TRUMiner: Mineração de Regras Temporais em Bases de Séries Multivariadas e Heterogêneas*

Eliane Karasawa[1], **Elaine P. M. Sousa**[1]

[1]Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP)
São Carlos, SP – Brasil

eligniechk@gmail.com, parros@icmc.usp.br

Abstract. *This work presents TRUMiner (Temporal RUles Miner), a new algorithm to mine temporal rules from multivariate time series. Our approach deals with missing data and heterogeneous time series, including variables with different numbers of observations. Furthermore, TRUMiner returns detailed temporal rules which allow referring to the respective occurrences in the original time series. We evaluated the TRUMiner algorithm on international trade multivariate data from several sources. Early results show the applicability of our algorithm to heterogeneous time series datasets, simplifying data integration and data pre-processing.*

Resumo. *Este trabalho apresenta o algoritmo TRUMiner (Temporal RUles Miner) para mineração de regras temporais em séries multivariadas. A abordagem proposta permite o tratamento de séries temporais heterogêneas, com suporte a dados faltantes e séries com números de observações diferentes entre variáveis. Além disso, o TRUMiner retorna regras temporais detalhadas, que possibilitam referenciar as respectivas ocorrências nas séries originais. A avaliação experimental do algoritmo foi realizada em séries temporais multivariadas do comércio internacional provenientes de várias fontes. Os resultados iniciais mostram a aplicabilidade do TRUMiner a bases de séries heterogêneas, simplificando a tarefa de integração e pré-processamento dos dados.*

1. Introdução

Na área de mineração de dados, as regras de associação consistem em um padrão útil que permite correlacionar em uma relação de causalidade o elemento inicial, denominado antecedente da regra, ao elemento final, o consequente [Agrawal et al. 1993]. A mineração de regras de associação é uma tarefa de grande interesse pelo potencial de explicabilidade e possibilidade de predição [Han et al. 2011]. Numa vertente mais recente, a área trata o fator temporal, agregando às regras conhecimento relevante [Segura-Delgado et al. 2020].

A análise do fator temporal é necessária para uma melhor compreensão dos dados, indicando a ordem de acontecimento e possibilitando a avaliação do período da observação. As regras temporais permitem identificar padrões com o respectivo intervalo de acontecimento [Segura-Delgado et al. 2020]. Por exemplo, uma regra temporal pode indicar que dois anos após queda da importação em um país, seu PIB volta a crescer, auxiliando na compreensão dessas variáveis e na predição do PIB.

*Os autores agradecem à CAPES e ao CNPq pelo apoio financeiro.

Muitos dos algoritmos propostos na literatura exigem etapas específicas de pré-processamento para adequação das séries e raramente são aplicáveis a dados de múltiplas fontes [Das et al. 1998, Harms and Deogun 2004]. Por exemplo, em análises da economia mundial, os dados históricos provenientes de fontes diversas, podem compor séries multivariadas de países de interesse, resultando numa base de séries heterogêneas, com observações faltantes e variáveis com períodos de cobertura distintos.

Neste trabalho, propõe-se o algoritmo TRUMiner (Temporal RUles Miner) para aplicação em séries temporais multivariadas integradas de várias fontes. O TRUMiner lida com variáveis com períodos de cobertura heterogêneos e observações faltantes; permite discretização adaptável aos dados analisados; retorna regras temporais no formato simplificado contendo antecedente, consequente e intervalo temporal, e regras no formato extenso que informa todas as séries de ocorrência e os respectivos intervalos temporais.

2. Trabalhos Relacionados

As regras (de associação) temporais são definidas como um par (A $\Rightarrow$ C, T), sendo T a característica temporal da regra A $\Rightarrow$ C [Chen and Petrounias 2000]. Essa característica pode ser dividida em componente integral em que o tempo participa integralmente da regra, ou implícito, ordenando os dados e/ou servindo como limite temporal para determinar a relevância de um dado em relação a outro [Segura-Delgado et al. 2020]. Alguns dos trabalhos específicos de mineração de regras temporais são sumarizados a seguir.

Em [Das et al. 1998], os autores propõem a discretização das séries temporais por meio da formação de subsequências utilizando janelas deslizantes e agrupamento. O intervalo temporal da regra é obtido a partir do número de elementos existentes entre antecedente e consequente. O algoritmo MOWCATL [Harms and Deogun 2004] realiza a mineração de regras a partir de elementos de interesse pré-determinados e uma janela de tempo entre antecedente e consequente fixa ou máxima.

O Clearminer [Romani et al. 2010] realiza o processo de mineração de regras temporais de séries multivariadas, com antecedente e consequente provenientes de variáveis distintas. O algoritmo permite delimitar a janela máxima de ocorrência da regra e as regras podem ser retornadas num formato extenso, detalhando a observação inicial, intermediária e final do antecedente e do consequente, bem como seus tempos inicial e final.

O algoritmo proposto em [Zhao and Zhang 2017] minera regras temporais em séries multivariadas com normalização min-max e agrupamento para reduzir os padrões gerados. O TRiER [Amaral and Sousa 2019] extrai regras temporais de exceção em séries multivariadas visando o máximo de variáveis para cada item. Em [de Oliveira et al. 2017] o foco é a mineração de regra de associação direcionada em grafos.

A solução de mineração de regras temporais proposta neste trabalho é baseada na abordagem apresentada no Clearminer [Romani et al. 2010], com outros métodos de discretização. Assim como em [Zhao and Zhang 2017], o tempo participa integralmente da regras geradas, mas sem o uso de agrupamento. Finalmente, o TRUMiner é capaz de lidar com observações faltantes e mapeia todas as ocorrências da regra nas séries originais.

3. O algoritmo TRUMiner

O TRUMiner visa simplificar o pré-processamento necessário e facilitar a integração de fontes de dados múltiplas para as variáveis que compõem as séries. O algoritmo é ca-

paz de lidar com variáveis faltantes e com quantidades distintas de observações. O único requisito é que as séries possam ser identificadas e definidas consistentemente, independente da ordem em que foram coletados os dados. Por exemplo, usar sempre "bra" para se referir às variáveis do Brasil. A Figura 1 apresenta as etapas realizadas no TRUMiner.

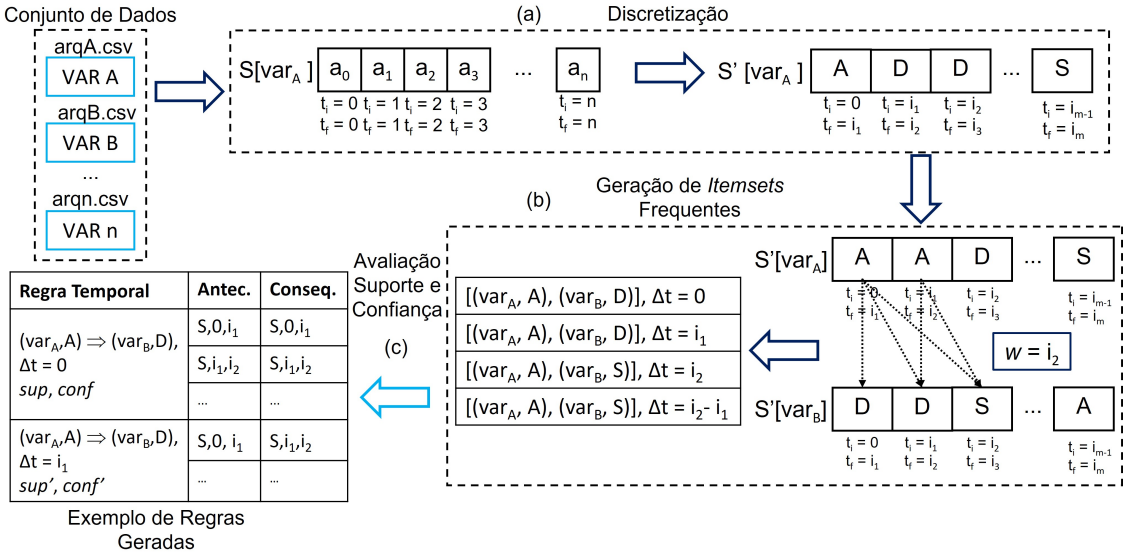

Figura 1. Representação esquemática do TRUMiner

As regras mineradas representam relacionamentos entre pares de variáveis das séries temporais multivariadas, onde antecedente e consequente são de variáveis distintas. É possível definir uma janela máxima w para gerar as regras temporais, sendo que o t_i (tempo inicial) do antecedente ocorre antes ou junto ao t_i do consequente e o início do consequente pode ser no máximo w após o início do antecedente.

3.1. Discretização

A Figura 1 (a) ilustra o processo de discretização que transforma as observações da variável A da série S (denotado por S[var_A]) em padrões discretizados (S'[var_A]). O tipo e o intervalo temporal dos padrões resultantes dependem da discretização escolhida. Os problemas de observações faltantes e de tamanhos heterogêneos entre as variáveis são tratados ao armazenar o t_i e o t_f (respectivamente os tempos inicial e final do padrão), permitindo duração distinta em cada padrão discretizado. Assim, ao gerar os *itemsets*, avalia-se o t_i e o t_f dos padrões envolvidos para impedir inconsistências.

As discretizações implementadas no algoritmo são: decis, quartis, SAX e comportamental. As discretizações decis e quartis permitem um maior detalhamento do comportamento do padrão, informando o percentual referente ao aumento ou queda. A discretização SAX [Lin et al. 2003] é uma abordagem clássica aplicada em vários domínios, comumente utilizada na descoberta de *motifs*. A discretização comportamental baseada no Clearminer [Romani et al. 2010] gera padrões de aumento (A), queda (D) e estabilidade (S) a partir de parâmetros (de usuário) que indicam: o máximo de observações a serem agrupadas, o número de observações mínimas para serem discretizadas em estabilidade, a diferença máxima entre duas observações para ser um elemento de estabilidade, e a mínima variação entre observações para ser discretizada em um padrão qualquer.

3.2. Geração de *Itemsets* Frequentes

A Figura 1 (b) ilustra o processo de geração de *itemsets* realizado pelo TRUMiner entre as variáveis var_A e var_B da série discretizada S'. O valor $w = i_2$, limita o t_i máximo do padrão final a i_2 após o início do padrão inicial. Alguns *itemsets* gerados estão apresentados em uma lista. A contabilização da frequência de um item considera o intervalo temporal em que o item está relacionado. Por exemplo, em um *itemset* do tipo $[(var_A, A), (var_B, D)]$ com Δt = i_1, a frequência de (var_A, A) somente será contabilizada a cada item A nas séries da var_A que componha um *itemset* do tipo $[(var_A, A), ()]$ com Δt = i_1.

3.3. Avaliação de Suporte e Confiança

As regras temporais são avaliadas através do suporte e da confiança, conforme indicado na Figura 1 (c). O suporte é dado pela frequência dos itens $f(A \cap C)$ sobre o número de *itemsets* e a confiança é $f(A \cap C)$ sobre a frequência do antecedente $f(A)$. O par de variáveis avaliado pode gerar dois tipos de *itemsets*, representados genericamente por $[var_1, var_2]$ e $[var_2, var_1]$. Assim, o suporte de uma regra temporal é limitado a 50% dado que cada par de padrão discretizado gera um *itemset* $[var_1, var_2]$ e outro $[var_2, var_1]$.

O algoritmo retorna as regras temporais acima do suporte mínimo (sup_{min}) e da confiança mínima ($conf_{min}$) informando a variável e o padrão antecedentes, a variável e o padrão consequentes e o intervalo de tempo entre antecedente e consequente (e.g. [(var_A, A) $\Rightarrow$ (var_B, D), Δt = 0] na tabela da Figura 1). Suporte e confiança são calculados e fornecidos conforme frequência descrita previamente. No formato extenso são retornados os identificadores das séries temporais em que a regra é verificada e o tempo inicial e final de cada antecendente e consequente (e.g. [(var_A, A) $\Rightarrow$ (var_B, D), Δt = 0], série S, antecedente e consequente t_i = 0 e t_f = i_1 na tabela da Figura 1).

A complexidade do TRUMiner é dada por $N^2.L.w$ onde N é o número de séries que constituem o conjunto e L é o maior número de observações, sendo a geração de *itemsets* o processo mais custoso do algoritmo. O TRUMiner foi escrito em C++ utilizando o conceito de classes. O uso de uma linguagem de baixo nível permite melhor utilização da memória, evitando o problema de falta de memória principal, fundamental para mineração de regras. O uso de classes facilita a compreensão e o reuso do código.

4. Resultados e Discussão

A avaliação experimental realizada neste trabalho explora dados do comércio internacional, com os índices: valores de importação e exportação dos países[1], Produto Interno Bruto (PIB)[2] e Índice de Complexidade Econômica (ECI)[3] que caracteriza a capacidade tecnológica de produção do país. A mineração de regras temporais permite compreender melhor o relacionamento entre variáveis econômicas e possibilita a predição de comportamentos econômicos. A base de dados constitui séries anuais com 4 variáveis: importação (IMP) e exportação (EXP), ambas contendo observações de 228 países de 1996 a 2020, PIB com o mesmo intervalo de cobertura e observações de 196 países, e ECI com observações de 133 países de 1996 a 2019.

[1]BACII http://www.cepii.fr/CEPII/en/bdd_modele/bdd_modele_item.asp?id=37
[2]GDP https://www.imf.org/en/Publications/WEO/weo-database/2022/April
[3]ECI https://atlas.cid.harvard.edu/rankings

A principal regra de cada discretização sobre o par de variáveis com maior confiança são apresentadas na Tabela 1. Todas as regras apresentam $\Delta t = 0$, já que este constitui-se o intervalo temporal com maior número de *itemsets* gerados. Além disso, a regra de quartis pode implicar que o aumento verificado na discretização comportamental é até 25% maior em relação à observação anterior, no antecedente e no consequente.

Tabela 1. Principal regra das variáveis (IMP, PIB) para cada discretização

Discretização	Regra Temporal
Comportamental	(IMP, A) ⇒ (PIB, A), Δt = 0 $sup = 4.58, conf = 83.21$
Decis	(IMP, A1) ⇒ (PIB, A1), Δt = 0 $sup = 1.25, conf = 59.97$
Quartis	(IMP, A1) ⇒ (PIB, A1), Δt = 0 $sup = 3.83, conf = 81.86$
SAX	(IMP, a) ⇒ (PIB, a), Δt = 0 $sup = 3.68, conf = 96.56$

As regras temporais de séries multivariadas com o fator temporal participando integralmente da regra apresentam baixo suporte devido a sua natureza combinatória. No conjunto avaliado, com $w = 5$, a média de regras geradas chega a 30 mil para cada par de variáveis. A regra mais frequente e confiante no par de variáveis (PIB, EXP) é (EXP, A) ⇒ (PIB, A), Δt = 0, sup = 4.45 e conf = 82.34. Essa regra indica que, a nível mundial, 82% das vezes que a exportação de um país sobe, no mesmo período o seu PIB também sobe. Nota-se que apesar do baixo suporte, as regras apresentam alta confiabilidade.

As regras extensas retornadas pelo TRUMiner permitem a fácil localização nas séries temporais em que aconteceram e o relacionamento da(s) observação(ões) ao padrão discretizado. Por exemplo, (EXP, D) ⇒ (PIB, A), Δt = 2, indica que 2 anos após a queda da exportação, o PIB aumenta com sup = 1.45 e conf = 72.72. Essa regra ocorreu no Brasil em (2009, 2011), (2015, 2017), na China (2009, 2011), (2016, 2018) e nos Estados Unidos (2001, 2003), (2002, 2004), (2009, 2011), (2015, 2017), (2016, 2018).

Apesar do período de cobertura menor para o ECI, observa-se a regra (ECI, D) ⇒ (PIB, D), Δt = 1 encontrada pelo TRUMiner, indicando que uma queda no ECI é seguida de uma queda no PIB após um ano. O formato extenso verifica a ocorrência da regra no Brasil entre 2019 (ECI) e 2020 (PIB), que pode ser contabilizado somente por algoritmos capazes de lidar com séries com variáveis de tamanhos heterogêneos, como o TRUMiner.

Numa análise sobre a série do Brasil, utilizou-se as variáveis PIB e importação e removeu-se de 0 a 4 observações (16%) aleatoriamente da importação. As 10 regras mais frequentes encontradas no caso sem observações faltantes compõem 60% ou mais das 10 regras mais frequentes em todos os casos com observações faltantes. Esse resultado indica robustez do algoritmo frente a observações faltantes e possibilidade de mineração sem o pré-processamento para tratamento desse problema.

5. Conclusão

A mineração de séries temporais multivariadas é uma área promissora pela capacidade de obter informações novas e relevantes indexadas ao seu intervalo temporal. As regras

temporais permitem maior explicabilidade dos padrões encontrados e a possibilidade de realizar previsões sobre os dados. Os métodos existentes frequentemente limitam-se à análise univariada ou necessitam de grande pré-processamento das séries. Aliado a isso, está a baixa aplicabilidade a conjuntos diversos, limitando eficiência ao conjunto avaliado.

O algoritmo TRUMiner visa a simplificação do processo de mineração, apresentando capacidade de lidar com dados de múltiplas fontes contendo séries distintas entre si, de duração temporal heterogênea e com observações faltantes. As regras temporais podem ser analisadas no formato extenso, verificando as séries temporais de ocorrência e seu intervalo temporal correspondente. Embora a natureza do problema implique em baixos valores do suporte, os resultados são promissores para a análise de bases de séries temporais multivariadas heterogêneas. Futuramente serão estudadas medidas alternativas ao baixo suporte. Visa-se também a geração eficiente de regras com 3 ou mais variáveis.

Referências

Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216.

Amaral, T. and Sousa, E. (2019). Trier: A fast and scalable method for mining temporal exception rules. In *Anais do XXXIV SBBD*, pages 1–12. SBC.

Chen, X. and Petrounias, I. (2000). Discovering temporal association rules: Algorithms, language and system. In *16th ICDE*, pages 306–306. IEEE.

Das, G., Lin, K.-I., Mannila, H., Renganathan, G., and Smyth, P. (1998). Rule discovery from time series. In *4th ACM KDD*, volume 98, pages 16–22.

de Oliveira, F. A., Costa, R. L., Goldschmidt, R. R., and Cavalcanti, M. C. (2017). Mineração de regras de associação multirrelação em grafos: Direcionando o processo de busca. In *SBBD (Short Papers)*, pages 270–275.

Han, J., Kamber, M., and Pei, J. (2011). Data mining: Concepts and techniques. *(3rd ed), Morgan Kauffman*.

Harms, S. K. and Deogun, J. S. (2004). Sequential association rule mining with time lags. *Journal of Intelligent Information Systems*, 22(1):7–22.

Lin, J., Keogh, E., Lonardi, S., and Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD*, DMKD '03, page 2–11, New York, NY, USA.

Romani, L. A. S., de Avila, A. M. H., Zullo, J., Chbeir, R., Traina, C., and Traina, A. J. M. (2010). Clearminer: a new algorithm for mining association patterns on heterogeneous time series from climate data. In *ACM*, SAC '10, page 900–905, New York, NY, USA.

Segura-Delgado, A., Gacto, M. J., Alcalá, R., and Alcalá-Fdez, J. (2020). Temporal association rule mining: An overview considering the time variable as an integral or implied component. *WIREs Data Mining and Knowledge Discovery*, 10(4):e1367.

Zhao, Y. and Zhang, T. (2017). Discovery of temporal association rules in multivariate time series. In *International Conference on Mathematics, Modelling and Simulation Technologies and Applications, 2017, Xiamen*, pages 294–300.