Redução no espaço de busca para mineração de itemset-like patterns

Samuel Lipovetsky
Universidade Federal de Minas Gerais
Belo Horizonte, Minas Gerais, Brasil
samuellpvt@gmail.com

Loic cerf
Universidade Federal de Minas Gerais
Belo Horizonte, Minas Gerais, Brasil
lcerf@dcc.ufmg.br

***Abstract*—In tensor data mining, discovering itemset-like patterns can be highly computationally intensive. This paper introduces a method for reducing the search space by pre-processing tensors to eliminate cells that cannot contribute to valid patterns. The proposed approach involves analyzing intersections between different tensor slices and applying noise limits to refine the search process. By removing subspaces that do not meet the specified criteria, we optimize the efficiency of the mining process without losing information about potential patterns. This technique enhances the computational feasibility of pattern discovery in large-scale tensor data and provides a practical solution for the problem.**

I. Introdução

Suponha que uma empresa está realizando uma pesquisa sobre aspectos de qualidade de diversos produtos com seus clientes. Isso gera um tensor tridimensional em que cada célula (u, p, a) representa uma nota X de 0 a 1. Nesta notação, u representa um usuário, p representa um produto e a representa um aspecto do produto avaliado. Assim, um analista desta empresa deseja obter um subconjunto desse tensor que seja próximo de ser composto somente de 1s, com um limite de ruído determinado pelo analista e que então represente um subconjunto de notas de usuários por aspecto de um produto em que a satisfação foi elevada.

	aspecto 1	aspecto 2	aspecto 3	aspecto 4
Alice	0.8	0.1	1	0.3
Bob	0.7	0.3	0.8	0.2
Carol	0.8	0.2	0.9	0.1

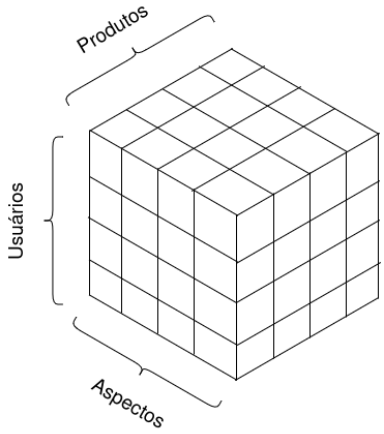
TABLE I: Fatia de Notas de usuários para aspectos de um produto

A maneira como o analista define o quão próximo de 1 o subconjunto minerado deve ser é a partir de um limite de ruido por dimensão do conjunto de dados. Por exemplo na tabela 1, temos uma fatia dos dados que fixa um produto e contem todas as notas dos aspectos por usuários daquele produto. Igualmente, existem dois outros tipo de fatias, uma que fixa um usuário e outra que fixa um aspecto. Assim, o ruido de uma fatia é soma das diferenças entre 1 e o valor de cada célula daquela fatia. Dessa forma, o analista deve definir os limites de ruido por fatia e o algoritmo multidupehack retorna subconjuntos do tensor em que o ruido das fatias desses subconjuntos sempre são menores que o limite definido. Ainda é possível definir outros limites para os padrões minerados que são descritos em [2].

Como minerar esses tipos de padrões é um processo extremamente caro computacionalmente, uma das maneiras de otimizar a velocidade de execução do algoritmo é diminuir o espaço de busca removendo células que com certeza não vão pertencer a nenhum padrão válido especificado pelos limites de ruídos. Assim, o foco deste trabalho é pré-processar os dados observando as interseções entre os diferentes tipos de fatias e como o limite de ruido definido para um tipo de fatia pode ser usado em outros tipos de fatias para reduzir o espaço de busca.

II. Visualizando o problema geometricamente

Ainda Utilizando o exemplo da pesquisa com usuários sobre aspectos de vários produtos, que pode ser descrita como um tensor tridimensional que atribui um valor entre 0 e 1 para cada célula (u, p, a) que pertencem respectivamente aos conjuntos $Usuarios$,$Produtos$ e $Aspectos$

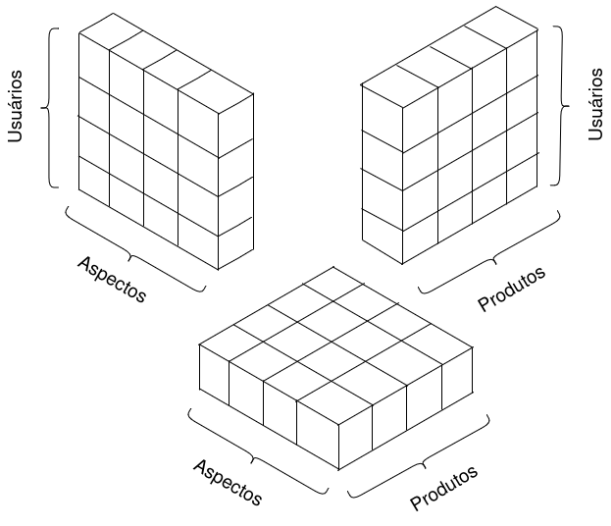


via

Fig. 1: Visualização geométrica do tensor.

Então são definidos valores limites para os ruídos e para os tamanhos mínimos de cada tipo de fatia dos subtensores que serão minerados.

Assim, para cada fatia do subtensor minerado, a soma dos ruídos de cada célula da fatia deve ser menor que o limite



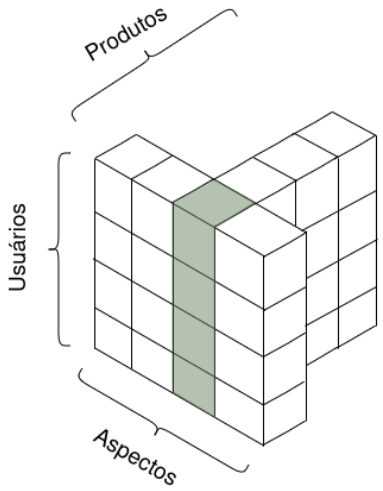
via

Fig. 2: Tipos de fatias do subtensor.

especificado. Sobre o número de fatias de cada tipo, para fatias do tipo $Usuario$ que representa as notas de um usuário sobre todos os aspectos de todos os produtos, como demostrado na tabela 1, existem $|Usuario|$'s desse tipo de fatia no tensor original , sendo que essa lógica se expande para todos os outros tipos de fatia.

A. Limites de ruídos ortogonais

Como existem conjuntos de elementos que são comuns entre fatias ortogonais, ou seja elementos comuns entre dois tipos de fatias diferentes, esses elementos que são interseções de fatias devem obedecer os limites de ruídos de ambas as fatias que os contem.



via

Fig. 3: Interseção de fatias

Dessa forma, é possível diminuir ainda mais o espaço de busca a partir da remoção de conjuntos de elementos que nunca vão pertence a um padrão válido sem perda de informação para os padrões minerados apos o pré-processamento. No exemplo da Figura 3, a coluna destacada deve obedecer aos limites de ruído definidos tanto para as fatias de $Usuarios \times Aspectos$ quanto para $Usuarios \times produtos$, uma vez que o subespaço destacado pertence aos dois tipos de fatia ao mesmo tempo. Assim, este trabalho busca provar que é possível remover subespaços do tensor observando esse tipo de interseção sem perda de informação nos padrões minerados.

III. Definição de padrões minerados

Sejam $D_1, .., D_n$ com $n \in \mathbb{N}$ conjuntos finitos assumidos disjuntos sem perda de generalidade. Existe um tensor T que mapeia cada N-tupla $\prod_{k=1}^{n} D_k$, em que $\prod$ representa o produto cartesiano, para um valor $T_t \in [0, 1]$ que é chamado de grau de pertencimento. Assim , podemos definir π que recebe um conjunto de E que é uma união de subconjuntos unitários de $D_1, ..., D_n$.

$$\pi : \begin{array}{c} 2^{\cup_{i=1}^{n} D_i} \to 2^{\Pi_{i=1}^{n} D_i} \\ (E, (D_1, .., D_n)) \mapsto \prod_{i=1}^{n} F_i \begin{cases} \text{se } E \cap D_i = \varnothing, F_i = D_i \\ \text{senão }, F_i = E \cap D_i \end{cases} \end{array} \tag{1}$$

Assim usando o exemplo da figura 2, com $E = \{Alice, produto1\}$ e $D = Usuarios, Produtos, Aspectos$.

$$\pi(\{Alice, produto_1\}, D) = \{Alice\} \times \{produto_1\} \times \{Aspectos\} \tag{2}$$

IV. Usando a função π para definir padrões válidos

O multidupehack recebe como entrada os conjuntos de dimensões com os graus de pertencimento de cada n-tupla, além de receber os limites de ruido $\{\epsilon_1, .., \epsilon_n\} \in \mathbb{R}_+$ e os tamanhos mínimos de cada dimensão $\{\gamma_1, .., \gamma_n\} \in \mathbb{N}_+$ e retorna os subconjuntos $X_1, .., X_n$ tal que $X_1 \subseteq D_1, .., X_n \subseteq D_n$ somente se:

$$\forall i \in \{1, ..., n\} \begin{cases} \forall e \in X_i, \sum_{t \in \pi(\{e\}, (X_1, .., X_n))} (1 - T_t) \leq \epsilon_i \\ |X_i| \geq \gamma_i \end{cases} \tag{3}$$

V. Zerando sub-espaços $\pi(E, (D_1, .., D_n))$ de T_t

Para remover qualquer n-tupla de T_t sem perda de informação nos padrões minerados pelo multidupehack, é necessário provar que as n-tuplas removidas não pertencem a nenhum padrão válido. Assim, dado um tensor R_t, que será chamado do tensor reduzido, que é obtido a partir de :

$$\begin{array}{c} (T, \pi(E, \{D_1, .., D_n\})) \mapsto R_t \\ R_t \mid \forall t \in R_t \begin{cases} \text{se } t \in \pi(E, \{D_1, ..., D_\}), t = 0 \\ \text{senão } t = Tt \end{cases} \end{array} \tag{4}$$

É preciso provar que os padrões válidos do tensor original T_t são os mesmos do que os do tensor reduzido R_t dado um subespaço $\pi(E,(D_1,..,D_n))$
Para auxiliar na prova será definida a função $index$ que retorna os índices das dimensões que não possuem interseção com E:

$$index: \begin{matrix} 2^{\cup_{i=1}^{n} D_i} \to I \subseteq \{1,..,n\} \\ E \mapsto \{i \mid D_i \cap E = \varnothing\} \end{matrix} \quad (5)$$

Assim, seja que $B = max(m, \pi(E,(D_1,...,D_n)))$ representa as m n-tuplas com maiores valores dentro de $\pi(E,(D_1,...,D_n)$, podemos remover esse subespaço sem perda de padrões válidos, somente se:

$$\nexists m \in \mathbb{N} \begin{cases} \sum\limits_{t \epsilon B} (1-T_t) \geq min(\{\epsilon_k | k \notin \text{index}(E)\}) \\ m \geq \prod\limits_{i \in \text{ index}(E)} \gamma_i \end{cases} \quad (6)$$

Ou seja, é possível remover um subespaço $\pi(E,(D_1,...D_n))$ somente se não existem pelo menos m células dentro desse subespaço tal que que as soma dos ruídos dessas m células é menor que o menor limite de ruido das dimensões com elementos em E , ou se m existir mas for menor que o produto dos limites de tamanhos das dimensões sem elementos em E.

Portanto, basta provar que para um dado subespaço $\pi(E,(D_1,...,D_n))$ se existe m que satisfaz (6) então $\pi(E,(D_1,...,D_n))$ não pertence a um padrão válido e pode ser removido.
Seja $(E_1,..,E_n)$ as dimensões de $\pi(E,(D_1,..,D_n)$ com os limites de ruido $(\epsilon_1,..,\epsilon_n) \in \mathbb{R}_+$ e os tamanhos mínimos de cada dimensão $(\gamma_1,..,\gamma_n) \in \mathbb{N}_+$, se existe m que satisfaz (6) então usando a propriedade que:

$$\sum_{t \in \max(m,\pi(E))} (1-T_t) \leq \sum_{t \epsilon \pi(E)} (1-T_t) \quad (7)$$

e que

$$\sum_{t \in \max(m,\pi(E))} (1-T_t) \geq min(\{\epsilon_k | k \notin \text{index}(E)\}) \quad (8)$$

é possível concluir que:

$$\exists e_i \in E \mid \sum_{t \epsilon (\pi(e,(E_1,..,E_n))} (1-T_t) \geq \epsilon_i \quad (9)$$

e assim $\pi(E,(D_1,...,D_n))$ não pode estar em um padrão válido, uma vez que a desigualdade (9) contradiz a definação do padrão descrito em (3).

VI. Conclusão

Portanto , utilizando o fato de que remover subespaços que não pertence a nenhum padrão válido não causa perda de informação sobre os padrões e que existem maneiras de checar se um subespaço pertence ou não à um padrão válido, é possível diminuir o espaço de busca sem perda de informação.

References

[1] Loic Cerf and Meira Jr., Wagner,Complete Discovery of High-Quality Patterns in Large Numerical Tensors, 2014.

[2] Loic Cerf , Enforcement of Minimal Size and Area Constraints before and while Mining Patterns in Fuzzy Tensor, 2023.