. Learning Dynamic Bayesian Networks. In C.L. Giles & M. Gori (eds.), *Adaptive Processing of Sequences and Data Structures*, *Lecture Notes in Artificial Intelligence*, vol. 1387, pp.168-197, Berlin, Springer-Verlag, 1998.

[16] J. Hartigan & M. Wong. A k-means clustering algorithm, ALGORITHM AS 136. *Applied Statistics*, Vol. 28, Number 1, pp. 100-108, 1979.

[17] A.C. Harvey. *Forecasting, structural time series models and the Kalman filter*. Cambridge University Press, 1994.

[18] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Macmillan College Publishing Company, New York, 1994.

[19] H.S. Hippert. Previsão de Cargas a Curto Prazo - Uma Avaliação da Viabilidade do Uso de Redes Neurais Artificiais. Tese de Doutorado, Departamento de Engenharia Elétrica, PUC/RJ, Março 2001.

[20] J.S.U. Hjorth. *Computer Intensive Statistical Methods. Validation Model Selection and Bootstrap*. Chapman & Hall. 1994.

[21] J.S.R Jang, C.T. Sun & E. Mizutani. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall Inc, USA, 1997.

[22] H. Jeffreys. *Theory of Probability* (3rd ed.). Oxford University Press, 1961.

[23] M.I. Jordan. *Learning in Graphical Models*. The MIT Press, 1999.

[24] N.N. Karnik & J.M. Mendel. Introduction to Type-2 Fuzzy Logic Systems. *Proc. 1998 IEEE FUZZY Conf.*, pp. 915-920, 1998.

[25] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, Berlim, Heidelberg, 1989.

[26] T. Koskela, M. Lehtokangas, J. Saarinen & K. Kaski. Time Series Prediction with Multilayer Perceptron, FIR and Elman Neural Networks. *Proc. of the World Congress on Neural Networks*, INNS Press, San Diego, USA, pp. 491-496, 1996.

[27] P.M. Lourenço. Um Modelo de Previsão de Curto Prazo de Cargas Elétricas Combinando Métodos Estatísticos e Inteligência Computacional. Tese de Doutorado, Departamento de Engenharia Elétrica, PUC/RJ, Junho 1998.

[28] P.M. Lourenço, C.R. Lourenço, G.F. Ribeiro & V.N.A.L. Silva. Short-Term Load Forecasting Using Fuzzy Logic and Calendar of Events. *19th International Symposium on Forecasting*, Washington DC, Junho 1999.

[29] M.A.S. Machado. Auxílio à Análise de Séries Temporais Não Sazonais Usando Redes Neurais Nebulosas. Tese de Doutorado, Departamento de Engenharia Elétrica, PUC/RJ, Junho 2000.

[30] T. Martinetz, S. Berkovich & K. Schulten. 'Neural-Gas' network for vector quantization and its application to times-series prediction. *IEEE Trans. on Neural Networks*, 4(4), pp. 558-569, 1993.

[31] J.M. Mendel. Fuzzy Logic Systems for Engineering: A Tutorial. *Proceedings of the IEEE*, vol.83, pp.345-377, Mar.1995.

[32] G.J. McLachlan & T. Krishnan. *The EM Algorithm and Extensions* (1st ed.). Wiley Interscience, 1997.

[33] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[34] M. Mohamed & P. Gader. Generalized hidden Markov models - part i: theoretical frameworks. *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 1, pp. 67-81, 2000.

[35] M. Mohamed & P. Gader. Generalized hidden Markov models - part ii: application to handwritten word recognition. *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 1, pp. 82-94, 2000.

[36] D. C. Montgomery, L. A. Johnson & J. S. Gardiner. *Forecasting and Time Series Analysis*. McGraw-Hill Companies, 1990.

[37] K.P. Murphy. Dynamic Bayesian Networks: Representation, Inference and Learning. PhD Thesis, UC Berkeley, Computer Science Division, Julho 2002.

[38] D.D. Nauck. Neuro-Fuzzy Systems: Review and Prospects. *Fifth European Congress on Intelligent Techniques and Soft Computing (EUFIT' 97),* pp. 1044-1053, 1997.

[39] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[40] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, vol. 77, no. 2, pp. 257-286, 1989.

[41] K. Revoredo & G. Zaverucha. Search-Based Class Discretization for Hidden Markov Model for Regression. *SBIA: 17th Brazilian Symposium on Artificial Intelligence, Lecture Notes in Computer Science*, vol. 3171, pp. 317-325, Springer, 2004.

[42] G.F.Ribeiro, V.N.A.L. da Silva, P.M. Lourenço, C.R.S.H. Lourenço & R. Linden. Comparative Studies of Short Term Load Forecasting Using Hybrid Neural Networks, *ISF'97*, Barbados, 1997.

[43] B.D. Ripley. Statistical ideas for selecting network architectures. In *Neural Networks: Artificial Intelligence and Industrial Applications*, eds B. Kappen and S. Gielen, Springer, pp. 183-190, 1995.

[44] S. Roweis & Z. Ghahramani. A Unifying Review of Linear Gaussian Models. *Neural Computation* 11(2) pp. 305-345, 1999.

[45] D.E. Rumelhart, G.E. Hinton & R.J. Willians. Learning Internal Representations by Error Propagation. *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, Vol. 1, pp. 318-361, MIT Press, 1986.

[46] S. Russell, J. Binder, D. Koller & K. Kanazawa. Local learning in probabilistic networks with hidden variables. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI95)*, pp. 1146-1152, 1995.

[47] S. Russell & P. Norvig. *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2.ª edição, 2002.

[48] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics* 6, pp. 461-464, 1978.

[49] D.W. Scott. Density Estimation. In P. Armitage & T. Colton, editors, *Encyclopedia of Biostatistics*, pp. 1134-1139. J. Wiley & Sons, Chichester, 1998.

[50] A.P.B. Sobral. Modelo de Previsão Horária de Carga Elétrica para LIGHT. Dissertação de Mestrado, Departamento de Engenharia Elétrica, PUC/RJ, Março 1999.

[51] R.C. Souza. Métodos Automáticos de Amortecimento Exponencial para Previsão de Séries Temporais. Monografia GSM-10/83, DEE PUC/RJ, Maio 1983.

[52] R.C. Souza. Modelos Estruturais para Previsão de Séries Temporais: Abordagens Clássica e Bayesiana. *17.º Colóquio Brasileiro de Matemática*. Instituto de Matemática Pura e Aplicada, 1989.

[53] R.C. Souza & M.E. Camargo. *Análise e Previsão de Séries Temporais: Os Modelos ARIMA*. SEDIGRAF, 1996.

[54] F.J. Souza, M.M.R. Vellasco & M.A.C. Pacheco. Hierarchical Neuro-Fuzzy Quadtree Models, *Fuzzy Set and Systems, IFSA*, Vol. 130(2), pp. 189-205, 2002.

[55] K. Swingler. Financial prediction: Some pointers, pitfalls and common errors. *Neural Computing Applications* 4(4), pp. 192-197, 1996.

[56] Task Force 38-06-06. Artificial neural networks for power systems. *Electra*, 159, pp. 77-101, 1995.

[57] M.A. Teixeira, G. Zaverucha, V. N. A. L. da Silva & G. F. Ribeiro. Previsão de Carga Elétrica Usando Redes de Elman. *V Simpósio Brasileiro de Redes Neurais (SBRN'98)*, pp. 161-164, 1998.

[58] M.A. Teixeira, G. Zaverucha, V.N.A.L. da Silva & G.F. Ribeiro. Evaluation and Comparison of Different Architectures Using Elman Networks Applied to Electric Load Forecasting. *Intelligent System Application to Power Systems (ISAP'99)*, pp. 3-7, 1999.

[59] M.A. Teixeira, G. Zaverucha, V.N.A.L. da Silva & G.F. Ribeiro. Recurrent Neural Gas in Electric Load Forecasting. *International Joint Conference on Neural Networks (IJCNN'99)*, Volume 5, pp. 3468-3473, 1999.

[60] M.A. Teixeira, G. Zaverucha, V.N.A.L.Silva & G.F. Ribeiro. Fuzzy Bayes Predictor in Electric Load Forecasting. *International Joint Conference on Neural Networks*, Washington DC, vol. 4, pp. 2339-234, Jullo 2001.

[61] M.A. Teixeira & G. Zaverucha. Fuzzy Markov Predictor in Electric Load Forecasting. *International Joint Conference on Neural Networks (IJCNN'2002),* vol. 3, pp. 2416-2421, 2002.

[62] M.A. Teixeira, K. Revoredo & G. Zaverucha. Hidden Markov model for regression in electric load forecasting. *ICANN/ICONIP*, Istanbul, Turkey, June 26-29, pp. 374-377. 2003.

[63] M.A. Teixeira & G. Zaverucha. Fuzzy Markov predictor in multi-step electric load forecasting. *International Joint Conference on Neural Networks (IJCNN)*, Portland, Oregon, USA, July 20-24, pp. 3065-3070, 2003.

[64] M.A. Teixeira & G. Zaverucha. Fuzzy Bayes and Fuzzy Markov Predictors. *Journal of Intelligent and Fuzzy Systems*. IOS Press, Amsterdam, The Netherlands, volume 13, numbers 2-4, pp. 155-165, 2003.

[65] M.A. Teixeira & G. Zaverucha. Fuzzy hidden Markov predictor in electric load forecasting. *International Joint Conference on Neural Networks*, Vol. 1, pp. 315-320, 2004.

[66] M.A. Teixeira & G. Zaverucha. A Partitioning Method for Fuzzy Probabilistic Predictors. *ICONIP, Lecture Notes in Computer Science,* vol. 3316, pp. 929-934, Springer, 2004.

[67] M.A. Teixeira & G. Zaverucha. Fuzzy multi-hidden Markov predictor in electric load forecasting. *International Joint Conference on Neural Networks*, aceito, 2005.

[68] M.A. Teixeira & G. Zaverucha. Um Método de Particionamento Baseado no K-Means para Preditores Probabilísticos Fuzzy. *VII Congresso Brasileiro de Redes Neurais (CBRN 2005)*, submetido, 2005.

[69] T. Terano, K. Asai & M. Sugeno. *Fuzzy Systems Theory and Its Applications*. Academic Press, Incorporated, 1992.

[70] S. Thrun, J. Langford & D. Fox. Monte Carlo hidden Markov models: Learning non-parametric models of partially observable stochastic processes. *Proc. of the International Conference on Machine Learning (ICML)*, pp. 415-424, 1999.

[71] E. Tito, G. Zaverucha, M. Vellasco & M.A. Pacheco. Applying Bayesian Neural Networks to Electric Load Forecasting. *Proc. Sixth IEEE International Conference on Neural Information Processing (ICONIP' 99)*, November, Perth, Australia, Volume 1, pp. 407-411, 1999.

[72] L. Wang & J.M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man and Cybernetics*, v.22, n.6, 1992.

[73] K. Warwick, A. Ekwue & R. Aggarwal. *Artificial Intelligence Techniques in Power Systems*. The Institution of Electrical Engineers, London, United Kingdom, 1997.

[74] M. West & J. Harrison. *Bayesian Forecasting and Dynamic Models* (2nd ed.). Springer, 1997.

[75] L.A. Zadeh. Probability measures of fuzzy events. *Jour. Math. Analysis and Appl.* 23, 421-427, 1968.

[76] R.S. Zebulum. Previsão de Carga em Sistemas Elétricos de Potência por Redes Neurais. Dissertação de Mestrado, Departamento de Engenharia Elétrica, PUC/RJ, Agosto 1995.

[77] G.P. Zhang, B.E. Patuwo & M.Y. Hu. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting* 14, pp. 35-62, 1998.

Apêndice A

Este apêndice apresenta as demonstrações de que os Preditores Probabilísticos Discretos e *Fuzzy* vistos nos Capítulos 4 e 5, respectivamente, estão estimando uma função densidade de probabilidade.

A.1 - *Naive Bayes for Regression*

O *Naive Bayes for Regression* (NBR) está estimando a seguinte função densidade de probabilidade:

$f(v|\mathbf{a}) = f(v \mid a_1, a_2, \ldots, a_m) = f(v).\prod_j f(a_j|v) / \int(f(v).\prod_j f(a_j|v))dv$

através do uso de estimativas de histograma [49]:

$f(v) \approx N(s) / (N \,.\, h_s) = P(s) / h_s \,,\; v \in s$

$f(a_j, v) \approx N(e_j \,, s) / (N \,.\, h_{e_j} \,.\, h_s) = P(e_j \,, s) / (h_{e_j} \,.\, h_s) \,,\; a_j \in e_j \text{ e } v \in s$

$f(a_j|v) = f(a_j, v) / f(v) \approx N(e_j \,, s) / (N(s) \,.\, h_{e_j}) = P(e_j|s) / h_{e_j} \,,\; a_j \in e_j \text{ e } v \in s$

onde h_{e_j} e h_s são os tamanhos dos intervalos e_j e s, respectivamente.

Substituindo as estimativas de f(v) e $f(a_j|v)$ no numerador de $f(v|\mathbf{a})$ obtém-se:

$f(v).\prod_j f(a_j|v) \approx (P(s) / h_s).\prod_j (P(e_j|s) / h_{e_j}) = P(s).\prod_j P(e_j|s) / (h_s.\prod_j h_{e_j})$

para $v \in s$, $a_1 \in e_1$, $a_2 \in e_2$, ... e $a_m \in e_m$.

O denominador de $f(v|\mathbf{a})$ é a integral desse numerador:

$\int(f(v).\prod_j f(a_j|v))dv \approx \int(P(s).\prod_j P(e_j|s) / (h_s.\prod_j h_{e_j}))dv$

para $v \in s$, $a_1 \in e_1$, $a_2 \in e_2$, ... e $a_m \in e_m$. O intervalo dessa integração é o domínio de v. Como a expressão dentro dessa integral é constante para qualquer v dentro de um mesmo intervalo s, ela é dividida em várias outras integrais onde cada uma possui um intervalo de integração igual aos limites de cada intervalo s:

$\int(f(v).\prod_j f(a_j|v))dv \approx \int(P(s).\prod_j P(e_j|s) / (h_s.\prod_j h_{e_j}))dv$ [usando estimativas de histograma]

$= \sum_s \int_s (P(s).\prod_j P(e_j|s) / (h_s.\prod_j h_{e_j}))dv$ [dividindo em várias integrais]

$= \sum_s \{P(s).\prod_j P(e_j|s) / (h_s.\prod_j h_{e_j})\}.(\int_s dv)$ [extraindo as expressões constantes das integrais]

$= \sum_s \{P(s).\prod_j P(e_j|s) / (h_s.\prod_j h_{e_j})\}.(c_s - b_s)$ [resolvendo as integrais]

$= \sum_s \{P(s).\prod_j P(e_j|s) / (h_s.\prod_j h_{e_j})\}.(h_s)$ [usando $h_s = (c_s - b_s)$]

$= (\sum_s P(s).\prod_j P(e_j|s)) / \prod_j h_{e_j}$

onde $\int_s$ indica um intervalo de integração igual aos limites do intervalo s, c_s é o limite superior de s e b_s é o limite inferior.

Colocando-se esse numerador e denominador em $f(v|\mathbf{a})$:

$f(v|\mathbf{a}) \approx P(s).\prod_j P(e_j|s) / (h_s.\sum_s P(s).\prod_j P(e_j|s)) = \alpha.P(s).\prod_j P(e_j|s) / h_s = P(s|\mathbf{e}) / h_s$

para $v \in s$, $a_1 \in e_1$, $a_2 \in e_2$, ... e $a_m \in e_m$.

Tendo em mãos a estimativa da função densidade de probabilidade $f(v|\mathbf{a})$, o próximo passo é calcular o valor esperado da variável contínua V (conhecidos os valores contínuos $a_1, a_2, ..., a_m$). Esse valor esperado é a previsão do NBR (v_{NBR}):

$v_{NBR} = \int v.f(v|\mathbf{a})dv \approx \int v.\{P(s|\mathbf{e}) / h_s\}dv$ [usando a estimativa de $f(v|\mathbf{a})$]

$= \sum_s \int_s v.\{P(s|\mathbf{e}) / h_s\}dv$ [dividindo em várias integrais]

$= \sum_s \{P(s|\mathbf{e}) / h_s\}.(\int_s v\, dv)$ [extraindo as expressões constantes das integrais]

$= \sum_s \{P(s|\mathbf{e}) / h_s\}.(c_s^2 - b_s^2) / 2$ [resolvendo as integrais]

$= \sum_s \{P(s|\mathbf{e}) / h_s\}.(c_s - b_s).(c_s + b_s) / 2$

$= \sum_s \{P(s|\mathbf{e}) / h_s\}.(h_s).m(s)$ [usando $h_s = (c_s - b_s)$ e $m(s) = (c_s + b_s) / 2$]

$= \sum_s m(s).P(s|\mathbf{e})$

que é exatamente o resultado da previsão contínua do NBR apresentado inicialmente no Capítulo 4.

A.2 - *Hidden Markov Model for Regression*

Da mesma forma que o NBR, o *Hidden Markov Model for Regression* (HMMR) também está estimando uma função densidade de probabilidade, neste caso a densidade da previsão da variável de evidência:

$f(a_{t+1,j} | \mathbf{a}_{1:t}) = \int \{f(a_{t+1,j} | v_{t+1}).f(v_{t+1} | \mathbf{a}_{1:t})\} dv_{t+1}$

onde a densidade da previsão da variável de estado é dada por

$f(v_{t+1} | \mathbf{a}_{1:t}) = \int \{f(v_{t+1} | v_t).f(v_t | \mathbf{a}_{1:t})\} dv_t$

e a densidade da filtragem da variável de estado é

se $t = 0$ então $f(v_t | \mathbf{a}_{1:t}) = f(v_t)$

se $t > 0$ então $f(v_t | \mathbf{a}_{1:t}) = \beta.f(\mathbf{a}_t | v_t).f(v_t | \mathbf{a}_{1:t-1}) = \beta.\{\prod_j f(a_{t,j}|v_t)\}.f(v_t | \mathbf{a}_{1:t-1})$

com a constante de normalização $\beta = 1 / \int \{f(\mathbf{a}_t | v_t).f(v_t | \mathbf{a}_{1:t-1})\} dv_t$.

A estimativa de $f(a_{t+1,j} \mid \mathbf{a}_{1:t})$ faz uso das seguintes estimativas de histograma:

$f(v_t) \approx N(s_t) / (N \,.\, h_{s_t}) = P(s_t) / h_{s_t}$, $v_t \in s_t$

$f(a_{t,j}, v_t) \approx N(e_{t,j}\,, s_t) / (N \,.\, h_{e_{t,j}} \,.\, h_{s_t}) = P(e_{t,j}\,, s_t) / (h_{e_{t,j}} \,.\, h_{s_t})$, $a_{t,j} \in e_{t,j}$ e $v_t \in s_t$

$f(a_{t,j}|v_t) = f(a_{t,j}, v_t) / f(v_t) \approx N(e_{t,j}\,, s_t) / (N(s_t) \,.\, h_{e_{t,j}}) = P(e_{t,j}|s_t) / h_{e_{t,j}}$, $a_{t,j} \in e_{t,j}$ e $v_t \in s_t$

$f(v_{t+1}, v_t) \approx N(s_{t+1}\,, s_t) / (N \,.\, h_{s_{t+1}} \,.\, h_{s_t}) = P(s_{t+1}\,, s_t) / (h_{s_{t+1}} \,.\, h_{s_t})$, $v_{t+1} \in s_{t+1}$ e $v_t \in s_t$

$f(v_{t+1}|v_t) = f(v_{t+1}, v_t) / f(v_t) \approx N(s_{t+1}, s_t) / (N(s_t).h_{s_{t+1}}) = P(s_{t+1}|s_t) / h_{s_{t+1}}$, $v_{t+1} \in s_{t+1}$ e $v_t \in s_t$

onde $h_{e_{t,j}}$ e h_{s_t} são os tamanhos dos intervalos $e_{t,j}$ e s_t, respectivamente. É importante lembrar que os tamanhos dos intervalos não mudam com o tempo, isto é, $h_{e_{t,j}} = h_{e_j}$ e $h_{s_t} = h_s$ para todo instante t.

Primeiro, temos que estimar as densidades da filtragem e da previsão do estado. Se considerarmos t = 1, a filtragem

$f(v_t|\mathbf{a}_{1:t}) = \{\prod_j f(a_{t,j}|v_t)\}.f(v_t|\mathbf{a}_{1:t-1}) / \int\{\prod_j f(a_{t,j}|v_t)\}.f(v_t|\mathbf{a}_{1:t-1})dv_t$

se torna igual a

$f(v_1|\mathbf{a}_{1:1}) = \{\prod_j f(a_{1,j}|v_1)\}.f(v_1) / \int\{\prod_j f(a_{1,j}|v_1)\}.f(v_1)dv_1$

que é praticamente idêntica a densidade $f(v|\mathbf{a})$ do NBR (com exceção dos índices de tempo). Logo, podemos substituir as estimativas de histograma em $f(v_1|\mathbf{a}_{1:1})$ e seguir os mesmos passos da dedução da estimativa de $f(v|\mathbf{a})$ para obter

$f(v_1|\mathbf{a}_{1:1}) \approx P(s_1).\prod_j P(e_{1,j}|s_1) / (h_s.\sum_{s_1} P(s_1).\prod_j P(e_{1,j}|s_1)) = P(s_1|\mathbf{e}_{1:1}) / h_s$

para $v_1 \in s_1$, $a_{1,1} \in e_{1,1}$, $a_{1,2} \in e_{1,2}$, ... e $a_{1,m} \in e_{1,m}$.

Para t = 2 a filtragem se torna igual a

$f(v_2|\mathbf{a}_{1:2}) = \{\prod_j f(a_{2,j}|v_2)\}.f(v_2|\mathbf{a}_{1:1}) / \int\{\prod_j f(a_{2,j}|v_2)\}.f(v_2|\mathbf{a}_{1:1})dv_2$

usando a previsão

$f(v_2 \mid \mathbf{a}_{1:1}) = \int\{f(v_2 \mid v_1).f(v_1 \mid \mathbf{a}_{1:1})\}dv_1$

A densidade $f(v_2 \mid \mathbf{a}_{1:1})$ é a integração de um produtório de densidades. Isso não é muito diferente do denominador de $f(v|\mathbf{a})$ e assim pode-se seguir os mesmos passos (substituir estimativas, dividir em várias integrais, extrair constantes das integrais, resolver integrais) empregados na dedução desse denominador:

$f(v_2|\mathbf{a}_{1:1}) \approx (\sum_{s_1} P(s_2 \mid s_1).P(s_1 \mid \mathbf{e}_{1:1})) / h_s = P(s_2 \mid \mathbf{e}_{1:1}) / h_s$

A dedução da densidade $f(v_2|\mathbf{a}_{1:2})$ segue a mesma feita para a densidade $f(v|\mathbf{a})$:

$f(v_2|\mathbf{a}_{1:2}) \approx \{\prod_j P(e_{2,j}|s_2)\}.P(s_2|\mathbf{e}_{1:1}) / (h_s \,.\, \sum_{s_2}\{\prod_j P(e_{2,j}|s_2)\}.P(s_2|\mathbf{e}_{1:1})) = P(s_2 \mid \mathbf{e}_{1:2}) / h_s$

Generalizando para t > 0:

$f(v_t|\mathbf{a}_{1:t}) \approx \{\prod_j P(e_{t,j}|s_t)\}.P(s_t|\mathbf{e}_{1:t-1}) / (h_s \,.\, \sum_{s_t}\{\prod_j P(e_{t,j}|s_t)\}.P(s_t|\mathbf{e}_{1:t-1})) = P(s_t \mid \mathbf{e}_{1:t}) / h_s$

$f(v_t|\mathbf{a}_{1:t-1}) \approx (\sum_{s_{t-1}} P(s_t \mid s_{t-1}).P(s_{t-1} \mid \mathbf{e}_{1:t-1})) / h_s = P(s_t \mid \mathbf{e}_{1:t-1}) / h_s$

Substituindo-se as estimativas de $f(v_t \mid \mathbf{a}_{1:t-1})$ e $f(a_{t,j} \mid v_t)$ em $f(a_{t+1,j} \mid \mathbf{a}_{1:t})$:

$f(a_{t+1,j}|\mathbf{a}_{1:t}) \approx \int\{P(e_{t+1,j}|s_{t+1}) / h_{e_j}\}.\{P(s_{t+1} \mid \mathbf{e}_{1:t}) / h_s\}dv_{t+1}$

$f(a_{t+1,j}|\mathbf{a}_{1:t}) \approx \sum_{s_{t+1}}\int_{s_{t+1}}\{P(e_{t+1,j}|s_{t+1}) / h_{e_j}\}.\{P(s_{t+1} \mid \mathbf{e}_{1:t}) / h_s\}dv_{t+1}$ [dividindo em integrais]

$f(a_{t+1,j}|\mathbf{a}_{1:t}) \approx \sum_{s_{t+1}}\{P(e_{t+1,j}|s_{t+1}) / h_{e_j}\}.\{P(s_{t+1} \mid \mathbf{e}_{1:t}) / h_s\}.\int_{s_{t+1}}dv_{t+1}$ [extraindo constantes]

$f(a_{t+1,j}|\mathbf{a}_{1:t}) \approx \sum_{s_{t+1}}\{P(e_{t+1,j}|s_{t+1}) / h_{e_j}\}.\{P(s_{t+1} \mid \mathbf{e}_{1:t}) / h_s\}.(c_{s_{t+1}} - b_{s_{t+1}})$ [resolvendo integrais]

$f(a_{t+1,j}|\mathbf{a}_{1:t}) \approx \sum_{s_{t+1}}\{P(e_{t+1,j}|s_{t+1}) / h_{e_j}\}.\{P(s_{t+1} \mid \mathbf{e}_{1:t}) / h_s\}.(h_s)$ [usando $h_s = (c_{s_{t+1}} - b_{s_{t+1}})$]

$f(a_{t+1,j}|\mathbf{a}_{1:t}) \approx \{\sum_{s_{t+1}}P(e_{t+1,j}|s_{t+1}).P(s_{t+1} \mid \mathbf{e}_{1:t})\} / h_{e_j}\}$

$f(a_{t+1,j}|\mathbf{a}_{1:t}) \approx P(e_{t+1,j}|\mathbf{e}_{1:t}) / h_{e_j}$ [usando a previsão de uma observação futura do HMM]

para $a_{t+1,j} \in e_{t+1,j}$, ($a_{1,1} \in e_{1,1}$, ... e $a_{1,m} \in e_{1,m}$) ... e ($a_{t,1} \in e_{t,1}$, ... e $a_{t,m} \in e_{t,m}$).

O valor esperado da variável contínua $A_{t+1,j}$ (condicionada pelas observações conhecidas até o instante t) é a previsão do HMMR ($a_{HMMR\ t,j}$):

$a_{HMMR\ t,j} = \int a_{t+1,j}.f(a_{t+1,j}|\mathbf{a}_{1:t})da_{t+1,j} \approx \int a_{t+1,j}.\{P(e_{t+1,j}|\mathbf{e}_{1:t}) / h_{e_j}\}da_{t+1,j}$ [estimativa de $f(a_{t+1,j}|\mathbf{a}_{1:t})$]

$= \sum_{e_{t+1,j}}\int_{e_{t+1,j}} a_{t+1,j}.\{P(e_{t+1,j}|\mathbf{e}_{1:t}) / h_{e_j}\}da_{t+1,j}$ [dividindo em várias integrais]

$= \sum_{e_{t+1,j}}\{P(e_{t+1,j}|\mathbf{e}_{1:t}) / h_{e_j}\}.(\int_{e_{t+1,j}}a_{t+1,j}\ da_{t+1,j})$ [extraindo expressões constantes das integrais]

$= \sum_{e_{t+1,j}}\{P(e_{t+1,j}|\mathbf{e}_{1:t}) / h_{e_j}\}.(c_{e_{t+1,j}}^{\ 2} - b_{e_{t+1,j}}^{\ 2})/2$ [resolvendo as integrais]

$= \sum_{e_{t+1,j}}\{P(e_{t+1,j}|\mathbf{e}_{1:t}) / h_{e_j}\}.(c_{e_{t+1,j}}-b_{e_{t+1,j}}).(c_{e_{t+1,j}}+b_{e_{t+1,j}})/2$

$= \sum_{e_{t+1,j}}\{P(e_{t+1,j}|\mathbf{e}_{1:t}) / h_{e_j}\}.(h_{e_j}).m(e_{t+1,j})$ [com $h_{e_j} = (c_{e_{t+1,j}}-b_{e_{t+1,j}})$ e $m(e_{t+1,j}) = (c_{e_{t+1,j}}+b_{e_{t+1,j}})/2$]

$= \sum_{e_{t+1,j}}\{m(e_{t+1,j}).P(e_{t+1,j}|\mathbf{e}_{1:t})\}$

que corresponde ao resultado da previsão contínua do HMMR apresentado antes no Capítulo 4.

A.3 - *Markov Model for Regression*

A função densidade de probabilidade sendo estimada *pelo Markov Model for Regression* (MMR) é a densidade da filtragem da variável de estado:

se $t = 0$ então $f(v_t \mid \mathbf{a}_{1:t}) = f(v_t)$

se $t > 0$ então $f(v_t \mid \mathbf{a}_{1:t}) = \beta.f(\mathbf{a}_t \mid v_t).f(v_t \mid \mathbf{a}_{1:t-1}) = \beta.\{\prod_j f(a_{t,j}|v_t)\}.f(v_t \mid \mathbf{a}_{1:t-1})$

com a constante de normalização $\beta = 1 / \int\{f(\mathbf{a}_t \mid v_t).f(v_t \mid \mathbf{a}_{1:t-1})\}dv_t$.

Através das mesmas estimativas de histograma do HMMR, obtém-se a seguinte estimativa de $f(v_t \mid \mathbf{a}_{1:t})$ para o MMR:

$f(v_t|\mathbf{a}_{1:t}) \approx P(s_t \mid \mathbf{e}_{1:t}) / h_s$

para $v_t \in s_t$, ($a_{1,1} \in e_{1,1}$, ... e $a_{1,m} \in e_{1,m}$) ... e ($a_{t,1} \in e_{t,1}$, ... e $a_{t,m} \in e_{t,m}$).

O valor esperado da variável contínua V_t (condicionada pelos valores contínuos $\mathbf{a}_{1:t}$) é a previsão do MMR ($v_{MMR\,t}$):

$v_{MMR\,t} = \int v_t.f(v_t|\mathbf{a}_{1:t})dv_t \approx \int v_t.\{P(s_t|\mathbf{e}_{1:t}) / h_s\}dv_t$ [usando a estimativa de $f(v_t|\mathbf{a}_{1:t})$]

$= \sum_{s_t}\{m(s_t).P(s_t|\mathbf{e}_{1:t})\}$ [obtido de forma similar ao HMMR]

que é o mesmo resultado inicialmente apresentado da previsão contínua do MMR no Capítulo 4.

A.4 - *Multi-Hidden Markov Model for Regression*

O *Multi-Hidden Markov Model for Regression* (MHMMR) está estimando a mesma função densidade de probabilidade que o HMMR, ou seja, a densidade da previsão da variável de evidência. A diferença é que agora o cálculo dessa densidade pressupõe que os dados contínuos são modelados por uma Rede Bayesiana Dinâmica (RBD) com w variáveis contínuas de estado $\mathbf{V}_t = (V_{t,1}, V_{t,2}, ..., V_{t,w})$ mas mantendo as mesmas m variáveis contínuas de evidência $\mathbf{A}_t = (A_{t,1}, A_{t,2}, ..., A_{t,m})$:

$f(a_{t+1,j} \mid \mathbf{a}_{1:t}) = \int\{f(a_{t+1,j} \mid \mathbf{v}_{t+1}).f(\mathbf{v}_{t+1} \mid \mathbf{a}_{1:t})\}d\mathbf{v}_{t+1}$

onde a densidade da previsão do conjunto das variáveis de estado é dada por

$f(\mathbf{v}_{t+1} \mid \mathbf{a}_{1:t}) = \int\{f(\mathbf{v}_{t+1} \mid \mathbf{v}_t).f(\mathbf{v}_t \mid \mathbf{a}_{1:t})\}d\mathbf{v}_t$

e a densidade da filtragem do conjunto das variáveis de estado é

se $t = 0$ então $f(\mathbf{v}_t \mid \mathbf{a}_{1:t}) = f(\mathbf{v}_t)$

se $t > 0$ então $f(\mathbf{v}_t \mid \mathbf{a}_{1:t}) = \beta.f(\mathbf{a}_t \mid \mathbf{v}_t).f(\mathbf{v}_t \mid \mathbf{a}_{1:t-1}) = \beta.\{\prod_j f(a_{t,j}|\mathbf{v}_t)\}.f(\mathbf{v}_t \mid \mathbf{a}_{1:t-1})$

com a constante de normalização $\beta = 1 / \int\{f(\mathbf{a}_t \mid \mathbf{v}_t).f(\mathbf{v}_t \mid \mathbf{a}_{1:t-1})\}d\mathbf{v}_t$.

A estimativa de $f(a_{t+1,j} \mid \mathbf{a}_{1:t})$ faz uso das seguintes estimativas de histograma:

$f(\mathbf{v}_t) = f(v_{t,1}, ..., v_{t,w}) \approx N(s_{t,1}, ..., s_{t,w}) / (N . h_{s_1} . ... . h_{s_w}) = P(s_{t,1}, ..., s_{t,w}) / \prod_k h_{s_k} =$

$= P(\mathbf{s}_t) / \prod_k h_{s_k}$, para $\mathbf{v}_t \in \mathbf{s}_t$ (ou seja, para $v_{1,1} \in s_{1,1}$, ... e $v_{1,m} \in s_{1,w}$)

$f(a_{t,j}, \mathbf{v}_t) = f(a_{t,j}, v_{t,1}, ..., v_{t,w}) \approx N(e_{t,j}, s_{t,1}, ..., s_{t,w}) / (N . h_{e_j} . h_{s_1} . ... . h_{s_w}) =$

$= P(e_{t,j}, s_{t,1}, ..., s_{t,w}) / (h_{e_j} . \prod_k h_{s_k}) = P(e_{t,j}, \mathbf{s}_t) / (h_{e_j} . \prod_k h_{s_k})$, $a_{t,j} \in e_{t,j}$ e $\mathbf{v}_t \in \mathbf{s}_t$

$f(a_{t,j}|\mathbf{v}_t) = f(a_{t,j}, \mathbf{v}_t) / f(\mathbf{v}_t) \approx P(e_{t,j}, \mathbf{s}_t) / (P(\mathbf{s}_t) . h_{e_j}) = P(e_{t,j}|\mathbf{s}_t) / h_{e_j}$, $a_{t,j} \in e_{t,j}$ e $\mathbf{v}_t \in \mathbf{s}_t$

$f(\mathbf{v}_{t+1}, \mathbf{v}_t) = f(v_{t+1,1}, ..., v_{t+1,w}, v_{t,1}, ..., v_{t,w}) \approx$

$\approx N(s_{t+1,1}, ..., s_{t+1,w}, s_{t,1}, ..., s_{t,w}) / (N \,.\, (\prod_k h_{s_k})^2) =$

$= P(s_{t+1,1}, ..., s_{t+1,w}, s_{t,1}, ..., s_{t,w}) / (\prod_k h_{s_k})^2 =$

$= P(\mathbf{s}_{t+1}, \mathbf{s}_t) / (\prod_k h_{s_k})^2$, $\mathbf{v}_{t+1} \in \mathbf{s}_{t+1}$ e $\mathbf{v}_t \in \mathbf{s}_t$

$f(\mathbf{v}_{t+1}|\mathbf{v}_t) = f(\mathbf{v}_{t+1}, \mathbf{v}_t) / f(\mathbf{v}_t) \approx P(\mathbf{s}_{t+1}, \mathbf{s}_t) / (P(\mathbf{s}_t) \,.\, \prod_k h_{s_k}) =$

$= P(\mathbf{s}_{t+1}|\mathbf{s}_t) / \prod_k h_{s_k}$, $\mathbf{v}_{t+1} \in \mathbf{s}_{t+1}$ and $\mathbf{v}_t \in \mathbf{s}_t$

Através dessas estimativas de histograma obtém-se estimativas para $f(\mathbf{v}_{t+1}|\mathbf{a}_{1:t})$ e $f(\mathbf{v}_t|\mathbf{a}_{1:t})$:

$f(\mathbf{v}_{t+1}|\mathbf{a}_{1:t}) \approx P(\mathbf{s}_{t+1}|\mathbf{e}_{1:t}) / \prod_k h_{s_k}$

$f(\mathbf{v}_t|\mathbf{a}_{1:t}) \approx P(\mathbf{s}_t|\mathbf{e}_{1:t}) / \prod_k h_{s_k}$

e a partir dessas estima-se $f(a_{t+1,j} \mid \mathbf{a}_{1:t})$:

$f(a_{t+1,j}|\mathbf{a}_{1:t}) \approx P(e_{t+1,j}|\mathbf{e}_{1:t}) / h_{e_j}$

para $a_{t+1,j} \in e_{t+1,j}$, $(a_{1,1} \in e_{1,1}$, ... e $a_{1,m} \in e_{1,m})$ ... e $(a_{t,1} \in e_{t,1}$, ... e $a_{t,m} \in e_{t,m})$.

A previsão do MHMMR ($a_{MHMMR\ t,j}$) é o valor esperado da variável contínua $A_{t+1,j}$ condicionada pelas observações conhecidas até o instante t:

$a_{MHMMR\ t,j} = \int a_{t+1,j}.f(a_{t+1,j}|\mathbf{a}_{1:t})da_{t+1,j} \approx \sum_{e_{t+1,j}}\{m(e_{t+1,j}).P(e_{t+1,j}|\mathbf{e}_{1:t})\}$

que é exatamente igual à previsão contínua do MHMMR vista anteriormente no Capítulo 4.

A.5 - *Fuzzy Bayes Predictor*

O *Fuzzy Bayes Predictor* (FBP) está aproximando a mesma função densidade de probabilidade estimada pelo NBR:

$f(v|\mathbf{a}) = f(v).\prod_j f(a_j|v) / \int(f(v).\prod_j f(a_j|v))dv$

através da seguinte fórmula de defuzzificação:

$f(v|\mathbf{a}) \approx (\sum_s \sum_{\mathbf{e} \in d} m_s(v).m_{\mathbf{e}}(\mathbf{a}).f_{s,\mathbf{e}}(v|\mathbf{a})) / (\sum_s \sum_{\mathbf{e} \in d} m_s(v).m_{\mathbf{e}}(\mathbf{a}))$

que é simplificada para (supondo um sistema de informação *fuzzy* ortogonal):

$f(v|\mathbf{a}) \approx \sum_s \sum_{\mathbf{e} \in d} m_s(v).m_{\mathbf{e}}(\mathbf{a}).f_{s,\mathbf{e}}(v|\mathbf{a})$

e cada $f_{s,\mathbf{e}}(v|\mathbf{a})$ sendo combinado pela defuzzificação é dado por:

$f_{s,\mathbf{e}}(v|\mathbf{a}) = P(s).\prod_j P(e_j|s) / (h_s.\sum_s P(s).\prod_j P(e_j|s)) = P(s|\mathbf{e}) / h_s$, para $m_s(v).m_{\mathbf{e}}(\mathbf{a}) > 0$

onde P(.) e P(.|.) são probabilidades *fuzzy*, e h_s é a área da região *fuzzy* s ($h_s = \int m_s(v)dv$). Se as funções de pertinência do FBP forem escolhidas de forma que a fuzzificação corresponda a uma discretização dos dados contínuos então a aproximação de f(v|**a**) feita pelo FBP torna-se igual àquela feita pelo NBR.

O valor esperado da variável contínua V condicionada pelos valores contínuos a_1, a_2, ..., a_m corresponde à previsão do FBP (v_{FBP}):

$v_{FBP} = \int v.f(v|\mathbf{a})dv \approx \int v.\{\sum_s \sum_{\mathbf{e} \in d} m_s(v).m_{\mathbf{e}}(\mathbf{a}).f_{s,\mathbf{e}}(v|\mathbf{a})\}dv$ [pela aproximação de f(v|**a**)]

$= \int v.\{\sum_s \sum_{\mathbf{e} \in d} m_s(v).m_{\mathbf{e}}(\mathbf{a}).P(s|\mathbf{e}) / h_s\}dv$ [substituindo pela expressão de $f_{s,\mathbf{e}}(v|\mathbf{a})$]

$= \sum_s \sum_{\mathbf{e} \in d} \int\{v.m_s(v).m_{\mathbf{e}}(\mathbf{a}).P(s|\mathbf{e}) / h_s\}dv$ [dividindo em várias integrais]

$= \sum_s \sum_{\mathbf{e} \in d} \{m_{\mathbf{e}}(\mathbf{a}).P(s|\mathbf{e}) / h_s\}.\int\{v.m_s(v)\}dv$ [extraindo constantes das integrais]

$= \sum_s \sum_{\mathbf{e} \in d} \{m_{\mathbf{e}}(\mathbf{a}).P(s|\mathbf{e}) / h_s\}.\ \overline{s}\ .\ h_s$ [usando $\overline{s} = \int v.m_s(v)dv / h_s$]

$= \sum_s\ \overline{s}\ .\ \sum_{\mathbf{e} \in d} \{m_{\mathbf{e}}(\mathbf{a}).P(s|\mathbf{e})\}$

$= \sum_s\ \overline{s}\ .\ P(s|d)$ [usando $P(s|d) = \sum_{\mathbf{e} \in d} m_{\mathbf{e}}(\mathbf{a}).P(s|\mathbf{e})$]

que é o mesmo resultado da previsão contínua do FBP mostrado anteriormente no Capítulo 5. Se as funções de pertinência do FBP forem selecionadas de forma que a fuzzificação corresponda a uma discretização então a previsão contínua do FBP torna-se igual a do NBR.

A.6 - *Fuzzy Hidden Markov Predictor*

A função densidade de probabilidade sendo aproximada pelo *Fuzzy Hidden Markov Predictor* (FHMP) é igual àquela estimada pelo HMMR:

$f(a_{t+1,j}|\mathbf{a}_{1:t}) = \int\{f(a_{t+1,j}|v_{t+1}).f(v_{t+1}|\mathbf{a}_{1:t})\}dv_{t+1}$

O FHMP utiliza a seguinte fórmula de defuzzificação para aproximar a densidade da previsão da variável de evidência:

$f(a_{t+1,j}|\mathbf{a}_{1:t}) \approx (\sum_{e_{t+1,j}} m_{e_{t+1,j}}(a_{t+1,j}).f_{e_{t+1,j}}(a_{t+1,j}|\mathbf{a}_{1:t})) / (\sum_{e_{t+1,j}} m_{e_{t+1,j}}(a_{t+1,j}))$

que é simplificada para (supondo um sistema de informação *fuzzy* ortogonal):

$f(a_{t+1,j}|\mathbf{a}_{1:t}) \approx \sum_{e_{t+1,j}} m_{e_{t+1,j}}(a_{t+1,j}).f_{e_{t+1,j}}(a_{t+1,j}|\mathbf{a}_{1:t})$

e cada $f_{e_{t+1,j}}(a_{t+1,j}|\mathbf{a}_{1:t})$ sendo combinado pela defuzzificação é dado por:

$f_{e_{t+1,j}}(a_{t+1,j}|\mathbf{a}_{1:t}) = P(e_{t+1,j}|d_{1:t}) / h_{e_j}$, para $m_{e_{t+1,j}}(a_{t+1,j}) > 0$

onde $P(e_{t+1,j}|d_{1:t})$ é a probabilidade de uma observação *fuzzy* futura $e_{t+1,j}$, e h_{e_j} é a área da região *fuzzy* $e_{t+1,j}$ ($h_{e_j} = \int m_{e_{t+1,j}}(a_{t+1,j})da_{t+1,j}$). Se for feita a escolha de funções de

pertinência a fim de que a fuzzificação corresponda a uma discretização dos dados contínuos então a aproximação de $f(a_{t+1,j}|\mathbf{a}_{1:t})$ realizada pelo FHMP torna-se igual àquela efetuada pelo HMMR.

O valor esperado da variável contínua $A_{t+1,j}$ condicionada pelos valores contínuos $\mathbf{a}_{1:t}$ equivale à previsão do FHMP ($a_{FHMP\ t,j}$):

$a_{FHMP\ t,j} = \int a_{t+1,j}.f(a_{t+1,j}|\mathbf{a}_{1:t})da_{t+1,j} \approx \int a_{t+1,j}.\{\sum_{e_{t+1,j}} m_{e_{t+1,j}}(a_{t+1,j}).f_{e_{t+1,j}}(a_{t+1,j}|\mathbf{a}_{1:t})\}da_{t+1,j}$ [pela aproximação de $f(a_{t+1,j}|\mathbf{a}_{1:t})$]

$= \int a_{t+1,j}.\{\sum_{e_{t+1,j}} m_{e_{t+1,j}}(a_{t+1,j}).P(e_{t+1,j}|d_{1:t}) / h_{e_j}\}da_{t+1,j}$ [substituindo $f_{e_{t+1,j}}(a_{t+1,j}|\mathbf{a}_{1:t})$]

$= \sum_{e_{t+1,j}} \int \{a_{t+1,j}.m_{e_{t+1,j}}(a_{t+1,j}).P(e_{t+1,j}|d_{1:t}) / h_{e_j}\}da_{t+1,j}$ [dividindo em várias integrais]

$= \sum_{e_{t+1,j}} \{P(e_{t+1,j}|d_{1:t}) / h_{e_j}\}\int \{a_{t+1,j}.m_{e_{t+1,j}}(a_{t+1,j})\}da_{t+1,j}$ [extraindo constantes das integrais]

$= \sum_{e_{t+1,j}} \{P(e_{t+1,j}|d_{1:t}) / h_{e_j}\}.\ \overline{e}_{t+1,j}\ .\ h_{e_j}$ [usando $\overline{e}_{t+1,j} = \int a_{t+1,j}.m_{e_{t+1,j}}(a_{t+1,j})da_{t+1,j} / h_{e_j}$]

$= \sum_{e_{t+1,j}} \overline{e}_{t+1,j}\ .\ P(e_{t+1,j}|d_{1:t})$

que corresponde à previsão contínua do FHMP apresentada inicialmente no Capítulo 5. Essa previsão torna-se igual a do HMMR quando as funções de pertinência são selecionadas de forma que a fuzzificação se iguale a uma discretização dos dados contínuos.

A.7 - *Fuzzy Markov Predictor*

A função densidade de probabilidade sendo aproximada pelo *Fuzzy Markov Predictor* (FMP) é igual àquela estimada pelo MMR, ou seja, a densidade da filtragem da variável de estado:

se $t = 0$ então $f(v_t|\mathbf{a}_{1:t}) = f(v_t)$

se $t > 0$ então $f(v_t|\mathbf{a}_{1:t}) = \beta.f(\mathbf{a}_t|v_t).f(v_t|\mathbf{a}_{1:t-1}) = \beta.\{\prod_j f(a_{t,j}|v_t)\}.f(v_t|\mathbf{a}_{1:t-1})$

com a constante de normalização $\beta = 1 / \int \{f(\mathbf{a}_t|v_t).f(v_t|\mathbf{a}_{1:t-1})\}dv_t$.

Essa densidade é aproximada pela seguinte fórmula de defuzzificação:

$f(v_t|\mathbf{a}_{1:t}) \approx (\sum_{s_t} m_{s_t}(v_t).f_{s_t}(v_t|\mathbf{a}_{1:t})) / (\sum_{s_t} m_{s_t}(v_t))$

que é simplificada para (em um sistema de informação *fuzzy* ortogonal):

$f(v_t|\mathbf{a}_{1:t}) \approx \sum_{s_t} m_{s_t}(v_t).f_{s_t}(v_t|\mathbf{a}_{1:t})$

e cada $f_{s_t}(v_t|\mathbf{a}_{1:t})$ sendo combinado é dado por:

$f_{s_t}(v_t|\mathbf{a}_{1:t}) = P(s_t|d_{1:t}) / h_s$, para $m_{s_t}(v_t) > 0$

onde $P(s_t|d_{1:t})$ é a probabilidade da filtragem do estado *fuzzy* s_t, e h_s é a área da região *fuzzy* s_t ($h_s = \int m_{s_t}(v_t)dv_t$). Através de uma escolha adequada das funções de pertinência, a fuzzificação pode corresponder a uma discretização dos dados contínuos e, assim, a aproximação de $f(v_t|\mathbf{a}_{1:t})$ feita pelo FMP torna-se igual àquela do MMR.

A previsão do FMP ($v_{FMP\,t}$) é o valor esperado da variável contínua V_t condicionada pelos valores contínuos $\mathbf{a}_{1:t}$:

$v_{FMP\,t} = \int v_t.f(v_t|\mathbf{a}_{1:t})dv_t \approx \int v_t.\{\sum_{s_t} m_{s_t}(v_t).f_{s_t}(v_t|\mathbf{a}_{1:t})\}dv_t$ [pela aproximação de $f(v_t|\mathbf{a}_{1:t})$]

$= \int v_t.\{\sum_{s_t} m_{s_t}(v_t).P(s_t|d_{1:t}) / h_s\}dv_t$ [substituindo $f_{s_t}(v_t|\mathbf{a}_{1:t})$]

$= \sum_{s_t} \int\{v_t.m_{s_t}(v_t).P(s_t|d_{1:t}) / h_s\}dv_t$ [dividindo em várias integrais]

$= \sum_{s_t} \{P(s_t|d_{1:t}) / h_s\}\int\{v_t.m_{s_t}(v_t)\}dv_t$ [extraindo constantes das integrais]

$= \sum_{s_t} \{P(s_t|d_{1:t}) / h_s\}.\ \overline{s_t}\ .\ h_s$ [usando $\overline{s_t} = \int v_t.m_{s_t}(v_t)dv_t / h_s$]

$= \sum_{s_t}\ \overline{s_t}\ .\ P(s_t|d_{1:t})$

que é o mesmo resultado da previsão contínua do FMP já vista no Capítulo 5. Essa previsão e a do MMR são iguais quando as funções de pertinência são ajustadas para que a fuzzificação seja uma discretização dos dados contínuos.

A.8 - *Fuzzy Multi-Hidden Markov Predictor*

O *Fuzzy Multi-Hidden Markov Predictor* (FMHMP) está aproximando a mesma função densidade de probabilidade que o FHMP, ou seja, a densidade da previsão da variável de evidência:

$$f(a_{t+1,j} \mid \mathbf{a}_{1:t}) = \int\{f(a_{t+1,j} \mid \mathbf{v}_{t+1}).f(\mathbf{v}_{t+1} \mid \mathbf{a}_{1:t})\}d\mathbf{v}_{t+1}$$

e usa a mesma fórmula de defuzzificação para essa aproximação (considerando um sistema de informação *fuzzy* ortogonal):

$$f(a_{t+1,j}|\mathbf{a}_{1:t}) \approx \sum_{e_{t+1,j}} m_{e_{t+1,j}}(a_{t+1,j}).f_{e_{t+1,j}}(a_{t+1,j}|\mathbf{a}_{1:t})$$

com

$$f_{e_{t+1,j}}(a_{t+1,j}|\mathbf{a}_{1:t}) = P(e_{t+1,j}|d_{1:t}) / h_{e_j}, \text{ para } m_{e_{t+1,j}}(a_{t+1,j}) > 0$$

onde o cálculo de $P(e_{t+1,j}|d_{1:t})$ utiliza um conjunto de variáveis de estado (ao invés de uma única variável de estado como no FHMP).

Através dessa aproximação pode-se provar que a previsão do FMHMP ($a_{FMHMP\ t,j}$) dada pelo valor esperado da variável $A_{t+1,j}$ condicionada por $\mathbf{a}_{1:t}$ é igual à previsão contínua do FMHMP já apresentada no Capítulo 5:

$a_{FMHMP\ t,j} = \int a_{t+1,j}.f(a_{t+1,j}|\mathbf{a}_{1:t})da_{t+1,j} \approx \int a_{t+1,j}.\{\sum_{e_{t+1,j}} m_{e_{t+1,j}}(a_{t+1,j}).f_{e_{t+1,j}}(a_{t+1,j}|\mathbf{a}_{1:t})\}da_{t+1,j} =$

$= \sum_{e_{t+1,j}} \overline{e}_{t+1,j} \,.\, P(e_{t+1,j}|d_{1:t}).$

Tanto essa previsão quanto a aproximação de $f(a_{t+1,j}|\mathbf{a}_{1:t})$ realizadas pelo FMHMP tornam-se idênticas àquelas do MHMMR quando as funções de pertinência são escolhidas de forma que a fuzzificação se iguale a uma discretização.

Apêndice B

Este apêndice apresenta as tabelas com os erros e as tabelas com os números de atributos e partições utilizados para as previsões *single-step* e *multi-step* mencionadas no Capítulo 7.

B.1 - Tabelas para as Previsões *Single-Step*

Tabela 1: Erros para distribuição uniforme das partições na série 1

	2000	2001	2002	2000-2002
FBP	2.24	3.61	3.50	3.11
FMP	2.38	3.65	2.59	2.87
FHMP	2.08	3.83	2.30	2.74
F2HMP	2.19	3.73	2.32	2.75
NBR	2.86	4.70	2.48	3.35
MMR	2.88	5.92	3.71	4.17
HMMR	2.45	3.84	2.20	2.83
2HMMR	2.01	4.16	2.08	2.75
FBP1e	2.18	3.67	2.24	2.70
FMP1e	2.22	3.71	2.23	2.72
FHMP1e	2.15	3.83	2.29	2.75
F2HMP1e	2.17	3.91	2.30	2.79
F3HMP1e	2.13	3.88	2.32	2.78
NBR1e	2.23	3.68	2.53	2.81
MMR1e	2.32	4.37	1.81	2.83
HMMR1e	2.20	4.19	2.27	2.89
2HMMR1e	2.06	4.08	2.36	2.83
3HMMR1e	2.10	4.09	2.36	2.85

Tabela 2: Erros para partições distribuídas através de DT na série 1

	2000	2001	2002	2000-2002
FBP	2.35	3.77	3.80	3.31
FMP	2.39	3.79	3.79	3.32
FHMP	2.11	3.99	2.81	2.97
F2HMP	2.03	4.04	2.81	2.96
F3HMP	2.12	4.01	2.81	2.98
NBR	2.64	3.91	3.17	3.24
MMR	2.89	3.82	4.26	3.66
HMMR	1.93	4.26	2.79	2.99
2HMMR	1.93	4.28	2.79	3.00
3HMMR	1.95	4.29	2.79	3.01
FBP1e	2.25	3.80	2.54	2.86
FMP1e	2.30	3.84	2.48	2.87
FHMP1e	2.17	3.99	2.81	2.99
F2HMP1e	2.16	3.99	2.81	2.99
F3HMP1e	2.16	3.99	2.81	2.99
NBR1e	2.56	3.91	3.03	3.17
MMR1e	2.80	3.67	3.04	3.17
HMMR1e	2.06	4.13	2.77	2.99
2HMMR1e	1.96	4.13	2.77	2.95
3HMMR1e	2.09	4.17	2.64	2.97

Tabela 3: Erros para partições obtidas pelo KM na série 1

	2000	2001	2002	2000-2002
FBP	2.23	4.53	4.77	3.84
FMP	2.25	4.59	4.83	3.89
FHMP	2.00	4.24	2.67	2.97
F2HMP	2.11	4.19	2.70	3.00
NBR	2.54	5.80	2.12	3.49
MMR	2.70	5.68	2.25	3.54
HMMR	2.02	3.91	2.41	2.78
2HMMR	2.01	3.81	2.08	2.63
FBP1e	2.38	3.65	2.56	2.86
FMP1e	2.38	3.94	2.62	2.98
FHMP1e	2.31	4.26	2.74	3.10
F2HMP1e	2.11	4.19	2.70	3.00
F3HMP1e	2.01	4.22	2.62	2.95
NBR1e	2.44	4.17	2.32	2.98
MMR1e	2.45	4.21	2.23	2.96
HMMR1e	2.17	4.05	2.91	3.04
2HMMR1e	2.12	4.20	2.31	2.88
3HMMR1e	2.31	4.24	2.42	2.99

Tabela 4: Erros para partições obtidas por DT e KM na série 1

	2000	2001	2002	2000-2002
FBP	2.63	3.82	2.33	2.93
FMP	2.96	4.90	3.06	3.64
FHMP	2.26	3.95	2.51	2.91
F2HMP	2.05	4.02	2.38	2.82
F3HMP	2.00	4.01	2.43	2.81
NBR	2.46	4.09	2.46	3.00
MMR	2.67	4.60	4.16	3.81
HMMR	2.14	3.57	1.74	**2.48**
2HMMR	2.17	4.45	1.70	2.77
3HMMR	2.17	4.82	**1.33**	2.78
FBP1e	2.23	3.68	2.35	2.75
FMP1e	2.25	4.05	2.52	2.94
FHMP1e	2.26	3.95	2.51	2.91
F2HMP1e	2.26	3.96	2.52	2.91
F3HMP1e	2.25	3.96	2.52	2.91
NBR1e	2.28	**3.46**	2.01	2.58
MMR1e	2.50	3.68	2.26	2.82
HMMR1e	2.23	3.70	1.78	2.57
2HMMR1e	2.23	4.18	1.53	2.65
3HMMR1e	2.17	4.44	1.47	2.69

Tabela 5: Erros para modelos de filtro de Kalman e métodos tradicionais na série 1

	2000	2001	2002	2000-2002
STAMP	1.77	6.11	1.47	3.12
BATS	**1.74**	12.04	8.07	7.28
Box-Jenkins	2.19	5.44	2.24	3.29
Winters	1.80	4.56	2.01	2.79

Tabela 6: Erros para distribuição uniforme das partições na série 2

	2000	2001	2002	2000-2002
FBP	3.67	5.41	5.83	4.97
FMP	3.54	5.73	5.47	4.91
FHMP	3.62	5.31	5.61	4.85
F2HMP	3.93	6.30	5.83	5.35
NBR	3.77	5.81	6.76	5.45
MMR	3.47	6.59	7.11	5.72
HMMR	3.67	6.22	6.50	5.46
2HMMR	3.69	5.76	7.05	5.50
FBP1e	3.56	6.55	6.67	5.59
FMP1e	3.60	6.25	6.36	5.40
FHMP1e	3.54	5.96	6.06	5.19
F2HMP1e	3.68	5.67	6.24	5.20
F3HMP1e	3.42	5.40	6.24	5.02
NBR1e	3.77	7.40	7.14	6.10
MMR1e	3.50	6.42	7.19	5.70
HMMR1e	3.67	6.22	6.50	5.46
2HMMR1e	3.69	5.76	7.05	5.50
3HMMR1e	3.71	5.69	7.18	5.53

Tabela 7: Erros para partições distribuídas através de DT na série 2

	2000	2001	2002	2000-2002
FBP	3.69	5.33	6.38	5.13
FMP	3.60	5.58	6.47	5.22
FHMP	3.42	6.09	6.20	5.23
F2HMP	3.43	5.76	5.70	4.96
F3HMP	3.45	5.53	5.79	4.92
NBR	3.71	6.34	6.75	5.60
MMR	3.47	6.59	7.11	5.72
HMMR	3.67	6.22	6.50	5.46
2HMMR	3.69	5.76	7.05	5.50
3HMMR	3.71	5.69	7.18	5.53
FBP1e	3.56	6.55	6.67	5.59
FMP1e	3.06	6.24	6.53	5.28
FHMP1e	3.64	6.92	6.87	5.81
F2HMP1e	3.54	6.84	6.82	5.74
F3HMP1e	3.46	6.63	6.65	5.58
NBR1e	3.65	6.84	6.62	5.70
MMR1e	3.01	7.17	7.21	5.80
HMMR1e	3.67	6.22	6.50	5.46
2HMMR1e	3.69	5.76	7.05	5.50
3HMMR1e	3.71	5.69	7.18	5.53

Tabela 8: Erros para partições obtidas pelo KM na série 2

	2000	2001	2002	2000-2002
FBP	3.71	5.43	6.33	5.16
FMP	3.92	5.25	6.65	5.27
FHMP	3.82	4.95	**4.86**	**4.54**
F2HMP	3.30	4.75	5.57	**4.54**
NBR	4.92	4.59	6.34	5.28
MMR	5.04	4.88	6.59	5.50
HMMR	3.64	5.92	5.10	4.89
2HMMR	3.65	5.81	5.79	5.08
FBP1e	3.62	5.95	6.95	5.51
FMP1e	3.55	7.15	6.64	5.78
FHMP1e	3.23	4.87	5.69	4.60
F2HMP1e	3.30	4.75	5.57	**4.54**
F3HMP1e	3.67	4.73	5.46	4.62
NBR1e	3.53	5.93	7.43	5.63
MMR1e	3.61	5.90	7.22	5.58
HMMR1e	3.64	5.92	5.10	4.89
2HMMR1e	3.65	5.81	5.79	5.08
3HMMR1e	3.65	5.87	5.81	5.11

Tabela 9: Erros para partições obtidas por DT e KM na série 2

	2000	2001	2002	2000-2002
FBP	4.58	7.40	7.43	6.47
FMP	4.35	7.57	7.25	6.39
FHMP	3.39	5.90	5.94	5.08
F2HMP	3.69	4.97	5.65	4.77
F3HMP	3.63	**4.56**	5.60	4.59
NBR	4.20	5.64	6.24	5.36
MMR	4.03	5.73	6.14	5.30
HMMR	3.68	5.76	5.44	4.96
2HMMR	3.68	5.42	5.22	4.77
3HMMR	3.68	4.86	5.80	4.78
FBP1e	3.55	6.10	6.95	5.53
FMP1e	**2.74**	6.57	7.41	5.57
FHMP1e	3.86	5.30	5.69	4.95
F2HMP1e	3.69	4.97	5.65	4.77
F3HMP1e	3.63	**4.56**	5.60	4.59
NBR1e	3.92	7.27	6.63	5.94
MMR1e	3.54	7.11	6.66	5.77
HMMR1e	3.68	5.76	5.44	4.96
2HMMR1e	3.68	5.42	5.22	4.77
3HMMR1e	3.68	4.86	5.80	4.78

Tabela 10: Erros para modelos de filtro de Kalman e métodos tradicionais na série 2

	2000	2001	2002	2000-2002
STAMP	3.08	5.85	5.01	4.65
BATS	3.23	13.04	10.54	8.93
Box-Jenkins	2.94	4.95	5.86	4.58
Winters	3.11	9.11	5.95	6.06

Tabela 11: Erros para distribuição uniforme das partições na série 3

	2000	2001	2002	2000-2002
FBP	1.83	**3.79**	4.02	3.21
FMP	1.76	4.03	4.29	3.36
FHMP	1.62	4.25	3.21	3.03
F2HMP	1.48	4.43	3.43	3.11
NBR	2.02	8.20	6.29	5.50
MMR	1.48	4.21	3.91	3.20
HMMR	1.66	4.29	3.31	3.09
2HMMR	1.80	4.36	3.47	3.21
FBP1e	1.49	4.36	3.87	3.24
FMP1e	1.52	4.98	4.33	3.61
FHMP1e	1.63	4.43	3.37	3.15
F2HMP1e	1.59	4.29	3.31	3.07
F3HMP1e	1.61	4.29	3.32	3.08
NBR1e	1.69	4.24	3.15	3.03
MMR1e	1.61	4.59	3.34	3.18
HMMR1e	1.66	4.29	3.31	3.09
2HMMR1e	1.80	4.36	3.47	3.21
3HMMR1e	1.43	4.04	3.18	2.88

Tabela 12: Erros para partições distribuídas através de DT na série 3

	2000	2001	2002	2000-2002
FBP	1.67	4.39	3.74	3.27
FMP	1.80	4.31	3.95	3.36
FHMP	1.52	4.35	3.31	3.06
F2HMP	1.50	4.36	3.31	3.06
F3HMP	1.47	4.37	3.29	3.04
NBR	1.77	4.80	3.78	3.45
MMR	1.76	4.78	3.96	3.50
HMMR	1.53	4.14	2.80	**2.82**
2HMMR	1.57	4.34	2.76	2.89
3HMMR	1.64	4.53	3.29	3.15
FBP1e	1.70	5.40	3.57	3.56
FMP1e	1.75	5.48	3.65	3.63
FHMP1e	1.69	4.48	3.34	3.17
F2HMP1e	1.68	4.47	3.34	3.16
F3HMP1e	1.68	4.46	3.34	3.16
NBR1e	1.69	5.16	3.16	3.34
MMR1e	1.61	5.20	3.19	3.33
HMMR1e	1.64	4.27	2.98	2.97
2HMMR1e	1.63	4.26	2.99	2.96
3HMMR1e	1.62	4.48	2.82	2.97

Tabela 13: Erros para partições obtidas pelo KM na série 3

	2000	2001	2002	2000-2002
FBP	1.99	4.30	5.87	4.05
FMP	2.44	4.53	5.92	4.30
FHMP	1.30	4.49	3.23	3.01
F2HMP	2.17	4.72	3.17	3.35
NBR	1.60	4.49	4.17	3.42
MMR	2.19	5.39	5.23	4.27
HMMR	1.87	5.06	3.11	3.35
2HMMR	1.59	4.74	3.16	3.16
FBP1e	1.71	4.90	4.64	3.75
FMP1e	1.80	4.53	4.59	3.64
FHMP1e	1.49	4.35	3.39	3.07
F2HMP1e	1.61	4.11	3.32	3.02
F3HMP1e	1.47	4.24	3.50	3.07
NBR1e	1.88	5.07	3.69	3.54
MMR1e	1.77	5.03	4.15	3.65
HMMR1e	1.87	4.28	2.91	3.02
2HMMR1e	1.59	4.74	3.16	3.16
3HMMR1e	1.59	4.97	3.09	3.22

Tabela 14: Erros para partições obtidas por DT e KM na série 3

	2000	2001	2002	2000-2002
FBP	2.20	4.01	4.63	3.61
FMP	2.19	4.03	5.04	3.75
FHMP	1.45	4.44	3.17	3.02
F2HMP	1.63	4.32	3.17	3.04
F3HMP	1.75	4.26	3.14	3.05
NBR	1.68	4.45	4.13	3.42
MMR	1.53	4.25	4.50	3.43
HMMR	1.65	4.22	2.98	2.95
2HMMR	1.64	4.12	2.89	2.88
3HMMR	1.60	4.60	3.05	3.09
FBP1e	1.62	4.74	3.76	3.38
FMP1e	1.60	4.75	3.93	3.42
FHMP1e	1.54	4.34	3.31	3.06
F2HMP1e	1.61	4.33	3.28	3.07
F3HMP1e	1.77	4.31	3.26	3.11
NBR1e	1.53	5.13	3.55	3.40
MMR1e	1.51	5.22	4.04	3.59
HMMR1e	1.64	4.14	3.05	2.94
2HMMR1e	1.64	4.12	2.89	2.88
3HMMR1e	1.64	4.34	**2.74**	2.91

Tabela 15: Erros para modelos de filtro de Kalman e métodos tradicionais na série 3

	2000	2001	2002	2000-2002
STAMP	1.43	6.05	3.14	3.54
BATS	1.82	11.53	9.08	7.48
Box-Jenkins	1.74	3.92	5.15	3.60
Winters	**1.34**	5.18	3.47	3.33

Tabela 16: Números de atributos / partições para distribuição uniforme das partições

	Série 1	Série 2	Série 3
FBP	4 / 3	6 / 9	4 / 6
FMP	4 / 4	6 / 9	3 / 6
FHMP	2 / 7	3 / 14	1 / 14
F2HMP	3 / 9	3 / 10	5 / 8
NBR	5 / 5	6 / 10	6 / 7
MMR	6 / 6	6 / 4	2 / 3
HMMR	3 / 5	1 / 4	1 / 10
2HMMR	2 / 6	1 / 4	1 / 6
FBP1e	1 / 3	1 / 5	1 / 8
FMP1e	1 / 3	1 / 3	1 / 8
FHMP1e	1 / 7	1 / 9	1 / 10
F2HMP1e	1 / 8	1 / 9	1 / 10
F3HMP1e	1 / 6	1 / 9	1 / 8
NBR1e	1 / 3	1 / 3	1 / 4
MMR1e	1 / 5	1 / 3	1 / 4
HMMR1e	1 / 10	1 / 4	1 / 10
2HMMR1e	1 / 10	1 / 4	1 / 6
3HMMR1e	1 / 10	1 / 4	1 / 7

Tabela 17: Números de atributos / partições para partições distribuídas por DT

	Série 1	Série 2	Série 3
FBP	3 / 4	6 / 5	3 / 4
FMP	3 / 4	6 / 5	3 / 4
FHMP	2 / 4	6 / 5	6 / 4
F2HMP	6 / 4	6 / 5	6 / 4
F3HMP	4 / 4	6 / 5	6 / 4
NBR	2 / 3	6 / 4	3 / 3
MMR	3 / 3	6 / 4	3 / 3
HMMR	3 / 3	1 / 4	2 / 3
2HMMR	3 / 3	1 / 4	3 / 3
3HMMR	3 / 3	1 / 4	6 / 3
FBP1e	1 / 4	1 / 5	1 / 4
FMP1e	1 / 4	1 / 5	1 / 4
FHMP1e	1 / 4	1 / 5	1 / 4
F2HMP1e	1 / 4	1 / 5	1 / 4
F3HMP1e	1 / 4	1 / 5	1 / 4
NBR1e	1 / 3	1 / 4	1 / 3
MMR1e	1 / 3	1 / 4	1 / 3
HMMR1e	1 / 3	1 / 4	1 / 3
2HMMR1e	1 / 3	1 / 4	1 / 3
3HMMR1e	1 / 3	1 / 4	1 / 3

Tabela 18: Números de atributos / partições para partições obtidas pelo KM

	Série 1	Série 2	Série 3
FBP	4 / 3	6 / 3	3 / 6
FMP	4 / 3	6 / 3	3 / 9
FHMP	1 / 12	1 / 14	2 / 11
F2HMP	1 / 10	1 / 10	4 / 9
NBR	5 / 13	6 / 4	2 / 3
MMR	5 / 13	6 / 4	2 / 13
HMMR	4 / 4	1 / 5	2 / 12
2HMMR	4 / 4	1 / 5	1 / 6
FBP1e	1 / 5	1 / 6	1 / 9
FMP1e	1 / 5	1 / 3	1 / 9
FHMP1e	1 / 10	1 / 10	1 / 9
F2HMP1e	1 / 10	1 / 10	1 / 10
F3HMP1e	1 / 10	1 / 9	1 / 9
NBR1e	1 / 3	1 / 3	1 / 3
MMR1e	1 / 3	1 / 3	1 / 3
HMMR1e	1 / 3	1 / 5	1 / 3
2HMMR1e	1 / 9	1 / 5	1 / 6
3HMMR1e	1 / 9	1 / 5	1 / 6

Tabela 19: Números de atributos / partições para partições obtidas por DT e KM

	Série 1	Série 2	Série 3
FBP	2 / 4	6 / 5	4 / 4
FMP	3 / 4	6 / 5	4 / 4
FHMP	1 / 4	4 / 5	6 / 4
F2HMP	3 / 4	1 / 5	4 / 4
F3HMP	3 / 4	1 / 5	4 / 4
NBR	2 / 3	6 / 4	3 / 3
MMR	3 / 3	6 / 4	3 / 3
HMMR	3 / 3	1 / 4	5 / 3
2HMMR	3 / 3	1 / 4	1 / 3
3HMMR	3 / 3	1 / 4	6 / 3
FBP1e	1 / 4	1 / 5	1 / 4
FMP1e	1 / 4	1 / 5	1 / 4
FHMP1e	1 / 4	1 / 5	1 / 4
F2HMP1e	1 / 4	1 / 5	1 / 4
F3HMP1e	1 / 4	1 / 5	1 / 4
NBR1e	1 / 3	1 / 4	1 / 3
MMR1e	1 / 3	1 / 4	1 / 3
HMMR1e	1 / 3	1 / 4	1 / 3
2HMMR1e	1 / 3	1 / 4	1 / 3
3HMMR1e	1 / 3	1 / 4	1 / 3

B.2 - Tabelas para as Previsões *Multi-Step*

Tabela 20: Erros para partições distribuídas através de DT na série 1

	2000	2001	2002	2000-2002
FHMP1e	4.07	16.61	10.73	10.47
F2HMP1e	3.87	16.65	10.68	10.40
F3HMP1e	3.85	16.69	10.82	10.45
aF2HMP1e	3.92	16.62	10.63	10.39
aF3HMP1e	3.59	16.88	11.18	10.55
HMMR1e	2.01	17.28	11.51	10.26
2HMMR1e	2.01	17.28	11.52	10.27
3HMMR1e	2.01	17.40	11.83	10.41
a2HMMR1e	2.54	**16.11**	**7.60**	**8.75**
a3HMMR1e	1.99	18.35	12.41	10.92

Tabela 21: Erros para partições obtidas por DT e KM na série 1

	2000	2001	2002	2000-2002
FHMP1e	1.76	20.01	19.25	13.67
F2HMP1e	1.72	19.88	18.73	13.45
F3HMP1e	1.76	22.08	22.66	15.50
aF2HMP1e	1.73	19.92	18.80	13.48
aF3HMP1e	1.75	17.90	13.10	10.91
HMMR1e	2.67	16.59	8.92	9.39
2HMMR1e	2.67	16.59	8.92	9.39
3HMMR1e	2.67	16.59	8.92	9.39
a2HMMR1e	2.20	16.84	9.99	9.67
a3HMMR1e	2.26	16.50	8.54	9.10

Tabela 22: Erros para modelo de filtro de Kalman e métodos tradicionais na série 1

	2000	2001	2002	2000-2002
BATS	3.87	21.42	17.89	14.39
Box-Jenkins	**1.18**	18.54	12.92	10.88
Winters	2.08	16.84	9.43	9.45

Tabela 23: Erros para partições distribuídas através de DT na série 2

	2000	2001	2002	2000-2002
FHMP1e	6.49	26.10	20.17	17.59
F2HMP1e	6.16	25.60	19.44	17.07
F3HMP1e	4.67	24.67	21.44	16.93
aF2HMP1e	6.25	25.75	19.68	17.23
aF3HMP1e	4.63	24.32	20.32	16.42
HMMR1e	5.09	22.33	14.89	14.11
2HMMR1e	5.09	22.34	14.90	14.11
3HMMR1e	5.08	22.35	14.94	14.12
a2HMMR1e	4.63	20.55	11.58	12.25
a3HMMR1e	4.50	21.28	14.87	13.55

Tabela 24: Erros para partições obtidas por DT e KM na série 2

	2000	2001	2002	2000-2002
FHMP1e	5.09	24.86	19.74	16.56
F2HMP1e	4.74	23.01	16.14	14.63
F3HMP1e	4.63	22.27	15.01	13.97
aF2HMP1e	4.75	23.25	16.60	14.87
aF3HMP1e	4.48	22.73	16.10	14.44
HMMR1e	4.00	19.11	9.03	10.71
2HMMR1e	4.00	19.11	9.03	10.71
3HMMR1e	4.00	19.12	9.04	10.72
a2HMMR1e	3.85	15.20	**7.29**	8.78
a3HMMR1e	3.43	**14.13**	8.44	**8.67**

Tabela 25: Erros para modelo de filtro de Kalman e métodos tradicionais na série 2

	2000	2001	2002	2000-2002
BATS	10.53	23.86	15.79	16.73
Box-Jenkins	3.05	19.54	16.31	12.96
Winters	**3.01**	19.18	16.19	12.79

Tabela 26: Erros para partições distribuídas através de DT na série 3

	2000	2001	2002	2000-2002
FHMP1e	3.06	18.31	11.20	10.86
F2HMP1e	3.06	18.31	11.20	10.86
F3HMP1e	3.06	18.31	11.20	10.86
aF2HMP1e	3.06	18.31	11.19	10.85
aF3HMP1e	3.05	18.29	11.16	10.83
HMMR1e	2.71	18.62	12.22	11.18
2HMMR1e	2.71	18.62	12.22	11.18
3HMMR1e	2.70	18.64	12.27	11.20
a2HMMR1e	2.89	18.47	12.52	11.29
a3HMMR1e	2.92	18.19	11.99	11.03

Tabela 27: Erros para partições obtidas por DT e KM na série 3

	2000	2001	2002	2000-2002
FHMP1e	1.86	20.27	16.38	12.84
F2HMP1e	1.82	20.35	16.63	12.93
F3HMP1e	**1.79**	20.40	16.80	13.00
aF2HMP1e	1.83	20.33	16.54	12.90
aF3HMP1e	1.80	20.36	16.75	12.97
HMMR1e	2.51	18.77	12.62	11.30
2HMMR1e	2.51	18.77	12.62	11.30
3HMMR1e	2.51	18.77	12.63	11.30
a2HMMR1e	2.72	18.28	11.62	10.88
a3HMMR1e	2.66	18.30	11.77	10.91

Tabela 28: Erros para modelo de filtro de Kalman e métodos tradicionais na série 3

	2000	2001	2002	2000-2002
BATS	3.28	**16.88**	**7.67**	9.27
Box-Jenkins	2.33	17.04	8.18	**9.18**
Winters	2.41	17.29	9.63	9.78

Tabela 29: Números de atributos / partições para partições distribuídas por DT

	Série 1	Série 2	Série 3
FHMP1e	1 / 5	1 / 5	1 / 5
F2HMP1e	1 / 5	1 / 5	1 / 5
F3HMP1e	1 / 5	1 / 5	1 / 5
aF2HMP1e	1 / 5	1 / 5	1 / 5
aF3HMP1e	1 / 5	1 / 5	1 / 5
HMMR1e	1 / 4	1 / 4	1 / 4
2HMMR1e	1 / 4	1 / 4	1 / 4
3HMMR1e	1 / 4	1 / 4	1 / 4
a2HMMR1e	1 / 4	1 / 4	1 / 4
a3HMMR1e	1 / 4	1 / 4	1 / 4

Tabela 30: Números de atributos / partições para partições obtidas por DT e KM

	Série 1	Série 2	Série 3
FHMP1e	1 / 5	1 / 5	1 / 5
F2HMP1e	1 / 5	1 / 5	1 / 5
F3HMP1e	1 / 5	1 / 5	1 / 5
aF2HMP1e	1 / 5	1 / 5	1 / 5
aF3HMP1e	1 / 5	1 / 5	1 / 5
HMMR1e	1 / 4	1 / 4	1 / 4
2HMMR1e	1 / 4	1 / 4	1 / 4
3HMMR1e	1 / 4	1 / 4	1 / 4
a2HMMR1e	1 / 4	1 / 4	1 / 4
a3HMMR1e	1 / 4	1 / 4	1 / 4