

MODELO ADAPTATIVO BASEADO EM REGRAS NEBULOSAS APLICADO À PREVISÃO DE SÉRIES TEMPORAIS

Ivette Luna¹, Secundino Soares², Rosângela Ballini³

^{1,2} Departamento de Engenharia de Sistemas – DENSIS
Faculdade de Engenharia Elétrica e de Computação – FEEC
Universidade Estadual de Campinas – UNICAMP
13083-970 Campinas – SP – Brasil
E-mail: iluna@cose.fee.unicamp.br; dino@cose.fee.unicamp.br

³ Departamento de Teoria Econômica – DTE
Instituto de Economia – IE
Universidade Estadual de Campinas – UNICAMP
13083-857 Campinas – SP – Brasil
E-mail: ballini@eco.unicamp.br

Resumo – Este trabalho propõe um modelo de aprendizado *online* baseado em regras nebulosas do tipo Takagi-Sugeno, para modelos de séries temporais. A estrutura do modelo é construída em duas etapas. Na primeira etapa, o modelo é inicializado utilizando um algoritmo de aprendizado construtivo *offline*, composto por duas regras nebulosas. Na segunda etapa, o modelo é modificado de forma dinâmica, utilizando um algoritmo de aprendizado recursivo baseado no algoritmo de maximização de verossimilhança e em operadores de adição e punição de regras. O aprendizado adaptativo reduz a complexidade do modelo, e define de forma automática a sua estrutura. A proposta é aplicada na construção de um modelo de previsão de vazões semanais e os resultados são comparados com o modelo evolutivo Takagi-Sugeno. O modelo é também utilizado na previsão até dezoito passos à frente de uma série financeira. Os resultados obtidos mostram que a proposta é uma alternativa eficiente para modelos de séries temporais, proporcionando uma estrutura simples e parcimoniosa.

Palavras Chaves – Previsão de séries temporais, algoritmo EM, aprendizado adaptativo.

1. Introdução

Redes neurais e sistemas nebulosos têm se mostrado eficientes em diversas aplicações, destacando-se os modelos baseados em regras nebulosas, devido a sua simplicidade e eficiência [1]. Existem diversas abordagens para a construção de sistemas baseados em regras. Uma técnica bastante divulgada é a utilização de algoritmos de agrupamento, via a partição nebulosa do espaço de entrada em M sub-regiões, como o algoritmo FCM [2] e o algoritmo da montanha [3]. Por outro lado, existem abordagens baseadas na alocação de modelos locais ou especialistas, as quais estão diretamente vinculadas ao princípio de dividir e conquistar, como é o caso dos modelos de mistura de especialistas [4].

Devido à complexidade dos problemas reais, os quais são em geral, de natureza não linear, torna-se impossível saber com antecedência o número adequado de partições ou especialistas necessários para a construção dos modelos. Como consequência, abordagens de aprendizado online tem sido desenvolvidas, fornecendo assim, estratégias de aprendizado com alta capacidade de adaptação a baixo custo computacional.

Seguindo esta direção, diversas técnicas de aprendizado adaptativas e não supervisionadas de modelos de redes neurais e de sistemas baseados em regras nebulosas vêm sendo desenvolvidas nos últimos anos. Os trabalhos descritos em [5], [6], [1], e [7] são apenas alguns exemplos deste tipo de aprendizado, sendo que, estas propostas consideram tanto a adaptação dos parâmetros como da estrutura do modelo.

Este artigo propõe um sistema adaptativo baseado em regras nebulosas (**A-FSM**), sendo o algoritmo de aprendizado baseado no algoritmo de maximização da verossimilhança (EM) [4]. A vantagem de adotar técnicas de aprendizado baseadas no algoritmo EM, é a convergência rápida do algoritmo e a obtenção de ótimos locais, quando comparado ao algoritmo de retro-propagação temporal [8].

Além disso, este trabalho sugere uma versão adaptativa do algoritmo EM tradicional, sendo este aplicado no ajuste de modelos baseados em regras nebulosas, efetuando o ajuste dos parâmetros e da estrutura do modelo à medida que os dados estão sendo apresentados ao sistema definindo, de forma automática, o número de regras necessárias a cada iteração.

O sistema adaptativo é aplicado na previsão de uma série de vazões semanais do posto de Sobradinho. O modelo **A-FSM** é comparado com o modelo evolutivo Takagi-Sugeno (**eTS**), proposto em [1]. Os resultados mostram que a técnica de

aprendizado sugerida é eficiente e mostra-se como uma alternativa promissória para o ajuste dinâmico de modelos de séries temporais.

Este artigo está organizado como segue. A Seção 2 descreve a estrutura geral do modelo **A-FSM**. A Seção 3 detalha o algoritmo de aprendizado. Os resultados de simulação são apresentados na Seção 4. Finalmente, as conclusões são descritas na Seção 5.

2. Estrutura do Modelo Adaptativo

A estrutura geral do sistema adaptativo proposto é composta por um conjunto de M regras nebulosas de primeira ordem, do tipo Takagi-Sugeno (TS) [9], é ilustrada na Figura 1, onde $\mathbf{x}^k = [x_1^k, x_2^k, \dots, x_p^k]$ é o vetor de entrada no instante k , $k \in \mathbb{Z}_0^+$; y^k é a saída do modelo, para uma determinada entrada \mathbf{x}^k .

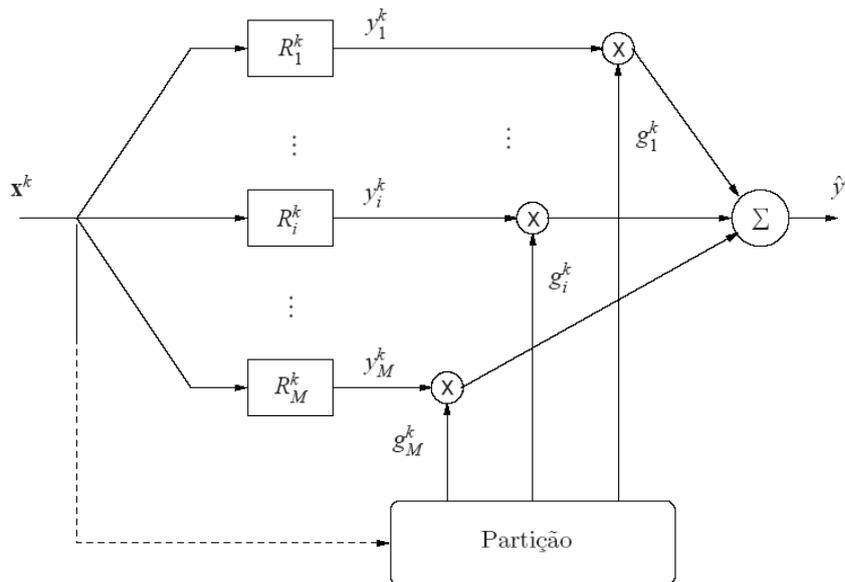


Figura 1: Estrutura do modelo proposto composto por M regras nebulosas no instante k .

A representação do sistema baseado em regras nebulosas pode ser descrita por camadas. A partição do espaço de entrada é representada pelas duas primeiras camadas. Cada sub-região gerada é definida por um centro $\mathbf{c}_i \in \mathbb{R}^p$ e por uma matriz de covariância \mathbf{V}_i de dimensão $p \times p$. Assim, cada vetor de entrada terá um grau de pertinência associado a cada uma das sub-regiões geradas pela base de regras.

A primeira camada é a responsável pelo cálculo das funções de pertinência a cada iteração k . Estas funções de pertinência são definidas pelo produto $\alpha_i P_i^k$, sendo α_i coeficientes positivos tal que $\sum_{i=1}^M \alpha_i = 1$ e $P_i^k = P[i|\mathbf{x}^k]$ definida como:

$$P_i^k = P[i|\mathbf{x}^k] = \left(\frac{1}{(2\pi)^{p/2} \det(\mathbf{V}_i)^{1/2}} \right) \times \left(\exp \left\{ -\frac{1}{2} (\mathbf{x}^k - \mathbf{c}_i) \mathbf{V}_i^{-1} (\mathbf{x}^k - \mathbf{c}_i)^T \right\} \right) \quad (1)$$

na qual \mathbf{V}_i e \mathbf{c}_i são a matriz de covariância e centro associados à i -ésima regra, respectivamente. A Eq. (1) representa a probabilidade de escolher a regra i , dado que apenas a entrada \mathbf{x}^k é conhecida, assim como os parâmetros \mathbf{V}_i e \mathbf{c}_i . Observa-se que $P[i|\mathbf{x}^k]$ é uma função de densidade de probabilidade condicional, que pode ser interpretada como a probabilidade do vetor de entrada \mathbf{x}^k pertencer à região ativa da i -ésima regra.

O grau de pertinência de cada padrão \mathbf{x}^k é representado por $g_i(\mathbf{x}^k) \in [0, 1]$, dado por:

$$g_i(\mathbf{x}^k) = g_i^k = \frac{\alpha_i P[i|\mathbf{x}^k]}{\sum_{q=1}^M \alpha_q P[q|\mathbf{x}^k]} \quad (2)$$

Estas funções de pertinência g_i^k são fornecidas pela segunda camada do modelo (Figura 1), uma vez que as funções de pertinência são normalizadas, tal que $\sum_{i=1}^M g_i^k = 1$.

Os coeficientes $\alpha_i, i = 1, \dots, M$ também podem ser interpretados como índices indiretos de relevância para cada regra gerada durante o treinamento, de tal forma que, quanto mais alto seja o valor de α_i , maior será a importância da i -ésima regra nebulosa.

A saída do sistema \hat{y}^k é calculada na última camada da estrutura do modelo, como uma combinação não linear dos conseqüentes y_i^k ponderados pelos seus respectivos graus de pertinência g_i^k , ou seja:

$$\hat{y}^k = \sum_{i=1}^M g_i^k \times y_i^k \quad (3)$$

Sendo, neste caso, a saída y_i^k de cada regra R_i^k , definida como:

$$y_i^k = \phi^k \times \theta_i^k \quad (4)$$

na qual $\phi^k = [1 \ x_1^k \ x_2^k \ \dots \ x_p^k]$ é um vetor que contém os dados de entrada \mathbf{x}^k e o termo constante; e $\theta_i^k = [\theta_{i0} \ \theta_{i1} \ \dots \ \theta_{ip}]$ é o vetor de coeficientes dos modelos locais lineares, de dimensão $1 \times (p+1)$ para $i = 1, \dots, M$, que representam os conseqüentes das regras nebulosas. Embora as Equações de (2)-(4) considerem apenas uma única saída, o modelo pode ser estendido para múltiplas saídas.

Note que a dinâmica do modelo pode ser interpretada como um mecanismo de inferência nebulosa, sendo a base de regras composta por regras $R_i^k, i = 1, \dots, M$, do tipo

$$R_i^k: \text{Se } \mathbf{x}^k \text{ pertence a } i - \text{ésima região com grau de pertinência } g_i^k \text{ Então } y_i^k = \phi^k \times \theta_i^k.$$

Por outro lado, o modelo proposto também pode ser interpretado como uma arquitetura modular, sendo que o mecanismo de inferência representa o sistema de controle para cada módulo ou especialista [10], sendo o módulo em si representado por cada um dos modelos lineares definidos.

A vantagem desta estrutura é a sua relativa simplicidade quando comparada a outras abordagens não lineares, assim como a sua capacidade de interpretação e adaptação. Além disso, esta permite uma série de variações no treinamento do modelo, obtendo ganhos tais como, a seleção automática da estrutura, um processo de aprendizado mais rápido, capacidade de adaptação, entre outras.

Diferentes trabalhos da literatura têm levantado comparações entre modelos puramente estatísticos e modelos computacionais baseados em redes neurais e/ou sistemas nebulosos, como é o caso, por exemplo, dos trabalhos descritos em [11], [12] e [13].

Ambas as abordagens são, em geral, utilizadas com o intuito de modelar dependências entre as variáveis envolvidas, sendo o modelo freqüentemente representado como segue:

$$\hat{y}^k = f[\mathbf{x}^k] = f[x_1^k, x_2^k, \dots, x_p^k]$$

onde $f[\cdot]$ é uma relação funcional entre a variável independente ou observável \mathbf{x}^k e a variável dependente y^k ; \hat{y}^k denota o valor estimado de y^k .

Desta forma, o objetivo dos modelos de regressão, assim como do modelo proposto neste trabalho, consiste na predição de uma variável dependente a partir de variáveis de entrada observáveis \mathbf{x}^k , comumente, mas não sempre, composta por atrasos das realizações da saída. O valor previsto \hat{y}^k é representado por uma relação não-linear definida pela Equação (3).

A principal diferença entre ambas as abordagens é que, o modelo de regressão é definido por um vetor de coeficientes enquanto os modelos baseados em redes neurais ou sistemas nebulosos, como o proposto neste trabalho, são definidos por um conjunto de parâmetros, incluindo centros \mathbf{c}_i , matrizes de covariância \mathbf{V}_i e os parâmetros dos modelos locais que representam os conseqüentes das regras nebulosas $\theta_i, i = 1, \dots, M$.

Portanto, enquanto os modelos lineares possuem uma forma fechada definida pelos seus coeficientes, os modelos com aprendizado construtivo ou *online* re-definem os seus parâmetros durante um processo de ajuste ou aprendizado. Apesar deste

“custo extra”, a vantagem de adotar modelos não-lineares com algoritmos de aprendizado como o proposto, é que aplicando estes algoritmos, tanto a estrutura do modelo como o conjunto de parâmetros são definidos de forma automática, diminuindo a dificuldade na escolha dos parâmetros e de uma estrutura adequada para a construção dos modelos de séries temporais.

3. Aprendizado Adaptativo

A identificação do modelo baseado em regras nebulosas é feita de tal forma que tanto a estrutura do modelo como os parâmetros sejam definidos simultaneamente durante o aprendizado. Assim, o objetivo do aprendizado adaptativo é:

- Definir o número de regras M que compõem a estrutura do modelo a cada iteração k ;
- Definir os parâmetros dos antecedentes das regras nebulosas, ou seja, os centros \mathbf{c}_i e as matrizes de covariância $\mathbf{V}_i, i = 1, \dots, M$;
- Definir os parâmetros dos conseqüentes ou modelos locais $\theta_i, i = 1, \dots, M$.

O processo de aprendizado pode ser dividido em duas etapas: inicialização do modelo e aprendizado online. Na primeira etapa, um método de aprendizado *offline* construtivo é usado para um ajuste inicial de uma base de regras composta por duas regras. Para o ajuste dessa base de regras é utilizada uma pequena parte do histórico da série temporal. Essa inicialização se faz necessário para que seja definido um ótimo local factível. Na segunda fase, a estrutura inicial e os parâmetros do modelo são ajustados de forma dinâmica conforme os dados são apresentados ao modelo via um método de aprendizado online. A seguir, cada uma dessas etapas é detalhada.

3.1 Inicialização

A inicialização do modelo utiliza um aprendizado construtivo baseado no algoritmo EM [4]. Este algoritmo é composto por uma seqüência iterativa de passos EM que maximiza a função de verossimilhança por meio do ajuste dos parâmetros do modelo. O algoritmo construtivo [14] é uma modificação do algoritmo EM tradicional, pois considera, durante o processo de ajuste dos parâmetros, operadores de adição de novas regras à estrutura, assim como, operadores de punição ou eliminação de regras existentes e pouco relevantes para o modelo de previsão. Este algoritmo é aplicado para inicializar o modelo adaptativo, através de um processo de ajuste *offline*, obtendo assim, uma inicialização adequada, que dará início à segunda etapa do processo de aprendizado.

Seja \mathbf{V}_i a matriz de covariância diagonal e definida positiva, de tal forma que soluções infactíveis sejam evitadas. De acordo com o algoritmo EM [4], e deduzindo as equações para a maximização da função de verossimilhança, uma solução ótima para o modelo proposto a cada passo M é definido por:

$$\alpha_i = \frac{1}{N} \sum_{k=1}^N h_i^k \quad (5)$$

$$\mathbf{c}_i = (\sum_{k=1}^N h_i^k \mathbf{x}^k) / \sum_{k=1}^N h_i^k \quad (6)$$

$$\mathbf{V}_i = (\sum_{k=1}^N h_i^k (\mathbf{x}^k - \mathbf{c}_i)^T (\mathbf{x}^k - \mathbf{c}_i)) / \sum_{k=1}^N h_i^k \quad (7)$$

para $i = 1, \dots, M$, onde N é o número de dados utilizado na fase de inicialização e $h_i^k, i = 1, \dots, M$, é a probabilidade a posteriori estimada no passo E do processo de otimização. Uma solução ótima para θ_i surge a partir da solução via mínimos quadrados, para a seguinte equação:

$$\sum_{k=1}^N \frac{h_i^k}{\sigma_i^2} (y^k - \phi^k \times \theta_i) = 0 \quad (8)$$

sendo y^k a saída desejada no instante k e σ_i o desvio padrão associado à saída local $y_i, i = 1, \dots, M$, sendo a variância σ_i^2 dada por:

$$\sigma_i^2 = (\sum_{k=1}^N h_i^k [y^k - \hat{y}^k]^2) / \sum_{k=1}^N h_i^k \quad (9)$$

Maiores detalhes referentes ao algoritmo EM podem ser encontrados em [4].

O algoritmo de aprendizado construtivo é utilizado para inicializar o modelo e definir de forma automática a base de regras inicial do aprendizado *online*. Como foi mencionado no início da seção, este algoritmo utiliza operadores de adição e eliminação de regras nebulosas. Estes operadores também serão utilizados como base para a adaptação da estrutura do modelo no aprendizado *online*.

3.1.1 Adição *offline* de regras nebulosas

O critério proposto para determinar uma situação na qual é necessário gerar uma nova regra nebulosa é chamado de critério do antecedente. Este critério verifica se a base de regras atual cobre o espaço de entrada de forma adequada. Assim, cada dado apresentado ao modelo deve pertencer a uma região ativa de pelo menos uma das regras nebulosas já existentes na estrutura do modelo, com uma probabilidade mínima superior a um limiar previamente definido.

Supondo que os dados tenham uma distribuição normal, para um dado nível de confiança de γ , os dados pertencentes à região ativa da i -ésima regra nebulosa estarão contidos no intervalo $[c_i - z_\gamma \sqrt{\text{diag}(V_i)}, c_i + z_\gamma \sqrt{\text{diag}(V_i)}]$, onde $\text{diag}(V_i)$ é a diagonal principal de V_i . Neste trabalho foi adotado um nível de confiança $\gamma = 72.86\%$, resultando em um $z_\gamma = 1.1$ (valor encontrado na tabela de distribuição normal) e um limiar igual a 0.1357. Esta condição implica que a máxima probabilidade $P[i | x^k]$ de x^k ativar pelo menos uma das regras nebulosas tem que ser maior que 0.1357. Ou seja:

$$\max(P[i | x^k])_{i=1, \dots, M} > 0.1357 \quad (10)$$

A situação descrita para a geração de uma nova regra nebulosa, considerando o dado de entrada no instante k é ilustrada na Figura 2. Assim, se a condição definida pela Eq. (10) não é satisfeita por um ou mais dados do conjunto de treinamento, significa que estes dados possuem uma probabilidade de pertencer a alguma região ativa coberta pela base de regras, inferior ao desejado.

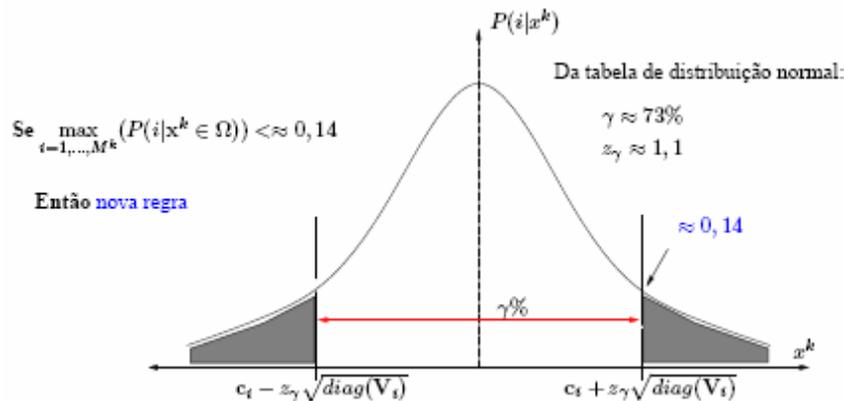


Figura 2: Critério de geração de novas regras nebulosas.

Denotando por Ω o conjunto contendo todos os dados entrada-saída que não satisfazem a Eq. (10), uma nova regra será inserida à estrutura do sistema. Assim,

$$M = M + 1 \quad (11)$$

e a estimativa inicial do centro desta nova regra c_M é dada por:

$$c_M = \frac{1}{N_\Omega} \sum_{i \in \Omega} x^i \quad (12)$$

sendo N_Ω o número de dados $[x^i \ y^i]$ em Ω . Além disso, os demais parâmetros associados a essa nova regra são inicializados da seguinte forma:

- $\sigma_M = 1.0$;
- $\theta_M = [\bar{y} \ 0 \ \dots \ 0]_{1 \times (p+1)}$, sendo \bar{y} o valor médio das saídas desejadas pertencentes ao conjunto Ω .

Os parâmetros α_i e V_i , para $i = 1, \dots, M$, devem ser re-inicializados de acordo com:

- $V_i = 10^{-4} I$, onde $I_{p \times p}$ é a matriz identidade;
- $\alpha_i = 1/M$.

Como α_i indica a relevância de cada regra nebulosa, a sua re-inicialização é necessária para que todas as regras tenham o mesmo peso, fazendo com que o ajuste dos parâmetros seja não-tendencioso, obtendo assim uma nova partição do espaço de entrada. A re-inicialização de V_i é necessária já que o processo de re-ajuste definirá novos valores de dispersão para cada regra nebulosa, de acordo com a nova partição gerada, ou seja, considerando neste processo a nova regra inserida na estrutura do modelo.

3.1.2 Eliminação *offline* de regras nebulosas

Como pode ser observado na Eq. (5), α_i pode ser interpretado como uma medida da relevância que cada regra nebulosa tem para a estrutura atual do modelo, quando comparadas entre si. Isto acontece porque α_i é proporcional a soma de todas as probabilidades h_i^k , as quais também podem ser consideradas como estimativas a posteriori das funções de pertinência g_i^k , sob o conjunto de dados de treinamento. Assim, quanto mais vezes a i -ésima regra nebulosa for ativada, maior será o valor de α_i .

Logo, α_i é o parâmetro que defini se existe ou não alguma regra a ser punida e eliminada da estrutura do modelo. Para tal, a regra com menor α_i será uma boa candidata e, para um dado limiar α_{\min} , toda regra com $\alpha_i < \alpha_{\min}$ a cada iteração, será eliminada da estrutura do sistema.

Após a inserção ou eliminação de alguma regra, o modelo será reajustado durante algumas iterações EM por meio das equações definidas na Seção 2 e Subseção 3.1. O processo global de aprendizado é finalizado após atingir a convergência e quando não existir mais mudanças na estrutura do modelo. Estes dois operadores de regras nebulosas também serão utilizados no aprendizado online, considerando algumas modificações que serão descritas a seguir.

3.2 Aprendizado *online*

O aprendizado online de modelos baseados em regras nebulosas, tem se tornado um método promissor, com diversos trabalhos propondo esta abordagem. Alguns exemplos de modelos com aprendizado online, que além de considerar o ajuste os parâmetros, também consideram a adaptação da estrutura são os trabalhos desenvolvidos por J. Er [2] e os modelos produzidos por P. Angelov [1].

O maior atrativo do aprendizado online, no caso da construção de modelos de séries temporais, é que não é necessário reajustar o modelo com todo o histórico da série, toda vez que um novo dado é disponibilizado. Além disso, a estrutura do modelo é modificada de acordo com a necessidade ao longo do tempo. Este artigo propõe um aprendizado online de modelos de séries temporais baseados em regras nebulosas, transformando as Eqs. (5)-(9) para uma versão recursiva, de tal forma que o ajuste da estrutura e dos parâmetros do modelo sejam feitos dinamicamente e simultaneamente.

O aprendizado adaptativo é realizado a cada instante de tempo $k, k = 1, 2, \dots$. Neste tipo de aprendizado as informações mais recentes têm uma maior influência no ajuste dos parâmetros e na estrutura do modelo. Observando a Eq. (5), e considerando uma janela no tempo $T = N$, esta equação pode ser re-escrita como segue:

$$\begin{aligned} \alpha_i^N &= \frac{1}{N} \sum_{k=1}^N h_i^k = \frac{1}{N} \sum_{k=1}^N (h_i^k - h_i^N) = \\ &= \frac{1}{N} [(N-1)\alpha_i^{N-1} + h_i^N] \\ &= \alpha_i^{N-1} + \frac{1}{N} [h_i^N - \alpha_i^{N-1}] \end{aligned} \quad (13)$$

onde h_i^N é a estimativa a posteriori de g_i^k e α_i^{N-1} é o valor estimado para α_i considerando apenas os $N-1$ dados iniciais. Como pode ser observado, a Eq. (13) é uma estimativa recursiva de (5). Seguindo este mesmo procedimento para um número genérico de iterações k e com uma janela no tempo T , as Eqs. (5)-(9) podem ser re-escritas recursivamente, obtendo as seguintes aproximações:

$$\alpha_i^{k+1} = \alpha_i^k + \frac{1}{T} [h_i^k - \alpha_i^k] \quad (14)$$

$$c_i^{k+1} = c_i^k + \frac{1}{\gamma_i^{k+1}} [x^k - c_i^k] \quad (15)$$

$$V_i^{k+1} = V_i^k + \frac{1}{\gamma_i^{k+1}} \left[(x^k - c_i^k)(x^k - c_i^k)' - V_i^k \right] \quad (16)$$

$$(\sigma_i^2)^{k+1} = (\sigma_i^2)^k + \frac{1}{\gamma_i^{k+1}} \left[(\hat{y}^k - y_i^k)^2 - (\sigma_i^2)^k \right] \quad (17)$$

na qual,

$$\frac{1}{\gamma_i^{k+1}} = \frac{h_i^{k+1}}{\sum_{t=1}^{k+1} h_i^t} \quad (18)$$

Denotando por $S_i^{k+1} = \sum_{t=1}^{k+1} h_i^t \alpha_i^t$, e $S_i(x^{k+1}) = h_i^{k+1}$, uma aproximação para $\sum_{t=1}^{k+1} h_i^t \alpha_i^t$ pode ser obtida como:

$$S_i^{k+1} \approx S_i(x^{k+1}) + \frac{T-1}{T} S_i^k \quad (19)$$

Re-escrevendo S_i^{k+1} tem-se:

$$S_i^{k+1} \approx S_i^k + \left[S_i(x^{k+1}) - \frac{S_i^k}{T} \right] \quad (20)$$

A Eq. (20) expõe alguns aspectos interessantes referentes às estimativas adotadas. S_i^k/T pode ser interpretada como uma estimativa do valor médio de α_i no intervalo de tempo T , prévio a $k+1$. Assim, quanto maior for S_i^k/T , maior será a importância da i -ésima regra para a próxima iteração. Se $S_i(x^k)$ obtém um valor baixo em T , então S_i^{k+1} diminuirá e a i -ésima regra terá um risco maior de ser eliminada.

Para estimar θ_i , aplica-se o algoritmo recursivo dos quadrados mínimos ponderados (wRLS), o qual considera um fator de esquecimento ao longo do tempo (f_{forget}). Maiores detalhes concernentes a este algoritmo de otimização podem ser encontrados em [15] e em [5]. Assim, as equações do algoritmo wRLS adaptadas ao problema de otimização de θ_i são definidas como:

$$\theta_i^{k+1} = \theta_i^k + C_i^{k+1} \phi^k \times h_i^k (y^k - y_i^k) \quad (21)$$

na qual:

$$C_i^{k+1} = \frac{C_i^k}{f_{forget}^k + h_i^k (\phi^k)^T C_i^k \phi^k} \quad (22)$$

é a matriz de covariância para cada θ_i durante o ajuste online. O fator de esquecimento $f_{forget} \in (0, 1]$, neste trabalho foi inicializado para $k=0$ como $f_{forget}^0 = 0.85$ e suavemente incrementado, tal que $f_{forget}^M \rightarrow 1.0$.

As condições iniciais para $\theta_i^0, i = 1, \dots, M$ foram dadas pelos valores obtidos na fase de inicialização.

Desta forma, e após a fase de inicialização, o aprendizado online é aplicado. Enquanto os parâmetros dos modelos são ajustados utilizando as Eqs. (14)-(22), para ajustar a estrutura do modelo a cada iteração é necessário adaptar os operadores de adição e eliminação utilizados na fase de inicialização.

3.2.1 Geração online de regras nebulosas

Neste caso, a geração de uma nova regra nebulosa e a sua inclusão na estrutura do modelo durante o aprendizado online será efetuada quando o dado atual de entrada x^k apresentado ao modelo não pertencer a nenhuma região de ativação das regras já existentes. Ou seja, a probabilidade do dado de entrada no instante k ativar alguma das regras nebulosas é inferior a um limiar previamente definido.

Da mesma forma que no caso do modelo construtivo, a probabilidade de x^k ativar alguma das regras nebulosas (ver Eq.(1)) deve ser superior a 0.1357, de acordo com a tabela de distribuição normal e considerando um nível de confiança de 72,86%. A condição definida pela Eq. (10) deve também ser satisfeita, sendo a condição verificada apenas para o dado atual já que não existe um conjunto de dados de treinamento, como no caso do modelo **C-FSM**.

Caso esta condição não seja satisfeita pelo k -ésimo dado de entrada apresentado uma nova regra é acrescentada ao sistema, sendo os seus parâmetros inicializados da seguinte maneira:

- $\alpha_{f+1}^k = x^k$;
- $\alpha_{f+1}^k = 1.0$;
- $\alpha_{f+1}^k = [y^k \ 0 \ \dots \ 0]_{1 \times p+1}$;
- $V_{f+1}^k = 10^{-4}I$, onde I é uma matriz identidade de ordem $p \times p$;
- $\alpha_{f+1}^k = 10^{-8}$. Embora este valor inicial seja pequeno o suficiente para inferir na dinâmica da estrutura atual, todos os parâmetros $\alpha_f, f = 1, \dots, M+1$ são re-normalizados tal que a soma de todos estes coeficientes seja igual à unidade.

3.2.1 Eliminação *online* de regras nebulosas

O operador de punição ou eliminação de regras nebulosas para o aprendizado *online* requer mais de uma condição a ser verificada antes da sua aplicação.

Após uma nova regra ser criada, o seu respectivo α_f terá um valor pequeno e igual à 10^{-8} . Se o operador de poda é aplicado imediatamente, esta regra nebulosa recém gerada será eliminada sem a devida verificação da sua relevância na estrutura do modelo.

Esta situação é resolvida com a criação de um novo índice, chamado de índice de permanência τ . Toda vez que uma nova regra é gerada, o seu respectivo índice de permanência τ_f também será criado. À medida que esta regra é ativada ao longo do tempo, este índice é incrementado:

$$\tau_f^{k+1} = \tau_f^k + 1 \quad (23)$$

Logo, uma regra nebulosa na estrutura poderá ser eliminada somente se o seu α_f possui um valor muito pequeno:

$$\alpha_f < \alpha_{min}$$

e, se:

$$\tau_f^k > \epsilon T \quad (24)$$

sendo $\epsilon > 0$ e T a janela de tempo utilizada durante o treinamento *online*.

Esta condição garante que nenhuma regra seja eliminada imediatamente após à sua criação, permitindo-lhe um tempo mínimo para a sua adaptação igual a ϵT , eliminando assim, possíveis oscilações desnecessárias e pouco eficientes na estrutura do modelo.

Após a verificação dos critérios de geração de novas regras e de poda, os parâmetros dos modelos são re-ajustados utilizando as equações recursivas e fornecendo a saída estimada para a próxima iteração, a qual é a estimação conhecida como previsão um passo à frente na teoria de análise de séries temporais [16].

4. Previsão de Séries Temporais

O modelo proposto é utilizado na construção de um modelo de série temporal para uma série de vazões semanais de um posto brasileiro. O modelo também é utilizado na modelagem de uma série financeira, a qual faz parte do conjunto reduzido de séries temporais utilizado na Competição Neural de Previsão 2006/2007 (NN3 Competition), como parte do International Joint Conference on Neural Networks'07.

4.1 Série de Vazões Semanais

A série de vazões semanais corresponde ao histórico referente ao posto de Sobradinho, localizado na bacia do rio São Francisco, região nordeste do Brasil. Estes dados abrangem o período da primeira semana de Janeiro de 1931 até a última semana de Dezembro de 2000.

Os dados são separados em dois conjuntos. O primeiro, composto apenas pelos primeiros cinco anos é utilizado na fase de inicialização do modelo. Este mesmo conjunto de dados é analisado para a seleção adequada de entradas para o modelo de previsão. Para tal, utilizou-se o critério de Informação Mútua Parcial (PMI), aplicando a metodologia descrita em [14]. De acordo com este critério, a partir de um conjunto inicial de possíveis entradas igual a 15, foram selecionados os atrasos 1, 2, 3 e 6, para a construção do modelo de série temporal.

Assim, os vetores de entrada-saída são construídos, e procede-se a inicialização do modelo, usando o algoritmo *offline*. O parâmetro α_{\min} para a fase de inicialização foi igual a 0.001 obtendo uma estrutura inicial composta por 18 regras nebulosas.

A fase de ajuste online foi inicializada a partir da primeira semana de Janeiro de 1936. O parâmetro α_{\min} para esta fase de ajuste dinâmico foi igual a 0.0025, $T=104$, o qual representa uma janela de tempo equivalente aos dois últimos anos, e $\epsilon = 0.50$. A Figura 3 ilustra a variação do número de regras durante o aprendizado *online*. Em geral, o número de regras durante o ajuste *online* da estrutura do modelo de série temporal é sempre menor ou igual ao número inicial de regras obtidas na inicialização *offline*. Isto ocorre, pois o aprendizado *online* permite que a estrutura do modelo mantenha apenas as regras nebulosas de maior relevância no período T imediatamente anterior ao instante atual $k+1$, mesmo definindo um α_{\min} maior que o utilizado na fase de inicialização. Para um instante $k > 500$, nota-se que a estrutura do modelo oscilou entre um total de 4 a 12 regras nebulosas.

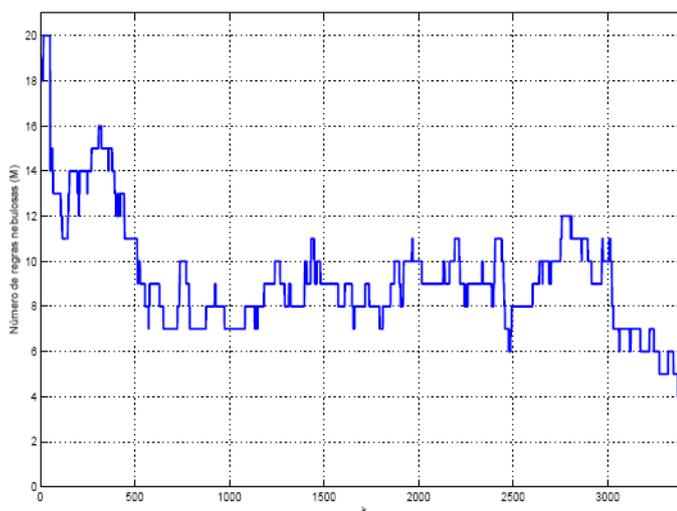


Figura 3: Variação do número de regras durante o aprendizado *online*.

O modelo **eTS** foi ajustado para a série de vazões semanais. Como o modelo pode ser inicializado sem nenhuma informação *a priori*, ou seja, sem nenhuma regra e com somente um dado, o modelo **eTS** foi aplicado as séries de vazões para previsão um passo à frente, a partir do ano de 1936.

A Figura 4 ilustra a análise dos resíduos durante o aprendizado online para verificar a adequação do modelo. A Figura 4-(a) ilustra a função de autocorrelação parcial estimada e a Figura 4-(b) ilustra a função de correlação cruzada estimada a cada iteração. Em ambas as figuras, as linhas tracejadas representam os limites $\pm 2/\sqrt{n}$, os quais fornecem uma indicação geral de possível quebra do comportamento de ruído branco, com n igual ao número de graus de liberdade do modelo obtido. Observa-se que, à medida que os dados são apresentados ao modelo, os resíduos estimados estão sendo menos correlacionados, localizando-se dentro dos limites permitidos.

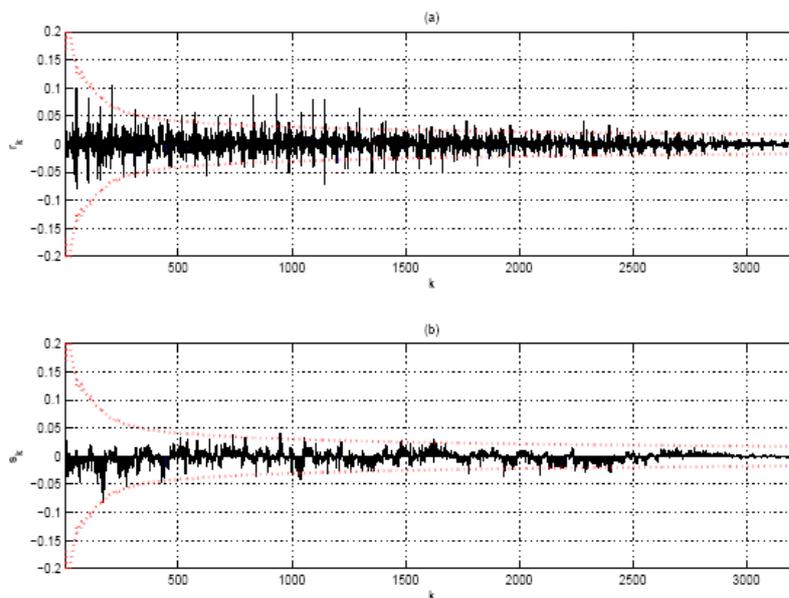


Figura 4: Análise dos resíduos.

Os indicadores de desempenho utilizados para esta aplicação são o erro relativo médio (**ERM (%)**), a raiz do erro quadrático médio (**REQM (m^2/s)**) e o erro absoluto médio (**EAM (m^2/s)**), os quais são definidos pelas seguintes equações:

$$ERM = \frac{100}{N} \sum_{k=1}^N \frac{|y^k - \hat{y}^k|}{\hat{y}^k} \tag{25}$$

$$REQM = \left(\frac{1}{N} \sum_{k=1}^N (y^k - \hat{y}^k)^2 \right)^{1/2} \tag{26}$$

$$EAM = \frac{1}{N} \sum_{k=1}^N |y^k - \hat{y}^k| \tag{27}$$

sendo N o número de dados a prever, y^k a saída desejada e \hat{y}^k a saída do modelo.

A Figura 5 apresenta as previsões da série de vazões para os modelos **A-FSM** e **eTS**, na previsão um passo à frente, a partir da primeira semana de Janeiro de 1991 até a última semana de Dezembro de 2000. A Tabela 1 apresenta os erros de previsão para os modelos **C-FSM**, **A-FSM** e **eTS**, no período de Janeiro de 1936 a Dezembro de 2000.

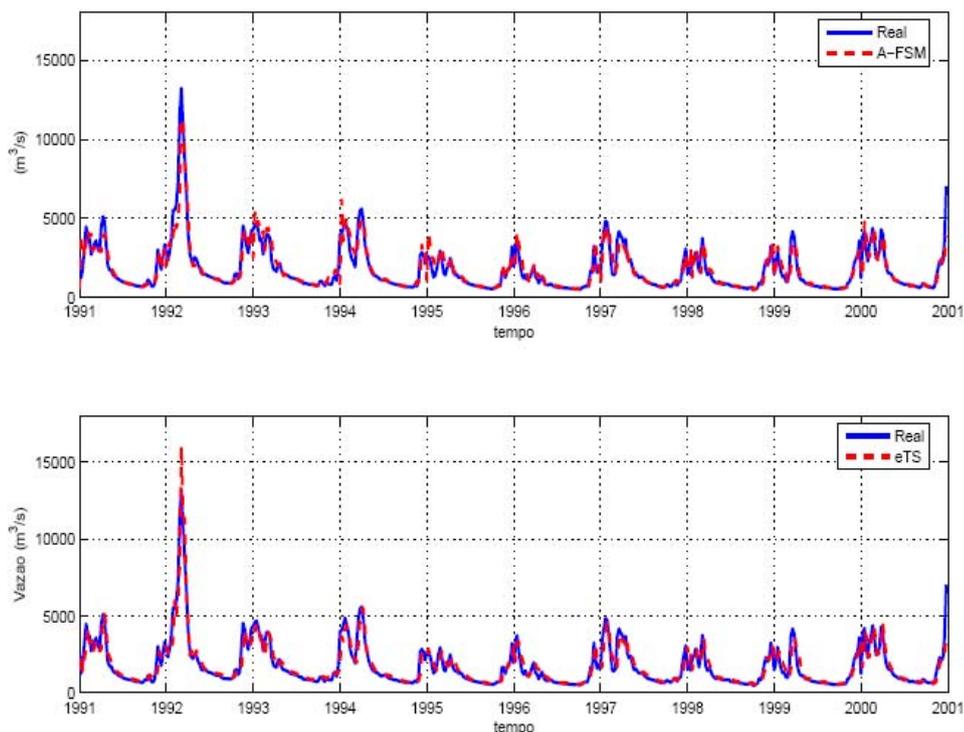


Figura 5: Previsão um passo à frente.

Tabela 1: Erros de previsão.

Modelo	M (regras)	ERM (%)	EQM (m^3/s)	EAM (m^3/s)
C-FSM	2-8	15.33	567.38	318.47
A-FSM	4-20	15.74	614.04	335.75
eTS	0-13	17.52	570.77	341.88

4.2 Série Financeira

A série escolhida arbitrariamente do conjunto utilizado na competição é a série NN3_109, de um total de 11 séries temporais. Este conjunto de séries é composto por séries financeiras, assim como por séries industriais, cujos dados faltantes, assim como maiores informações e características particulares serão disponibilizados em meados do ano de 2008. Os resultados de previsão um e até dezoito passos à frente são comparados aos obtidos pelo modelo construtivo C-FSM [14].

A série NN3_109 é composta por 123 observações mensais, no período de Janeiro de 1979 a Março de 1989, sendo os respectivos valores ilustrados na Figura 6-(a). Neste caso, a série é definida como uma série não sazonal, sendo composta por um componente de tendência de primeira ordem e o componente aleatório. A série foi transformada, trabalhando assim, como a primeira diferença da série a qual é ilustrada na Figura 6-(b).

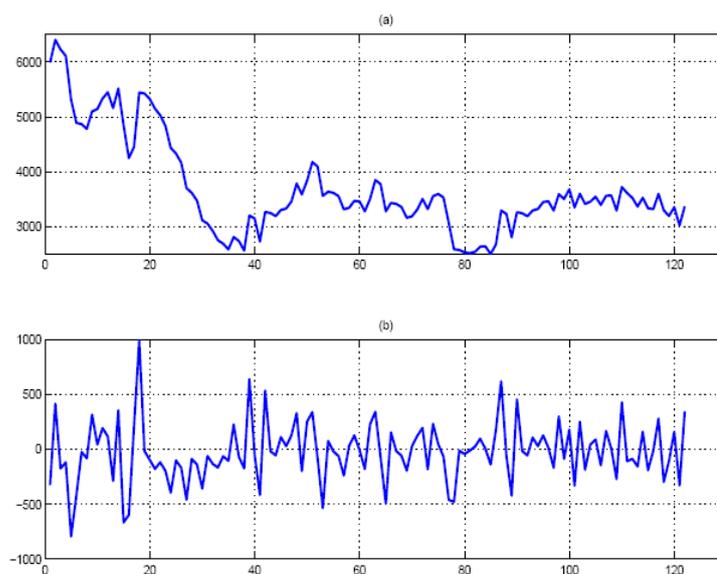


Figura 2: Série NN3_109: (a) histórico, (b) primeira diferença.

Deve-se observar que, embora o aprendizado seqüencial permita a adaptação do modelo com a apresentação de um novo dado, neste caso, o aprendizado foi finalizado antes de efetuar o teste, tal que o mesmo modelo seja utilizado para o cálculo das previsões um e múltiplos passos à frente. Por tal motivo, foram definidos dois conjuntos de dados: o conjunto de treinamento, composto pelos dados do histórico até Setembro de 1987, e o conjunto de teste ou validação, composto pelos últimos dezoito valores do histórico, o qual tem por objetivo avaliar o desempenho do modelo na previsão um e até dezoito passos à frente.

Neste caso, a seleção de entradas utilizou a abordagem combinada FNN-PMI [18], devido ao histórico curto da série temporal, pois, quanto menor a quantidade de dados disponíveis para a estimativa do PMI, menor será a precisão das mesmas. Assim, o conjunto inicial de possíveis entradas deve ser o mínimo possível, deixando neste caso, a definição do conjunto inicial de possíveis entradas para o algoritmo FNN.

No caso do modelo **C-FSM**, a seleção de entradas foi realizada considerando o conjunto completo de dados de treinamento. No caso do modelo adaptativo **A-FSM**, a seleção de entradas foi realizada considerando apenas os primeiros quatro anos do histórico da série, os mesmos que serão utilizados na etapa de inicialização do modelo, deixando os outros cinco anos para a etapa de aprendizado seqüencial.

O conjunto inicial de possíveis entradas para o modelo é definido pelo algoritmo FNN ou algoritmo dos Falsos vizinhos mais próximos [19], sendo composto pelos primeiros seis atrasos da série temporal no caso do modelo construtivo e pelos primeiros quatro atrasos da série no caso do modelo **A-FSM**. Após a construção deste conjunto inicial, o critério PMI é verificado para cada uma das possíveis variáveis de entrada.

No final do processo iterativo de seleção de variáveis, o conjunto final de entradas para o modelo **C-FSM** é composto pelos atrasos 2, 3 e 6, como mostra a Tabela 2. A primeira coluna mostra o número da iteração do processo de seleção de entradas utilizando o critério PMI, a segunda coluna mostra o atraso ou variável de entrada escolhido nessa iteração. A terceira coluna mostra o valor de PMI obtido pelo atraso selecionado, que corresponde ao máximo valor de PMI obtido pelas variáveis avaliadas nessa iteração. A quarta coluna mostra o limiar do PMI estimado para cada iteração do processo de seleção de entradas, o qual, como pode ser visto, finaliza quando o valor de PMI obtido na iteração final é inferior ao limiar calculado.

As próximas colunas (quinta, sexta e sétima), mostram os resultados obtidos na seleção de entradas para o modelo com aprendizado seqüencial (**A-FSM**), ou seja, considerando apenas os quatro primeiros anos do histórico da série. Neste caso, como a Tabela 2 ilustra, a única variável de entrada a ser considerada para a construção do modelo adaptativo é o atraso 3.

Uma vez que as entradas dos modelos são definidas, os modelos construtivo e *online* são inicializados e ajustados.

Tabela 2: Resultados do algoritmo FNN-PMI para a série NN3_109.

	NN3_109 (aprox. 9 anos)			NN3_109 (aprox. 4 anos)		
Iteração	Lag	PMI	Limiar	Lag	PMI	Limiar
1	3	0.3037	0.2285	3	0.3449	0.3005
2	2	0.2030	0.1995	2	0.2024	0.2517
3	6	0.1418	0.1169	-	-	-
4	1	0.0609	0.0783	-	-	-

Como já foi mencionado, o modelo **C-FSM** utiliza os primeiros 9 anos do histórico da série durante a etapa de inicialização da estrutura, assim como durante o aprendizado do modelo. O valor de α_{\min} considerado para ambas as etapas foi igual a 0.01 e o processo foi finalizado após 40 épocas, sendo a estrutura final composta por $M=4$ regras nebulosas. Já o modelo **A-FSM** utilizou apenas os primeiros quatro anos do histórico da série durante a etapa de inicialização da estrutura, e os próximos quatro anos da série para a etapa de ajuste sequencial, com $\alpha_{\text{util}} = 0.01$, $f_{\text{forget}} = 0.95$, $T = 20$ e $\epsilon = 0.50$ para o aprendizado *online*, sendo composto por $M=2$ regras nebulosas no final do aprendizado.

As métricas de erro utilizadas para verificar o desempenho dos modelos foram o erro relativo médio simétrico (**ERMs (%)**) e o erro absoluto médio (**EAM**). O **ERMs** é calculado utilizando a seguinte equação:

$$\text{ERMs} = \frac{100}{N} \sum_{k=1}^N \frac{|y^k - \hat{y}^k|}{(|y^k - \hat{y}^k|)^{1/2}} \quad (28)$$

onde N é o número de padrões no conjunto de teste ou validação; y^k é a saída desejada e \hat{y}^k é a saída estimada pelo modelo de previsão.

O **ERMs** é uma medida de erro empregada na Competição de Previsores com Redes Neurais (*Neural Networks Competition - NN3*). Utilizando esta medida, elimina-se o problema de obter erros grandes quando y^k é próximo de zero, e de diferenças muito grandes entre os erros percentuais absolutos quando y^k é maior que \hat{y}^k e vice-versa [20]. Por exemplo, quando $y^k = 100$ e $\hat{y}^k = 50$, o erro percentual médio é igual a **ERM**=50%, enquanto que, para $y^k = 50$ e $\hat{y}^k = 100$ o erro percentual médio é igual a **ERM**=100%. Já no caso do **ERMs**, este será igual a 66.67% para ambos os casos. Além disso, o **ERM** é ilimitado, enquanto o **ERMs** adota valores no intervalo $[-200\%, 200\%]$.

A Tabela 3 apresenta os erros de previsão obtidos para o conjunto de treinamento e na etapa de validação dos modelos, para a previsão um e até dezoito passos à frente, assim como o número de regras nebulosas que compõe cada modelo de série temporal no final do aprendizado.

Em geral, os dois modelos conseguiram resultados de previsão bastante próximos quando comparados entre si, porém a degradação do erro de previsão do **A-FSM** foi inferior ao apresentado pelo modelo construtivo, como ilustrado na Figura 7, o qual apresenta a variação do **ERMs** para diferentes horizontes de previsão.

Ambos os modelos atingiram patamares similares de erros relativos simétricos, mostrando-se o modelo **A-FSM** como o mais estável, já que o **ERMs** mantém-se praticamente constante para os diferentes horizontes de previsão avaliados.

Tabela 3: Erros de previsão para a série NN3_109.

Modelo	Treinamento			Validação		Validação	
	M	ERMs (%)	EAM (u)	ERMs (%)	EAM (u)	ERMs (%)	EAM (u)
				1 passo à frente		Até 18 passos à frente	
C-FSM	4	3.68	133.95	3.41	116.49	3.74	128.41
A-FSM	2	4.15	54.74	3.48	121.79	3.57	125.28

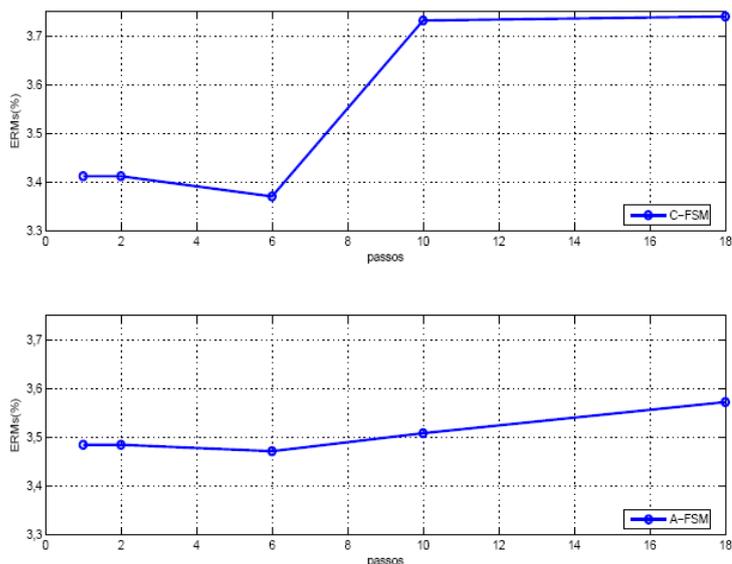


Figura 7: Variação do ERMs de 1 até 18 passos à frente para os modelos de série temporal obtidos, série NN3_109.

Os resultados de previsão, um e até dezoito passos à frente são ilustrados na Figura 8, para cada um dos modelos ajustados.

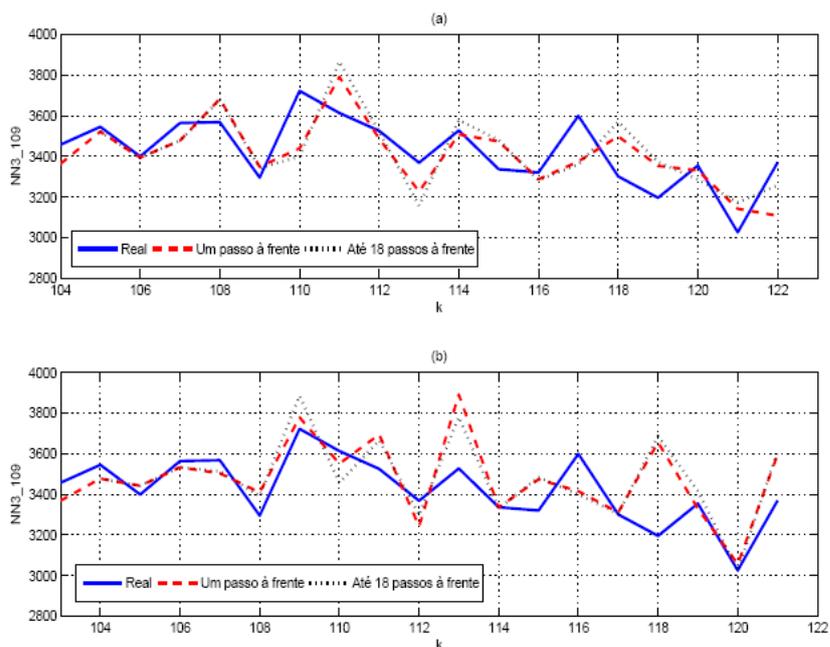


Figura 8: Previsão vários passos à frente para a série NN3_109: (a) C-FSM, (b) A-FSM.

5. Conclusões

Este trabalho apresenta um modelo baseado em regras nebulosas com aprendizado adaptativo para previsão de séries temporais. A estratégia de aprendizado é baseada no algoritmo de otimização EM, e em operadores de adição e eliminação de regras nebulosas da estrutura do modelo. Estes operadores são aplicados durante o aprendizado definindo de forma automática a estrutura do modelo. O modelo **A-FSM** foi aplicado para a geração de um modelo de previsão de vazões semanais do posto de Sobradinho, aplicado na previsão um passo à frente e os resultados foram comparados com o modelo **eTS**. Em uma segunda aplicação, o modelo foi ajustado e avaliado na previsão um e até dezoito passos à frente de uma série financeira. A análise apresentada mostrou que o sistema nebuloso proposto fornece modelos de previsão adequados e parcimoniosos, tanto na previsão um passo à frente, como na previsão múltiplos passos à frente. Experimentos computacionais, considerando um problema de modelagem de processo não linear, sugerem que, quando comparadas com estruturas similares propostas na literatura, as redes neurais nebulosas são eficazes para aproximar modelos de sistemas dinâmicos não lineares com erros de aproximação satisfatórios.

6. Agradecimentos

Os autores agradecem à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) processo 2003/10019-9, e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) processo 305796/2005-4, pelo auxílio.

7. Referências

- [1] P. P. Angelov e D. P. Filev: "An Approach to Online Identification of Takagi-Sugeno Fuzzy Models", IEEE Transactions on Systems, Man and Cybernetics - Part B, vol. 34, no 1, 2004, pp. 484 – 498.
- [2] J. Bezdek: "Pattern Recognition with Fuzzy Objective Function Algorithms", New York, USA: Plenum Press, 1981.
- [3] R. Yager e D. P. Filev: "Essentials of Fuzzy Modeling and Control", John Wiley and Sons. Inc., 1994.
- [4] R. Jacobs, M. Jordan, S. Nowlan e G. Hinton: "Adaptive Mixture of Local Experts", Neural Computation, vol. 3, no. 1, 1991, pp. 79-87.
- [5] V. Ramamurti e J. Ghosh: "Structural Adaptation in Mixture of Experts", 13th International Conference on Pattern Recognition, vol. 4, 1996, pp. 704-708.
- [6] M. J. Er e S. Wu: "A fast learning algorithm for parsimonious fuzzy neural systems", Fuzzy Sets and Systems, vol. 126, 2002, pp. 337 – 351.
- [7] G. Leng, T. McGinnity, e G. Prasad: "An approach for on-line extraction of fuzzy rules using a self-organizing", Fuzzy Sets and Systems, vol. 150, no. 2, 2005, pp. 211 – 243.
- [8] S. Haykin: "Redes Neurais, Princípios e Prática", 2 edn, Bookman, 2001.
- [9] T. Takagi e M. Sugeno: "Fuzzy Identification of Systems and Its Applications to Modeling and Control", IEEE Transactions on Systems, Man and Cybernetics, vol. 1, 1985, pp. 116 – 132.
- [10] P. Guo e L. Xu: "Relationship between Mixture of Experts and Ensemble Neural Networks", Proceedings of the 6th International Conference on Neural Information Processing, vol. 1, 1999, pp. 246-250.
- [11] B. Warner e M. Misra: "Understanding Neural Networks as Statistical Tools", The American Statistician, vol. 50, no. 4, 1996, pp. 284-293.
- [12] J. M. Kihoro, R. O. Otieno e C. Wafula: "Seasonal time series forecasting: a comparative study of ARIMA and ANN models", African Journal of Science and Technology, vol. 5, no. 2, 2004, pp. 41-49.
- [13] R. S. Tsay: "Analysis of Financial Time Series", 2nd edn, Wiley & Sons, 2005.
- [14] I. Luna, S. Soares e R. Ballini: "A Constructive-Fuzzy Rule Based System for Time Series Forecasting", International Joint Conference on Neural Networks - IJCNN'07, 2007.
- [15] S. Haykin: "Kalman Filtering and Neural Networks", John Wiley & Sons, Inc.2001.
- [16] L. A. Aguirre: "Introdução à Identificação de Sistemas: Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais", 2 edn, Editora UFMG, Belo Horizonte, BH, Brasil, 2004.
- [17] I. Luna, S. Soares, R. Ballini: "Técnica de identificação de modelos lineares e não-lineares de séries temporais", Sba: Controle & Automação Sociedade Brasileira Automática, vol. 17, pp. 245-256, 2006.
- [18] I. Luna, S. Soares, R. Ballini: "A technique for identifying time series models", XVI Congresso Brasileiro de Automática - CBA'06, 2006.
- [19] S. Akaho: "Conditionally independent component analysis for supervised feature extraction", Neurocomputing, vol. 49, pp. 139-150, 2002.
- [20] S. Makridakis e M. Hibon: "The M-3 Competition: results, conclusions and implications", International Journal of Forecasting, vol. 16, pp. 541-476, 2000.