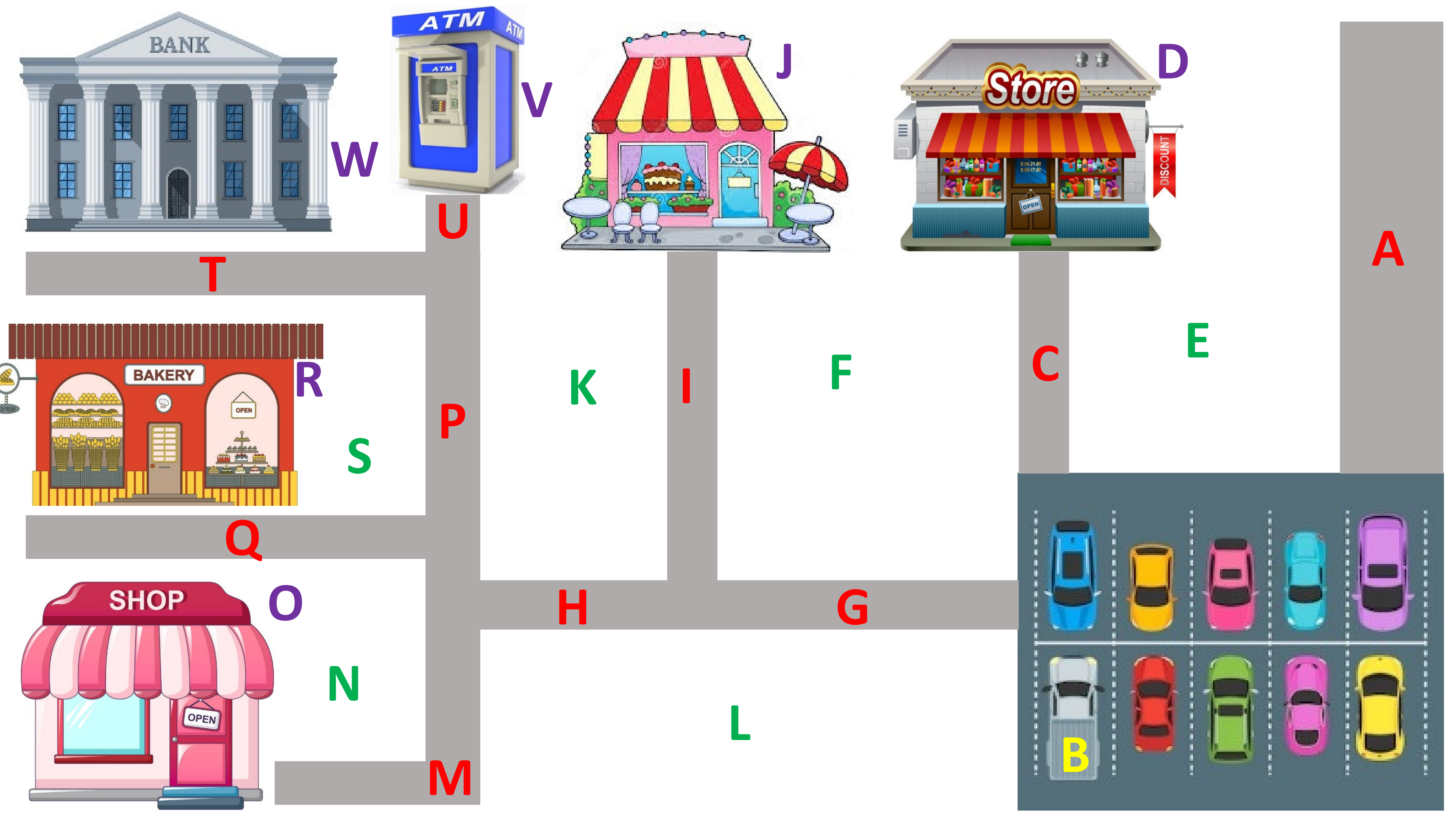
Hidden Markov Models

Prithwijit Guha
Dept. of EEE, IIT Guwahati

Unsupervised Learning

- Parametric Clustering Algorithms
- Generic Clustering Algorithms
- Estimation Theory
- **Generative Models**
- Pattern Mining

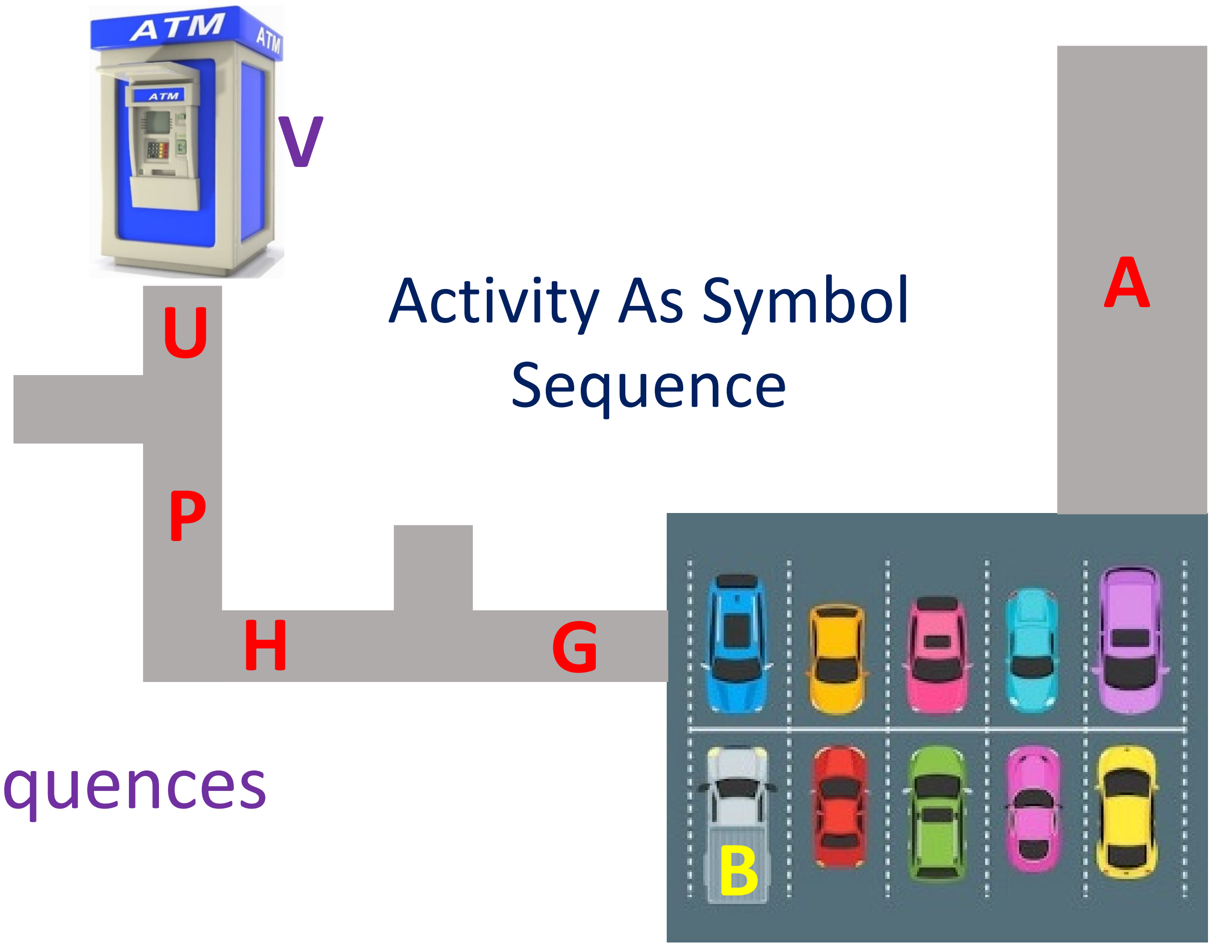
Monitoring a Shopping Area



BANK
ATM
W
V
U
J
D
Store
DISCOUNT
OPEN
T
A
E
BAKERY
OPEN
R
K
I
F
C
S
P
Q
O
SHOP
OPEN
H
G
N
L
B
M

Visiting the ATM

$A\ (N_A) \rightarrow B\ (N_B) \rightarrow G\ (N_G) \rightarrow H\ (N_H) \rightarrow P\ (N_P) \rightarrow U\ (N_U) \rightarrow V\ (N_V) \rightarrow U\ (N_U) \rightarrow$



Task: Modeling Symbol Sequences Generated by Activities

Time Series of Vectors

$$S_i = \left\{x_t^{(i)}; t = 1, \dots T_i\right\}; \mathrm{i} = 1, \dots n$$

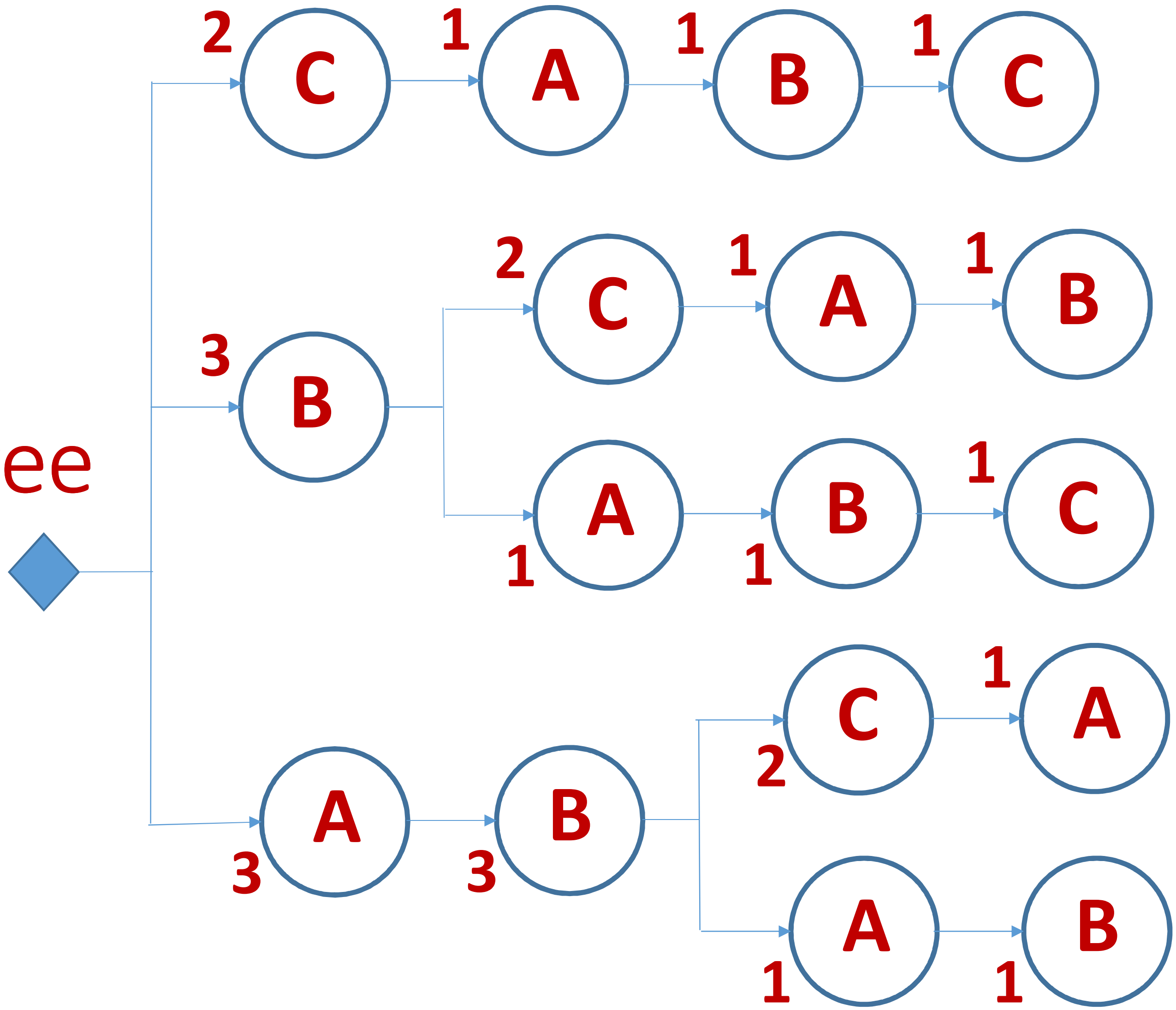
Time Series Datasets $\boldsymbol{S}_i$ are of Different Duration T_i

Task-1: Learning Model $\boldsymbol{M}_i$ for Time Series $\boldsymbol{S}_i$

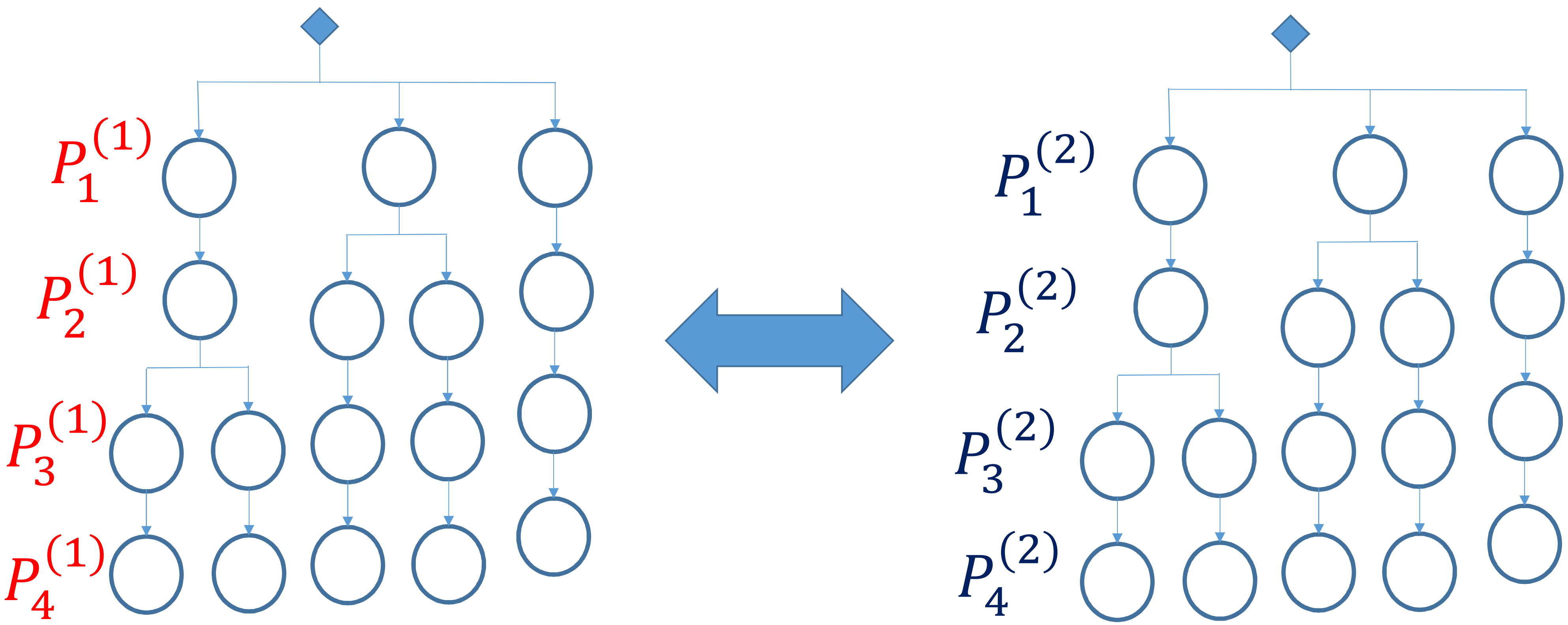
Task-2: Grouping Models $\boldsymbol{M}_i$ using Clustering Algorithms

Learning a Suffix Tree

A B A B C D B D



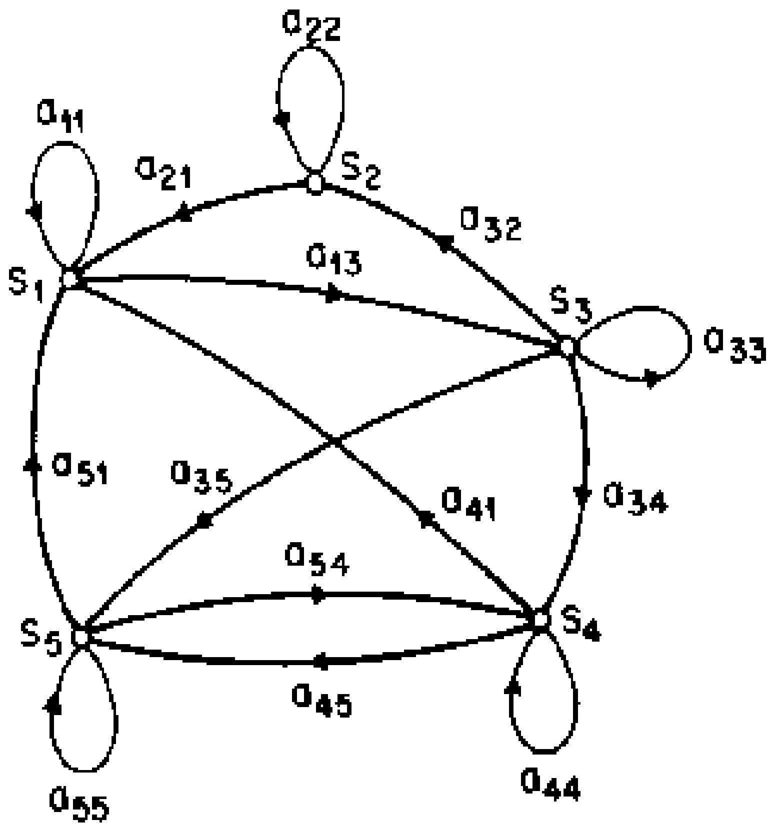
Similarity between Two Suffix Trees



Similarity between Two Suffix Trees

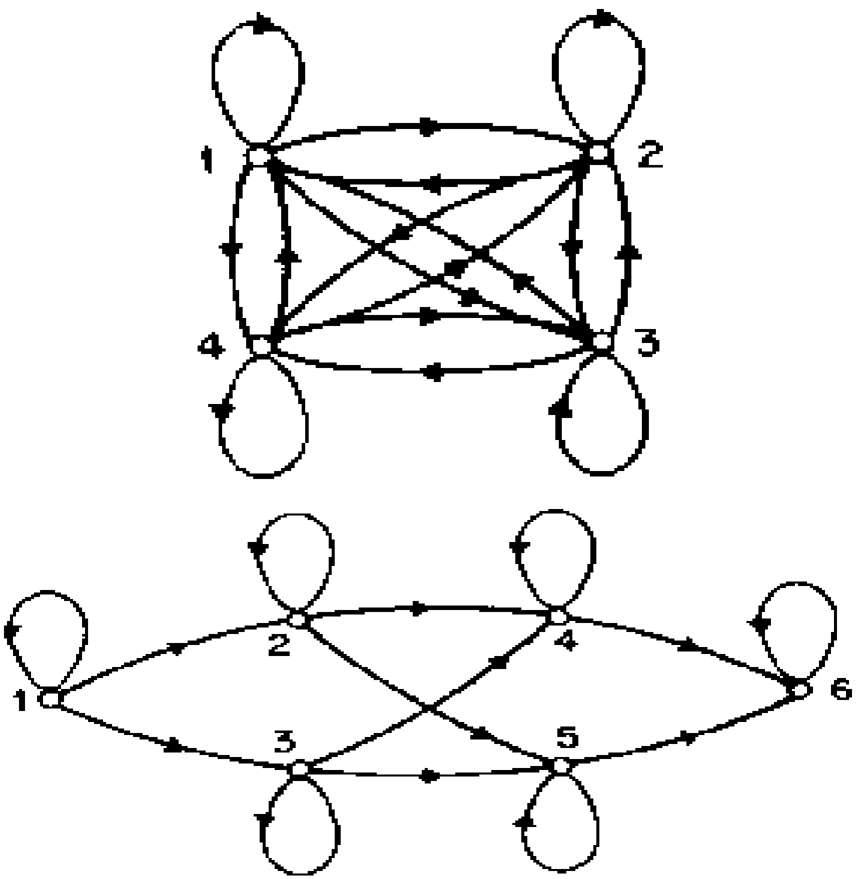
$$\mu_s(T_1, T_2) = \frac{\sum_{d=1}^{D} \omega_d BC\left(P_d^{(1)}, P_d^{(2)}\right)}{\sum_{d=1}^{D} \omega_d}$$

Markov Process



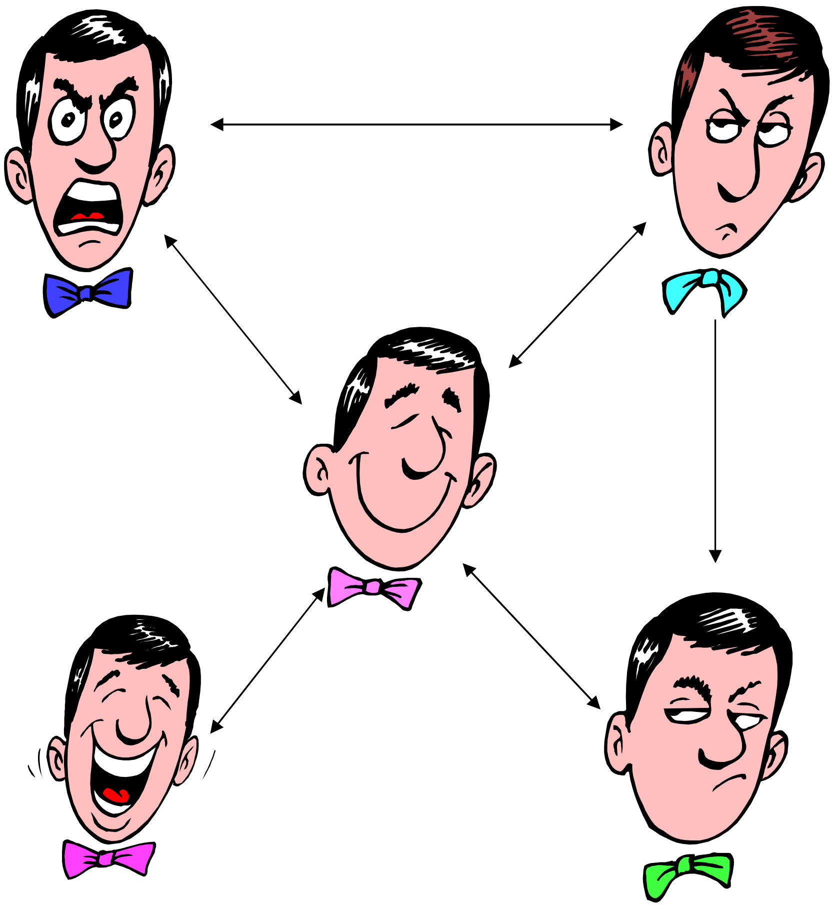
- Stochastic State Machine
- Observable States
- Observable Sequence
- Signal Generation
  - State Transitions
  - State-Output Associations

The State Machine



- System Model
  - Set of Internal States
  - Transition Dynamics
    - Deterministic
    - Stochastic
- Signal Model
  - External Manifestation
  - Internal State
  - Output As
    - Deterministic Reasoning
    - Probabilistic Inferencing

The Hidden States



- A Special Case
  - Assuming an Abstract System
  - Abstract System States
  - No Physical Association
  - Mathematical Significance
- Hidden Markov Process
  - Stochastic Transitions
  - Probabilistic Association
- Analogy of Behavior
  - Person as System
  - Mood as Hidden States
  - Output as Facial Expression

Notations

$\boldsymbol{S} = \{s_i; i = 1, \ldots N\}$: Set of States

$\boldsymbol{Q} = \{q_t; t = 1, \ldots T\}$: Ordered Sequence of States

$\boldsymbol{O} = \{o_t; t = 1, \ldots T\}$: Ordered Sequence of Observations

$\boldsymbol{V} = \{v_k; k = 1, \ldots M\}$: Set of Observable Symbols

$\boldsymbol{N}$ Hidden States

$\boldsymbol{M}$ Observable Symbols

$\boldsymbol{T}$ Observations

Notations

$q_t = s_i$: State at Instant t is s_i

$o_t = v_k$: Observation at Instant t is v_k

$\boldsymbol{Q_t}$: Time Ordered Set of States till Instant t

$\boldsymbol{O_t}$: Time Ordered Set of Observations till Instant t

$\boldsymbol{Q}^{\star}$: Optimal State Sequence

Notations

State Transition Probability

$$a_{ij} = P(q_{t+1} = s_j \mid q_t = s_i)$$

State-Output Association Probability

$$b_{jk} = b_j(v_k) = P(o_t = v_k \mid q_t = s_j)$$

Initial State Probability

$$\pi_i = P(q_1 = s_i)$$

Notations

State Transition Probability Matrix

$$\boldsymbol{A}_{(N\times N)} = \{a_{ij}\}$$

State-Output Association Probability Matrix

$$\boldsymbol{B}_{(N\times M)} = b_{jk}$$

Initial State Probability Array

$$\boldsymbol{\pi}_{(\mathbf{1}\times \boldsymbol{N})} = \{\pi_i\}$$

Hidden Markov Model

$$\boldsymbol{\lambda} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$$

Assumptions

First Order Process

$$P(\, q_{t+1} = s_j \mid q_t = s_i, q_{t-1} = s_m, \ldots \,) = P(q_{t+1} = s_j \mid q_t = s_i)$$

Stationarity or Time Homogeneity

$$P(\, q_{t+m+1} = s_j \mid q_{t+m} = s_i \,) = P(q_{t+1} = s_j \mid q_t = s_i)$$

Observation Independence

$$P(\, \boldsymbol{O} \mid \boldsymbol{\lambda} \,) = P(\, o_1 \mid \boldsymbol{\lambda} \,)P(\, o_2 \mid \boldsymbol{\lambda} \,) \ldots P(\, o_T \mid \boldsymbol{\lambda} \,)$$

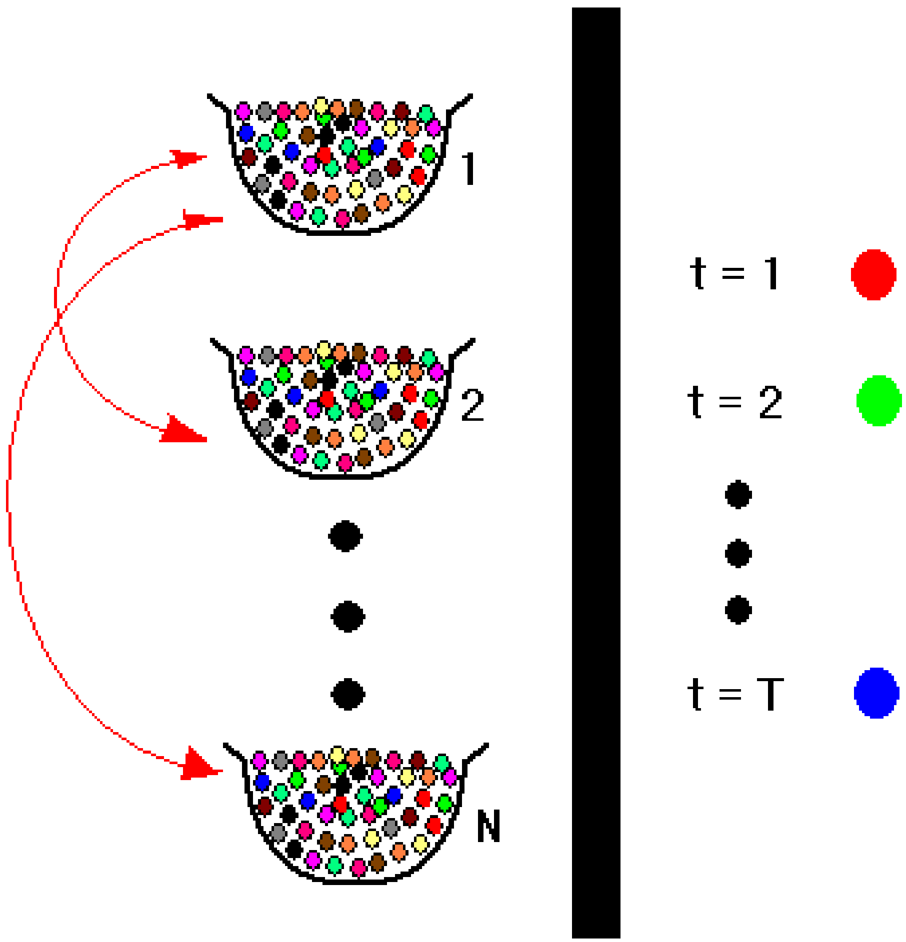
Constraints

Initialization Probabilities $\forall i \; \pi_i \geq 0; \; \sum_{i=1}^{N} \pi_i = 1$

State Transition Probabilities $\forall i, j \; a_{ij} \geq 0; \; \sum_{j=1}^{N} a_{ij} = 1$

Output Association Probabilities $\forall j, k \; b_{jk} \geq 0; \; \sum_{k=1}^{M} b_{jk} = 1$

The Urn-Ball Model



- Experiment Setup
  - Consider N Urns
  - M Balls of Distinct Colors
  - Different Ball Distribution in Urns
  - Urns “Hidden” in a Room
  - Experimenter in Room
- Experiment
  - Urn Chosen Randomly
  - Ball Picked Up Randomly
  - Observation as Color
- Explanation
  - Urns as Hidden States
  - Balls as Output Symbols
  - Experiment as Generating Process

HMM: Illustration Purposes

$\boldsymbol{N} = \mathbf{3}$ Hidden States

$\boldsymbol{M} = \mathbf{3}$ Observable Symbols

$\boldsymbol{T} = \mathbf{3}$ Observations

$$\boldsymbol{A} = \begin{bmatrix} 0.1 & 0.5 & 0.4 \\ 0.6 & 0.1 & 0.3 \\ 0.3 & 0.6 & 0.1 \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} 0.3 & 0.4 & 0.3 \\ 0.4 & 0.2 & 0.4 \\ 0.6 & 0.2 & 0.2 \end{bmatrix}$$

$$\boldsymbol{\pi} = \begin{bmatrix} 0.3 & 0.4 & 0.3 \end{bmatrix}$$

$$\boldsymbol{\lambda_I} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$$

$$\boldsymbol{O} = \begin{bmatrix} v_2 & v_3 & v_1 \end{bmatrix}$$

HMM: Evaluation

Given Observation Sequence $\boldsymbol{O}$ and Hidden Markov Model $\boldsymbol{\lambda} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$

$P(\boldsymbol{O} \mid \boldsymbol{\lambda})$: Probability that $\boldsymbol{\lambda}$ has generated $\boldsymbol{O}$

Direct Method

Backward-Forward Algorithm

Probability Recaps

Conditional Probability

$$P(X, Y \mid Z) = P(X \mid Y, Z)P(Y \mid Z)$$

Marginalization

$$\sum_{i=1}^{K} P(X, Y_i \mid Z) = P(X \mid Z)$$

Corollary

$$\sum_{i=1}^{K} P(X, V_i \mid Z)P(Y, V_i \mid Z) = P(X, Y \mid Z)$$

Evaluation: Direct Method

$$P(\boldsymbol{O} \mid \boldsymbol{\lambda}) = \sum_{\boldsymbol{Q} \in \{\boldsymbol{Q}\}} P(\boldsymbol{O} \mid \boldsymbol{Q}, \boldsymbol{\lambda}) P(\boldsymbol{Q} \mid \boldsymbol{\lambda})$$

N^T Possible Paths

$$P(\boldsymbol{O} \mid \boldsymbol{Q}, \boldsymbol{\lambda}) = \prod_{t=1}^{T} P(o_t \mid q_t, \boldsymbol{\lambda}) = \prod_{t=1}^{T} b_{q_t}(o_t)$$

T Multiplications

Evaluation: Direct Method

$$P(\boldsymbol{Q} \mid \boldsymbol{\lambda}) = P(q_t \mid \boldsymbol{Q}_{t-1}, \boldsymbol{\lambda})P(\boldsymbol{Q}_{t-1} \mid \boldsymbol{\lambda})$$

$$= P(q_t \mid q_{t-1}, \boldsymbol{Q}_{t-2}, \boldsymbol{\lambda})P(\boldsymbol{Q}_{t-1} \mid \boldsymbol{\lambda}) = P(q_t \mid q_{t-1}, \boldsymbol{\lambda})P(\boldsymbol{Q}_{t-1} \mid \boldsymbol{\lambda})$$

$$P(\boldsymbol{Q} \mid \boldsymbol{\lambda}) = P(q_t \mid q_{t-1}, \boldsymbol{\lambda})P(q_t \mid q_{t-1}, \boldsymbol{\lambda}) \ldots P(q_2 \mid q_1, \boldsymbol{\lambda})P(q_1 \mid \boldsymbol{\lambda})$$

$$P(\boldsymbol{Q} \mid \boldsymbol{\lambda}) = \left(\prod_{t=2}^{T} P(q_t \mid q_{t-1}, \boldsymbol{\lambda})\right)P(q_1 \mid \boldsymbol{\lambda})$$

T Multiplications

Evaluation: Forward Variable

The Forward Variable

$$\alpha_t(i) = P(\boldsymbol{O}_t, q_t = s_i \mid \boldsymbol{\lambda}) = P(o_1, \ldots o_t, q_t = s_i \mid \boldsymbol{\lambda})$$

Initialization

$$\alpha_1(i) = P(o_1, q_1 = s_i \mid \boldsymbol{\lambda})$$

$$P(o_1, q_1 = s_i \mid \boldsymbol{\lambda}) = P(o_1 \mid q_1 = s_i, \boldsymbol{\lambda})P(q_1 = s_i \mid \boldsymbol{\lambda})$$

$$\alpha_1(i) = b_i(o_1)\pi_i$$

Forward Variable: Induction

$$\alpha_{t+1}(j) = P(\boldsymbol{O}_{t+1}, q_{t+1} = s_j \mid \boldsymbol{\lambda}) = P(o_1, \ldots o_t, o_{t+1}, q_{t+1} = s_j \mid \boldsymbol{\lambda})$$

$$P(\boldsymbol{O}_t, o_{t+1}, q_{t+1} = s_j \mid \boldsymbol{\lambda}) = P(\boldsymbol{O}_t \mid o_{t+1}, q_{t+1} = s_j, \boldsymbol{\lambda}) P(o_{t+1}, q_{t+1} = s_j \mid \boldsymbol{\lambda})$$

$$= P(\boldsymbol{O}_t \mid q_{t+1} = s_j, \boldsymbol{\lambda}) P(o_{t+1} \mid q_{t+1} = s_j, \boldsymbol{\lambda}) P(q_{t+1} = s_j \mid \boldsymbol{\lambda})$$

$$= P(\boldsymbol{O}_t, q_{t+1} = s_j \mid \boldsymbol{\lambda}) b_j(o_{t+1})$$

$$= \left(\sum_{i=1}^{N} P(\boldsymbol{O}_t, q_t = s_i \mid \boldsymbol{\lambda}) P(q_{t+1} = s_j \mid q_t = s_i, \lambda) \right) b_j(o_{t+1})$$

Forward Variable: Induction

$$\alpha_{t+1}(j) = \left(\sum_{i=1}^{N} P(\boldsymbol{O}_t, q_t = s_i \mid \boldsymbol{\lambda})\, a_{ij} \right) b_j(o_{t+1})$$

N Evaluations

N Multiplications

N^2 Multiplications per instant

Evaluation: Forward Variable

$$P(\boldsymbol{O} \mid \boldsymbol{\lambda}) = \sum_{i=1}^{N} P(\boldsymbol{O}_T, q_T = s_i \mid \lambda)$$

$$P(\boldsymbol{O} \mid \boldsymbol{\lambda}) = \sum_{i=1}^{N} \alpha_T(i)$$

Evaluation: Backward Variable

Backward Variable $\beta_t(i) = P(o_{t+1}, \dots o_T \mid q_t = s_i, \boldsymbol{\lambda})$

Initialization $\beta_T(i) = 1$

Induction $\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$

Final Evaluation $P(\boldsymbol{O} \mid \boldsymbol{\lambda}) = \sum_{i=1}^{N} b_i(o_1) \beta_1(i)$

Evaluation: Computation Gain

Direct Method	Forward/Backward
➢ Computation Intensive ➢ 2T Computations/Path ➢ N^T Possible Paths ➢ $O(2TN^T)$	➢ Very Efficient ➢ N^2 Computations/Instant ➢ T Possible Instants ➢ $O(TN^2)$

HMM: State Sequencing

Given Observation Sequence $\boldsymbol{O}$ and Hidden Markov Model $\boldsymbol{\lambda} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$

Optimal State Sequence to generate $\boldsymbol{O}$ using $\boldsymbol{\lambda}$

Instantaneous Best State Approach

Viterbi Algorithm

HMM: State Sequence Estimation

- How to Choose the Best???
- Given, the Observation and Model
- Deciding Criteria
  - Individualistic Approach
  - Group Approach
- Optimizing w.r.t. Path
  - Maximize $P(Q|O,\lambda)$
  - Viterbi Algorithm

State Occupancy Measure $\boldsymbol{\gamma_t(i)}$

$$\gamma_t(i) = P(q_t = s_i \mid \boldsymbol{O}, \boldsymbol{\lambda}) = \frac{P(\boldsymbol{O}, q_t = s_i \mid \boldsymbol{\lambda})}{P(\boldsymbol{O} \mid \boldsymbol{\lambda})}$$

$$P(\boldsymbol{O}, q_t = s_i \mid \boldsymbol{\lambda}) = P(\mathrm{o}_1, \dots o_t, o_{t+1}, \dots o_T, q_t = s_i \mid \boldsymbol{\lambda})$$

$$= P(\mathrm{o}_1, \dots o_t, q_t = s_i \mid \boldsymbol{\lambda}) P(\mathrm{o}_{\mathrm{t+1}}, \dots o_T \mid q_t = s_i, \boldsymbol{\lambda}) = \alpha_t(i)\beta_t(i)$$

$$\sum_{i=1}^{N} P(\boldsymbol{O}, q_t = s_i \mid \boldsymbol{\lambda}) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) = P(\boldsymbol{O} \mid \boldsymbol{\lambda})$$

Individually, Most Likely…

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}$$

Note: γ induces a Probability Measure

Algorithm

- Compute for each Instant
- Assign the highest one
- Proceed to form the Path

Problems

- Self-Centered Approach
- Do not Consider Neighbors
- May form Invalid Transitions

Viterbi Algorithm

Partial Path Measure

$$\delta_t(i) = \max_{q_1,\dots q_{t-1}} P(q_1, \dots q_{t-1}, q_t = s_i, o_1, \dots o_t \mid \boldsymbol{\lambda})$$

Initialization

$$\delta_1(i) = P(\, q_1 = s_i, o_1 \mid \boldsymbol{\lambda}\,)$$

$$\delta_1(i) = P(\, o_1 \mid q_1 = s_i, \boldsymbol{\lambda}\,)\mathrm{P}(\, \mathrm{q}_1 = \mathrm{s}_\mathrm{i} \mid \boldsymbol{\lambda}\,) = b_i(o_1)\pi_i$$

$$\psi_1(i) = 0$$

Viterbi Algorithm

$$\delta_{t-1}(i)a_{ij} = \max_{q_1,\dots q_{t-2}} P(\underbrace{q_1,\dots q_{t-2}}_{\text{Partially Optimal Path}}, \underbrace{q_{t-1} = s_i, o_1,\dots o_{t-1} \mid \boldsymbol{\lambda})P(q_t = s_j \mid q_{t-1} = s_i, \boldsymbol{\lambda})}_{\text{Jump from i to j}}$$

Induction

$$\delta_t(j) = \max_{1 \le i \le N} \{\delta_{t-1}(i)a_{ij}\}\, b_i(o_1)$$

$$\psi_t(j) = \underset{1 \le i \le N}{argmax} \{\delta_{t-1}(i)a_{ij}\}$$

Viterbi Algorithm

Final Evaluation

$$P^\star = \max_{1 \le i \le N} \delta_T(i) \qquad\qquad Q^\star = \operatorname*{argmax}_{1 \le i \le N} \delta_T(i)$$

Back-Tracking

$$q_t^\star = \psi_{t+1}(q_{t+1}^\star)$$

$$1 \le t \le T - 1$$

HMM: Learning

Given Observation Sequence $\boldsymbol{O}$ and Hidden Markov Model $\boldsymbol{\lambda} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$

Adjusting $\boldsymbol{\lambda} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$ to Maximize $P(\boldsymbol{O} \mid \boldsymbol{\lambda})$

Baum-Welch Re-estimation Algorithm

HMM: Learning

- Given, the Observation Sequence $\boldsymbol{O}$
- Search in Model Space $\{\boldsymbol{\lambda}\}$
- Best Model to Generate Given Sequence
- To Maximize $\mathrm{P}(\boldsymbol{O} \mid \boldsymbol{\lambda})$
- Optimization w.r.t. $(\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$
- Maximization through
  - Constrained Gradient Ascent Optimization
  - Expectation Maximization (Baum-Welch) Algorithm

Joint State Measure $\eta_t(i, j)$

$$\eta_t(i,j) = P(q_t = s_i, q_{t+1} = s_j \mid \boldsymbol{O}, \boldsymbol{\lambda})$$

$$P\left(q_t = s_i, q_{t+1} = s_j \mid \boldsymbol{O}, \boldsymbol{\lambda} \right) = \frac{P(\boldsymbol{O}, q_t = s_i, q_{t+1} = s_j \mid \boldsymbol{\lambda})}{P(\boldsymbol{O} \mid \boldsymbol{\lambda})}$$

$$P\left(q_t = s_i, q_{t+1} = s_j \mid \boldsymbol{O}, \boldsymbol{\lambda} \right) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{l=1}^{N} \sum_{r=1}^{N} \alpha_t(l) a_{lr} b_r(o_{t+1}) \beta_{t+1}(r)}$$

$$\sum_{j=1}^{N} \eta_t(i,j) = \gamma_t(i)$$

Joint State Measure: Observations

$$\eta_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}$$

$$\sum_{j=1}^{N}\eta_t(i,j) = \gamma_t(i)$$

η induces a Probability Measure

$\sum_{t=1}^{T-1}\gamma_t(i)$: Expected Number of Transitions from s_i

$\sum_{t=1}^{T-1}\eta_t(i,j)$: Expected Number of Transitions from s_i to s_j

Baum-Welch Re-estimation (π)

$\bar{\pi}_i$ =Expected Number of Times at State s_i at $t = 1$

$$\bar{\pi}_i = \gamma_1(i)$$

Baum-Welch Re-estimation (A)

$$\overline{a_{ij}} = \frac{\text{Expected Number of Transitions from } s_i \text{ to } s_j}{\text{Expected Number of Transitions from } s_i}$$

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1} \eta_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

Baum-Welch Re-estimation (B)

$$\overline{b_{jk}} = \frac{\text{Expected Number of Times in State } s_j \text{ and Observing } v_k}{\text{Expected Number Times in State } s_j}$$

$$\overline{b_{jk}} = \frac{\sum_{t=1}^{T} \gamma_t(j)\delta[o_t; v_k]}{\sum_{t=1}^{T} \gamma_t(j)}$$

$$\delta[o_t; v_k] = \begin{cases} 1, & o_t = v_k \\ 0, & o_t \neq v_k \end{cases}$$

Notes on Re-estimation

- Ensures $P(\boldsymbol{O} \mid \bar{\boldsymbol{\lambda}}) \geq P(\boldsymbol{O} \mid \boldsymbol{\lambda})$
- Proposed By Baum-Welch
- Automatically Satisfies Stochastic Constraints
- Other Approach through Gradient Ascent
- Stochastic Constraints by Lagrange Multipliers
- Both Leads to same formulae
- Global Maxima is not Assured
- Frequently Local Maxima is Satisfactory

Continuous Data Sequences

Express $b_j(O)$ *as*

$$b_j(O)=\sum_{m=1}^{M} c_{jm}\Psi(O;\mu_{jm},U_{jm})$$

Satisfying Constraints

$$\sum_{m=1}^{M} c_{jm}=1 \; AND \; c_{jm}\geq 0$$

Define a Probability Measure

$$\gamma_t(j,k)=\left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}\right]\left[\frac{c_{jk}\Psi(o_t;\mu_{jk},U_{jk})}{\sum_{m=1}^{M}c_{jm}\Psi(o_t;\mu_{jm},U_{jm})}\right]$$

Re − estimation

$$\overline{c_{jk}}=\frac{\sum_{t=1}^{T}\gamma_t(j,k)}{\sum_{t=1}^{T}\sum_{k=1}^{M}\gamma_t(j,k)}$$

$$\overline{\mu_{jk}}=\frac{\sum_{t=1}^{T}\gamma_t(j,k)\cdot o_t}{\sum_{t=1}^{T}\gamma_t(j,k)}$$

$$\overline{U_{jk}}=\frac{\sum_{t=1}^{T}\gamma_t(j,k)\cdot(o_t-\mu_{jk})(o_t-\mu_{jk})^T}{\sum_{t=1}^{T}\gamma_t(j,k)}$$

Implementation Issues

- Finite Register Length Limitations
- Handling Multiple Training Data
- Tackling Insufficient Learning Data
- Parameter Initialization
- Choice of Model Dimensions

Scaling

$$Define \quad Scaling \quad Variable$$

$$c_t = \frac{1}{\sum_{i=1}^{N} \alpha_t(i)}$$

$$Scale \quad Forward \quad / \, Backward \quad Variables \quad as$$

$$\overset{scaled}{\alpha}_t(i) = c_t \alpha_t(i) \quad AND \quad \overset{scaled}{\beta}_t(i) = c_t \beta_t(i)$$

$$Evaluation \quad Pr\,obability$$

$$\ln[P(O \mid \lambda)] = -\sum_{t=1}^{T} c_t$$

Problems with Probability Values

- Less than 1
- Often very small
- Underflow when multiplied in long chain
- Occurs in Evaluation and Learning

Training Data Issues

Multiple Training Data

Express Data Set Evaluation as

$$P[O^{(1)}, O^{(1)}, \dots, O^{(K)} \mid \lambda) = \prod_{k=1}^{K} P[O^{(k)} \mid \lambda] = \prod_{k=1}^{K} P_k$$

New Learning Rules

$$\overline{a_{ij}} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k - 1} \alpha_t^k(i) a_{ij} b_j(o_{t+1}^k) \beta_{t+1}^k(j)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k - 1} \alpha_t^k(i) \beta_t^k(j)}$$

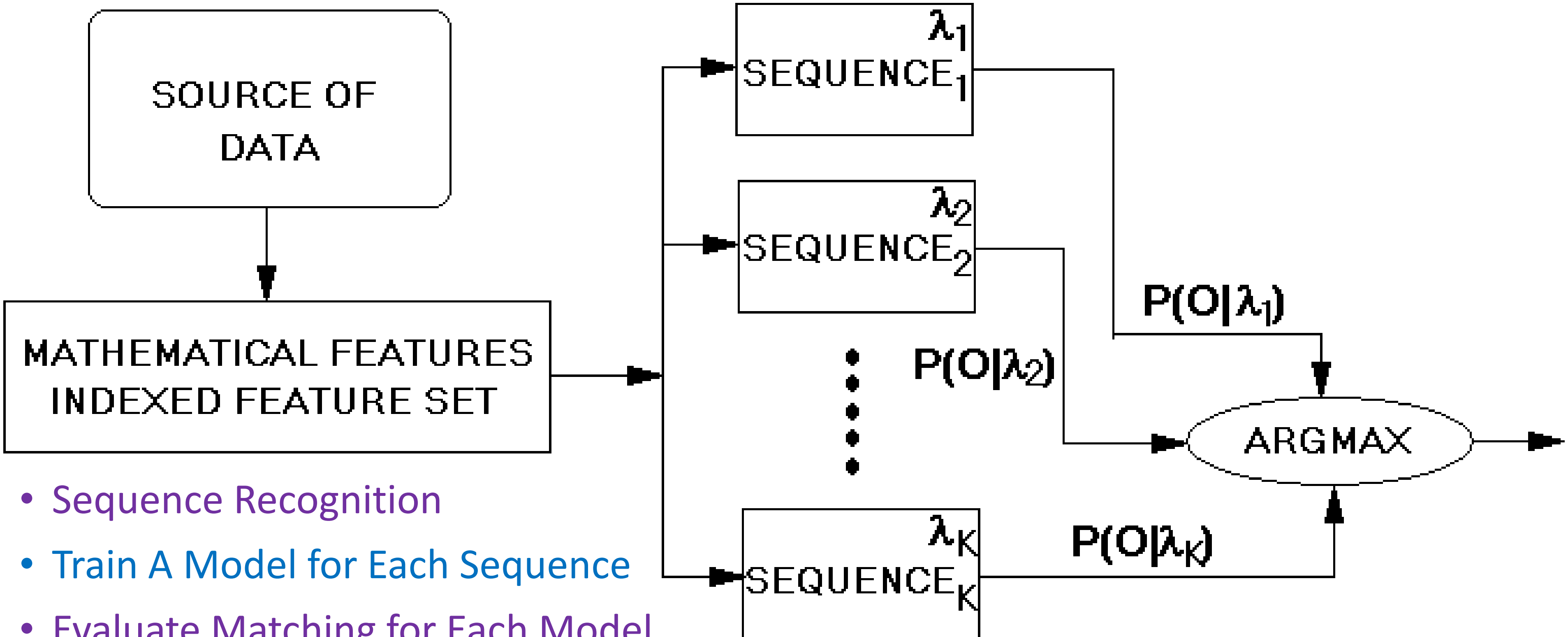
Insufficient Training Data

- Interpolate to Expand Data Set
- Model Interpolation

Parameter Issues

- Model Structure
  - Graphical Model Selection
  - Number of States
    - Observation Clustering
    - Growing and Pruning
- Initial Estimates
  - (A, π): Equally Likely Events
  - B: Statistical Analysis of Observation
    - Manual Segmentation
    - Mixture Models
    - K-Means Clustering

HMM: Classification



- Sequence Recognition
- Train A Model for Each Sequence
- Evaluate Matching for Each Model
- Highest Match Indicates the Sequence
- Speech/Gesture Recognition

Summary

- Hidden Markov Models
- Evaluation, State Sequencing & Learning
- Training Issues
- HMM Execution

Thank You