

Proposta de uma Função de Previsão de Dados Temporais para o Algoritmo dos Vizinhos mais Próximos

Carlos Andres Ferrero^{1,2}, Maria Carolina Monard¹,
Huei Diana Lee², and Wu Feng Chung²

¹ Instituto de Ciências Matemáticas e de Computação – ICMC/USP
Laboratório de Inteligência Computacional – LABIC
Caixa Postal 668, 13560-970 – São Carlos, SP, Brasil

² Centro de Engenharias e Ciências Exatas – CECE/UNIOESTE
Laboratório de Bioinformática – LABI
Caixa Postal 39, 85856-970 – Foz do Iguaçu, PR, Brasil

anfer86@gmail.com, mcmonard@icmc.usp.br, {hueidianalee, wufengchung}@gmail.com

Abstract. *The k -Nearest Neighbor - Time Series Prediction — k NN-TSP — algorithm is an approximator based on the non-linear model approach for time series prediction problems. However, some issues related to how to determine the algorithm's parameters still remain. In this work, we focus in the prediction function parameter. To this end, we propose a prediction function which is not disturbed in the presence of patterns at different levels of the time series. This function was empirically evaluated on several artificial time series, including chaotic time series, comparing the results with the ones obtained using a frequently used prediction function. Results show that in most cases the proposed prediction function is significantly better than the other function.*

Key words: *Nearest neighbors, non-linear prediction, chaotic data*

1 Introdução

A previsão de dados ou séries temporais permite prever dados desconhecidos a partir de um conjunto de informações conhecidas. Para isso, têm sido propostas abordagens lineares e não-lineares para realizar a previsão. Em geral, abordagens lineares assumem que os dados respeitam alguma distribuição estatística e, com base nessa informação, são definidos parâmetros para ajustar o modelo aos dados. Porém, uma grande parte dos dados temporais, no mundo real, envolve fenômenos naturais, os quais são não-lineares. As abordagens para modelagem não-linear, também denominadas de regressões não-paramétricas, são comumente classificadas como globais e locais [1]. As primeiras utilizam a série temporal inteira para a construção de um modelo que represente toda a série, enquanto que as segundas utilizam somente um subconjunto de sequências, consideradas mais importantes, para estimar o valor futuro.

Uma das estratégias para contornar o problema de previsão local para comportamentos não-lineares trata da adaptação do algoritmo de aprendizado de máquina *k-Nearest Neighbor* — *kNN* — para a previsão de dados temporais [2]. O algoritmo *kNN* para classificação foi proposto em [3] e consiste em prever a classe, ou rótulo, de um novo exemplo com base em exemplos similares já rotulados. A adaptação desse algoritmo para dados temporais é denominada, neste trabalho, *k-Nearest Neighbor - Time Series Prediction* — *kNN-TSP*, apresentando como uma das principais vantagens a sua simplicidade [4]. Entretanto, ainda que ele frequentemente apresente melhores previsões para séries temporais parcialmente periódicas, várias questões relacionadas aos seus parâmetros continuam em aberto.

Neste trabalho é proposta uma função de previsão que tem a propriedade de manter bom desempenho na presença de padrões em níveis diferentes da série temporal. A função proposta foi avaliada experimentalmente e comparada com uma abordagem convencional frequentemente utilizada na literatura [5]. Nessas avaliações, foram usados diversos tipos de séries temporais artificiais, inclusive séries caóticas. Os resultados obtidos, na grande maioria dos casos, mostram que a função proposta é superior, com diferenças estatisticamente significativas.

O trabalho está organizado da seguinte forma: na Seção 2, é descrito o algoritmo para previsão de dados temporais; na Seção 3, é apresentada a configuração para a realização dos experimentos; na Seção 4, é descrita a avaliação experimental, bem como os resultados e a discussão; e, na Seção 5, são apresentadas as conclusões deste trabalho.

2 Algoritmo *k-Nearest Neighbor - Time Series Prediction*

Para prever valores futuros de uma série temporal é necessário adaptar a abordagem convencional do algoritmo *kNN*, que calcula valores discretos, para previsão de dados contínuos [5]. Neste trabalho, essa adaptação deve considerar que os dados provêm de séries temporais, representadas por $X = (x_1, x_2, \dots, x_n)$ de tamanho n , em que cada elemento x_t da série indica o valor amostrado. A ideia do algoritmo *kNN-TSP* consiste em, considerando os últimos w registros ocorridos, os quais constituem a sequência de referência, encontrar as sequências de tamanho w que apresentaram comportamentos similares no passado. Com base nas informações dessas sequências, é realizado o cálculo do valor futuro \hat{x}_{n+1} , que é uma aproximação do verdadeiro, mas desconhecido, valor x_{n+1} da série temporal. Os parâmetros de *kNN-TSP*, ilustrados na Figura 1, são:

- (a) tamanho w da janela para extrair as sequências;
- (b) conjunto de exemplos de treinamento;
- (c) medida de similaridade;
- (d) cardinalidade do conjunto de vizinhos mais próximos;
- (e) função de previsão.

Assim, para a execução do *kNN-TSP* devem ser levadas em consideração as seguintes três fases: (1) preparação do conjunto de exemplos de treinamento, (2)

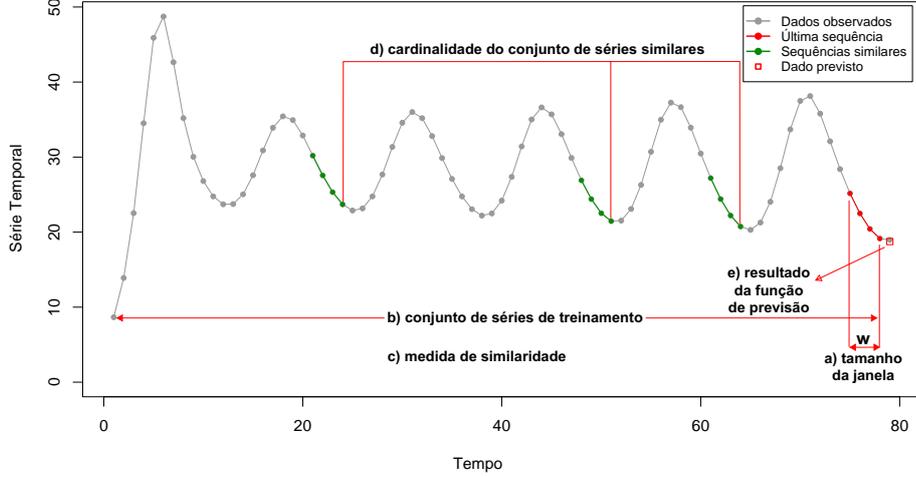


Figura 1. Parâmetros do algoritmo *kNN-TSP*

obtenção dos vizinhos mais próximos e (3) cálculo do valor futuro da sequência de referência, as quais são descritas a seguir.

2.1 Fase 1 — Preparação do conjunto de exemplos de treinamento

Para construir o conjunto de exemplos de treinamento é utilizada uma janela de tamanho w , de modo a extrair da série sequências de tamanho fixo, que representem parte do comportamento dessa série. Seja a série temporal $X = (x_1, x_2, \dots, x_n)$ de tamanho n . Seja o par ordenado (\mathbf{x}_n, x_{n+1}) o exemplo de referência, em que \mathbf{x}_n corresponde à sequência de referência, *i.e.*, aos últimos w valores de X , $(x_{n-(w-1)}, x_{n-(w-2)}, \dots, x_n)$, e \hat{x}_{n+1} ao valor futuro da sequência de referência, *i.e.*, ao valor a ser previsto. Utilizando essa notação, cada elemento do conjunto de exemplos de treinamento $S = \{S_w, S_{w+1}, S_{w+2}, \dots, S_{n-1}\}$ é definido pelo conjunto de Equações 1. Desse modo, cada elemento de $S_i \in S$ consiste em um par ordenado (\mathbf{x}_i, x_{i+1}) , onde a primeira componente corresponde à i -ésima sequência de treinamento, dada pela série $(x_{i-(w-1)}, x_{i-(w-2)}, \dots, x_i)$, e a segunda refere-se ao valor futuro da sequência de treinamento (classe).

2.2 Fase 2 — Obtenção dos vizinhos mais próximos

Considerando o par (\mathbf{x}_n, x_{n+1}) , que contém a sequência de referência, são procurados em S os exemplos das sequências de treinamento que descrevam comportamento semelhante ao da sequência de referência. O objetivo consiste em selecionar os vizinhos próximos a serem utilizados para estimar x_{n+1} . Para isso, devem ser levados em consideração a cardinalidade k do conjunto de vizinhos mais próximos a serem utilizados na estimativa do valor futuro, bem como o critério de seleção de vizinhos próximos.

$$\begin{aligned}
S_w &= (\mathbf{x}_w, x_{w+1}) = ((x_1, x_2, \dots, x_w), x_{w+1}) \\
S_{w+1} &= (\mathbf{x}_{w+1}, x_{w+2}) = ((x_2, x_3, \dots, x_{w+1}), x_{w+2}) \\
&\vdots \\
S_{w+i} &= (\mathbf{x}_{w+i}, x_{w+i+1}) = ((x_{i+1}, x_{i+2}, \dots, x_{w+i}), x_{w+i+1}) \\
&\vdots \\
S_{w+j} &= (\mathbf{x}_{w+j}, x_{w+j+1}) = ((x_{j+1}, x_{j+2}, \dots, x_{w+j}), x_{w+j+1}) \\
&\vdots \\
S_{n-1} &= (\mathbf{x}_{n-1}, x_n) = ((x_{n-w}, x_{n-(w-1)}, \dots, x_{n-1}), x_n)
\end{aligned} \tag{1}$$

O valor de k pode ser definido *a priori* ou pode ser utilizado um limiar de similaridade l , tal que, todos os exemplos das sequências de treinamento com valor de similaridade maior que l são considerados para compor o conjunto de vizinhos mais próximos. Em relação à segunda questão, deve ser definido o critério de seleção de vizinhos próximos a serem utilizados para a previsão do valor futuro. Geralmente utiliza-se uma medida de similaridade, no intuito de encontrar os exemplos de treinamento que mais se assemelham ao comportamento da sequência de referência. Diversas medidas têm sido propostas na literatura e a combinação de várias dessas medidas constitui uma área interesse [6]. No entanto, de acordo com trabalho relacionado [7], a distância Euclidiana é a mais frequentemente utilizada, devido ao fato de ser uma medida intuitiva e de baixo custo computacional. Essa medida determina a distância entre dois pontos no espaço \mathfrak{R}^w , no qual w corresponde ao tamanho das sequências.

O resultado da execução desta fase consiste na definição do subconjunto $S' = \{S'_1, S'_2, \dots, S'_k\} \subset S$, tal que $|S'| = k$ e a série S'_i é um par ordenado $(s_{i,w}, s_{i,w+1})$ que corresponde ao exemplo de treinamento que contém a i -ésima sequência mais próxima à sequência de referência \mathbf{x}_n . Por exemplo, considerando o conjunto de exemplos de treinamento definido na Seção 2.1 utilizado para estimar o valor x_{n+1} , para o qual foram encontrados os seguintes três vizinhos mais próximos $\{S_{w+1}, S_{w+i}, S_{w+j}\}$, então $S' = \{S'_1, S'_2, S'_3\}$ verifica o conjunto de Equações 2.

$$\begin{aligned}
S'_1 &= ((s'_{1,1}, s'_{1,2}, \dots, s'_{1,w}), s'_{1,w+1}) = S_{w+1} = ((x_2, x_3, \dots, x_{w+1}), x_{w+2}) \\
S'_2 &= ((s'_{2,1}, s'_{2,2}, \dots, s'_{2,w}), s'_{2,w+1}) = S_{w+i} = ((x_{i+1}, x_{i+2}, \dots, x_{w+i}), x_{w+i+1}) \\
S'_3 &= ((s'_{3,1}, s'_{3,2}, \dots, s'_{3,w}), s'_{3,w+1}) = S_{w+j} = ((x_{j+1}, x_{j+2}, \dots, x_{w+j}), x_{w+j+1})
\end{aligned} \tag{2}$$

2.3 Fase 3 — Cálculo do valor futuro da sequência de referência

Nesta fase, a função de previsão $f(S')$ é responsável pela estimativa \hat{x}_{n+1} do valor futuro da sequência de referência, que é desconhecido. As funções de previsão

comumente utilizadas aproximam esse valor pela média local ou pela média ponderada dos valores da classe, dados por $s'_{i,w+1}$ de cada sequência $S'_i \in S'$ [1]. Diversos critérios têm sido propostos na literatura para definir os pesos dos $s'_{i,w+1}$, os quais podem ser combinados linearmente ou utilizando, por exemplo, uma função exponencial [8]. A função de previsão Média de Valores — MV — f_{MV} , comumente utilizada na literatura, utiliza a média local para calcular \hat{x}_{n+1} e é definida pela Equação 3.

$$f_{MV}(S') = \frac{\sum_{i=1}^k s'_{i,w+1}}{k} = \hat{x}_{n+1} \quad (3)$$

Neste trabalho, propõe-se a função de previsão que foi denominada Média de Valores Relativos — MVR — f_{MVR} , a qual calcula \hat{x}_{n+1} utilizando o valor de x_n acrescido da média local da diferença dos valores da classe, $s'_{i,w+1}$, e o valor $s'_{i,w}$ de cada exemplo $S'_i \in S'$. A função é definida pela Equação 4.

$$f_{MVR}(S') = x_n + \frac{\sum_{i=1}^k \Delta s'_{i,w+1}}{k} = \hat{x}_{n+1} \quad (4)$$

onde $\Delta s'_{i,w+1} = s'_{i,w+1} - s'_{i,w}$.

Uma das vantagens da função f_{MVR} em relação à função f_{MV} é que ela permite prever valores futuros considerando padrões localizados em níveis diferentes de tendência.

2.4 Descrição do Algoritmo

O Algoritmo 1 descreve o pseudocódigo do algoritmo *kNN-TSP*, onde

- $X = (x_1, x_2, \dots, x_n)$ é a série temporal que gera o conjunto de exemplos de treinamento e a sequência de referência;
- w é o tamanho da janela para extrair as sequências;
- S é o conjunto de exemplos de treinamento;
- \mathbf{x}_n é a sequência de referência;
- M_s é a medida de similaridade;
- C_k é o critério de seleção dos vizinhos mais próximos;
- k é o número de vizinhos mais próximos;
- S' é o subconjunto de exemplos de treinamento referentes às sequências mais próximas a \mathbf{x}_n ; e
- f é a função de previsão e \hat{x}_{n+1} é o valor futuro a ser calculado do exemplo de referência.

A complexidade da função de preparação do conjunto de exemplos de treinamento, $exemplos_de_treinamento(X, w)$, é da ordem de $O(n)$. Em relação à função $vizinhos_próximos(S, \mathbf{x}_n, M_s, C_k, k)$, a qual é responsável por encontrar os k vizinhos mais próximos da sequência de referência \mathbf{x}_n , a complexidade depende da medida de similaridade utilizada. Usando a distância Euclidiana, a complexidade dessa função é $O(n \times w)$. A função de previsão, $f(S')$, apresenta complexidade $O(k)$. Desse modo, a complexidade do algoritmo *kNN-TSP* é da ordem de $O(n \times w)$.

Algorithm 1: *kNN-TSP*

Input: X, w, M_s, C_k, k, f **Output:** \hat{x}_{n+1} // Construir o conjunto S a partir da série X e w $S \leftarrow \text{exemplos_de_treinamento}(X, w);$ // Obtenção dos k exemplos de treinamento referentes às sequências mais próximas a \mathbf{x}_n contidos em S , considerando M_s e C_k $S' \leftarrow \text{vizinhos_próximos}(S, \mathbf{x}_n, M_s, C_k, k);$ // Cálculo do valor futuro do exemplo de referência, utilizando $f(S')$ $\hat{x}_{n+1} \leftarrow f(S');$ **Return** $\hat{x}_{n+1};$

3 Configuração dos Experimentos

O método para avaliar a qualidade do algoritmo *kNN-TSP* consiste das seguintes três etapas:

Etapa 1 — Pré-processamento de séries temporais: responsável pela representação das séries temporais no formato adequado para a aplicação do algoritmo *kNN-TSP*.

Etapa 2 — Configuração de experimentos e previsão: inicialmente é definida a quantidade de valores a serem previstos. Com essa informação, são definidos os dados a serem utilizados pelo algoritmo *kNN-TSP* para prever cada valor futuro. A abordagem utilizada neste trabalho consiste em prever um valor futuro utilizando apenas valores reais no conjunto de treinamento. Ou seja, dada a série temporal $ST = (x_1, x_2, \dots, x_n)$, o *kNN-TSP* prevê o valor \hat{x}_{n+1} de x_{n+1} . Entretanto, para avaliar o algoritmo é necessário conhecer o valor de x_{n+1} para medir o erro entre esse valor real x_{n+1} e o valor previsto \hat{x}_{n+1} . Esse processo deve ser repetido um certo número m de vezes, com o objetivo de obter uma amostra mais representativa dos erros de previsão do algoritmo. Isso pode ser realizado gerando m pares ordenados $(\mathbf{X}_i, x_{n-(m-i)})$, em que o primeiro termo é a série temporal que gera o conjunto de exemplos de treinamento e o segundo termo refere-se à classe. Assim, o algoritmo *kNN-TSP* deverá ser executado m vezes, utilizando como série temporal para construir o conjunto de sequências de treinamento, em cada iteração i , a série \mathbf{X}_i para estimar os m últimos valores da série ST .

Etapa 3 — Avaliação de resultados e pós-processamento: é realizado o cálculo das medidas para quantificar o desempenho dos algoritmos de previsão. Considerando $ST = (x_1, x_2, \dots, x_n)$ a série temporal de valores observados, e $\hat{X} = (\hat{x}_{n-(m-1)}, \hat{x}_{n-(m-2)}, \dots, \hat{x}_n)$ os m valores estimados utilizando o *kNN-TSP*, são calculados o Erro Médio Absoluto (EMA) e o coeficiente de correlação de Spearman (r) [9]. O EMA permite quantificar a diferença média entre os valores observados e os previstos e, o r , a correlação entre os valores observados e os previstos.

A utilização dessas medidas permite comparar objetivamente diferentes configurações do algoritmo. Neste trabalho, as comparações são realizadas utilizando o teste estatístico não-paramétrico de Wilcoxon para amostras emparelhadas [9].

4 Avaliação Experimental

A avaliação do algoritmo $kNN-TSP$ foi realizada empiricamente, utilizando conjuntos de dados que consistem de séries temporais artificiais, *i.e.*, geradas por funções matemáticas, de modo a realizar uma avaliação focada no valor de k , de número de séries similares consideradas na previsão e nas funções de previsão. Os dados foram gerados utilizando modelos (funções) pertencentes a duas famílias: (A) séries temporais de modelos sazonais e (B) sistemas caóticos. Os modelos da família (A) contêm tendência e/ou mudança de amplitude ao longo do tempo e são definidos pelas seguintes séries temporais, extraídas de [5]: (A.1) dependência sazonal, (A.2) sazonalidade multiplicativa e (A.3) alta frequência. Os modelos da família (B) geram séries com comportamento pouco previsível, com ciclos que não se repetem. A avaliação por meio desse tipo de séries é importante, devido ao fato de que parte das séries temporais que representam eventos naturais apresentam comportamento caótico. Neste trabalho, são utilizados os sistemas de (B.1) Lorenz e (B.2) Mackey-Glass, para gerar séries temporais com comportamento caótico.

Cada série temporal foi submetida ao método apresentado na Seção 3. Para cada conjunto de dados é utilizado um valor, definido em [5], do parâmetro (a) — tamanho w da janela. Deve ser observado que não é foco deste trabalho a investigação da influência da dimensão de imersão da série temporal na previsão de dados. Quanto ao parâmetro (b), conjunto de exemplos de treinamento, foi construído utilizando os elementos da série somente até, mas sem incluir, o valor a ser previsto. Na Tabela 1 são apresentados o tamanho (número de pontos), os valores do parâmetro (a) e a quantidade de valores previstos, m , para cada série considerada.

Tabela 1. Configuração do parâmetro (a) e do número de valores previstos

Série Temporal	Tamanho	(a)	#Valores previstos
(A.1)	2200	100	220
(A.2)	590	15	88
(A.3)	550	70	55
(B.1)	2050	25	100
(B.2)	2050	7	100

Em relação ao parâmetro (c), medida de similaridade, foi utilizada a distância Euclidiana e para o parâmetro (d) foram considerados os valores de cardinalidade $k = 1, 2, 3, 4$ e 5 . Em relação ao parâmetro (e) foram utilizadas as funções de previsão f_{MV} e f_{MVR} .

O desempenho do algoritmo $kNN-TSP$ foi avaliado baseado nas funções de previsão. As comparações foram realizadas utilizando o teste não-paramétrico de Wilcoxon para dados emparelhados, considerando nível de significância de 95%, no intuito de verificar a existência de diferença estatisticamente significativa — **d.e.s** — nos resultados. Os gráficos das previsões para cada série temporal, nas diversas configurações, encontram-se no documento $kNN-TSP$ Resultados Experimentais [10]. Na Tabela 2 são apresentados os resultados da comparação entre as funções f_{MV} e f_{MVR} , em que o símbolo \checkmark indica os resultados com **d.e.s**. Deve ser observado que essas **d.e.s** são válidas para todos os valores de $k = 1, 2, 3, 4$ e 5 , ainda que, dependendo da série, alguns resultados foram superiores para certos valores específicos de k .

Tabela 2. Resumo das comparações entre as funções f_{MV} e f_{MVR}

Série Temporal	EMA	r	Melhor desempenho
Dependência sazonal	\checkmark	\checkmark	f_{MVR}
Sazonalidade multiplicativa	\checkmark		f_{MVR}
Alta frequência			
Lorenz	\checkmark	\checkmark	f_{MVR}
Mackey-Glass	\checkmark	\checkmark	f_{MVR}

Para ilustrar, na Figura 2 são apresentados os primeiros 40 valores de previsão de acordo com as funções f_{MV} e f_{MVR} , para cada série temporal considerada, utilizando $k = 1$.

De acordo com esses resultados pode ser constatado que a eficiência do algoritmo de previsão depende das características de cada série temporal. Devido às séries temporais artificiais utilizadas apresentarem padrões similares em diferentes níveis, de modo geral, a função f_{MVR} apresentou melhor desempenho em relação à função f_{MV} . Especificamente as séries de dependência sazonal, sazonalidade multiplicativa e alta frequência apresentam padrões ao longo de tendências, motivando a aplicação da função f_{MVR} no intuito de reproduzir esses padrões mais adequadamente. As séries temporais de Lorenz e Mackey-Glass não apresentam padrões ao longo de uma linha de tendência específica, porém, as sequências similares foram encontradas, em geral, em diferentes níveis, o que torna interessante a aplicação da função f_{MVR} . Entretanto, a variação local de amplitude ao longo do tempo pode estar influenciando esses resultados, devido a que essa característica não foi considerada pela função de previsão, constituindo uma limitação da função f_{MVR} .

5 Conclusão

Neste trabalho foi apresentado o algoritmo $kNN-TSP$ e várias questões relacionadas ao problema de aprendizado desse algoritmo foram consideradas para o problema de previsão de dados temporais. Foi proposta uma nova função de

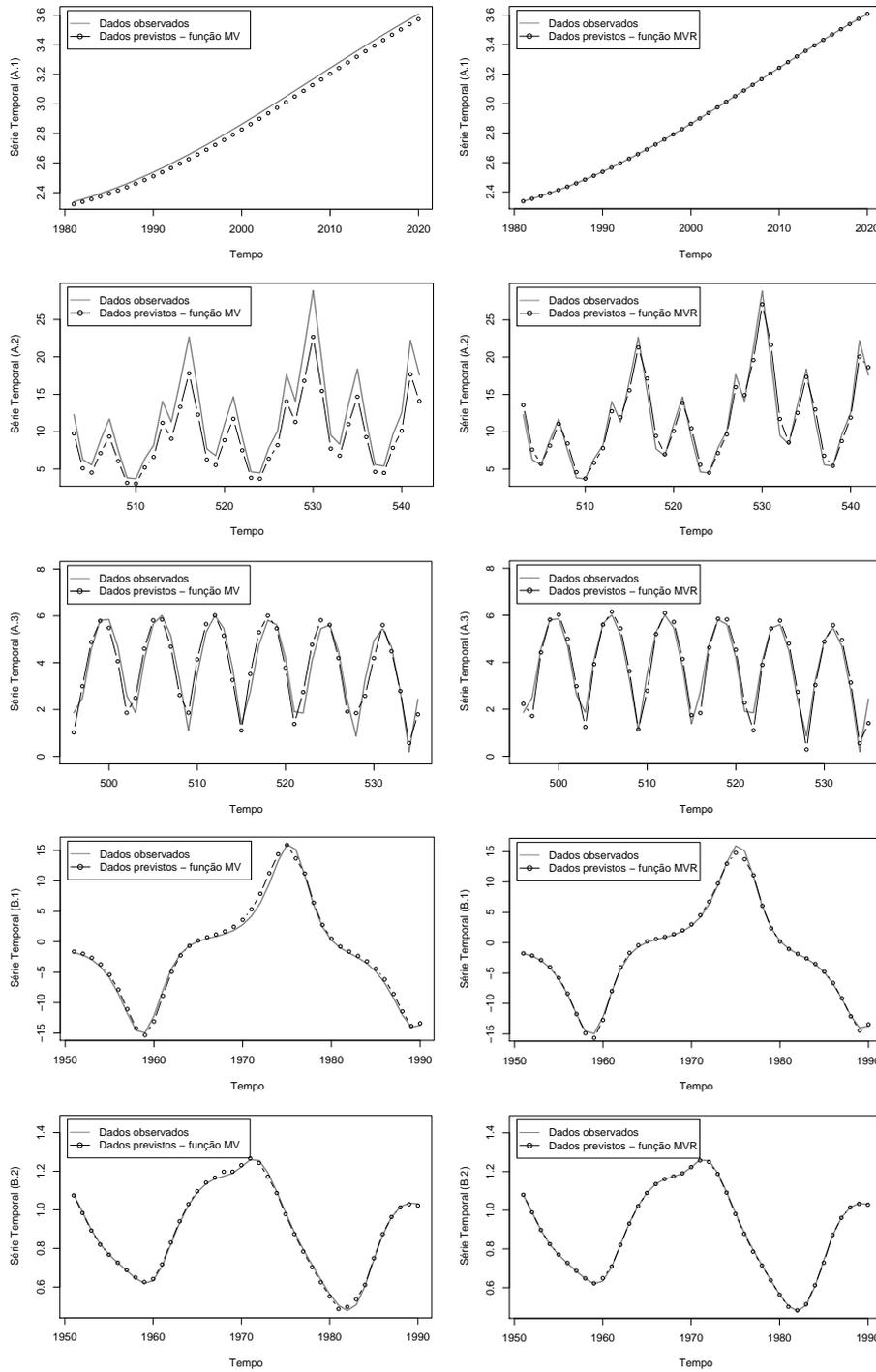


Figura 2. Gráficos das previsões usando o critério de seleção por similaridade e as funções de previsão f_{MV} e f_{MVR}

previsão no intuito de contornar o problema de previsão de dados na abordagem tradicional desse algoritmo quando as séries temporais apresentam padrões em diferentes níveis de tendência. De acordo com a avaliação experimental realizada, para as séries temporais de dependência sazonal, sazonalidade multiplicativa, Lorenz e Mackey-Glass foi possível constatar com **d.e.s** em que a função f_{MVR} proposta neste trabalho apresentou melhor desempenho em relação à função f_{MV} . Em relação à série de alta frequência não foi possível evidenciar **d.e.s** entre o desempenho de ambas as funções.

Trabalhos futuros incluem a utilização de outros critérios de seleção de vizinhos próximos que permitam selecionar sequências que resultem em previsões mais precisas; a utilização do cálculo de média ponderada na função f_{MVR} ; e a aplicação de $kNN-TSP$ a dados temporais reais.

Agradecimentos

À Fundação Parque Tecnológico Itaipu — FPTI-BR — pelo apoio por meio da linha de financiamento de bolsas e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico — CNPq. Aos revisores pelas sugestões.

Referências

1. D. S. K. Karunasinghe, S.-Y. Liong, Chaotic time series prediction with a global model: Artificial neural network, *Journal of Hydrology* 323 (1-4) (2006) 92–105.
2. J. McNames, Innovations in local modeling for time series prediction, Ph.D. thesis, Stanford University (1999).
3. D. W. Aha, D. Kibler, M. K. Albert, Instance-based learning algorithms, *Machine Learning* 6 (1991) 37–66.
4. C. A. Ferrero, M. C. Monard, H. D. Lee, S. F. Benassi, F. C. Wu, Previsão da temperatura da água no reservatório de Itaipu utilizando o método não-linear k -Nearest Neighbor, in: III Congresso da Academia Trinacional de Ciências, Foz do Iguaçu — PR, Brasil, 2008, pp. 1–10.
5. M. Kulesh, M. Holschneider, K. Kurennaya, Adaptive metrics in the nearest neighbours method, *Physica D: Nonlinear Phenomena* 237 (3) (2008) 283–291.
6. F. Fabris, I. Drago, F. M. Varejão, A multi-measure nearest neighbor algorithm for time series classification, in: IBERAMIA '08: Proceedings of the 11th Ibero-American Conference on AI, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 153–162.
7. E. Keogh, S. Kasetty, On the need for time series data mining benchmarks: a survey and empirical demonstration, in: Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining, New York, USA, 2002, pp. 102–110.
8. D. Solomatine, M. Maskey, D. Shrestha, Eager and lazy learning methods in the context of hydrologic forecasting, *Neural Networks, 2006. IJCNN '06. International Joint Conference on* 1 (1) (2006) 4847–4853.
9. D. Freedman, R. Pisani, R. Purves, *Statistics*, 3rd Edition, Norton, New York — NY, USA, 1998.
10. C. A. Ferrero, Documento interno – $kNN-TSP$ Resultados Experimentais, disponível em: <http://labic.icmc.usp.br/resultados/anfer/> (2009).