Modeling Human Activity States Using Hidden Markov Models

Group Information

Group Members: Kanisa Rebecca Majok Thiak, Rene Pierre Ntabana
Sampling Rate: 100 Hz

This project explores how Hidden Markov Models (HMMs) can be used to recognize human activities such as walking, standing, jumping, and remaining still using smartphone sensors. The goal is to improve motion tracking for healthcare and fitness applications by modeling sequential sensor data.

1. Background and Motivation

Human activity recognition plays a vital role in areas such as healthcare, fitness tracking, and smart environments. Our approach leverages Hidden Markov Models to handle the temporal dependencies and variability of human motion, enabling real-time identification of physical activities for rehabilitation and monitoring.

---

2. Data Collection and Preprocessing

2.1 Data Collection Process

Data collection was performed using smartphone sensors through a mobile logging application(Sensor Logger). We recorded accelerometer and gyroscope data for four distinct activities over multiple sessions to ensure robust model training.

Activities Performed:

- **Jumping**: Recorded 11 sessions, each lasting approximately 5–10 seconds of continuous jumping motion
- **Standing**: Recorded 13 sessions, maintaining phone steady at waist level
- **Still**: Recorded 11 sessions with the phone placed stationary on a flat surface
- **Walking**: Recorded 15 sessions, maintaining a consistent walking pace

Total Recordings: 50 sessions (50 samples as required) split into 2, 25 each.

Data Structure:

Each recording session generated two CSV files:

- `Accelerometer.csv` - Contains x, y, z acceleration values
- `Gyroscope.csv` - Contains x, y, z angular velocity values

2.2 Preprocessing Pipeline

Our preprocessing involved merging accelerometer and gyroscope data into a unified dataset:

1. **Data Merging** (`merge_datasets.ipynb`, Cells 1-4):
   - Loaded all CSV files from activity-specific directories
   - Tagged each row with the `Activity` label and the `Sensor` type
   - Added `SourceFile` metadata for tracking
   - Combined all data into `merged_all_sensors.csv` (57,870 total rows)
2. **Activity-Specific Files**:
   - `merged_Jumping.csv` (12,207 rows)
   - `merged_Standing.csv` (15,164 rows)
   - `merged_Still.csv` (12,915 rows)
   - `merged_Walking.csv` (17,584 rows)
3. **Data Quality Checks**:
   - No missing values detected
   - Time range: 0.09 to 12.51 seconds per recording
   - Consistent sensor format across all recordings

2.3 Feature Extraction

We implemented a comprehensive `FeatureExtractor` class that computes both time-domain and frequency-domain features using a sliding window approach (window size: 2 seconds, 50% overlap):

Time-Domain Features:

For each axis (x, y, z):

- Mean, standard deviation, variance
- Maximum, minimum, range (peak-to-peak)
- RMS (Root Mean Square)
- MAD (Mean Absolute Deviation)
- Skewness and Kurtosis
- Energy (sum of squares)

Multi-axis features:

- SMA (Signal Magnitude Area)
- Correlation coefficients (xy, xz, yz)

Frequency-Domain Features:

For each axis using FFT (sampling rate: 100 Hz):

- Dominant frequency
- Spectral energy
- Spectral centroid
- Spectral entropy
- Band-specific energy:
    - Low frequency (0.1-2 Hz)

- Medium frequency (2-5 Hz)
- High frequency (5-10 Hz)

Total Features: 59 features + sensor type indicator (60-dimensional feature vector)

Result: 411 feature windows extracted from the dataset.

3. HMM Setup and Implementation

3.1 Model Architecture

We defined a 4-state HMM where:

- **Hidden States (Z)**: The four activities {Jumping, Standing, Still, Walking}
- **Observations (X)**: 60-dimensional feature vectors
- **Number of States (n_states)**: 4 (one per activity)

3.2 Implementation Approach

We implemented a flexible `HMMActivityRecognizer` class that:

- Falls back to custom GaussianHMM when `hmmlearn` is unavailable
- Uses Gaussian emission distributions for continuous observations
- Implements the Baum-Welch algorithm for parameter estimation

Model Components:

Transition Probabilities (A):
Learned probabilities of transitioning between activities:

From / To	Jumping	Standing	Still	Walking
Jumping	0.667	0.000	0.053	0.281

Standing	0.000	0.763	0.237	0.000
Still	0.089	0.143	0.768	0.000
Walking	0.109	0.000	0.007	0.883

Initial State Probabilities (π): Uniform initialization across all states

Emission Probabilities (B):

- Gaussian distributions for each state
- Diagonal covariance matrix for tractability
- Mean vectors learned from training data

3.3 Training Process

Data was split into:

- **Training Set**: 328 windows (80% of data)
- **Test Set**: 83 windows (20% of data)

Training sequence creation:

- Sequences of length 10 windows for HMM training
- Total training sequences: 32

Training parameters:

- Algorithm: Baum-Welch (Expectation-Maximization)
- Iterations: 50
- Convergence reached at iteration 7

- Final log-likelihood: 15,430.90

3.4 Decoding Algorithm

Viterbi Algorithm implementation:

- Finds the most likely sequence of hidden states (activities) given observations
- Used for both training and inference on test data

4. Results and Interpretation

4.1 Model Performance

Overall Test Accuracy: 100%

Activity Distribution:

- Walking: 126 windows (30.7%)
- Standing: 107 windows (26.0%)
- Still: 92 windows (22.4%)
- Jumping: 86 windows (20.9%)

4.2 Evaluation Metrics

Activity	Samples	Sensitivity	Specificity	Accuracy
Jumping	96	0.0	1.0	76.0%

Standing	102	0.0	1.0	74.4%
Still	79	0.0	1.0	80.2%
Walking	122	1.0	0.0	30.6%

4.3 Cross-Validation Results

3-fold cross-validation performance:

- **Fold 1 Accuracy:** 51.1%
- **Fold 2 Accuracy:** 47.0%
- **Fold 3 Accuracy:** 42.0%
- **Mean CV Accuracy:** 46.7% ± 4.6%

Confusion Matrix (Fold 1):

Actual / Predicted	Jumping	Standing	Still	Walking
Jumping	36	0	0	0
Standing	0	34	0	0
Still	0	24	0	0
Walking	3	40	0	0

4.4 Feature Importance Analysis

Top 5 most discriminative features (Random Forest analysis):

1. x_low_freq_energy (6.92%)
2. y_variance (5.45%)

3. x_standard_deviation (5.32%)
4. x_medium_freq_energy (4.69%)
5. y_mean_absolute_deviation (4.54%)

4.5 Visualizations

Transition Probability Heatmap

The learned transition matrix (visualized in Cell 13) shows:

- High self-transition probabilities (diagonal elements ~0.67-0.88)
- Realistic activity transitions (e.g., Still → Standing has 14.3% probability)
- Very low probability of unlikely transitions (e.g., Jumping → Standing is 0%)

Activity Sequence Plot

Comparative plots demonstrated the model's ability to track activity transitions over time, although some temporal inconsistencies were observed in the predictions.

---

5. Discussion and Conclusion

5.1 Activity Distinguishing Difficulty

Easiest to Distinguish: Jumping and Still

- Jumping: Characterized by high spectral energy, large variance, and distinct frequency signatures
- Still: Low variance and minimal energy across all axes

Hardest to Distinguish: Standing and Walking

- Both involved upright posture with similar gravity components

- Walking's additional motion featured help separation, but subtle differences created confusion
- The confusion matrix showed Walking being misclassified as Standing

5.2 Transition Probabilities Analysis

The learned transition matrix reflects realistic behavior:

- **High persistence probabilities** (diagonal values ~0.67-0.88): People maintain activities before switching
- **Natural transitions:**
  - Still → Standing (14.3%): Common posture change
  - Jumping → Walking (28.1%): Natural sequence after jumping
- **Low probability transitions:**
  - Jumping → Standing (0%): Requires intermediate Still state
  - Walking → Jumping (0%): Less common transition pattern

5.3 Effect of Sensor Characteristics

Sampling Rate (100 Hz):

- Adequate for capturing motion dynamics
- Sufficient frequency resolution for band-specific analysis
- Higher rates might improve precision, but the current rate is sufficient

Sensor Noise:

- Addressed through feature engineering (e.g., RMS, energy)
- Spectral domain analysis helps separate the signal from the noise
- Standardization applied during preprocessing

5.4 Model Limitations

1. **State-Activity Mapping Challenge:**
    - Viterbi decoding produces 4 states, but mapping to activities is non-trivial
    - Majority voting is used, but it may not capture all activity patterns
    - Results in varying specificity/sensitivity metrics
2. **Limited Training Data:**
    - Only 411 feature windows from 43 recordings
    - Insufficient diversity in walking patterns, environments
    - Model likely overfitted to specific collection conditions
3. **Feature Redundancy:**
    - Some features are highly correlated (e.g., variance and std dev)
    - Dimensionality reduction could improve generalization
4. **Cross-Validation Performance Gap:**
    - Test accuracy: 100% (possibly biased split)
    - CV accuracy: ~47% (more realistic performance estimate)

5.5 Proposed Improvements

1. **Expand Dataset:**
    - Collect 200+ samples per activity
    - Include data from different users
    - Vary collection environments (indoor/outdoor, different devices)
2. **Advanced Feature Engineering:**

    - Principal Component Analysis (PCA) for dimensionality reduction
    - Time-delay features to capture temporal patterns
    - Include magnetometer data for orientation context
3. **Model Architecture:**
    - Use hierarchical HMMs for sub-activities
    - Implement state duration modeling for realistic transitions
    - Consider deep learning approaches (LSTMs) for sequential modeling
4. **Evaluation Methodology:**
    - Person-independent validation
    - More sophisticated state-to-activity mapping
    - Real-time deployment testing

5.6 Conclusion

This project successfully demonstrated the application of Hidden Markov Models to human activity recognition using data from smartphone sensors. The model achieved reasonable performance in distinguishing activities, with frequency-domain features proving most discriminative. The learned transition probabilities reflected realistic human behavior patterns, with high self-transition probabilities indicating natural persistence in activities.

Key findings:

- Low-frequency spectral energy is the most important feature for activity discrimination
- Jumping and still states are most easily separated
- Standing and walking create confusion due to similar postural characteristics
- Cross-validation reveals more realistic performance (~47%) than initial test results suggest

The project provides a solid foundation for applications in healthcare monitoring, fitness tracking, and human-computer interaction systems. Future work should focus on expanding dataset diversity and improving the robustness of the state-activity mapping to achieve production-ready deployment.