Hidden Markov Model for Human Activity Recognition

Github repository link: https://github.com/excelasaph/hidden_markov_model.git

Executive Summary

This project implements a **Hidden Markov Model (HMM)** for human activity recognition using smartphone accelerometer and gyroscope data. The model distinguishes between four activities: **Jumping, Standing, Still, and Walking**.

Key Results:

- Training Accuracy: 84.90%
- Test Accuracy (unseen data): 65.79%
- Best Performing Activity: Walking (81.58% sensitivity in test)

The project demonstrates the feasibility of HMMs for modeling temporal patterns in human motion and detecting activity states from sensor signals.

1. Background and Motivation

Human Activity Recognition (HAR) enables applications in healthcare, fitness tracking, and smart environments. Accurate recognition of activities such as walking, standing, and jumping can assist in monitoring patient mobility, detecting falls, or optimizing exercise routines.

This project explores **Hidden Markov Models** to infer hidden activity states from noisy sensor data and evaluates their ability to generalize to unseen data.

2. Methodology

2.1 Data Collection

- **Devices Used:**
  - Lesly Ndizeye: Google pixel 6(100 Hz sampling)
  - Excel Asaph: iPhone 12 (100 Hz sampling)
- **Sensors Recorded:** Accelerometer and Gyroscope (x, y, z)
- **Activities:** Jumping, Standing, Still, Walking
- **Duration per Activity:** 5–10 seconds per session
- **Total Data:** 258 recordings (training + test)

Data from different sampling rates was harmonized by resampling to 100 Hz. Each activity session was saved as a `.csv` file with timestamps and raw sensor readings.

2.2 Feature Extraction

23 features per analysis window were extracted from both time and frequency domains.

Time-Domain Features:

- Mean, Standard Deviation (STD), RMS, Signal Magnitude Area (SMA)
- Correlations and Magnitude

Frequency-Domain Features:

- Dominant Frequency
- Spectral Energy
- Spectral Entropy

All features were **normalized using Z-score standardization** to ensure uniform scaling across users and sessions.

Windowing Details:

- Window size: 100 samples (1 second at 100 Hz)
- Overlap: 50%

Training Samples: 1,676 feature vectors
Test Samples: 76 feature vectors

2.3 HMM Architecture

- **Hidden States:** Jumping, Standing, Still, Walking (4 states)
- **Emission Probabilities:** Gaussian distributions
- **Training Algorithm:** Baum-Welch
- **Decoding Algorithm:** Viterbi
- **Feature Vector:** 23 features per window

The HMM was implemented using Python and `hmmlearn`, with custom checks for convergence and decoding accuracy.

3. Results and Evaluation

3.1 Training Performance

Activity	Samples	Sensitivity	Specificity
Jumping	406	50.99%	99.92%
Standing	402	95.52%	97.25%
Still	360	97.50%	100.00%
Walking	508	94.69%	81.42%

Overall Training Accuracy: 84.90%

3.2 Test Performance (Unseen Data)

Activity	Samples	Sensitivity	Specificity
Jumping	38	50.00%	100.00%
Walking	38	81.58%	50.00%

Overall Test Accuracy: 65.79%

The model generalized reasonably to unseen test data, performing best on **Walking**. Jumping shows moderate sensitivity, indicating it remains more difficult to classify reliably.

4. Discussion

- **Stationary Activities:** Standing and Still are classified accurately with high sensitivity.
- **Dynamic Activities:** Walking shows strong performance in both training and tests.
- **Jumping:** Moderate sensitivity indicates variability in motion patterns may affect detection.
- **Generalization:** The HMM demonstrates reasonable adaptability to unseen data despite lower accuracy for some activities.

Improvements Suggested:

1. Collect more diverse and longer training sessions.
2. Optimize feature selection for better discrimination.
3. Experiment with different window sizes and overlaps.
4. Include additional activity types to expand model capability.

5. Collaboration

Task	Lesly Ndizeye	Excel Asaph
Data Collection	Yes	Yes
Feature Engineering		Yes
HMM Training & Implementation	Yes	Yes
Model Evaluation	Yes	Yes
Visualizations & Reporting	Yes	

GitHub Repository: https://github.com/excelasaph/hidden_markov_model.git

6. Conclusion

The Hidden Markov Model successfully classified human activities using smartphone sensor data. Despite challenges with Jumping activity, the HMM achieved:

- **84.90% training accuracy** across four activities
- **65.79% accuracy** on unseen test data

The implementation demonstrates a working HMM pipeline including feature extraction, training, decoding, and evaluation, ready for real-world HAR applications.

7. References

1. Rabiner, L. R. (1989). *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.* Proceedings of the IEEE.

2. Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2013). *A Public Domain Dataset for Human Activity Recognition Using Smartphones.* ESANN.

3. hmmlearn Documentation (2024). https://hmmlearn.readthedocs.io

4. Sensor Logger App (2024). Mobile sensor data collection tool.