SCC5900
Relatório Trabalho 3 – DTW

Rafael Miranda Lopes

nUSP 6520554

Implementação

Foram implementados o DTW básico, o DTW com banda de Sakoe-Chiba e a extensão para séries multidimensionais. Esta foi implementada como uma generalização para N > 0 dimensões e inclui o caso de séries unidimensionais. Durante testes na implementação, foi verificado que essa generalização não causa grande impacto no tempo de execução e, por isso, foi feita uma implementação única genérica em relação às dimensões, mas distintas quanto ao uso da banda. Para fazer a leitura dos arquivos para a memória, é necessário fornecer o número de dimensões das séries. Para fazer uma consulta, é necessário fornecer os conjuntos de séries a serem comparados, o número de dimensões das séries e o valor da banda, sendo que valores negativos indicam que a banda não é utilizada. Uma única matriz de memorização é utilizada para todos os cálculos de distâncias.

DTW N-Dimensional

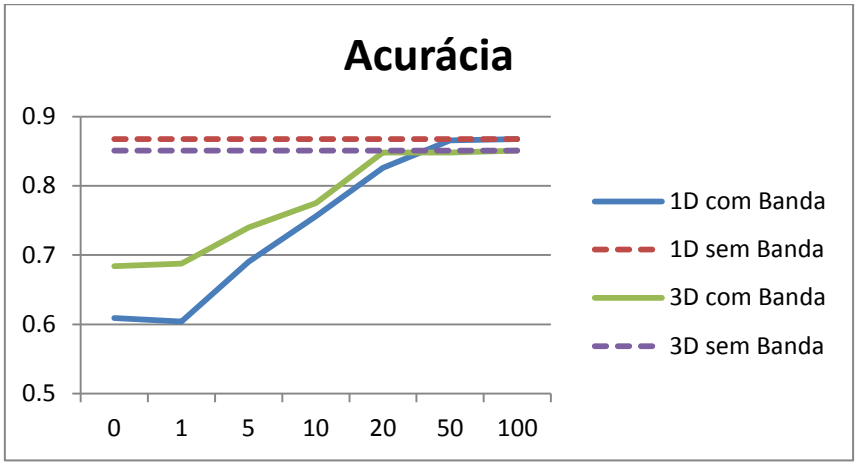
Para calcular as distâncias entre séries N-dimensionais, basta fornecer o valor de N > 0 na entrada dos dados. A função *PointDist()* calcula a distância entre dois pontos de duas séries. Esse cálculo é utilizado considerado que as N dimensões dos acelerômetros são dependentes e é realizado em uma matriz de memorização única para todas as dimensões. Essa função foi implementada para compor como resultado, alternativamente, a distância euclidiana ao quadrado (soma de quadrados das distâncias ponto a ponto) e para a distância de Manhattan (soma dos absolutos). Esta mostrou-se bem mais rápida e sem efeito considerável na acurácia e foi, portanto, utilizada por padrão.

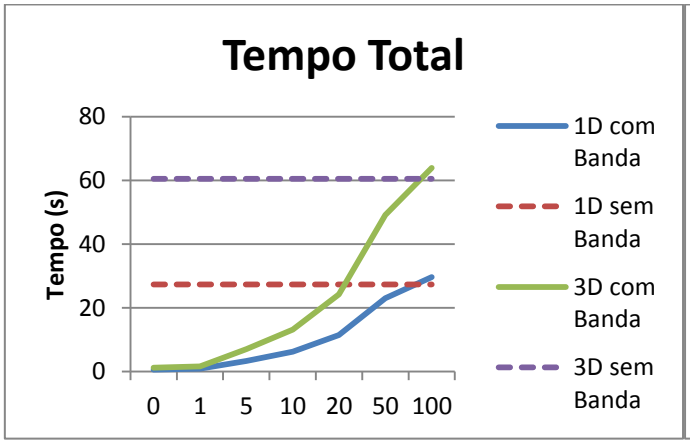
Banda Sakoe-Chiba

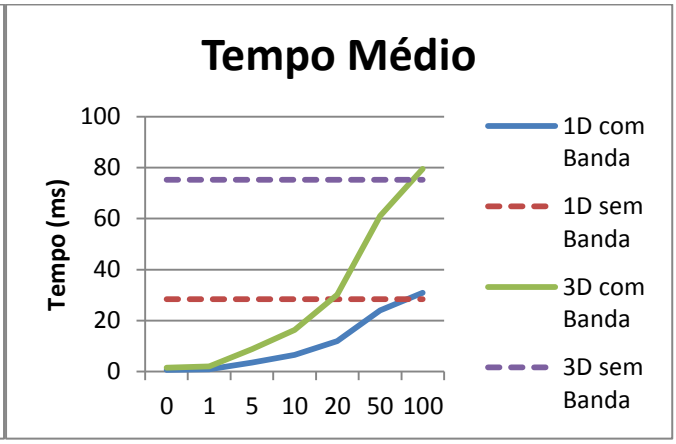
A banda de Sakoe-Chiba foi implementada para aceitar um valor de 0 a 100, em que esse valor é a porcentagem do comprimento da maior série entre o par a ser comparado e o resultado deste cálculo é o valor absoluto da banda, *bandW*. Como, para séries de tamanhos distintos e bandas muito pequenas, há a possibilidade do vértice final nunca ser alcançado (quando w < |len1-len2|), a série de maior comprimento foi cortada à direita, quando necessário, para que |len1-len2| <= w. Isso traz um problema: algumas séries são muito pequenas comparadas a outras e, como o valor da distância é crescente ao longo da matriz de memorização, séries muito pequenas acabam tendo o valor de distância reduzido, o que prejudica a acurácia. Para resolvê-lo, a distância calculada na matriz é dividida por min(len1, len2), após o corte anteriormente explicado, resultando em grande ganho de acurácia, especialmente para bandas muito pequenas. O mesmo ajuste foi feito para o algoritmo sem banda; neste, sem mudanças significativas na acurácia.

Outro detalhe da implementação importante de ressaltar é a inicialização da matriz de memorização. A cada comparação entre duas séries, apenas são preenchidas com INFINITO as duas retas paralelas à diagonal (tamanho d) que delimitam a largura da banda, apresentando complexidade de tempo O(d) em vez de $O(d^2)$ – caso a matriz seja completamente preenchida (para i,j > 0). Essa decisão teve grande impacto no tempo de execução (redução de até cerca de 50% em alguns casos).

Resultados







Pode-se perceber que os resultados para uma dimensão ou três dimensões apresentou tendências muito próximas, tanto de acurácia quanto de tempo. Bandas de até 20% reduziram a menos da metade o tempo de execução, sem grande prejuízo à acurácia, especialmente para três dimensões. Bandas menores podem ainda tornar o algoritmo ainda muito mais rápido, porém com maior perda de acurácia, especialmente para os casos de uma dimensão.

A complexidade do algoritmo é O(t*st*sv), onde *t* é o número de casos no conjunto de treinamento, *st* é o tamanho dos vetores nesse conjunto e *sv* é o tamanho do vetor a ser classificado. O cálculo da distância tem grande impacto no tempo de execução, como pode ser percebido pela comparação dos tempos de execução utilizando distância euclidiana ao quadrado (sem tirar a raiz ao final) ou a distância de Manhattan, conforme a tabela abaixo, indicando que a euclidiana foi cerca de 50% mais lenta.

	Tempo Médio (ms)	
	Euclidiana	Manhattan
1D	20	13
3D	53	34