

Modele Markov Ascunse

De la Teorie la Aplicații

Alexandru Sorici, Tudor Berariu

Asociația Română pentru Inteligență Artificială

31 octombrie 2012

Outline

- 1 Aplicații în Învățarea Automată pentru MMA
 - Învățarea Automată
 - MMA în Învățarea Automată

Outline

- 1 Aplicații în Învățarea Automată pentru MMA
 - Învățarea Automată
 - MMA în Învățarea Automată
- 2 Teoria MMA
 - Cele Trei Probleme ale MMA
 - Fundamente Matematice

Outline

- 1 Aplicații în Învățarea Automată pentru MMA
 - Învățarea Automată
 - MMA în Învățarea Automată
- 2 Teoria MMA
 - Cele Trei Probleme ale MMA
 - Fundamente Matematice
- 3 Implementarea MMA
 - Problema Evaluării: Algoritmul Forward-Backward
 - Problema Interpretării: Algoritmul Viterbi
 - Problema Estimării: Algoritmul Baum-Welch
 - Learning from Observations: Baum-Welch algorithm

Outline

- 1 Aplicații în Învățarea Automată pentru MMA
 - Învățarea Automată
 - MMA în Învățarea Automată
- 2 Teoria MMA
 - Cele Trei Probleme ale MMA
 - Fundamente Matematice
- 3 Implementarea MMA
 - Problema Evaluării: Algoritmul Forward-Backward
 - Problema Interpretării: Algoritmul Viterbi
 - Problema Estimării: Algoritmul Baum-Welch
 - Learning from Observations: Baum-Welch algorithm
- 4 Demo: Recunoașterea Simbolurilor

Outline

- 1 Aplicații în Învățarea Automată pentru MMA
 - Învățarea Automată
 - MMA în Învățarea Automată
- 2 Teoria MMA
 - Cele Trei Probleme ale MMA
 - Fundamente Matematice
- 3 Implementarea MMA
 - Problema Evaluării: Algoritmul Forward-Backward
 - Problema Interpretării: Algoritmul Viterbi
 - Problema Estimării: Algoritmul Baum-Welch
 - Learning from Observations: Baum-Welch algorithm
- 4 Demo: Recunoașterea Simbolurilor
- 5 Tipuri de MMA

Outline

- 1 Aplicații în Învățarea Automată pentru MMA
 - Învățarea Automată
 - MMA în Învățarea Automată
- 2 Teoria MMA
 - Cele Trei Probleme ale MMA
 - Fundamente Matematice
- 3 Implementarea MMA
 - Problema Evaluării: Algoritmul Forward-Backward
 - Problema Interpretării: Algoritmul Viterbi
 - Problema Estimării: Algoritmul Baum-Welch
 - Learning from Observations: Baum-Welch algorithm
- 4 Demo: Recunoașterea Simbolurilor
- 5 Tipuri de MMA
- 6 Discuții și Concluzii

Outline

- 1 Aplicații în Învățarea Automată pentru MMA
 - Învățarea Automată
 - MMA în Învățarea Automată
- 2 Teoria MMA
 - Cele Trei Probleme ale MMA
 - Fundamente Matematice
- 3 Implementarea MMA
 - Problema Evaluării: Algoritmul Forward-Backward
 - Problema Interpretării: Algoritmul Viterbi
 - Problema Estimării: Algoritmul Baum-Welch
 - Learning from Observations: Baum-Welch algorithm
- 4 Demo: Recunoașterea Simbolurilor
- 5 Tipuri de MMA
- 6 Discuții și Concluzii

What is Machine Learning?

Machine Learning

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .

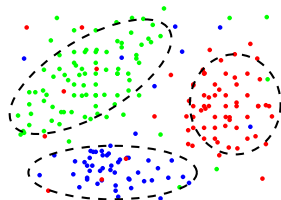
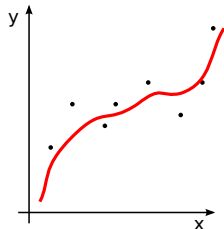
Machine Learning Applications

- Computer Vision: Google Car
- Machine Translation
- Speech Recognition
- Recommender Systems
- Intelligent Advertising

Machine Learning Classification

Types of Machine Learning Problems

- Regression
- Classification
- Reinforcement Learning
- supervised learning (eg. ..)
- unsupervised

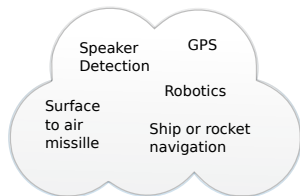


Outline

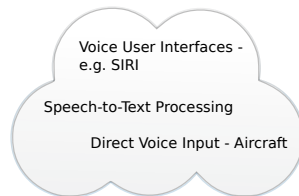
- 1 Aplicații în Învățarea Automată pentru MMA
 - Învățarea Automată
 - MMA în Învățarea Automată
- 2 Teoria MMA
 - Cele Trei Probleme ale MMA
 - Fundamente Matematice
- 3 Implementarea MMA
 - Problema Evaluării: Algoritmul Forward-Backward
 - Problema Interpretării: Algoritmul Viterbi
 - Problema Estimării: Algoritmul Baum-Welch
 - Learning from Observations: Baum-Welch algorithm
- 4 Demo: Recunoașterea Simbolurilor
- 5 Tipuri de MMA
- 6 Discuții și Concluzii

Sequence / Temporal problems (I)

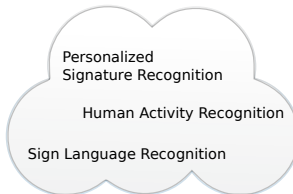
OBJECT TRACKING



SPEECH RECOGNITION



GESTURE RECOGNITION



Sequence / Temporal problems (II)

BIOINFORMATICS

Protein Sequencing

Modeling of a Gene
Regulatory Network

ECONOMICS

Stock Price Prediction

Econometrics

- estimate a country's economic indicators across time -

Probabilistic Reasoning over Time - Models

Consider some of the previously presented problems ...

Probabilistic Reasoning over Time - Models

Consider some of the previously presented problems ...

How do we model such dynamic situations?

Probabilistic Reasoning over Time - Models

Consider some of the previously presented problems ...

How do we model such dynamic situations?

States and Observations

- The process of change is viewed as a series of **time slices (snapshots)**
- Each time slice contains a set of random variables
 - \mathbf{O}_t - set of all **observable** evidence variables at time t
 - \mathbf{Q}_t - set of all **unobservable / hidden** state variables at time t

Probabilistic Reasoning over Time - Assumptions

Consider some of the previously presented problems ...

Probabilistic Reasoning over Time - Assumptions

Consider some of the previously presented problems ...

What **assumptions** (if any) do we make?

Probabilistic Reasoning over Time - Assumptions

Consider some of the previously presented problems ...

What **assumptions** (if any) do we make?

Stationary Process

The process of change is governed by laws **that do not themselves change over time**.

Implication: we need to specify conditional distributions only for the variables within a *representative* timeslice.

Probabilistic Reasoning over Time - Assumptions

Consider some of the previously presented problems ...

What **assumptions** (if any) do we make?

Stationary Process

The process of change is governed by laws **that do not themselves change over time**.

Implication: we need to specify conditional distributions only for the variables within a *representative* timeslice.

Markov Assumption

The current state in a process of change depends only on a **finite history** of previous states.

Implication: there is a **bounded** number of “parents” for the variables in each time slice.

$$P(Q_t | Q_{1:t-1}) = P(Q_t | Q_{t-1}) \quad P(O_t | Q_{1:t}, Q_{1:t-1}) = P(O_t | Q_t)$$

Probabilistic Reasoning over Time - Inference

What are the basic inference tasks that must be solved?

Probabilistic Reasoning over Time - Inference

What are the basic inference tasks that must be solved?

Filtering (monitoring)

The task of computing the **belief state** - the posterior distribution over the **current state**, given all evidence to date.

$$P(\mathbf{Q}_t | \mathbf{o}_{1:t})$$

Probabilistic Reasoning over Time - Inference

What are the basic inference tasks that must be solved?

Filtering (monitoring)

The task of computing the **belief state** - the posterior distribution over the **current state**, given all evidence to date.

$$P(\mathbf{Q}_t | \mathbf{o}_{1:t})$$

Evaluation (likelihood)

The task of computing the **likelihood** of the evidence up to present.

$$P(\mathbf{o}_{1:t})$$

Probabilistic Reasoning over Time - Inference

Prediction

The task of computing the posterior distribution over the **future state**, given all evidence to date.

$P(\mathbf{Q}_{t+k} | \mathbf{o}_{1:t})$, for some $k > 0$

Probabilistic Reasoning over Time - Inference

Prediction

The task of computing the posterior distribution over the **future state**, given all evidence to date.

$P(\mathbf{Q}_{t+k} | \mathbf{o}_{1:t})$, for some $k > 0$

Smoothing (hindsight)

The task of computing the posterior distribution over a **past state**, given all evidence to the present.

$P(\mathbf{Q}_k | \mathbf{o}_{1:t})$, for some $1 \leq k < t$

Provides a better estimate of the state than was available at the time.

Probabilistic Reasoning over Time - Inference

Most likely explanation

Given a *sequence of observations*, find the **sequence of states** that is **most likely** to have generated those observations. $\operatorname{argmax}_{q_{1:t}} \mathbf{P}(\mathbf{q}_{t+k} | \mathbf{o}_{1:t})$, for some $k > 0$

Probabilistic Reasoning over Time - Inference

Most likely explanation

Given a *sequence of observations*, find the **sequence of states** that is **most likely** to have generated those observations. $\operatorname{argmax}_{q_{1:t}} \mathbf{P}(\mathbf{q}_{t+k} | \mathbf{o}_{1:t})$, for some $k > 0$

Learning

Given a set of *observation sequences*, find a method to learn the **transition** (e.g. $\mathbf{P}(\mathbf{q}_{t+1} = s_j | \mathbf{q}_t = s_i)$, $1 \leq i, j < N$) and **sensor** ($\mathbf{P}(\mathbf{o}_t | \mathbf{q}_t)$) **models** from the observations.

Probabilistic Reasoning over Time - Known Methods

Dynamic Bayesian Networks (DBN)

A DBN is Bayesian network that represents a temporal probability model.

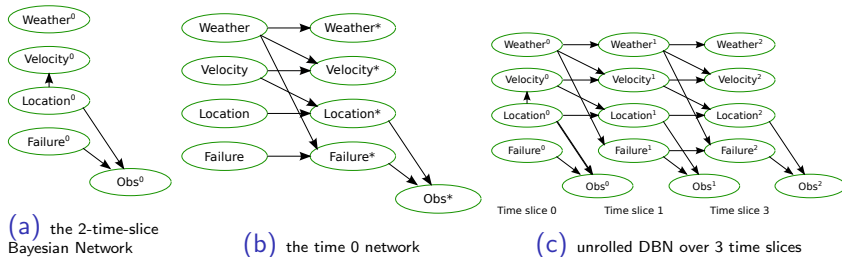


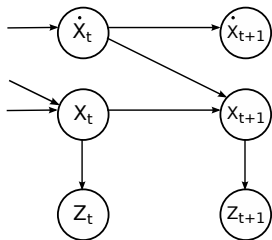
Figura: A highly simplified DBN for monitoring a vehicle [KF09]

Applied in problems like: object tracking, human activity recognition, protein sequencing etc.

Probabilistic Reasoning over Time - Known Methods

Kalman Filters (Linear Dynamical Systems)

A temporal model of one or more real-valued variables that **evolve linearly** over time, with some **Gaussian noise**.



- can be viewed as DBNs where all variables are continuous and all dependencies are linear gaussian
- wide application in **object tracking**

Figura: BN structure for a linear dynamical system with position X_t , velocity \dot{X}_t , and position measurement Z_t

Probabilistic Reasoning over Time - Known Methods

Hidden Markov Models (HMM)

An HMM is a temporal probabilistic model in which the state of the process is described by a **single discrete** random variable. The possible values of the variable are the possible states of the world.

Used successfully in applications like:

- Handwriting Recognition
- Gesture Recognition
- Speech Recognition
- Part-of-Speech Tagging
- DNA Sequencing

Outline

- 1 Aplicații în Învățarea Automată pentru MMA
 - Învățarea Automată
 - MMA în Învățarea Automată
- 2 Teoria MMA
 - Cele Trei Probleme ale MMA
 - Fundamente Matematice
- 3 Implementarea MMA
 - Problema Evaluării: Algoritmul Forward-Backward
 - Problema Interpretării: Algoritmul Viterbi
 - Problema Estimării: Algoritmul Baum-Welch
 - Learning from Observations: Baum-Welch algorithm
- 4 Demo: Recunoașterea Simbolurilor
- 5 Tipuri de MMA
- 6 Discuții și Concluzii

The 3 fundamental problems [Rab89]

- Particularization of temporal inference problems to the HMM case
- The restricted structure of the HMM allows for elegant implementations of all the basic algorithms

The 3 fundamental problems [Rab89]

- Particularization of temporal inference problems to the HMM case
- The restricted structure of the HMM allows for elegant implementations of all the basic algorithms

Evaluation Problem

Given a model and a sequence of observations, how do we compute the probability that the **observed sequence** was produced by the model?

The 3 fundamental problems [Rab89]

- Particularization of temporal inference problems to the HMM case
- The restricted structure of the HMM allows for elegant implementations of all the basic algorithms

Evaluation Problem

Given a model and a sequence of observations, how do we compute the probability that the **observed sequence** was produced by the model?

Best Explanation of Observations Problem

Given a model and a sequence of observations how do we choose a corresponding sequence of **states** which *gives meaning* to the observations?
How do we *uncover* the hidden part of the model?

The 3 fundamental problems [Rab89]

- Particularization of temporal inference problems to the HMM case
- The restricted structure of the HMM allows for elegant implementations of all the basic algorithms

Evaluation Problem

Given a model and a sequence of observations, how do we compute the probability that the **observed sequence** was produced by the model?

Best Explanation of Observations Problem

Given a model and a sequence of observations how do we choose a corresponding sequence of **states** which *gives meaning* to the observations?
How do we *uncover* the hidden part of the model?

Model Estimation (Training) Problem

Given some observed sequences, how do we adjust the **parameters** of an HMM model that best tries to explain the observations?

Outline

- 1 Aplicații în Învățarea Automată pentru MMA
 - Învățarea Automată
 - MMA în Învățarea Automată
- 2 Teoria MMA
 - Cele Trei Probleme ale MMA
 - Fundamente Matematice
- 3 Implementarea MMA
 - Problema Evaluării: Algoritmul Forward-Backward
 - Problema Interpretării: Algoritmul Viterbi
 - Problema Estimării: Algoritmul Baum-Welch
 - Learning from Observations: Baum-Welch algorithm
- 4 Demo: Recunoașterea Simbolurilor
- 5 Tipuri de MMA
- 6 Discuții și Concluzii

Exemplu: Urmărirea stărilor emoționale

Să considerăm următorul exemplu:

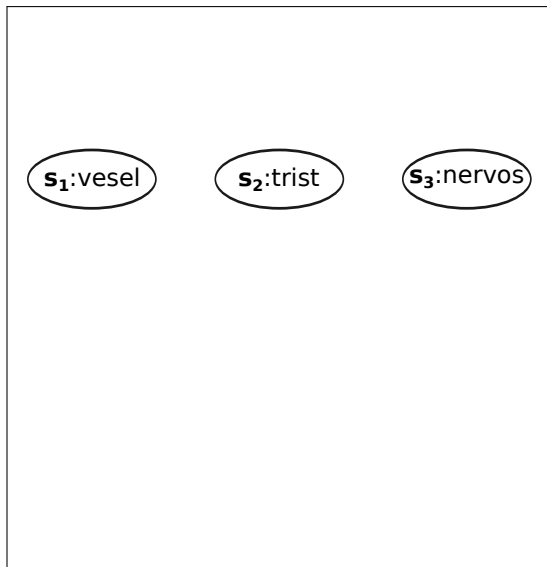
- un robot ce urmărește evoluția stărilor emoționale ale unui om

Senzor:

- cameră video

Să modelăm împreună această problemă definind componentele unui Model Markov Ascuns!

Exemplu: Urmărirea stărilor emoționale



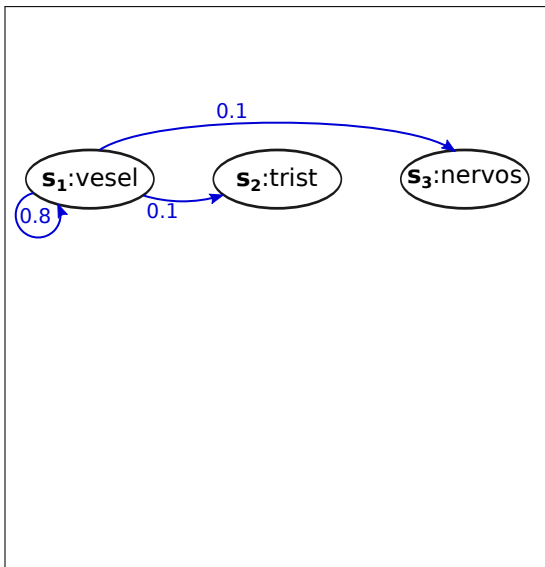
N - numărul de stări ascunse

$$N = 3$$

Stări:

- s_1 : vesel
- s_2 : trist
- s_3 : nervos

Exemplu: Urmărirea stărilor emoționale



A - matricea distribuțiilor de probabilitate ale tranzițiilor între stări

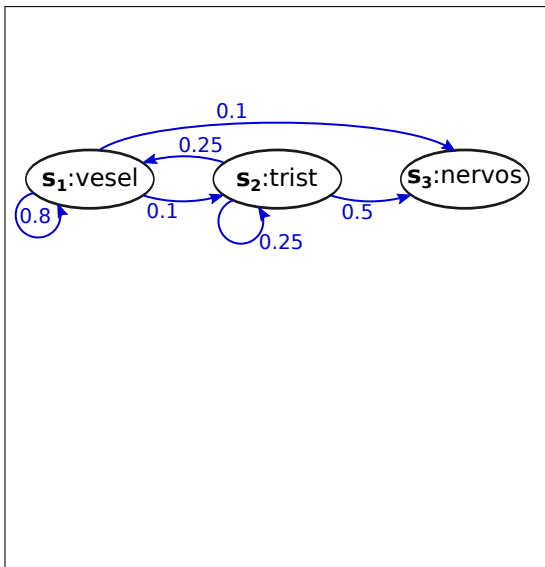
$$\mathbf{A} = \{a_{i,j}\}, \quad 1 \leq i, j \leq N$$

$$a_{i,j} = P(q_{t+1} = s_j | q_t = s_i)$$

$$\sum_{j=1}^N a_{i,j} = 1, \quad 1 \leq i \leq N$$

$$\mathbf{A} = \begin{matrix} & \begin{matrix} s_1 & s_2 & s_3 \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} & \begin{pmatrix} 0.8 & 0.1 & 0.1 \end{pmatrix} \end{matrix}$$

Exemplu: Urmărirea stărilor emoționale



A - matricea distribuțiilor de probabilitate ale tranzițiilor între stări

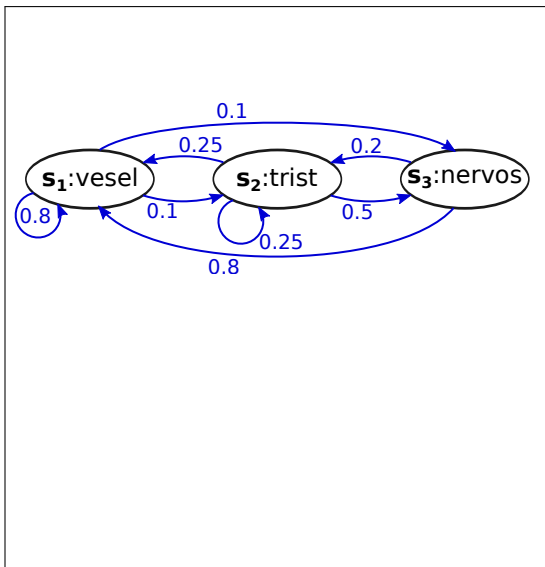
$$\mathbf{A} = \{a_{i,j}\}, \quad 1 \leq i, j \leq N$$

$$a_{i,j} = P(q_{t+1} = s_j | q_t = s_i)$$

$$\sum_{j=1}^N a_{i,j} = 1, \quad 1 \leq i \leq N$$

$$\mathbf{A} = \begin{matrix} & \begin{matrix} s_1 & s_2 & s_3 \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} & \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.25 & 0.25 & 0.5 \end{pmatrix} \end{matrix}$$

Exemplu: Urmărirea stărilor emoționale



A - matricea distribuțiilor de probabilitate ale tranzițiilor între stări

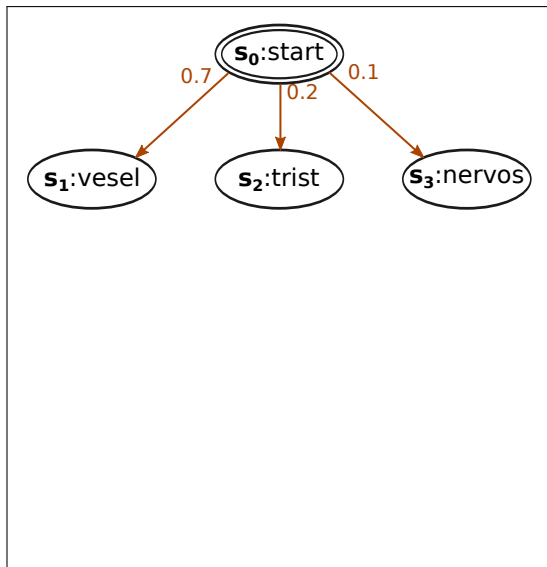
$$\mathbf{A} = \{a_{i,j}\}, \quad 1 \leq i, j \leq N$$

$$a_{i,j} = P(q_{t+1} = s_j | q_t = s_i)$$

$$\sum_{j=1}^N a_{i,j} = 1, \quad 1 \leq i \leq N$$

$$\mathbf{A} = \begin{matrix} & \begin{matrix} s_1 & s_2 & s_3 \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} & \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.25 & 0.25 & 0.5 \\ 0.8 & 0.2 & 0 \end{pmatrix} \end{matrix}$$

Exemplu: Urmărirea stărilor emoționale



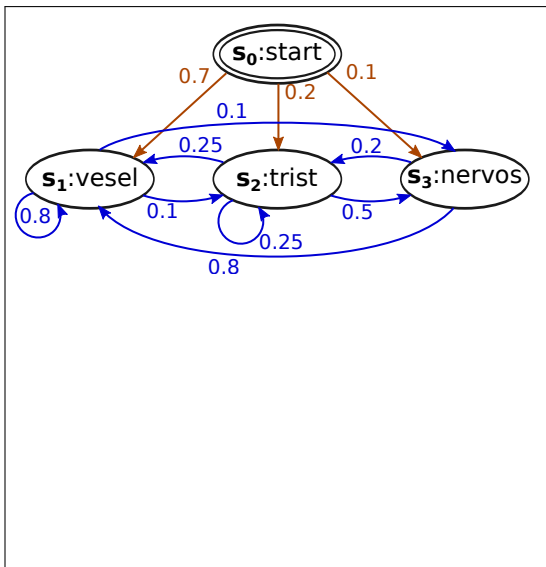
Π - distribuția stării inițiale

$$\Pi = \{\pi_i\}, \quad 1 \leq i \leq N$$

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

$$\Pi = \begin{matrix} & s_1 & s_2 & s_3 \\ \begin{pmatrix} 0.7 & 0.2 & 0.1 \end{pmatrix} \end{matrix}$$

Exemplu: Urmărirea stărilor emoționale

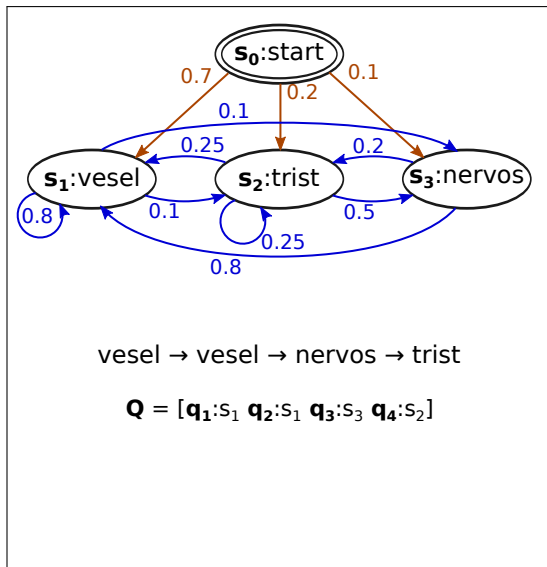


Deocamdată am descris un lanț Markov.

$$A = \begin{matrix} & s_1 & s_2 & s_3 \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} & \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.25 & 0.25 & 0.5 \\ 0.8 & 0.2 & 0 \end{pmatrix} \end{matrix}$$

$$\Pi = \begin{matrix} & s_1 & s_2 & s_3 \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} & \begin{pmatrix} 0.7 & 0.2 & 0.1 \end{pmatrix} \end{matrix}$$

Exemplu: Urmărirea stărilor emoționale



Deocamdată am descris un lanț Markov.

$$A = \begin{matrix} & s_1 & s_2 & s_3 \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} & \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.25 & 0.25 & 0.5 \\ 0.8 & 0.2 & 0 \end{pmatrix} \end{matrix}$$

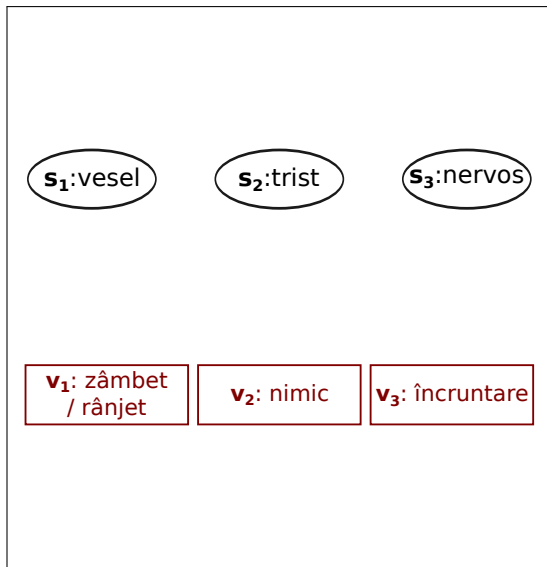
$$\Pi = \begin{matrix} & s_1 & s_2 & s_3 \\ \begin{pmatrix} 0.7 & 0.2 & 0.1 \end{pmatrix} \end{matrix}$$

Notatie: $\mathbf{Q} = [q_1 q_2 \cdots q_T]$

$$P(\mathbf{Q}|\mathbf{A}, \Pi) = \pi_{q_1} a_{q_1, q_2} \cdots a_{q_{T-1}, q_T}$$

$$\begin{aligned} P(s_1, s_1, s_3, s_2 | \mathbf{A}, \Pi) &= \pi_1 \cdot a_{1,1} \cdot a_{1,3} \cdot a_{3,2} \\ &= 0.8 \cdot 0.8 \cdot 0.1 \cdot 0.2 = 0.0128 \end{aligned}$$

Exemplu: Urmărirea stărilor emoționale



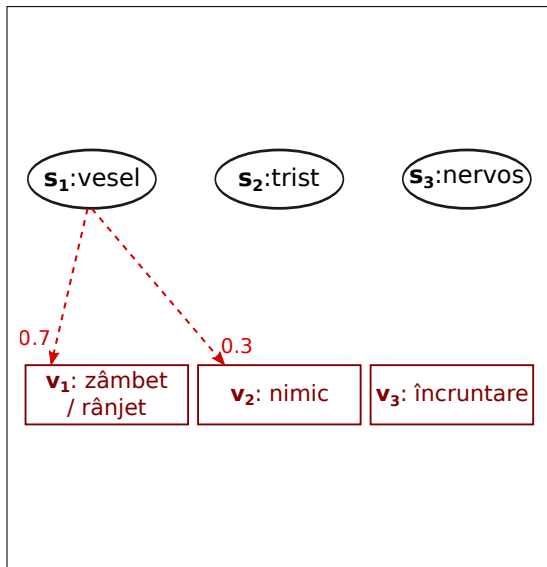
M - numărul de valori observabile distincte

$$M = 3$$

valori observabile:

- v_1 : zâmbet / rânjete
- v_2 : nimic
- v_3 : încruntare

Exemplu: Urmărirea stărilor emoționale



B - matricea distribuțiilor de probabilitate ale valorilor observabile

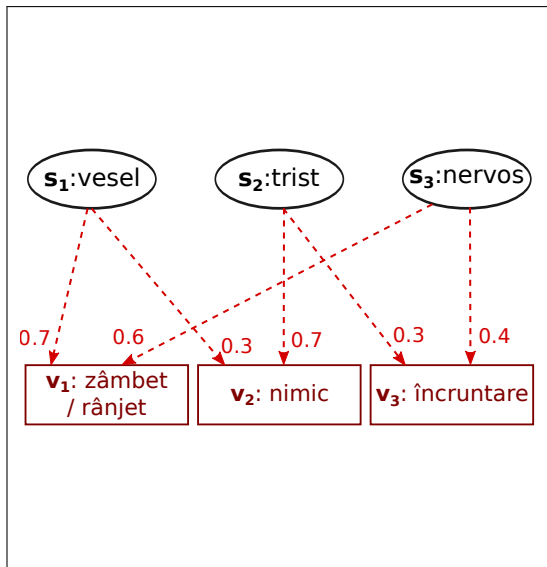
$$\mathbf{B} = \{b_{j,k}\} \quad 1 \leq j \leq N, 1 \leq k \leq M$$

$$\begin{aligned} b_{j,k} &= b_j(v_k) \\ &= P(o_t = v_k | q_t = s_j) \end{aligned}$$

$$\sum_{k=1}^M b_{j,k} = 1, \quad 1 \leq j \leq N$$

$$\mathbf{B} = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} & \begin{pmatrix} 0 & 0.2 & 0.8 \end{pmatrix} \end{matrix}$$

Exemplu: Urmărirea stărilor emoționale



B - matricea distribuțiilor de probabilitate ale valorilor observabile

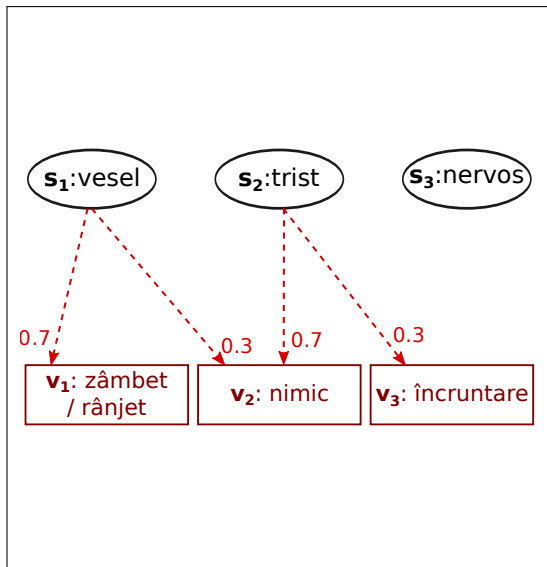
$$\mathbf{B} = \{b_{j,k}\} \quad 1 \leq j \leq N, 1 \leq k \leq M$$

$$b_{j,k} = b_j(v_k) \\ = P(o_t = v_k | q_t = s_j)$$

$$\sum_{k=1}^M b_{j,k} = 1, \quad 1 \leq j \leq N$$

$$\mathbf{B} = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} & \begin{pmatrix} 0 & 0.2 & 0.8 \\ 0.3 & 0.7 & 0 \end{pmatrix} \end{matrix}$$

Exemplu: Urmărirea stărilor emoționale



B - matricea distribuțiilor de probabilitate ale valorilor observabile

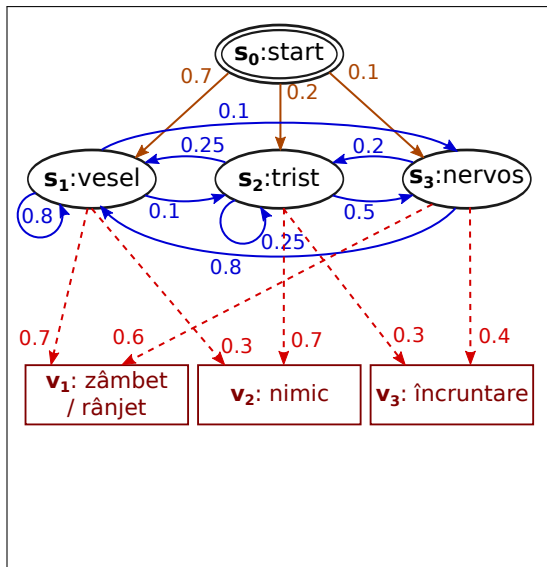
$$\mathbf{B} = \{b_{j,k}\} \quad 1 \leq j \leq N, 1 \leq k \leq M$$

$$b_{j,k} = b_j(v_k) \\ = P(o_t = v_k | q_t = s_j)$$

$$\sum_{k=1}^M b_{j,k} = 1, \quad 1 \leq j \leq N$$

$$\mathbf{B} = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} & \begin{pmatrix} 0 & 0.2 & 0.8 \\ 0.3 & 0.7 & 0 \\ 0.4 & 0 & 0.6 \end{pmatrix} \end{matrix}$$

Exemplu: Urmărirea stărilor emoționale



λ - parametrii Modelului Markov Ascuns

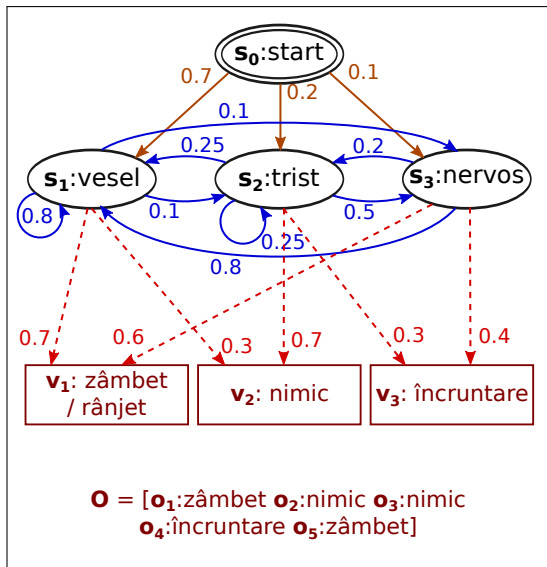
$$\lambda = (A, B, \Pi)$$

A - matricea distribuțiilor de probabilitate ale tranzițiilor între stări

B - matricea distribuțiilor de probabilitate ale valorilor observabile

Π - distribuția stării inițiale

Exemplu: Urmărirea stărilor emoționale

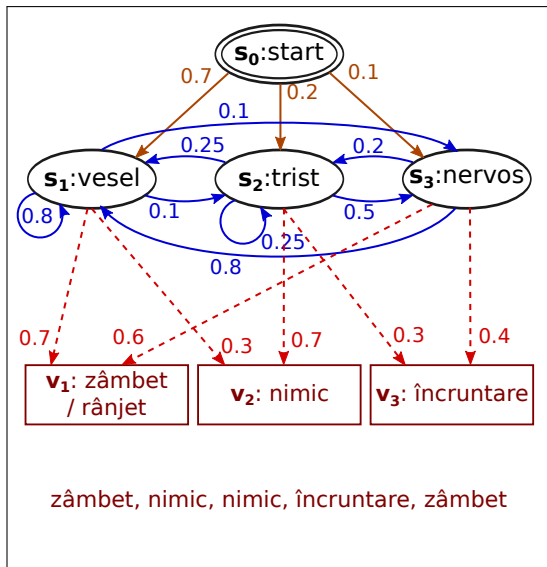


O - secvența de observații

T - lungimea secvenței de observații

$$O = [o_1 o_2 \cdots o_T]$$

Exemplu: Urmărirea stărilor emoționale

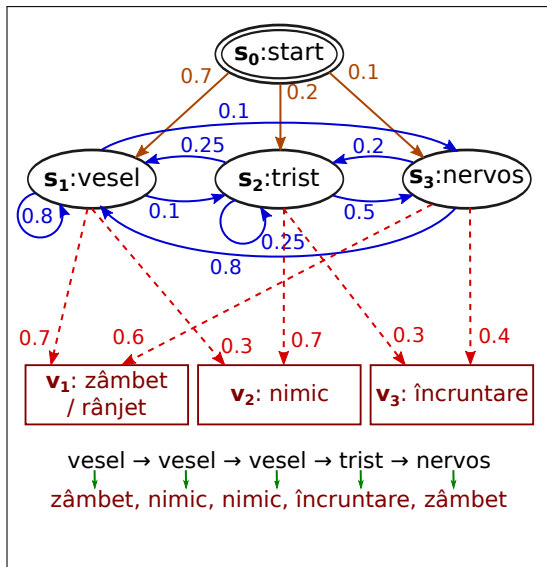


O - secvența de observații

T - lungimea secvenței de observații

$$O = [o_1 o_2 \cdots o_T]$$

Exemplu: Urmărirea stărilor emoționale

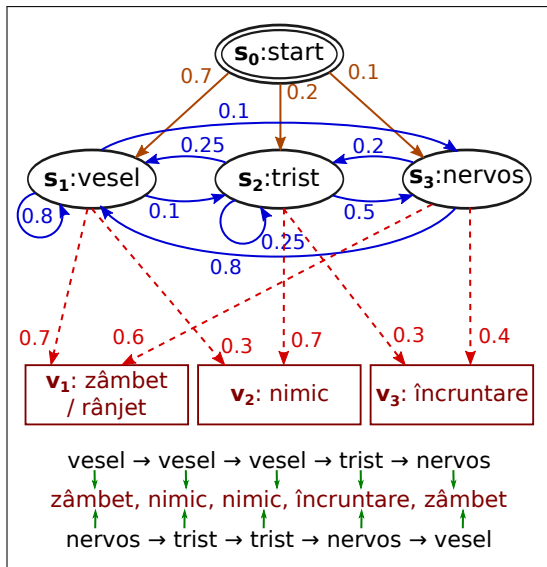


O - secvența de observații

T - lungimea secvenței de observații

$$O = [o_1 o_2 \cdots o_T]$$

Exemplu: Urmărirea stărilor emoționale



O - secvența de observații

T - lungimea secvenței de observații

$$O = [o_1 o_2 \cdots o_T]$$

Exemplu: Urmărirea stărilor emoționale

- Exemplul a fost adaptat după:

R. Zubek. Introduction to hidden markov models. *AI Game Programming Wisdom*, 3:633–646, 2006

Reformularea celor 3 probleme fundamentale ale MMA

Problema evaluării

Date fiind un model și o secvență de observații, cum calculăm probabilitatea ca secvența de observații să fi fost generată de acel model?

Reformularea celor 3 probleme fundamentale ale MMA

Problema evaluării

Date fiind un model $\lambda = (A, B, \Pi)$ și o secvență de observații O , cum calculăm probabilitatea $P(O|\lambda)$ ca secvența de observații să fi fost generată de acel model?

Reformularea celor 3 probleme fundamentale ale MMA

Problema evaluării

Date fiind un model $\lambda = (A, B, \Pi)$ și o secvență de observații $O = [o_1 o_2 \cdots o_T]$, cum calculăm probabilitatea ca secvența de observații să fi fost generată de acel model?

Reformularea celor 3 probleme fundamentale ale MMA

Problema evaluării

Date fiind un model $\lambda = (A, B, \Pi)$ și o secvență de observații $O = [o_1 o_2 \cdots o_T]$, cum calculăm probabilitatea $P(O|\lambda)$ ca secvența de observații să fi fost generată de acel model?

Reformularea celor 3 probleme fundamentale ale MMA

Problema evaluării

Date fiind un model $\lambda = (A, B, \Pi)$ și o secvență de observații $O = [o_1 o_2 \cdots o_T]$, cum calculăm probabilitatea $P(O|\lambda)$ ca secvența de observații să fi fost generată de acel model?

- Prin enumerarea tuturor secvențelor posibile de stări:

$$P(O|\lambda) = \sum_{\text{all } Q} P(O|Q, \lambda) \cdot P(Q|\lambda) \quad (1)$$

Reformularea celor 3 probleme fundamentale ale MMA

$$P(O|\lambda) = \sum_{\text{all } Q} P(O|Q, \lambda) \cdot P(Q|\lambda) \quad (1)$$

Reformularea celor 3 probleme fundamentale ale MMA

$$P(O|\lambda) = \sum_{\text{all } Q} P(O|Q, \lambda) \cdot P(Q|\lambda) \quad (1)$$

$$P(O|Q, \lambda) = \prod_{t=1}^T P(o_t|q_t, \lambda) = \prod_{t=1}^T b_{q_t}(o_t) = b_{q_1}(o_1) \cdot \dots \cdot b_{q_T}(o_T) \quad (2)$$

Reformularea celor 3 probleme fundamentale ale MMA

$$P(O|\lambda) = \sum_{\text{all } Q} P(O|Q, \lambda) \cdot P(Q|\lambda) \quad (1)$$

$$P(O|Q, \lambda) = \prod_{t=1}^T P(o_t|q_t, \lambda) = \prod_{t=1}^T b_{q_t}(o_t) = b_{q_1}(o_1) \cdot \dots \cdot b_{q_T}(o_T) \quad (2)$$

$$P(Q|\lambda) = \pi_{q_1} \prod_{t=2}^T a_{q_{t-1}, q_t} = \pi_{q_1} \cdot a_{q_1, q_2} \cdot a_{q_2, q_3} \cdot \dots \cdot a_{q_{T-1}, q_T} \quad (3)$$

Reformularea celor 3 probleme fundamentale ale MMA

$$P(O|\lambda) = \sum_{\text{all } Q} P(O|Q, \lambda) \cdot P(Q|\lambda) \quad (1)$$

$$P(O|Q, \lambda) = \prod_{t=1}^T P(o_t|q_t, \lambda) = \prod_{t=1}^T b_{q_t}(o_t) = b_{q_1}(o_1) \cdot \dots \cdot b_{q_T}(o_T) \quad (2)$$

$$P(Q|\lambda) = \pi_{q_1} \prod_{t=2}^T a_{q_{t-1}, q_t} = \pi_{q_1} \cdot a_{q_1, q_2} \cdot a_{q_2, q_3} \cdot \dots \cdot a_{q_{T-1}, q_T} \quad (3)$$

$$\begin{aligned} P(O|\lambda) &= \sum_{\text{all } Q} P(O, Q|\lambda) = \sum_{\text{all } Q} P(O, |Q, \lambda) \cdot P(Q, \lambda) \\ &= \sum_{\text{all } Q} \left(\pi_{q_1} \cdot b_{q_1}(o_1) \cdot \prod_{t=2}^T b_{q_t}(o_t) a_{q_{t-1}, q_t} \right) \end{aligned} \quad (1)$$

Reformularea celor 3 probleme fundamentale ale MMA

Problema explicării unei secvențe de observații

Date fiind un model și o secvență de observații
, cum alegem o secvență corespunzătoare de stări
care *să dea un înțeles* observațiilor? Cum *descoperim*
partea ascunsă a modelului?

Reformularea celor 3 probleme fundamentale ale MMA

Problema explicării unei secvențe de observații

Date fiind un model $\lambda = (A, B, \Pi)$ și o secvență de observații
, cum alegem o secvență corespunzătoare de stări
care *să dea un înțeles* observațiilor? Cum *descoperim*
partea ascunsă a modelului?

Reformularea celor 3 probleme fundamentale ale MMA

Problema explicării unei secvențe de observații

Date fiind un model $\lambda = (A, B, \Pi)$ și o secvență de observații

$O = [o_1 o_2 \cdots o_T]$, cum alegem o secvență corespunzătoare de stări
care să dea un înțeles observațiilor? Cum descoperim
partea ascunsă a modelului?

Reformularea celor 3 probleme fundamentale ale MMA

Problema explicării unei secvențe de observații

Date fiind un model $\lambda = (A, B, \Pi)$ și o secvență de observații

$O = [o_1 o_2 \cdots o_T]$, cum alegem o secvență corespunzătoare de stări

$Q = [q_1 q_2 \cdots q_T]$ care să dea un înțeles observațiilor? Cum descoperim partea ascunsă a modelului?

Reformularea celor 3 probleme fundamentale ale MMA

Problema explicării unei secvențe de observații

Date fiind un model $\lambda = (A, B, \Pi)$ și o secvență de observații $O = [o_1 o_2 \cdots o_T]$, cum alegem o secvență corespunzătoare de stări $Q = [q_1 q_2 \cdots q_T]$ care să dea un înțeles observațiilor? Cum descoperim partea ascunsă a modelului?

- Există mai multe criterii pentru cea mai bună secvență
 - Secvența celor mai probabile stări (luate individual):

$$Q_{\text{best}} = [\hat{q}_1 \hat{q}_2 \cdots \hat{q}_T], \quad \hat{q}_t = \underset{s_i}{\operatorname{argmax}} P(q_t = s_i | O, \lambda) \quad (4)$$

Reformularea celor 3 probleme fundamentale ale MMA

Problema explicării unei secvențe de observații

Date fiind un model $\lambda = (A, B, \Pi)$ și o secvență de observații $O = [o_1 o_2 \cdots o_T]$, cum alegem o secvență corespunzătoare de stări $Q = [q_1 q_2 \cdots q_T]$ care să dea un înțeles observațiilor? Cum descoperim partea ascunsă a modelului?

- Există mai multe criterii pentru cea mai bună secvență
 - Secvența celor mai probabile stări (luate individual):

$$Q_{\text{best}} = [\hat{q}_1 \hat{q}_2 \dots \hat{q}_T], \quad \hat{q}_t = \underset{s_i}{\operatorname{argmax}} P(q_t = s_i | O, \lambda) \quad (4)$$

- Cea mai bună cale (de dimensiune T)

$$Q_{\text{best}} = \underset{Q}{\operatorname{argmax}} P(Q | O, \lambda) = \underset{Q}{\operatorname{argmax}} P(Q, O | \lambda) \quad (5)$$

Reformularea celor 3 probleme fundamentale ale MMA

Problema Estimării Modelului (Învățării)

Date fiind niște secvențe de observații , cum *ajustăm* **parametrii** ai unui MMA astfel încât să explice cel mai bine observațiile?

Reformularea celor 3 probleme fundamentale ale MMA

Problema Estimării Modelului (Învățării)

Date fiind niște secvențe de observații $\mathcal{O} = [O_1 O_2 \cdots O_L]$, cum *ajustăm parametrii* ai unui MMA astfel încât să explice cel mai bine observațiile?

Reformularea celor 3 probleme fundamentale ale MMA

Problema Estimării Modelului (Învățării)

Date fiind niște secvențe de observații $\mathcal{O} = [O_1 O_2 \cdots O_L]$, cum *ajustăm parametrii* $\lambda = (A, B, \Pi)$ ai unui MMA astfel încât să explice cel mai bine observațiile?

Reformularea celor 3 probleme fundamentale ale MMA

Problema Estimării Modelului (Învățării)

Date fiind niște secvențe de observații $\mathcal{O} = [O_1 O_2 \cdots O_L]$, cum *ajustăm parametrii* $\lambda = (A, B, \Pi)$ ai unui MMA astfel încât să explice cel mai bine observațiile?

- Întrebarea se poate reformula matematic:

$$\lambda_{\text{best}} = \underset{\lambda}{\operatorname{argmax}} P(\mathcal{O}|\lambda) \quad (6)$$

:)

Outline

- 1 Aplicații în Învățarea Automată pentru MMA
 - Învățarea Automată
 - MMA în Învățarea Automată
- 2 Teoria MMA
 - Cele Trei Probleme ale MMA
 - Fundamente Matematice
- 3 Implementarea MMA
 - Problema Evaluării: Algoritmul Forward-Backward
 - Problema Interpretării: Algoritmul Viterbi
 - Problema Estimării: Algoritmul Baum-Welch
 - Learning from Observations: Baum-Welch algorithm
- 4 Demo: Recunoașterea Simbolurilor
- 5 Tipuri de MMA
- 6 Discuții și Concluzii

:)

Outline

- 1 Aplicații în Învățarea Automată pentru MMA
 - Învățarea Automată
 - MMA în Învățarea Automată
- 2 Teoria MMA
 - Cele Trei Probleme ale MMA
 - Fundamente Matematice
- 3 Implementarea MMA
 - Problema Evaluării: Algoritmul Forward-Backward
 - **Problema Interpretării: Algoritmul Viterbi**
 - Problema Estimării: Algoritmul Baum-Welch
 - Learning from Observations: Baum-Welch algorithm
- 4 Demo: Recunoașterea Simbolurilor
- 5 Tipuri de MMA
- 6 Discuții și Concluzii

:)

Outline

- 1 Aplicații în Învățarea Automată pentru MMA
 - Învățarea Automată
 - MMA în Învățarea Automată
- 2 Teoria MMA
 - Cele Trei Probleme ale MMA
 - Fundamente Matematice
- 3 Implementarea MMA
 - Problema Evaluării: Algoritmul Forward-Backward
 - Problema Interpretării: Algoritmul Viterbi
 - **Problema Estimării: Algoritmul Baum-Welch**
 - Learning from Observations: Baum-Welch algorithm
- 4 Demo: Recunoașterea Simbolurilor
- 5 Tipuri de MMA
- 6 Discuții și Concluzii

Outline

- 1 Aplicații în Învățarea Automată pentru MMA
 - Învățarea Automată
 - MMA în Învățarea Automată
- 2 Teoria MMA
 - Cele Trei Probleme ale MMA
 - Fundamente Matematice
- 3 Implementarea MMA
 - Problema Evaluării: Algoritmul Forward-Backward
 - Problema Interpretării: Algoritmul Viterbi
 - Problema Estimării: Algoritmul Baum-Welch
 - Learning from Observations: Baum-Welch algorithm
- 4 Demo: Recunoașterea Simbolurilor
- 5 Tipuri de MMA
- 6 Discuții și Concluzii

Learning from observations - Reminder

Model Estimation (Training) Problem

Given some observed sequences, how do we adjust the **parameters** of an HMM model that best tries to explain the observations?

Learning from observations - Reminder

Model Estimation (Training) Problem

Given some observed sequences, how do we adjust the **parameters** of an HMM model that best tries to explain the observations?

Adjust the model parameters $\lambda = (A, B, \Pi)$ to obtain $\max_{\lambda} P(O|\lambda)$

Learning from observations - Reminder

Model Estimation (Training) Problem

Given some observed sequences, how do we adjust the **parameters** of an HMM model that best tries to explain the observations?

Adjust the model parameters $\lambda = (A, B, \Pi)$ to obtain $\max_{\lambda} P(O|\lambda)$

The observation sequence used to adjust the model parameters is called a **training** sequence.

Training problem is crucial - allows to create best models for real phenomena.

Learning from observations - Aspects of the approach

Learning from observations - Aspects of the approach

Problem

There is no known way to analytically solve for the model which maximizes the probability of the observation sequence.

Learning from observations - Aspects of the approach

Problem

There is no known way to analytically solve for the model which maximizes the probability of the observation sequence.

Solution

We can choose $\lambda = (A, B, \Pi)$ such that $\max_{\lambda} P(O|\lambda)$ is **locally maximized** using an **iterative procedure** such as *Baum-Welch*.

The method is an instance of the *EM algorithm* [DLR77] for HMMs.

Baum-Welch algorithm (I)

Baum-Welch algorithm (I)

We first define some auxiliary variables:

$$\xi_{t,i,j} = \xi_t(i,j) = P(q_t = s_i, q_{t+1} = s_j | O, \lambda)$$

The probability of being in state s_i at time t and in state s_j at time $t + 1$, given the model and the observation sequence.

Baum-Welch algorithm (I)

We first define some auxiliary variables:

$$\xi_{t,i,j} = \xi_t(i,j) = P(q_t = s_i, q_{t+1} = s_j | O, \lambda)$$

The probability of being in state s_i at time t and in state s_j at time $t + 1$, given the model and the observation sequence.

$$\gamma_{t,i} = \gamma_t(i) = P(q_t = s_i | O, \lambda)$$

The probability of being in state s_i at time t , given the model and the observation sequence.

Baum-Welch algorithm (I)

We first define some auxiliary variables:

$$\xi_{t,i,j} = \xi_t(i,j) = P(q_t = s_i, q_{t+1} = s_j | O, \lambda)$$

The probability of being in state s_i at time t and in state s_j at time $t + 1$, given the model and the observation sequence.

$$\gamma_{t,i} = \gamma_t(i) = P(q_t = s_i | O, \lambda)$$

The probability of being in state s_i at time t , given the model and the observation sequence.

From the definitions it follows that:

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

Baum-Welch algorithm (II)

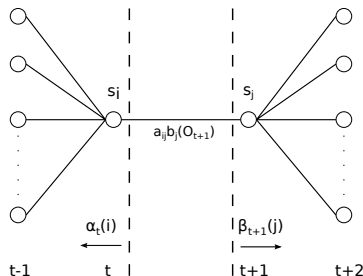


Figura: Sequence of operations required for the computation of the joint event that the system is in state S_i at time t and state S_j at time $t+1$ [Rab89]

$$\alpha_{t,i} = P(o_1, o_2, \dots, o_t, q_t = S_i | \lambda)$$

$$\beta_{t,i} = P(o_{t+1} o_{t+2} \dots o_T | q_t = S_i, \lambda)$$

$$\begin{aligned} \xi_t(i, j) &= \frac{\alpha_{t,i} \cdot a_{i,j} \cdot b_j(o_{t+1}) \cdot \beta_{t+1,j}}{P(O | \lambda)} \\ &= \frac{\alpha_{t,i} \cdot a_{i,j} \cdot b_j(o_{t+1}) \cdot \beta_{t+1,j}}{\sum_{k=1}^N \sum_{l=1}^N \alpha_{t,k} \cdot a_{k,l} \cdot b_l(o_{t+1}) \cdot \beta_{t+1,l}} \end{aligned}$$

Baum-Welch algorithm (III)

How do these auxiliary variables help?

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{expected number of transitions from } S_i$$

$$\sum_{t=1}^{T-1} \xi_t(i, j) = \text{expected number of transitions from } S_i \text{ to } S_j$$

Baum-Welch algorithm (IV)

$\bar{\pi}_i =$ expected no. of times in state S_i at time $(t = 1) = \gamma_t(i)$

Baum-Welch algorithm (IV)

$\bar{\pi}_i =$ expected no. of times in state S_i at time $(t = 1) = \gamma_t(i)$

$$\begin{aligned} a_{i,j} &= \frac{\text{expected no. of transitions from } S_i \text{ to } S_j}{\text{expected no. of transition from } S_i} \\ &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \end{aligned}$$

Baum-Welch algorithm (IV)

$\bar{\pi}_i =$ expected no. of times in state S_i at time $(t = 1) = \gamma_t(i)$

$$\begin{aligned} a_{i,j}^- &= \frac{\text{expected no. of transitions from } S_i \text{ to } S_j}{\text{expected no. of transition from } S_i} \\ &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \end{aligned}$$

$$\begin{aligned} b_{j,k}^- &= \frac{\text{expected no. of times in } S_j \text{ observing symbol } v_k}{\text{expected no. of times in } S_j} \\ &= \frac{\sum_{t=1, O_t=v_k}^T \gamma_t(j)}{T} \end{aligned}$$

Baum-Welch algorithm (V)

The routine for the general case:

Baum-Welch - Let's write some code

LET'S WRITE SOME CODE :-)

A simple symbol recognition application

Features:

A simple symbol recognition application

Features:

Define

Define, organize and visualize a dataset of symbols defined with mouse movements.

A simple symbol recognition application

Features:

Define

Define, organize and visualize a dataset of symbols defined with mouse movements.

Train

Train a HMM-based recognition engine on a symbol dataset.

A simple symbol recognition application

Features:

Define

Define, organize and visualize a dataset of symbols defined with mouse movements.

Train

Train a HMM-based recognition engine on a symbol dataset.

Recognize

Recognize new symbols and view classification metrics.

A simple symbol recognition application

Features:

Define

Define, organize and visualize a dataset of symbols defined with mouse movements.

Train

Train a HMM-based recognition engine on a symbol dataset.

Recognize

Recognize new symbols and view classification metrics.

Default included symbols: **left arrow, right arrow, circle, square, infinity**

A simple symbol recognition application - A View

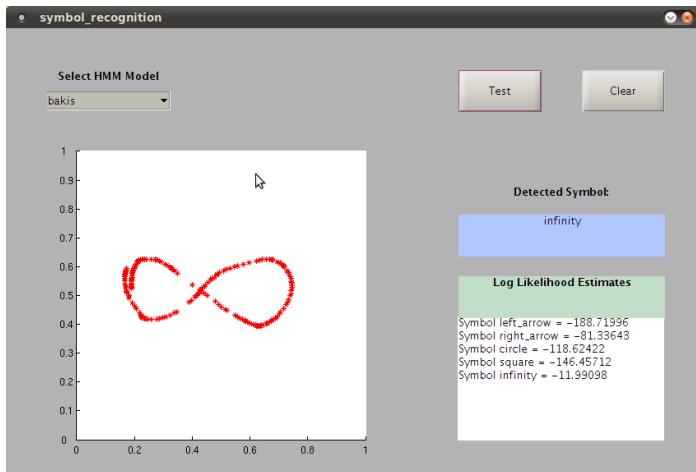
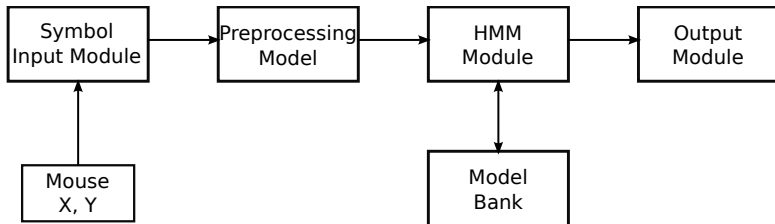


Figura: A view of the symbol recognition application GUI

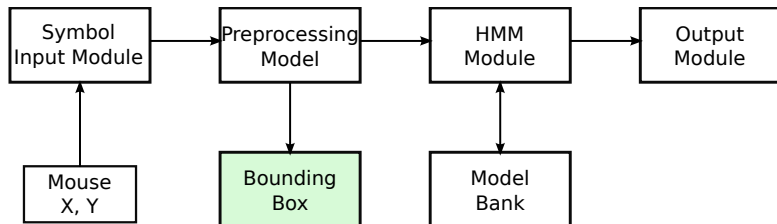
A simple symbol recognition application - Approach (I)

Adapted from [YX94].



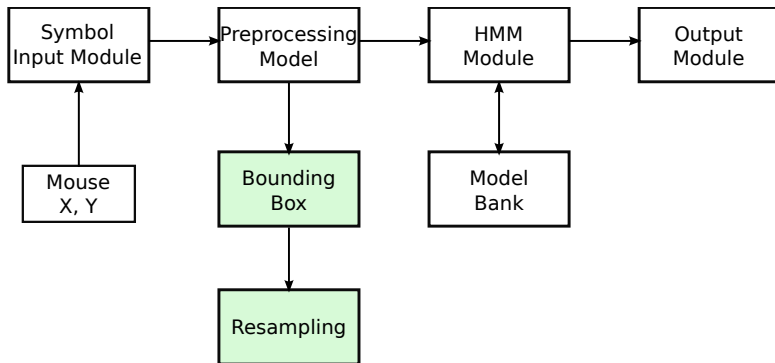
A simple symbol recognition application - Approach (I)

Adapted from [YX94].



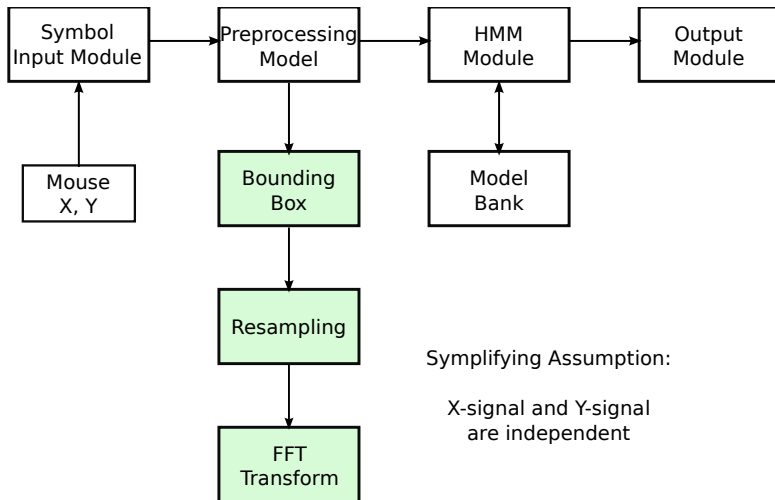
A simple symbol recognition application - Approach (I)

Adapted from [YX94].



A simple symbol recognition application - Approach (I)

Adapted from [YX94].



Simplifying Assumption:

X-signal and Y-signal
are independent

A simple symbol recognition application - Approach (II)

Adapted from [YX94].

HMM Structure

$N(\text{number of states}) = 8$

2 discrete observable variables per state - $coef_{FFT}(x)$, $coef_{FFT}(y)$

$M(\text{number of values for each observable variable}) = 256$

Transition model:

- Bakis
- Ergodic

A simple symbol recognition application - Results

Dataset size

5 symbols: left arrow, right arrow, circle, square, infinity

100 samples per symbol: 50 training, 10 validation, 40 testing

```
>> symbol_performance_test('ergodic')
===== Testing trained HMM models =====
## Results for the model of symbol "left_arrow":
Accuracy: 0.97500
Precision: 1.00000
Recall: 0.97500
Confusion matrix line: 39 0 1 0 0 0

## Results for the model of symbol "right_arrow":
Accuracy: 1.00000
Precision: 1.00000
Recall: 1.00000
Confusion matrix line: 0 40 0 0 0 0

## Results for the model of symbol "circle":
Accuracy: 0.90244
Precision: 0.97368
Recall: 0.92500
Confusion matrix line: 0 0 37 2 1 0

## Results for the model of symbol "square":
Accuracy: 0.95238
Precision: 0.95238
Recall: 1.00000
Confusion matrix line: 0 0 0 40 0 0

## Results for the model of symbol "infinity":
Accuracy: 0.97561
Precision: 0.97561
Recall: 1.00000
Confusion matrix line: 0 0 0 0 40 0
```

```
>> symbol_performance_test('bakis')
===== Testing trained HMM models =====
## Results for the model of symbol "left_arrow":
Accuracy: 0.90000
Precision: 1.00000
Recall: 0.90000
Confusion matrix line: 36 0 1 0 0 3

## Results for the model of symbol "right_arrow":
Accuracy: 1.00000
Precision: 1.00000
Recall: 1.00000
Confusion matrix line: 0 40 0 0 0 0

## Results for the model of symbol "circle":
Accuracy: 0.97561
Precision: 0.97561
Recall: 1.00000
Confusion matrix line: 0 0 40 0 0 0

## Results for the model of symbol "square":
Accuracy: 0.97500
Precision: 1.00000
Recall: 0.97500
Confusion matrix line: 0 0 0 39 0 1

## Results for the model of symbol "infinity":
Accuracy: 1.00000
Precision: 1.00000
Recall: 1.00000
Confusion matrix line: 0 0 0 0 40 0
```

:)

- [DLR77] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.
- [KF09] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
- [Rab89] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [YX94] J. Yang and Y. Xu. Hidden markov model for gesture recognition. Technical report, DTIC Document, 1994.
- [Zub06] R. Zubek. Introduction to hidden markov models. *AI Game Programming Wisdom*, 3:633–646, 2006.

Thank you!

Baftă, șailor!