

Determinação de Modelo Markoviano para Canal Sem Fio Empregando Aprendizado de Máquina

Higor Í. dos Santos, Daniel P. B. Chaves e Cecilio Pimentel

Resumo—Séries temporais são a maneira natural de obter informações sobre sistemas ou processos dinâmicos em uma variedade de aplicações científicas, de engenharia e financeiras. A complexidade dessas pode tornar imperativo o uso de métodos baseados em dados. Entre os quais destacamos a técnica de Filtragem por Dinâmica Simbólica, que envolve a determinação de modelos markovianos para expressar a estrutura causal do comportamento dinâmico observado. Neste trabalho apresentamos um algoritmo para gerar esses modelos baseado em cadeias de Markov de comprimento variável, aprendizado de máquina e algoritmo de minimização de estados de grafos. Para validar o algoritmo, modelamos um canal de comunicação sem fio, demonstrando que o modelo obtido é superior aos gerados por outras técnicas.

Palavras-Chave— Clusterização, comunicação sem fio, minimização de grafos, modelos de Markov de comprimento variado.

Abstract—Time series are the natural way for accessing information about dynamical systems or processes in a variety of scientific, engineering and financial applications. Due to their complexity the use of data-based methods is imperative. Among which we highlight the Symbolic Dynamic Filtering technique, which involves the determination of Markovian models to express the causal structure of the observed dynamic behavior. We propose a new algorithm for obtain these models based on variable length Markov chains, machine learning algorithms and graph minimization techniques. To validate the algorithm, we model a wireless communication channel, showing that the obtained model is superior to those generated by other techniques.

Keywords— Clustering, wireless communication, graph minimization, variable length Markov models.

I. INTRODUÇÃO

Pode-se citar dois métodos para tratar o problema de modelagem/análise de processos ou sistemas dinâmicos: baseado em modelo e orientado a dados [1]. O primeiro procura descrever analiticamente o sistema dinâmico; enquanto o segundo estabelece uma relação entre as variáveis do espaço de estados do sistema mesmo sem o conhecimento explícito do seu comportamento dinâmico, tornando-se atrativo quando envolve sistemas complexos, já que essa característica pode tornar impraticável a obtenção de modelos analíticos [1], [2]. Este método pode ser caracterizado por capturar o mecanismo responsável pela geração dos dados observados, sendo de particular valia para a análise de séries temporais, em especial quando se trata de sistemas não lineares [3]. A análise de séries temporais envolve fatores experimentais que podem afetar a

confiabilidade dos resultados, tais como: taxa de amostragem, resposta dinâmica dos instrumentos e relação sinal ruído. A Filtragem por Dinâmica Simbólica (SDF, *symbolic dynamic filtering*) é uma técnica de processamento de séries caracterizada por capturar o comportamento dinâmico subjacente à série [4], [5], além de apresentar maior robustez ao efeito do ruído.

A forma usual de aplicar SDF à análise de séries temporais é pela determinação de um autômato probabilístico de estados finitos (PFSA, *Probabilistic Finite State Automata*) que reflita a estrutura causal do comportamento dinâmico observado. Um algoritmo clássico para determinar um PFSA a partir de uma série temporal é o D -Markov, que tem como maior limitação o crescimento exponencial do número de estados em função da memória que se deseja capturar. Isto decorre da suposição que todas as sequências de um dado comprimento, no caso D , são necessárias para refletir a estrutura causal da dinâmica. Uma proposta que tenta contornar essa explosão de estado é apresentada em [6], contudo, requer o pós-processamento do PFSA inicialmente gerado, o que pode elevar exponencialmente o número de estados do PFSA final. Em [7] supõe-se a existência de palavras de sincronização para gerar PFSA compactos. Nesse caso, tal suposição restringe significativamente o escopo do algoritmo, já que não é satisfeita em muitos casos e não há como determinar previamente se será satisfeita. Em [5] são utilizadas operações sucessivas de colapso e quebra de estados sobre um PFSA inicial, o que tem por objetivo eliminar estados considerados redundantes sem comprometer a capacidade do PFSA de refletir a dinâmica subjacente à série temporal. Contudo, essas operações requerem buscas exaustivas sobre o conjunto de estados, tornando a complexidade do algoritmo elevada.

Em [8] os autores propõem o emprego de clusterização e algoritmos de minimização de estados para obter um PFSA reduzido a partir de um D -Markov. Os PFSA gerados, para diversas classes de sistemas dinâmicos, apresentam uma redução no número de estados que pode superar 50%. Contudo, o algoritmo requer a determinação de um D -Markov na sua primeira etapa, cujo parâmetro D dita a máxima memória capturada pelo PFSA. Nesse contexto, trazemos três contribuições neste artigo: (1) Especificamos como construir uma cadeia de Markov de comprimento variável em substituição ao D -Markov empregado em [8], com isso nos atemos as sequências relevantes para refletir a estrutura causal da dinâmica para a construção do PFSA inicial; (2) Propomos uma modificação para o cálculo dos centroides no algoritmo de clusterização que conduz a um ganho de desempenho dos PFSA gerados quando comparados ao caso tradicional; (3) Demonstramos a

viabilidade do algoritmo proposto para modelagem de canais sem fio discretos.

Este artigo é composto de seis seções. Na Seção II discorremos sobre conceitos preliminares. Na Seção III apresentamos o algoritmo proposto. Os quantificadores de desempenho empregados para comparar nossa proposta com outros algoritmos são abordados na Seção IV. Os resultados são apresentados na Seção V e, por fim, tecemos as conclusões na Seção VI.

II. PRELIMINARES

Nesta seção abordaremos conceitos sobre: grafos [9], PFSA [5] e clusterização [10]; que são relevantes para o algoritmo proposto. Uma sequência finita u de símbolos de um alfabeto Σ é chamada uma palavra e seu comprimento é denotado por $|u|$. A palavra vazia ε é uma sequência de comprimento zero, tal que $u\varepsilon = \varepsilon u = \varepsilon$. O conjunto de todas as possíveis palavras de comprimento n sobre Σ é Σ^n . Neste trabalho, por simplicidade de notação, denotamos por grafo um grafo direcional rotulado, conforme a Definição 1.

Definição 1 (Grafo): Um grafo G sobre o alfabeto Σ consiste em uma tripla (Q, Σ, δ) :

- Q é o conjunto finito de estados com cardinalidade $|Q|$;
- Σ é o alfabeto finito com cardinalidade $|\Sigma|$;
- δ é a função de transição de estados tal que $Q \times \Sigma \rightarrow Q$.

É possível estender a definição da função de transição para que esta aceite também palavras, e não apenas símbolos [8].

Definição 2 (Contexto à direita): O contexto à direita de um estado $q \in Q$ de um grafo G é definido como o conjunto de todas as palavras geradas por caminhos que começam no estado q e terminam em algum estado de Q .

Um grafo é chamado de reduzido se os contextos à direita de cada estado são distintos. Dado um grafo original, existem dois algoritmos para gerar um grafo reduzido: Moore e Hopcroft [11]; também conhecidos como algoritmo de minimização de estados.

Definição 3 (PFSA): Um PFSA é um par (G, π) no qual G é um grafo e π é uma atribuição de probabilidade $\pi : Q \times \Sigma \rightarrow [0, 1]$ que satisfaz

$$\sum_{\sigma \in \Sigma} \pi(q, \sigma) = 1, \forall q \in Q. \quad (1)$$

Consideramos na Definição 3 que todos os estados de Q são estados iniciais e finais tal que um PFSA é uma ênupla (Q, Σ, δ, π) . Dado um estado $q \in Q$, o vetor distribuição de probabilidade $\mathcal{V}(q) = (\pi(q, \sigma_1), \pi(q, \sigma_2), \dots, \pi(q, \sigma_{|\Sigma|}))$ é chamado de *morph* deste estado.

A. D -Markov

A máquina D -Markov [5] é um PFSA que gera símbolos que dependem apenas de D símbolos passados. Esta gera um processo Markoviano $\{s_n\}$ de ordem D

$$\Pr(s_n | \dots s_{n-D} \dots s_{n-1}) = \Pr(s_n | s_{n-D} \dots s_{n-1}), \quad (2)$$

em que $\Pr(\cdot)$ é uma medida de probabilidade. Para construir uma máquina D -Markov, $|\Sigma|^D$ estados são criados gerando o conjunto Q em que cada estado é rotulado com uma sequência de comprimento D sobre um alfabeto Σ . Para cada $\tau \in \Sigma$,

existe uma transição de estados rotulada com o símbolo τ que parte do estado q com rótulo $\sigma_1 \sigma_2 \dots \sigma_D$, $\sigma_i \in \Sigma$ para o estado q' com rótulo $\sigma_2 \dots \sigma_D \tau$, isto é $\delta(q, \tau) = q'$, com probabilidade

$$\pi(q, \tau) = \frac{\Pr(q\tau)}{\Pr(q)} = \frac{\Pr(\sigma_1 \sigma_2 \dots \sigma_D \tau)}{\Pr(\sigma_1 \sigma_2 \dots \sigma_D)}, \quad (3)$$

desde que $\Pr(\sigma_1 \sigma_2 \dots \sigma_D) > 0$. Após a remoção de estados sem aresta incidente, caso haja, uma máquina D -Markov essencial é gerada.

B. Árvore de Contexto

A determinação de um PFSA D -Markov envolve a determinação de $|\Sigma|^D (|\Sigma| - 1)$ parâmetros livres. Esse valor provém da memória constante e igual a D atribuída ao D -Markov. Contudo, a depender da sequência observada até um dado instante, o histórico relevante pode variar (com comprimento limitado a D), o que implica em um valor de memória médio menor que D . Dá-se o nome de cadeia de Markov de comprimento variável (VLMC - *Variable length Markov Chain*) ao processo cuja memória varia a depender dos símbolos passados até o instante atual [12], [13].

A determinação das sequências relevantes para a construção do D -Markov é simples, pois se trata de todas aquelas de comprimento D . No caso da VLMC, antes de mais nada, é preciso determinar o que caracteriza uma sequência relevante. Se o sistema original é de ordem finita D (máxima memória), sendo uma sequência s relevante, então possui comprimento máximo D . Além disso, se s é sufixo de qualquer outra sequência w , a dinâmica a partir desta sequência pode ser determinada observando a dinâmica a partir de s . Isso conceitua a função contexto ou função sufixo; $c : \Sigma^D \mapsto \mathcal{C}$, tal que, para $w \in \Sigma^D$ há $s \in \mathcal{C}$ que satisfaz $\Pr(\sigma|w) = \Pr(\sigma|s)$, para qualquer $\sigma \in \Sigma$, de forma que $c(w) = s$. Observe que \mathcal{C} é um conjunto mínimo, de forma que $c(\cdot)$ é bem definida.

A determinação do conjunto \mathcal{C} pode ser feita a partir de uma árvore de contexto (CT - *Context Tree*). Neste trabalho, para determinar a CT se gera recorrentemente novos nós, iniciando pelo nó raiz ε , até que os rótulos de todas as extensões a partir de um nó apresentem o mesmo *morph* ou, na prática, que tenham *morphs* similares (adotaremos como critério de similaridade entre *morphs* a distância euclidiana menor que um certo limiar T). Nesse caso, o nó pai é especificado como uma folha da CT. A partir de cada nó divergem $|\Sigma|$ ramos, cada qual tendo como rótulo um elemento de Σ . O rótulo de um caminho na CT é a concatenação dos rótulos dos ramos percorridos, sendo esse também o rótulo da folha alcançada.

Os parâmetros necessários para determinar a CT são sua profundidade, L , e a distância euclidiana máxima permitida, T , entre *morphs* de nós para que estes ainda sejam considerados similares. Os *morphs* dos nós são especificados como em (3), em que a probabilidade das sequências são estimadas a partir da série temporal. Logo, para simplificar a notação, podemos representar o *morph* de um estado q por $\mathcal{V}(q)$ ou por $\mathcal{V}(w)$, em que w é a sequência que rotula q . Inicia-se a construção da CT a partir do nó de rótulo ε . Neste caso, se $\|\mathcal{V}(\sigma_i \varepsilon) - \mathcal{V}(\sigma_j \varepsilon)\| > T$ (em que $\|\cdot\|$ denota distância

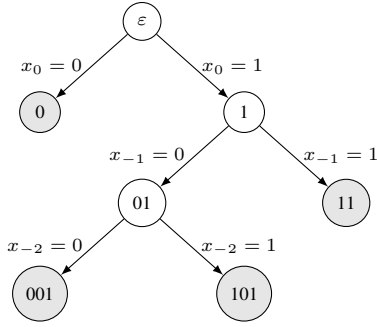


Fig. 1. Exemplo de árvore de contextos com $\Sigma = \{0,1\}$ e $L = 3$.

euclidiana entre dois vetores) para algum par $\sigma_i, \sigma_j \in \Sigma$, então para cada elemento em Σ é criado um nó filho. Caso o nó raiz seja estendido, a árvore será percorrida em pré-ordem de forma iterativa. A cada novo nó q com rótulo w visitado, verifica-se a desigualdade $\|\mathcal{V}(\sigma_i w) - \mathcal{V}(\sigma_j w)\| > T$ para algum par $\sigma_i, \sigma_j \in \Sigma$, que quando observada implica na criação dos correspondentes nós filhos, isso se a profundidade máxima da árvore, L , não tiver sido alcançada; caso contrário, o nó q é inserido no conjunto de folhas, \mathcal{C} . A Fig. 1 exemplifica essa construção para $\mathcal{C} = \{0, 11, 001, 101\}$. A partir da Fig. 1, podemos afirmar que as sequências 10 e 00 possuem o mesmo *morph*, logo o símbolo 0 é uma folha da CT.

C. Clusterização

Neste artigo aplicamos o algoritmo de aprendizado de máquina *K-Means* [10] para definir classes de estados com *morphs* similares do *D-Markov* e do VL_{MC}, que são empregadas como partições iniciais para o algoritmo de minimização dos algoritmos descritos nas próximas seções. O *K-Means* é empregado para particionar, em K clusters, M pontos de um espaço de dimensão I . Cada um dos K clusters é representado por um vetor $\mathbf{m}^{(k)}$ denominado centroide do cluster. Os pontos são representados por \mathbf{x}_n , com $n = 1, \dots, M$. Os centroides dos clusters são inicializados em pontos de \mathbb{R}^I e agrupam-se os pontos \mathbf{x}_n em torno dos centroides mais próximos. Em seguida, o valor do centroide é redefinido iterativamente de forma a minimizar a função objetiva $J_{obj} = \sum_{n=1}^M \sum_{k=1}^K r_{nk} \|\mathbf{x}_n - \mathbf{m}^{(k)}\|^2$, em que r_{nk} é igual a um se $k = \arg \min_j \|\mathbf{x}_n - \mathbf{m}^{(j)}\|^2$, caso contrário é igual a zero; isso implica em recalculer os centroides empregando a expressão $\mathbf{m}^{(k)} = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}}$ a cada nova iteração; o processo cessa quando os centroides ficam inalterados em iterações sucessivas. Neste trabalho os pontos \mathbf{x}_n correspondem aos *morphs* dos estados do *D-Markov* ou do VL_{MC}.

D. Algoritmo DCGraM

Pode-se reduzir o número de estados de um *D-Markov* identificando-se estados equivalentes e os colapsando. Entende-se por estados equivalentes aqueles com mesma distribuição de probabilidade para o conjunto de sequências gerados a partir destes. Dois aspectos práticos devem ser ressaltados: 1) Podem ser geradas infinitas sequências a partir de um estado, o que impede a determinação de equivalência

por comparação direta; 2) As probabilidades das sequências são estimadas a partir de séries temporais finitas, logo, as estimativas diferem do valor intrínseco (supomos que o processo estocástico é estacionário). Para lidar com essas questões de forma computacionalmente atrativa, foi proposto em [8] um algoritmo para construção de um PFSA denominado DCGraM (*D-Markov with Clustering and Graph Minimization*).

O DCGraM é dividido essencialmente em três passos: 1) Geração de um *D-Markov*, logo, pressupõe-se ou estima-se a memória do sistema igual a D ; 2) Aplicação do *K-Means* ao conjunto de *morphs* das sequências de comprimento D , identificando-se estados (a partir dos seus rótulos) com estatísticas de primeira ordem similares; 3) Aplicação de algum algoritmo de minimização de estados (e.g., o algoritmo de Moore ou o algoritmo de Hopcroft), o que permite identificar estados que gerem sequências de comprimento maior que dois com probabilidades distintas. Os PFSA gerados com o DCGraM possuem tipicamente menos estados que o *D-Markov* correspondente sem apresentar degradação na qualidade [5].

III. ALGORITMO VL-DCGRAM

Em caso de sistemas que precisem de valores elevados de D para capturar sua memória, o emprego de um *D-Markov* como PFSA de entrada para o DCGraM pode ser proibitivo. Observe que o número de estados obtidos com o *D-Markov* cresce exponencialmente com D . Para contornar essa limitação, propomos neste artigo o algoritmo VL-DCGraM (*Variable Length DCGraM*). Assim como o DCGraM, o VL-DCGraM é dividido em três etapas, diferenciando-se nas duas primeiras: 1) Construção de um VL_{MC} a partir da série temporal disponibilizada; 2) Aplicação do *K-Means* com média ponderada para cálculo dos centroides; 3) Aplicação de um algoritmo de minimização de estados. As propostas deste trabalho para as duas primeiras etapas são descritas a seguir.

A. Construção do VL_{MC}

Como contribuição deste trabalho, especificamos a seguir como gerar um VL_{MC} (Q, Σ, δ, π) a partir das folhas \mathcal{C} de uma CT especificada na Subseção II-B. Define-se o conjunto de estados inicial $Q = \mathcal{C}$ e calcula-se a função de transição $\delta(q, \sigma)$, para todo $q \in Q$ e $\sigma \in \Sigma$. Define-se $\delta(q, \sigma)$ como o estado em Q com rótulo igual ao máximo sufixo de $w\sigma$, sendo w o rótulo de q ; caso esse não exista, inclui-se um estado em Q com rótulo $w\sigma$, repetindo-se o cálculo. Caso w tenha comprimento igual à profundidade da árvore L e $w\sigma$ não possua sufixo entre os rótulos dos estados em Q , então um estado com rótulo igual ao máximo sufixo próprio de $w\sigma$ é incluído em Q . A partir da CT mostrada na Fig. 1 o grafo apresentado na Fig. 2 é construído. Observa-se que apesar da sequência 01 não rotular uma folha da CT, essa rotula um vértice do grafo. Sua inclusão se deve a inexistência no conjunto \mathcal{C} de um sufixo para a extensão de 0 pelo símbolo 1. Dessa forma, a construção do grafo segue com a inclusão da sequência 01 no conjunto Q . Os *morphs* dos estados são calculados como especificado em (3).

Para evitar desvios estatísticos da sequência gerada pelo PFSA, como consequência da presença de estados com alta

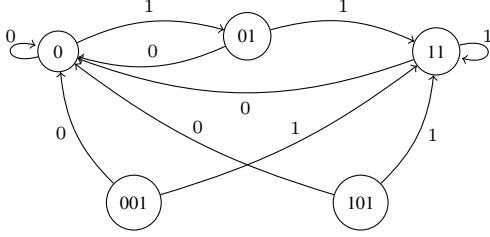


Fig. 2. Grafo obtido a partir da árvore de contexto da Fig. 1.

probabilidade de visitação se comparada com seus pares, o PFSA gerado passa por um processo de quebra de estados. Para um PFSA com $|Q|$ estados, define-se inicialmente o limiar l_μ pela equação

$$l_\mu = 10^\alpha \sum_{i=1}^N \mu_i \log_{10}(\mu_i) = 10^{-\alpha h}, \quad (4)$$

em que $\mu = (\mu_1, \dots, \mu_{|Q|})$ é o vetor estacionário do PFSA (que especifica, em regime estacionário, a probabilidade de visitação dos estados), α é um parâmetro a ser determinado empiricamente e h é a entropia calculada sobre a distribuição de probabilidade de visitação dos estados. Nessa etapa também é especificado um comprimento máximo para o rótulo de um estado, M . Sempre que um estado q_i com rótulo w possuir probabilidade de visitação μ_i superior ou igual a l_μ , então esse será substituído no conjunto Q por estados com rótulos $\sigma_i w$, $\sigma_i \in \Sigma$, caso $|w| < M$; contudo, se $|w| = M$, então q_i permanece no conjunto Q . Para os novos estados, a função de transição é calculada. Esse processo segue até que todos os estados com rótulos de comprimentos inferiores a M tenham probabilidade de visitação menor que l_μ .

K-Means com média ponderada

No VL-DCGraM a expressão da função objetiva empregada no *K*-Means é

$$J_{pond} = \sum_{n=1}^M \sum_{k=1}^K r_{nk} p_n \| \mathbf{x}_n - \mathbf{m}^{(k)} \|^2, \quad (5)$$

em que p_n é a probabilidade do estado do PFSA com morf \mathbf{x}_n , logo $\mathbf{x}_n = \mathcal{V}(q_n)$ para um estado q_n com probabilidade p_n . Como no caso apresentado na Seção II-C, para minimizar J_{pond} , a cada nova iteração os valores dos centroides são recalculados com a expressão

$$\mathbf{m}^{(k)} = \frac{\sum_{n=1}^M r_{nk} p_n \mathbf{x}_n}{\sum_{n=1}^M r_{nk} p_n}, \quad (6)$$

que é a média ponderada dos pontos no k -ésimo *cluster*.

IV. QUANTIFICADORES DE DESEMPENHO

O primeiro quantificador de desempenho de um PFSA é a entropia condicional de ordem ℓ dada por $h_\ell \triangleq H(X_\ell | X_1 X_2 \dots X_{\ell-1})$, em que $H(X|Y)$ é a entropia condicional de X dado Y [14]. O objetivo com o cálculo de h_ℓ é verificar se o PFSA é capaz de capturar a memória da série temporal. Em seguida empregaremos a divergência de Kullback-Leiber de ordem ℓ , D_ℓ , para avaliar a semelhança

entre as distribuições de probabilidade das sequências de comprimento ℓ geradas pela série temporal $P_1(\omega)$ e pelo PFSA $P_2(\omega)$, dada por

$$D_\ell = \sum_{\omega \in \Sigma^\ell} P_1(\omega) \log \left(\frac{P_1(\omega)}{P_2(\omega)} \right). \quad (7)$$

Por fim, a função autocorrelação ($R[m] = E[Z_k Z_{k+m}]$, em que $E[\cdot]$ indica o valor esperado) das sequências geradas pelos PFSA são comparadas com a obtida da série temporal. No caso da série temporal o cálculo é feito por simulação, enquanto que para os PFSA se emprega-se a matriz de transição de probabilidade, como detalhado em [15].

V. RESULTADOS

Nesta seção o VL-DCGraM é comparado com o *D*-Markov e o DCGraM. O sistema considerado é o canal com desvanecimento discreto (DFC, *discrete fading channel*). A entrada é um processo binário i.i.d. $\{X_k\}_{k=1}^\infty$, $X_k \in \{0,1\}$, e a saída também é binária $\{Y_k\}_{k=1}^\infty$, $Y_k \in \{0,1\}$. O DFC é composto por um modulador BPSK, um canal com desvanecimento plano Rayleigh correlacionado no tempo, ruído AWGN e um demodulador coerente quantizado. O símbolo recebido no k -ésimo intervalo de sinalização é escrito como

$$R_k = \sqrt{E_s} A_k S_k + N_k, \quad k = 1, 2, \dots, \quad (8)$$

em que $\{S_k\} = \{(2X_k - 1)\}$, E_s é a energia do sinal transmitido e $\{N_k\}$ é a k -ésima amostra do ruído AWGN com variância $N_0/2$. A relação sinal-ruído (SNR) é definida como E_s/N_0 . Além disso, $\{A_k\}$ é o processo de desvanecimento com $A_k = |G_k|$, em que $\{G_k\}$ é um processo Gaussiano complexo com média zero, variância unitária e função autocorrelação de Clarke $R[m] = J_0(2\pi f_D T |m|)$ [16], em que $J_0(x)$ é a função de Bessel de primeira espécie de ordem zero e $f_D T$ é a máxima frequência Doppler normalizada. A variável aleatória A_k possui função densidade de probabilidade $p_A(a) = 2ae^{-a^2}$, com segundo momento unitário para $a > 0$. O símbolo de saída é $Y_k = 0$ se $R_k \leq 0$ e $Y_k = 1$ se $R_k > 0$.

Os símbolos de entrada e saída são relacionados por um símbolo de ruído $Z_k \in \{0,1\}$ através da expressão $Y_k = X_k \oplus Z_k$, em que \oplus denota adição módulo 2. Os símbolos X_k e Z_k são estatisticamente independentes. Para as simulações é empregado um DFC com parâmetros $f_D T = 0,005$ e SNR = 10 dB. A sequência binária de ruído $\{Z_k\}$ empregada é gerada por simulação e tem comprimento $N = 5 \times 10^7$. A partir desta sequência, PFSA são gerados pelos algoritmos *D*-Markov, DCGraM e VL-DCGraM.

Os quantificadores abordados na Seção IV são utilizados para avaliar o desempenho relativo dos PFSA gerados. Os algoritmos *D*-Markov e DCGraM usam como parâmetros *D* de 4 a 9 e, para este último, é empregado $K = 5$ no *K*-Means. Para o VL-DCGraM, $M = 15$ e L de 4 a 9; ainda, o limiar adotado em todos os casos é $T = 0,05$ (definido na Subseção II-B), $\alpha = 2$, ao passo que $K = 5$ no *K*-Means.

A Fig. 3 apresenta o divergente de Kullback-Leiber de ordem quinze versus o número de estados. Observa-se que o VL-DCGraM é capaz de capturar a distribuição de probabilidade das subsequências de tamanho quinze da sequência

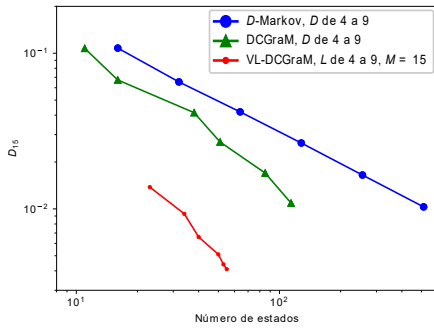


Fig. 3. Divergente de Kullback-Leiber de ordem quinze entre a série temporal e os PFSA's gerados pelos algoritmos D -Markov, DCGraM ($4 \leq D \leq 9$) e VL-DCGraM ($M = 15$ e $4 \leq L \leq 9$).

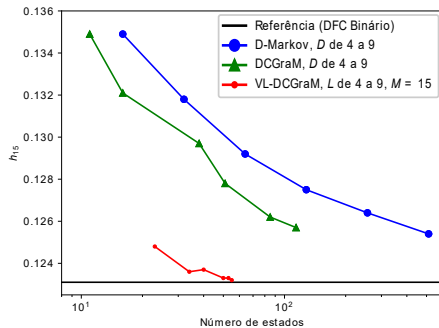


Fig. 4. Entropia condicional de ordem quinze para D -Markov, DCGraM ($4 \leq D \leq 9$) e VL-DCGraM ($M = 15$ e $4 \leq L \leq 9$).

$\{Z_k\}$ de forma mais fidedigna que o D -Markov e o DCGraM. A partir de $L = 5$, o VL-DCGraM obteve D_{15} inferior ao de todos DCGraM mostrados na figura, sempre com uma redução significativa no número de estados. No melhor caso $D_{15} = 4 \times 10^{-3}$, quando $L = 9$. Na Fig. 4, em que temos a entropia condicional de ordem quinze em função do número de estados, novamente, o PFSA gerado pelo VL-DCGraM possui um comportamento superior aos demais. Nesse caso, vale destacar que esse foi o único capaz de gerar o valor de h_{15} que coincide com o da sequência $\{Z_k\}$; que corresponde a linha horizontal na figura. Isso indica que para as condições de teste, o VL-DCGraM capturou a memória da série temporal até comprimento quinze, para $L = 9$; o que não foi observado para o D -Markov e DCGraM para nenhum dos valores de D . Por fim, na Fig. 5 são apresentadas as curvas da função autocorrelação para D -Markov, DCGraM com $D = 9$, VL-DCGraM com $M = 15$ e $L = 9$ (mantidos os demais parâmetros) e DFC. Inicialmente, observa-se que o VL-DCGraM gera um PFSA com menos de 50% do número de estados dos demais e, ainda assim, a curva adere visivelmente melhor a do DFC em toda a sua extensão.

VI. CONCLUSÕES

Neste trabalho apresentamos um novo algoritmo para a determinação de PFSA's a partir de séries temporais. Empregando técnicas de inferência estatística, aprendizado de máquina não

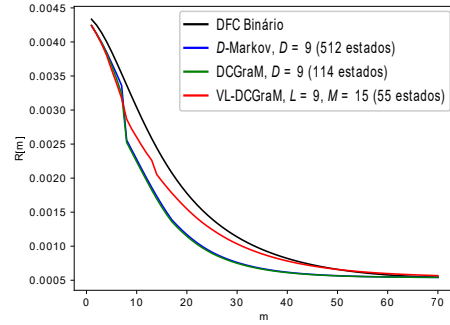


Fig. 5. Função autocorrelação da série temporal (DFC binário) e do D -Markov, DCGraM ($D = 9$) e VL-DCGraM ($M = 15$ e $L = 9$).

supervisionado e teoria de autômatos, obtém-se um modelo markoviano da dinâmica subjacente às sequências simbólicas com desempenho superior ao de outras propostas da literatura. Como melhorias, propomos a determinação analítica dos parâmetros T , L e M empregados na etapa um do algoritmo.

REFERÊNCIAS

- [1] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques — Part I: Fault diagnosis with model-based and signal-based approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, Jun. 2015.
- [2] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis - a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, April 2014.
- [3] E. Bradley and H. Kantz, "Nonlinear time-series analysis revisited," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 25, no. 9, p. 097610, 2015.
- [4] F. Li, H. Wang, G. Zhou, D. Yu, J. Li, and H. Gao, "Anomaly detection in gas turbine fuel systems using a sequential symbolic method," *Energies*, vol. 10, no. 5, pp. 1–22, May 2017.
- [5] K. Mukherjee and A. Ray, "State splitting and merging in probabilistic finite state automata for signal representation and analysis," *Signal Processing*, vol. 104, pp. 105–119, April 2014.
- [6] C. R. Shalizi and K. L. Shalizi, "Blind construction of optimal nonlinear recursive predictors for discrete sequences," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, 2004, pp. 504–511.
- [7] I. Chattopadhyay, Y. Wen, A. Ray, and S. Phoha, "Unsupervised inductive learning in symbolic sequences via recursive identification of self-similar semantics," in *Proceedings of the 2011 American Control Conference*, 2011, pp. 125–130.
- [8] D. K. Franch, D. P. Chaves, C. Pimentel, and D. M. Hamilton, "Markov modeling of dynamical systems via clustering and graph minimization," *Digital Signal Processing*, p. 102769, 2020.
- [9] D. Lind and B. Marcus, *An Introduction To Symbolic Dynamics and Codings*. Cambridge University Press, 1995.
- [10] C. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. Springer New York, 2016.
- [11] J. Berstel, L. Boasson, O. Carton, and I. Fagnot, "Minimization of automata," *arXiv:1010.5318*, 2010.
- [12] P. Bühlmann, "Model selection for variable length Markov chains and tuning the context algorithm," *Annals of the Institute of Statistical Mathematics*, vol. 52, pp. 287–315, 2000.
- [13] F. Babich, O. E. Kelly, and G. Lombardi, "A context-tree based model for quantized fading," *IEEE Communications Letters*, vol. 3, no. 2, pp. 46–48, Feb. 1999.
- [14] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 2012.
- [15] C. Pimentel and I. F. Blake, "Enumeration of Markov chains and burst error statistics for finite state channel models," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 2, pp. 415–428, 1999.
- [16] R. Clarke, "A statistical theory of mobile-radio reception," *Bell Labs Technical Journal*, vol. 47, no. 6, pp. 957–1000, 1968.