Hidden Markov Model for Sequential Data

Dr.-Ing. Michelle Karg
mekarg@uwaterloo.ca

Electrical and Computer Engineering
Cheriton School of Computer Science
Sequential Data

• Measurement of time series:

  Example: Motion data

• Others:
  – Characters in a Sentence
  – Nucleotide base pairs along a strand of DNA sequence

  Example: Speech data [1]
Sequential Data

• Characteristic:
  – Dependence on previous observations
    \[ P(q_t = S_i \mid q_{t-1} = S_j, q_{t-2} = S_k, \ldots) = P(q_t = S_i \mid q_{t-1} = S_j) \]
    More recent observations likely to be more relevant
  – Stationary versus nonstationary sequential distributions
    Stationary: generative distribution not evolving with time

• Tasks:
  – Predict next value in a time series
  – Classify time series
Methods

Deterministic Models:
• Frequency analysis
• Statistical Features: (e.g., mean) + classification
• Dynamic time warping

Probabilistic Models:
• Hidden Markov Models
Frequency Analysis

- Fourier transform
  - Amplitude of frequency

- Pro:
  - Visualization

- Disadvantage:
  - No information about previous state

Example: speech data [1]
Statistical Features

• Transformation of time series into a \textit{set of features} → Conventional classification

• Example: Emotion recognition in gait [2]
  Step length, time, velocity: 84 % (NN)
  Min, mean, max: 93 % (Naive Bayes)

\begin{center}
\begin{figure}
\centering
\includegraphics[width=\textwidth]{time_series.png}
\caption{Time series [2]}
\end{figure}
\end{center}
Statistical Features

- Questions:
  - Which descriptors to calculate?
    - Feature Selection
  - Window size?
Statistical Features

- Questions:
  - Which descriptors to calculate?
    - Feature Selection
  - Window size?

- Pro:
  - Simple approach, fast

- Disadvantage:
  - Could be easily distorted by noise
Dynamic Time Warping

• Similarity measure between two sequences:
  Spatio-temporal correspondence
• Minimize error between sequence and reference:

\[ E_c[\xi, \tau] = \int \left[ |\xi(t)|^2 + \lambda \tau(t)^2 \right] dt \]
Dynamic Time Warping

• Computation: 1. Local cost measure
  – Distance measure (e.g., Euclidean, Manhattan)
  – Sampled at equidistant points in time

Cost matrix $C$ for time series $X$ and $Y$ [6]
Dynamic Time Warping

• Computation: 1. Local cost measure
  – Distance measure (e.g., Euclidean, Manhattan)
  – Sampled at equidistant points in time

Cost matrix $C$ for time series $X$ and $Y$ [6]
Dynamic Time Warping

- Computation: 1. Local cost measure
  - Distance measure (e.g., Euclidean, Manhattan)
  - Sampled at equidistant points in time

Cost matrix $C$ for time series $X$ and $Y$ [6]
Dynamic Time Warping

2. Find **optimal warping path**:
   - Boundary condition: \( p_1 = (1,1) \) and \( p_L = (N,M) \)
   - Monotonicity condition: \( n_1 \leq n_2 \leq \ldots \leq n_L \) and \( m_1 \leq m_2 \leq \ldots \leq m_L \)
   - Step size condition

Which figure fulfills all conditions? [6]
Dynamic Time Warping

- Result: Optimal warping path

\[ D(n, m) = \min\{D(n-1, m-1), D(n-1, m), D(n, m-1)\} + c(n, m) \]
Dynamic Time Warping

**Pro:**
- Very accurate
- Cope with different speeds
- Can be used for generation

**Disadvantages:**
- Alignment of segments?
  (e.g., different length)
- Computationally intensive
- Usually applied to low-dimensional data (1-dim.)

Generation: Morphing [3]
Methods

Deterministic Models:
• Frequency analysis
• Statistical Features: (e.g., mean) + classification
• Dynamic time warping

Probabilistic Models:
• Hidden Markov Models
**Hidden Markov Model**

- Sequence of hidden states
- **Observations** in each state
- Markov property
- Parameters: Transition matrix, observation, prior

[5] “A Tutorial on HMM and Selected Applications in Speech Recognition”
Hidden Markov Model

- Topology of transition matrix
- Model for the observations
- Methodology (3 basic problems)
- Implementation Issues
Topology of Transition Matrix A

• Markov Chain:
  Considering the previous state!

• Transition matrix A:
  – $0 \leq a_{ij} \leq 1$
  Transactions of the hidden states

• Topologies:
  – Ergodic or fully connected
  – Left-right or Bakis model (cyclic, noncyclic)
  – Note: the more “0”, the faster computation!

• What happens if ....
  – All entries of A are equal?
  – All entries in a row/column are zero except for diagonal?
Example for Markov Chain

• Given 3 states and A
  – **State 1**: rain or snow, **state 2**: cloudy, **state 3**: sunny
    
    \[
    A = \begin{bmatrix}
    0.4 & 0.3 & 0.3 \\
    0.2 & 0.6 & 0.2 \\
    0.1 & 0.1 & 0.8 \\
    \end{bmatrix}
    \]

• Questions:
  – If the sun shines now, what is the most probable weather for tomorrow?
  – What is the probability that the weather for the next 7 days will be: “sun – rain – rain – sun – cloudy – sun”
    Given, that the sun shines today;
Hidden Markov Model

- Markov Chain: States are observable
- HMM: states are not observable, only the observations

Observations are either
- Discrete, e.g., icy - cold – warm
- Continuous, e.g., temperature

Comparison of Markov Chain and HMM [4]
HMM – Discrete Observations

- A number of $M$ distinct observation symbols per state:
  - Vector quantization of continuous data
- Observation Matrix $B$
Continuous Density HMM

- Example: Identification using gait [7]
  - Extract silhouette from video

  *Width vector profile during gait steps [7]*

  - FED vector for observation:
    * 5 stances: \( e_n \)
    * Distance
      \[ f_{ed}^n(t) = d(x(t), e_n) \]
    * Distance: Gaussian distributed

*Gait as biometric [7]*

*FED vector components during a step [7]*
Design of an HMM

• An HMM is characterized by
  – The number of states: N
  – The number of distinct observation symbols M (only discrete !)
  – The state transition probabilities: A
  – The observation probability probability distributions
  – The initial state distribution $\pi$

• A model is described by the parameter set $\lambda$
  – $\lambda = (A, B, \pi)$
3 Basic Problems

1. **Learning:**
   
   Given:
   - Number of states $N$
   - The number of observations $M$
   - Structure of the model
   - Set of training observations

   *How to estimate the probability matrices $A$ and $B$?*

   **Solution:** *Baum-Welch* algorithm
   
   *(It can result in local maxima and the results depend on the initial estimates of $A$ and $B)*

   **Application:** Required for any HMM
Similarity Measure for HMMS

- Kullback-Leibler divergence
- Example: Movement imitation in robotics
  - Encode observed behavior as HMM
  - Calculate Kullback-Leibler divergence:
    - Existing or new behavior?
  - Build tree of human motions

Why not a metric?

Clustering human movement [8]

General Concept [8]
3 Basic Problems

2. Evaluation:

Given:
- Trained HMM $\lambda = (A, B, \pi)$
- Observation sequence $V = [v(1), v(2), \ldots, v(T)]$

What is the conditional probability $P(V|\lambda)$ that the observation sequence $V$ is generated by the model $\lambda$?

Solution: Forward-backward algorithm

(Straight-forward calculation of $P(V|\lambda)$ would be too computationally intensive)

Application: Classification of time series
Classification of Time Series

- Examples: Happy versus neutral gait
- Concept: An HMM is trained for each class c
  \[ \lambda_h = (A_h, B_h, \pi) \text{ and } \lambda_{neu} = (A_{neu}, B_{neu}, \pi) \]
- Calculation of the probabilities \( P(V|\lambda) \) for sequence \( V \)
  \[ P_h(V | \lambda_h) \text{ and } P_{neu}(V | \lambda_{neu}) \]
- Comparison:
  \[ P_h(V | \lambda_h) > P_{neu}(V | \lambda_{neu}) \]

Concept of HMM [4]
3 Basic Problems

3. Decoding:

Given:
- Trained HMM $\lambda = (A, B, \pi)$
- Observation sequence $V = [v(1), v(2), \ldots, v(T)]$

What is the most likely sequence of hidden states?

Solution: Viterbi algorithm

Application: Activity recognition
Implementation Issues

• Scaling
  – Rescale of forward and backward variables to avoid that computed variables exceed the precision range of machines

• Multiple observation sequences
  – Training

• Initial estimates of the HMM parameters
  – Random or uniform of $\pi$ and $A$ is adequate
  – Observation distributions: good initial estimate crucial

• Choice of the model
  – Topology of Markov Chain
  – Observation: discrete or continuous?
Implementation

An HMM can be used to
- Estimate a state sequence
- Classify sequential data
- Predict next value
- Build a generative model (e.g., application in robotics for motion imitation)

Real-world issues:
- Incomplete sequences
- Data differing in length
References