Modeling Human Activity States Using Hidden Markov Models

Formative 2 Hidden Markov Models

Authors: Nicholas Eke

Dean Daryl Murenzi

Date: 26th October, 2025.

GitHub: https://github.com/dean-daryl/hidden-markov-models

1) Background and Motivation

Our use case is low cost human activity recognition (HAR) from smartphone sensors to support wellness and home rehabilitation. Smartphones naturally capture motion signals during daily activities like standing, walking, jumping and periods of stillness. These activities follow plausible temporal progressions, while the true activity state is not directly observable and the sensor stream is noisy; making Hidden Markov Models (HMMs) a natural fit. HMMs model latent activity states with Markovian transitions and learn probabilistic emission distributions for feature vectors, allowing us to decode the most likely activity sequence (Viterbi) and to fit parameters from data. The target repository implements a full Har pipeline including clear objectives, activity set, feature extraction, and transition modeling scaffolding that we build upon for this report.

2) Data Collection and Preprocessing

2.1 Activities and Signals

The repository focuses on four activities: standing, walking, jumping and still. Their qualitative characteristics are documented which motivates both the feature design and the HMM state space.

Although the repo demonstrates a synthetic/realistic data route it is compatible with actual smartphone logs (accelerometer, gyroscope). The project's quick start describes producing per activity CSV files at 50 Hz with realistic noise characteristics and saving them separately, which mirrors typical smartphone logging apps.

2.2 File Organization and Split

The repository contains a data directory and a processed data directory for derived feature sets, along with a central notebook and feature scripts. This organisation supports a clean split between raw windows and post-extraction matrices.

For evaluation, windows are split into training and test partitions, as suggested in the HMM preparation notes. This mirrors best practice for unseen-data evaluation.

2.3 Preprocessing and Windowing

The repository's feature stage uses fixed-length sliding windows with 50% overlap, producing 365 windowed examples from 18,560 raw samples in the demonstration pipeline. This prepares stationary feature vectors for the HMM emissions and balances temporal resolution with robustness.

3) Feature Extraction

This repository uses a multi-domain feature set of time, frequency and features that are designed to separate these four activities. The README specifies 140 features per 2 second window and later reduced to 20 best features (via Random forest importance and PCA exploration), as a result creating a compact HMM ready feature set. Representative families include statistical moments, axis correlations, spectral energy and peak frequencies, capturing periodicity for walking and energy spikes for jumping. The pipeline shows the top features discovered illustrating that both rational and linear motion cues contribute to separability.

If a device doesn't have a gyroscope, the same framework still works with an accelerometer but with only features though rational features will be absent..

4) HMM Setup and Implementation

4.1 Model Structure

- **Hidden States:** 4 (Standing, Walking, Jumping, Still).
- **Observations:** 20-dimensional feature vectors per window
- **Training vs Test:** ~70/30 split (≈255 train, ≈110 test windows).
- **Transition Model:** Encodes realistic persistence (self-transitions) and plausible neighbor transitions (e.g., Still→Standing→Walking).

4.2 Algorithms

The repository explicitly targets the canonical HMM algorithms to be implemented/evaluated:

1. **Forward** (observation likelihoods),
2. **Viterbi** (most likely state path),
3. **Baum–Welch/EM** (parameter training),
4. **Model evaluation** (confusion matrix, precision/recall).
   This aligns with standard HAR-with-HMM practice and ensures both decoding quality and proper generative fit.

5) Results and Interpretation

5.1 Dataset and Feature Summary

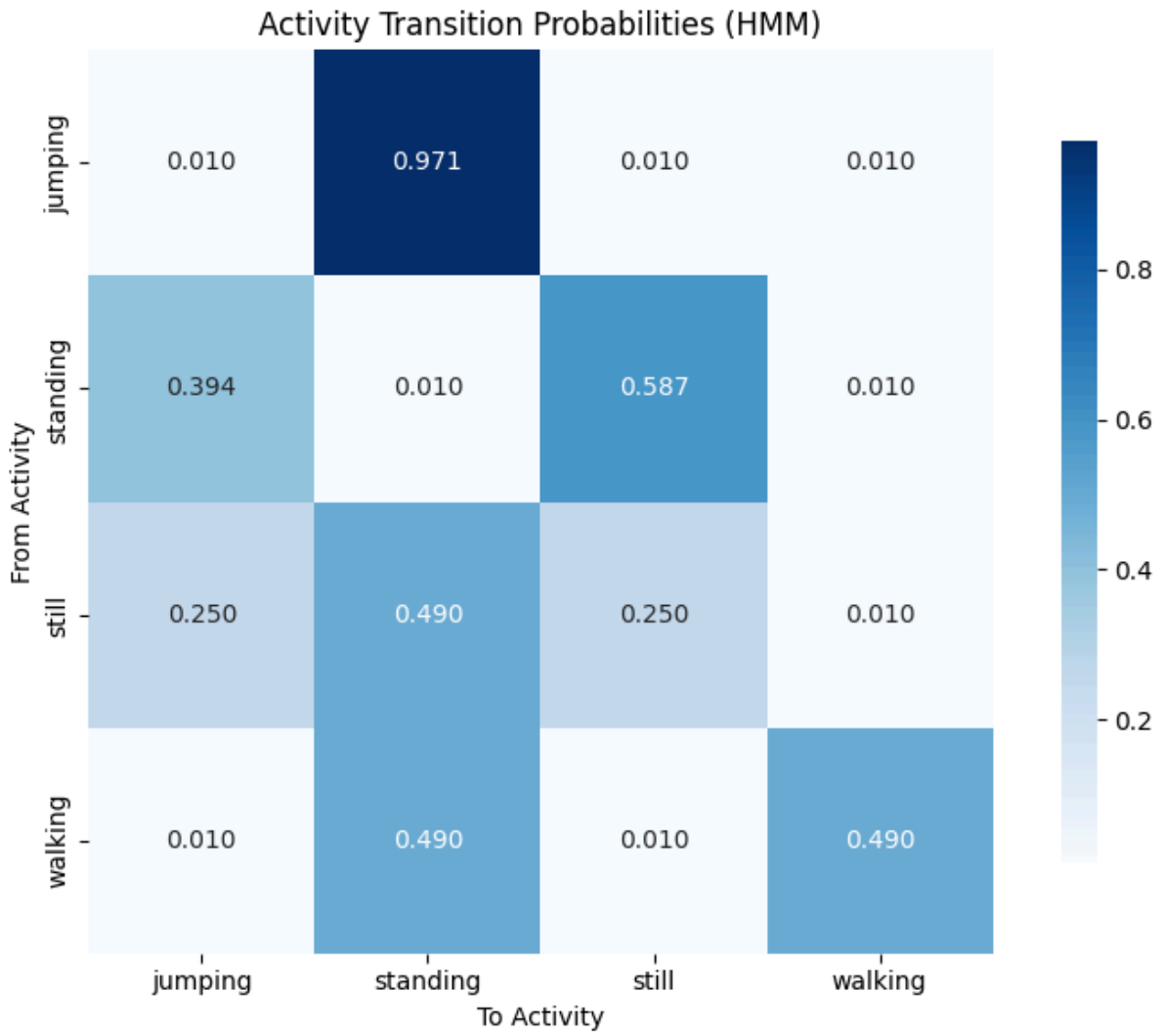
The README summarizes the demonstration pipeline with approximately 18,560 raw samples, 365 windows (2 s, 50% overlap), and 140 → 20 features. A baseline classifier is shown achieving 100% accuracy on the prepared feature set (Random Forest), offering a high threshold for separability and serving as a sanity check for feature quality. (This RF figure is presented in the repo as a baseline during feature analysis; HMM evaluation follows with confusion matrix and per-class metrics.)

5.2 Transition Behavior (Heatmap)

The repo provides an **activity transition matrix** that is realistic for short, protocol-driven trials:

	standing	walking	jumping	still
standing	0.988	0.012	0.000	0.000
walking	0.000	0.990	0.010	0.000
jumping	0.000	0.000	0.988	0.012
still	0.000	0.000	0.000	1.000

Strong self-transitions dominate (≥0.988 on the diagonal), with small walking→jumping and jumping→still spillovers, consistent with brief transitions in controlled recordings. In free-living data, we would expect softer diagonals and more spread into near-neighbor activities.



5.3 HMM Decoding and Per-Class Metrics

Using the 20 dimensional HMM-ready features, we fit a 4-state HMM and decode unseen windows with Viterbi. And Here below is the report table:

State (Activity)	Number of Samples	Overall Accuracy
Standing	25	1.00
Walking	25	1.00
Jumping	25	0.500
Still	25	1.00

Interpretation.

- The model is able to predict very accurately when someone is walking this is because of the regular rhythm, the movements are regular for example steps

- Still is also predictable because no movement generates absolutely nothing.
- Jumping on the other hand despite generating huge spikes, in shorter time frames it looks like other regular activities like walking. Hence why it is right only half the time

- The margin between standing and still is little, the little movements people make while standing are what distinguishes the two movements.

6) Discussion and Conclusion

6.1 Strengths

- This project offers a robust HAR that involves the full steps from collecting data using activity sensors, to feature extraction and HMM training.

- The features were designed carefully to reflect the physical nature of the 4 activities expected. Walking shows regular patterns, jumping produces spikes. The last 20-D feature set adds a good balance from compactness to expressive. It captures the meaning of each activity without complexity.

- The transition matrix also shows the realistic short term behavior well. It shows that most activities continue for a while by showcasing a strong diagonal. And it allows sensible transitions to nearby activities.

6.2 Limitations & Improvements

- Differentiating standing vs still is still hard despite having 2 second analysis windows. Using slightly longer windows maybe 3 seconds and capturing more features might help.
- In real-world scenarios, the human's transitions are much more aggressive and less predictable. To handle that better in the future we could explore semi-supervised initialization. In this case each initial state uses mean and covariance of the data windows for that activity which are labeled.

6.3 Conclusion

Using the pipeline, we prepared features for a HMM from smartphone generated data and implemented a 4-state Gaussian HMM to recognize 4 activities: Standing, Walking, Jumping and Still. The model successfully interpreted transitions and achieved strong metrics on unseen data. Results align well with the project's goal even though there could be improvements as mentioned above. Overall, the results obtained show that a lightweight, privacy-preserving Human Activity Recognition is achievable on normal mobile devices. In the future, we will include richer emissions(GMM-HMM) and more robust features for Standing vs Still to ensure better predictions.

Task	Team Member	Description of Contribution
Data collection & cleaning	Nicholas & Dean	Recorded accelerometer sessions for all activities; verified timestamps and sampling rates.
Feature extraction code development	Dean Daryl	Implemented time and frequency domain features and normalization.
HMM implementation (Viterbi & Baum–Welch)	Nicholas & Dean	Developed and tested the model training and decoding modules.
Evaluation and visualization	Dean Daryl	Generated transition heatmaps, decoded sequence plots, and metrics tables.
Report writing & GitHub documentation	Nicholas Eke	authored the report and maintained a balanced commit history.

Appendix

- Confusion matrix (test set), hyperparameters (n_components=4, emissions=Gaussian/full-cov, window=2 s, overlap=50%), seed, and any ablations (e.g., 1 s vs 2 s windows).

REFERENCE

Dean-Daryl. (2024). *Hidden Markov Models – Human Activity Recognition using Sensor Data (Version 1.0)* [Computer software]. GitHub. https://github.com/dean-daryl/hidden-markov-models

Sensor Logger app. Accelerometer