Classifying Human Movement Using Discrete Fréchet and DTW Distances

Hugo A. Akitaya
University of Massachusetts Lowell, USA

Joao Carlos Canto de Almeida
University of Massachusetts Lowell, USA

Lucas Fonseca
University of Nottingham, UK

Gabriel Shahrouzi
University of Massachusetts Lowell, USA

Individuals with motor disabilities often rely on assistive devices for support. However, conditions that affect upper limbs, particularly hand function, pose a challenge in the user interface design. In this context, inertial sensor-based movement classification rises as a method to interact with these devices by exploring residual motor capabilities [17]. State-of-the-art movement classification systems are based on machine learning algorithms. These, even the ones described as "explainable", are not transparent and user-friendly when it comes to movement analysis and understanding, which is vital for this demographic [9, 16, 10].

Addressing these concerns, we present a novel movement classifier that uses Fréchet and Dynamic Time Warping (DTW) distances based on inertial data. Our classifier aims to be customizable to individuals with various levels of motor impairment, following the work by Fonseca et al. (2019 and 2022) [9, 8]. In these studies, sensors were placed on the upper limbs of participants with tetraplegia. Inertial data was captured in pre-defined time windows and used to calculate features as input for machine learning classifiers. In [9], Fonseca et al. developed an adapted version of the k-nearest neighbour using PCA for feature dimension reduction. In [8], Fonseca et al. achieved superior results with an SVM algorithm. However, in both cases, unknown movements were incorrectly classified as one of the trained ones. Furthermore, incorrect classifications were difficult to assess, particularly by the participants.

The classifier proposed here introduces several enhancements over existing models. Our nearest centroid classifier is straightforward in identifying movements outside existing classes, providing an intuitive visualization of movements, and is independent of specific features. The integration of "time warping" distance functions introduces flexibility to variances in the speed of movements, no longer requiring a time window. Finally, our classifier facilitates a swifter classification (both Fréchet and DTW distances take quadratic time to compute). This is particularly beneficial in assistive technologies, where rapid actuation is crucial.

Contribution. We present a novel nearest centroid classifier for movements that utilizes Fréchet and Dynamic Time Warping (DTW) distances. We aim to enhance explainability, a critical yet often underemphasized aspect in assistive technology interfaces [10]. To define the centroids under DTW for each class, we use DTW Barycenter Averaging (DBA). For Fréchet distance, we define a DBA-like heuristic to produce a representative curve for each class. We then generate a simple data set to test our system.

Related work. The training data for each class is given by a set of m polygonal curves, each with at most n points. We wish to obtain a representative (centroid) curve for each class to act as the class centroid. A natural choice would be to compute an "average" curve. A natural definition of an average curve is the curve that minimizes the sum of distances from the average curve to the curves in the set (as in [4]). This is also called the *mean* curve. Unfortunately, Bulteau, Froese and Niedermeier [6] showed that, under DTW distance, computing the mean curve is NP- and W[1]-hard (when parameterized by m). Buchin, Driemel and Struijs [4] show the same under discrete (and continuous) Fréchet distance. They also describe approximation algorithms that run in time polynomial in m, n and the length ℓ of the median curve. Petitjean, Ketterlin and Gançarski [13] introduced a popular heuristic algorithm called DTW Barycenter Averaging (DBA) to approximate the mean curve under DTW. Another good candidate for centroid is the *center* curve, defined as the curve minimizing the maximum distance from itself to a curve in the set. Buchin et al. [3] showed that computing the center curve under discrete Fréchet distance is NP-hard, and they also give approximation algorithms whose runtime depend on the length ℓ of the output curve. In the context of clustering, a method similar to DBA was proposed by Buchin et al. [5] for the continuous Fréchet distance to improve a candidate for a center curve. Our method is basically the same but applied to the discrete Fréchet distance. We are unaware of literature investigating the computation of the center curve under DTW distance.

Considerable previous work has been done regarding k-clustering using Fréchet and DTW distances ([4, 3, 7, 11] to name a few). There are some applications that use DTW in nearest cluster classifiers [14, 12]. However, we are unaware of any application using Fréchet distance in the same setting.

1 Methodology

We utilize sets of training and test data labeled with their classes. Each data file represents a sequence of movements interspersed with rests (periods that the subject do not move). In our target application, the classification of the movements happen *online*, meaning that each point of the input is received streaming one at a time. It is desirable that a classifier outputs quickly after the end of a movement, or even before the movement ends, so that the appropriate action by the assistive device is performed. Note that we do not mean *online learning* where the training data is received in an online fashion. We take this fact into consideration to define three problems which must be addressed by our system:

- Segment the input data into rest and movement segments.
- Compute a centroid for each class using the movement segments of the training data.
- Implement an efficient nearest centroid classifier that uses the centroids computed in the previous step.

Our source code is available at:
`https://anonymous.4open.science/r/curves_classifier-DD4F`

Data. Data were collected using one Xsens (Henderson, USA) inertial measurement unit (IMU) from one able body subject (34 years old, right-handed). The IMU was placed on their left shoulder, and the sample frequency was set to 100Hz. The sensor is connected wirelessly to its base, which is interfaced with the computer by the Xsens MTManager software. The data is extracted from the software as unit quaternions, which are calculated based on the fusion of 3 degrees-of-freedom (3 DoFs) accelerometer, gyroscope and magnetometer data with the Xsens Kalman Filter for 3DoF orientation for Human Motion (XKF3hm). The subject was asked to perform three different shoulder movements: forward, upward and backward, always starting and finishing in a rest position. Figure 3 (in the appendix) illustrates the sensor placement and the movements directions. For each of the three required movements, a total of 20 repetitions were collected in the training scenario. For the test scenario, each sequence consisted of 10 repetitions.

Data Segmentation. We use two parameters: k, called *window size*, and r, called *threshold radius*. If the radius of the minimum enclosing ball of k subsequent data points is less than or equal to r, then these data points are defined to be *at rest*. A rest segment is a maximal contiguous subsequence of data points that are at rest. A maximal contiguous subsequence of data points not at rest is a movement segment.

We implement a straightforward greedy segmentation algorithm using CGAL [15] for minimum enclosing ball computations. For constant k, each computation takes $O(1)$ time.

Centroid Computation. For each class, we wish to compute a representative curve using the training data. We start by computing all pairwise distances between movement segments of a class. We noticed that, for both distance functions, the distance between a pair grew depending on the time gap between the execution of the movements, suggesting that the IMU sensor was slowly drifting during the data collection, or that the subject initial position was not maintained during the data collection. In [9, 8] this was not noticeable because the data pre-processing involved a differentiation step, which minimizes low frequencies in the signal, virtually disregarding the movement initial position. To attenuate this effect, we applied a translation for each movement segment to align their starting points at the origin. We offset each point on a movement segment to the radius of the minimum enclosing ball created with the last k points on the last seen rest segment.

For both DTW and Fréchet distances, we use a copy of one of the movement segments as a centroid and incrementally improve it using heuristics until convergence. We use the pairwise distance computations to choose the initial guess as the movement segment that minimizes the maximum distance between itself and the others in the same class. We then apply DBA and the Fréchet analog to improve the centroid for DTW and Fréchet distances.

Nearest Centroid Classifier. To perform the segmentation of the test data online, we use a simple state machine, similar to Fonseca et al. [9], but with only two states: rest and movement. We start at rest state. For each new data point, we check whether the radius of the minimum enclosing ball of the last k points is within r. We used the same parameters k and r as for the training data. As soon as this condition is not met, we enter movement state. If we are in movement state and the condition is met, we enter rest state.

As soon as we enter a movement state, we start computing DTW and Fréchet distances. We also apply a translation to bring the first data point to the origin. Recall that both distances have very similar Dynamic Programming (DP) implementations. We have a DP table for each centroid (two per class, one for each distance function, with a total of 6 tables). Let ℓ be the length of a centroid curve P. For each new point, we add a new row of length ℓ to the corresponding DP table. This can be done in $O(\ell)$ time. At each moment, we can compare the distances from the current movement segment to each centroid. We perform one classification for each distance function at the end of the movement segment.

2 Results

Segmentation. We manually found parameters $k = 20$ and $r = .005$ that correctly segment each training data file, obtaining 20 movement segments for each class. Figure 1 shows an example of the training data files and the computed segmentation.

Centroid computation. After applying the translation to each movement segment, the starting data point was set to the origin. Figure 2 show an example of the obtained centroids for the DBA and Fréchet analogue methods. We note that the centers using DBA showed "spike artifacts." This is a known effect of DBA [1]. The DBA and FCH computations performed a maximum of 35 and 22 iterations, respectively.

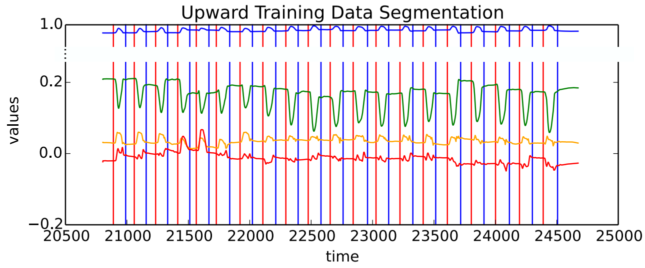

Figure 1 Training Data Segmentation. The vertical red lines represent when a movement start, and the blue vertical line represent the start of a rest. The blue, green, yellow and red curves represent the four coordinates of the quaternion data.

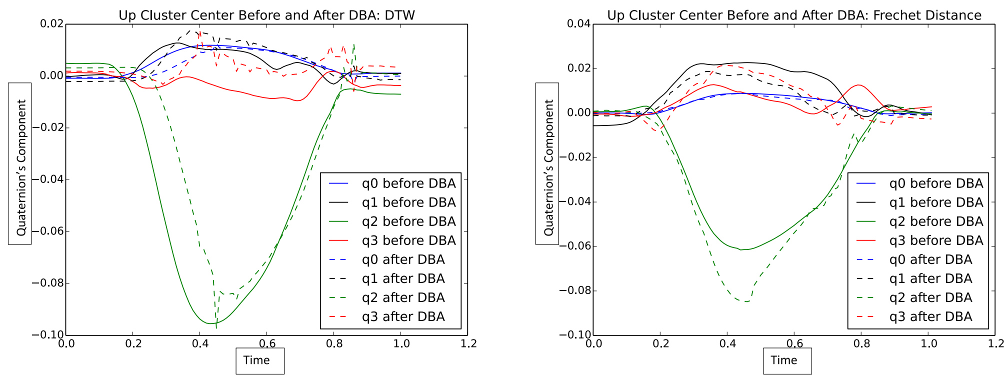

Figure 2 Comparison between the centroids before FHC is applied to them and after.

Classification. Both instances, the one using DTW and the one using Frèchet distance, achieved 100% accuracy. Tables 1 and 2 (in the appendix) show respectively the distances between the test and each centroid. In every case, the distance between the test curve and the centroid of the class it's labeled as is smaller than the distance between the test curve and the other centroid. Although 100% accuracy was achieved for both distance methods, we observed that the difference in the distances between the test curves and all three centroid was more prominent when we used DTW as the distance function.

3 Future Work

In our simple data set, both our classifiers obtained 100% accuracy. The data collected for this work was meant to serve as a baseline for a proof-of-concept and do not represent a real-life scenario. In future work, we will analyze the behavior of these classifiers in a more complex and realistic data set with more variation on the length of movements, and with data collected from subjects with motor disabilities. We also leave for future work to develop a method that allows the classification to happen as early as possible (as opposed to when a new rest starts) while maintaining the accuracy.

While DBA is widely used and has a fast convergence in practice for many applications, we don't know if that is the case for the Fréchet analogue. Indeed, our results suggest that the convergence for FCH is faster. An interesting open problem is to try to generalize the analysis by Brüning et al. [2] to the Fréchet distance.

References

1 Milutin Brankovic, Kevin Buchin, Koen Klaren, André Nusser, Aleksandr Popov, and Sampson Wong. (k, ℓ)-Medians Clustering of Trajectories Using Continuous Dynamic Time Warping. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, pages 99–110, 2020.

2 Frederik Brüning, Anne Driemel, Alperen Ergür, and Heiko Röglin. On the number of iterations of the dba algorithm. 2023.

3 Kevin Buchin, Anne Driemel, Joachim Gudmundsson, Michael Horton, Irina Kostitsyna, Maarten Löffler, and Martijn Struijs. Approximating (k, ℓ)-center clustering for curves. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2922–2938. SIAM, 2019.

4 Kevin Buchin, Anne Driemel, and Martijn Struijs. On the Hardness of Computing an Average Curve. In Susanne Albers, editor, *17th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2020)*, volume 162 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:19, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. URL: `https://drops.dagstuhl.de/opus/volltexte/2020/12266`, `doi:10.4230/LIPIcs.SWAT.2020.19`.

5 Kevin Buchin, Anne Driemel, Natasja van de L'Isle, and André Nusser. klcluster: Center-based clustering of trajectories. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 496–499, 2019.

6 Laurent Bulteau, Vincent Froese, and Rolf Niedermeier. Tight hardness results for consensus problems on circular strings and time series. *SIAM Journal on Discrete Mathematics*, 34(3):1854–1883, 2020.

7 Anne Driemel, Amer Krivošija, and Christian Sohler. Clustering time series under the fréchet distance. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 766–785. SIAM, 2016.

8 Lucas Fonseca, David Guiraud, Arthur Hiairrassary, Charles Fattal, and Christine Azevedo-Coste. A residual movement classification based user interface for control of assistive devices by persons with complete tetraplegia. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 30:569–578, 2022.

9 Lucas Fonseca, Wafa Tigra, Benjamin Navarro, David Guiraud, Charles Fattal, Antônio Bó, Emerson Fachin-Martins, Violaine Leynaert, Anthony Gélis, and Christine Azevedo-Coste. Assisted grasping in individuals with tetraplegia: Improving control through residual muscle contraction and movement. *Sensors*, 19(20):4532, 2019.

10 Jessica Gantenbein, Jan Dittli, Jan Thomas Meyer, Roger Gassert, and Olivier Lambercy. Intention detection strategies for robotic upper-limb orthoses: A scoping review considering usability, daily life application, and user evaluation. 16. MAG ID: 4212909647 S2ID: 3273379374f295db57ca5c3727ac5ebd31394c98. `doi:10.3389/fnbot.2022.815693`.

11 Abhinandan Nath and Erin Taylor. k-median clustering under discrete fréchet and hausdorff distances. In *36th International Symposium on Computational Geometry (SoCG 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

12 François Petitjean, Germain Forestier, Geoffrey I Webb, Ann E Nicholson, Yanping Chen, and Eamonn Keogh. Dynamic time warping averaging of time series allows faster and more accurate classification. In *2014 IEEE international conference on data mining*, pages 470–479. IEEE, 2014.

13 François Petitjean, Alain Ketterlin, and Pierre Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern recognition*, 44(3):678–693, 2011.

14 Saeid Soheily-Khah. Fast and accurate temporal data classification using nearest weighted centroid. In *2018 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO)*, pages 90–94. IEEE, 2018.

15 The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 5.6 edition, 2023. URL: https://doc.cgal.org/5.6/Manual/packages.html.

16 Richard B. Woodward, Sandra J. Shefelbine, and Ravi Vaidyanathan. Integrated grip switching and grasp control for prosthetic hands using fused inertial and mechanomyography measurement. pages 1–8. MAG ID: 1898006011. doi:10.1109/shbi.2015.7321689.

17 Sebastjan Šlajpah, Roman Kamnik, and Marko Munih. Kinematics based sensory fusion for wearable motion assessment in human walking. 116(2):131–144. MAG ID: 2096709246. doi:10.1016/j.cmpb.2013.11.012.

A Appendix

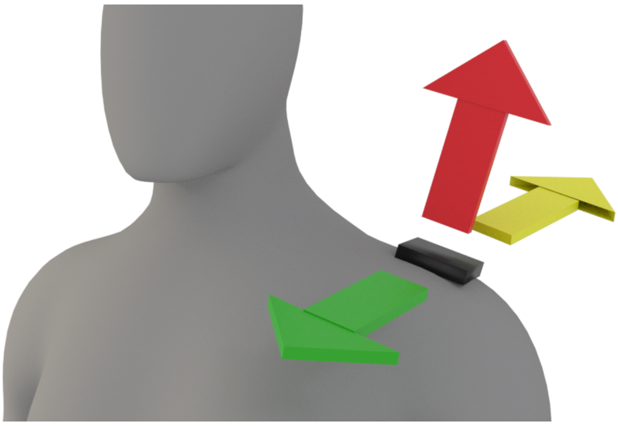

Figure 3 Sensor placement and shoulder movement directions.

	Forward (2.93564)	Up (2.81197)	Back (1.51583)
Up	5.08359	1.76751	4.36188
Forward	1.82604	4.76715	5.61986
Up	5.0193	1.92057	4.32075
Back	6.78597	4.03471	3.49427
Forward	2.12297	5.28419	6.12749
Back	6.09581	3.82701	2.76364
Forward	1.74797	4.85909	6.10829
Up	4.87847	2.31186	4.36646
Forward	1.59563	4.52471	5.70002
Forward	1.80807	4.96765	6.26574
Up	4.54756	1.84179	3.87357
Up	4.36569	2.14165	4.42327
Forward	2.51372	5.03914	5.93439
Up	4.51764	1.72154	3.77269
Forward	2.43589	5.072	6.08485
Forward	2.03448	4.66486	5.93898
Up	6.35574	2.93209	4.22205
Forward	1.90855	4.89245	5.90856
Back	6.83771	4.73186	3.42432
Back	6.96181	3.85901	2.53403
Back	5.90154	3.63211	2.5899
Forward	1.9541	5.34266	6.78012
Up	5.74237	2.30056	3.598
Back	6.82603	4.66437	3.62769
Up	5.8303	2.8356	5.0788
Forward	2.1149	5.08505	6.30179
Forward	3.27591	4.91543	6.29541
Forward	2.57902	5.44063	6.47033
Up	5.56865	2.19873	4.00986
Up	5.21602	2.17869	3.98014

Table 1 DTW: Distances between test curve and centroid

	Forward (0.112586)	Up (0.0931776)	Back (0.0971985)
Up	0.108214	0.0354834	0.0757051
Forward	0.0555512	0.0674876	0.0757051
Up	0.102874	0.0488166	0.0864274
Back	0.105634	0.0806102	0.0610774
Forward	0.0496826	0.0855315	0.0849349
Back	0.107653	0.0879584	0.0488565
Forward	0.0492905	0.0.0879081	0.086147
Up	0.103817	0.0565963	0.0850903
Forward	0.0516548	0.0730864	0.0850903
Forward	0.0481151	0.0848726	0.0871019
Up	0.1082	0.0436746	0.0787153
Up	0.107375	0.0556671	0.0787153
Forward	0.0515642	0.0752174	0.0787153
Up	0.103852	0.0382095	0.0787153
Forward	0.0488261	0.0847435	0.0788105
Forward	0.0525987	0.0773703	0.078932
Up	0.124464	0.0661044	0.0663126
Forward	0.0494689	0.0745088	0.0749716
Back	0.108041	0.0972703	0.0558157
Back	0.111256	0.080132	0.0487224
Back	0.104614	0.0855337	0.0495288
Forward	0.0476759	0.0710307	0.0944706
Up	0.108117	0.0529512	0.0641397
Back	0.107322	0.0892985	0.0600195
Up	0.105665	0.0520633	0.0841367
Forward	0.0499374	0.0776967	0.0841367
Forward	0.0525162	0.0833213	0.0846168
Forward	0.0546177	0.0853659	0.0868744
Up	0.108339	0.0592446	0.0801229
Up	0.108237	0.0559229	0.0746656

Table 2 Frechet Distance: Distances between test curve and centroid