Universidade de São Paulo
Instituto de Ciências e Matemáticas e Computação - ICMC

Relatório final - Iniciação Científica

Leonardo Vinícius de Oliveira Toledo

Orientadora: Profa. Dra. Elaine Parros Machado de Sousa

Fevereiro de 2019
São Carlos - SP

SUMÁRIO

1 - Introdução

Em uma época na qual o poder de geração e armazenamento de dados cresce exponencialmente, métodos computacionais para extração de padrões vem tornando-se necessários, visto a inviabilidade de extração manual. Diante disso, busca-se tornar as bases de dados o mais inteligentemente organizadas possível, de modo a extrair padrões interessantes para o propósito e evitar informações irrelevantes. Nesse contexto, o trabalho em questão visa a implementação de métodos para tratamento desses dados.

Especificamente, são aplicados três métodos: Piecewise Aggregate Approximation, Symbolic Aggregate Approximation e ClipsMiner.

O foco do trabalho são dados de NDVI (Normalized Difference Vegetation Index) e clima localizados no estado de São Paulo. Entretanto, os métodos são aplicáveis a qualquer base de dados.

2 - Resultados e Discussões

2.1 - Piecewise Aggregate Approximation

O método Piecewise Aggregate Approximation tem por objetivo transformar uma série $X = \{x_1, x_2, ..., x_n\}$ em uma série $X' = \{x'_1, x'_2, ..., x'_m\}$ tal que $m < n$ [1].
O procedimento utilizado para tal processo consiste na divisão da série *X* em *k* segmentos de tamanho igual, sendo feita, posteriormente, uma média dos valores de cada segmento. Cada média (m_i) de um segmento $\{x_a, ..., x_b\}$ será um novo elemento da série transformada *X'*. A equação 1.0 para esse cálculo é dada a seguir:

$$x'_i = \frac{m}{n} \sum_{j=\frac{n}{m(i-1)+1}}^{(\frac{n}{m})i} \qquad (1.0)$$

Para implementar esse algoritmo foi utilizada linguagem C++. No processo, foi criada uma classe "Elemento", que possui em si uma variável char chamada "tipo", que indica se o valor é uma latitude, longitude ou Ndvi (Normalized Difference Vegetation Index) e uma variável "value", que armazena o valor do elemento da série temporal em questão. Desse modo, cada valor x_i de uma série é um Elemento.

A função que realiza o cálculo do PAA consiste, basicamente, em um loop externo que percorre uma série por vez, seguido de um loop interno, que percorre cada elemento de cada série. Ao longo desse loop interno, ao passo que é percorrida a série, é feita a média de cada segmento de tamanho pré-definido pelo usuário, que formará, portanto, parte da série transformada. Ao fim desse processo, é gerado um arquivo de dados com os dados tratados.

Abaixo, temos um exemplo de uma série temporal tratada neste processo:

Série de Entrada :

2	6	1	5	7	2	1	8	6	6	2	1

Quantidade de Segmentos Escolhida: 4

Série Transformada:

3	4.66667	5	3

Graficamente, temos o arranjo abaixo. A cada 3 elementos da série original, forma-se um novo elemento da série transformada(parte em vermelho).

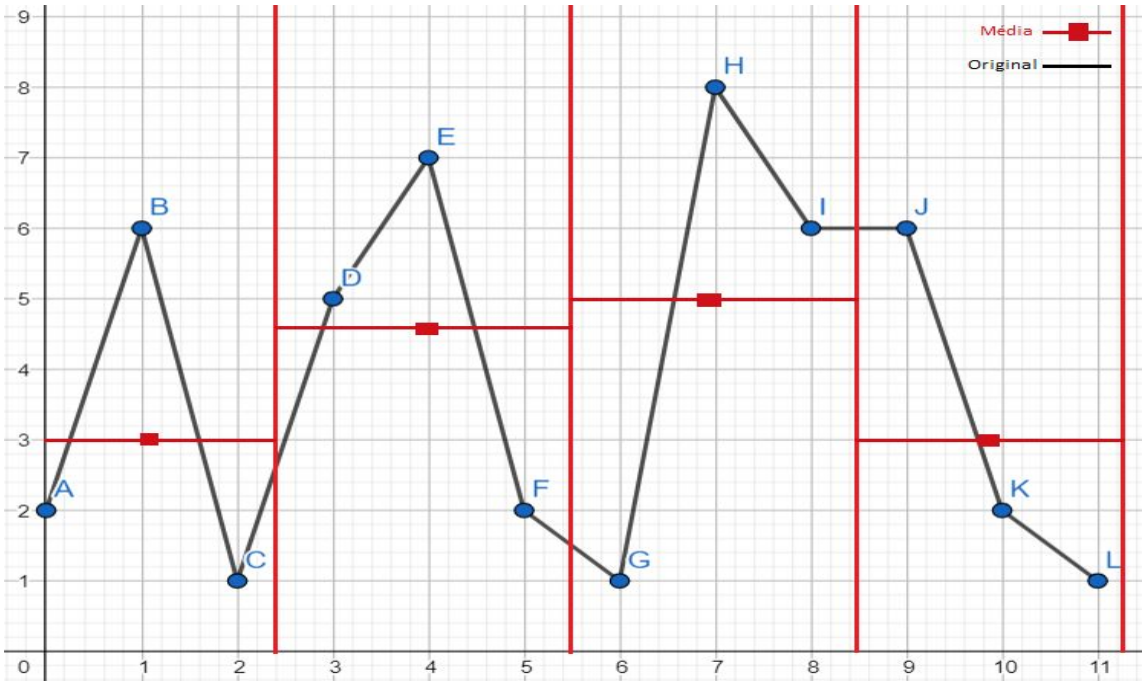


Figura 1.0: Exemplo de representação PAA. Adaptada de AMARAL, B. F.. Classificação semissupervisionada de séries temporais extraídas de imagens de satélite. 2016. 95 f.

2.2 - Symbolic Aggregate Approximation

Proposto por Eamonn Keogh and Jessica Lin em 2002, esse algoritmo tem como base o Piecewise Aggregate Approximation e utiliza-se de uma representação simbólica para os dados [2].

O funcionamento se dá da seguinte forma: Primeiramente, a série temporal passa pelo PAA conforme explicado anteriormente. Posteriormente, assumindo uma distribuição normal N(0,1), a série é discretizada em X partes, tendo cada parte $\frac{1}{X}$ de tamanho. A cada parte é associado um símbolo e a linha divisória de duas partes é chamada de Breakpoint. Neste caso, foram usadas letras do alfabeto como símbolos para representação [3].

Para implementar esse algoritmo foi usada linguagem C++. No processo, foi primeiro utilizado o já implementado PAA, e, posteriormente, é usada uma função que aplica o método SAX.

A função SAX pede como parâmetro a quantidade de símbolos desejada (Q). A quantidade de Breakpoints a ser feita será dada pela quantidade de símbolos mais um. Posteriormente, é calculado o maior (V_{max}) e menor(V_{min}) valor da série temporal, sendo, então, feito o seguinte cálculo:

$$Faixa\ de\ valor\ entre\ dois\ breakpoints = \frac{V_{max} - V_{min}}{Q}$$

O primeiro Breakpoint será instalado no menor valor da série temporal gerada pelo PAA e o elemento em cima dele terá o símbolo do intervalo logo acima. Já o último Breakpoint será implantado no maior valor da série, e o valor em cima dele terá o símbolo do intervalo logo abaixo. Os Breakpoints intermediários serão classificados de acordo com o símbolo da faixa na qual eles se encontrarem.

Veja como uma série temporal se comporta passando por esse processo:

Série de Entrada :

2	6	1	5	7	2	1	8	6	6	2	1

Quantidade de Segmentos Escolhida: 4 ***(Usada no cálculo do PAA)***

Série Transformada pelo PAA:

3	4.66667	5	3

Quantidade de Símbolos Escolhida: 2 ***(Os símbolos serão A e B)***

Como temos 2 símbolos, A e B, teremos 3 Breakpoints, com o primeiro começando no menor valor (3) e o último terminando no maior valor (5). Abaixo, temos uma representação gráfica para maior entendimento. As médias são os valores calculados pelo PAA.

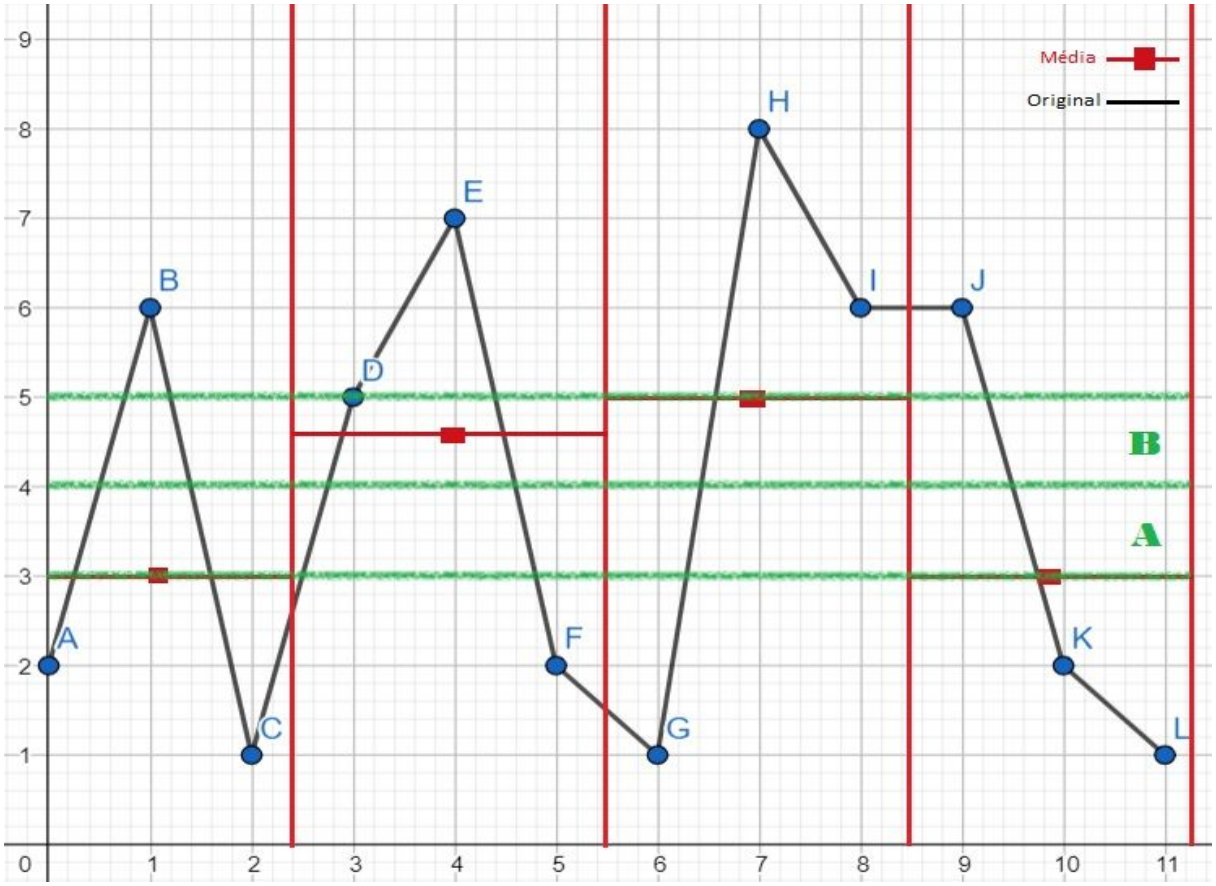


Figura 2.0 - Exemplo gráfico de funcionamento do SAX. As faixas em verde representam BreakPoints.

Com isso, ficaremos com a série {3, 4.66667, 5, 3} alterada para símbolos A e B do seguinte modo:

PAA	**3**	**4.66667**	**5**	**3**
SAX	**A**	**B**	**B**	**A**

2.3 - ClipsMiner

Criado por Luciana Alvim Santos Romani, é um algoritmo que consiste na extração de padrões de pico, vale e platô em uma série temporal e tem destaque em uso para dados climáticos [4].

Nesse algoritmo, temos três parâmetros dados pelo usuário:

- δ (*delta*) : valor, em módulo, para classificar o crescimento ou decrescimento mínimo de uma sequência para ela ser ou não enquadrada como um evento estável. Se um elemento estiver na faixa de valor de delta, ele é um evento estável.

- λ (*comprimento de platô*) : Quantidade mínima de elementos em um intervalo estável para que ele seja considerado um platô.

- ρ (*fator de relevância*) : É uma porcentagem da amplitude total da série temporal, usada para classificar um evento como crescente ou decrescente.

Seu funcionamento se dá, inicialmente, pela criação de um vetor de diferenças $d = \{d_1, d_2, ..., d_n\}$, no qual cada elemento é dado pela subtração $x_{i+1} - x_i$, sendo $x = \{x_1, x_2, ..., x_n\}$ uma série temporal. Posteriormente, são classificados 3 tipos de sequência: S_{es}, S_{ea}, S_{ed} .

S_{es} é um evento estável, que ocorre quando $|d_i| < \delta$. S_{ea} é um evento ascendente, que ocorre quando $d_i > \delta$. S_{ed} é um evento decrescente, que ocorre quando $d_i < -\delta$.

Uma vez classificadas as sequências, é feita então a concatenação delas. A partir daí, temos 3 tipos de concatenação que geram padrões:

-Montanha: é gerada pela concatenação de uma sequência de crescimento com uma sequência de decrescimento, ou seja, $S_{ea} \cup S_{ed}$.

-Vale: é gerado pela concatenação de uma sequência de decrescimento com uma sequência de crescimento, ou seja, $S_{ed} \cup S_{ea}$.

-Platô: é gerado pela concatenação de sequências estáveis, desde que o número dessas sequências sejam maior ou igual a λ , ou seja, $S_{es1} \cup S_{es2} \cup ... \cup S_{esn} / n \geq \lambda$.

A implementação desse método foi feita em C++ e a seguir temos um exemplo de como uma série temporal genérica se comporta.

Obs: Para os cálculos a seguir, foi usado $\delta = 0.1,\ \lambda = 2\ e\ \rho = 40\%$

Série de Entrada :

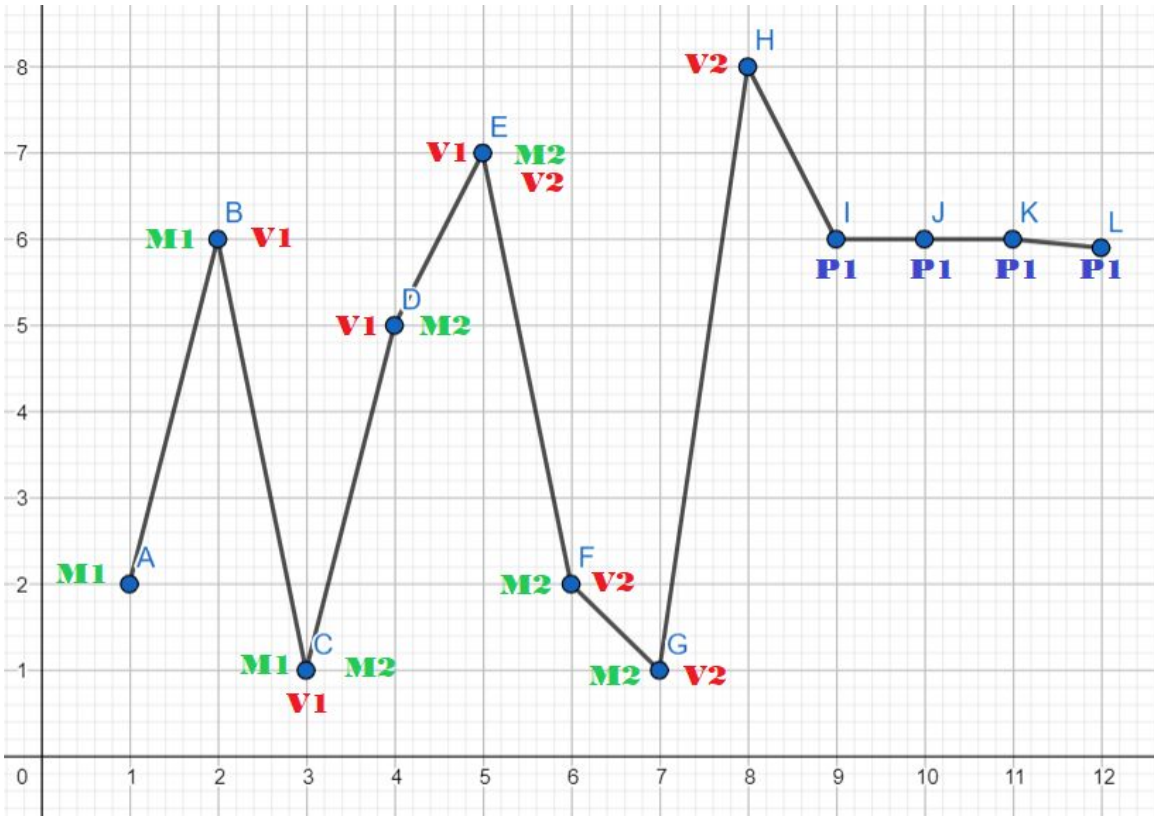
2	**6**	**1**	**5**	**7**	**2**	**1**	**8**	**6**	**6**	**6**	**5.9**

Na saída do programa, a representação de um padrão é feita pelos seus elementos e pela posição inicial do padrão: (Elemento, Posição inicial), (Elemento, Posição inicial).

Série de Saída:

Montanha	(2, 0)	(6, 0)	(1, 0)		
Vale	(6, 1)	(1, 1)	(5, 1)	(7, 1)	
Montanha	(1, 2)	(5, 2)	(7, 2)	(2, 2)	(1, 2)
Vale	(7, 4)	(2, 4)	(1, 4)	(8, 4)	
Platô	(6, 8)	(6, 8)	(6, 8)	(5.9, 8)	

Graficamente, temos o seguinte arranjo:



Conclusões:

Nas bases de dados utilizadas para teste, ficou evidente a complexidade de lidar com grandes volumes de dados. Nesse contexto, ficou evidente a necessidade de simplificar ao máximo tais dados, para melhor aproveitamento. Portanto, a aplicação dos métodos apresentados é imprescindível para melhor extração de padrões, pois, além de reduzir a dimensionalidade das séries temporais, como no PAA e no SAX, também simplificam os dados, seja por utilização de símbolos(SAX) ou representação por padrões(ClipsMiner).

Referências:
[1] - https://jmotif.github.io/sax-vsm_site/morea/algorithm/PAA.html
[2]- http://www.cs.ucr.edu/~eamonn/SAX.htm
[3]- AMARAL, B. F.. Classificação semissupervisionada de séries temporais extraídas de imagens de satélite. 2016. 95 f.
[4]- ROMANI, A New Algorithm for Mining Association Patterns on Heterogeneous Time Series from Climate Data, 2010