

Universidade de São Paulo – USP
Instituto de Ciências Matemáticas e de Computação – ICMC
Departamento de Ciências de Computação – SCC

SCC0602 - Algoritmos e Estruturas de Dados I (2018)

Professor André Carlos Ponce de Leon Ferreira de Carvalho
andre@icmc.usp.br
PAE Rafael

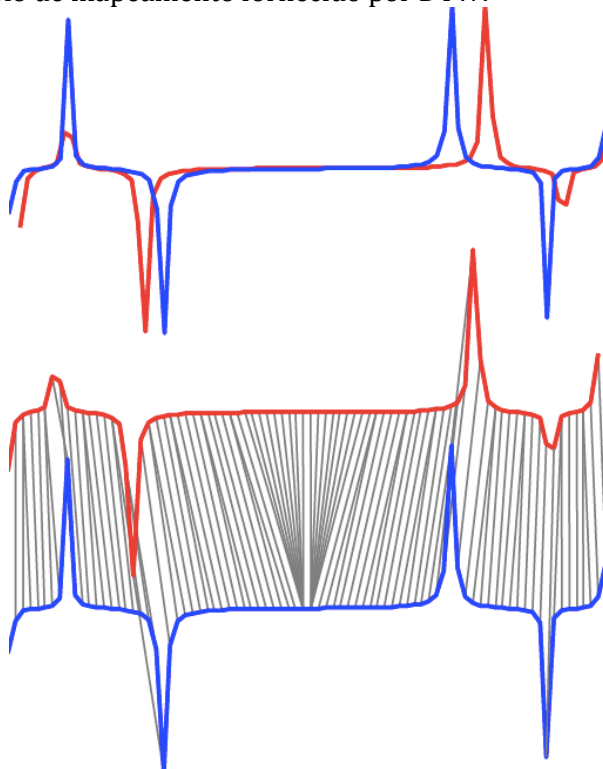
Projeto – Programação Dinâmica **Data de entrega: 07/06**

O objetivo deste projeto é implementar um algoritmo de classificação de séries temporais utilizando a distância *Dynamic Time Warping* (DTW) e dados obtidos a partir de acelerômetros.

Este projeto possui três partes descritas a seguir:

Parte I – Implementação da DTW

Nesta parte será implementado o algoritmo DTW que utiliza a programação dinâmica para estimar a distância entre duas séries temporais, conforme visto em sala de aula. A Figura 1 apresenta um exemplo de mapeamento fornecido por DTW.



A distância DTW entre duas séries temporais A e B é definida pela seguinte relação de recorrência [1]:

$$DTW(x_i, y_j) = c(x_i, y_j) + \min \begin{cases} DTW(x_{i-1}, y_{j-1}) \\ DTW(x_i, y_{j-1}) \\ DTW(x_{i-1}, y_j) \end{cases}$$

na qual $c(x_i, y_j)$ é uma métrica de distância, como a Euclidiana:

$$c(x_i, y_j) = \sqrt{(x_i - y_i)^2} = |x_i - y_i|$$

Parte II – Implementação do classificador de séries temporais

Nesta etapa, o objetivo é utilizar o algoritmo DTW para auxiliar um algoritmo de aprendizado de máquina a reconhecer e classificar séries temporais dentre um conjunto de classes específicas de movimentos realizados com um controle do videogame Wii. Para tal, será fornecida uma base de dados com séries temporais coletadas do acelerômetro do videogame[2].

Nessa base, existem 12 movimentos diferentes, tais como direita, esquerda, cima, baixo, quadrado, movimento em Z, etc. O arquivo rotulos.txt possui uma breve descrição dos movimentos.

As séries temporais estão armazenadas nos arquivos treino.txt e teste.txt. No total existem 1200 séries temporais nesses dois arquivos. Cada classe (movimento) é representada por 100 séries que são resultados de um voluntário executando o mesmo movimento 100 vezes.

O formato dos dados tanto no arquivo treino.txt quanto no teste.txt é o mesmo. Cada linha corresponde a uma série temporal. O primeiro valor de cada linha corresponde à classe da série temporal. Esse valor é um número de 1 a 12 correspondente aos rótulos armazenados no arquivo rotulos.txt. Até o final da linha ('\n'), segue uma quantidade variável de valores numéricos reais que são os valores obtidos do acelerômetro do controle do Wii.

Para classificar as séries temporais nós vamos utilizar um classificador bastante simples, conhecido como k -vizinhos mais próximos. Esse classificador consiste em dada uma série temporal Q do arquivo teste.txt, verificar quais k séries S no arquivo treino.txt são mais similares a ela (ou seja, quais são seus k -vizinhos mais próximos). Para a série Q é atribuída a classe majoritária dentre seus k vizinhos.

Alguns detalhes devem ser considerados nessa implementação:

- Qual o método de ordenação será empregado para ordenar os vizinhos por distância? Tente escolher aquele que é mais adequado para a operação.
- Como proceder se houver empate, ou seja, quando duas classes tiverem o mesmo número de votos (ex: $k = 5$, dentre os vizinhos há 2 votos para classe 1, 2 votos para classe 6 e um voto para a classe 10)? Resoluções inteligentes para esse problema poderão ser bonificadas.

Parte II – Experimentos e relatório

Uma pergunta importante é quão efetivo é esse classificador. Por exemplo, podemos pensar em um jogo real, que dada uma série temporal resultante de um movimento realizado por um jogador, precisamos classificar a série para que o personagem faça alguma ação. O classificador precisa ser preciso para que o jogo interprete corretamente as ações do jogador.

Nós podemos estimar o desempenho desse classificador utilizando o arquivo teste.txt. Nesse arquivo estão armazenadas séries temporais que serão utilizadas apenas para medir a qualidade de classificação. O procedimento é o seguinte:

Para cada série temporal Q do arquivo de teste, procure pelas k séries temporais mais similares a ela utilizando a distância DTW no arquivo de treino e faça a classificação com base nas classes das séries vizinhas. Se a classe atribuída for igual à classe de Q , então contabilizamos um acerto. Se forem diferentes, contabilizamos um erro. A medida final de desempenho é a taxa de acerto, ou seja, o número de acertos dividido pelo número total de séries temporais de teste.

Para que os experimentos sejam realizados, compare a taxa de acertos para execuções com valores $k = \{1, 3, 5, 10, 20\}$. Mostre esses resultados em tabelas e/ou gráficos. Mostre também gráficos de tempo de execução para as rotinas de DTW e ordenação de vizinhos separadamente, e por fim todo o conjunto de execução do algoritmo k -vizinhos mais próximos, para cada execução.

Faça um curto relatório (2 a 4 páginas) explicando a sua implementação e os resultados obtidos. Faça a análise de complexidade do algoritmo.

Observações:

1. Trabalho pode ser feito em duplas. Coloque o nome e NUSP de ambos em todos os arquivos.
2. O trabalho poderá ser feito em qualquer linguagem de programação, **contanto que** esteja bem documentado. Códigos não documentados terão desconto de até 10% na nota final.
3. Entrega deverá ser feita em um arquivo .zip ou .rar contendo relatório e códigos. Coloque o nome do arquivo como o NUSP dos dois alunos.

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

[2] Eduardo T. Bogue, Edson T. Matsubara, Anderson C. Bessa. *Uma Abordagem em Reconhecimento de Movimentos Utilizando TRKNN e Dynamic Time Warping*. ENIA 2012.