Projeto 3 - Dynamic Time Warping (DTW)

Victor Augusto Alves Catanante - 10839918

I. Implementação

A implementação do projeto foi feita com a linguagem de programação C++. Um script *bash* para realizar dez testes para cada abordagem também foi construído. As informações sobre a forma de execução dos programas estão descritas no arquivo README.txt.

A classe DynamicTimeWarping implementa todas as operações. Foram criados tipos de dados com as estruturas *pair* e *vector* da linguagem C++ em prol de armazenar as séries temporais lidas dos arquivos no método *loadTimeSeries*. O valor da distância entre cada elemento das séries temporais foi calculado por meio da distância euclidiana, conforme mostra a equação 1:

$$d(x, y) = (x - y)^2 \tag{1}$$

Um método baseado em comparações dos valores tomados dois a dois foi criado para encontrar o valor mínimo entre as três possibilidades de escolha da abordagem de programação dinâmica; devido ao fato de que a linguagem C++ não oferece uma função nativa que calcula o mínimo entre diversos números, optou-se por implementar a comparação ao invés de usar a função $min(x, y)$ de forma aninhada.

A classificação e a medida de acurácia foram feitas com o método *classify*. Este percorre o conjunto de treinamento *S* e realiza o cálculo da distância DTW para cada elemento do conjunto teste *Q*; concomitantemente, procura pelo menor valor de distância em relação a tal elemento teste. Ao final, classifica cada série de *Q* com a classe de *S*, relacionada ao menor valor de DTW. O valor da acurácia é definido pela quantidade de classificações corretas em relação ao conjunto *Q*.

As complexidades de tempo e espaço do algoritmo DTW (implementado como método *DTWDistance*) são $O(NM)$, sendo N e M os comprimentos das séries temporais. A leitura do arquivo possui complexidade linear de acordo com o tamanho de cada série, e o algoritmo do 1-vizinho mais próximo possui complexidades de tempo e espaço iguais ao DTW, visto que também percorre todos os elementos de ambas as séries.

II. Resultados

Os dados sobre o ambiente de teste, acurácia incial e tempo médio de execução estão nas tabelas I e II.

A implementação inicial, a qual executa a classificação das séries temporais unidimensionais fornecidas por [1], necessitou de 20 segundos em média para calcular a matriz de distâncias e realizar a classificação. Conforme descrito em [3], a distância euclidiana pode ser calculada por meio da equação 2:

$$d(x, y) = |x - y| \tag{2}$$

Esta alteração foi relevante e resultou em uma redução do tempo de execução para 18 segundos em média. A acurácia do método foi 83.8542%, ou seja, aproximadamente 84%.

Tabela I
Ambiente de Teste

OS	Debian 9.4
Compilador	gcc 6.3.0 20170516
CPU	Intel Core i7
Memória	8GB

Tabela II
Resultados Iniciais

Acurácia	Acertos	Total	Tempo (s)
83.8542	805	960	18.0381

III. Extensões e seus Resultados

Foram implementadas as duas extensões propostas para o projeto. A primeira extensão é a banda de Sakoe-Chiba para o aumento da acurácia e diminuição do tempo de execução. A adaptação foi feita de acordo com a proposição em [3]: adiciona-se uma restrição ao segundo laço de repetição no cálculo da matriz DTW. Multiplica-se o tamanho da banda pelo maior valor entre as dimensões das séries temporais, e utiliza-se a diferença absoluta entre as mesmas como comparação. O maior valor entre estes é o parâmetro que irá ponderar os limites inferiores e superiores para a banda. O trecho de código da figura 1 indica a alteração.

A utilização das bandas propiciou resultados melhores em relação à acurácia e ao tempo de execução, sem alterações na complexidade. Conforme a tabela III, a banda de 20% trouxe a melhor acurácia em um tempo menor que o de outros tamanhos de banda, inclusive em relação à implementação inicial.

Tabela III
Resultados com Banda de Sakoe-Chiba

Banda	Acurácia	Acertos	Total	Tempo (s)
0%	69.4792	667	960	12.81363
1%	69.4792	667	960	12.79068
5%	75.3125	723	960	12.88595
10%	81.4583	782	960	13.10823
20%	84.6875	813	960	14.5925
50%	83.75	804	960	18.89807
100%	83.8542	805	960	22.50858

```
int w = static_cast<int>(bandSize * (max(m, n)));
int constraint = abs(m-n);
w = max(w, constraint);

double DTW[m][n];

for(int i = 0; i < m; i++) {
    for(int j = 0; j < n; j++) {
        DTW[i][j] = infinity;
    }
}

DTW[0][0] = 0;

for (int i = 1; i < m; i++) {
    for (int j = max(1, i - w); j < min(n, i + w); j++) {
        double cost = euclideanDistance(s.second[i], t.second[j]);
        DTW[i][j] = cost +  minimum(DTW[i-1][j],        // insertion
                                    DTW[i][j-1],        // deletion
                                    DTW[i-1][j-1]);     // match
    }
}
```

Figura 1. Trecho de código da banda de Sakoe-Chiba.

A segunda extensão é relacionada à distância DTW entre séries temporais multidimensionais. Existem duas abordagens propostas *a priori* por [2]: a primeira, denominada *DTW_D*, consiste em somar as distâncias euclidianas de cada dimensão e realizar o cálculo da matriz por meio de tal somatório; já a abordagem *DTW_I* envolve o cálculo da distâncias DTW em cada dimensão e a posterior soma das mesmas. No código, a abordagem implementada foi a *DTW_D*.

A extensão do algoritmo deu-se por meio de um novo método para calcular a distância euclidiana. Este retorna o somatório das distâncias em cada dimensão, conforme a equação 3:

$$d(q_i, c_j) = \sum_{m=1}^{M} |q_{i,m} - c_{j,m}| \tag{3}$$

Onde M é a quantidade de dimensões e q e c são elementos de séries temporais multidimensionais. O resultado médio de tempos de execução e de acurácia para o caso tridimensional proposto é denotado pela tabela IV.

Tabela IV
RESULTADOS PARA O CASO TRIDIMENSIONAL

Acurácia	Acertos	Total	Tempo (s)
81.9652	659	804	58.38434

Verifica-se que a extensão tridimensional também obteve uma medida de acurácia considerável. O tempo de execução aumentou devido ao fato da quantidade de cálculos de distância euclidiana ser o triplo do original. As complexidades de espaço e tempo dos algoritmos de DTW em si e de 1-vizinho mais próximo continuaram as mesmas. Uma possível alteração para melhorar o tempo e a acurácia, de acordo com os resultados obtidos na extensão anterior, seria a implementação da banda de Sakoe-Chiba.

REFERÊNCIAS

[1] Bogue, E. T.; Matsubara, A. C.; Bessa, U. Uma Abordagem em Reconhecimento de Movimentos Utilizando TRKNN e Dynamic Time Warping. *ENIA 2012*, 2012.

[2] Shokoohi-Yekta, M.; Wang, J.; Keogh, E. On the Non-Trivial Generalization of Dynamic Time Warping to the Multi-Dimensional Case. *SDM 2015*, 2015.

[3] Dynamic Time Warping. https://en.wikipedia.org/wiki/Dynamic_time_warping.