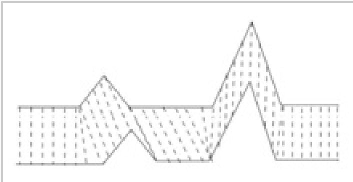
Dynamic time warping

From Wikipedia, the free encyclopedia

Not to be confused with the Time Warp mechanism for discrete event simulation, or the Time Warp Operating System that used this mechanism.



Dynamic Time Warping

In time series analysis, **dynamic time warping** (DTW) is an algorithm for measuring similarity between two temporal sequences which may vary in time or speed. For instance, similarities in walking patterns could be detected using DTW, even if one person was walking faster than the other, or if there were accelerations and decelerations during the course of an observation. DTW has been applied to temporal sequences of video, audio, and graphics data — indeed, any data which can be turned into a linear sequence can be analyzed with DTW. A well known application has been automatic speech recognition, to cope with different speaking speeds. Other applications include speaker recognition and online signature recognition. Also it is seen that it can be used in partial shape matching application.

In general, DTW is a method that calculates an optimal match between two given sequences (e.g. time series) with certain restrictions. The sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension. This sequence alignment method is often used in time series classification. Although DTW measures a distance-like quantity between two given sequences, it doesn't guarantee the triangle inequality to hold.

Contents [hide]

Implementation [edit]

This example illustrates the implementation of the dynamic time warping algorithm when the two sequences `s` and `t` are strings of discrete symbols. For two symbols `x` and `y`, `d(x, y)` is a distance between the symbols, e.g. `d(x, y)` = `| x - y |`

```
int DTWDistance(s: array [1..n], t: array [1..m]) {
    DTW := array [0..n, 0..m]

    for i := 1 to n
        DTW[i, 0] := infinity
    for i := 1 to m
        DTW[0, i] := infinity
    DTW[0, 0] := 0

    for i := 1 to n
        for j := 1 to m
            cost:= d(s[i], t[j])
            DTW[i, j] := cost + minimum(DTW[i-1, j  ],    // insertion
                                        DTW[i  , j-1],    // deletion
                                        DTW[i-1, j-1])    // match

    return DTW[n, m]
}
```

We sometimes want to add a locality constraint. That is, we require that if `s[i]` is matched with `t[j]`, then `| i - j |` is no larger than `w`, a window parameter.

We can easily modify the above algorithm to add a locality constraint (differences marked in ***bold italic***). However, the above given modification works only if `| n - m |` is no larger than `w`, i.e. the end point is within the window length from diagonal. In order to make the algorithm work, the window parameter `w` must be adapted so that `| n - m | ≤ w` (see the line marked with (*) in the code).

```
int DTWDistance(s: array [1..n], t: array [1..m], w: int) {
    DTW := array [0..n, 0..m]

    w := max(w, abs(n-m)) // adapt window size (*)

    for i := 0 to n
        for j:= 0 to m
            DTW[i, j] := infinity
    DTW[0, 0] := 0

    for i := 1 to n
        for j := max(1, i-w) to min(m, i+w)
            cost := d(s[i], t[j])
            DTW[i, j] := cost + minimum(DTW[i-1, j  ],    // insertion
                                        DTW[i, j-1],    // deletion
                                        DTW[i-1, j-1])    // match

    return DTW[n, m]
```

Fast computation [edit]

Computing the DTW requires $O(N^2)$ in general. Fast techniques for computing DTW include SparseDTW[1] and the FastDTW.[2] A common task, retrieval of similar time series, can be accelerated by using lower bounds such as LB_Keogh[3] or LB_Improved.[4] In a survey, Wang et al. reported slightly better results with the LB_Improved lower bound than the LB_Keogh bound, and found that other techniques were inefficient.[5]

Average sequence [edit]

Averaging for Dynamic Time Warping is the problem of finding an average sequence for a set of sequences. The average sequence is the sequence that minimizes the sum of the squares to the set of objects. NLAAF[6] is the exact method for two sequences. For more than two sequences, the problem is related to the one of the Multiple alignment and requires heuristics. DBA[7] is currently the reference method to average a set of sequences consistently with DTW. COMASA[8] efficiently randomizes the search for the average sequence, using DBA as a local optimization process.

Supervised Learning [edit]

Dynamic Time Warping is used as an elastic distance measure for the Nearest Neighbor Classifier, achieving state-of-the-art prediction quality.[9]

Alternative approach [edit]

An alternative technique for DTW is based on functional data analysis, in which the time series are regarded as discretizations of smooth (differentiable) functions of time and therefore continuous mathematics is applied.[10] Optimal nonlinear time warping functions are computed by minimizing a measure of distance of the set of functions to their warped average. Roughness penalty terms for the warping functions may be added, e.g., by constraining the size of their curvature. The resultant warping functions are smooth, which facilitates further processing. This approach has been successfully applied to analyze patterns and variability of speech movements.[11][12]

Open Source software [edit]

- The lbimproved C++ library implements Fast Nearest-Neighbor Retrieval algorithms under the GNU General Public License (GPL). It also provides a C++ implementation of Dynamic Time Warping as well as various lower bounds.

- The FastDTW library is a Java implementation of DTW and a FastDTW implementation that provides optimal or near-optimal alignments with an O(N) time and memory complexity, in contrast to the O(N^2) requirement for the standard DTW algorithm. FastDTW uses a multilevel approach that recursively projects a solution from a coarser resolution and refines the projected solution..
- FastDTW fork (Java) published to Maven Central
- The R package dtw implements most known variants of the DTW algorithm family, including a variety of recursion rules (also called step patterns), constraints, and substring matching.
- The mlpy Python library implements DTW.
- The pydtw C++/Python library implements the Manhattan and Euclidean flavoured DTW measures including the LB_Keogh lower bounds.
- What about the dtw python library?
- The cudadtw C++/CUDA library implements subsequence alignment of Euclidean-flavoured DTW and z-normalized Euclidean Distance similar to the popular UCR-Suite on CUDA-enabled accelerators.
- The JavaML machine learning library implements DTW.
- The ndtw C# library implements DTW with various options.
- Sketch-a-Char uses Greedy DTW (implemented in JavaScript) as part of LaTeX symbol classifier program.
- The MatchBox implements DTW to match Mel-Frequency Cepstral Coefficients of audio signals.
- Sequence averaging: a GPL Java implementation of DBA.[7]
- C/Python library implements DTW with some variations(distance functions, step patterns and windows)

References [edit]

1. ^ Al-Naymat, G., Chawla, S., & Taheri, J. (2012). SparseDTW: A Novel Approach to Speed up Dynamic Time Warping
2. ^ Stan Salvador & Philip Chan, FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space. KDD Workshop on Mining Temporal and Sequential Data, pp. 70-80, 2004
3. ^ Keogh, E.; Ratanamahatana, C. A. (2005). "Exact indexing of dynamic time warping". *Knowledge and Information Systems* **7** (3): 358–386. doi:10.1007/s10115-004-0154-9.
4. ^ Lemire, D. (2009). "Faster Retrieval with a Two-Pass Dynamic-Time-Warping Lower Bound". *Pattern Recognition* **42** (9): 2169–2180. doi:10.1016/j.patcog.2008.11.030.
5. ^ Wang, Xiaoyue et al. "Experimental comparison of representation methods and distance measures for time series data". *Data Mining and Knowledge Discovery* **2010**: 1–35.
6. ^ Gupta, L.; Molfese, D. L.; Tammana, R.; Simos, P. G. (1996). "Nonlinear alignment and averaging for estimating the evoked potential". *IEEE Transactions on Biomedical Engineering* **43** (4): 348–356. doi:10.1109/10.486255. PMID 8626184.
7. ^ *a* *b* Petitjean, F. O.; Ketterlin, A.; Gançarski, P. (2011). "A global averaging method for dynamic time warping, with applications to clustering". *Pattern Recognition* **44** (3): 678. doi:10.1016/j.patcog.2010.09.013.
8. ^ Petitjean, F. O.; Gançarski, P. (2012). "Summarizing a set of time series by averaging: From Steiner sequence to compact multiple alignment". *Theoretical Computer Science* **414**: 76. doi:10.1016/j.tcs.2011.09.029.
9. ^ Ding, Hui; Trajcevski, Goce; Scheuermann, Peter; Wang, Xiaoyue; Keogh, Eamonn (2008). "Querying and mining of time series data: experimental comparison of representations and distance measures". *Proc. VLDB Endow* **1** (2): 1542–1552. doi:10.14778/1454159.1454226.
10. ^ Lucero, J. C.; Munhall, K. G.; Gracco, V. G.; Ramsay, J. O. (1997). "On the Registration of Time and the Patterning of Speech Movements". *Journal of Speech, Language, and Hearing Research* **40**: 1111–1117.
11. ^ Howell, P.; Anderson, A.; Lucero, J. C. (2010). "Speech motor timing and fluency". In Maassen, B.; van Lieshout, P. *Speech Motor Control: New Developments in Basic and Applied Research*. Oxford University Press. pp. 215–225. ISBN 978-0199235797.
12. ^ Koenig, Laura L.; Lucero, Jorge C.; Perlman, Elizabeth (2008). "Speech production variability in fricatives of children and adults: Results of functional data analysis". *The Journal of the Acoustical Society of America* **124** (5): 3158–3170. doi:10.1121/1.2981639. ISSN 0001-4966. PMC 2677351. PMID 19045800.

Further reading [edit]

- Vintsyuk, T.K. (1968). "Speech discrimination by dynamic programming". *Kibernetika* **4**: 81–88.
- Sakoe, H.; Chiba (1978). "Dynamic programming algorithm optimization for spoken word recognition". *IEEE Transactions on Acoustics, Speech and Signal Processing* **26** (1): 43–49. doi:10.1109/tassp.1978.1163055.
- C. S. Myers and L. R. Rabiner.
  A comparative study of several dynamic time-warping algorithms for connected word recognition.
  The Bell System Technical Journal, 60(7):1389-1409, September 1981.
- L. R. Rabiner and B. Juang.
  Fundamentals of speech recognition.

Prentice-Hall, Inc., 1993 (Chapter 4)

- Muller, M., Information Retrieval for Music and Motion, Ch. 4 (available online at http://www.springer.com/cda/content/document/cda_downloaddocument/9783540740476-1.pdf?SGWID=0-0-45-452103-p173751818), Springer, 2007, ISBN 978-3-540-74047-6
- Rakthanmanon, Thanawin (September 2013). "Addressing Big Data Time Series: Mining Trillions of Time Series Subsequences Under Dynamic Time Warping". *ACM Transactions on Knowledge Discovery from Data* **7** (3): 10:1–10:31. doi:10.1145/2510000/2500489.

See also [edit]

- Levenshtein distance
- Elastic matching

Categories: Dynamic programming | Machine learning algorithms | Time series analysis