Recent Advances in Assessing Time Series Similarity Through Dynamic Time Warping

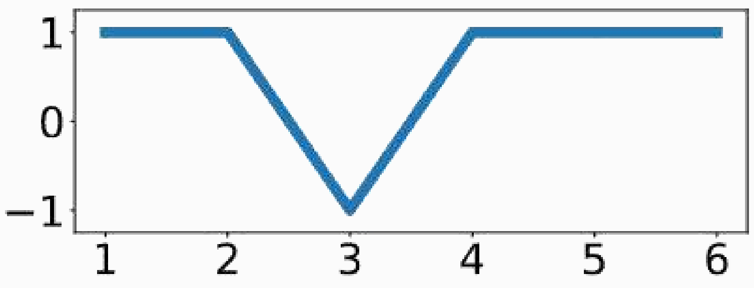
Geoff Webb
Monash University
http://i.giwebb.com
Work with C. W. Tan, M Herrmann & F. Petitjean

Outline

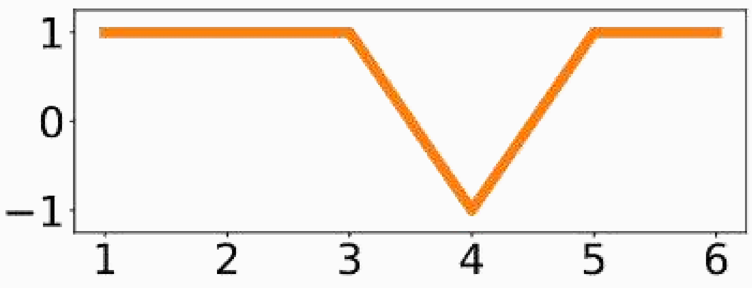
- Similarity assessment for time series
  - Dynamic Time Warping (DTW)
- Fast DTW computation
  - Early abandoning and pruning
- DTW variants
  - Cost function tuning
  - Amerced Dynamic Time Warping

Time series

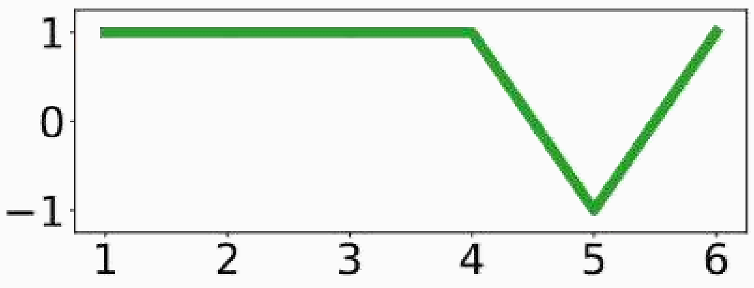
Time series



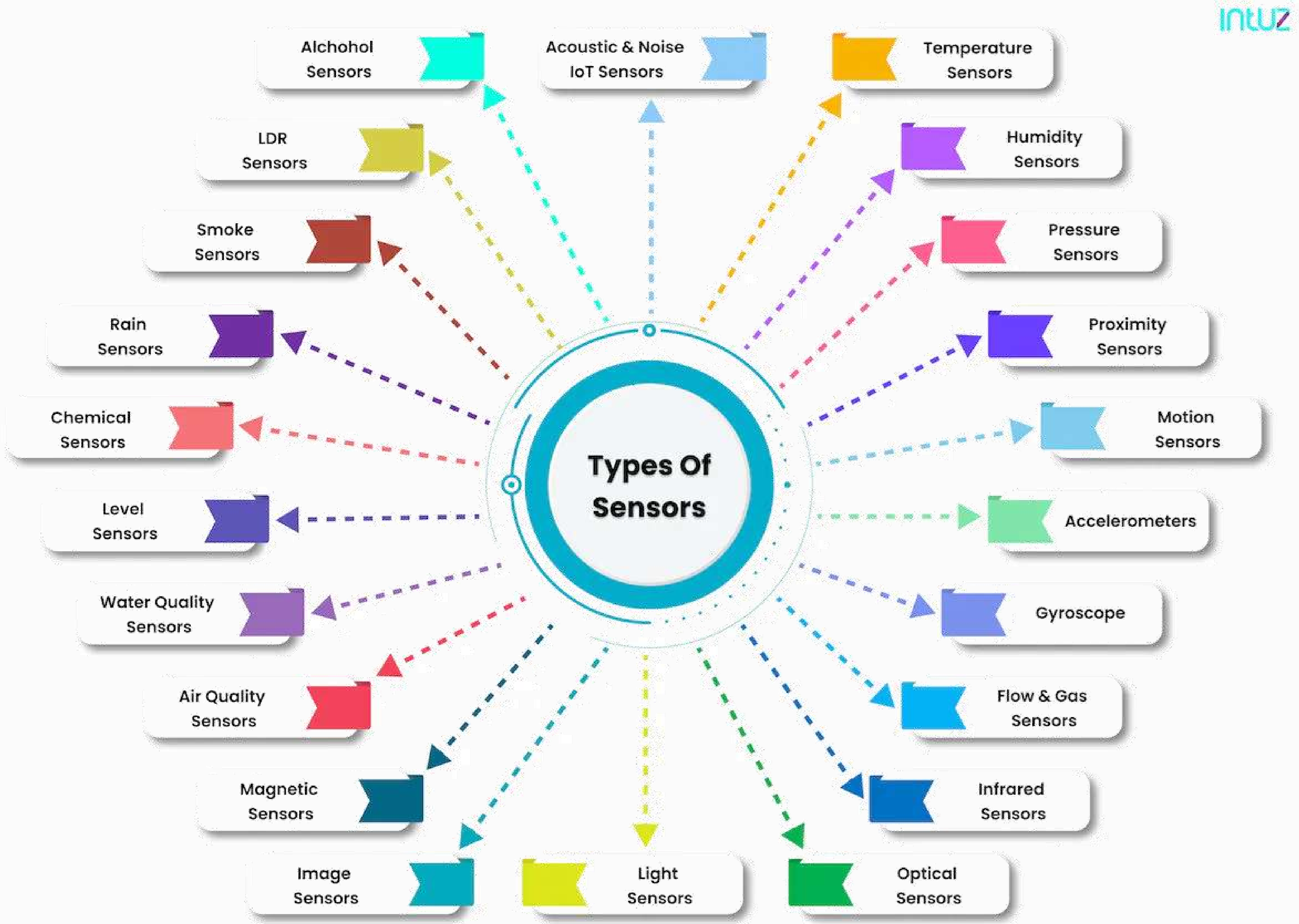
(a) $S = \{1, 1, -1, 1, 1, 1\}$



(b) $T = \{1, 1, 1, -1, 1, 1\}$



(c) $U = \{1, 1, 1, 1, -1, 1\}$



https://www.intuz.com/guide-on-top-iot-sensor-types

Applications

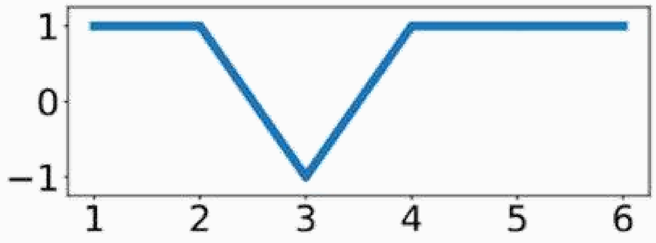
- Finance
- Health
- Environmental monitoring
- Equipment monitoring and control
- Process monitoring and control
- Online systems
- Logistics

Similarity assessment for time series

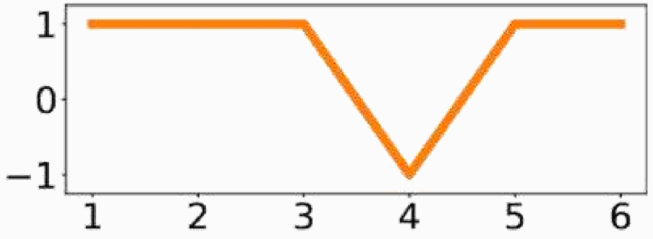
Similarity assessment is foundational for data science

Underpins

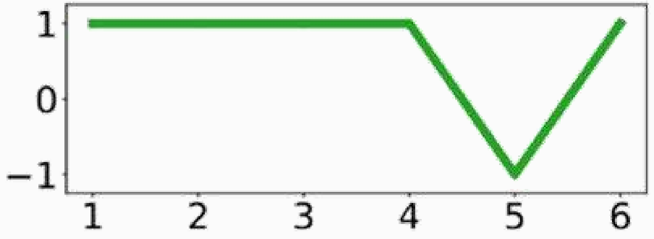
- Classification
- Regression
- Clustering
- Anomaly & outlier detection
- Sequence alignment
- Recommender Systems
- Feature extraction
- Information retrieval



(a) $S = \{1, 1, -1, 1, 1, 1\}$



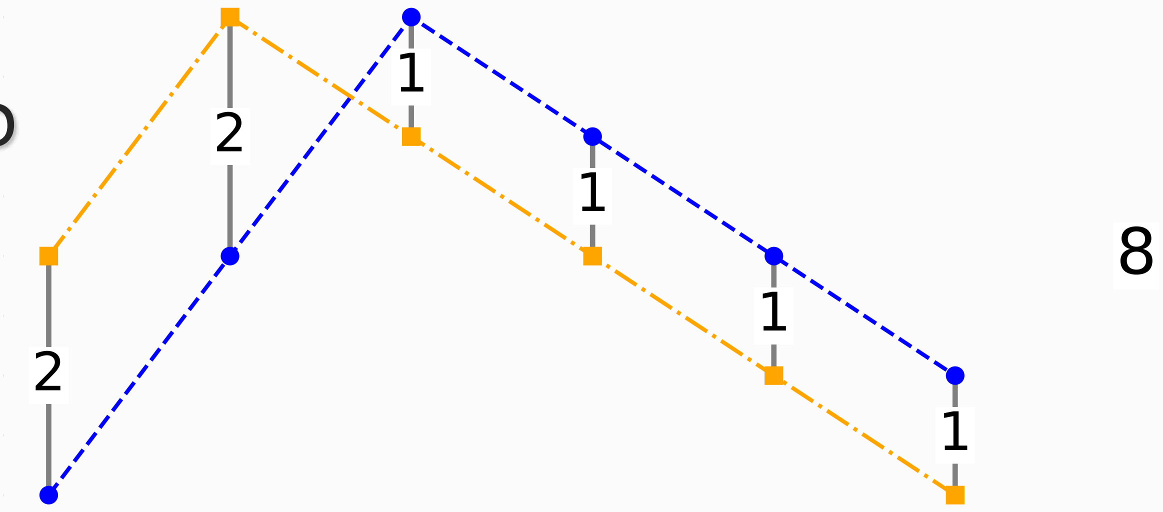
(b) $T = \{1, 1, 1, -1, 1, 1\}$



(c) $U = \{1, 1, 1, 1, -1, 1\}$

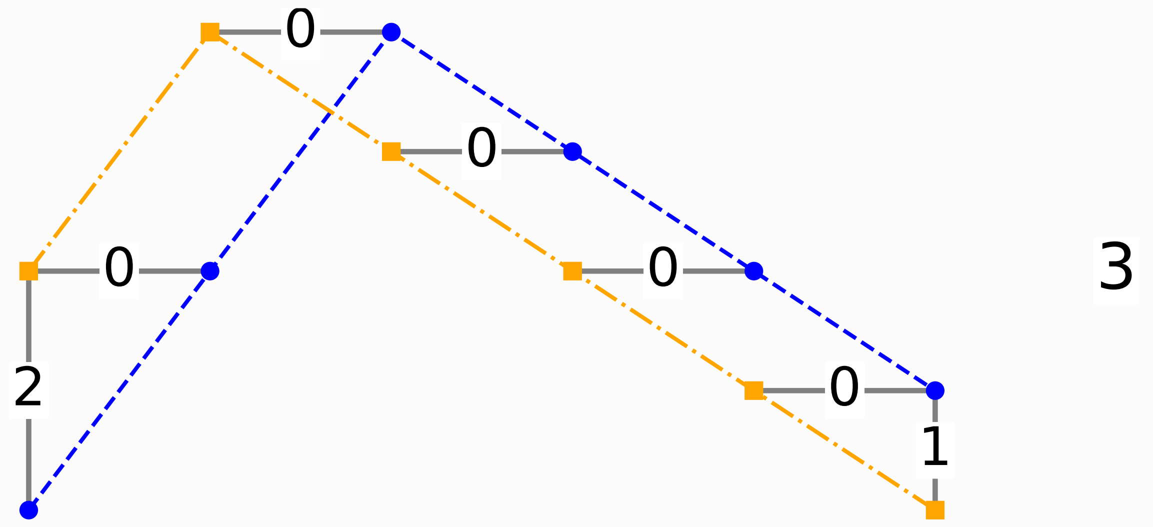
Time series distance measures

- Assess similarity in terms of *distance* between series
- *Direct Alignment* sums differences between points at same time step



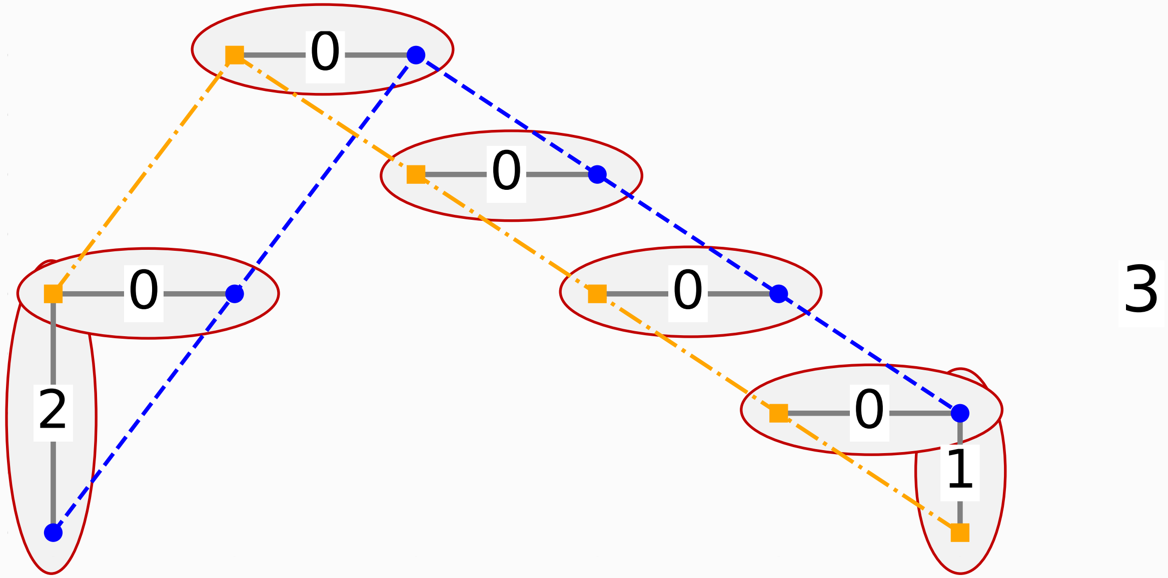
Time series distance measures

- Assess similarity in terms of *distance* between series
- *Direct Alignment* sums differences between points at same time step
- *Dynamic Time Warping* allows alignments across time steps



Dynamic Time Warping (DTW)

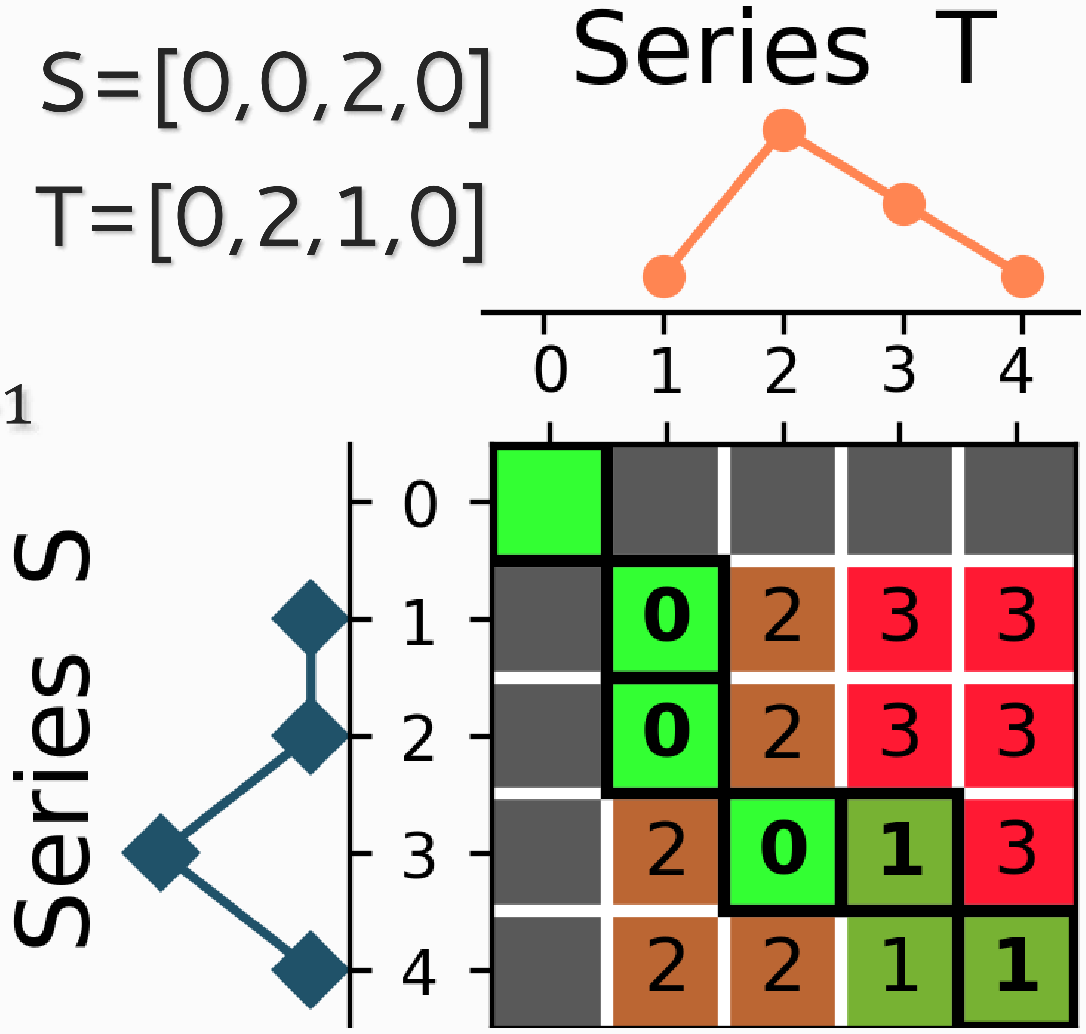
- Popular distance measure for time series
- First points are aligned
- Last points are aligned
- Successive alignments advance by at most one time step along each series



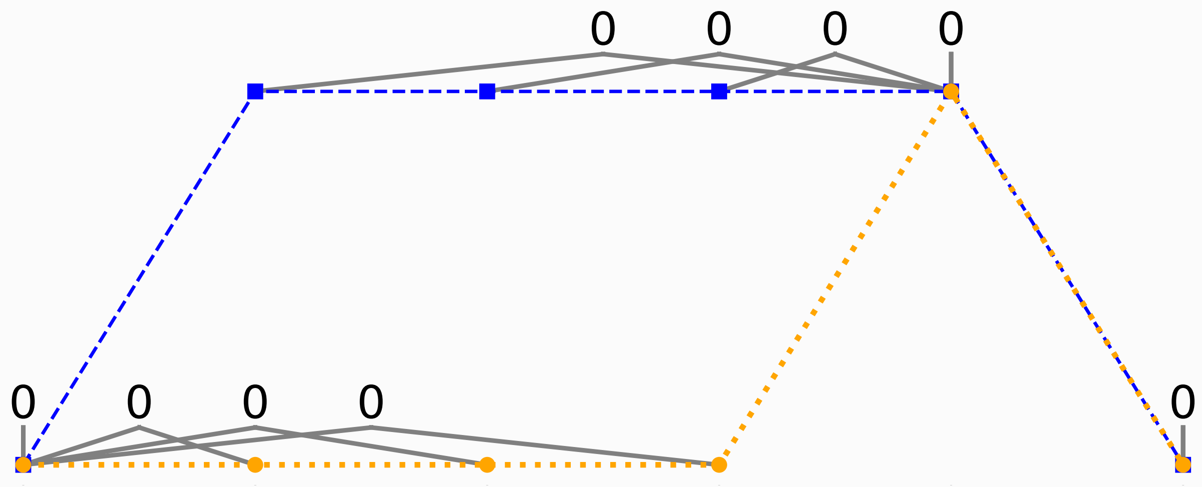
- Distance = minimum cost path that satisfies these constraints

Dynamic programming calculates DTW efficiently

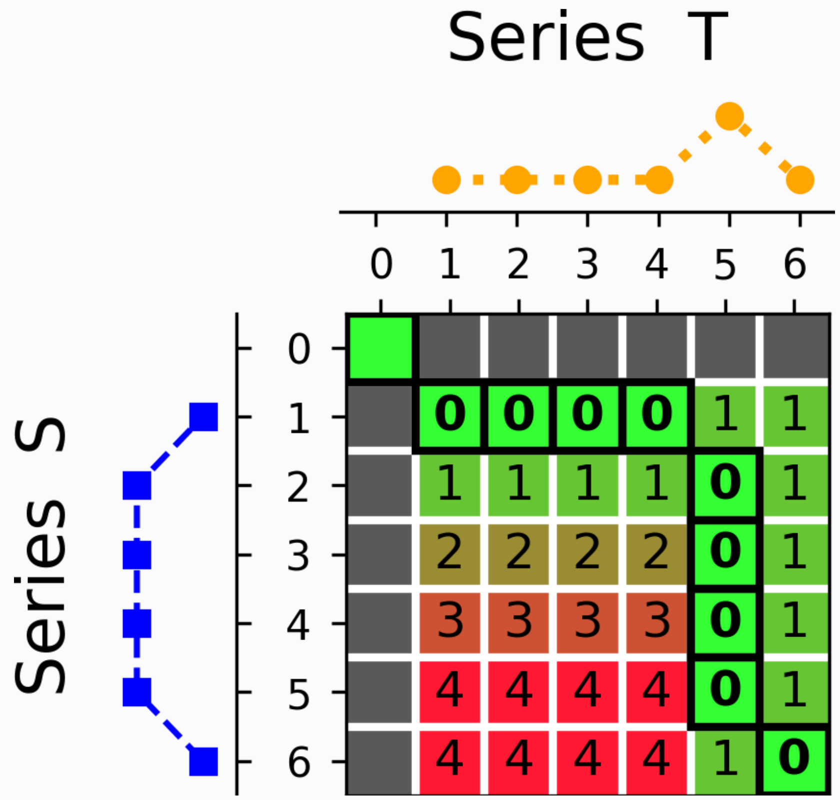
- $\mathrm{M}_{0,0}=0$
- $\mathrm{M}_{0,j} = \mathrm{M}_{i,0} = \infty$
- $\mathrm{M}_{i,j} = \lambda(S_i, T_j) + \min\begin{cases}\mathrm{M}_{i-1,j-1}\\ \mathrm{M}_{i-1,j}\\ \mathrm{M}_{i,j-1}\end{cases}$
- $\mathrm{DTW}(S,T) = \mathrm{M}_{\mathrm{len}(S),\mathrm{len}(T)}$
- Original $\lambda(S_i, T_j) = |S_i - T_j|$



DTW’s warping can be too permissive



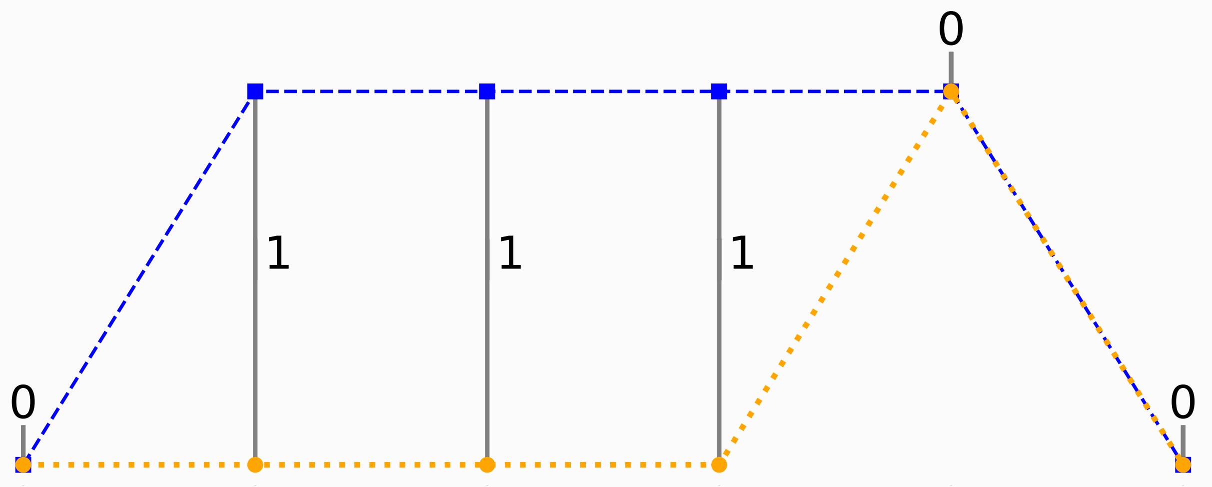
DTW = 0



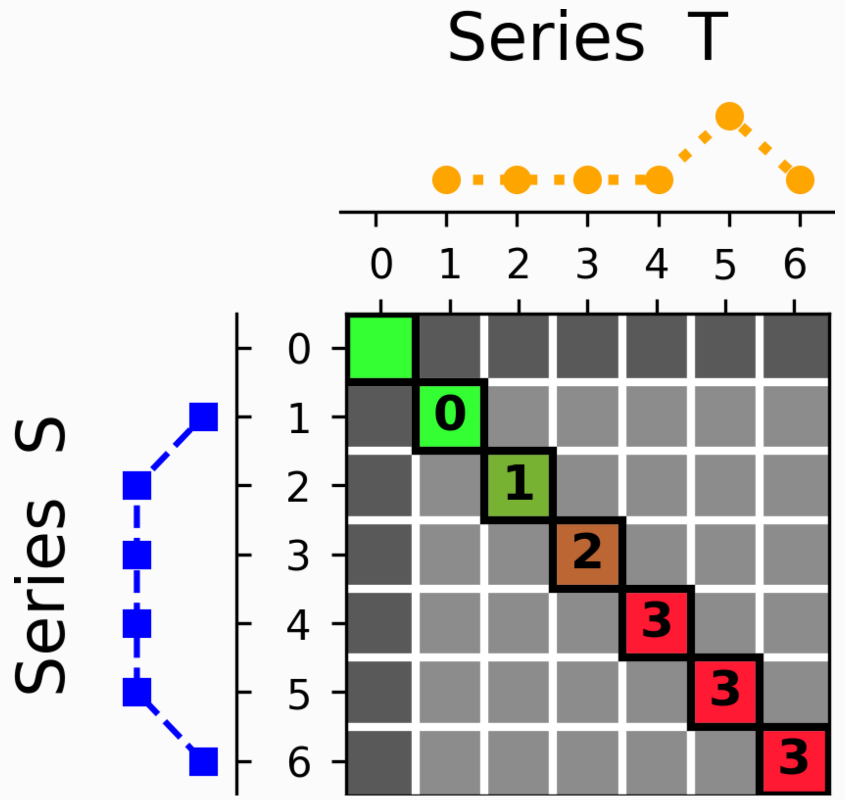
Windowing seeks to control this

- Adds further constraint: points cannot be aligned if separated by more than WINDOW time steps
- Distance is path with minimum cost that obeys constraints

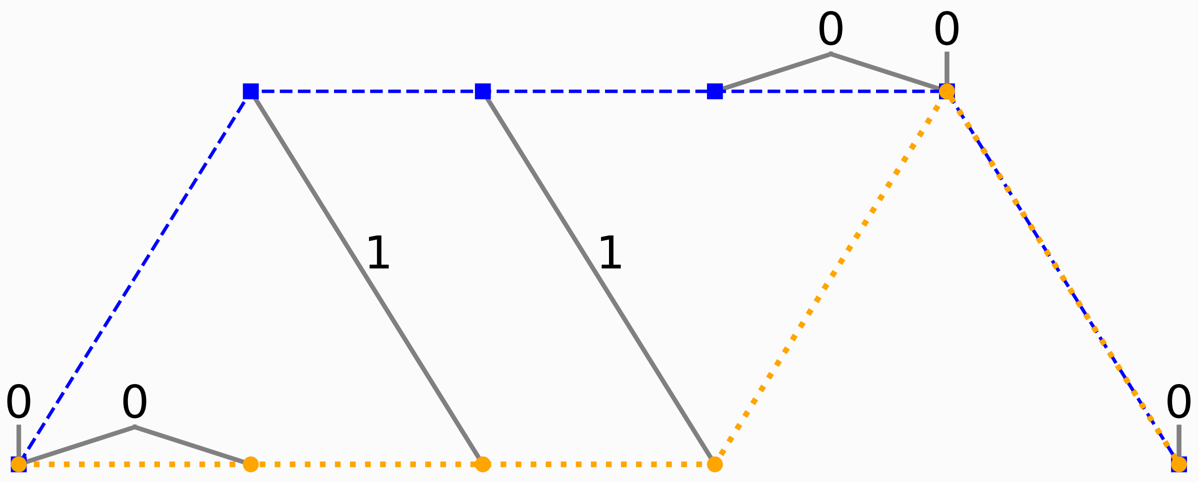
Windowing seeks to control this

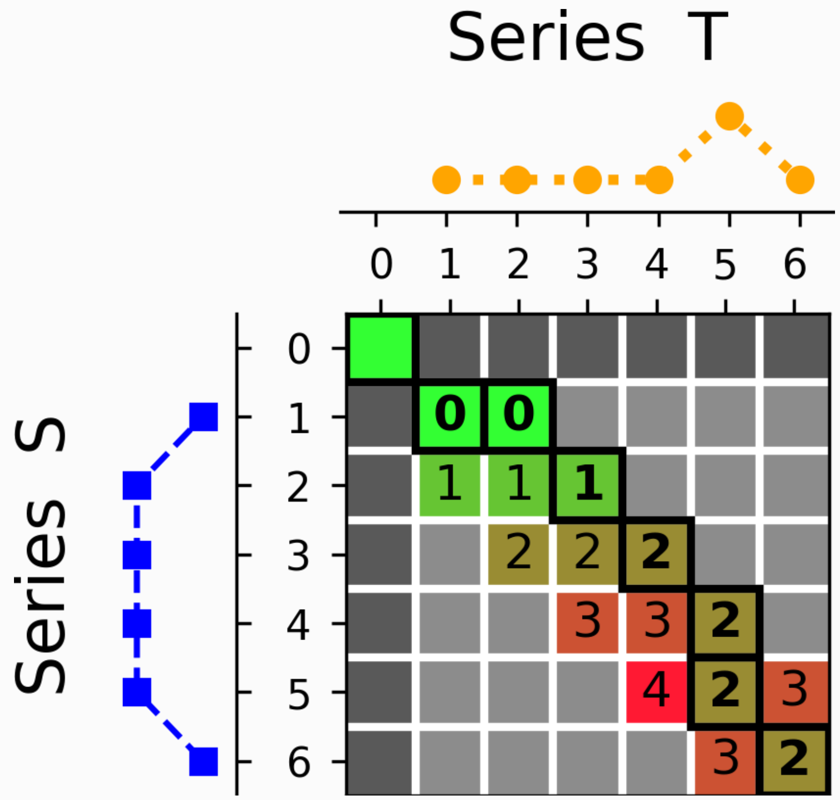


Window=0
DTW = 3



Windowing seeks to control this

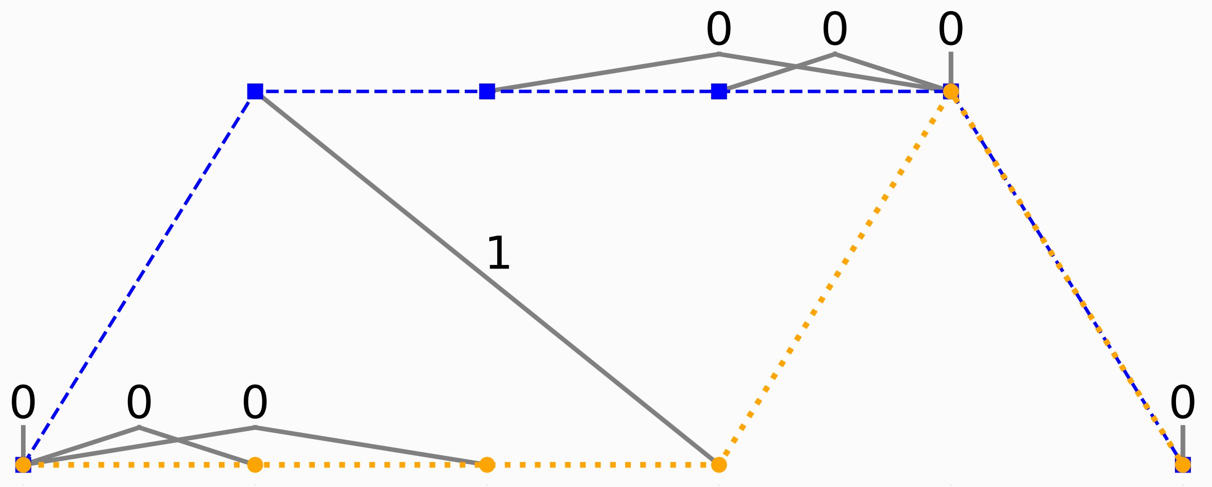


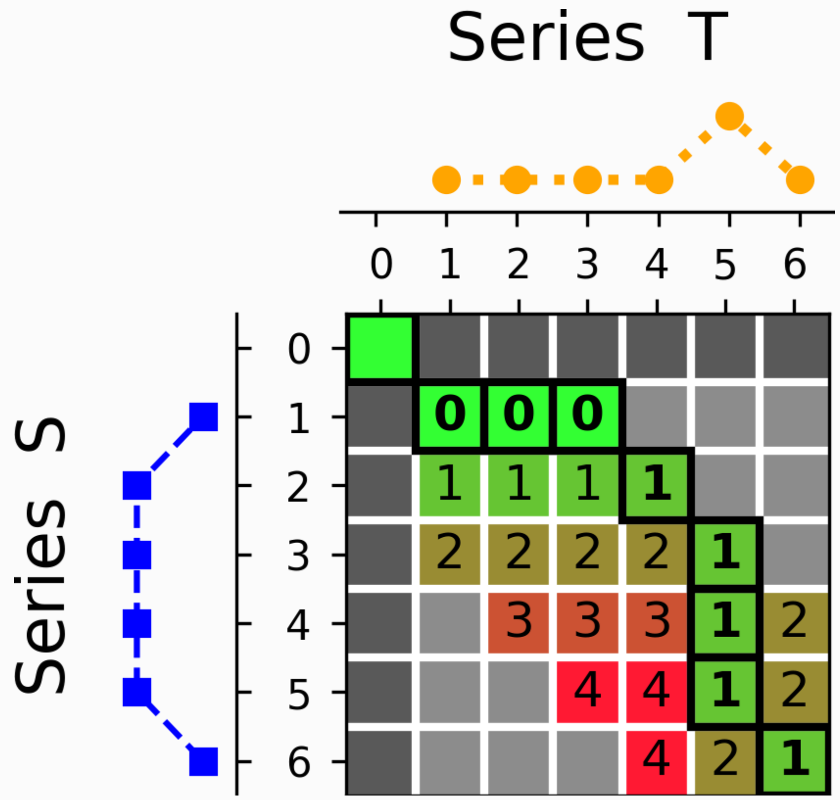


Window=1

DTW = 2

Windowing seeks to control this

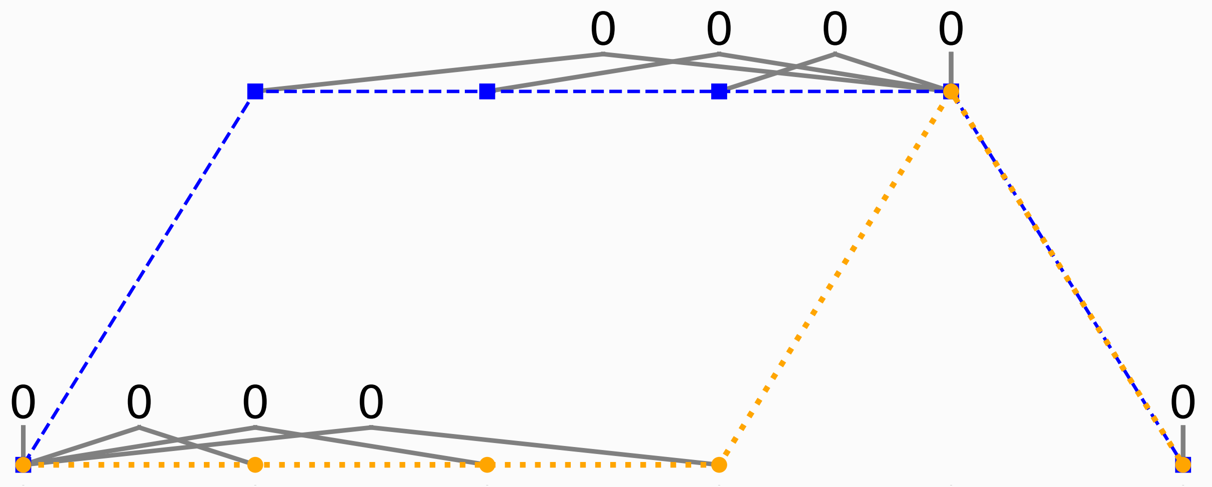




Window=2

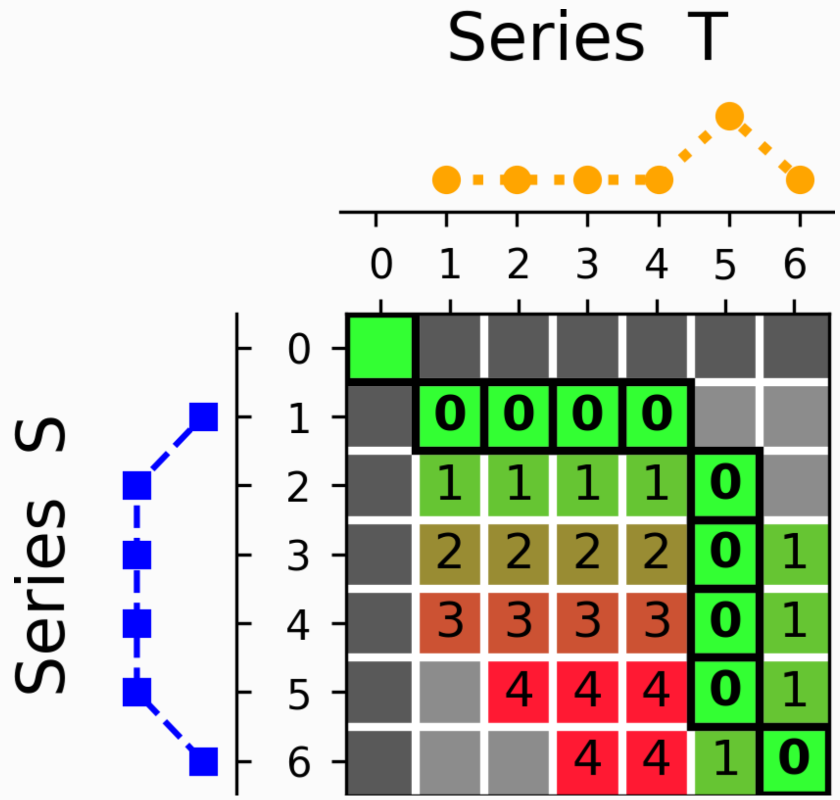
DTW = 1

Windowing seeks to control this

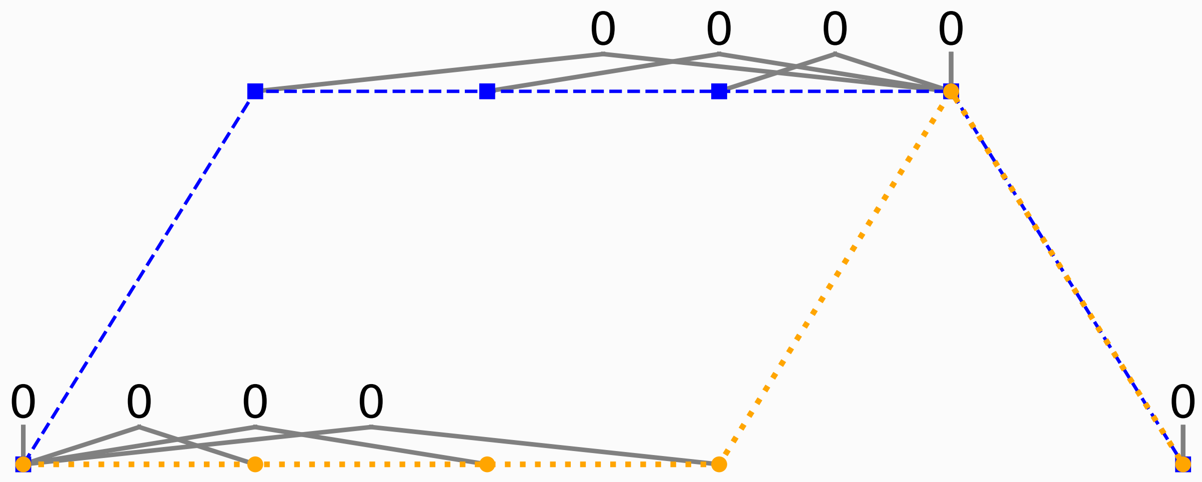


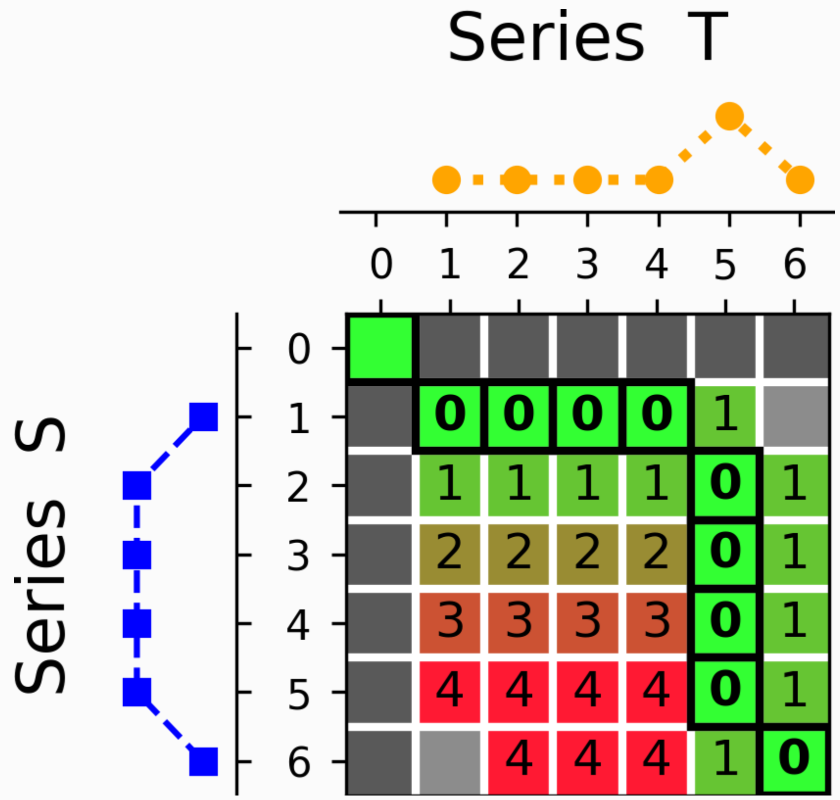
Window=3

DTW = 0



Windowing seeks to control this





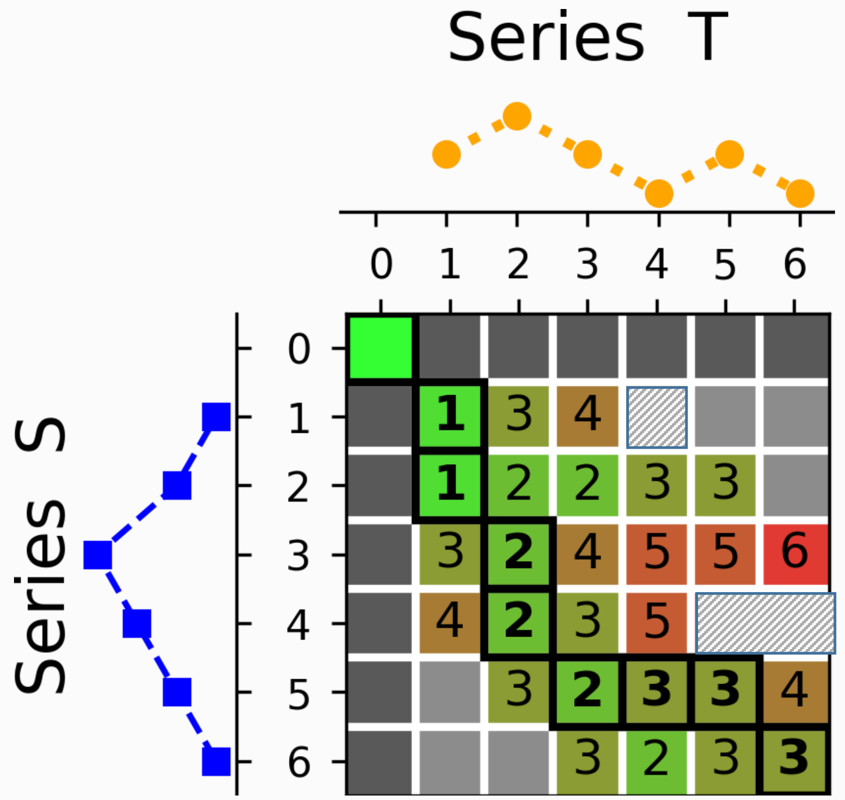
Window=4

DTW = 0

Fast DTW computation

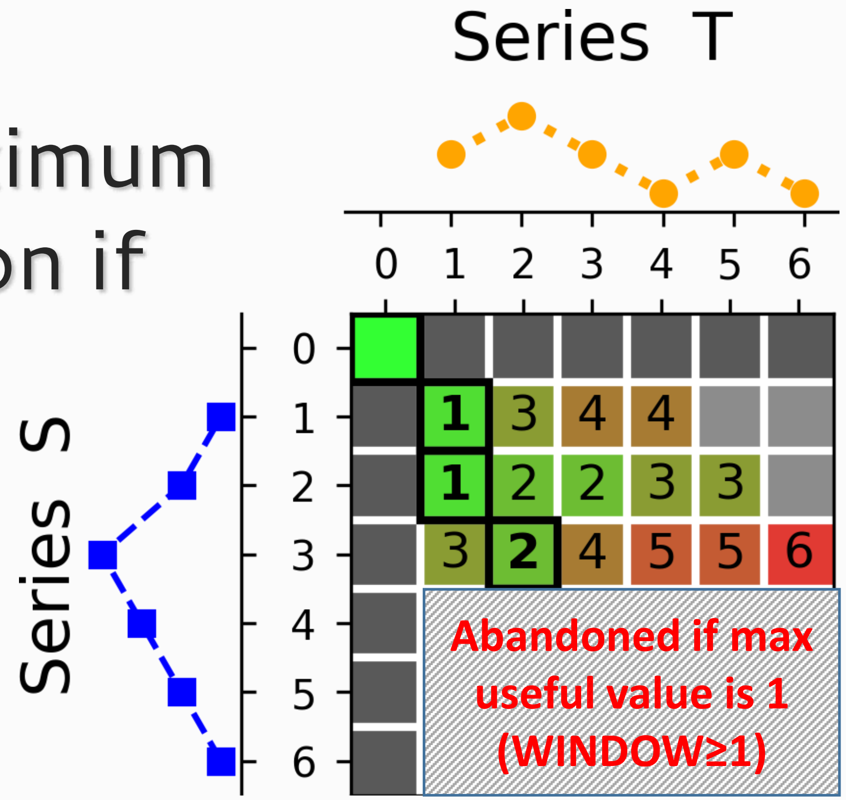
Fast DTW computation

- Naïve approach must fill the entire matrix
  - ➢ $O\big(\mathrm{len}(S) \times \mathrm{len}(T)\big)$
- **Pruning**: Given a maximum useful value, skip computation of cells that are on paths that exceed that value
  - ➢ Either cost of direct alignment path or an external factor such as the distance to the closest neighbour found so far



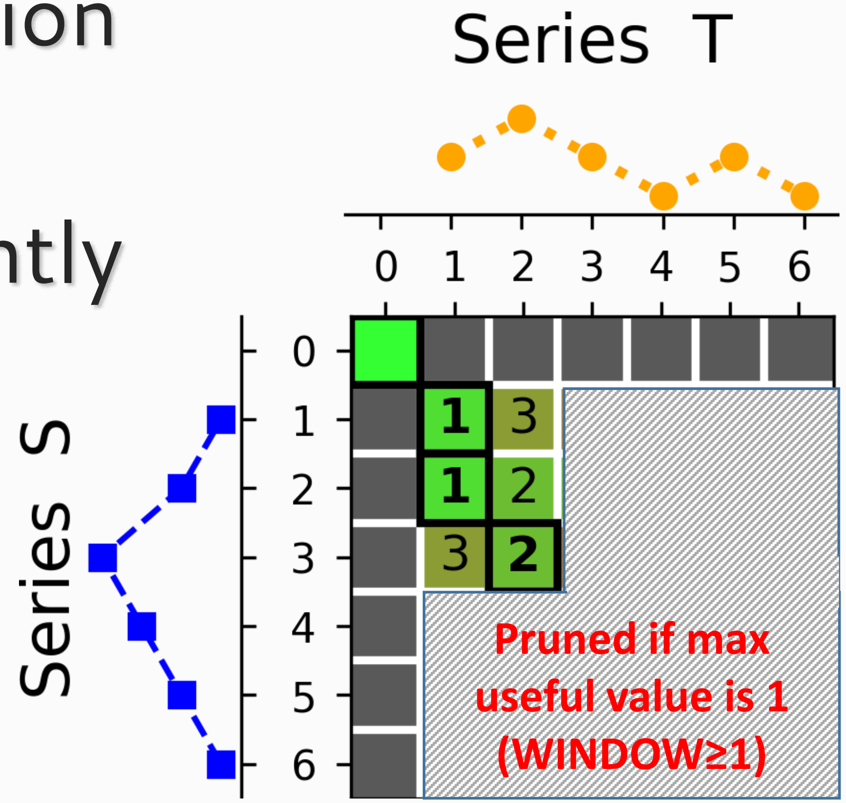
Fast DTW computation

- Naïve approach must fill the entire matrix
  - $O(\mathrm{len}(S) \times \mathrm{len}(T))$
- **Early Abandoning**: Given a maximum useful value, abandon computation if all paths exceed that value



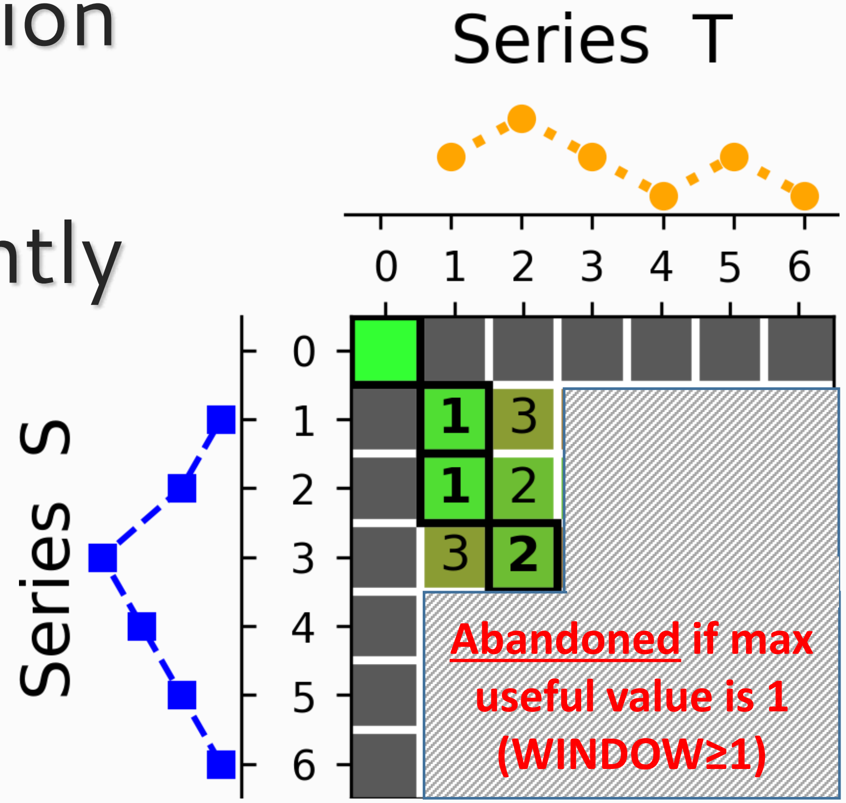
Our Method Early Abandoning AND Pruning

- Based on realization that when **all paths are pruned** the computation should be **abandoned**.
- Implements pruning more efficiently than previous approaches
- Unlike previous approaches, achieves abandoning without any significant computational overhead

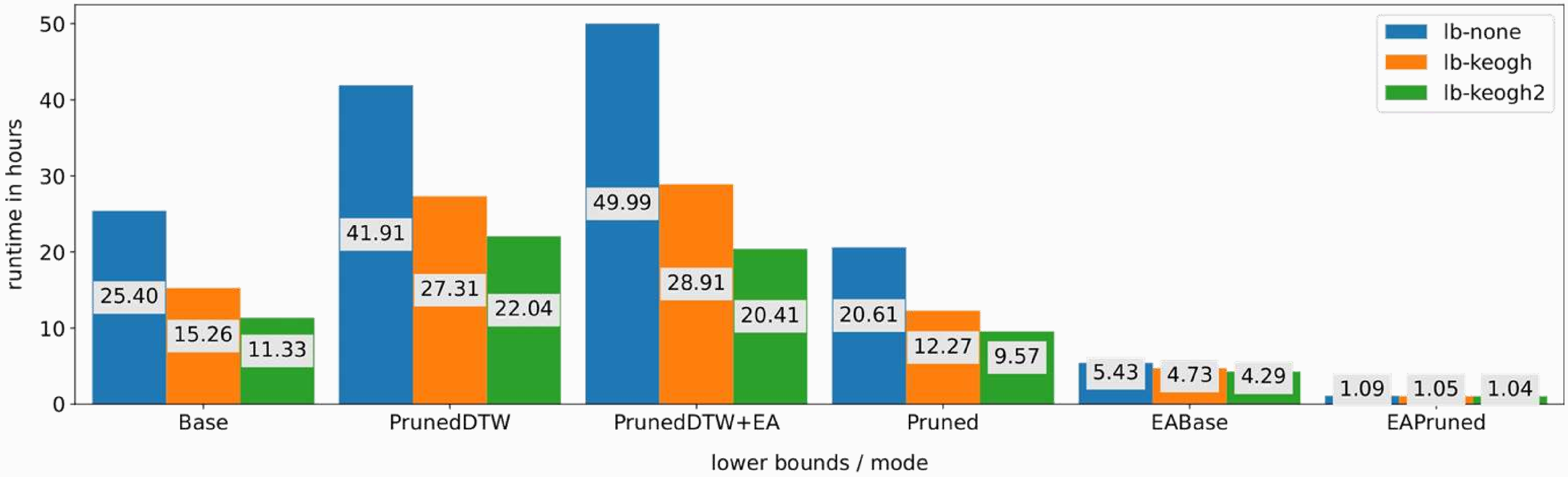


Our Method Early Abandoning AND Pruning

- Based on realization that when **all paths are pruned** the computation should be **abandoned**.
- Implements pruning more efficiently than previous approaches
- Unlike previous approaches, achieves abandoning without any significant computational overhead



Time in hours to process the UCR benchmark



Cost function tuning

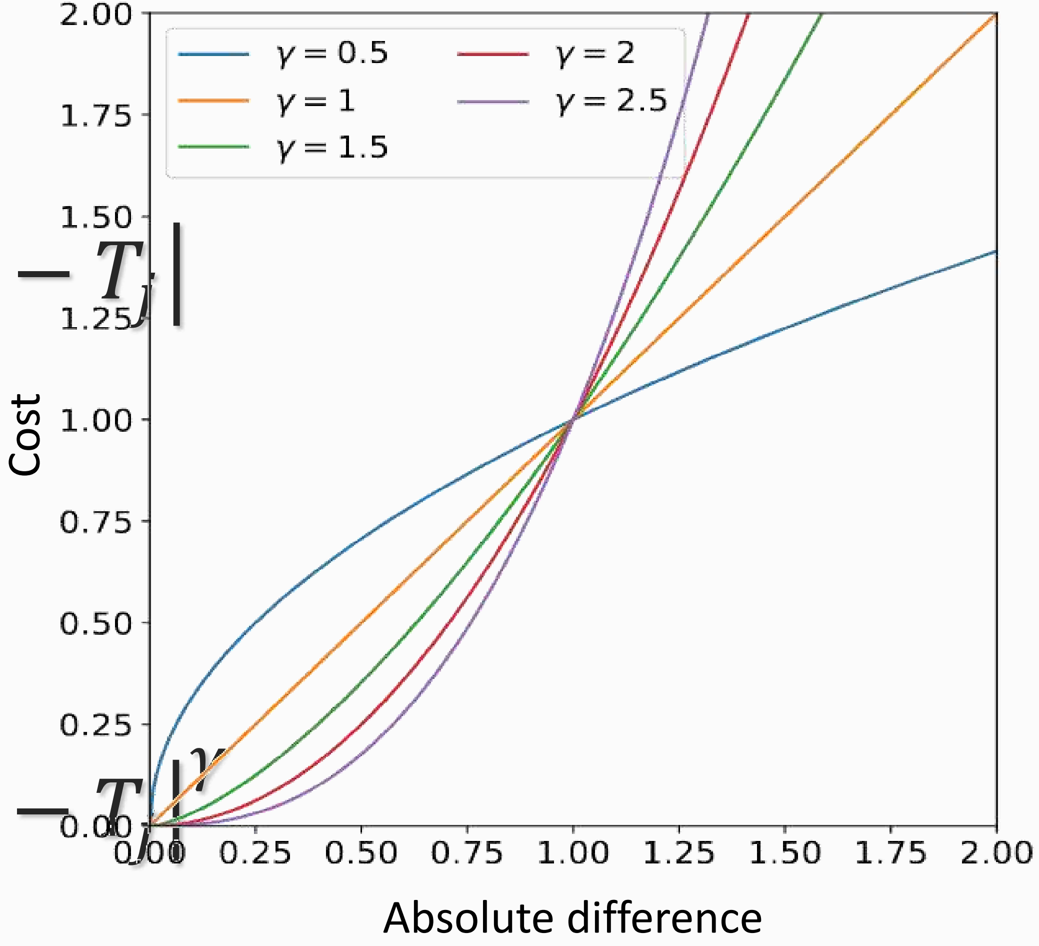
Cost function tuning

- The cost function determines the penalty for each alignment of two points
- The original cost function was

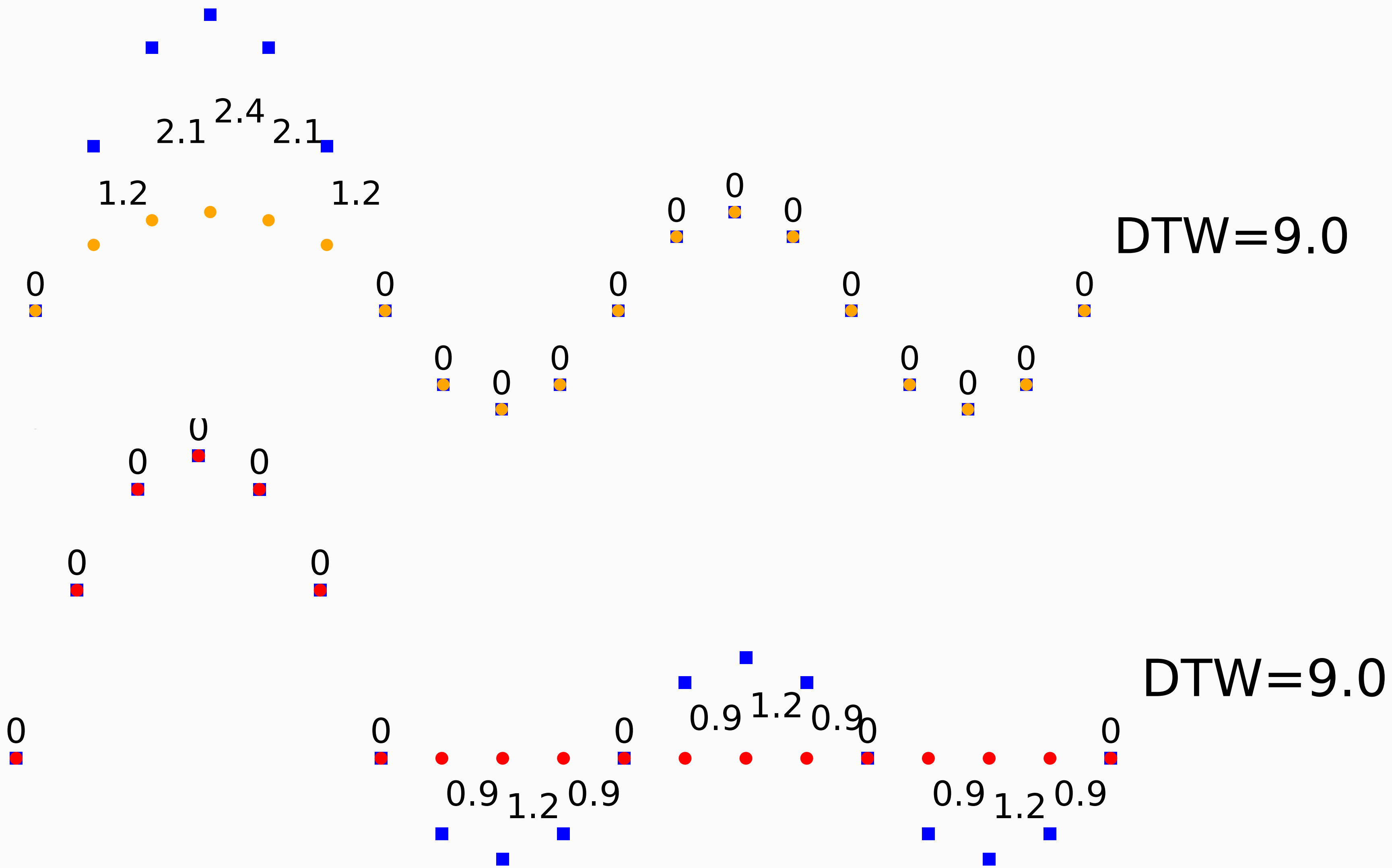
$$\lambda(S_i - T_j) = |S_i - T_j|$$

- $\lambda(S_i - T_j) = (S_i - T_j)^2$ also popular
- We explore

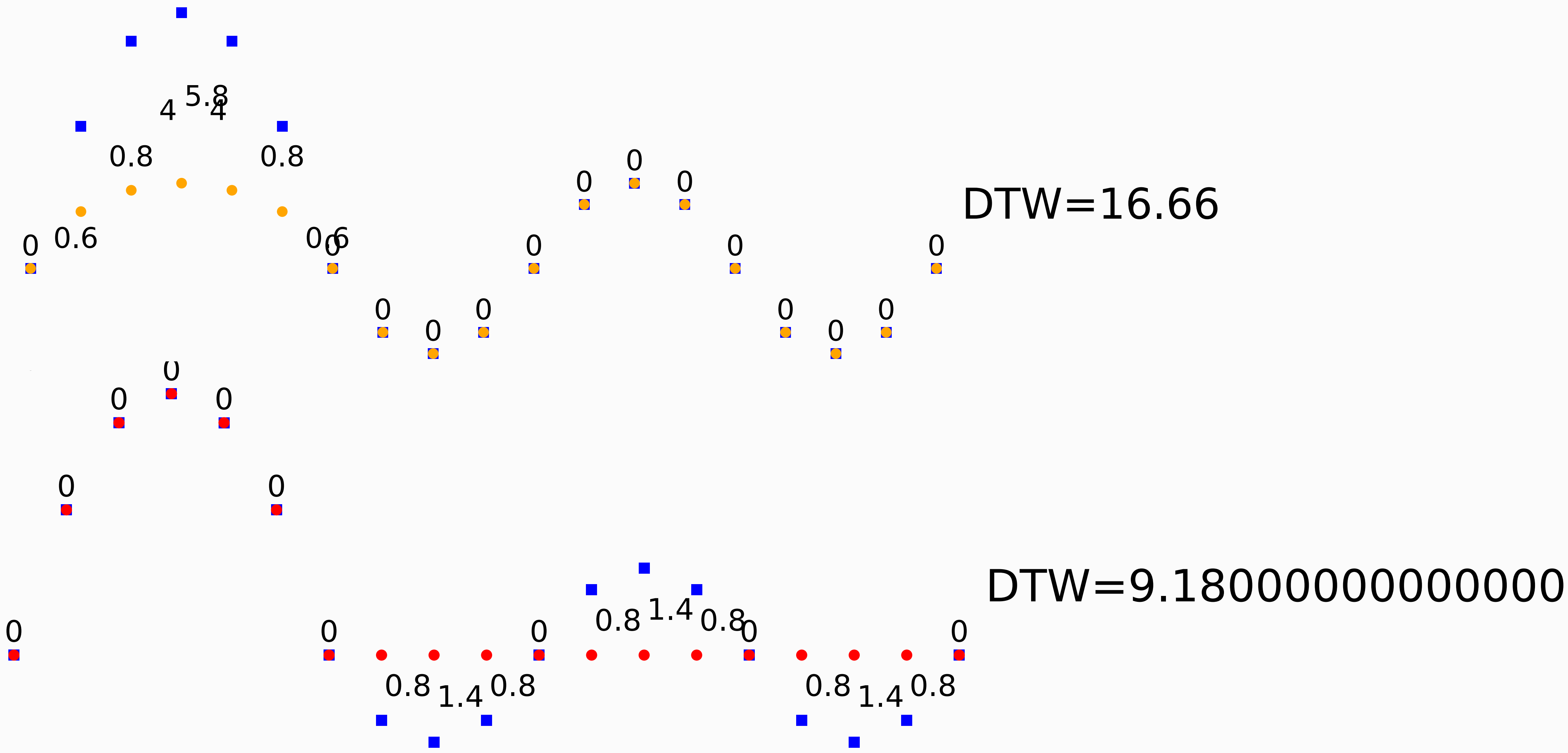
$$\lambda_\gamma(S_i - T_j) = |S_i - T_j|^\gamma$$



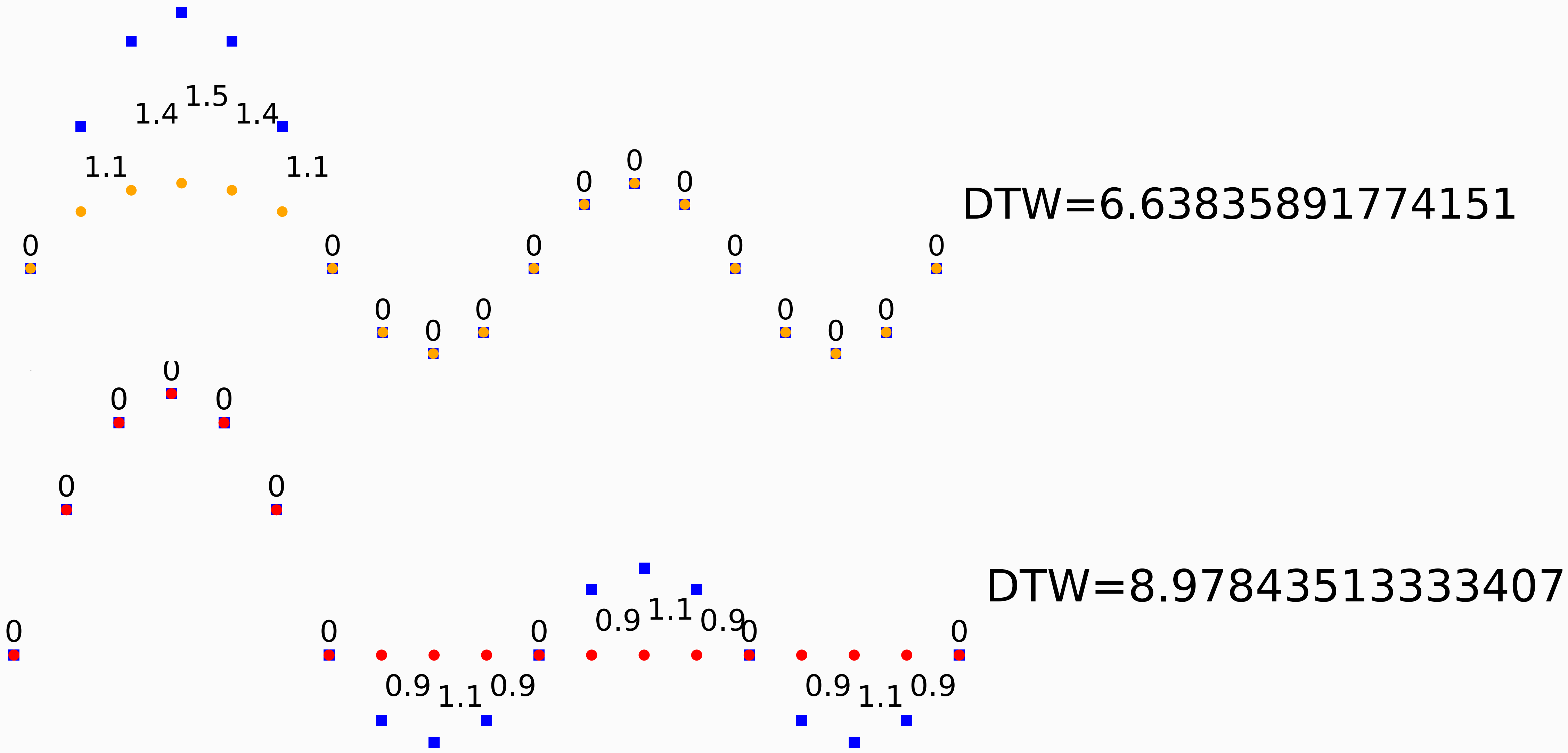
Distances using $\lambda(S_i - T_j) = |S_i - T_j|$



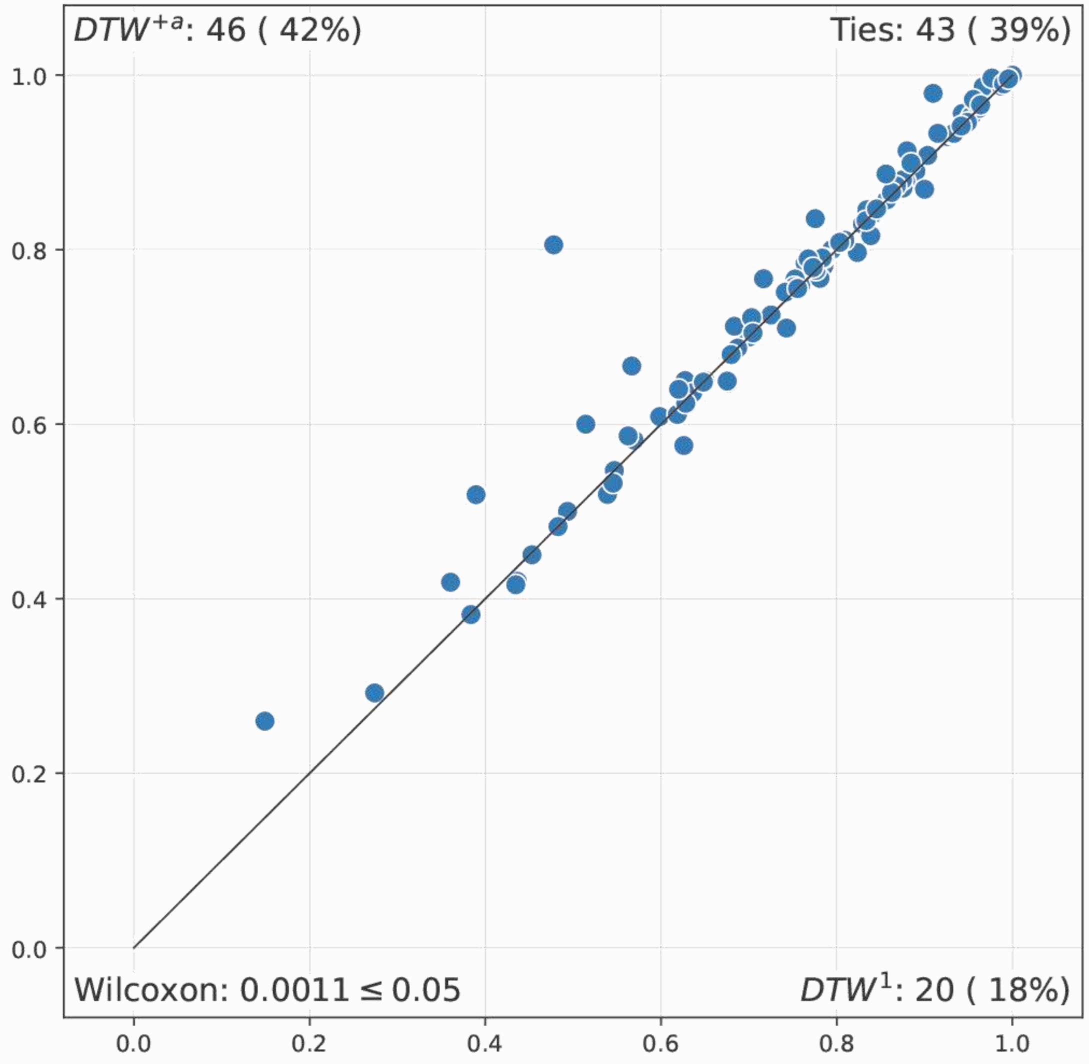
Distances using $\lambda(S_i - T_j) = (S_i - T_j)^2$

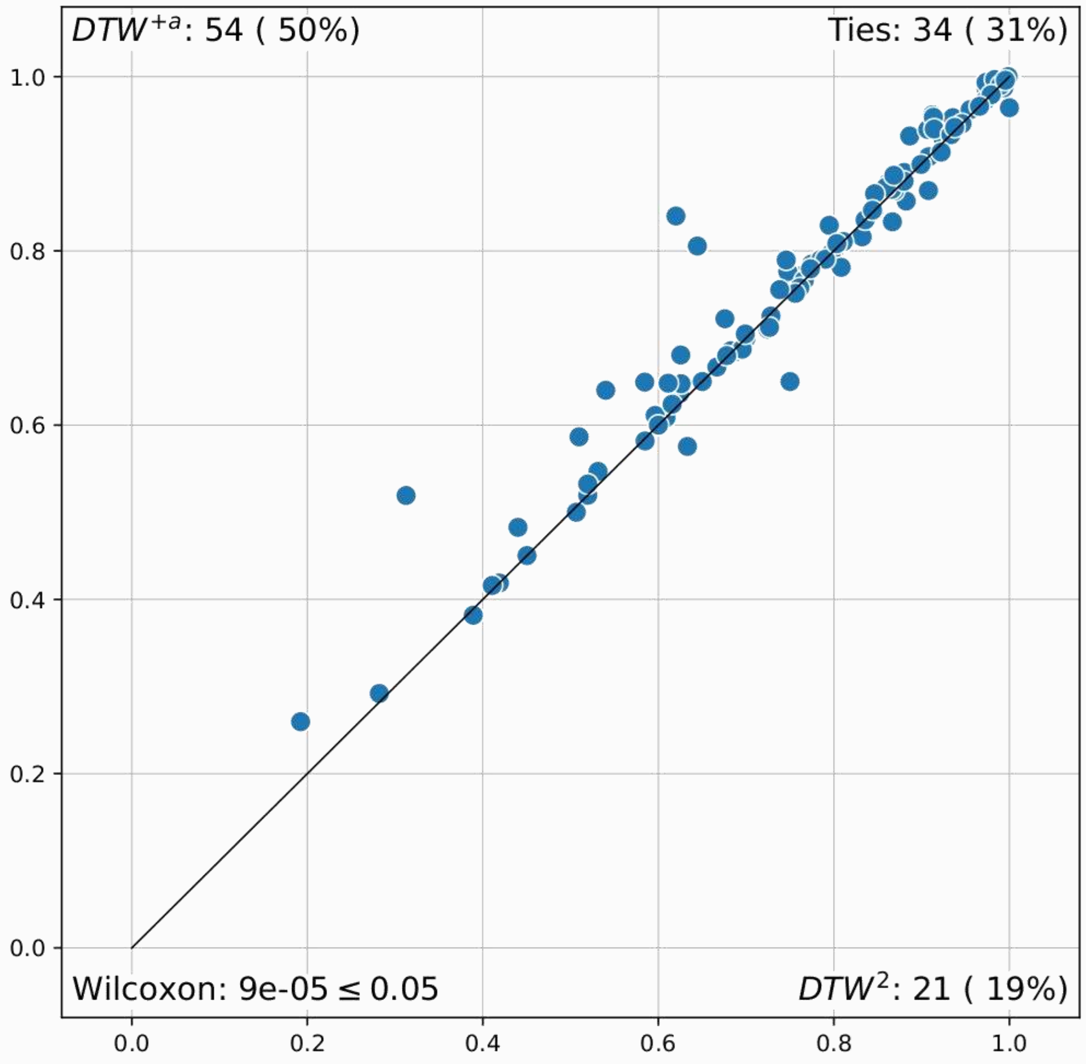


Distances using $\lambda_\gamma(S_i - T_j) = |S_i - T_j|^{0.5}$



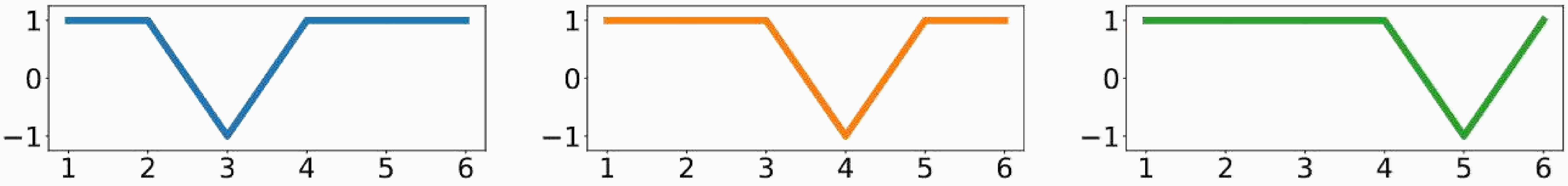
Cost tuning against fixed cost – UCR





Amerced Dynamic Time Warping

Amerced Dynamic Time Warping (ADTW)

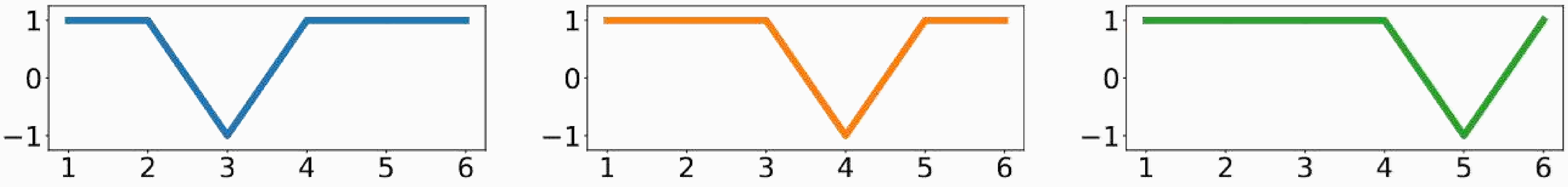


(a) $S = \{1, 1, -1, 1, 1, 1\}$ (b) $T = \{1, 1, 1, -1, 1, 1\}$ (c) $U = \{1, 1, 1, 1, -1, 1\}$

- Intuitively S closer to itself than T, and S closer to T than U
  dist(S,S) < dist(S,T) < dist(T,U)

- With DTW_∞ = DTW with no window, we have
  $DTW_\infty(S,S) = DTW_\infty(S,T) = DTW_\infty(S,U) = 0$

- With DTW, we have a "step function"
  - $w \geq 2$, DTW(S,S) = DTW(S,T) = DTW(S,U)=0
  - w = 1, DTW(S,S) = DTW(S,T) = 0 < DTW(S,U)=8
  - w = 0, DTW(S,S) = 0 < DTW(S,T) = DTW(S,U)=8



(a) $S = \{1, 1, -1, 1, 1, 1\}$ (b) $T = \{1, 1, 1, -1, 1, 1\}$ (c) $U = \{1, 1, 1, 1, -1, 1\}$

- New distance ADTW with additive penalty omega ω

  $\omega=0$, ADTW(S,S) **=** ADTW(S,T) **=** ADTW(S,U)

  $0<\omega<4$, ADTW(S,S) **<** ADTW(S,T) **<** ADTW(S,U)

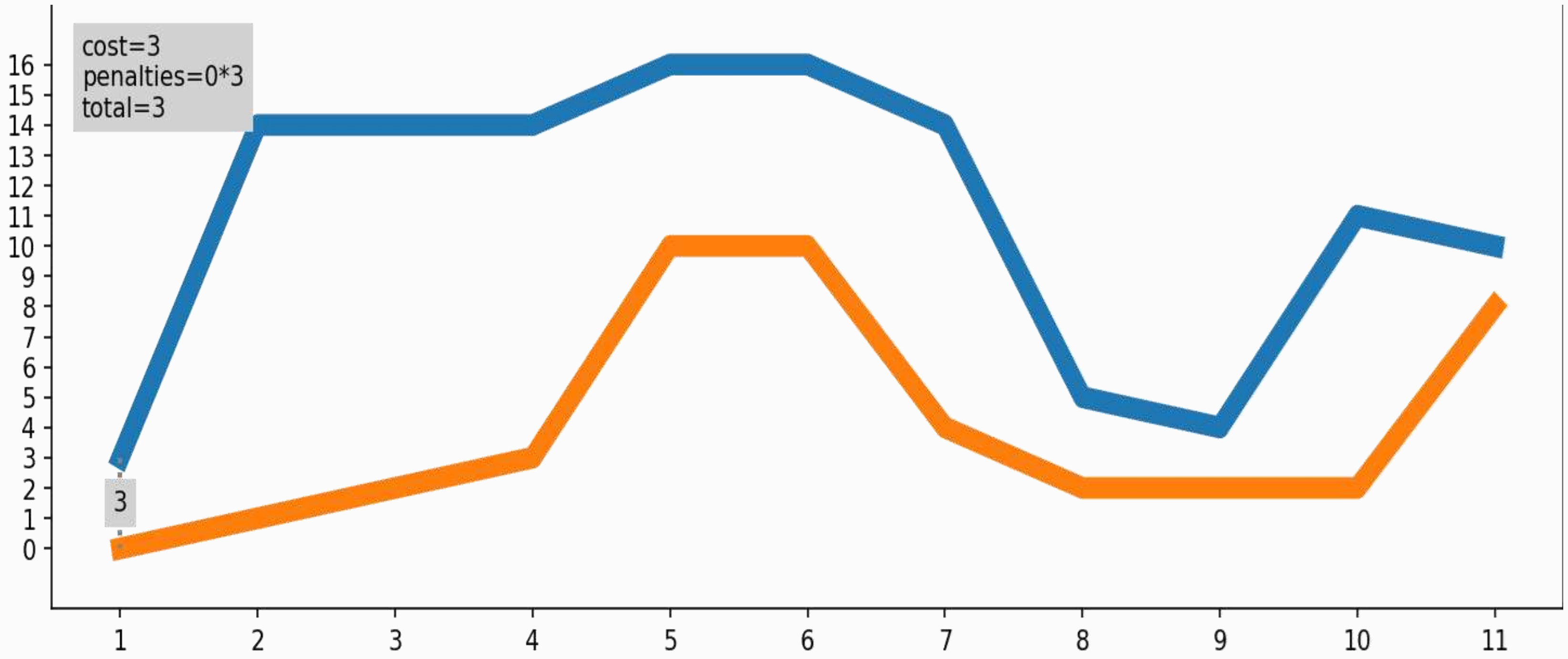
  $\omega\geq 4$, ADTW(S,S) **<** ADTW(S,T) **=** ADTW(S,U)

DTW

- $M_{0,0}=0$
- $M_{0,j} = M_{i,0} = \infty$
- $M_{i,j} = \gamma(S_i, T_j) + \min \begin{cases} M_{i-1,j-1} \\ M_{i-1,j} \\ M_{i,j-1} \end{cases}$

ADTW

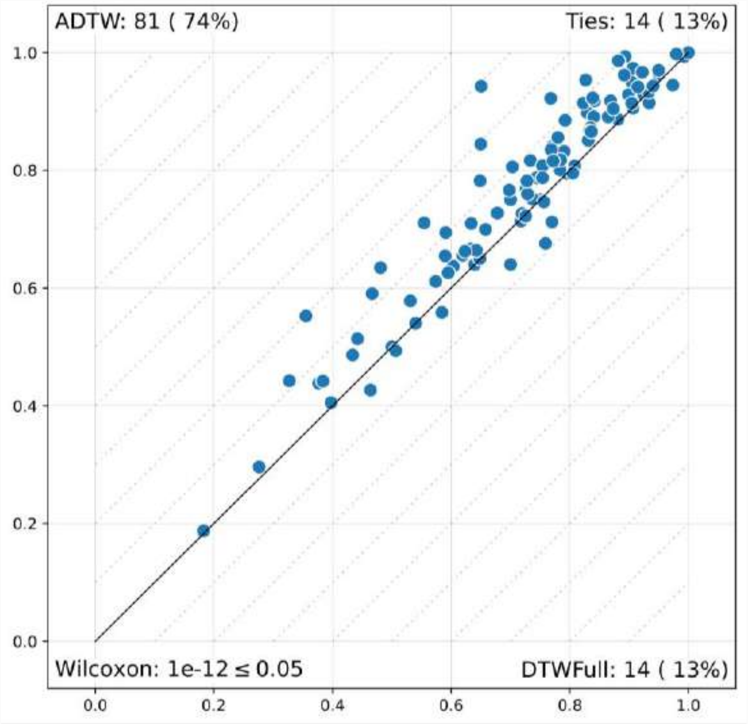
- $M_{0,0}=0$
- $M_{0,j} = M_{i,0} = \infty$
- $M_{i,j} = \gamma(S_i, T_j) + \min \begin{cases} M_{i-1,j-1} \\ M_{i-1,j} + \omega \\ M_{i,j-1} + \omega \end{cases}$



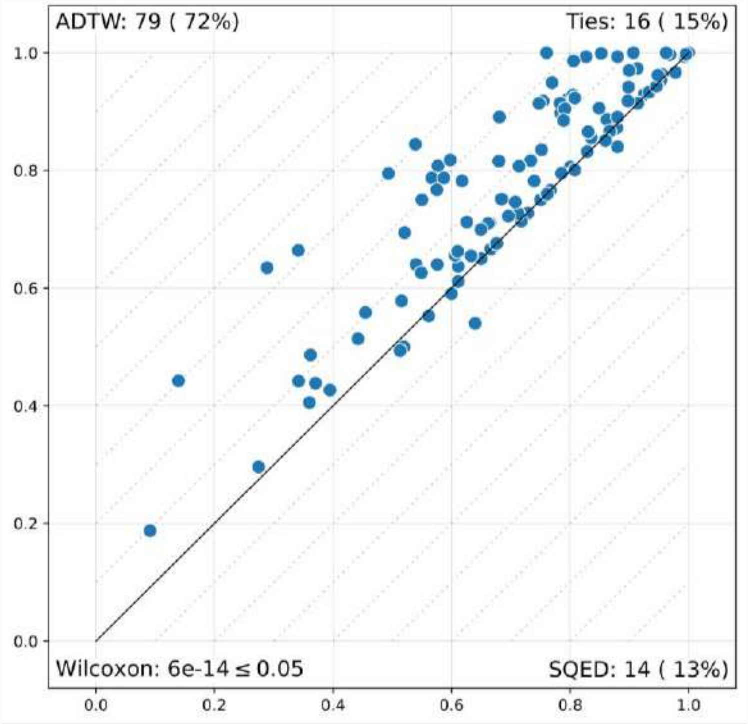
ADTW properties

- Symmetric: ADTW(S,T) = ADTW(T,S)
- ADTW(S,T) = ADTW(reverse(S), reverse(T))
- Monotonic with respect to ω
- $ADTW_0(S,T) = DTW_\infty(S,T)$
- $ADTW_\infty(S,T) = DTW_0(S,T)$
- So with $0 \leq \omega \leq \infty$ we have
  $DTW_\infty(S,T) \leq ADTW(S,T) \leq DTW_0(S,T)$

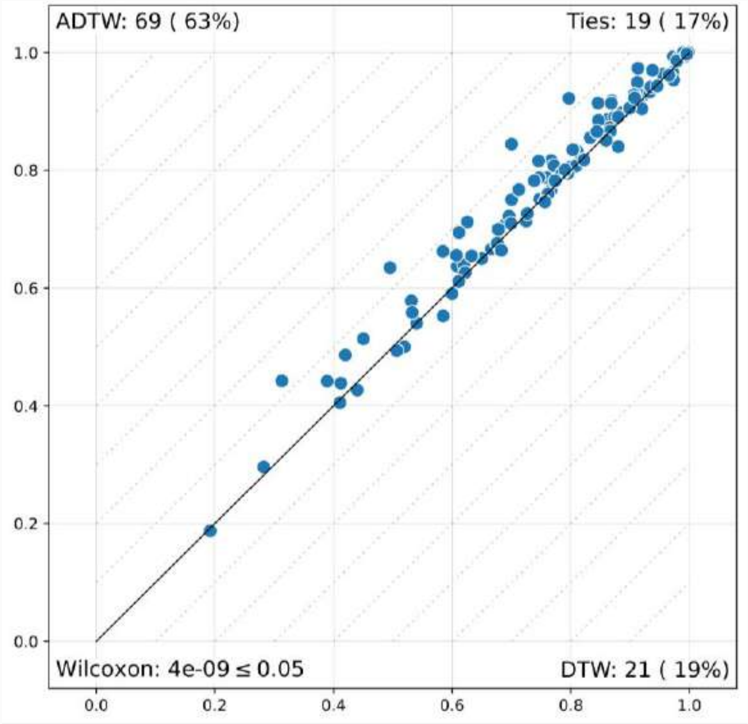
Comparison with DTW



ADTW vs DTW_{∞}



ADTW vs DTW_0



ADTW vs DTW_w

Concluding remarks

Research opportunities

- How to select meta parameters for tasks like clustering without objective performance measures
  - w for DTW
  - ω for ADTW
  - Cost function for all DTW variants
- Other classes of cost function
- Evaluate cost function tuning and ADTW in other tasks

Conclusions

- **Early Abandoning and Pruning** supports very fast exact calculation of DTW and its variants
- **Cost Function Tuning** can greatly improve DTW utility
- **ADTW** is an effective alternative to windowing for constraining warping in DTW
- We believe in reproducible research:
  - https://github.com/MonashTS/tempo

Matthieu Herrmann and Geoffrey I. Webb (2021) Early abandoning and pruning for elastic distances including dynamic time warping. *Data Mining and Knowledge Discovery*. 35(6): 2577–2601. doi: 10.1007/s10618-021-00782-4.

Matthieu Herrmann, Chang Wei Tan and Geoffrey I. Webb (in press) Parameterizing the cost function of Dynamic Time Warping with application to time series classification. *Data Mining and Knowledge Discovery*.

Matthieu Herrmann and Geoffrey I. Webb (2023) Amercing: An Intuitive and Effective Constraint for Dynamic Time Warping. *Pattern Recognition*. 137: article no 109333. doi: 10.1016/j.patcog.2023.109333.

Questions?

Matthieu Herrmann and Geoffrey I. Webb (2021) Early abandoning and pruning for elastic distances including dynamic time warping. *Data Mining and Knowledge Discovery*. 35(6): 2577-2601. doi: 10.1007/s10618-021-00782-4.

Matthieu Herrmann, Chang Wei Tan and Geoffrey I. Webb (in press) Parameterizing the cost function of Dynamic Time Warping with application to time series classification. *Data Mining and Knowledge Discovery*.

Matthieu Herrmann and Geoffrey I. Webb (2023) Amercing: An Intuitive and Effective Constraint for Dynamic Time Warping. *Pattern Recognition*. 137: article no 109333. doi: 10.1016/j.patcog.2023.109333.

http://i.giwebb.com/
https://github.com/MonashTS/tempo