

**UNIVERSIDADE DE SÃO PAULO**  
**Instituto de Ciências Matemáticas e de Computação**

---

**Projeto 2**  
**Programação Dinâmica**

**Venilton Falvo Jr – 7902878**

---



# 1. Introdução

Este relatório documenta o segundo trabalho desenvolvido para a disciplina de Projetos de Algoritmos (SCC5900), que consiste na implementação de uma solução para a classificação de séries temporais utilizando a distância *Dynamic Time Warping* (DTW) e dados obtidos a partir de acelerômetros. O código-fonte submetido na plataforma Tidia-Ae também encontra-se disponível no GitHub<sup>1</sup>.

O objetivo deste documento é explicar a estratégia de implementação utilizada e discutir brevemente seus resultados. Para isso, a Seção 2 apresenta maiores detalhes sobre os elementos da solução desenvolvida. A Seção 3 expõem uma análise de performance, resumizando os resultados obtidos. Por fim, a Seção 4 apresentada a conclusão deste trabalho, bem como suas limitações e trabalhos futuros.

## 2. Implementação

A solução implementada utiliza um algoritmo de classificação de séries temporais baseado na distância DTW, toda lógica foi construída na linguagem Java em sua versão 8. Tal implementação conta com algumas classes responsáveis por domínios específicos do problema em questão. A seguir, as principais dentre elas são explanadas de maneira sucinta.

### 2.1. br.usp.falvojr.dtw.Main

A classe *Main* é a classe principal do projeto. Nela os argumentos de execução são processados com o objetivo de computar os dados dos sensores, especificados nos arquivos de entrada. Tais argumentos são detalhados na tabela abaixo:

**Tabela 1:** Argumentos de Execução

Argumento	Descrição	Sintaxe
-d [path]	Define um diretório base para a obtenção dos arquivos de treino e testes.	\$ java -jar Project2.jar -d [path]
-3d	Ativa a estratégia que utiliza 3 dimensões.	\$ java -jar Project2.jar -d [path] -3d
-w [value]	Especifica o valor da restrição de banda Sakoe-Chiba.	\$ java -jar Project2.jar -d [path] -w [value] \$ java -jar Project2.jar -d [path] -3d -w [value]

<sup>1</sup> Disponível em: <https://github.com/falvojr-phd/scc5900/tree/master/Project2>

A partir dos argumentos, a classe *Main* processa os arquivos de rótulos, treino e teste, considerando suas respectivas nomenclaturas e variações tridimensionais (sufixo 3D). Com isso, a classe interpreta os arquivos de entrada estruturando e executando o algoritmo DTW com suas devidas extensões.

## 2.2. **br.usp.falvojr.dtw.algorithm.DynamicTimeWarping**

Classe responsável pela implementação do algoritmo DTW e suas respectivas variações. Em termos técnicos, essa classe implementa o padrão de projeto Singleton, garantindo uma única instância para seus objetos. Além disso, a classe expõe os seguintes métodos para o cálculo da distância DTW:

- **dtwDistance**: método responsável pela implementação básica (considerando apenas a relação de recorrência em questão) da distância DTW;
- **dtwDistance (sobrecarga)**: representa uma variação de sua implementação básica, considerando um parâmetro adicional  $w$ , que representa a grandeza da banda Sakoe-Chiba (Extensão 1).

## 2.3. **br.usp.falvojr.dtw.classification.OneNearestNeighbor**

Classe que implementa a interface *NearestNeighbor*, responsável por definir os contratos para os métodos de classificadores do tipo K-Vizinhos Mais Próximos (do inglês, K-Nearest Neighbor (KNN)). Sua lógica representa o classificador 1-Nearest Neighbor (1NN) e possui as seguintes operações:

- **computeAccuracyRate**: identifica com base em duas séries temporais a série de treino mais similar às séries de testes, utilizando a distância DTW no arquivo de treino. A medida final de retorno é o número de acertos considerando o classificador em questão;
- **computeAccuracyRate (sobrecarga)**: representa uma variação do método descrito acima, com um parâmetro adicional  $w$ , que representa o valor da banda Sakoe-Chiba (Extensão 1), repassado para o cálculo da distância DTW.

## 2.4. **br.usp.falvojr.dtw.classification.TreeNearestNeighbor**

Assim como a classe descrita anteriormente, essa também é uma implementação da interface *NearestNeighbor*. Entretanto, provê um classificador baseado nos 3-Nearest Neighbor (3NN), implementado para que o algoritmo DTW seja aplicado em um contexto tridimensional.

### 3. Performance e Complexidade

Para a análise de performance duas variáveis foram mensuradas: tempo de execução e taxa de acerto. Para calcular o tempo foi utilizada a diferença entre duas chamadas do método `System.nanoTime` do Java. Os tempos apresentados nesta seção são as médias de 10 execuções realizadas. Por outro lado, a taxa de acertos é medida com base na razão de series compatíveis considerando os dados de treino e testes. As execuções foram realizadas usando o SO Ubuntu 14.04 LTS x64 em um notebook com processador 2.00 GHz Intel Core i7-2630QM e 8 GB de RAM.

As tabelas a seguir resumizam os tempos de execução e taxas de acerto para os arquivos de entrada propostos para este projeto. A Tabela 2<sup>2</sup> apresenta os resultados obtidos através da implementação básica do algoritmo DTW com o classificador 1NN.

**Tabela 2:** Resultados da medida DTW com o classificador 1NN

Dimensão	Tempo (seg)	Razão de Acertos	Taxa de Acertos
1D	11.968	814/960	<b>84.79%</b>
3D	142.040	636/804	<b>79.10%</b>

Considerando a medida final de desempenho como a taxa de acerto, podemos considerar que o algoritmo DTW com o simples classificador 1NN foi capaz de resolver em um tempo razoável. Entretanto pode-se verificar que para as entradas 3D o tempo de execução cresceu consideravelmente.

Adicionalmente, considerando a Extensão 1, procuramos melhorar o desempenho da medida DTW tanto em termos de tempo de execução quanto em termos de desempenho de classificação. Para isso, a banda Sakoe-Chiba (representada pela letra W) foi aplicada. Nesse sentido, seus resultados são exibidos na Tabela 3<sup>2</sup>.

**Tabela 3:** Resultados da medida DTW com a banda Sakoe-Chiba e classificador 1NN

Dimensão	W (%)	Tempo (seg)	Razão de Acertos	Taxa de Acertos
1D	0	11.897	773/960	<b>80.52%</b>
	1	11.790	773/960	<b>80.52%</b>
	5	11.766	783/960	<b>81.56%</b>
	10	11.956	799/960	<b>83.23%</b>
	20	12.411	823/960	<b>85.73%</b>
	50	14.220	815/960	<b>84.90%</b>
	100	16.311	814/960	<b>84.79%</b>

<sup>2</sup> A planilha com os detalhes dos dados coletados está disponível em: <https://goo.gl/WdoScx>.

3D	0	132.810	660/804	<b>82.09%</b>
	1	132.231	660/804	<b>82.09%</b>
	5	131.669	659/804	<b>81.97%</b>
	10	134.289	653/804	<b>81.22%</b>
	20	140.032	650/804	<b>80.85%</b>
	50	162.532	636/804	<b>79.10%</b>
	100	185.518	636/804	<b>79.10%</b>

Analisando os resultados obtidos utilizando a banda Sakoe-Chiba é possível identificar um aumento considerável na taxa de acertos, principalmente considerando a dimensão 3D, que obteve todas as taxas maiores ou iguais à obtida na Tabela 2. Por outro lado, em termos de tempo de execução, os resultados foram equivalentes de modo geral.

Finalmente, em atenção a Extensão 2, foi implementada uma variação da solução para trabalhar com séries temporais com três dimensões. Para isso o classificador 1NN foi adaptado para considerar os três vizinhos mais próximos (3NN). Os resultados obtidos para essa estratégia são sintetizados na Tabela 4<sup>3</sup>.

**Tabela 4:** Resultados da medida DTW com a banda Sakoe-Chiba e classificador 3NN

Dimensão	W (%)	Tempo (seg)	Razão de Acertos	Taxa de Acertos
3D	0	43.000	201/268	<b>75.00%</b>
	1	41.995	201/268	<b>75.00%</b>
	5	42.532	200/268	<b>74.63%</b>
	10	43.690	199/268	<b>74.25%</b>
	20	45.605	201/268	<b>75.00%</b>
	50	52.077	199/268	<b>74.25%</b>
	100	61.252	198/268	<b>73.88%</b>

A partir dos resultados obtidos é possível concluir que essa estratégia reduziu consideravelmente o tempo de execução, provavelmente devido a diminuição das iterações na implementação do classificador 3NN. Entretanto, se mostrou ligeiramente menos eficiente em termos de taxa de acertos.

Por fim, analisando a complexidade da solução, conclui-se que ela é de  $O(n^4)$ . Isso porque a implementação final deste projeto contou com duas partes fundamentais: o algoritmo DTW e o classificador KNN, ambos com complexidade de  $O(n^2)$ .

<sup>3</sup> A planilha com os detalhes dos dados coletados está disponível em: <https://goo.gl/WdoScx>.

## **4. Conclusão e Trabalhos Futuros**

Com base nos resultados obtidos foi possível visualizar de modo prático a aplicação do conceito de classificação de séries temporais utilizando a distância DTW. A medida final de desempenho se mostrou bastante elevada, independentemente da estratégia de implementação utilizada, ficando geralmente com cerca de 80% na taxa de acerto.

Como trabalhos futuros, os classificadores 1NN e 3NN podem ser refatorados de uma maneira genérica, evitando múltiplas implementações dessa mesma abordagem. Alternativamente, a implementação tridimensional pode ser reavaliada, considerando uma evolução que contemple essa lógica no algoritmo DTW, diferentemente da estratégia utilizada neste projeto, que está vinculada ao classificador 3NN.