Page intentionally left blank. Figure must be placed on a previous page in order to properly appear.

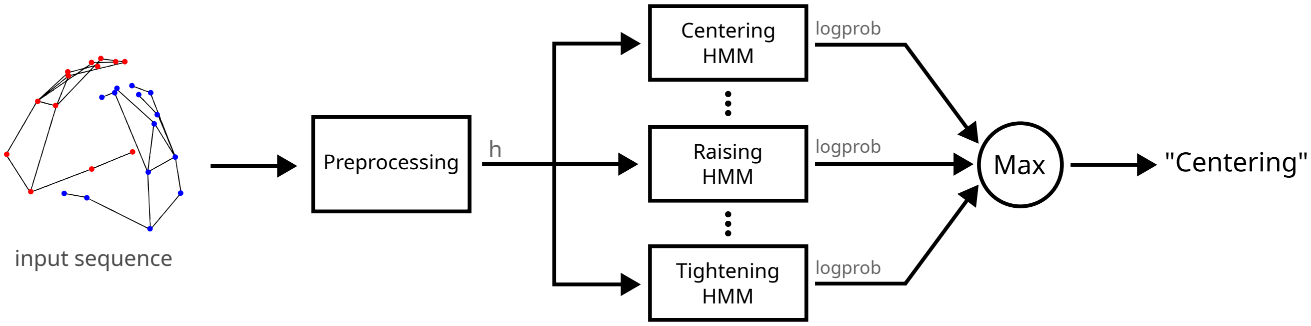

Fig. 1. Architecture of the proposed HMM-based method.

1. HMM-based classification

The proposed method utilizes an array of Hidden Markov Models (HMMs) with Gaussian mixture emissions.

HMMs are known to be particularly well suited for modelling and classifying signals that demonstrate intrinsic temporality, like human speech [1] and movement [2]. This makes them a promising choice for the present task of hand action recognition.

1.1. Architecture

The basic architecture of the proposed solution is illustrated in Fig. 1. Each input sequence is initially filtered, processed and flattened to a single vector (h), which is then fed to N distinct HMMs. Each HMM models one of the observed actions (classes) and, using a scoring function, evaluates the (log) probability of the given input sequence. The most likely match can then be extracted using a simple voting system based on the generated probabilities.

Each processing step is described in detail in the following paragraphs.

1.1.1. Preprocessing

Each of the provided examples is compromised of a sequence of frames, with each frame containing the coordinates of each marker. In order to train the HMMs, each sequence has to be converted to a single vector. Different ways of generating this representation were tested and compared, with the most efficient ultimately being interlacing the position data with estimated velocity data:

$$h_t = [x_1\ y_1\ z_1\ \Delta x_1\ \Delta y_1\ \Delta z_1\ x_2\ y_2\ z_2\ \Delta x_2\ \Delta y_2\ \Delta z_2\ \cdots] \quad (1)$$

$$h = [h_0\ h_1\ h_2\ \cdots] \quad (2)$$

The velocity of each marker is estimated as the difference between the current coordinates of the marker and those of the previous frame.

Another useful preprocessing step identified during testing was filtering the data by keeping only markers placed on the subjects fingertips (*THM, IDX, MID, RNG* and *PNK*), wrist (*IWR* and *OWR*) and center of the hand (*IHAND*). This improves training speed without affecting the models performance, as the positions of the other markers seem to provide mostly redundant information.

Finally, each sequence can be downsampled by only keeping every n frames. This improves training speed and, in some cases, also improves performance as the delta values become more intensified.

1.1.2. HMMs

One fully connected first order HMM is fitted to model the provided training examples of each separate class using the Expectation-Maximization (EM) algorithm [3]. The observations for each state are modeled using a Gaussian Mixture Model (GMM) with a full covariance matrix. The number of states of each HMM, as well as the number of states of each GMM are considered free variables.

The implementation of HMMs used was provided by the hmmlearn[1] python library, while hyperparameter optimization was performed based on leave-one-out cross validation (LOOCV) manually and automcatically using Optuna [4].

1.1.3. Results

The best observed result during LOOCV across all 50 examples had an overall accuracy of 90%. This translated to $83.\overline{3}\%$ accuracy on the provided test set (10/12).

References

[1] Rabiner, L. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 1989;77(2):257–286. doi:10.1109/5.18626.

[2] Papadopoulos, GT, Axenopoulos, A, Daras, P. Real-Time Skeleton-Tracking-Based Human Action Recognition Using Kinect Data. In: Gurrin, C, Hopfgartner, F, Hurst, W, Johansen, H, Lee, H, O'Connor, N, editors. MultiMedia Modeling. Cham: Springer International Publishing. ISBN 978-3-319-04114-8; 2014, p. 473–483. doi:10.1007/978-3-319-04114-8_40.

[3] Dempster, AP, Laird, NM, Rubin, DB. Maximum Likelihood from Incomplete Data Via the EM Algorithm. Journal of the Royal Statistical Society: Series B (Methodological) 1977;39(1):1–22. doi:10.1111/j.2517-6161.1977.tb01600.x.

[4] Akiba, T, Sano, S, Yanase, T, Ohta, T, Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. 2019. doi:10.48550/arXiv.1907.10902. arXiv:1907.10902.

[1] https://github.com/hmmlearn/hmmlearn