

# Modele Markov Ascunse

## De la Teorie la Aplicații

Alexandru Sorici, Tudor Berariu

Asociația Română pentru Inteligență Artificială  
în colaborare cu  
Laboratorul AI-MAS

10 Noiembrie 2012



# Cuprins

## 1 Aplicații în Învățarea Automată pentru MMA

- Învățarea Automată
- MMA în Învățarea Automată

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

# Cuprins

- 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

# Cuprins

- 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
- 4 Demo: Recunoașterea Simbolurilor

# Cuprins

- 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
- 4 Demo: Recunoașterea Simbolurilor
- 5 Tipuri de MMA

# Cuprins

- 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
- 4 Demo: Recunoașterea Simbolurilor
- 5 Tipuri de MMA
- 6 Discuții și Concluzii



# Cuprins

- 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
- 4 Demo: Recunoașterea Simbolurilor
- 5 Tipuri de MMA
- 6 Discuții și Concluzii



# Unde suntem...

## 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

## 4 Demo: Recunoașterea Simbolurilor

## 5 Tipuri de MMA

## 6 Discuții și Concluzii



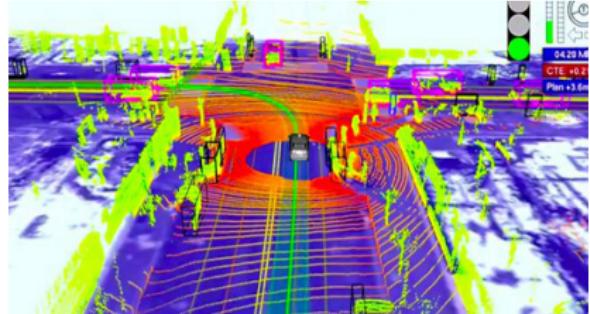
# Ce este Învățarea Automată?

## Învățarea automată

Un program se spune că învăță dintr-o mulțime de experiențe  $E$  relativ la o clasă de task-uri  $T$  și o măsură a performanței  $P$  dacă performanța acestuia la execuția task-urilor  $T$ , măsurată de  $P$ , se îmbunătățește în urma experiențelor  $E$ . [Mit97]

# Aplicații ale Învățării Automate

- Self-Driving Car: Google Car
- Traducere Automată: Google Translate
- Sisteme de Recomandare
  - Filme: ImDB, NetFlix
  - Publicitate Inteligentă: Google Ads, Facebook Ads



imagini de pe [www.nytimes.com](http://www.nytimes.com)



# Aplicații ale Învățării Automate

- Self-Driving Car: Google Car
- Traducere Automată: Google Translate
- Sisteme de Recomandare
  - Filme: ImDB, NetFlix
  - Publicitate Inteligentă: Google Ads, Facebook Ads

The screenshot shows a translation interface between Romanian and English. On the left, the Romanian text "Bine ați venit la atelierul de lucru!" is displayed with a "Romanian - detected" label above it. On the right, the English translation "Welcome to the workshop!" is shown. Both panels have language selection buttons at the top: "Romanian", "English", and "Spanish". Below the text, each panel has a set of icons: a speech bubble, a speaker icon, and a checkmark. A note at the bottom says: "New! Hold down the shift key, click, and drag the words above to reorder. [Dismiss](#)".



# Aplicații ale Învățării Automate

- Self-Driving Car: Google Car
- Traducere Automată: Google Translate
- Sisteme de Recomandare
  - Filme: ImDB, NetFlix
  - Publicitate Intelligentă: Google Ads, Facebook Ads

**Recommended for you**

[Learn more](#)

**Vicky Cristina Barcelona** (2008)

PG-13 Drama | Romance

★★★★★ ★★★★★ 7.2/10

Two girlfriends on a summer holiday in Spain become enamored with the same painter, unaware that his ex-wife, with whom he has a tempestuous relationship, is about to re-enter the picture.

Director: Woody Allen  
Stars: Rebecca Hall and Scarlett Johans...

Add to Watchlist

◀ Prev 6 Next ▶

No Next

Recommended because of your interest in Gegen die Wand and Closer.



# Aplicații ale Învățării Automate

- Self-Driving Car: Google Car
- Traducere Automată: Google Translate
- Sisteme de Recomandare
  - Filme: ImDB, NetFlix
  - Publicitate Inteligentă: Google Ads, Facebook Ads

A screenshot of a Google search results page. The search bar at the top contains the query "gripa". Below the search bar, the word "Search" is followed by "About 7,250,000 results (0.20 seconds)". On the left, there is a sidebar with links for "Web", "Images", "Videos", "News", and "More". The "Web" section shows a search result for "Ad related to gripa" with a link to "Adio raceala – Tratament natural impotriva racelii" from "www.daciaplanit.ro/imunitate". Below this, there is a tip about specifying search language in preferences. On the right, there is a snippet for "Gripa" with a link to "www.sfatulmedicului.ro/gripa". The snippet describes gripa as a acute respiratory infection caused by the influenza virus A or B, appearing in epidemic outbreaks.

Google

gripa

Search About 7,250,000 results (0.20 seconds)

Web Ad related to gripa ⓘ  
[Adio raceala – Tratament natural impotriva racelii](#)  
www.daciaplanit.ro/imunitate  
Fara efecte secundare.

Images

Videos

News Tip: Search for English (UK) results only. You can specify your search language in Preferences

More

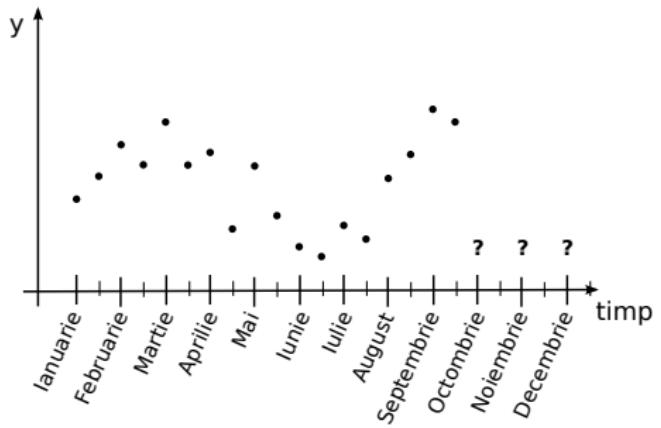
Gripa  
[www.sfatulmedicului.ro/gripa](#) · Translate this page  
Gripa - Gripa este o boala infecțioasă acută a cailor respiratorii, foarte contagioasă, cauzată de virusul gripal A sau B, care apare în izbucniri epidemice de ...

Bucharest Change location

# Învățarea Automată: Tipuri de probleme

## Tipuri de probleme

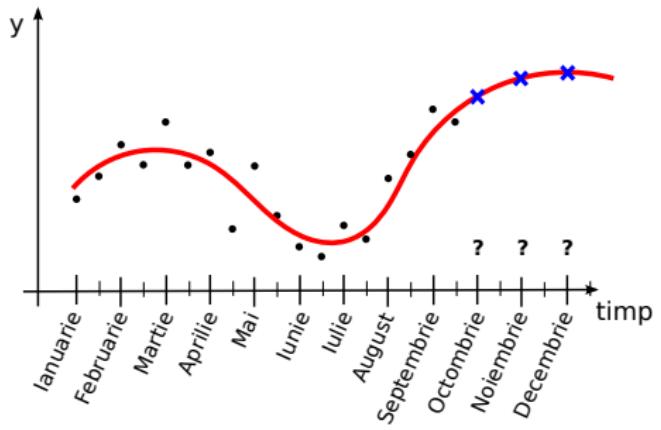
- Regresie
  - predicția evoluției prețului unui bun
- Clasificare
  - clasificarea obiectelor dintr-o imagine



# Învățarea Automată: Tipuri de probleme

## Tipuri de probleme

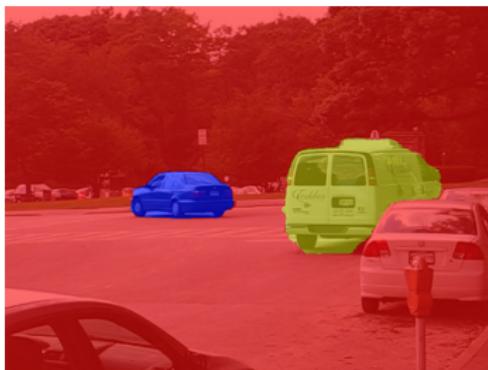
- Regresie
  - predicția evoluției prețului unui bun
- Clasificare
  - clasificarea obiectelor dintr-o imagine



# Învățarea Automată: Tipuri de probleme

## Tipuri de probleme

- Regresie
  - predictia evolutiei pretului unui bun
- Clasificare
  - clasificarea obiectelor dintr-o imagine



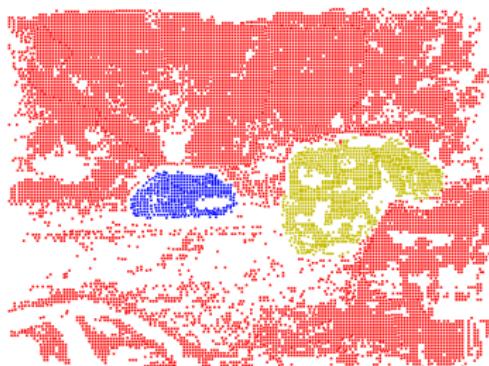
Imagine de la Albert-Ludwigs-Universität,

Lehrstuhl für Mustererkennung und Bildverarbeitung

# Învățarea Automată: Tipuri de probleme

## Tipuri de probleme

- Regresie
  - predictia evolutiei pretului unui bun
- Clasificare
  - clasificarea obiectelor dintr-o imagine



Imagine de la Albert-Ludwigs-Universität,

Lehrstuhl für Mustererkennung und Bildverarbeitung



# Tipuri de Învățare

## Învățare Supervizată

Învățarea **supervizată** se face pe baza unor date de antrenare etichetate.

- construirea unui model climatic pe baza datelor din ultimii 30 de ani
- construirea unui detector de mesaje spam



# Tipuri de Învățare

## Învățare Supervizată

Învățarea **supervizată** se face pe baza unor date de antrenare etichetate.

- construirea unui model climatic pe baza datelor din ultimii 30 de ani
- construirea unui detector de mesaje spam

## Învățare Nesupervizată

În **învățarea nesupervizată** nu există date etichetate.

- detectarea profilurilor de comportament ale cumpărătorilor



# Tipuri de Învățare

## Învățare Supervizată

Învățarea **supervizată** se face pe baza unor date de antrenare etichetate.

- construirea unui model climatic pe baza datelor din ultimii 30 de ani
- construirea unui detector de mesaje spam

## Învățare Nesupervizată

În **învățarea nesupervizată** nu există date etichetate.

- detectarea profilurilor de comportament ale cumpărătorilor

## Învățare prin Recompensă

În **învățarea prin recompensă** un *agent* caută să execute acele acțiuni care îi aduc o recompensă mai mare din partea mediului.

- inteligența artificială pentru jocuri



# Unde suntem...

## 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

## 4 Demo: Recunoașterea Simbolurilor

## 5 Tipuri de MMA

## 6 Discuții și Concluzii



# Probleme cu Secvențe Temporale (I)

## URMĂRIREA OBIECTELOR

Urmărirea  
Vorbitorului

GPS

Rachete  
Sol-Aer

Sistem de  
Navigare

## RECUNOAȘTEREA VORBIRII

Interfețe  
utilizator vocale

Procesare speech-to-text

Comandă Vocală

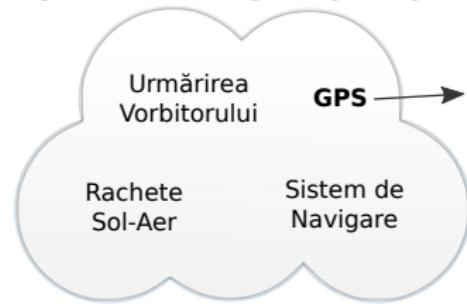
Recunoașterea  
Personalizată a  
Semnăturilor

Recunoașterea  
Limbajului Semnelor

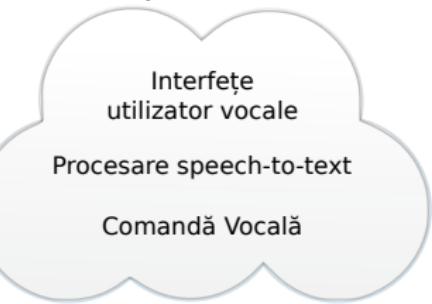
## RECUNOAȘTEREA GESTURILOR

# Probleme cu Secvențe Temporale (I)

## URMĂRIREA OBIECTELOR



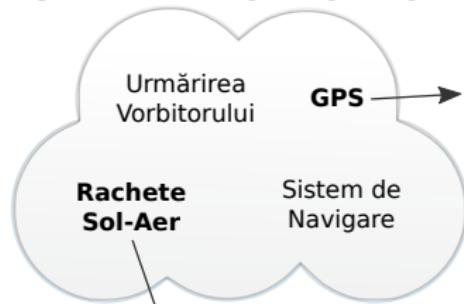
## RECUNOAȘTEREA VORBIRII



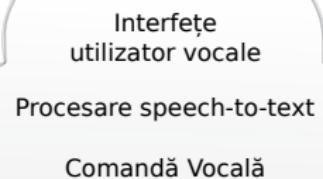
## RECUNOAȘTEREA GESTURIILOR

# Probleme cu Secvențe Temporale (I)

## URMĂRIREA OBIECTELOR



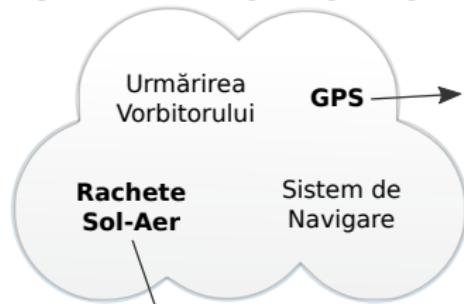
## RECUNOAȘTEREA VORBIRII



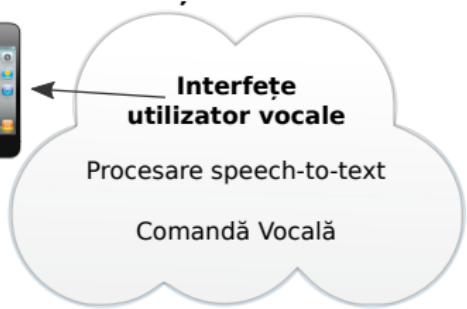
## RECUNOAȘTEREA GESTURILOR

# Probleme cu Secvențe Temporale (I)

## URMĂRIREA OBIECTELOR



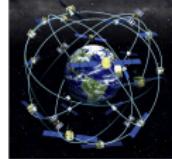
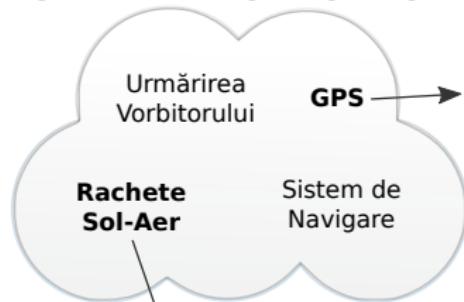
## RECUNOAȘTEREA VORBIRII



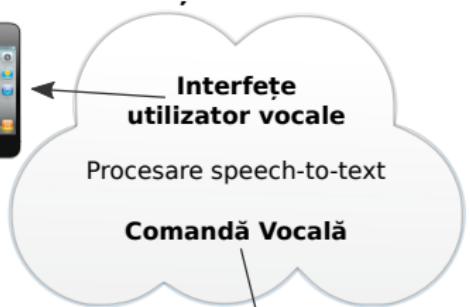
## RECUNOAȘTEREA GESTURILOR

# Probleme cu Secvențe Temporale (I)

## URMĂRIREA OBIECTELOR



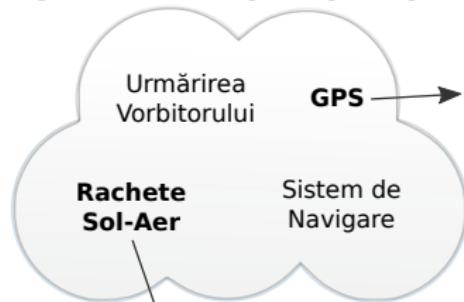
## RECUNOAȘTEREA VORBIRII



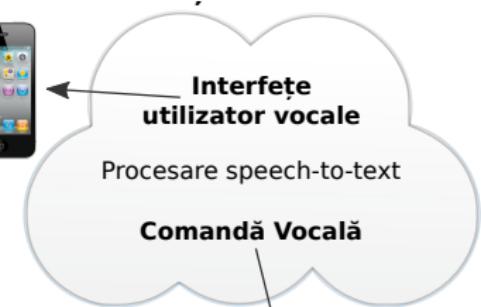
## RECUNOAȘTEREA GESTURILOR

# Probleme cu Secvențe Temporale (I)

## URMĂRIREA OBIECTELOR



## RECUNOAȘTEREA VORBIRII



RECUNOAȘTEREA GESTURIILOR





# Probleme cu Secvențe Temporale (II)

## BIOINFORMATICĂ

Secvențierea  
Proteinelor

Modelarea unei  
Rețele Regulatoare

## ECONOMIE

Predictia  
Valorilor

Econometrie  
- estimarea în timp a indicatorilor economici ai unei țări -

# Probleme cu Secvențe Temporale (II)

## BIOINFORMATICĂ

Secvențierea  
Proteinelor

Modelarea unei  
Rețele Regulatoare



## ECONOMIE

Predictia  
Valorilor

Econometrie  
- estimarea în timp a indicatorilor economici ai unei țări -



# Ratiōnament Probabilistic Temporal - Modele

Să ne gândim la unele din problemele anterioare ...



# Ratiōnament Probabilistic Temporal - Modele

Să ne gândim la unele din problemele anterioare ...

Cum modelăm astfel de situații dinamice?

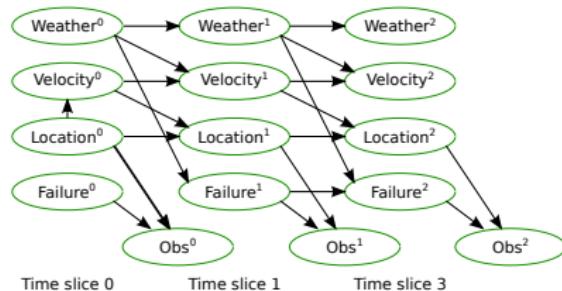
# Rationament Probabilistic Temporal - Modele

Să ne gândim la unele din problemele anterioare ...

Cum modelăm astfel de situații dinamice?

## Stări și Observații

- Procesul de schimbare este văzut ca o serie de **snapshoturi**
- Fiecare snapshot conține un set de variabile aleatoare
  - $\mathbf{O}_t$  - setul tuturor variabilelor de măsurare (*observable*) la momentul  $t$
  - $\mathbf{Q}_t$  - setul tuturor variabilelor de stare (*neobservable / ascunse*) la momentul  $t$



Exemplu: problemă de localizare a unui vehicul [KF09]



# Rationament Probabilistic Temporal - Presupuneri

Să ne gândim la unele din problemele anterioare ...



# Ratiōnament Probabilistic Temporal - Presupunerī

Să ne gândim la unele din problemele anterioare ...

Ce **presupunerī** facem?



# Rationament Probabilistic Temporal - Presupuneri

Să ne gândim la unele din problemele anterioare ...

Ce **presupuneri** facem?

## Proces staționar

Procesul de schimbare este guvernat de legi care **nu se schimbă in timp**.

**Urmare:** trebuie să specificăm relațiile între variabile doar pentru un snapshot *reprezentativ*.



# Rationament Probabilistic Temporal - Presupuneri

Să ne gândim la unele din problemele anterioare ...

Ce **presupuneri** facem?

## Proces staționar

Procesul de schimbare este guvernat de legi care **nu se schimbă in timp**.

**Urmare:** trebuie să specificăm relațiile între variabile doar pentru un snapshot *reprezentativ*.

## Presupunerea Markov

Starea curentă a unui proces de schimbare depinde doar de o **istorie finită** de stări anterioare.

**Urmare:** avem un număr **limitat** de “părinti” pentru variabilele din fiecare snapshot.



# Rationament Probabilistic Temporal - Inferență

Care sunt principalele inferențe ce se doresc făcute?



# Raționament Probabilistic Temporal - Inferență

Care sunt principalele inferențe ce se doresc făcute?

## Filtrare (monitorizare)

Sarcina de a calcula **starea de fapt** - distribuția posterioară de probabilitate a **stării curente**, date fiind toate observațiile de până acum.



# Ratiونament Probabilistic Temporal - Inferență

Care sunt principalele inferențe ce se doresc făcute?

## Filtrare (monitorizare)

Sarcina de a calcula **starea de fapt** - distribuția posterioară de probabilitate a **stării curente**, date fiind toate observațiile de până acum.

## Evaluare

Sarcina de a calcula **probabilitatea (likelihood)** a observațiilor făcute până în prezent.



# Ratiونament Probabilistic Temporal - Inferență

Care sunt principalele inferențe ce se doresc făcute?

## Predictie

Sarcina de a calcula distribuția posterioară de probabilitate peste o **stare viitoare**, date fiind toate observațiile de până acum.



# Ratiونament Probabilistic Temporal - Inferență

Care sunt principalele inferențe ce se doresc făcute?

## Predictie

Sarcina de a calcula distribuția posterioară de probabilitate peste o **stare viitoare**, date fiind toate observațiile de până acum.

## Netezire (hindsight)

Sarcina de a calcula distribuția posterioară de probabilitate peste o **stare anterioară**, date fiind toate observațiile de până acum.

Furnizează o estimare mai bună asupra stării respective, decât a fost posibil la momentul respectiv.



# Raționament Probabilistic Temporal - Inferență

Care sunt principalele inferențe ce se doresc făcute?

Cea mai probabilă explicație

Dându-se o secvență de observații, se cere găsirea celei mai probabile secvențe de stări care a generat acele observații.



# Raționament Probabilistic Temporal - Inferență

Care sunt principalele inferențe ce se doresc făcute?

Cea mai probabilă explicație

Dându-se o secvență de observații, se cere găsirea celei mai probabile secvențe de stări care a generat acele observații.

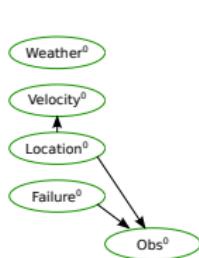
Învățare

Dându-se un set de secvențe de observații, găsește o metodă de a învăța modelele de tranzitie și senzoriale / de măsurare pe baza acestor observații.

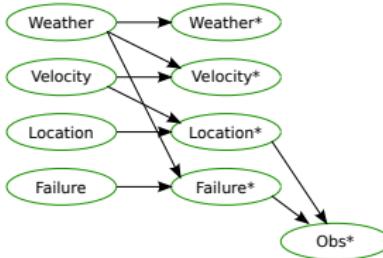
# Raționament Probabilistic Temporal - Metode

## Rețele Bayesiene Dinamice (RBD)

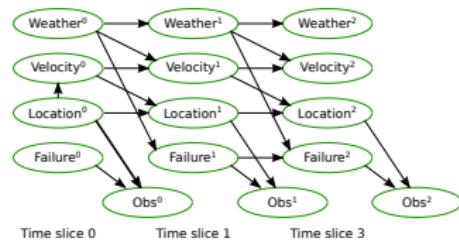
O RBD este o rețea Bayesiană ce reprezintă un model temporal de probabilitate.



(a) Rețeaua la momentul 0



(b) Rețeaua Bayesiană în două snapshot-uri



(c) RBD desfășurată pe 3 momente de timp

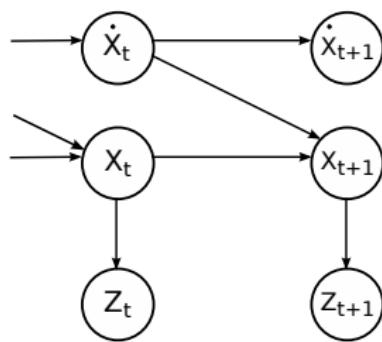
RBD simplificată pentru monitorizarea unui vehicul [KF09]

Aplicată în probleme precum: urmărirea obiectelor, recunoașterea activității umane, sevențierea proteinelor, etc.

# Rationament Probabilistic Temporal - Metode

## Filtre Kalman (Sistem Dinamice Lineare)

Un model temporal având una sau mai multe variabile care evoluează linear în timp, la care se adaugă **zgomot Gaussian**.



- Poate fi văzut ca o RBD în care toate variabilele sunt continue, iar dependențele sunt linear gaussiane.
- Aplicații multiple în **urmărirea obiectelor**

Structura unei RB pentru un sistem linear dinamic cu variabile de poziție  $X_t$ , viteză  $\dot{X}_t$ , și măsurare a poziției  $Z_t$



# Raționament Probabilistic Temporal - Metode

## Modele Markov Ascunse (MMA)

Un MMA (HMM) este un model probabilistic temporal în care *starea* procesului de schimbare este descrisă de **o singură variabilă aleatoare discretă**. Valorile posibile ale variabilei reprezintă stările posibile ale lumii modelate.

Utilizat cu succes în aplicații precum:

- Recunoașterea Scrisului
- Recunoașterea Gesturilor
- Recunoașterea Vorbirii
- Determinarea Părților de Vorbire (Part-of-Speech Tagging)
- Secvențiere ADN



# Cuprins

## 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

## 4 Demo: Recunoașterea Simbolurilor

## 5 Tipuri de MMA

## 6 Discuții și Concluzii



# Unde suntem...

## 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

## 4 Demo: Recunoașterea Simbolurilor

## 5 Tipuri de MMA

## 6 Discuții și Concluzii



# Cele 3 Probleme ale MMA

3 probleme fundamentale [Rab89]

- Particularizarea inferenței în probleme cu secvențe temporale pe cazul MMA
- Structura restricționată a MMA permite implementări elegante ale tuturor algoritmilor de bază



# Cele 3 Probleme ale MMA

3 probleme fundamentale [Rab89]

- Particularizarea inferenței în probleme cu secvențe temporale pe cazul MMA
- Structura restricționată a MMA permite implementări elegante ale tuturor algoritmilor de bază

## Problema Evaluării

Dându-se un model și o secvență de observații, cum calculăm probabilitatea ca **secvența observată** să fi fost produsă de model?



# Cele 3 Probleme ale MMA

3 probleme fundamentale [Rab89]

- Particularizarea inferenței în probleme cu secvențe temporale pe cazul MMA
- Structura restricționată a MMA permite implementări elegante ale tuturor algoritmilor de bază

## Problema Evaluării

Dându-se un model și o secvență de observații, cum calculăm probabilitatea ca secvența observată să fi fost produsă de model?

## Problema Interpretării (cea mai bună explicație a observațiilor)

Dându-se un model și o secvență de observații, cum alegem o secvență corespunzătoare de stări care *dau sens* observațiilor?



# Cele 3 Probleme ale MMA

3 probleme fundamentale [Rab89]

- Particularizarea inferenței în probleme cu secvențe temporale pe cazul MMA
- Structura restricționată a MMA permite implementări elegante ale tuturor algoritmilor de bază

## Problema Evaluării

Dându-se un model și o secvență de observații, cum calculăm probabilitatea ca secvența observată să fi fost produsă de model?

## Problema Interpretării (cea mai bună explicație a observațiilor)

Dându-se un model și o secvență de observații, cum alegem o secvență corespunzătoare de stări care dă sens observațiilor?

## Problema Estimării (Antrenării) Modelului

Dându-se mai multe secvențe de observații, cum putem ajusta parametrii modelului MMA care explică cel mai bine observațiile făcute?



# Unde suntem...

## 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

## 4 Demo: Recunoașterea Simbolurilor

## 5 Tipuri de MMA

## 6 Discuții și Concluzii



# O perspectivă teoretică asupra MMA

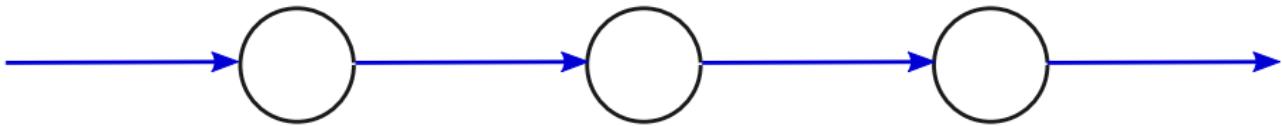
- Vom răspunde la următoarele întrebări:
  - ➊ Cum definim un Model Markov Ascuns?
  - ➋ Cum exprimăm cele trei întrebări fundamentale cu ajutorul probabilităților?

# Ce este un MMA?

## Definiție

Un **Model Markov Ascuns** este un dublu proces stocastic cu două componente:

- un proces Markov

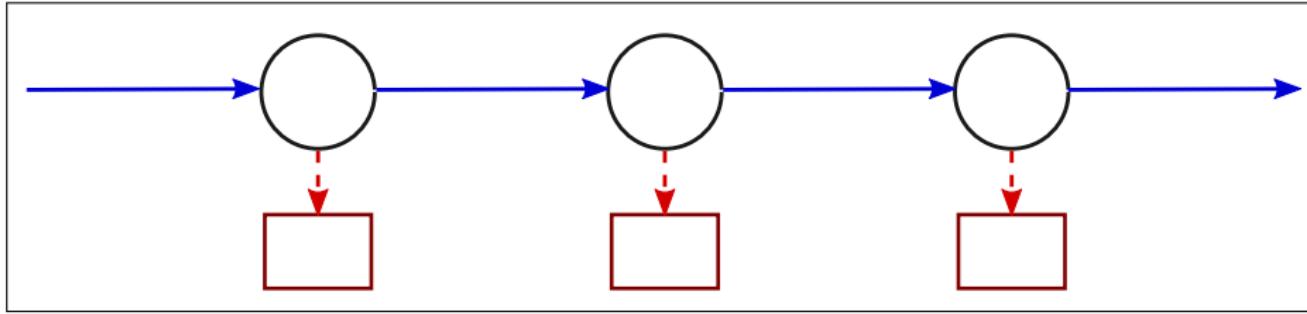


# Ce este un MMA?

## Definiție

Un **Model Markov Ascuns** este un dublu proces stocastic cu două componente:

- un proces Markov *neobservabil (ascuns)*,
- un set de procese stocastice care produc partea *observabilă*.





# Exemplu: Urmărirea stărilor emoționale

Să considerăm următorul exemplu:

- un robot ce urmărește evoluția stărilor emoționale ale unui om

Senzor:

- cameră video

Exemplu adaptat după :

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



## Exemplu: Urmărirea stărilor emoționale



**N** - numărul de stări ascunse  
**S** - mulțimea stărilor

---

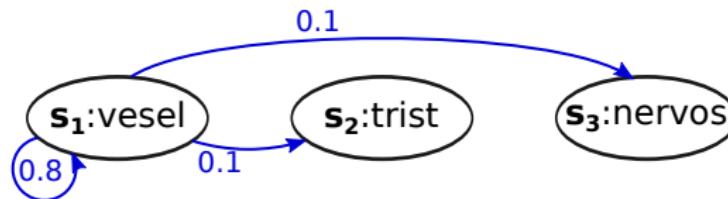
$$N = 3$$

Stări:

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

Starea - variabilă aleatoare multinomială

# Exemplu: Urmărirea stărilor emoționale



**A** - matricea distribuțiilor de probabilitate ale tranzițiilor între stări

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

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

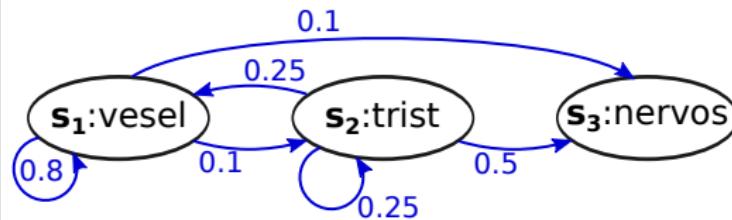

---

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


---

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

# Exemplu: Urmărirea stărilor emoționale



**A** - matricea distribuțiilor de probabilitate ale tranzițiilor între stări

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

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

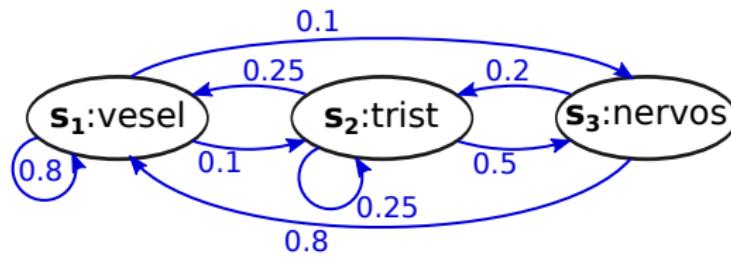

---

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


---

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

# Exemplu: Urmărirea stărilor emoționale



**A** - matricea distribuțiilor de probabilitate ale tranzițiilor între stări

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

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

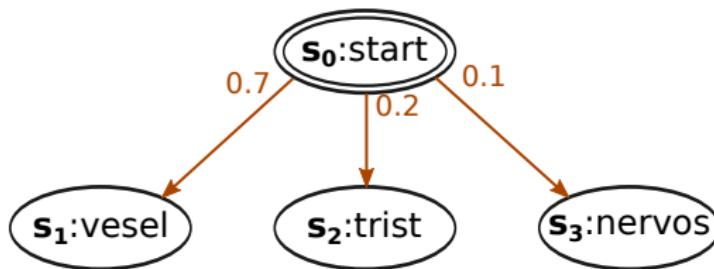

---

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


---

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

## Exemplu: Urmărirea stărilor emoționale



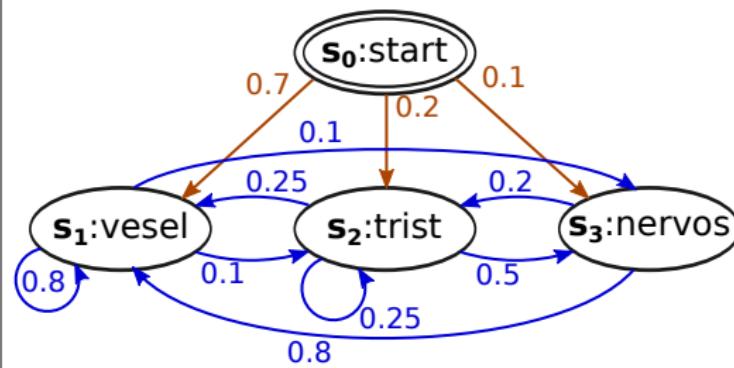
$\Pi$  - distribuția stării inițiale

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

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

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

# Exemplu: Urmărirea stărilor emoționale



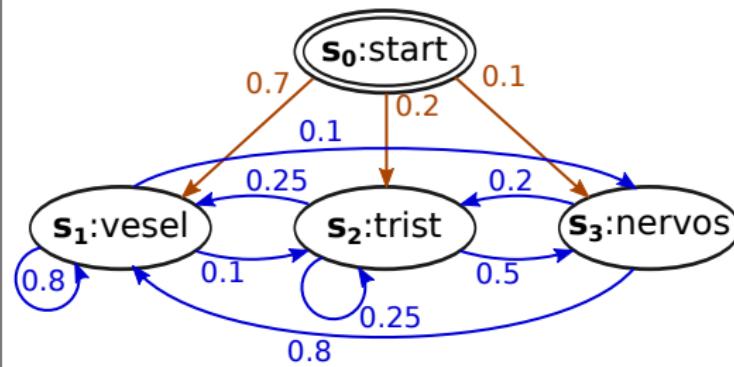
Deocamdată am descris un proces Markov.

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

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


---

# Exemplu: Urmărirea stărilor emoționale



vesel → vesel → nervos → trist

$$\mathbf{Q} = [q_1:s_1 \ q_2:s_1 \ q_3:s_3 \ q_4:s_2]$$

Deocamdată am descris un proces Markov.

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

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

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

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

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



# Exemplu: Urmărirea stărilor emoționale

$s_1$ : vesel       $s_2$ : trist       $s_3$ : nervos

$v_1$ : zâmbet / rânjet

$v_2$ : nimic

$v_3$ : încruntare

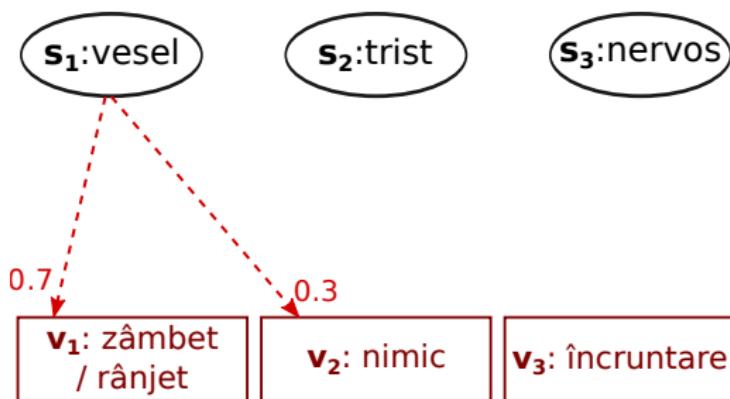
**M** - numărul de valori observabile distincte

$$M = 3$$

valori observabile:

- $v_1$ : zâmbet / rânjet
- $v_2$ : nimic
- $v_3$ : încruntare

# Exemplu: Urmărirea stărilor emoționale



**B** - matricea distribuțiilor de probabilitate ale valorilor observabile

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

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

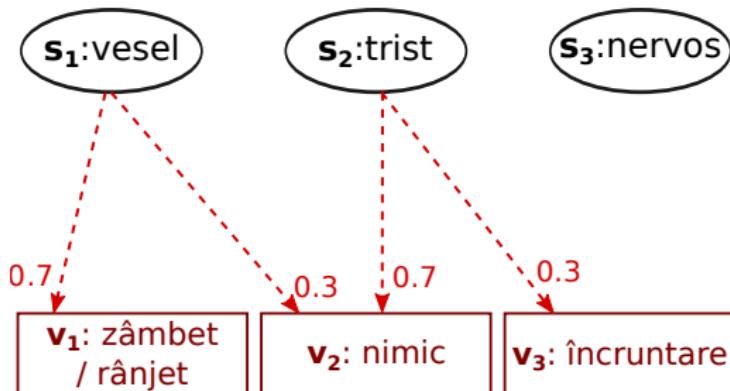
---


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


---

$$\mathbf{B} = \begin{matrix} & v_1 & v_2 & v_3 \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} & \left( \begin{matrix} 0.7 & 0.3 & 0 \end{matrix} \right) \end{matrix}$$

# Exemplu: Urmărirea stărilor emoționale



**B** - matricea distribuțiilor de probabilitate ale valorilor observabile

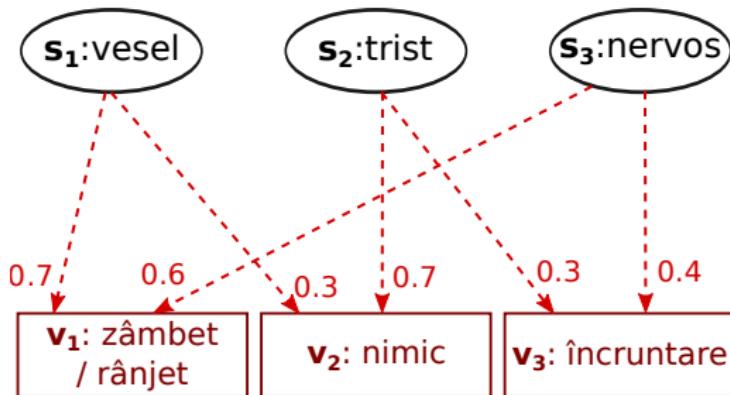
$$\mathbf{B} = \{b_{j,k}\}_{1 \leq j \leq N, 1 \leq k \leq M}$$

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

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

$$\mathbf{B} = \begin{matrix} & v_1 & v_2 & v_3 \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} & \left( \begin{matrix} 0.7 & 0.3 & 0 \\ 0 & 0.7 & 0.3 \end{matrix} \right) \end{matrix}$$

# Exemplu: Urmărirea stărilor emoționale



**B** - matricea distribuțiilor de probabilitate ale valorilor observabile

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

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

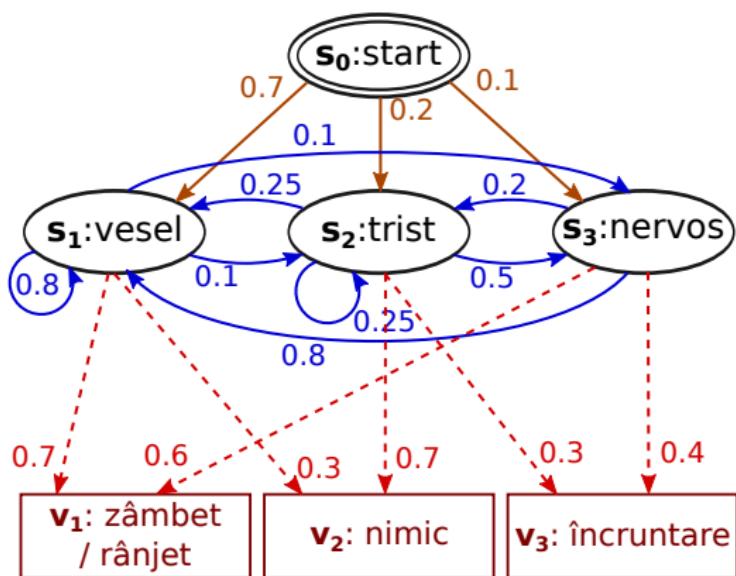

---

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


---

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

## Exemplu: Urmărirea stărilor emoționale



$\lambda$  - parametrii Modelului Markov Ascuns

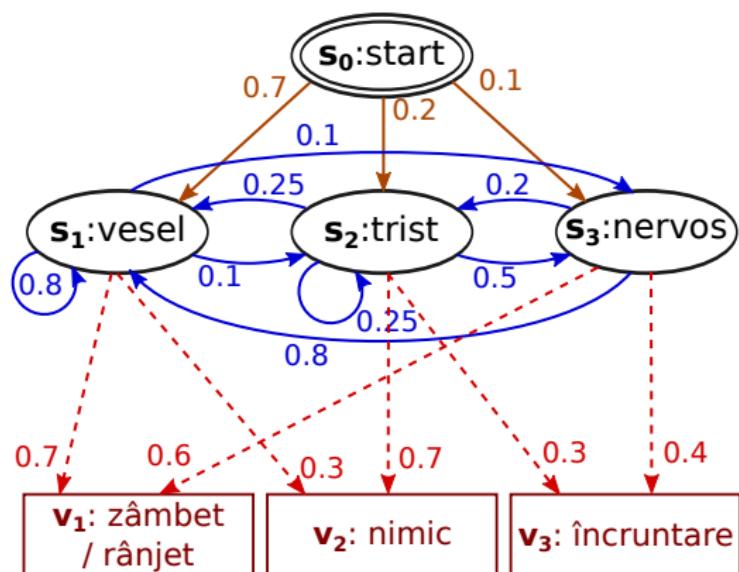
$$\lambda = (A, B, \Pi)$$

$A$  - matricea distribuțiilor de probabilitate ale tranzițiilor între stări

$B$  - matricea distribuțiilor de probabilitate ale valorilor observabile

$\Pi$  - distribuția stării inițiale

# Exemplu: Urmărirea stărilor emoționale



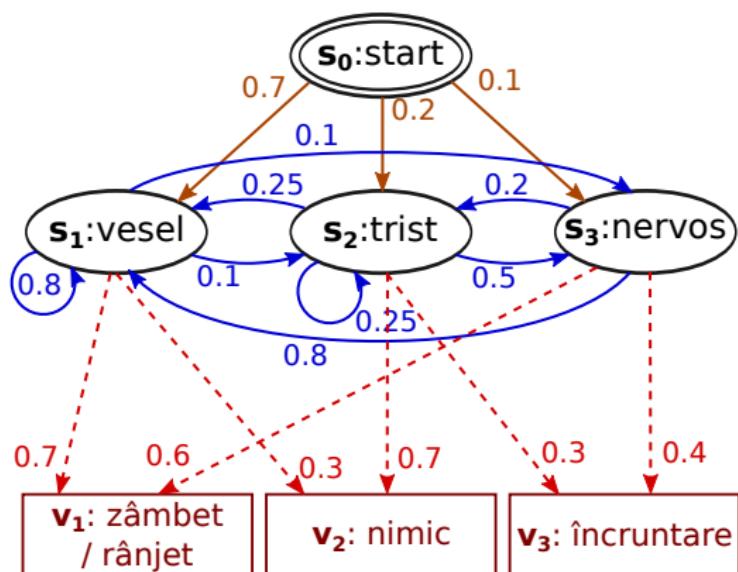
$$\mathbf{o} = [\mathbf{o}_1:zâmbet \ \mathbf{o}_2:nimic \ \mathbf{o}_3:nimic \\ \mathbf{o}_4:încruntare \ \mathbf{o}_5:zâmbet]$$

**O** - secvența de observații

**T** - lungimea secvenței de observații

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

# Exemplu: Urmărirea stărilor emoționale



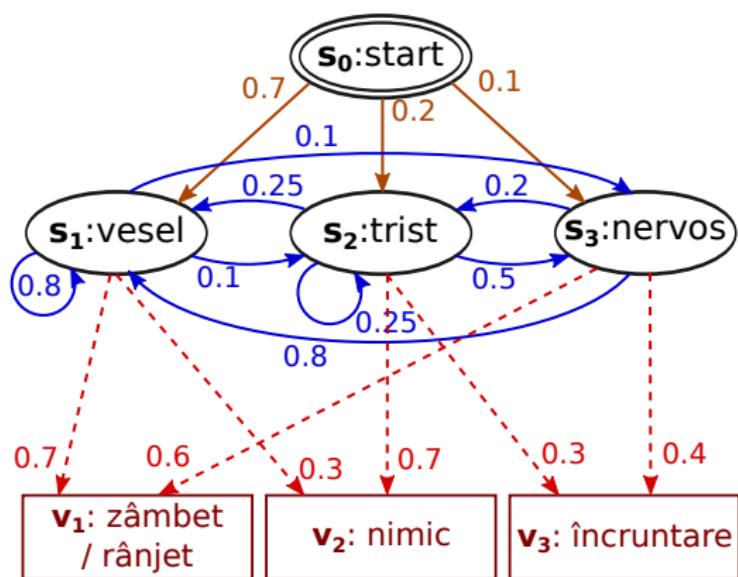
zâmbet, nimic, nimic, încruntare, zâmbet

**O** - secvența de observații

**T** - lungimea secvenței de observații

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

# Exemplu: Urmărirea stărilor emoționale



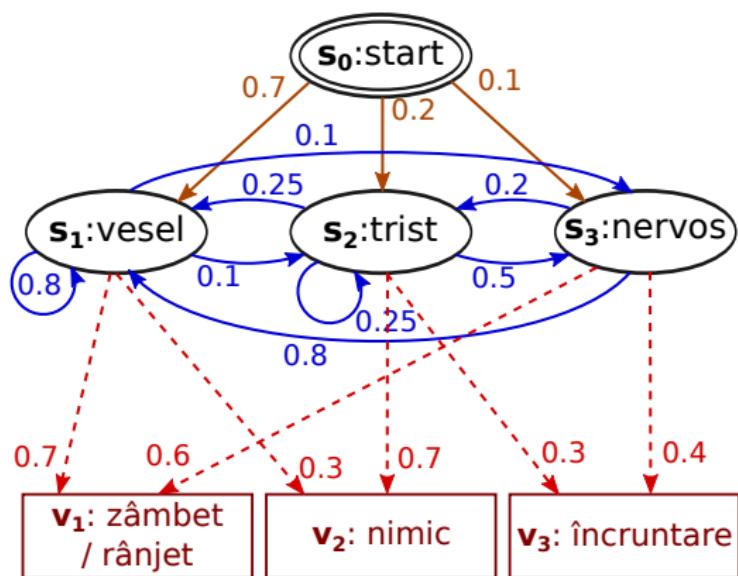
$\text{vesel} \rightarrow \text{vesel} \rightarrow \text{vesel} \rightarrow \text{trist} \rightarrow \text{nervos}$   
 $\text{zâmbet}, \text{nimic}, \text{nimic}, \text{încruntare}, \text{zâmbet}$

**O** - secvența de observații

**T** - lungimea secvenței de observații

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

# Exemplu: Urmărirea stărilor emotionale



**O** - secvența de observații

**T** - lungimea secvenței de observații

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

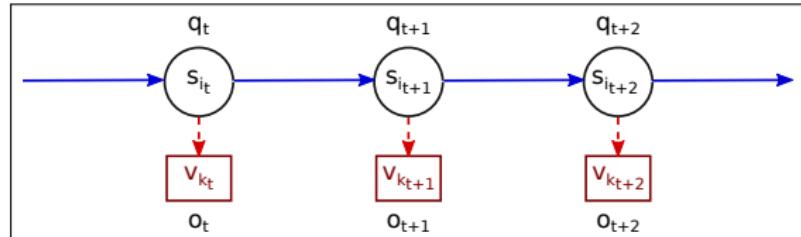
vesel → vesel → vesel → trist → nervos  
 zâmbet, nimic, nimic, încruntare, zâmbet  
 nervos → trist → trist → nervos → vesel

# Modele Markov Ascunse

## Definiție

Un **Model Markov Ascuns** este un tuplu  $\langle S, V, A, B, P \rangle$ :

- $S$  - mulțimea stărilor
- $V$  - mulțimea valorilor observabile
- $A$  - matricea de tranziție
- $B$  - matricea de emisie
- $\Pi$  - matricea distribuției stării inițiale
- Notație:  $\lambda = (A, B, \Pi)$  - parametrii modelului





# Reformularea problemelor 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 problemelor 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 problemelor 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:

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

(legea probabilității totale)



# Reformularea problemelor fundamentale ale MMA

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

---



# Reformularea problemelor fundamentale ale MMA

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

---

Primul factor (independență condițională):

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



# Reformularea problemelor fundamentale ale MMA

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

Primul factor (independență condițională):

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

Al doilea factor (presupunerea Markov):

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

# Reformularea problemelor fundamentale ale MMA

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

Primul factor (independență condițională):

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

Al doilea factor (presupunerea Markov):

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

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



# Reformularea problemelor fundamentale ale MMA

Problema interpretării 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 problemelor fundamentale ale MMA

Problema interpretării 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 problemelor fundamentale ale MMA

Problema interpretării 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?

- Cum alegem criteriul pentru *cea mai bună* secvență?



# Reformularea problemelor fundamentale ale MMA

Problema interpretării 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?

- Cum alegem criteriul pentru *cea mai bună* secvență?
  - Secvența celor mai probabile stări (luate individual):

$$Q_{\text{best}} = [q_{1_{\text{best}}} \ q_{2_{\text{best}}} \ \dots \ q_{T_{\text{best}}}], \quad q_{t_{\text{best}}} = \underset{s_i}{\operatorname{argmax}} P(q_t = s_i | O, \lambda) \quad (4)$$



# Reformularea problemelor fundamentale ale MMA

Problema interpretării 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?

- Cum alegem criteriul pentru *cea mai bună secvență*?
  - Secvența celor mai probabile stări (luate individual):

$$Q_{\text{best}} = [q_{1_{\text{best}}} \ q_{2_{\text{best}}} \ \dots \ q_{T_{\text{best}}}], \quad q_{t_{\text{best}}} = \underset{s_i}{\operatorname{argmax}} P(q_t = s_i | O, \lambda) \quad (4)$$

- 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 problemelor 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 problemelor 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 problemelor 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:

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



# Mediu de lucru

Pentru început

- ① Deschideți Matlab / Octave
- ② Schimbați directorul de lucru cu aria-hmm:  
`cd ("aria-hmm")`
- ③ Adăugați toate subdirectoarele în path:  
`addpath (genpath ('.', ))`



# Platforma de lucru - fișierele stub

## Fișierele .stub

- le veți folosi ca schelet de cod pentru sesiunile de implementare
- au secțiuni delimitate de `<label>-start` și `<label>-end` între care veți adăuga liniile de cod
- înlăturați sufixul `.stub` când rezolvați task-urile.

```
1 N = size(Pi, 2); % The number of states
2 T = size(O, 2); % The number of observations
3
4 Scale = zeros (1, T); % Scale is an 1 x T matrix
5 Alpha = zeros (T, N); % Alpha is a T x N matrix
6 Beta = ones (T, N); % Beta is a T x N matrix
7
8 %% Forward variables
9 % alpha_disc-start – Write code below
10
11 % alpha_disc-end – Write code above
```



# Platforma de lucru - tester

- Testerul se rulează folosind comanda `hmm-test`.

```
1 Please choose a test to run!
2 Type "list" to list all available tests.
3 Type "quit" to exit this tester.
4 Test:
```

- Cu `list` afișați toate testele disponibile.
- Pentru un anumit task numele testului coincide cu eticheta care delimită secțiunea de completat.



# Cuprins

- 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
- 4 Demo: Recunoașterea Simbolurilor
- 5 Tipuri de MMA
- 6 Discuții și Concluzii



# Unde suntem...

## 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

## 4 Demo: Recunoașterea Simbolurilor

## 5 Tipuri de MMA

## 6 Discuții și Concluzii



# Algoritmul Forward-Backward

## Problema Evaluării

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

$$P(O|\lambda) = ? \quad (7)$$



# Algoritmul Forward-Backward

## Problema Evaluării

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

$$P(O|\lambda) = ? \quad (7)$$

## Rezolvare

$P(O|\lambda)$  se calculează *eficient* cu algoritmul **Forward - Backward**.

Calculul se face cu ajutorul unor variabile auxiliare  $\alpha$  și  $\beta$ .



# Variabilele $\alpha$ - Motivație

- Calcul conform legii probabilităților totale:

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

- Pentru secvențele de stări care încep cu o secvență comună  $[q_1 q_2 \dots q_Z]$ , următorii factori sunt comuni:

$$\pi_{q_1} \cdot b_{q_1}(o_1) \cdot \prod_{z=2}^Z b_{q_z}(o_z) a_{q_{z-1}, q_z}$$

- Calculul de  $(T - Z)^N$  ori ar fi redundant!



## Variabilele $\alpha$ - Definiție

Definim variabilele  $\alpha$  astfel:

$$\alpha_{t,i} = P(o_1, o_2, \dots, o_t, q_t = s_i | \lambda) \quad (8)$$

$1 \leq t \leq T, 1 \leq i \leq N$

(probabilitatea de a fi observat primele  $t$  valori observate ajungând la momentul  $t$  în starea  $s_i$ , dați fiind parametrii  $\lambda$ )



## Variabilele $\alpha$ - Definiție

Definim variabilele  $\alpha$  astfel:

$$\alpha_{t,i} = P(o_1, o_2, \dots, o_t, q_t = s_i | \lambda) \quad (8)$$

$1 \leq t \leq T, 1 \leq i \leq N$

(probabilitatea de a fi observat primele  $t$  valori observate ajungând la momentul  $t$  în starea  $s_i$ , dați fiind parametrii  $\lambda$ )

Relația dintre  $P(O|\lambda)$  și variabilele  $\alpha$  este:

$$P(O|\lambda) = \sum_{i=1}^N \alpha_{T,i} \quad (9)$$

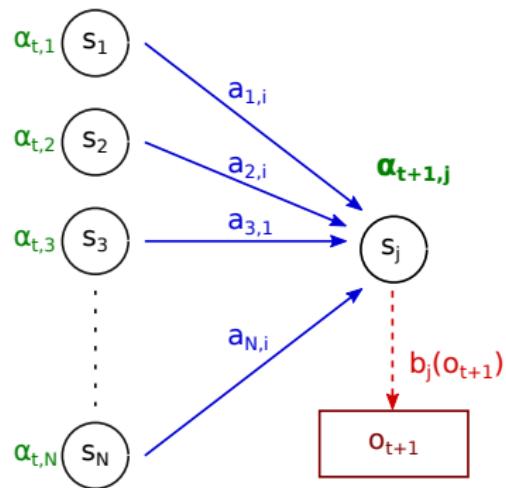
(conform legii probabilităților totale)



# Calculul variabilelor $\alpha$

- Initializare ( $t = 1$ )

$$\alpha_{1,i} = \pi_i b_i(o_1), \quad 1 \leq i \leq N$$

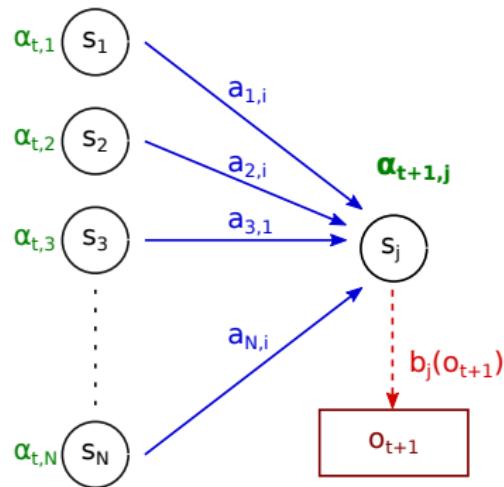




# Calculul variabilelor $\alpha$

- Initializare ( $t = 1$ )

$$\alpha_{1,i} = \pi_i b_i(o_1), \quad 1 \leq i \leq N$$



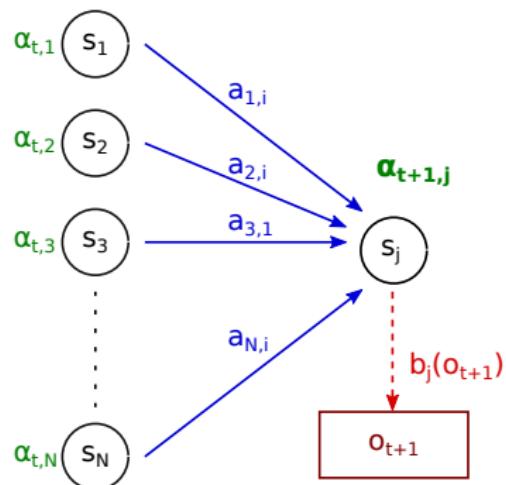
- Inducție ( $t > 1$ )

$$\alpha_{t+1,j} = \left[ \sum_{i=1}^N \alpha_{t,i} a_{i,j} \right] b_j(o_{t+1}), \quad \begin{matrix} 1 \leq t \leq T-1 \\ 1 \leq j \leq N \end{matrix}$$

# Calculul variabilelor $\alpha$

- Initializare ( $t = 1$ )

$$\alpha_{1,i} = \pi_i b_i(o_1), \quad 1 \leq i \leq N$$



- Inducție ( $t > 1$ )

$$\alpha_{t+1,j} = \left[ \sum_{i=1}^N \alpha_{t,i} a_{i,j} \right] b_j(o_{t+1}), \quad \begin{matrix} 1 \leq t \leq T-1 \\ 1 \leq j \leq N \end{matrix}$$

- Probabilitatea secvenței observate:

$$P(O|\lambda) = \sum_{i=1}^N \alpha_{T,i}$$



## Variabilele $\beta$ - Definiție

Definim variabilele  $\beta$  astfel:

$$\beta_{t,i} = P(o_{t+1} o_{t+2} \cdots o_T | q_t = s_i, \lambda) \quad (10)$$

(probabilitatea de a fi observate valorile secvenței de la  $t + 1$  la  $T$ , condiționată de aflarea la momentul  $t$  în starea  $s_i$  și de parametrii  $\lambda$ )



## Variabilele $\beta$ - Definiție

Definim variabilele  $\beta$  astfel:

$$\beta_{t,i} = P(o_{t+1} o_{t+2} \cdots o_T | q_t = s_i, \lambda) \quad (10)$$

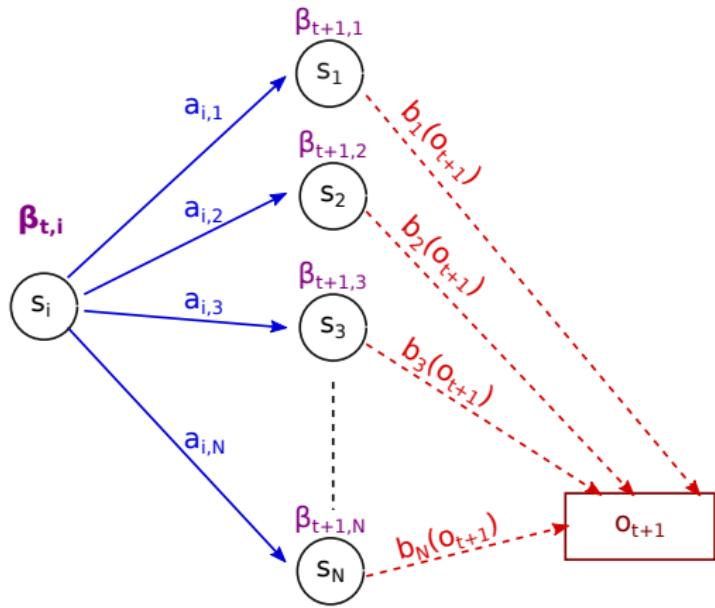
(probabilitatea de a fi observate valorile secvenței de la  $t + 1$  la  $T$ , condiționată de aflarea la momentul  $t$  în starea  $s_i$  și de parametrii  $\lambda$ )

Relația dintre  $P(O|\lambda)$  și variabilele  $\beta$  este:

$$P(O|\lambda) = \sum_{i=1}^N \beta_{1,i} \quad (11)$$

(conform legii probabilităților totale)

# Calculul Variabilelor $\beta$



- Inițializare ( $t = T$ )

$$\beta_{T,i} = 1, \quad 1 \leq i \leq N \quad (12)$$

- Pas de inducție ( $t < T$ )

$$\beta_{t,i} = \sum_{j=1}^N a_{i,j} b_j(o_{t+1}) \beta_{t+1,j}, \quad t = T-1, T-2, \dots, 1, 1 \leq i \leq N$$



# Probleme numerice

- să revedem definiția  $P(O|\lambda)$ :

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



# Probleme numerice

- să revedem definiția  $P(O|\lambda)$ :

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

- toți cei  $2T$  factori ai unui produs sunt subunitari
- pentru secvențe mari, produsele se apropie mult de zero
- se depășește precizia disponibilă pentru reprezentare
- trebuie introdus un mecanism de **scalare**



# Algoritmul Forward-Backward cu scalare

- $\hat{\alpha}_{t,i}$  - variabilele  $\alpha$  scalate
- $\hat{\beta}_{t,i}$  - variabilele  $\beta$  scalate
- $C_t$  - coeficienții de scalare
- Variabilele  $\alpha$  scalate

$$\hat{\alpha}_{t,i} = C_t \cdot \alpha_{t,i} \quad (13)$$

- Variabilele  $\beta$  scalate:

$$\hat{\beta}_{t,j} = C_t \cdot \beta_{t,j} \quad (14)$$

(notățiile sunt adoptate din literatură)



# Calculul variabilelor $\alpha$ scalate

- Inițializare  
 $(t = 1)$

$$1 \leq i \leq N$$

$$\ddot{\alpha}_{1,i} = \alpha_{1,i} \quad (15)$$

$$c_1 = \frac{1}{\sum_{i=1}^N \ddot{\alpha}_{1,i}} \quad (16)$$

$$\hat{\alpha}_{1,i} = c_1 \cdot \ddot{\alpha}_{1,i} \quad (17)$$



# Calculul variabilelor $\alpha$ scalate

- Inițializare  
( $t = 1$ )

$$1 \leq i \leq N$$

$$\ddot{\alpha}_{1,i} = \alpha_{1,i} \quad (15)$$

$$c_1 = \frac{1}{\sum_{i=1}^N \ddot{\alpha}_{1,i}} \quad (16)$$

$$\hat{\alpha}_{1,i} = c_1 \cdot \ddot{\alpha}_{1,i} \quad (17)$$


---

- Pas de inducție  
( $t > 1$ )

$$1 \leq j \leq N$$

$$\ddot{\alpha}_{t+1,j} = \left[ \sum_{i=1}^N \hat{\alpha}_{t,i} a_{i,j} \right] b_j(o_{t+1}) \quad (18)$$

$$c_{t+1} = \frac{1}{\sum_{j=1}^N \ddot{\alpha}_{t+1,j}} \quad (19)$$

$$\hat{\alpha}_{t+1,j} = c_{t+1} \cdot \ddot{\alpha}_{t+1,j} \quad (20)$$



# Coeficientii de scalare

Vom nota:  $C_t = c_1 \cdot c_2 \cdot \dots \cdot c_t$

**Ipoteza:**  $\hat{\alpha}_{t,i} = C_t \alpha_{t,i}$



# Coeficientii de scalare

Vom nota:  $C_t = c_1 \cdot c_2 \cdot \dots \cdot c_t$

**Ipoteza:**  $\hat{\alpha}_{t,i} = C_t \alpha_{t,i}$

Demonstrație prin inducție:

- ( $t = 1$ ):

$$\hat{\alpha}_{1,i} = c_1 \cdot \ddot{\alpha}_{1,i} = c_1 \cdot \alpha_{1,i}$$



# Coeficientii de scalare

Vom nota:  $C_t = c_1 \cdot c_2 \cdot \dots \cdot c_t$

**Ipoteza:**  $\hat{\alpha}_{t,i} = C_t \alpha_{t,i}$

Demonstrație prin inducție:

- ( $t = 1$ ):

$$\hat{\alpha}_{1,i} = c_1 \cdot \ddot{\alpha}_{1,i} = c_1 \cdot \alpha_{1,i}$$

- ( $t > 1$ ): Presupunem adevărat:  $\hat{\alpha}_{t,i} = C_t \alpha_{t,i}$

$$\ddot{\alpha}_{t+1,j} = \left[ \sum_{i=1}^N \hat{\alpha}_{t,i} a_{i,j} \right] b_j(o_{t+1}) = \left[ \sum_{i=1}^N C_t \alpha_{t,i} a_{i,j} \right] b_j(o_{t+1}) = C_t \alpha_{t+1,j}$$

$$\hat{\alpha}_{t+1,j} = c_{t+1} \cdot \ddot{\alpha}_{t+1,j} = c_{t+1} \cdot C_t \cdot \alpha_{t+1,j} = C_{t+1} \cdot \alpha_{t+1,j}$$



## Coeficientii de scalare

Vom nota:  $C_t = c_1 \cdot c_2 \cdot \dots \cdot c_t$

**Ipoteza:**  $\hat{\alpha}_{t,i} = C_t \alpha_{t,i}$

Atunci:  $P(\mathbf{O}|\lambda) = \sum_{i=1}^N \alpha_{T,i} = C_T^{-1} \cdot \sum_{i=1}^N \hat{\alpha}_{T,i} = C_T^{-1} = \prod_{t=1}^T c_t^{-1}$



## Coeficientii de scalare

Vom nota:  $C_t = c_1 \cdot c_2 \cdot \dots \cdot c_t$

**Ipoteza:**  $\hat{\alpha}_{t,i} = C_t \alpha_{t,i}$

Atunci:  $P(\mathbf{O}|\lambda) = \sum_{i=1}^N \alpha_{T,i} = C_T^{-1} \cdot \sum_{i=1}^N \hat{\alpha}_{T,i} = C_T^{-1} = \prod_{t=1}^T c_t^{-1}$

Calculăm  $P(O|\lambda)$ ?



# Coeficientii de scalare

Vom nota:  $C_t = c_1 \cdot c_2 \cdot \dots \cdot c_t$

**Ipoteza:**  $\hat{\alpha}_{t,i} = C_t \alpha_{t,i}$

Atunci:  $P(\mathbf{O}|\lambda) = \sum_{i=1}^N \alpha_{T,i} = C_T^{-1} \cdot \sum_{i=1}^N \hat{\alpha}_{T,i} = C_T^{-1} = \prod_{t=1}^T c_t^{-1}$

Calculăm  $P(O|\lambda)$ ? **NU**



# Coeficientii de scalare

Vom nota:  $C_t = c_1 \cdot c_2 \cdot \dots \cdot c_t$

**Ipoteza:**  $\hat{\alpha}_{t,i} = C_t \alpha_{t,i}$

$$\text{Atunci: } P(\mathbf{O}|\lambda) = \sum_{i=1}^N \alpha_{T,i} = C_T^{-1} \cdot \sum_{i=1}^N \hat{\alpha}_{T,i} = C_T^{-1} = \prod_{t=1}^T c_t^{-1}$$

Calculăm  $P(O|\lambda)$ ? **NU**

Calculăm:

$$\log(P(O|\lambda)) = \log\left(\prod_{t=1}^T c_t^{-1}\right) = \sum_{t=1}^T \log(c_t^{-1}) = -\sum_{t=1}^T \log(c_t) \quad (21)$$



# Calculul variabilelor $\beta$ scalate

- Inițializare

$$(t = T)$$

$$\ddot{\beta}_{T,i} = \beta_{T,i} = 1 \quad (22)$$

$$1 \leq i \leq N$$

$$\hat{\beta}_{T,i} = c_T \cdot \ddot{\beta}_{T,i} \quad (23)$$



# Calculul variabilelor $\beta$ scalate

- Inițializare

$$(t = T)$$

$$\ddot{\beta}_{T,i} = \beta_{T,i} = 1 \quad (22)$$

$$1 \leq i \leq N$$


---

$$\hat{\beta}_{T,i} = c_T \cdot \ddot{\beta}_{T,i} \quad (23)$$

- Pas de inducție

$$(t < T)$$

$$\ddot{\beta}_{t,i} = \sum_{j=1}^N a_{i,j} b_j(o_{t+1}) \hat{\beta}_{t+1,i} \quad (24)$$

$$1 \leq i \leq N$$

$$\hat{\beta}_{t,i} = c_t \cdot \ddot{\beta}_{t,i} \quad (25)$$

- Se demonstrează:  $\hat{\beta}_{t,i} = c_t \cdot c_{t+1} \cdot \dots \cdot c_T \cdot \beta_{t,i}$



## Exemplu de aplicație

- Să reluăm exemplul de mai devreme cu robotul.
- Robotul acționează diferit în fața a 2 tipuri de personalitate:
  - *joivialul*
  - *morocănosul*.
- Pentru fiecare există un set de parametri:
  - $\lambda^1 = (A^1, B^1, \Pi^1)$
  - $\lambda^2 = (A^2, B^2, \Pi^2)$
- Robotul recunoaște următoarele gesturi:  
 $(zâmbet \mid rânjet) \longrightarrow (zâmbet \mid rânjet) \longrightarrow \text{încruntare}$
- **Întrebare:** Cărei personalități este mai probabil să-i aparțină cel monitorizat?



# Jovialul

## Jovialul



**s<sub>1</sub>:**vesel

**s<sub>2</sub>:**trist

**s<sub>3</sub>:**nervos

$$\lambda^1 = (A^1, B^1, \Pi^1)$$

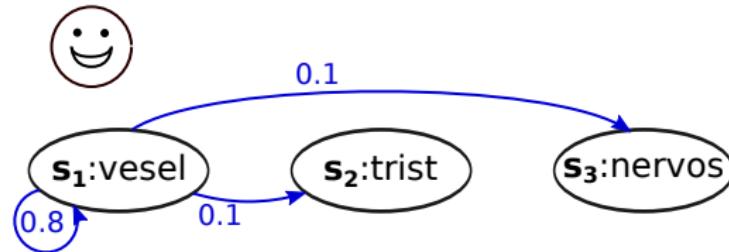
$$A^1 = \begin{matrix} s_1 & s_2 & s_3 \\ s_1 & & \\ s_2 & & \\ s_3 & & \end{matrix}$$

$$\Pi^1 = \begin{pmatrix} s_1 & s_2 & s_3 \end{pmatrix}$$

$$B^1 = \begin{matrix} v_1 & v_2 & v_3 \\ s_1 & & \\ s_2 & & \\ s_3 & & \end{matrix}$$

# Jovialul

## Jovialul



$$\lambda^1 = (A^1, B^1, \Pi^1)$$

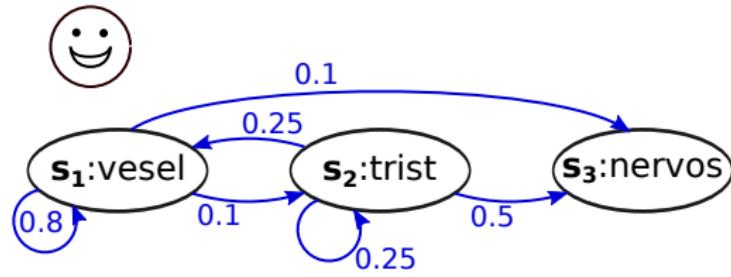
$$A^1 = \begin{matrix} & s_1 & s_2 & s_3 \\ s_1 & 0.8 & 0.1 & 0.1 \\ s_2 & & & \\ s_3 & & & \end{matrix}$$

$$\Pi^1 = \begin{pmatrix} & s_1 & s_2 & s_3 \\ & & & \\ & & & \end{pmatrix}$$

$$B^1 = \begin{matrix} & v_1 & v_2 & v_3 \\ s_1 & & & \\ s_2 & & & \\ s_3 & & & \end{matrix}$$

# Jovialul

## Jovialul



$$\lambda^1 = (A^1, B^1, \Pi^1)$$

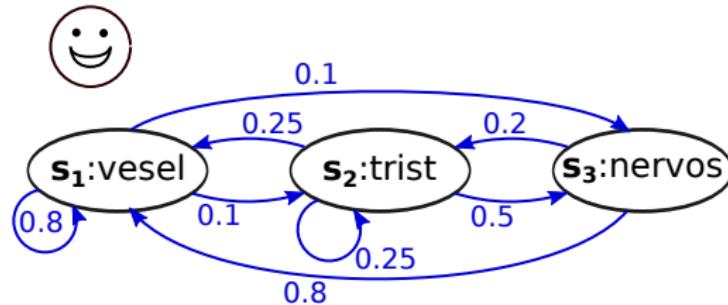
$$A^1 = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_1 & 0.8 & 0.1 & 0.1 \\ s_2 & 0.25 & 0.25 & 0.5 \\ s_3 & & & \end{pmatrix}$$

$$\Pi^1 = \left( \begin{array}{ccc} s_1 & s_2 & s_3 \end{array} \right)$$

$$B^1 = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

# Jovialul

## Jovialul



$$\lambda^1 = (A^1, B^1, \Pi^1)$$

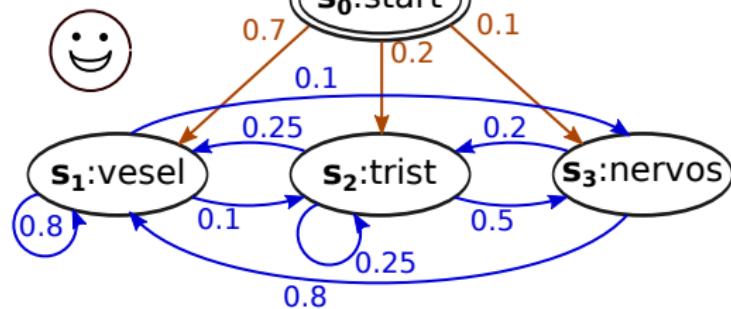
$$A^1 = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_1 & 0.8 & 0.1 & 0.1 \\ s_2 & 0.25 & 0.25 & 0.5 \\ s_3 & 0.8 & 0.2 & 0 \end{pmatrix}$$

$$\Pi^1 = \left( \begin{array}{ccc} s_1 & s_2 & s_3 \end{array} \right)$$

$$B^1 = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

# Jovialul

**Jovialul**



$$\lambda^1 = (A^1, B^1, \Pi^1)$$

$$A^1 = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_1 & 0.8 & 0.1 & 0.1 \\ s_2 & 0.25 & 0.25 & 0.5 \\ s_3 & 0.8 & 0.2 & 0 \end{pmatrix}$$

$$\Pi^1 = \begin{pmatrix} s_1 & s_2 & s_3 \\ 0.7 & 0.2 & 0.1 \end{pmatrix}$$

$$B^1 = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix}$$



# Jovialul

## Jovialul



**s<sub>1</sub>:** vesel

**s<sub>2</sub>:** trist

**s<sub>3</sub>:** nervos

**v<sub>1</sub>:** zâmbet  
/ rânjet

**v<sub>2</sub>:** nimic

**v<sub>3</sub>:** încruntare

$$\lambda^1 = (A^1, B^1, \Pi^1)$$

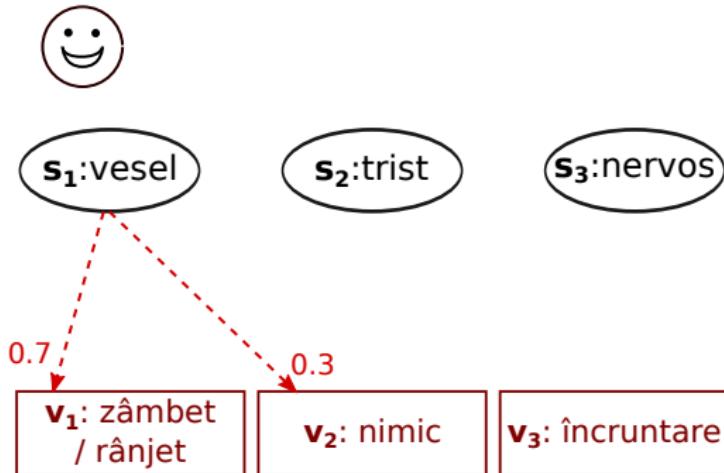
$$A^1 = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_1 & 0.8 & 0.1 & 0.1 \\ s_2 & 0.25 & 0.25 & 0.5 \\ s_3 & 0.8 & 0.2 & 0 \end{pmatrix}$$

$$\Pi^1 = \begin{pmatrix} s_1 & s_2 & s_3 \\ 0.7 & 0.2 & 0.1 \end{pmatrix}$$

$$B^1 = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

# Jovialul

## Jovialul



$$\lambda^1 = (A^1, B^1, \Pi^1)$$

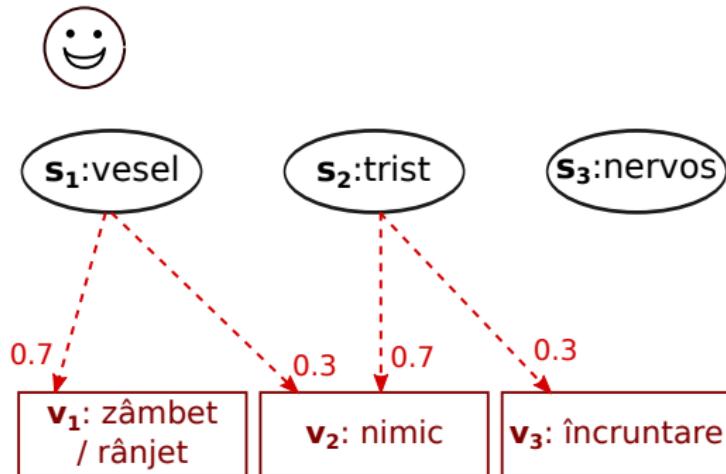
$$A^1 = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_1 & 0.8 & 0.1 & 0.1 \\ s_2 & 0.25 & 0.25 & 0.5 \\ s_3 & 0.8 & 0.2 & 0 \end{pmatrix}$$

$$\Pi^1 = \begin{pmatrix} s_1 & s_2 & s_3 \\ 0.7 & 0.2 & 0.1 \end{pmatrix}$$

$$B^1 = \begin{pmatrix} v_1 & v_2 & v_3 \\ s_1 & 0.7 & 0.3 & 0 \\ s_2 & \\ s_3 & \end{pmatrix}$$

# Jovialul

## Jovialul



$$\lambda^1 = (A^1, B^1, \Pi^1)$$

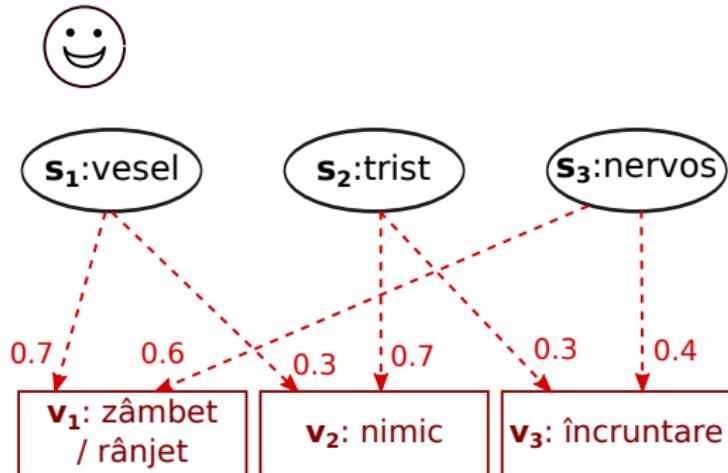
$$A^1 = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_1 & 0.8 & 0.1 & 0.1 \\ s_2 & 0.25 & 0.25 & 0.5 \\ s_3 & 0.8 & 0.2 & 0 \end{pmatrix}$$

$$\Pi^1 = \begin{pmatrix} s_1 & s_2 & s_3 \\ 0.7 & 0.2 & 0.1 \end{pmatrix}$$

$$B^1 = \begin{pmatrix} v_1 & v_2 & v_3 \\ s_1 & 0.7 & 0.3 & 0 \\ s_2 & 0 & 0.7 & 0.3 \\ s_3 & 0 & 0 & 0 \end{pmatrix}$$

# Jovialul

## Jovialul



$$\lambda^1 = (A^1, B^1, \Pi^1)$$

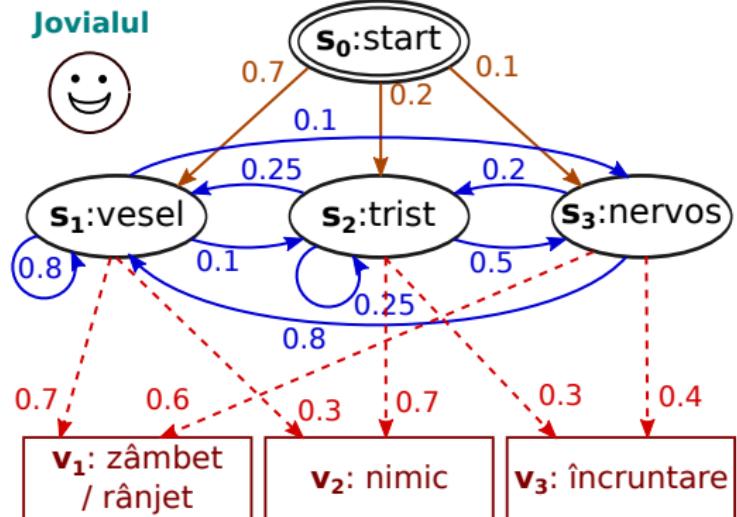
$$A^1 = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_1 & 0.8 & 0.1 & 0.1 \\ s_2 & 0.25 & 0.25 & 0.5 \\ s_3 & 0.8 & 0.2 & 0 \end{pmatrix}$$

$$\Pi^1 = \begin{pmatrix} s_1 & s_2 & s_3 \\ 0.7 & 0.2 & 0.1 \end{pmatrix}$$

$$B^1 = \begin{pmatrix} v_1 & v_2 & v_3 \\ s_1 & 0.7 & 0.3 & 0 \\ s_2 & 0 & 0.7 & 0.3 \\ s_3 & 0.6 & 0 & 0.4 \end{pmatrix}$$

# Jovialul

**Jovialul**



$$\lambda^1 = (A^1, B^1, \Pi^1)$$

$$A^1 = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_1 & 0.8 & 0.1 & 0.1 \\ s_2 & 0.25 & 0.25 & 0.5 \\ s_3 & 0.8 & 0.2 & 0 \end{pmatrix}$$

$$\Pi^1 = \begin{pmatrix} s_1 & s_2 & s_3 \\ 0.7 & 0.2 & 0.1 \end{pmatrix}$$

$$B^1 = \begin{pmatrix} v_1 & v_2 & v_3 \\ s_1 & 0.7 & 0.3 & 0 \\ s_2 & 0 & 0.7 & 0.3 \\ s_3 & 0.6 & 0 & 0.4 \end{pmatrix}$$



# Morocănosul

## Morocănosul



**s<sub>1</sub>**:vesel

**s<sub>2</sub>**:trist

**s<sub>3</sub>**:nervos

$$\lambda^2 = (A^2, B^2, \Pi^2)$$

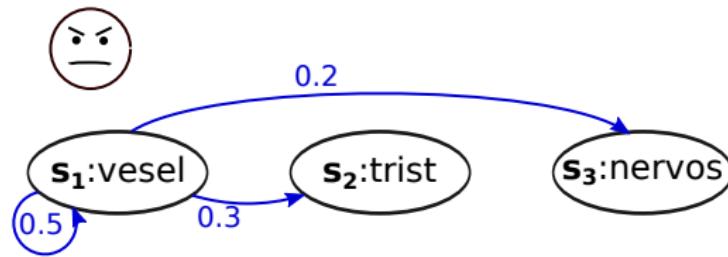
$$A^2 = \begin{matrix} s_1 & s_2 & s_3 \\ s_1 \\ s_2 \\ s_3 \end{matrix} \left( \begin{array}{c} \\ \\ \\ \end{array} \right)$$

$$\Pi^2 = \begin{matrix} s_1 & s_2 & s_3 \\ ( & & ) \end{matrix}$$

$$B^2 = \begin{matrix} v_1 & v_2 & v_3 \\ s_1 \\ s_2 \\ s_3 \end{matrix} \left( \begin{array}{c} \\ \\ \\ \end{array} \right)$$

# Morocănosul

## Morocănosul



$$\lambda^2 = (A^2, B^2, \Pi^2)$$

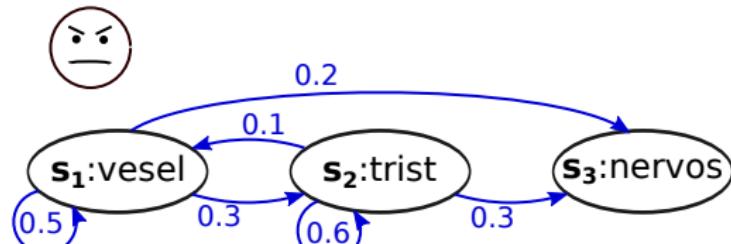
$$A^2 = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_1 & 0.5 & 0.3 & 0.2 \\ s_2 & & & \\ s_3 & & & \end{pmatrix}$$

$$\Pi^2 = \left( \begin{array}{ccc} s_1 & s_2 & s_3 \\ v_1 & v_2 & v_3 \end{array} \right)$$

$$B^2 = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix}$$

# Morocănosul

## Morocănosul



$$\lambda^2 = (A^2, B^2, \Pi^2)$$

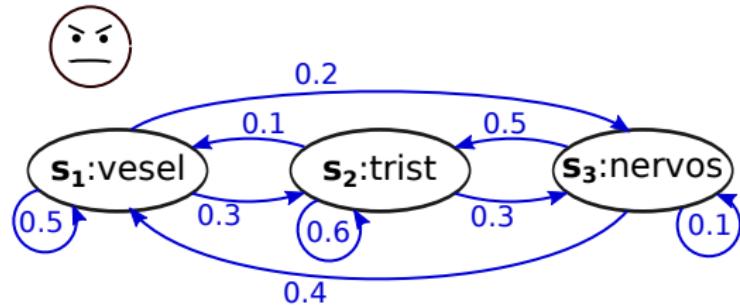
$$A^2 = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_1 & 0.5 & 0.3 & 0.2 \\ s_2 & 0.1 & 0.6 & 0.3 \\ s_3 & & & \end{pmatrix}$$

$$\Pi^2 = \left( \begin{array}{ccc} s_1 & s_2 & s_3 \\ & & \end{array} \right)$$

$$B^2 = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ v_1 & v_2 & v_3 \end{pmatrix}$$

# Morocănosul

## Morocănosul



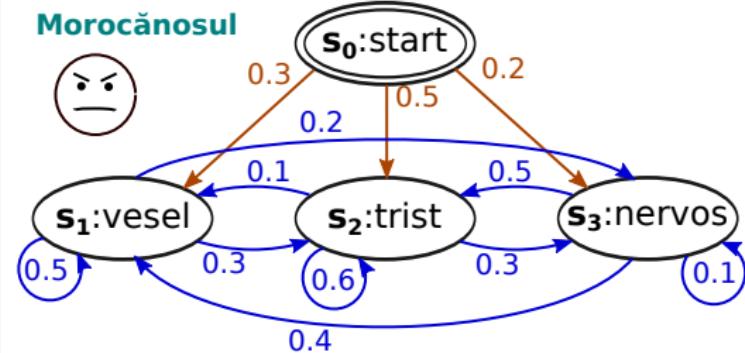
$$\lambda^2 = (A^2, B^2, \Pi^2)$$

$$A^2 = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_1 & 0.5 & 0.3 & 0.2 \\ s_2 & 0.1 & 0.6 & 0.3 \\ s_3 & 0.4 & 0.5 & 0.1 \end{pmatrix}$$

$$\Pi^2 = \left( \begin{array}{ccc} s_1 & s_2 & s_3 \end{array} \right)$$

$$B^2 = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix}$$

# Morocănosul



$$\lambda^2 = (A^2, B^2, \Pi^2)$$

$$A^2 = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_1 & 0.5 & 0.3 & 0.2 \\ s_2 & 0.1 & 0.6 & 0.3 \\ s_3 & 0.4 & 0.5 & 0.1 \end{pmatrix}$$

$$\Pi^2 = \begin{pmatrix} s_1 & s_2 & s_3 \\ 0.3 & 0.5 & 0.2 \end{pmatrix}$$

$$B^2 = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix}$$

# Morocănosul

## Morocănosul



**s<sub>1</sub>:** vesel

**s<sub>2</sub>:** trist

**s<sub>3</sub>:** nervos

**v<sub>1</sub>:** zâmbet / rânger

**v<sub>2</sub>:** nimic

**v<sub>3</sub>:** încruntare

$$\lambda^2 = (A^2, B^2, \Pi^2)$$

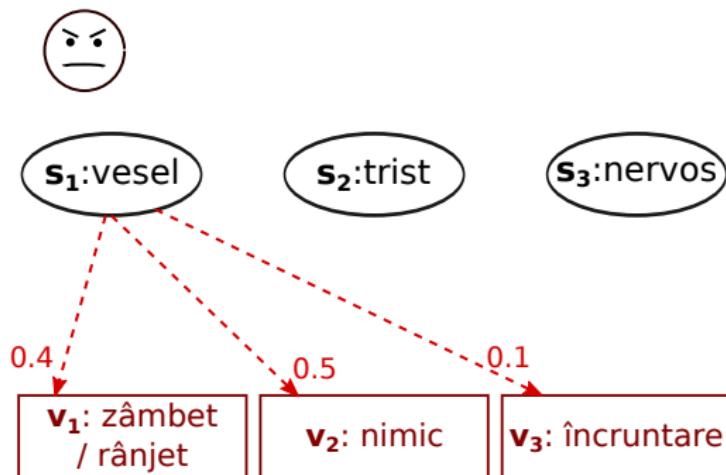
$$A^2 = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_1 & 0.5 & 0.3 & 0.2 \\ s_2 & 0.1 & 0.6 & 0.3 \\ s_3 & 0.4 & 0.5 & 0.1 \end{pmatrix}$$

$$\Pi^2 = \begin{pmatrix} s_1 & s_2 & s_3 \\ 0.3 & 0.5 & 0.2 \end{pmatrix}$$

$$B^2 = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix}$$

# Morocănosul

## Morocănosul



$$\lambda^2 = (A^2, B^2, \Pi^2)$$

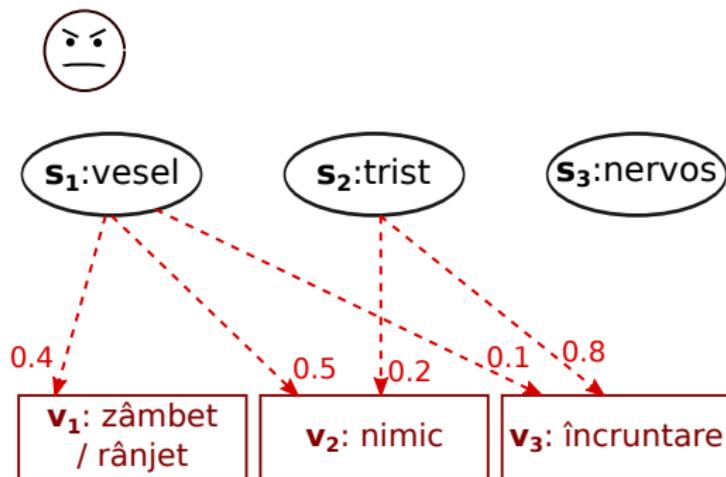
$$A^2 = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_1 & 0.5 & 0.3 & 0.2 \\ s_2 & 0.1 & 0.6 & 0.3 \\ s_3 & 0.4 & 0.5 & 0.1 \end{pmatrix}$$

$$\Pi^2 = \begin{pmatrix} s_1 & s_2 & s_3 \\ 0.3 & 0.5 & 0.2 \end{pmatrix}$$

$$B^2 = \begin{pmatrix} v_1 & v_2 & v_3 \\ s_1 & 0.4 & 0.5 & 0.1 \\ s_2 & \\ s_3 & \end{pmatrix}$$

# Morocănosul

## Morocănosul



$$\lambda^2 = (A^2, B^2, \Pi^2)$$

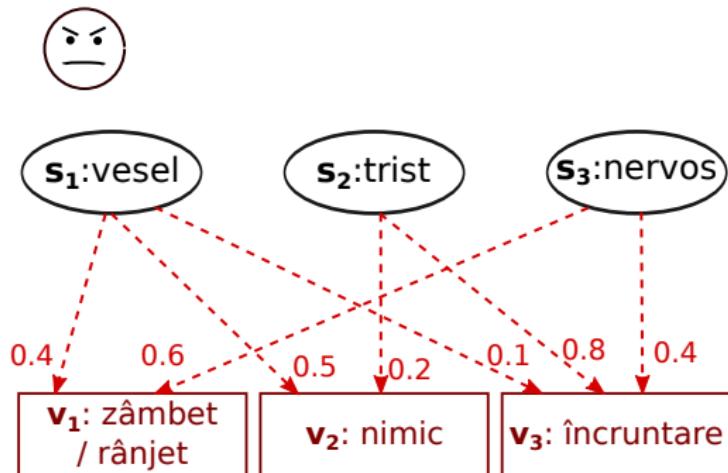
$$A^2 = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_1 & 0.5 & 0.3 & 0.2 \\ s_2 & 0.1 & 0.6 & 0.3 \\ s_3 & 0.4 & 0.5 & 0.1 \end{pmatrix}$$

$$\Pi^2 = \begin{pmatrix} s_1 & s_2 & s_3 \\ 0.3 & 0.5 & 0.2 \end{pmatrix}$$

$$B^2 = \begin{pmatrix} v_1 & v_2 & v_3 \\ s_1 & 0.4 & 0.5 & 0.1 \\ s_2 & 0 & 0.2 & 0.8 \\ s_3 & \dots & \dots & \dots \end{pmatrix}$$

# Morocănosul

## Morocănosul



$$\lambda^2 = (A^2, B^2, \Pi^2)$$

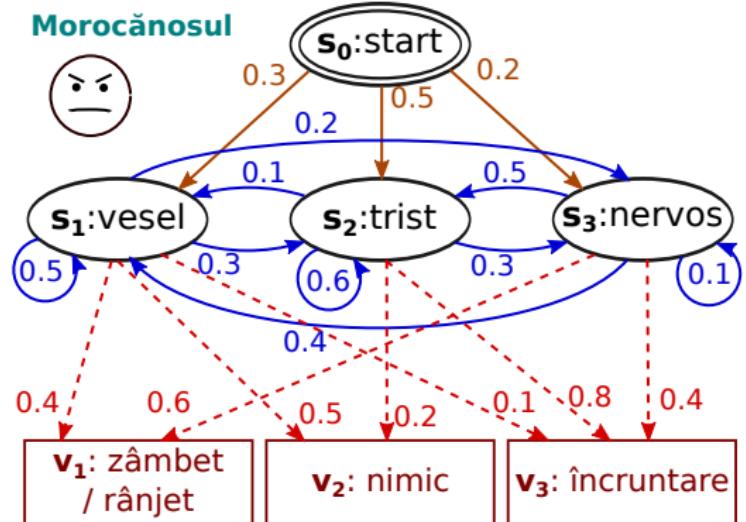
$$A^2 = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_1 & 0.5 & 0.3 & 0.2 \\ s_2 & 0.1 & 0.6 & 0.3 \\ s_3 & 0.4 & 0.5 & 0.1 \end{pmatrix}$$

$$\Pi^2 = \begin{pmatrix} s_1 & s_2 & s_3 \\ 0.3 & 0.5 & 0.2 \end{pmatrix}$$

$$B^2 = \begin{pmatrix} v_1 & v_2 & v_3 \\ s_1 & 0.4 & 0.5 & 0.1 \\ s_2 & 0 & 0.2 & 0.8 \\ s_3 & 0.6 & 0 & 0.4 \end{pmatrix}$$

# Morocănosul

**Morocănosul**



$$\lambda^2 = (A^2, B^2, \Pi^2)$$

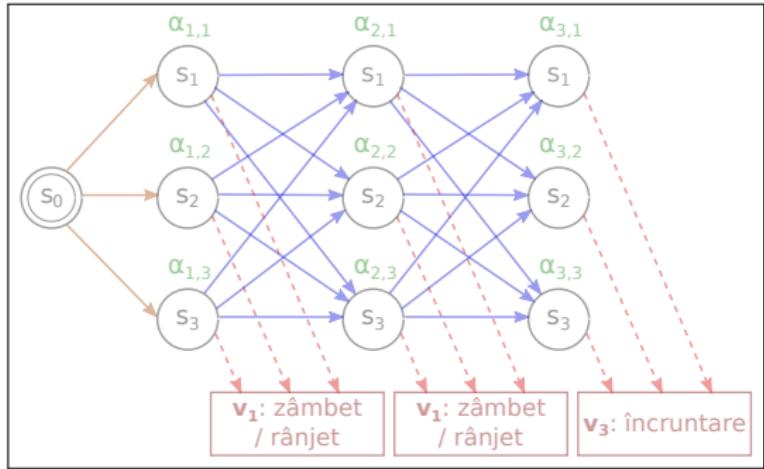
$$A^2 = \begin{pmatrix} s_1 & s_2 & s_3 \\ s_1 & 0.5 & 0.3 & 0.2 \\ s_2 & 0.1 & 0.6 & 0.3 \\ s_3 & 0.4 & 0.5 & 0.1 \end{pmatrix}$$

$$\Pi^2 = \begin{pmatrix} s_1 & s_2 & s_3 \\ 0.3 & 0.5 & 0.2 \end{pmatrix}$$

$$B^2 = \begin{pmatrix} v_1 & v_2 & v_3 \\ s_1 & 0.4 & 0.5 & 0.1 \\ s_2 & 0 & 0.2 & 0.8 \\ s_3 & 0.6 & 0 & 0.4 \end{pmatrix}$$

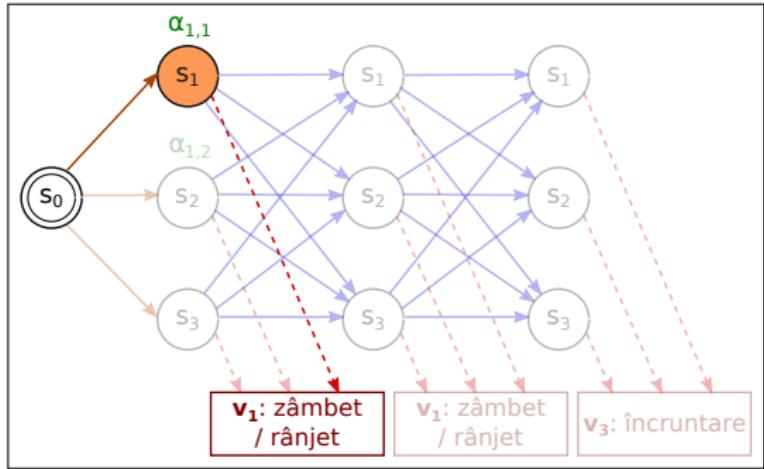


# Calculul variabilelor $\alpha$ :



$$\hat{\alpha} = \left[ \quad \right] \quad c = \left[ \quad \right]$$

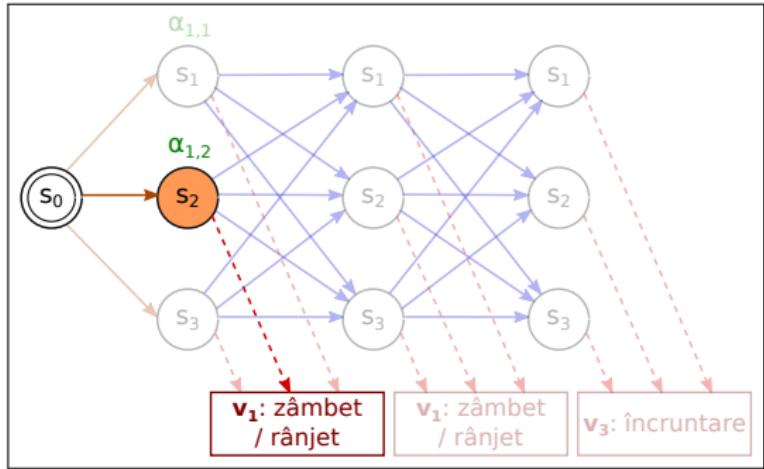
# Calculul variabilelor $\alpha$ :



$$\ddot{\alpha}_{1,1} = \pi_{1,1} \cdot b_1(v_1) = 0.7 \cdot 0.7 = 0.49$$

$$\hat{\alpha} = \left[ \quad \right] \quad c = \left[ \quad \right]$$

# Calculul variabilelor $\alpha$ :

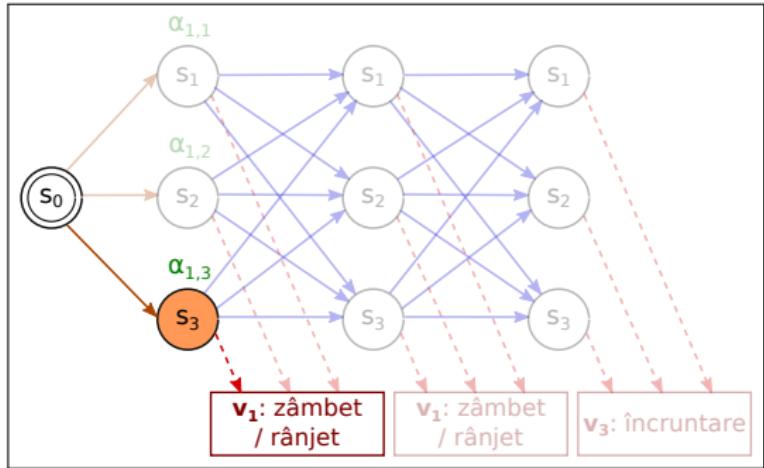


$$\ddot{\alpha}_{1,1} = \pi_{1,1} \cdot b_1(v_1) = 0.7 \cdot 0.7 = 0.49$$

$$\ddot{\alpha}_{1,2} = \pi_{1,2} \cdot b_2(v_1) = 0.2 \cdot 0 = 0$$

$$\hat{\alpha} = \left[ \quad \right] c = \left[ \quad \right]$$

# Calculul variabilelor $\alpha$ :



$$\ddot{\alpha}_{1,1} = \pi_{1,1} \cdot b_1(v_1) = 0.7 \cdot 0.7 = 0.49$$

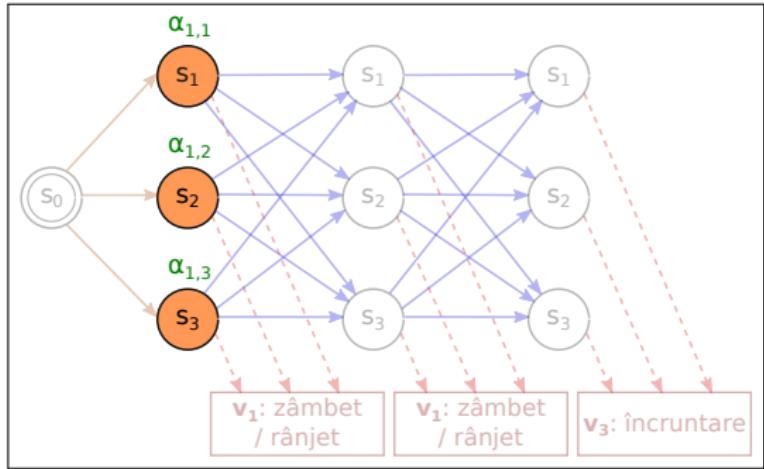
$$\ddot{\alpha}_{1,2} = \pi_{1,2} \cdot b_2(v_1) = 0.2 \cdot 0 = 0$$

$$\ddot{\alpha}_{1,3} = \pi_{1,3} \cdot b_3(v_1) = 0.1 \cdot 0.6 = 0.6$$

$$\hat{\alpha} = \left[ \quad \right] c = \left[ \quad \right]$$



# Calculul variabilelor $\alpha$ :



$$\ddot{\alpha}_{1,1} = \pi_{1,1} \cdot b_1(v_1) = 0.7 \cdot 0.7 = 0.49$$

$$\ddot{\alpha}_{1,2} = \pi_{1,2} \cdot b_2(v_1) = 0.2 \cdot 0 = 0$$

$$\ddot{\alpha}_{1,3} = \pi_{1,3} \cdot b_3(v_1) = 0.1 \cdot 0.6 = 0.6$$

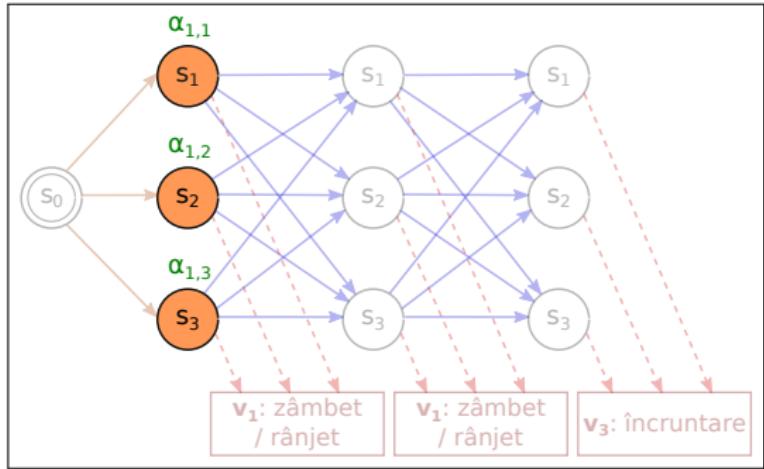

---

$$c_1 = \frac{1}{\ddot{\alpha}_{1,1} + \ddot{\alpha}_{1,2} + \ddot{\alpha}_{1,3}}$$

$$c_1 = 1 / 0.54 \approx 1.8182$$

$$\hat{\alpha} = \left[ \begin{array}{c} \alpha_{1,1} \\ \alpha_{1,2} \\ \alpha_{1,3} \end{array} \right] \quad c = \left[ \begin{array}{c} 1.8182 \\ \vdots \\ \vdots \end{array} \right]$$

# Calculul variabilelor $\alpha$ :



$$\ddot{\alpha}_{1,1} = \pi_{1,1} \cdot b_1(v_1) = 0.7 \cdot 0.7 = 0.49$$

$$\ddot{\alpha}_{1,2} = \pi_{1,2} \cdot b_2(v_1) = 0.2 \cdot 0 = 0$$

$$\ddot{\alpha}_{1,3} = \pi_{1,3} \cdot b_3(v_1) = 0.1 \cdot 0.6 = 0.6$$


---

$$c_1 = \frac{1}{\ddot{\alpha}_{1,1} + \ddot{\alpha}_{1,2} + \ddot{\alpha}_{1,3}}$$

$$c_1 = 1/0.54 \approx 1.8182$$


---

$$\hat{\alpha} = \begin{bmatrix} 0.8909 & 0 & 0.1091 \end{bmatrix} \quad c = \begin{bmatrix} 1.8182 \end{bmatrix}$$

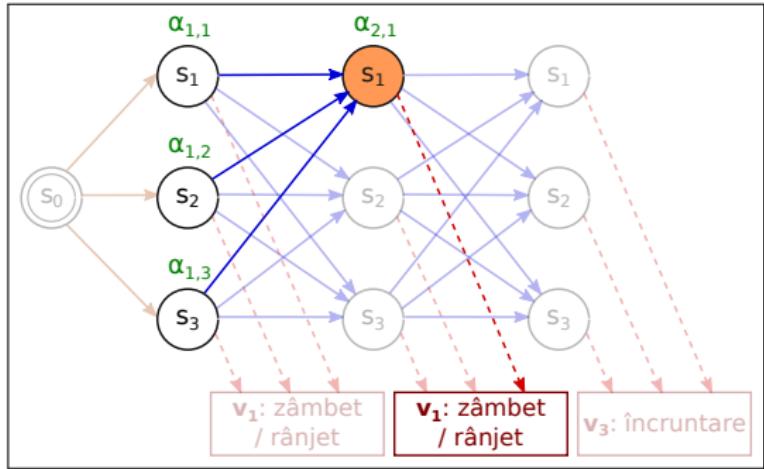
$$\hat{\alpha}_{1,1} = c_1 \cdot \ddot{\alpha}_{1,1} = 0.8909$$

$$\hat{\alpha}_{1,2} = c_1 \cdot \ddot{\alpha}_{1,2} = 0$$

$$\hat{\alpha}_{1,3} = c_1 \cdot \ddot{\alpha}_{1,3} = 0.1091$$



# Calculul variabilelor $\alpha$ :

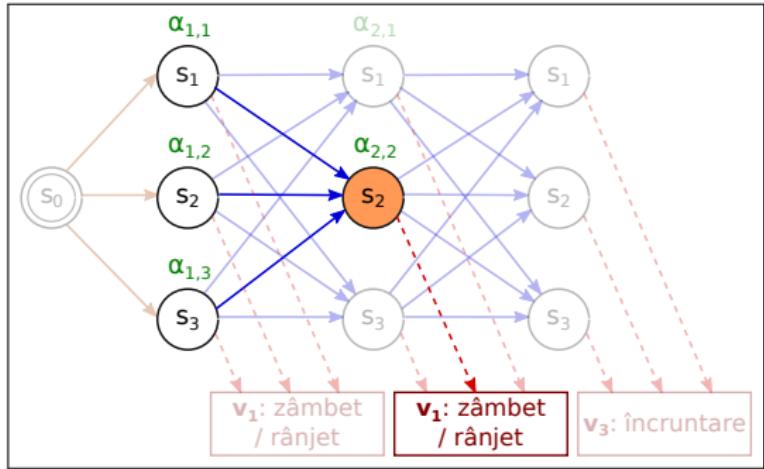


$$\ddot{\alpha}_{2,1} = b_1(v_1) \sum \hat{\alpha}_{1,i} a_{i,1} \\ = \dots = 0.56$$

$$\hat{\alpha} = \begin{bmatrix} 0.8909 & 0 & 0.1091 \end{bmatrix} \quad c = \begin{bmatrix} 1.8182 \end{bmatrix}$$



# Calculul variabilelor $\alpha$ :



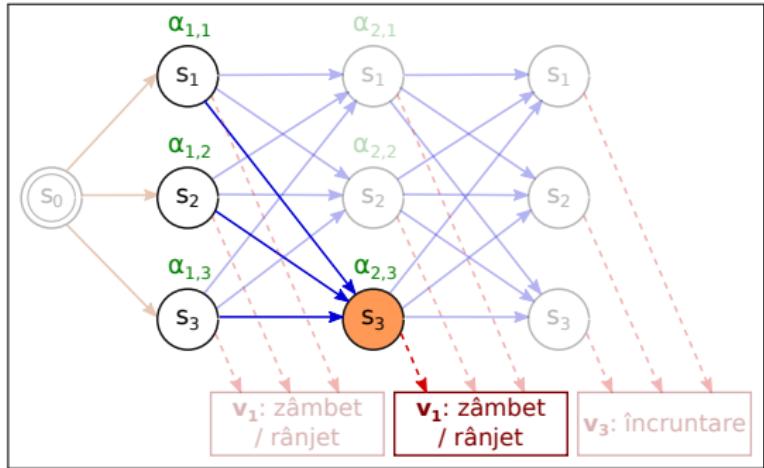
$$\ddot{\alpha}_{2,1} = b_1(v_1) \sum \hat{\alpha}_{1,i} a_{i,1} \\ = \dots = 0.56$$

$$\ddot{\alpha}_{2,2} = b_2(v_1) \sum \hat{\alpha}_{1,i} a_{i,2} = 0$$

$$\hat{\alpha} = \begin{bmatrix} 0.8909 & 0 & 0.1091 \end{bmatrix} \quad c = \begin{bmatrix} 1.8182 \end{bmatrix}$$



# Calculul variabilelor $\alpha$ :



$$\ddot{\alpha}_{2,1} = b_1(v_1) \sum \hat{\alpha}_{1,i} a_{i,1} \\ = \dots = 0.56$$

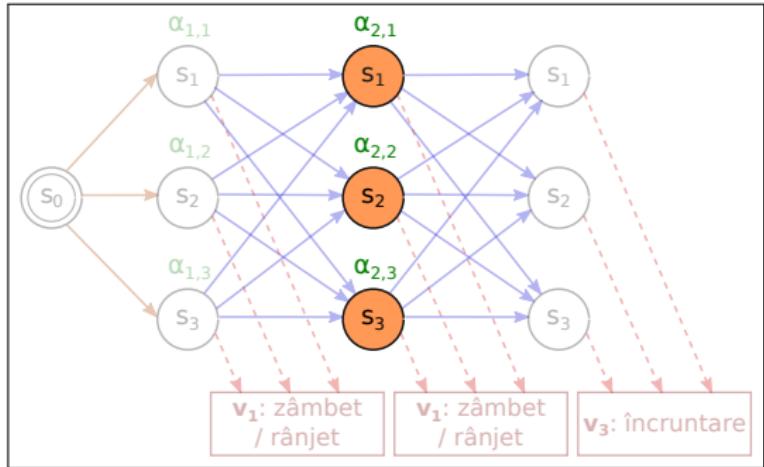
$$\ddot{\alpha}_{2,2} = b_2(v_1) \sum \hat{\alpha}_{1,i} a_{i,2} = 0$$

$$\ddot{\alpha}_{2,3} = b_3(v_1) \sum \hat{\alpha}_{1,i} a_{i,3} = 0.0535$$

$$\hat{\alpha} = \begin{bmatrix} 0.8909 & 0 & 0.1091 \end{bmatrix} \quad c = \begin{bmatrix} 1.8182 \end{bmatrix}$$



# Calculul variabilelor $\alpha$ :



$$\ddot{\alpha}_{2,1} = b_1(v_1) \sum \hat{\alpha}_{1,i} a_{i,1} \\ = \dots = 0.56$$

$$\ddot{\alpha}_{2,2} = b_2(v_1) \sum \hat{\alpha}_{1,i} a_{i,2} = 0$$

$$\ddot{\alpha}_{2,3} = b_3(v_1) \sum \hat{\alpha}_{1,i} a_{i,3} = 0.0535$$


---

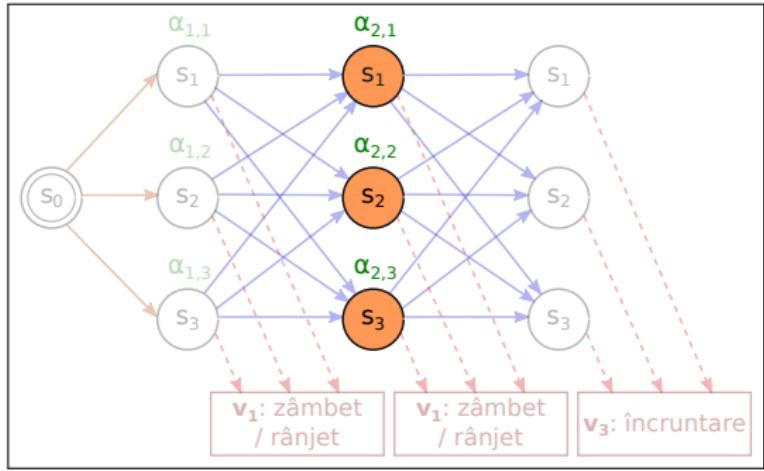
$$c_2 = \frac{1}{\ddot{\alpha}_{2,1} + \ddot{\alpha}_{2,2} + \ddot{\alpha}_{2,3}}$$

$$c_2 = \frac{1}{0.484 + 0.0776 + 0} = 1.63$$

$$\hat{\alpha} = \begin{bmatrix} 0.8909 & 0 & 0.1091 \end{bmatrix} \quad c = \begin{bmatrix} 1.8182 \\ 1.63 \end{bmatrix}$$



# Calculul variabilelor $\alpha$ :



$$\hat{\alpha} = \begin{bmatrix} 0.8909 & 0 & 0.1091 \\ 0.9128 & 0 & 0.0872 \end{bmatrix} \quad c = \begin{bmatrix} 1.8182 \\ 1.63 \end{bmatrix}$$

$$\ddot{\alpha}_{2,1} = b_1(v_1) \sum \hat{\alpha}_{1,i} a_{i,1} \\ = \dots = 0.56$$

$$\ddot{\alpha}_{2,2} = b_2(v_1) \sum \hat{\alpha}_{1,i} a_{i,2} = 0$$

$$\ddot{\alpha}_{2,3} = b_3(v_1) \sum \hat{\alpha}_{1,i} a_{i,3} = 0.0535$$


---

$$c_2 = \frac{1}{\ddot{\alpha}_{2,1} + \ddot{\alpha}_{2,2} + \ddot{\alpha}_{2,3}}$$

$$c_2 = \frac{1}{0.484 + 0.0776 + 0} = 1.63$$


---

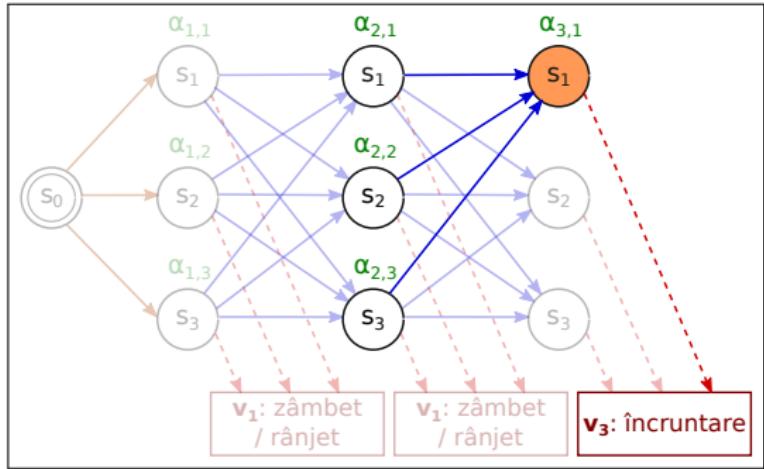
$$\hat{\alpha}_{2,1} = c_2 \cdot \ddot{\alpha}_{2,1} = 0.9128$$

$$\hat{\alpha}_{2,2} = c_2 \cdot \ddot{\alpha}_{2,2} = 0$$

$$\hat{\alpha}_{2,3} = c_2 \cdot \ddot{\alpha}_{2,3} = 0.0872$$



# Calculul variabilelor $\alpha$ :

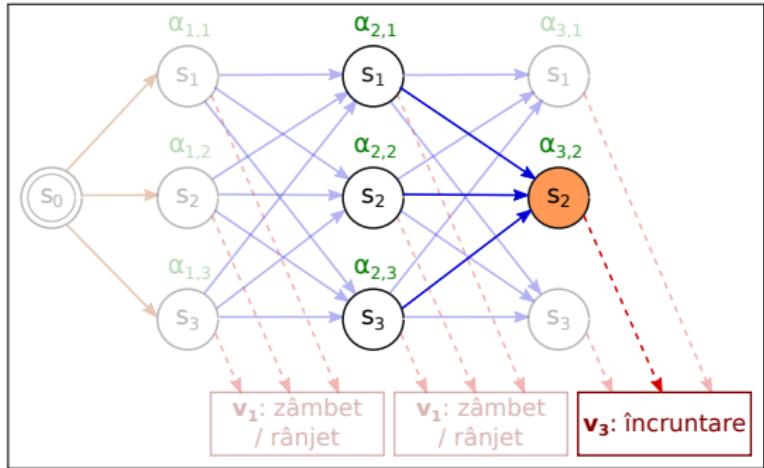


$$\begin{aligned}\ddot{\alpha}_{3,1} &= b_1(v_3) \sum \hat{\alpha}_{2,i} a_{i,1} \\ &= \dots = 0\end{aligned}$$

$$\hat{\alpha} = \begin{bmatrix} 0.8909 & 0 & 0.1091 \\ 0.9128 & 0 & 0.0872 \end{bmatrix} \quad c = \begin{bmatrix} 1.8182 \\ 1.63 \end{bmatrix}$$



# Calculul variabilelor $\alpha$ :



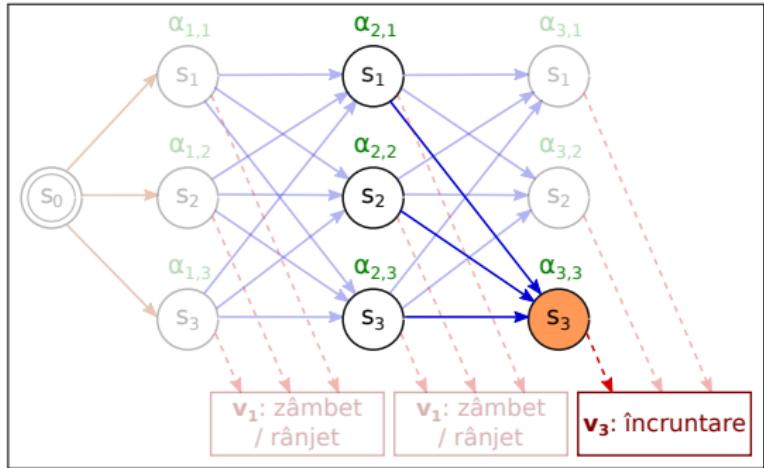
$$\ddot{\alpha}_{3,1} = b_1(v_3) \sum \hat{\alpha}_{2,i} a_{i,1} \\ = \dots = 0$$

$$\ddot{\alpha}_{3,2} = b_2(v_3) \sum \hat{\alpha}_{2,i} a_{i,2} = 0.0326$$

$$\hat{\alpha} = \begin{bmatrix} 0.8909 & 0 & 0.1091 \\ 0.9128 & 0 & 0.0872 \end{bmatrix} \quad c = \begin{bmatrix} 1.8182 \\ 1.63 \end{bmatrix}$$



# Calculul variabilelor $\alpha$ :



$$\ddot{\alpha}_{3,1} = b_1(v_3) \sum \hat{\alpha}_{2,i} a_{i,1} = \dots = 0$$

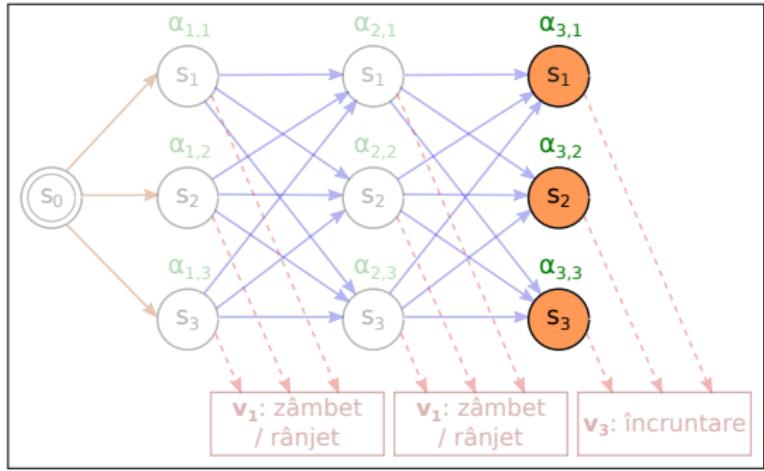
$$\ddot{\alpha}_{3,2} = b_2(v_3) \sum \hat{\alpha}_{2,i} a_{i,2} = 0.0326$$

$$\ddot{\alpha}_{3,3} = b_3(v_3) \sum \hat{\alpha}_{2,i} a_{i,3} = 0.0365$$

$$\hat{\alpha} = \begin{bmatrix} 0.8909 & 0 & 0.1091 \\ 0.9128 & 0 & 0.0872 \end{bmatrix} \quad c = \begin{bmatrix} 1.8182 \\ 1.63 \end{bmatrix}$$



# Calculul variabilelor $\alpha$ :



$$\ddot{\alpha}_{3,1} = b_1(v_3) \sum \hat{\alpha}_{2,i} a_{i,1} \\ = \dots = 0$$

$$\ddot{\alpha}_{3,2} = b_2(v_3) \sum \hat{\alpha}_{2,i} a_{i,2} = 0.0326$$

$$\ddot{\alpha}_{3,3} = b_3(v_3) \sum \hat{\alpha}_{2,i} a_{i,3} = 0.0365$$


---

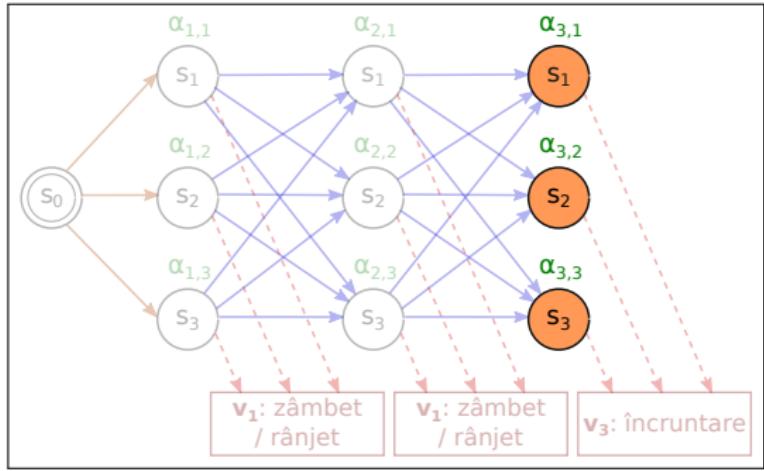
$$c_3 = \frac{1}{\ddot{\alpha}_{3,1} + \ddot{\alpha}_{3,2} + \ddot{\alpha}_{3,3}}$$

$$c_3 = \frac{1}{0.069} = 14.4718$$

$$\hat{\alpha} = \begin{bmatrix} 0.8909 & 0 & 0.1091 \\ 0.9128 & 0 & 0.0872 \end{bmatrix} \quad c = \begin{bmatrix} 1.8182 \\ 1.63 \\ 14.4718 \end{bmatrix}$$



# Calculul variabilelor $\alpha$ :



$$\ddot{\alpha}_{3,1} = b_1(v_3) \sum \hat{\alpha}_{2,i} a_{i,1} \\ = \dots = 0$$

$$\ddot{\alpha}_{3,2} = b_2(v_3) \sum \hat{\alpha}_{2,i} a_{i,2} = 0.0326$$

$$\ddot{\alpha}_{3,3} = b_3(v_3) \sum \hat{\alpha}_{2,i} a_{i,3} = 0.0365$$


---

$$c_3 = \frac{1}{\ddot{\alpha}_{3,1} + \ddot{\alpha}_{3,2} + \ddot{\alpha}_{3,3}}$$

$$c_3 = \frac{1}{0.069} = 14.4718$$


---

$$\hat{\alpha} = \begin{bmatrix} 0.8909 & 0 & 0.1091 \\ 0.9128 & 0 & 0.0872 \\ 0 & 0.4718 & 0.5282 \end{bmatrix} \quad c = \begin{bmatrix} 1.8182 \\ 1.63 \\ 14.4718 \end{bmatrix}$$

$$\hat{\alpha}_{3,1} = c_3 \cdot \ddot{\alpha}_{3,1} = 0$$

$$\hat{\alpha}_{3,2} = c_3 \cdot \ddot{\alpha}_{3,2} = 0.4718$$

$$\hat{\alpha}_{3,3} = c_3 \cdot \ddot{\alpha}_{3,3} = 0.5282$$



# Morocănos sau jovial?

- $c = [1.8182 \quad 1.63 \quad 14.4718]$
- Probabilitatea ca sevența observată să fi fost generată de un *jovial*:  
 $\log(P(O|\lambda^1)) = - \sum \log(c_i) = -3.7583$



# Morocănos sau jovial?

- $c = [1.8182 \quad 1.63 \quad 14.4718]$
- Probabilitatea ca sevența observată să fi fost generată de un *jovial*:  
 $\log(P(O|\lambda^1)) = -\sum \log(c_i) = -3.7583$
- Probabilitatea ca sevența observată să fi fost generată de un *morocănos*:  
 $\log(P(O|\lambda^2)) = -3.6362$
- $\log(P(O|\lambda^2)) > \log(P(O|\lambda^1))$
- Este mai probabil să avem de-a face cu un...



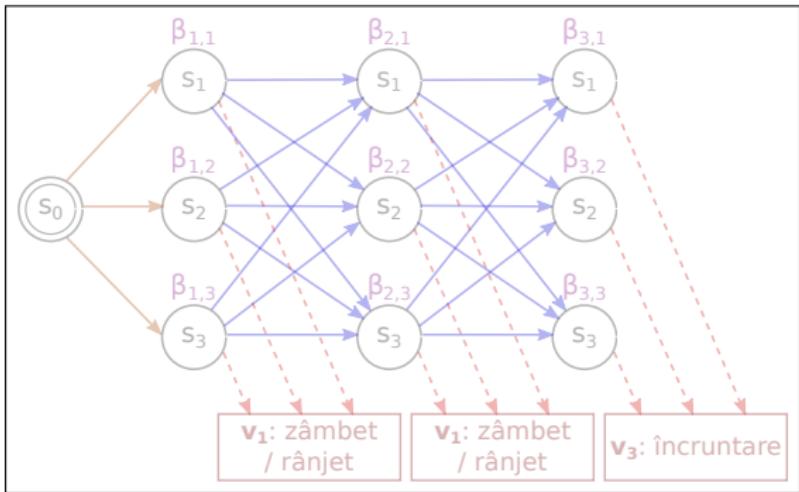
# Morocănos sau jovial?

- $c = [1.8182 \quad 1.63 \quad 14.4718]$
- Probabilitatea ca sevența observată să fi fost generată de un *jovial*:  
 $\log(P(O|\lambda^1)) = -\sum \log(c_i) = -3.7583$
- Probabilitatea ca sevența observată să fi fost generată de un *morocănos*:  
 $\log(P(O|\lambda^2)) = -3.6362$
- $\log(P(O|\lambda^2)) > \log(P(O|\lambda^1))$
- Este mai probabil să avem de-a face cu un...

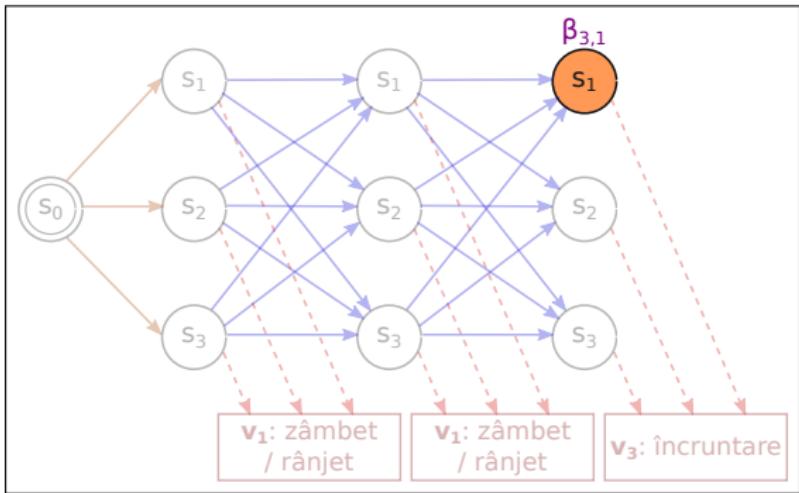
**Morocănos**



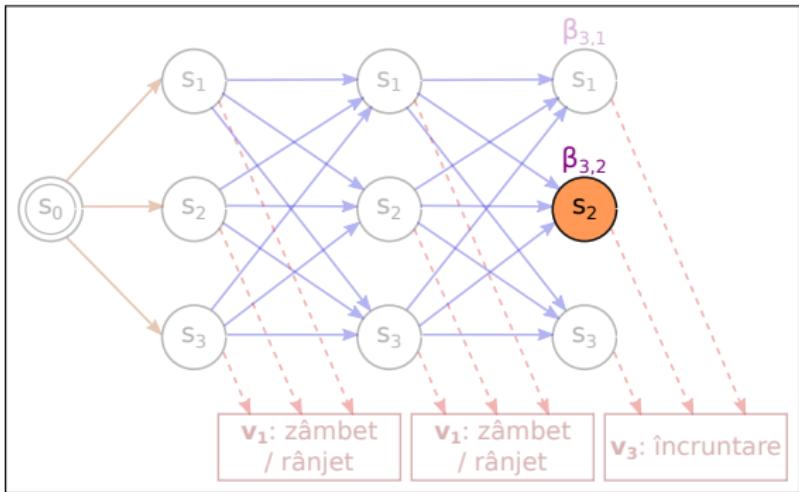
# Calculul variabilelor $\beta$ :



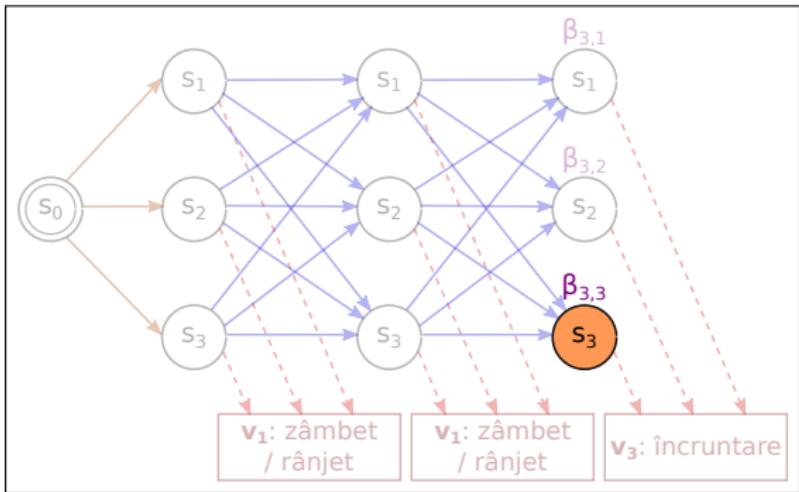
# Calculul variabilelor $\beta$ :



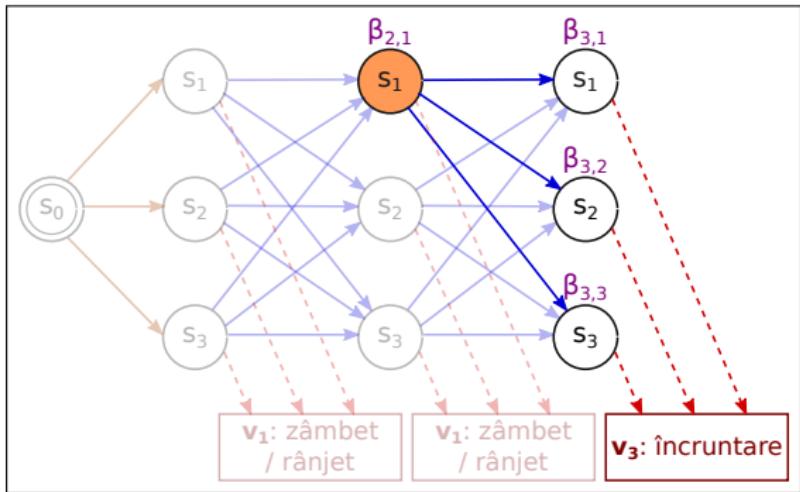
# Calculul variabilelor $\beta$ :



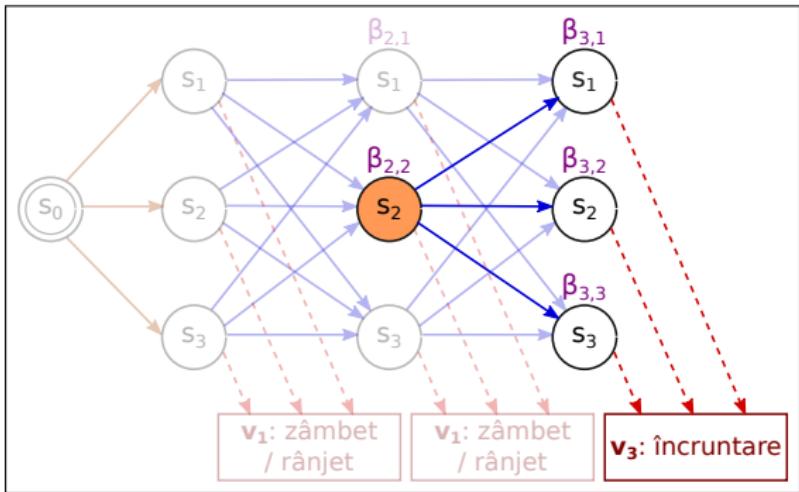
# Calculul variabilelor $\beta$ :



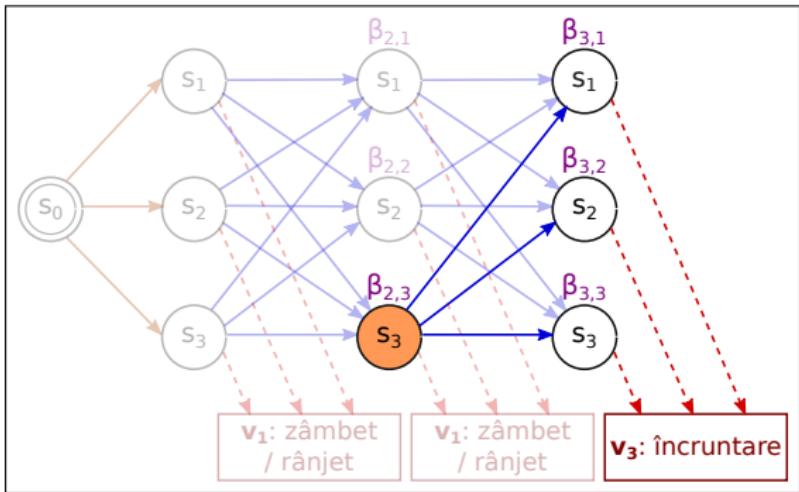
# Calculul variabilelor $\beta$ :



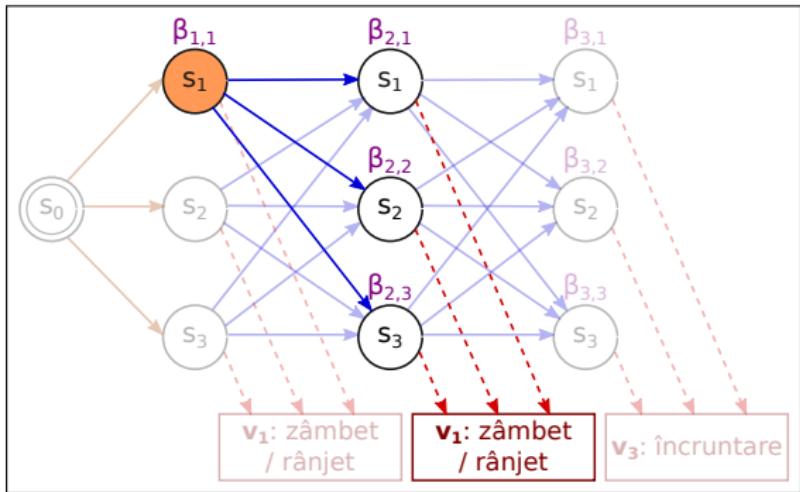
# Calculul variabilelor $\beta$ :



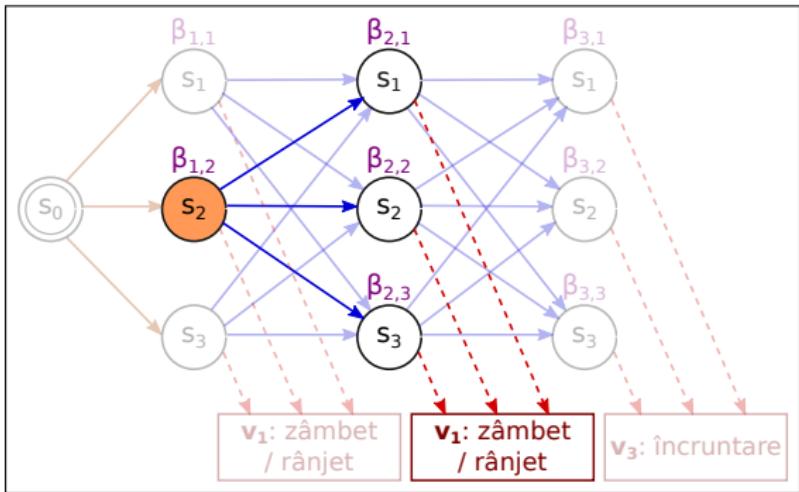
# Calculul variabilelor $\beta$ :



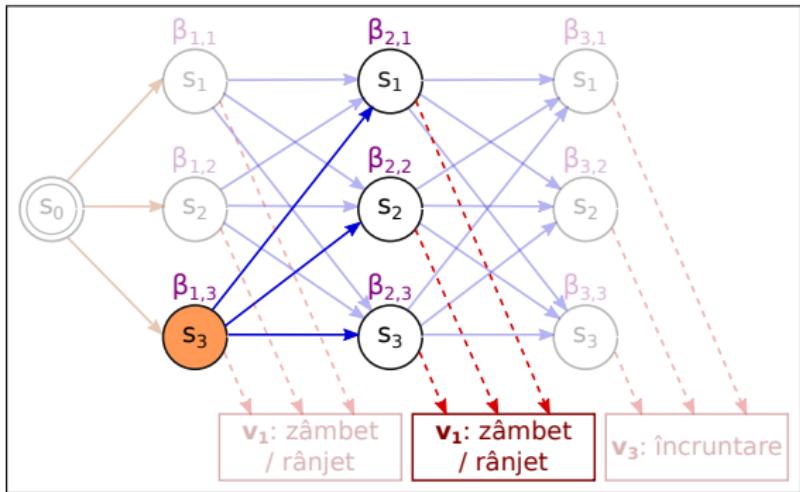
# Calculul variabilelor $\beta$ :



# Calculul variabilelor $\beta$ :



# Calculul variabilelor $\beta$ :





# Algoritmul Forward-Backward

## Algoritmul 1 Calculul variabilelor $\alpha$

```

for i = 1 to N do
2:    $\ddot{\alpha}_{1,i} \leftarrow \pi_i \cdot b_i(o_1)$ 
end for
4:  $c_1 \leftarrow \left( \sum_{i=1}^N \ddot{\alpha}_{1,i} \right)^{-1}$ 
for i = 1 to N do
6:    $\hat{\alpha}_{1,i} \leftarrow c_1 \cdot \ddot{\alpha}_{1,i}$ 
end for
8: for t = 1 to T - 1 do
    for j = 1 to N do
10:       $\ddot{\alpha}_{t+1,j} \leftarrow \left[ \sum_{i=1}^N \hat{\alpha}_{t,i} a_{i,j} \right] b_j(o_{t+1})$ 
    end for
12:       $c_{t+1} \leftarrow \left( \sum_{i=1}^N \ddot{\alpha}_{t+1,i} \right)^{-1}$ 
    for i = 1 to N do
14:       $\hat{\alpha}_{t+1,i} \leftarrow c_{t+1} \cdot \ddot{\alpha}_{t+1,i}$ 
    end for
16: end for

```

## Algoritmul 2 Calculul $P(O|\lambda)$

$$\log P \leftarrow - \sum_{t=1}^T \log c_t$$

## Algoritmul 3 Calculul variabilelor $\beta$

```

for i = 1 to N do
2:    $\hat{\beta}_{T,i} \leftarrow c_T$ 
end for
4: for t = (T - 1) to 1 do
    for i = 1 to N do
6:       $\hat{\beta}_{t,i} \leftarrow \sum_{j=1}^N a_{i,j} b_j(o_{t+1}) \hat{\beta}_{t+1,j} \cdot c_t$ 
    end for
8: end for

```



# A venit vremea să scriem cod

- 1 Faceți o copie a fișierului `forward_backward_disc.m.stub` și denumiți-o `forward_backward_disc.m` (eliminați sufixul `.stub`). Veți implementa funcția:

```
function [logP, Alpha, Beta, Scale] = ...
forward_backward_disc(O,Pi, A, B)
```



# A venit vremea să scriem cod

- 1 Faceți o copie a fișierului `forward_backward_disc.m.stub` și denumiți-o `forward_backward_disc.m` (eliminați sufixul `.stub`). Veți implementa funcția:

```
function [logP, Alpha, Beta, Scale] = ...
forward_backward_disc(O,Pi, A, B)
```

- 2 Completați cele trei secțiuni (calcul *Alpha*, *Beta* și *logP*).

```
1 ...
2 Scale = zeros (1, T); % Scale is an 1 x T matrix
3 Alpha = zeros (T, N); % Alpha is a T x N matrix
4 Beta = ones (T, N); % Beta is a T x N matrix
5 %% Forward variables
6 % alpha_disc_start - Write code below
7
8 % alpha_disc_end - Write code above
9 ...
```

- 3 Folosiți `hmm_test` pentru a vă testa codul.



# Unde suntem...

## 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

## 4 Demo: Recunoașterea Simbolurilor

## 5 Tipuri de MMA

## 6 Discuții și Concluzii



# Problema interpretării

## Problema interpretării 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  $Q_{\text{best}} = [q_{1_{\text{best}}} q_{2_{\text{best}}} \cdots q_{T_{\text{best}}}]$  care să dea un înțeles observațiilor? Cum descoperim partea ascunsă a modelului?



# Problema interpretării

## Problema interpretării 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  $Q_{\text{best}} = [q_{1_{\text{best}}} q_{2_{\text{best}}} \cdots q_{T_{\text{best}}}]$  care să dea un înțeles observațiilor? Cum descoperim partea ascunsă a modelului?

- Răspunsul depinde de criteriul cu care alegem *cea mai bună* secvență



# Problema interpretării

## Problema interpretării 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  $Q_{\text{best}} = [q_{1_{\text{best}}} q_{2_{\text{best}}} \cdots q_{T_{\text{best}}}]$  care să dea un înțeles observațiilor? Cum descoperim partea ascunsă a modelului?

- Răspunsul depinde de criteriul cu care alegem *cea mai bună* secvență
  - **secvența celor mai probabile stări**  $q_{t_{\text{best}}}$  (luate individual), date fiind modelul și secvența observată:  

$$q_{t_{\text{best}}} = \underset{s_i}{\operatorname{argmax}} P(q_t = s_i | O, \lambda)$$
  - **cea mai probabilă secvență de stări**  $Q$  (per ansamblu), date fiind modelul și secvența observată:  

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



# Secvența celor mai probabile stări

$$\text{Notație: } P(q_t = s_i | O, \lambda) = \gamma_{t,i} = \frac{\alpha_{t,i} \beta_{t,i}}{\sum_{j=1}^N \alpha_{t,j} \beta_{t,j}}$$

- Este un criteriu satisfăcător?



## Secvența celor mai probabile stări

Notăție:  $P(q_t = s_i | O, \lambda) = \gamma_{t,i} = \frac{\alpha_{t,i}\beta_{t,i}}{\sum_{j=1}^N \alpha_{t,j}\beta_{t,j}}$

- Este un criteriu satisfăcător?
- **NU!**

Pot exista  $q_t$  și  $q_{t+1}$  astfel încât  $a_{q_t, q_{t+1}} = 0$



# Algoritmul Viterbi

- Criteriu care ia în considerare distribuțiile de probabilitate ale tranzițiilor între stări
- Cea mai bună cale:  $Q_{\text{best}} = [q_{1_{\text{best}}} q_{2_{\text{best}}} \cdots q_{T_{\text{best}}}]$

$$Q_{\text{best}} = \underset{Q}{\operatorname{argmax}} P(Q|O, \lambda) = \underset{Q}{\operatorname{argmax}} P(Q, O|\lambda) \quad (6)$$

- **Algoritmul Viterbi** - programare dinamică [Vit67]
- Vom introduce variabilele  $\delta$ .



# Variabilele $\delta$ - intuiție

## Întrebare

Dacă  $Q = [q_1, q_2, \dots, q_{t-1}, q_t]$  este cea mai bună secvență care explică  $O = [o_1, o_2, \dots, o_{t-1}, o_t]$ , atunci putem afirma că  $Q[1 : t - 1]$  este cea mai bună secvență de stări care explică  $O[1 : t - 1]$ ?



# Variabilele $\delta$ - intuiție

## Întrebare

Dacă  $Q = [q_1, q_2, \dots, q_{t-1}, q_t]$  este cea mai bună secvență care explică  $O = [o_1, o_2, \dots, o_{t-1}, o_t]$ , atunci putem afirma că  $Q[1 : t - 1]$  este cea mai bună secvență de stări care explică  $O[1 : t - 1]$ ?

- NU!

$$\begin{aligned} P(q_1 = s_{i_1}, q_2 = s_{i_2}, \dots, q_{t-1} = s_{i_{t-1}}, q_t = s_{i_t} | O, \lambda) &= \\ P(q_1, q_2, \dots, q_{t-1} | O, \lambda) \cdot P(q_t = s_{i_t} | q_{t-1} = s_{i_{t-1}}, \lambda) \cdot P(o_t | q_t = s_{i_t}, \lambda) \end{aligned}$$



# Variabilele $\delta$

- Vom numi variabile  $\delta$ :

$$\delta_{t,i} = \max_{q_1, \dots, q_{t-1}} P([q_1 q_2 \dots q_{t-1} s_i], [o_1, o_2, \dots, o_t] | \lambda) \quad (26)$$

- $\delta_{t,i}$  - cea mai mare probabilitate a unei secvențe de stări de lungime  $t$  care ajunge în  $s_i$  și explică primele  $t$  valori observate

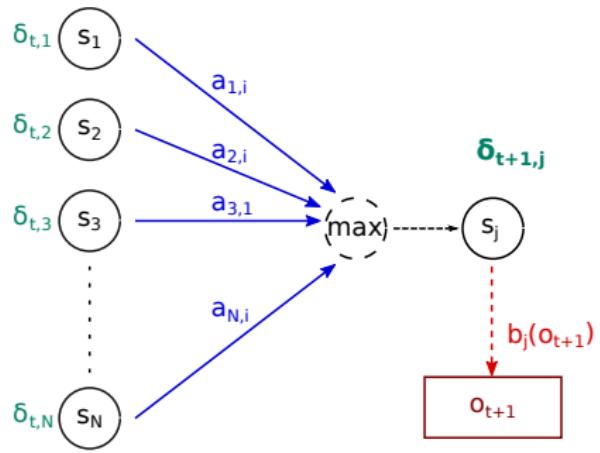
# Variabilele $\delta$

- Vom numi variabile  $\delta$ :

$$\delta_{t,i} = \max_{q_1, \dots, q_{t-1}} P([q_1 q_2 \dots q_{t-1} s_i], [o_1, o_2, \dots, o_t] | \lambda) \quad (26)$$

- $\delta_{t,i}$  - cea mai mare probabilitate a unei secvențe de stări de lungime  $t$  care ajunge în  $s_i$  și explică primele  $t$  valori observate
- relația dintre variabilele  $\delta$ :

$$\delta_{t+1,j} = [\max_i \delta_{t,i} \cdot a_{i,j}] \cdot b_j(o_{t+1}) \quad (27)$$





# Algoritmul Viterbi - Privire de ansamblu

## Pași înainte

Calculăm cea mai mare probabilitate de a ajunge în starea  $s_i$  la momentul  $t$  pe baza celor mai mari probabilități de a fi ajuns în toate stările  $s_j$  la  $t - 1$ .



# Algoritmul Viterbi - Privire de ansamblu

## Pași înainte

Calculăm cea mai mare probabilitate de a ajunge în starea  $s_i$  la momentul  $t$  pe baza celor mai mari probabilități de a fi ajuns în toate stările  $s_j$  la  $t - 1$ .

## La final ( $t = T$ )

Starea finală este acea stare  $s_i$  cu cea mai mare probabilitate de a ajunge la ea după observarea tuturor valorilor.



# Algoritmul Viterbi - Privire de ansamblu

## Pași înainte

Calculăm cea mai mare probabilitate de a ajunge în starea  $s_i$  la momentul  $t$  pe baza celor mai mari probabilități de a fi ajuns în toate stările  $s_j$  la  $t - 1$ .

## La final ( $t = T$ )

Starea finală este acea stare  $s_i$  cu cea mai mare probabilitate de a ajunge la ea după observarea tuturor valorilor.

## Pași înapoi

Refacem *cea mai bună cale* alegând pentru fiecare moment  $t$  starea care a dus la probabilitatea maximă pentru starea de la  $t + 1$ .



# Algoritmul Viterbi (I)

1 Inițializare:

$$\begin{aligned}\delta_{1,i} &= \pi_i b_i(o_1), \quad 1 \leq i \leq N \\ \psi_{1,i} &= 0\end{aligned}\tag{28}$$

2 Recursivitate:

$$\begin{aligned}\delta_{t,j} &= \left[ \max_i \delta_{t-1,i} \cdot a_{i,j} \right] \cdot b_j(o_t) \quad 2 \leq t \leq T, 1 \leq j \leq N \\ \psi_{t,i} &= \operatorname{argmax}_i \delta_{t-1,i} \cdot a_{i,j} \quad 2 \leq t \leq T, 1 \leq j \leq N\end{aligned}\tag{29}$$



# Algoritmul Viterbi (II)

3 Terminare:

$$\begin{aligned} P(Q_{\text{best}} | O, \lambda) &= \max_i \delta_{T,i} \\ q_{T_{\text{best}}} &= \operatorname{argmax}_i \delta_{T,i} \end{aligned} \tag{30}$$

4 Backtracking:

$$q_{t_{\text{best}}} = \psi_{t+1}(q_{t+1_{\text{best}}}), \quad t=T-1, T-2, \dots, 1 \tag{31}$$



# Probleme numerice

- Cine este  $P(Q_{\text{best}}|O, \lambda)$ ?

$$P(Q_{\text{best}}|O, \lambda) = \delta_{T,i_T}$$

$$P(Q_{\text{best}}|O, \lambda) = \delta_{T-1,i_{T-1}} \cdot a_{i_{T-1},i_T} \cdot b_T(o_T)$$

$$P(Q_{\text{best}}|O, \lambda) = \delta_{T-2,i_{T-2}} \cdot a_{i_{T-2},i_{T-1}} \cdot b_{T-1}(o_{T-1}) \cdot a_{i_{T-1},j_T} \cdot b_T(o_T)$$

$$P(Q_{\text{best}}|O, \lambda) = \prod \dots$$

- Cum putem evita apropierea rapidă de zero?



# Probleme numerice

- Cine este  $P(Q_{\text{best}}|O, \lambda)$ ?

$$P(Q_{\text{best}}|O, \lambda) = \delta_{T,i_T}$$

$$P(Q_{\text{best}}|O, \lambda) = \delta_{T-1,i_{T-1}} \cdot a_{i_{T-1},i_T} \cdot b_T(o_T)$$

$$P(Q_{\text{best}}|O, \lambda) = \delta_{T-2,i_{T-2}} \cdot a_{i_{T-2},i_{T-1}} \cdot b_{T-1}(o_{T-1}) \cdot a_{i_{T-1},i_T} \cdot b_T(o_T)$$

$$P(Q_{\text{best}}|O, \lambda) = \prod \dots$$

- Cum putem evita apropierea rapidă de zero?
- Calculăm  $\log(P)$

$$\log(P(Q_{\text{best}}|O, \lambda)) = \log(\delta_{T,i_T})$$

$$\log(P(Q_{\text{best}}|O, \lambda)) = \log(\delta_{T-1,i_{T-1}}) + \log(a_{i_{T-1},i_T}) + \log(b_T(o_T))$$

$$\log(P(Q_{\text{best}}|O, \lambda)) = \log(\delta_{T-2,i_{T-2}}) + \log(a_{i_{T-2},i_{T-1}}) + \log(b_{T-1}(o_{T-1}))$$

$$\log(P(Q_{\text{best}}|O, \lambda)) = \sum \dots$$



# Probleme numerice - Rezolvare

- Notație:

$$\phi_{t,i} = \max_{q_1, \dots, q_{t-1}} \log(P(q_1, \dots, q_{t-1}, q_t = s_i, o_1, \dots, o_t | \lambda)) = \log(\delta_{t,i})$$

- Matricele  $\log(\Pi)$ ,  $\log(A)$  și  $\log(B)$  pot fi precalculate.



# Algoritmul Viterbi - $\log(P)$ (I)

1 Inițializare:

$$\begin{aligned}\phi_{1,i} &= \log(\pi_i) + \log(b_i(o_1)), \quad 1 \leq i \leq N \\ \psi_{1,i} &= 0\end{aligned}\tag{32}$$

2 Recursivitate:

$$\begin{aligned}\phi_{t,j} &= [\max_i \phi_{t-1,i} + \log(a_{i,j})] + \log(b_j(o_t)) \quad 2 \leq t \leq T, 1 \leq j \leq N \\ \psi_{t,i} &= \operatorname{argmax}_i \phi_{t-1,i} + \log(a_{i,j}) \quad 2 \leq t \leq T, 1 \leq j \leq N\end{aligned}\tag{33}$$



## Algoritmul Viterbi - $\log(P)$ (II)

### 3 Terminare:

$$\begin{aligned}\log(P(Q_{\text{best}} | O, \lambda)) &= \max_i \phi_{T,i} \\ q_{T_{\text{best}}} &= \operatorname{argmax}_i \phi_{T,i}\end{aligned}\tag{34}$$

### 4 Backtracking:

$$q_{t_{\text{best}}} = \psi_{t+1}(q_{t+1_{\text{best}}}), \quad t=T-1, T-2, \dots, 1\tag{35}$$



## Înapoi la exemplu

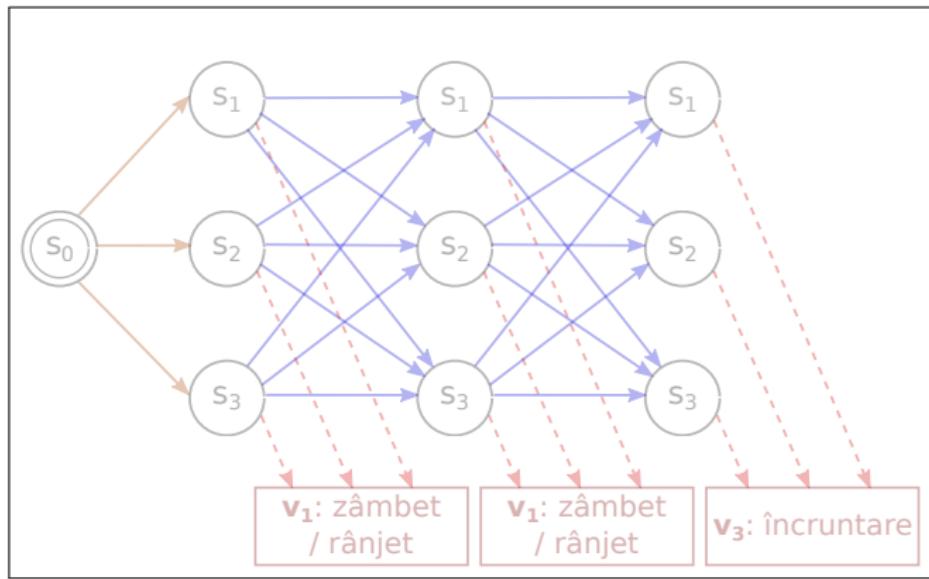
- Să reluăm exemplul cu robotul.
- Robotul a recunoscut secvența de gesturi:  
**(zâmbet | rânjet) → (zâmbet | rânjet) → încruntare**
- Am stabilit:  $P(O|\lambda^1) > P(O|\lambda^2)$



## Înapoi la exemplu

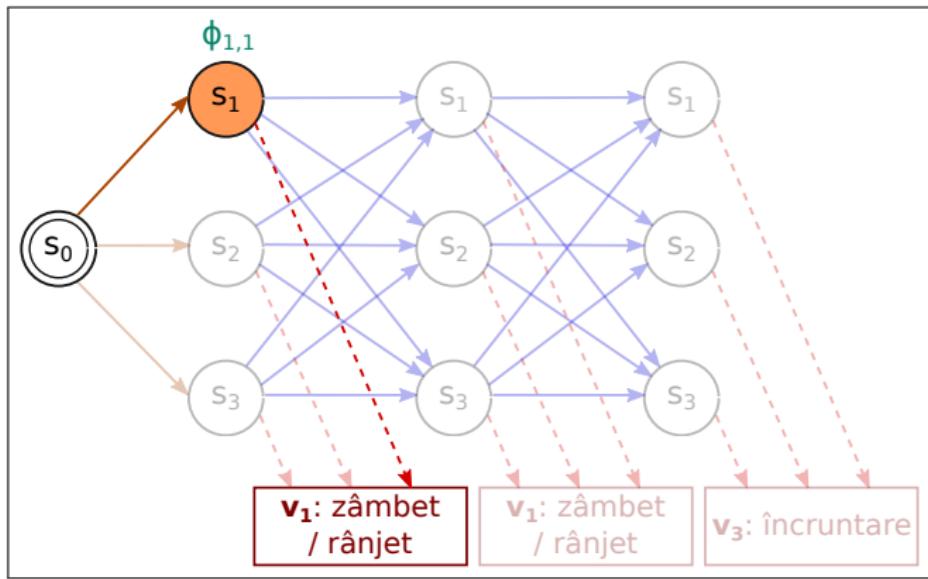
- Să reluăm exemplul cu robotul.
- Robotul a recunoscut secvența de gesturi:  
 $(zâmbet \mid rânjet) \longrightarrow (zâmbet \mid rânjet) \longrightarrow \text{încruntare}$
- Am stabilit:  $P(O|\lambda^1) > P(O|\lambda^2)$   
Deci, robotul se confruntă [ , probabil] cu un morocănos!
- A doua întrebare: **Prin ce stări a trecut acesta?**

# Jovialul



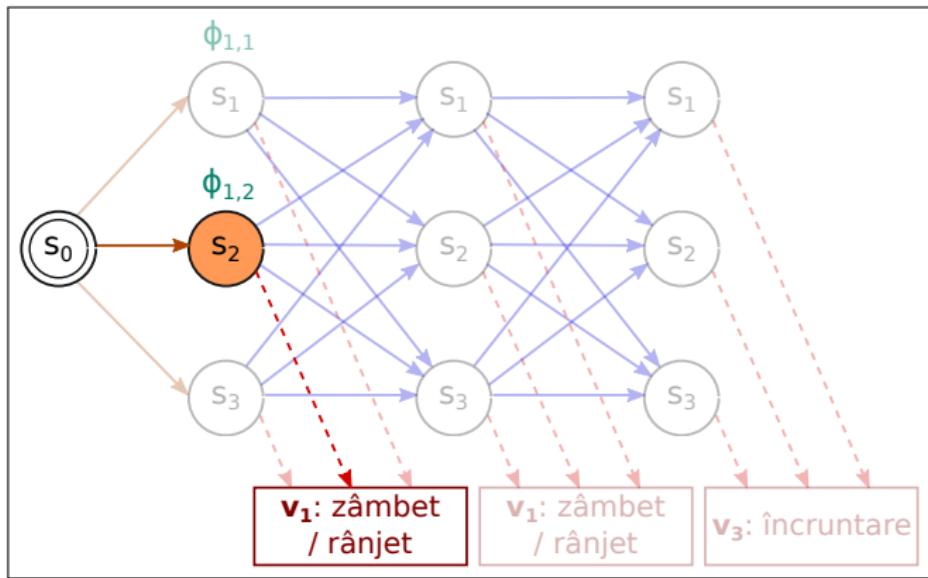
$$\Phi = \left[ \quad \right] \quad \Psi = \left[ \quad \right] \quad Q_{best} = [ \quad ]$$

# Jovialul



$$\Phi = \begin{bmatrix} -2.1202 \end{bmatrix} \quad \Psi = \begin{bmatrix} 0 \end{bmatrix} \quad Q_{best} = [ ]$$

# Jovialul

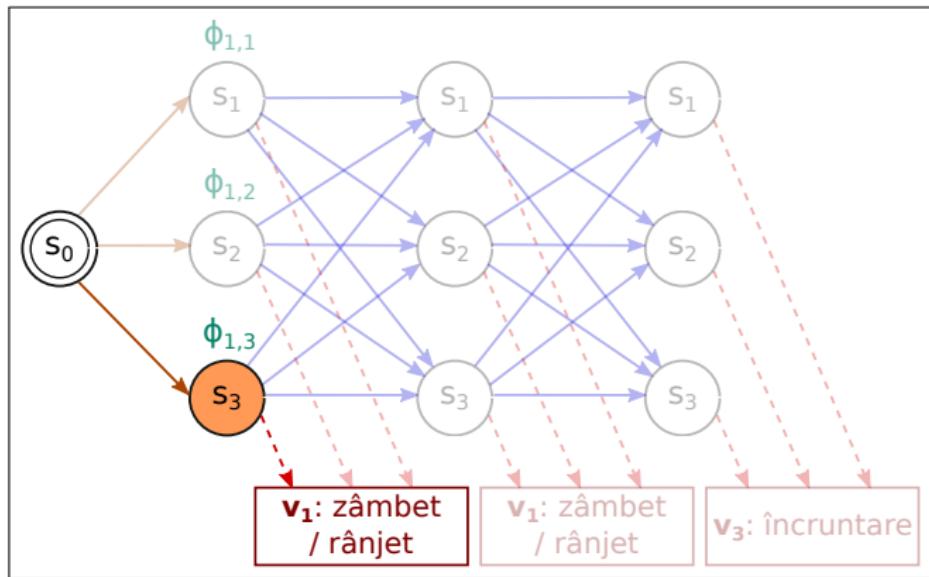


$$\phi_{1,2} = \log(\pi_2) + \log(b_{2,1})$$

$$\psi_{1,2} = 0$$

$$\Phi = \begin{bmatrix} -2.1202 & -Inf \end{bmatrix} \quad \Psi = \begin{bmatrix} 0 & 0 \end{bmatrix} \quad Q_{best} = [ ]$$

# Jovialul

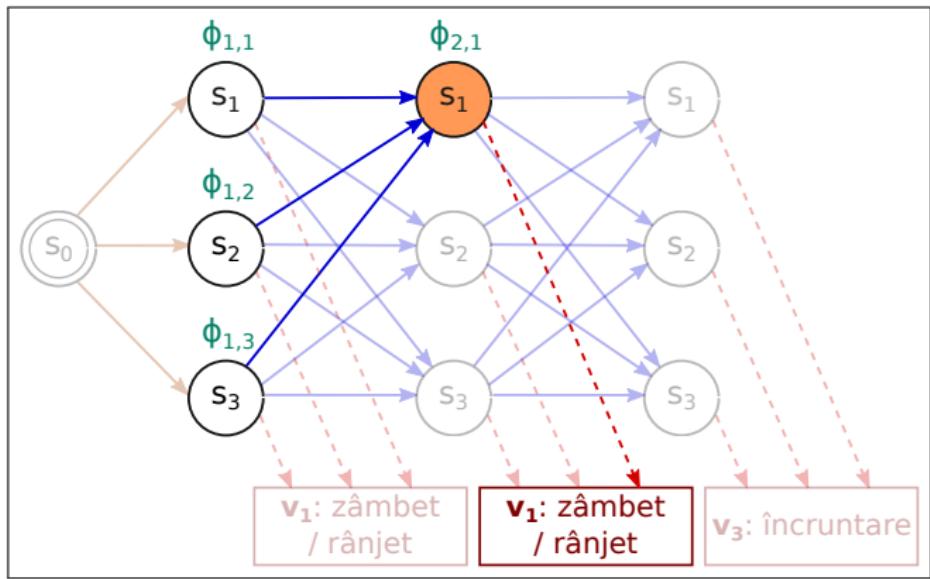


$$\phi_{1,3} = \log(\pi_3) + \log(b_{3,1})$$

$$\psi_{1,3} = 0$$

$$\Phi = \begin{bmatrix} -2.1202 & -\text{Inf} & -2.1202 \end{bmatrix} \quad \Psi = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \quad Q_{best} = [ ]$$

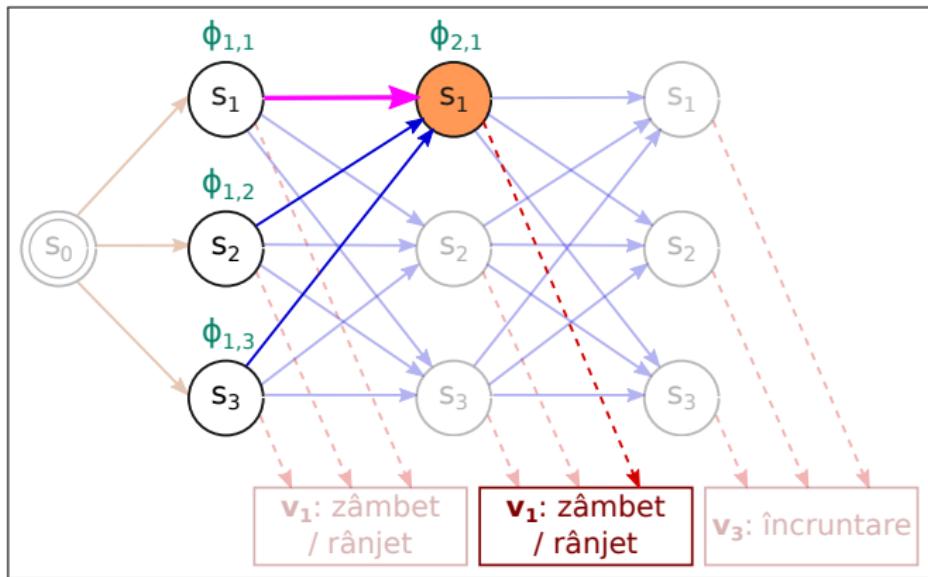
# Jovialul



$$\phi_{1,3} = \max \left\{ \begin{array}{l} \log(\phi_{1,1}) + \log(a_{1,1}), \\ \log(\phi_{2,1}) + \log(a_{2,1}), \\ \log(\phi_{1,3}) + \log(a_{3,1}) \end{array} \right\} + \log(b_{1,1})$$

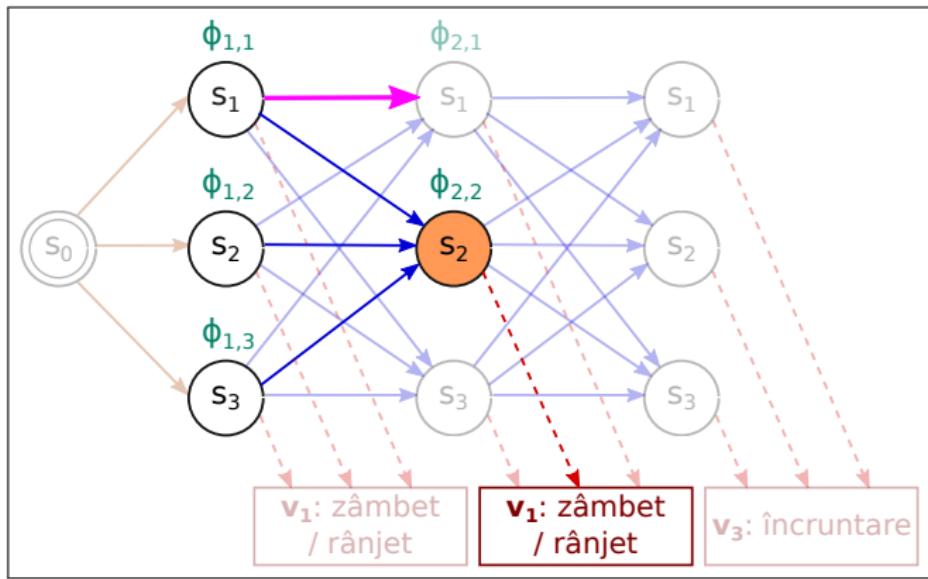
$$\Phi = \begin{bmatrix} -2.1202 & -Inf & -2.1202 \\ -3.7297 & & \end{bmatrix} \quad \Psi = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \quad Q_{best} = [ \quad ]$$

# Jovialul



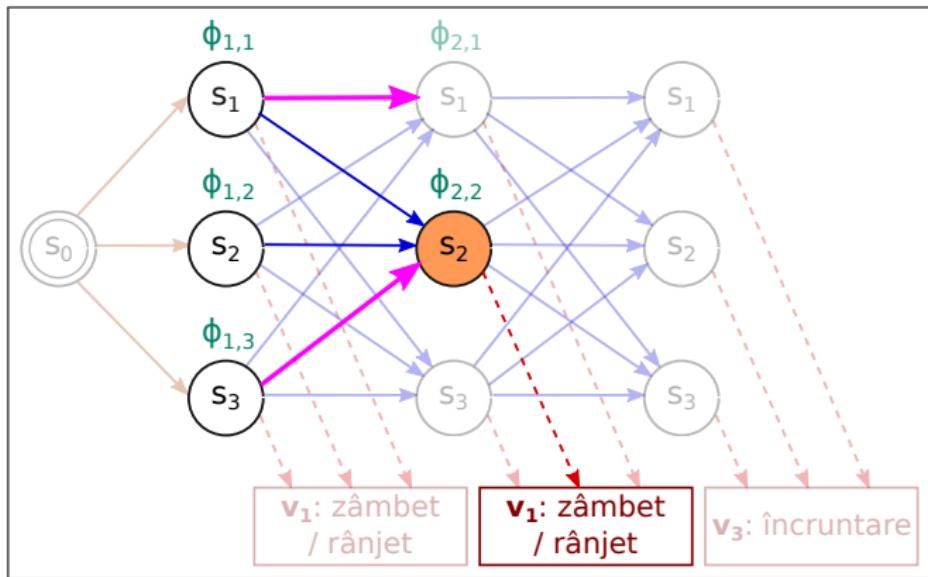
$$\Phi = \begin{bmatrix} -2.1202 & -\text{Inf} & -2.1202 \\ -3.7297 & & \end{bmatrix} \quad \Psi = \begin{bmatrix} 0 & 0 & 0 \\ 1 & & \end{bmatrix} \quad Q_{best} = [ ]$$

# Jovialul



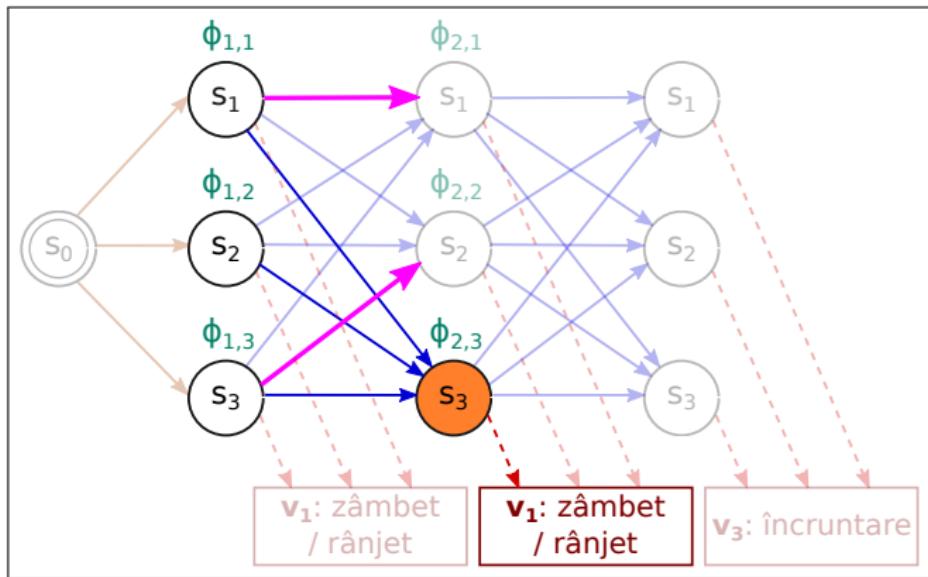
$$\Phi = \begin{bmatrix} -2.1202 & -Inf & -2.1202 \\ -3.7297 & -Inf & \end{bmatrix} \quad \Psi = \begin{bmatrix} 0 & 0 & 0 \\ 1 & \end{bmatrix} \quad Q_{best} = [ ]$$

# Jovialul



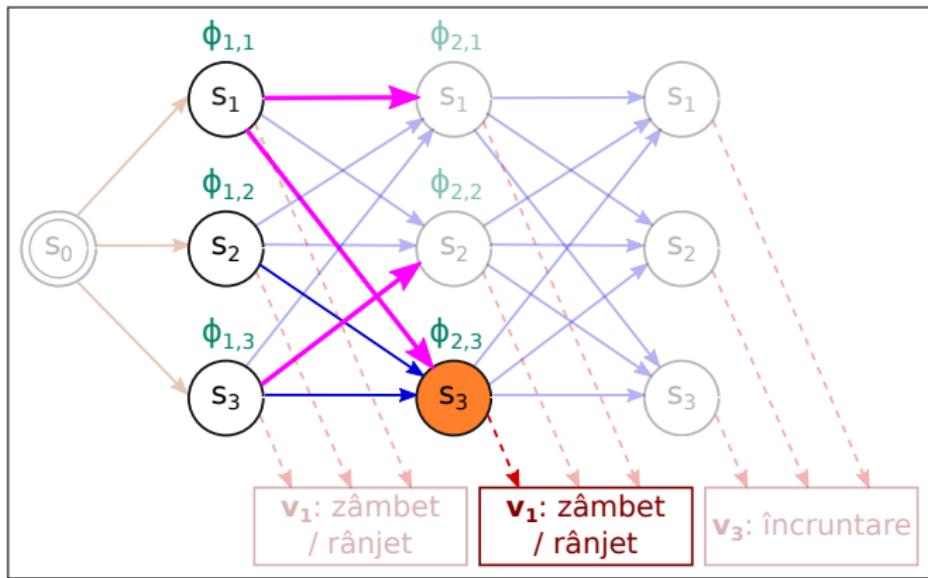
$$\Phi = \begin{bmatrix} -2.1202 & -\text{Inf} & -2.1202 \\ -3.7297 & -\text{Inf} & \end{bmatrix} \quad \Psi = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 3 & \end{bmatrix} \quad Q_{best} = [ ]$$

# Jovialul



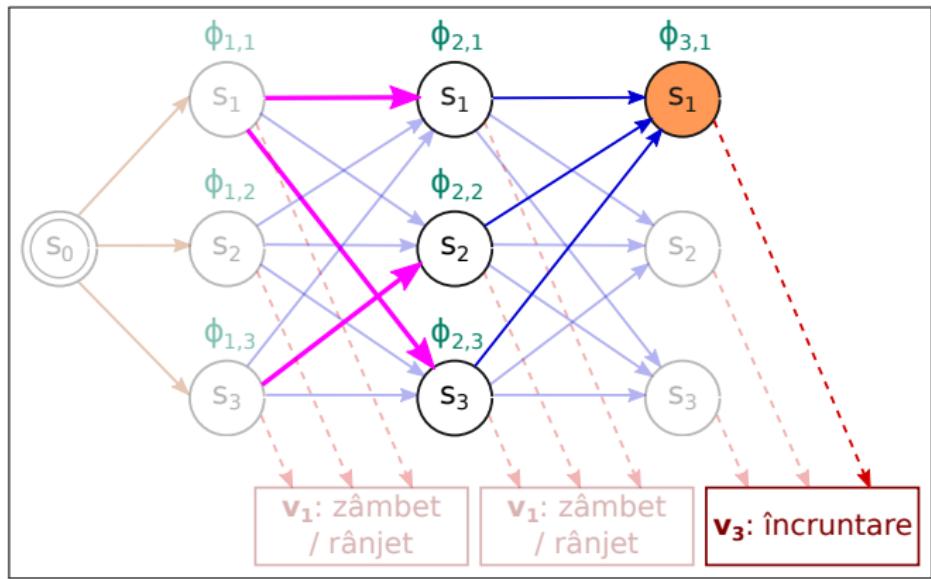
$$\Phi = \begin{bmatrix} -2.1202 & -\text{Inf} & -2.1202 \\ -3.7297 & -\text{Inf} & -4.2205 \end{bmatrix} \quad \Psi = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 3 & 0 \end{bmatrix} \quad Q_{best} = [ ]$$

# Jovialul



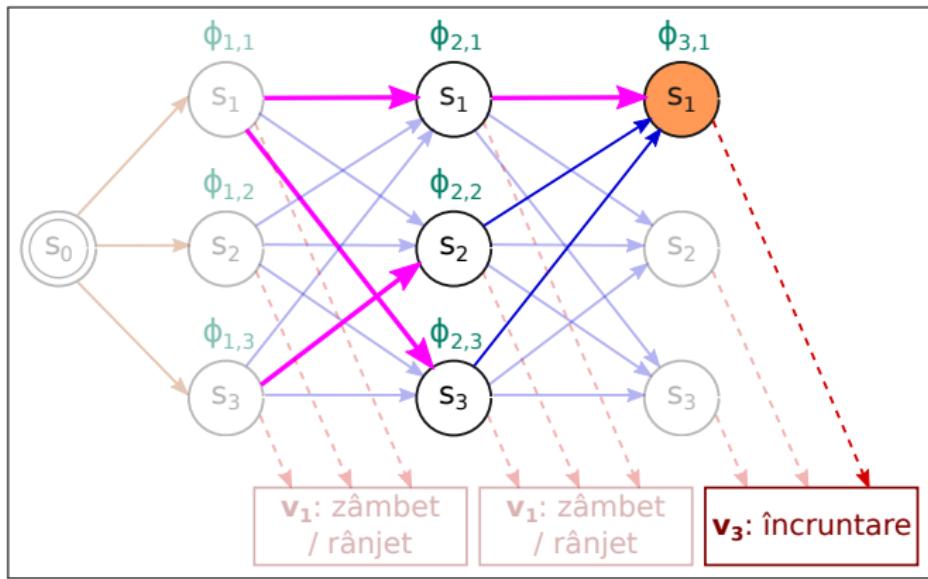
$$\Phi = \begin{bmatrix} -2.1202 & -\text{Inf} & -2.1202 \\ -3.7297 & -\text{Inf} & -4.2205 \end{bmatrix} \quad \Psi = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 3 & 1 \end{bmatrix} \quad Q_{best} = [ ]$$

# Jovialul



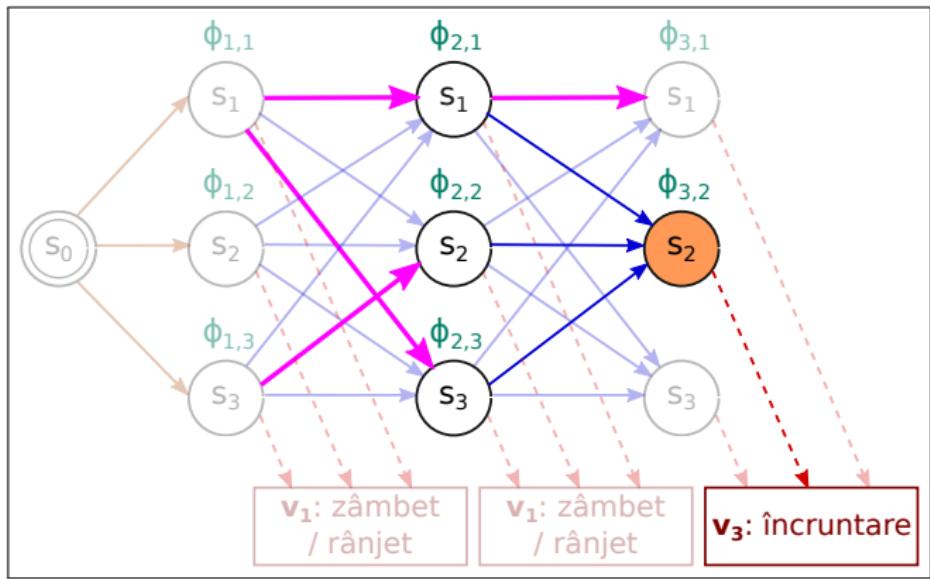
$$\Phi = \begin{bmatrix} -2.1202 & -\text{Inf} & -2.1202 \\ -3.7297 & -\text{Inf} & -4.2205 \\ -6.7254 & & \end{bmatrix} \quad \Psi = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 3 & 1 \end{bmatrix} \quad Q_{best} = [ ]$$

# Jovialul



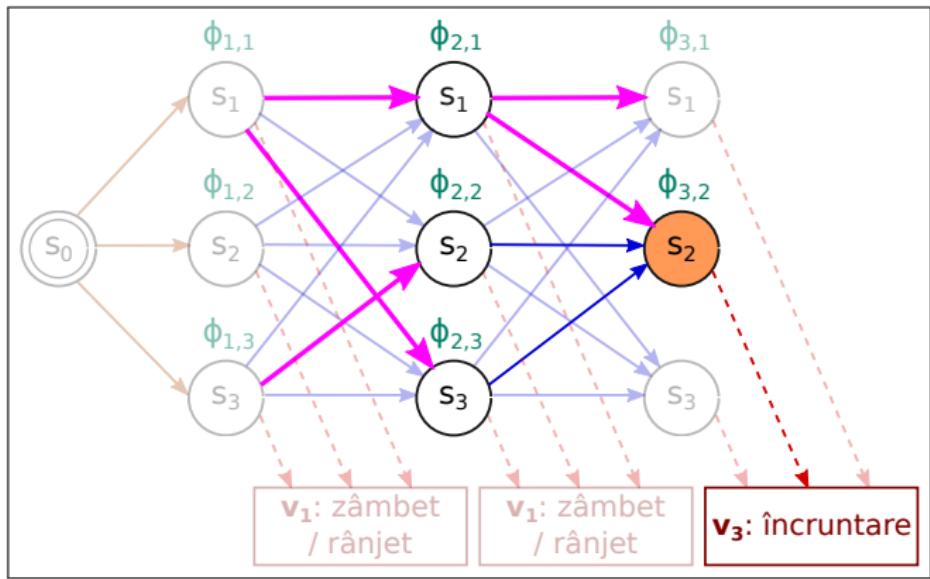
$$\Phi = \begin{bmatrix} -2.1202 & -\text{Inf} & -2.1202 \\ -3.7297 & -\text{Inf} & -4.2205 \\ -6.7254 & & \end{bmatrix} \quad \Psi = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 3 & 1 \\ 1 & & \end{bmatrix} \quad Q_{best} = [ ]$$

# Jovialul



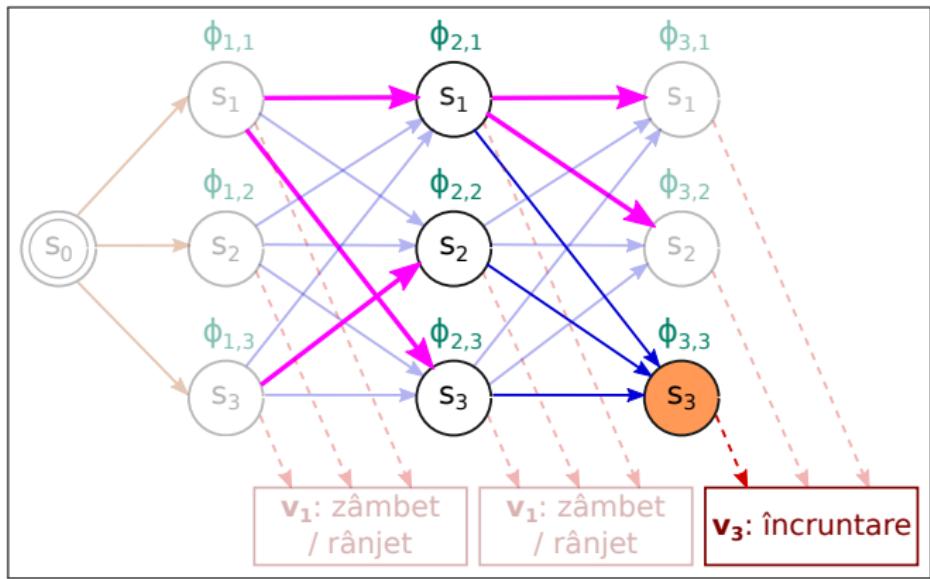
$$\Phi = \begin{bmatrix} -2.1202 & -Inf & -2.1202 \\ -3.7297 & -Inf & -4.2205 \\ -6.7254 & -5.1567 \end{bmatrix} \quad \Psi = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 3 & 1 \\ 1 \end{bmatrix} \quad Q_{best} = [ ]$$

# Jovialul



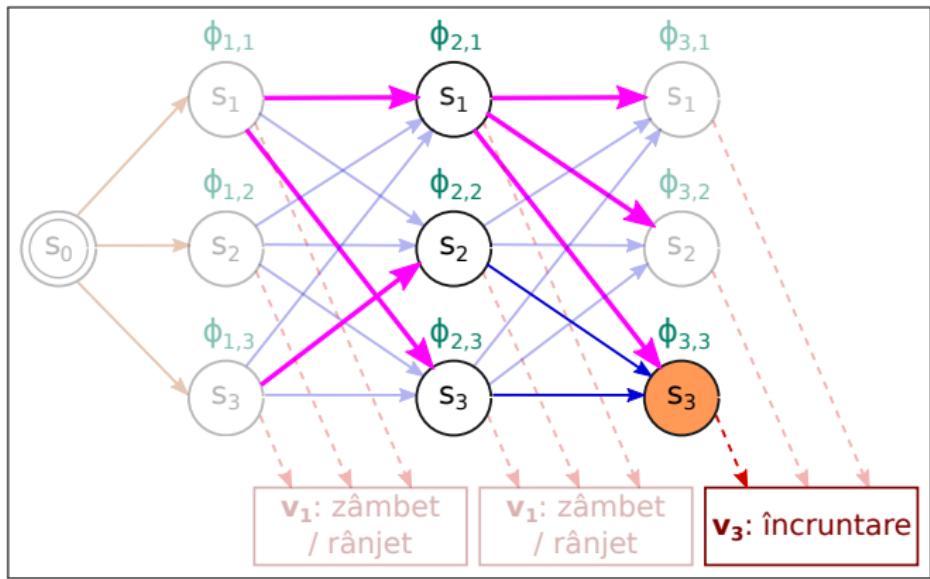
$$\Phi = \begin{bmatrix} -2.1202 & -Inf & -2.1202 \\ -3.7297 & -Inf & -4.2205 \\ -6.7254 & -5.1567 \end{bmatrix} \quad \Psi = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 3 & 1 \\ 1 & 1 \end{bmatrix} \quad Q_{best} = [ ]$$

# Jovialul



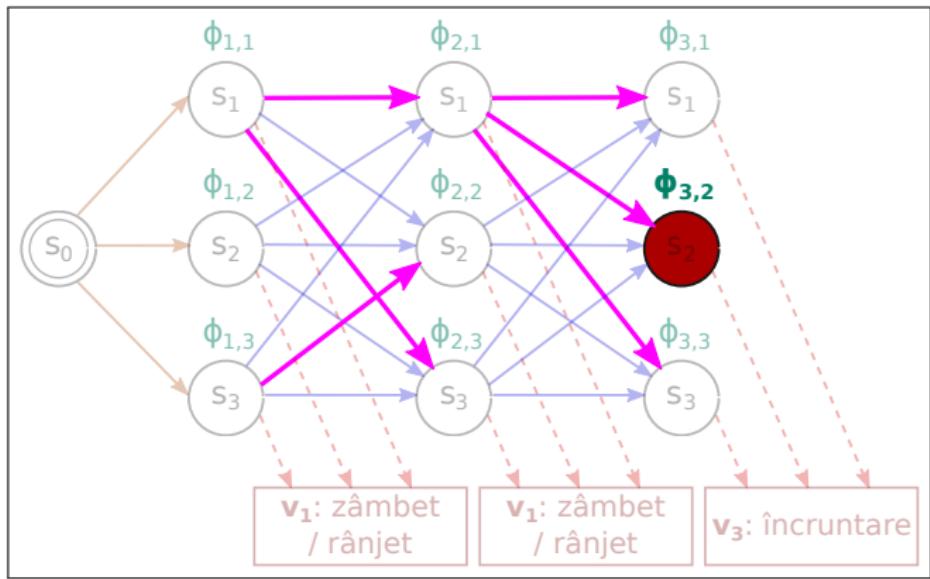
$$\Phi = \begin{bmatrix} -2.1202 & -\text{Inf} & -2.1202 \\ -3.7297 & -\text{Inf} & -4.2205 \\ -6.7254 & -5.1567 & -5.3391 \end{bmatrix} \quad \Psi = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 3 & 1 \\ 1 & 1 & \end{bmatrix} \quad Q_{best} = [ ]$$

# Jovialul



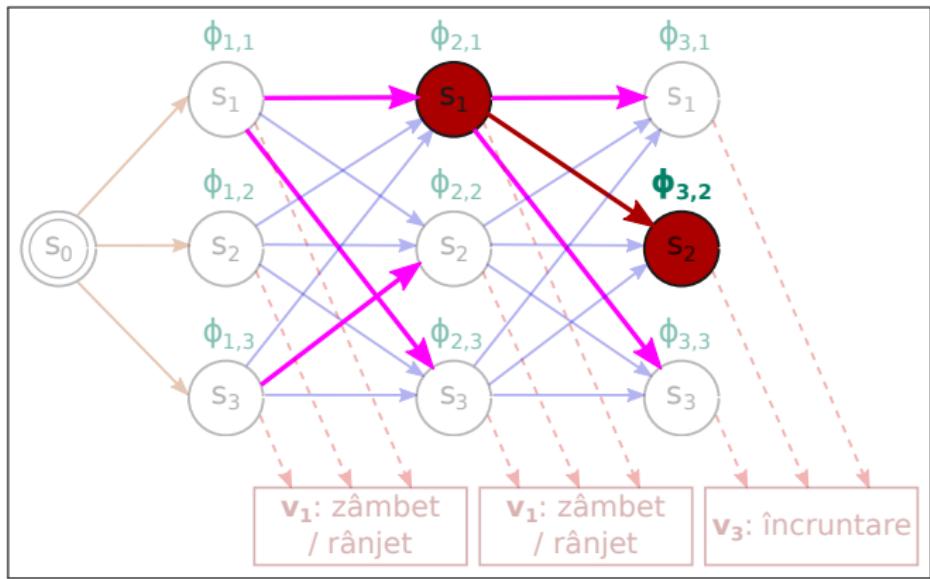
$$\Phi = \begin{bmatrix} -2.1202 & -\text{Inf} & -2.1202 \\ -3.7297 & -\text{Inf} & -4.2205 \\ -6.7254 & -5.1567 & -5.3391 \end{bmatrix} \quad \Psi = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad Q_{best} = [ ]$$

# Jovialul



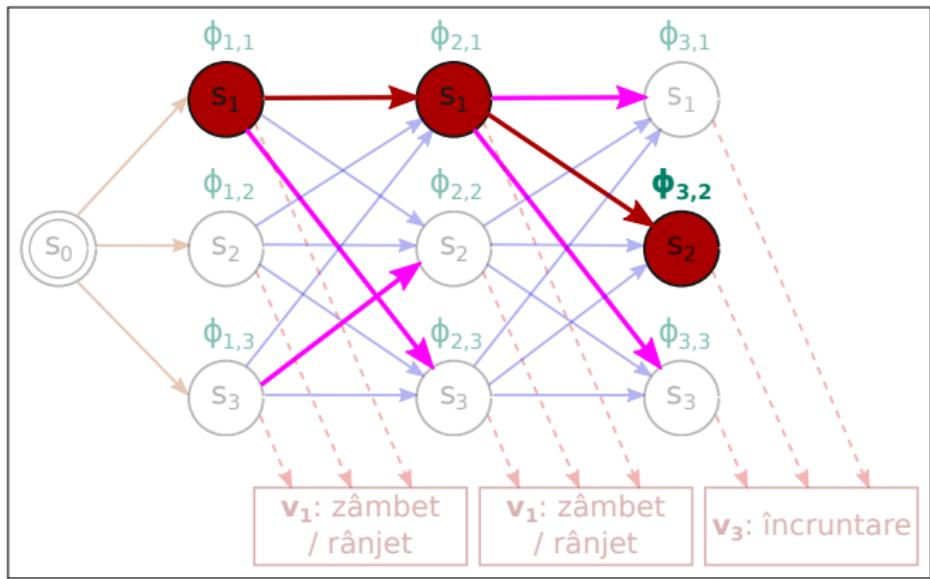
$$\Phi = \begin{bmatrix} -2.1202 & -\text{Inf} & -2.1202 \\ -3.7297 & -\text{Inf} & -4.2205 \\ -6.7254 & -5.1567 & -5.3391 \end{bmatrix} \quad \Psi = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad Q_{best} = [ \quad \quad \quad 2 ]$$

# Jovialul



$$\Phi = \begin{bmatrix} -2.1202 & -\text{Inf} & -2.1202 \\ -3.7297 & -\text{Inf} & -4.2205 \\ -6.7254 & -5.1567 & -5.3391 \end{bmatrix} \quad \Psi = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad Q_{best} = [ \quad 1 \quad 2 ]$$

# Jovialul



$$\Phi = \begin{bmatrix} -2.1202 & -Inf & -2.1202 \\ -3.7297 & -Inf & -4.2205 \\ -6.7254 & -5.1567 & -5.3391 \end{bmatrix} \quad \Psi = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad Q_{best} = [1 \ 1 \ 2]$$



# Algoritmul Viterbi

## Algoritmul 4 Calculul celei mai probabile secvențe $Q_{best}$

```

for  $i = 1$  to  $N$  do
2:    $\phi_{1,i} \leftarrow \log(\pi_i) + \log(b_i(o_1))$ 
     $\psi_{1,i} \leftarrow 0$ 
4: end for
for  $t = 2$  to  $T$  do
6:   for  $j = 1$  to  $N$  do
       $\phi_{t,j} \leftarrow [\max_i \phi_{t-1,i} + \log(a_{i,j})] + \log(b_j(o_t))$ 
8:    $\psi_{t,j} \leftarrow \operatorname{argmax}_i \phi_{t-1,i} + \log(a_{i,j})$ 
    end for
10: end for
     $\log(P(Q_{best}|O, \lambda)) \leftarrow \max_i \phi_{T,i}$ 
12:  $q_{T_{best}} \leftarrow \operatorname{argmax}_i \phi_{T,i}$ 
    for  $t = T - 1$  to  $1$  do
14:    $q_{t_{best}} \leftarrow \psi_{t+1}(q_{t+1_{best}})$ 
    end for

```



# A venit vremea să scriem cod

- 1 Faceți o copie a fișierului `viterbi_disc.m.stub` și denumiți-o `viterbi_disc.m` (eliminați sufixul `.stub`).

Veți implementa funcția:

```
function [logP, Q] = viterbi_disc(O,Pi, A, B)
```



# A venit vremea să scriem cod

- ① Faceți o copie a fișierului `viterbi_disc.m.stub` și denumiți-o `viterbi_disc.m` (eliminați sufixul `.stub`).

Veți implementa funcția:

```
function [logP, Q] = viterbi_disc(O, Pi, A, B)
```

- ② Completați cele două secțiuni.

```

1 %% Recursion
2 % phi_psi_disc-start – Write code below
3
4 % phi_psi_disc-end – Write code above
5 %% logP
6 [logP, Q(T)] = max(Phi(T, :));
7 %% Backtracking to compute the path Q
8 % path_disc-start – Write code below
9
10 % path_disc-end – Write code above

```

- ③ Folosiți `hmm_test` pentru a vă testa codul.



# Unde suntem...

## 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**

## 4 Demo: Recunoașterea Simbolurilor

## 5 Tipuri de MMA

## 6 Discuții și Concluzii



# Învățarea din observații - Amintire

## Problema Estimării (Antrenării) Modelului

Dându-se un set de secvențe observate  $\mathcal{O} = [O_1 O_2 \cdots O_L]$ , cum ajustăm parametrii  $\lambda = (A, B, \Pi)$  ai unui MMA care încearcă să explice cel mai bine acele observații?



# Învățarea din observații - Amintire

## Problema Estimării (Antrenării) Modelului

Dându-se un set de secvențe observate  $\mathcal{O} = [O_1 O_2 \cdots O_L]$ , cum ajustăm parametrii  $\lambda = (A, B, \Pi)$  ai unui MMA care încearcă să explice cel mai bine acele observații?

Secvențele de observații folosite pentru ajustarea parametrilor modelului se numesc secvențe **de antrenare**.

Problema antrenării este esențială - ea permite crearea celor mai bune modele pentru fenomene reale.



# Învățarea din observații - Abordare



# Învățarea din observații - Abordare

## Problemă

Nu se cunoaște o metodă analitică de căutare a parametrilor modelului care *maximizează* probabilitatea secvențelor observate.



# Învățarea din observații - Abordare

## Problemă

Nu se cunoaște o metodă analitică de căutare a parametrilor modelului care *maximizează* probabilitatea secvențelor observate.

## Soluție

Putem totuși găsi  $\lambda = (A, B, \Pi)$ , astfel încât  $\max_{\lambda} P(O|\lambda)$  corespunde unui **maxim local**, utilizând o **procedură iterativă** precum *algoritmul Baum-Welch*. Această metodă este o instanță a *algoritmului EM (Expectation Maximization)* [DLR77] pentru cazul MMA.



# Algoritmul Baum-Welch (I)

Procedura în descriere conceptuală:

- ① Avem MMA  $\lambda = (A, B, \Pi)$  și o secvență observată  $O$



# Algoritmul Baum-Welch (I)

Procedura în descriere conceptuală:

- ① Avem MMA  $\lambda = (A, B, \Pi)$  și o secvență observată  $O$
- ② Calculăm folosindu-ne de parametrii  $\alpha_t(i)$  și  $\beta_t(i)$ 
  - nr. estimat de tranzitii din  $S_i$ , pentru fiecare  $1 \leq i \leq N$
  - nr. estimat de tranzitii din  $S_i$  la  $S_j$ , pentru fiecare  $1 \leq i \leq N, 1 \leq j \leq N$
  - nr. estimat de vizite în  $S_j$  observând simbolul  $v_k$ , pentru fiecare  $1 \leq j \leq N, 1 \leq k \leq M$



# Algoritmul Baum-Welch (I)

Procedura în descriere conceptuală:

- ① Avem MMA  $\lambda = (A, B, \Pi)$  și o secvență observată  $O$
- ② Calculăm folosindu-ne de parametrii  $\alpha_t(i)$  și  $\beta_t(i)$ 
  - nr. estimat de tranzitii din  $S_i$ , pentru fiecare  $1 \leq i \leq N$
  - nr. estimat de tranzitii din  $S_i$  la  $S_j$ , pentru fiecare  $1 \leq i \leq N, 1 \leq j \leq N$
  - nr. estimat de vizite în  $S_j$  observând simbolul  $v_k$ , pentru fiecare  $1 \leq j \leq N, 1 \leq k \leq M$
- ③ Dacă modelul este corect ne așteptăm ca
  - (a)  $\Pi_i = \text{nr. estimat de vizite în starea } S_i \text{ la momentul } (t=1) = \bar{\Pi}_i$
  - (b)  $a_{i,j} = \frac{\text{nr. estimat de tranzitii din } S_i \text{ la } S_j}{\text{nr. estimat de tranzitii din } S_i} = \bar{a}_{i,j}$
  - (c)  $b_{j,k} = \frac{\text{nr. estimat de vizite în } S_j \text{ observând simbolul } v_k}{\text{nr. estimat de vizite în } S_j} = \bar{b}_{j,k}$



# Algoritmul Baum-Welch (I)

Procedura în descriere conceptuală:

- ① Avem MMA  $\lambda = (A, B, \Pi)$  și o secvență observată  $O$
- ② Calculăm folosindu-ne de parametrii  $\alpha_t(i)$  și  $\beta_t(i)$ 
  - nr. estimat de tranzitii din  $S_i$ , pentru fiecare  $1 \leq i \leq N$
  - nr. estimat de tranzitii din  $S_i$  la  $S_j$ , pentru fiecare  $1 \leq i \leq N, 1 \leq j \leq N$
  - nr. estimat de vizite în  $S_j$  observând simbolul  $v_k$ , pentru fiecare  $1 \leq j \leq N, 1 \leq k \leq M$
- ③ Dacă modelul este corect ne așteptăm ca
  - (a)  $\Pi_i = \text{nr. estimat de vizite în starea } S_i \text{ la momentul } (t=1) = \bar{\Pi}_i$
  - (b)  $a_{i,j} = \frac{\text{nr. estimat de tranzitii din } S_i \text{ la } S_j}{\text{nr. estimat de tranzitii din } S_i} = \bar{a}_{i,j}$
  - (c)  $b_{j,k} = \frac{\text{nr. estimat de vizite în } S_j \text{ observând simbolul } v_k}{\text{nr. estimat de vizite în } S_j} = \bar{b}_{j,k}$
- ④ Vedem că raporturile calculate din presupunerea noastră explică mai bine observația decât parametrii anteriori, i.e.  $P(O|\bar{\lambda}) > P(O|\lambda)$



# Algoritmul Baum-Welch (I)

Procedura în descriere conceptuală:

- ① Avem MMA  $\lambda = (A, B, \Pi)$  și o secvență observată  $O$
- ② Calculăm folosindu-ne de parametrii  $\alpha_t(i)$  și  $\beta_t(i)$ 
  - nr. estimat de tranzitii din  $S_i$ , pentru fiecare  $1 \leq i \leq N$
  - nr. estimat de tranzitii din  $S_i$  la  $S_j$ , pentru fiecare  $1 \leq i \leq N, 1 \leq j \leq N$
  - nr. estimat de vizite în  $S_j$  observând simbolul  $v_k$ , pentru fiecare  $1 \leq j \leq N, 1 \leq k \leq M$
- ③ Dacă modelul este corect ne așteptăm ca
  - (a)  $\Pi_i = \text{nr. estimat de vizite în starea } S_i \text{ la momentul } (t=1) = \bar{\Pi}_i$
  - (b)  $a_{i,j} = \frac{\text{nr. estimat de tranzitii din } S_i \text{ la } S_j}{\text{nr. estimat de tranzitii din } S_i} = \bar{a}_{i,j}$
  - (c)  $b_{j,k} = \frac{\text{nr. estimat de vizite în } S_j \text{ observând simbolul } v_k}{\text{nr. estimat de vizite în } S_j} = \bar{b}_{j,k}$
- ④ Vedem că raporturile calculate din presupunerea noastră explică mai bine observația decât parametrii anteriori, i.e.  $P(O|\bar{\lambda}) > P(O|\lambda)$
- ⑤ Atunci repetăm procesul până ce suntem mulțumiți (convergență):  $P(O|\bar{\lambda}) - P(O|\lambda) \leq \epsilon$



# Algoritmul Baum-Welch (II)



## Algoritmul Baum-Welch (II)

Definim întâi niște variabile auxiliare:

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

Probabilitatea de a fi în starea  $s_i$  la momentul  $t$  și în starea  $s_j$  la momentul  $t + 1$ , condiționat de parametrii modelului curent și secvența observată.



## Algoritmul Baum-Welch (II)

Definim întâi niște variabile auxiliare:

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

Probabilitatea de a fi în starea  $s_i$  la momentul  $t$  și în starea  $s_j$  la momentul  $t + 1$ , condiționat de parametrii modelului curent și secvența observată.

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

Probabilitatea de a fi în starea  $s_i$  la momentul  $t$ , condiționat de parametrii modelului curent și secvența observată.



## Algoritmul Baum-Welch (II)

Definim întâi niște variabile auxiliare:

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

Probabilitatea de a fi în starea  $s_i$  la momentul  $t$  și în starea  $s_j$  la momentul  $t + 1$ , condiționat de parametrii modelului curent și secvența observată.

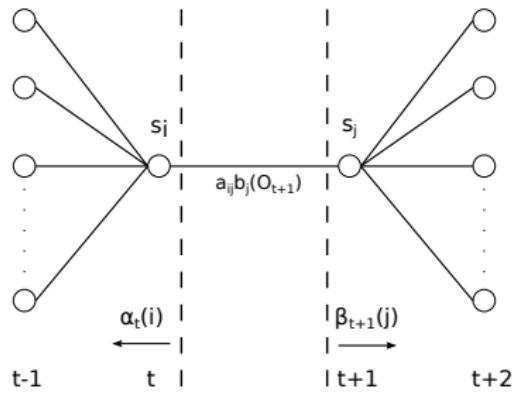
$$\gamma_{t,i} = \gamma_t(i) = P(q_t = s_i | O, \lambda)$$

Probabilitatea de a fi în starea  $s_i$  la momentul  $t$ , condiționat de parametrii modelului curent și secvența observată.

Din definiții rezultă că:

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

# Algoritmul Baum-Welch (III)



$$\alpha_{t,i} = P(o_1, o_2, \dots, o_t, q_t = S_i | \lambda) \quad (36)$$

$$\beta_{t,i} = P(o_{t+1} o_{t+2} \cdots o_T | q_t = S_i, \lambda) \quad (37)$$

Secvența de operații necesară pentru calculul evenimentului mixt ca sistemul se află în starea  $S_i$  la momentul  $t$  și în starea  $S_j$  la momentul  $t + 1$

$$\begin{aligned} \xi_t(i,j) &= \frac{\alpha_{t,i} \cdot a_{i,j} \cdot b_j(o_{t+1}) \cdot \beta_{t+1,j}}{P(O|\lambda)} \\ &= \frac{\alpha_{t,i} \cdot a_{i,j} \cdot b_j(o_{t+1}) \cdot \beta_{t+1,j}}{\sum_{k=1}^N \sum_{l=1}^N \alpha_{t,k} \cdot a_{k,l} \cdot b_l(o_{t+1}) \cdot \beta_{t+1,l}} \end{aligned} \quad (38)$$



## Algoritmul Baum-Welch (IV)

Cum ne ajută aceste variabile auxiliare?

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{numărul estimat de tranziții din } S_i$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{numărul estimat de tranziții din } S_i \text{ la } S_j$$



# Algoritmul Baum-Welch (V)

$$\bar{\pi}_i = \text{nr. estimat de vizite în starea } S_i \text{ la momentul } (t = 1) = \gamma_1(i) \quad (39)$$



# Algoritmul Baum-Welch (V)

$\bar{\pi}_i = \text{nr. estimat de vizite în starea } S_i \text{ la momentul } (t = 1) = \gamma_1(i)$  (39)

$$\begin{aligned} \bar{a}_{i,j} &= \frac{\text{nr. estimat de tranziții din } S_i \text{ la } S_j}{\text{nr. estimat de tranziții din } S_i} \\ &= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \end{aligned} \quad (40)$$



# Algoritmul Baum-Welch (V)

$$\bar{\pi}_i = \text{nr. estimat de vizite în starea } S_i \text{ la momentul } (t=1) = \gamma_1(i) \quad (39)$$

$$\begin{aligned} \bar{a}_{i,j} &= \frac{\text{nr. estimat de tranziții din } S_i \text{ la } S_j}{\text{nr. estimat de tranziții din } S_i} \\ &= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \end{aligned} \quad (40)$$

$$\begin{aligned} \bar{b}_{j,k} &= \frac{\text{nr. estimat de vizite în } S_j \text{ observând simbolul } v_k}{\text{nr. estimat de vizite în } S_j} \\ &= \frac{\sum_{t=1, O_t=v_k}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \end{aligned} \quad (41)$$



# Algoritmul Baum-Welch (VI)

## Algoritm 5 Algoritm Baum-Welch

```

1: intrări:  $O \leftarrow$  secvența de observații,  $\epsilon \leftarrow$  prag de convergență
2:
   {Initializare}
3: init. uniformă  $\Pi$  ( $\Pi_i = 1/N, 1 \leq i \leq N$ )
4: init. aleatoare  $a_{i,j}$ , a. î.  $\sum_{j=1}^N a_{i,j} = 1, \forall i = 1, \dots, N$ 
5: init. uniformă  $b_{j,k}$  ( $b_{j,k} = 1/M, 1 \leq j \leq N, 1 \leq k \leq M$ )
6: init  $\log(P(O|\bar{\lambda})) = 0$ 

7: repeat
8:    $\log(P(O|\lambda)) = \log(P(O|\bar{\lambda}))$ 
9:
   {E STEP - calculeaza variantele scalate pentru  $\alpha$  și  $\beta$  și probabilitatea curentă (log likelihood -  $\log(P(O|\bar{\lambda}))$ ) a secvenței observate}
10:   $[\log(P(O|\bar{\lambda})), \hat{\alpha}, \hat{\beta}, Scale] = forward\_backward(O, \Pi, A, B)$ 
11:
   {M STEP - recalculeaza  $\Pi$ ,  $A$  și  $B$ }
12:   $\Pi = update\_pi\_procedure(\hat{\alpha}, \hat{\beta}, Scale)$ 
13:   $A = update\_A\_procedure(O, \hat{\alpha}, \hat{\beta}, Scale)$ 
14:   $B = update\_B\_procedure(O, \hat{\alpha}, \hat{\beta}, Scale)$ 
15: until  $\log(P(O|\bar{\lambda})) - \log(P(O|\lambda)) < \epsilon$ 

```



# Algoritmul Baum-Welch (VI)

## Algoritmul 6 Algoritm Baum-Welch

---

```

1: Function update_pi_procedure( $\hat{\alpha}$ ,  $\hat{\beta}$ , Scale)
2: for  $i = 1$  to  $N$  do
3:    $\Pi_i = \frac{\hat{\alpha}_1(i) \cdot \hat{\beta}_1(i) / Scale(1)}{\sum_{j=1}^N \hat{\alpha}_1(j) \cdot \hat{\beta}_1(j) / Scale(1)}$ 
4: end for
5: return  $\Pi$ 
6: EndFunction

7: Function update_A_procedure( $O$ ,  $\hat{\alpha}$ ,  $\hat{\beta}$ , Scale)
8: for  $i = 1$  to  $N$  do
9:   for  $j = 1$  to  $N$  do
10:     $a_{i,j} = \frac{\sum_{t=1}^{T-1} \hat{\alpha}_{t,i} \cdot a_{ij} \cdot b_j(o_{t+1}) \cdot \hat{\beta}_{t+1,j}}{\sum_{t=1}^{T-1} \sum_{j=1}^N \hat{\alpha}_{t,i} \cdot a_{i,j} \cdot b_j(o_{t+1}) \cdot \hat{\beta}_{t+1,j}}$ 
11:   end for
12: end for
13: return  $a$ 
14: EndFunction
  
```

---



# Algoritmul Baum-Welch (VI)

---

## Algoritm 7 Algoritm Baum-Welch

---

```
1: Function update_B_procedure( $O$ ,  $\hat{\alpha}$ ,  $\hat{\beta}$ , Scale)
2:   for  $j = 1$  to  $N$  do
3:     for  $k = 1$  to  $M$  do
4:        $b_{j,k} = \frac{\sum_{t=1, O(t)=v_k}^T \hat{\alpha}_t(j) \cdot \hat{\beta}_t(j) / Scale(t)}{\sum_{t=1}^T \hat{\alpha}_t(j) \cdot \hat{\beta}_t(j) / Scale(t)}$ 
5:     end for
6:   end for
7:   return  $b$ 
8: EndFunction
```

---



# A venit vremea să scriem cod

- 1 Faceți o copie a fișierului `baum_welch_disc.m.stub` și denumiți-o `baum_welch_disc.m` (eliminați sufixul `.stub`).

Veți implementa funcția:

```
function [Pi, A, B] = baum_welch_disc(O, T, N, M, ...
model, max_iter)
```



# A venit vremea să scriem cod

- 1 Faceți o copie a fișierului `baum_welch_disc.m.stub` și denumiți-o `baum_welch_disc.m` (eliminați sufixul `.stub`).

Veți implementa funcția:

```
function [Pi, A, B] = baum_welch_disc(O, T, N, M, ...
model, max_iter)
```

- 2 Completați secțiunea `maximization` în care reestimați matricele  $A$ ,  $B$ .

```
1 %% Reestimate A, B
2 % maximization_disc_start – Write code below
3
4 % maximization_disc_end – Write code above
```

- 3 Folosiți `hmm_test` pentru a vă testa codul.



# Cuprins

- 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
- 4 Demo: Recunoașterea Simbolurilor
- 5 Tipuri de MMA
- 6 Discuții și Concluzii



# Aplicație de recunoaștere a simbolurilor

Funcționalități:



# Aplicație de recunoaștere a simbolurilor

Funcționalități:

## Definire

Definire, organizare și vizualizare a unui set de date de simboluri captate prin mișcarea mouse-ului.



# Aplicație de recunoaștere a simbolurilor

Funcționalități:

## Definire

Definire, organizare și vizualizare a unui set de date de simboluri captate prin mișcarea mouse-ului.

## Antrenare

Antrenarea unui motor de recunoaștere a simbolurilor prin metoda MMA.



# Aplicație de recunoaștere a simbolurilor

Funcționalități:

## Definire

Definire, organizare și vizualizare a unui set de date de simboluri captate prin mișcarea mouse-ului.

## Antrenare

Antrenarea unui motor de recunoaștere a simbolurilor prin metoda MMA.

## Recunoaștere

Recunoașterea unui nou simbol și vizualizarea unor metrii de clasificare.



# Aplicație de recunoaștere a simbolurilor

Funcționalități:

## Definire

Definire, organizare și vizualizare a unui set de date de simboluri captate prin mișcarea mouse-ului.

## Antrenare

Antrenarea unui motor de recunoaștere a simbolurilor prin metoda MMA.

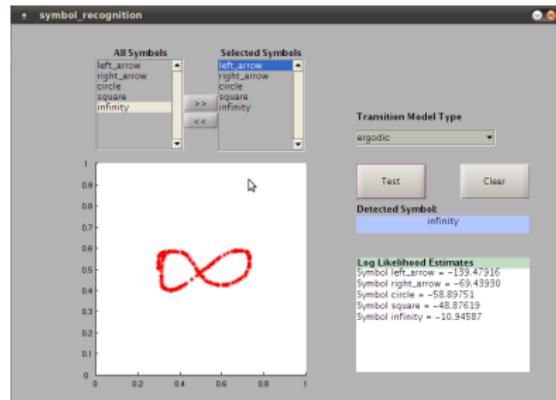
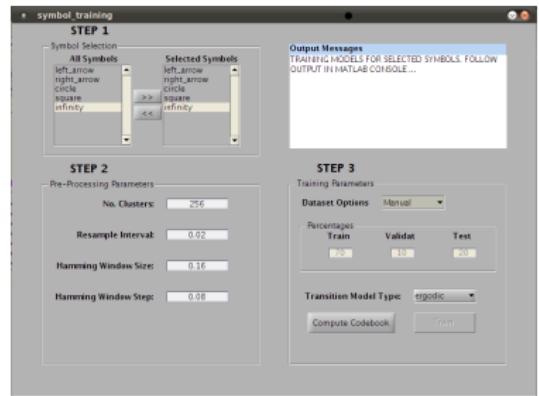
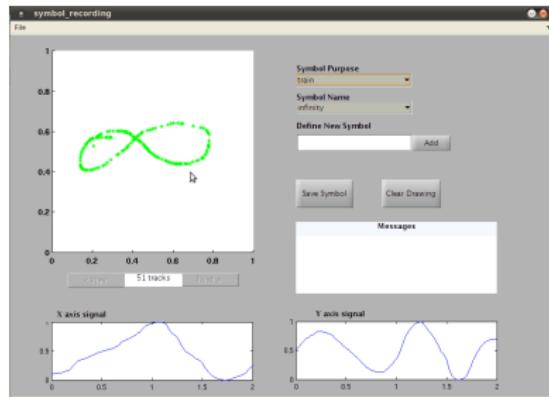
## Recunoaștere

Recunoașterea unui nou simbol și vizualizarea unor metrii de clasificare.

Simboluri default incluse: **săgeată stânga, săgeată dreapta, cerc, pătrat, infinit**

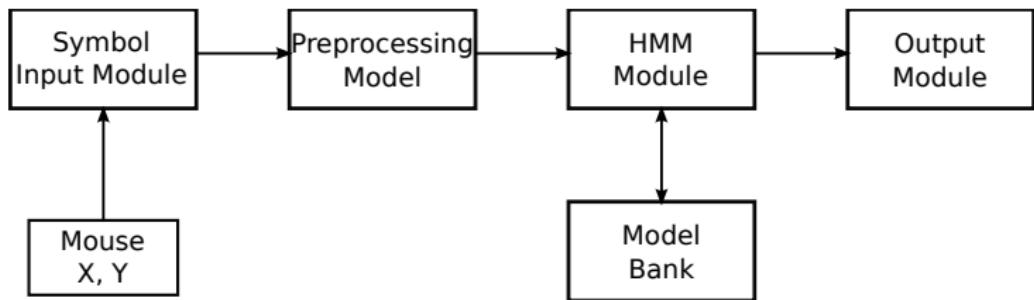


# Aplicație de recunoaștere a simbolurilor - Preview



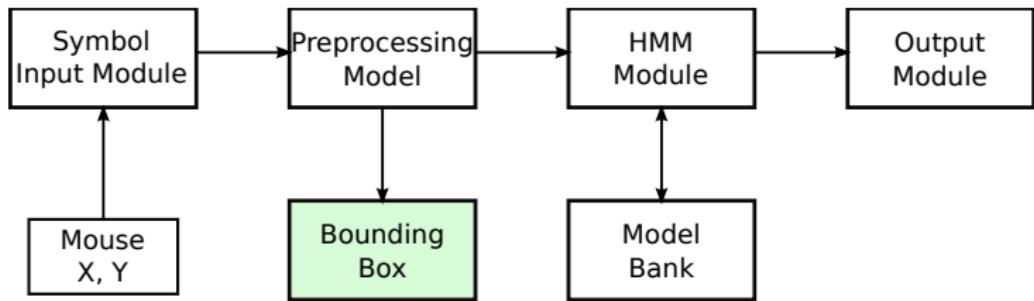
# Aplicație de recunoaștere a simbolurilor - Abordare

Adapted from [YX94].



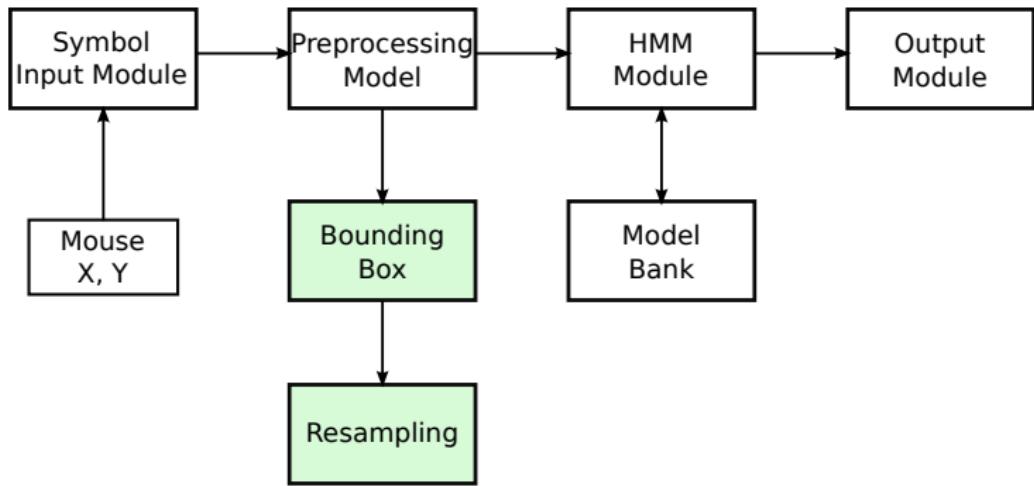
# Aplicație de recunoaștere a simbolurilor - Abordare

Adapted from [YX94].



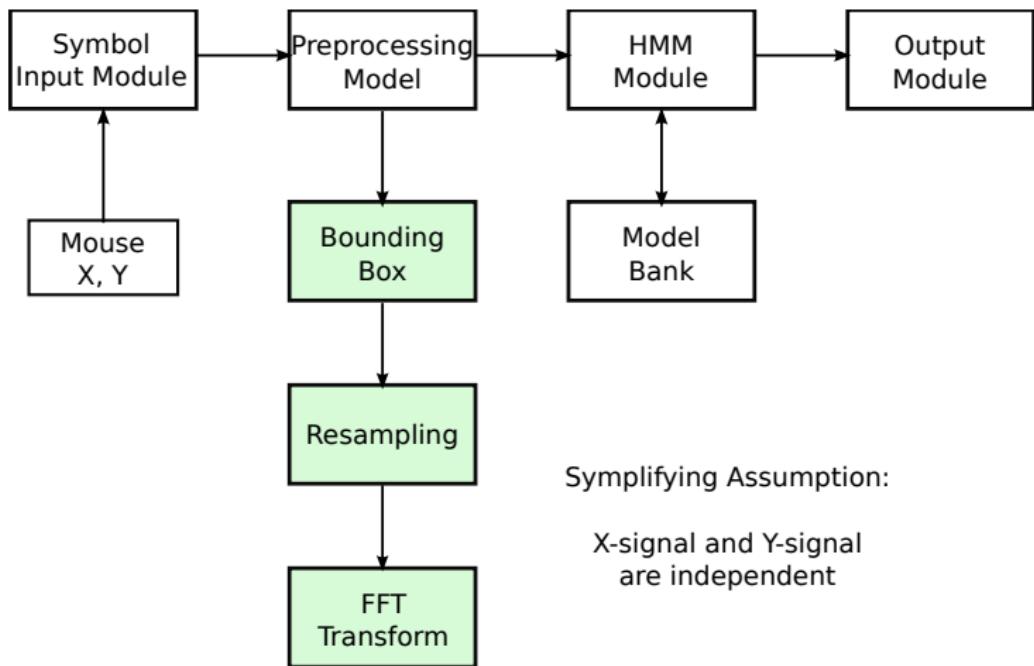
# Aplicație de recunoaștere a simbolurilor - Abordare

Adapted from [YX94].



# Aplicație de recunoaștere a simbolurilor - Abordare

Adapted from [YX94].



# Aplicație de recunoaștere a simbolurilor - Abordare

Adaptare după [YX94].

## Structura MMA

$$N(\text{număr de stări}) = 8$$

2 variabile observabile de tip discrete per stare -  $\text{coef}_{FFT}(x)$ ,  $\text{coef}_{FFT}(y)$

$$M(\text{număr de valori pentru fiecare variabilă observabilă}) = 256$$

Modelul de tranziție:

- Bakis
- Ergodic



# Aplicație de recunoaștere a simbolurilor - Abordare

Procedură de recunoaștere:

- ① Construim și antrenăm câte un MMA pentru fiecare simbol în parte



# Aplicație de recunoaștere a simbolurilor - Abordare

Procedură de recunoaștere:

- ① Construim și antrenăm câte un MMA pentru fiecare simbol în parte
- ② Folosim un set de date (simboluri) de validare pentru a stabili *praguri* de recunoaștere, i.e. dacă probabilitatea secvenței observate este prea mică pentru fiecare MMA, marcăm simbolul drept *necunoscut*

# Aplicație de recunoaștere a simbolurilor - Abordare

Procedură de recunoaștere:

- ① Construim și antrenăm câte un MMA pentru fiecare simbol în parte
- ② Folosim un set de date (simboluri) de validare pentru a stabili *praguri* de recunoaștere, i.e. dacă probabilitatea secvenței observate este prea mică pentru fiecare MMA, marcăm simbolul drept *necunoscut*
- ③ Recunoaștere:
  - Calculăm  $P(O|\lambda_i)$  pentru fiecare MMA construit pentru simbolurile  $i = 1, \dots, nr\_simboluri$
  - Alegem  $\max(P(O|\lambda_i))$  ca și simbol candidat. Dacă  $P(O|\lambda_i) > prag_i$ , atunci am recunoscut *simbolul i*, altfel marcăm *necunoscut*



# Aplicație de recunoaștere a simbolurilor - Rezultate

Mărime set de date

**5 simboluri: săgeată stânga, săgeată dreapta, cerc, pătrat, infinit**

**100 exemple per simbol: 50 antrenare, 10 validare, 40 testare**

```
>> 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
```

# Cuprins

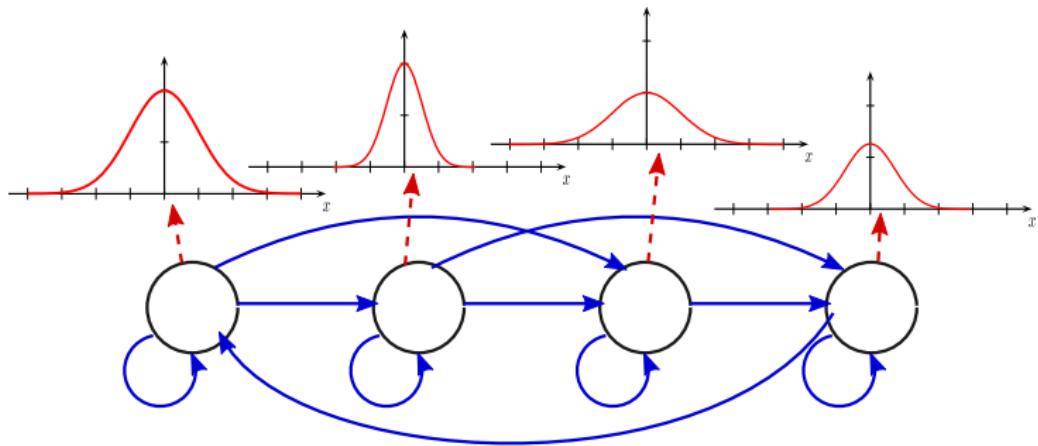
- 1 Aplicații în Învățarea Automată pentru MMA
  - Învățarea Automată
  - MMA în Învățarea Automată
- 2 Teoria MMA
  - Cele Trei Probleme ale MMA
  - Fundamente Matematice
- 3 Implementarea MMA
  - Problema Evaluării: Algoritmul Forward-Backward
  - Problema Interpretării: Algoritmul Viterbi
  - Problema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

# Emisii continue

- Probabilitatea emisiilor este modelată cu o lege de distribuție continuă (e.g. normală).

$$b_j(o) = \mathcal{N}(o, \mu_j, \sigma_j^2) \quad (42)$$

- Parametrii modelului:  $\mu_j, \sigma_j^2 \quad 1 \leq j \leq N$

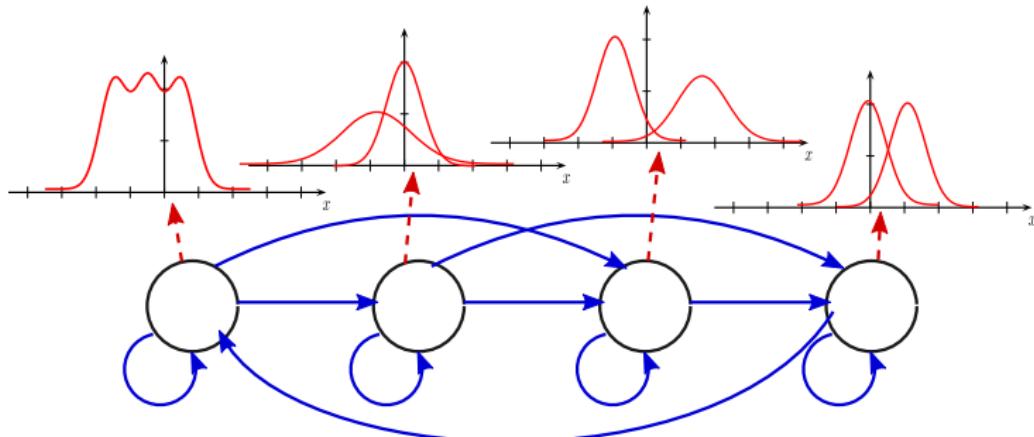


# Mixturi Gaussiene

- Probabilitatea emisiilor este modelată printr-o mixtură de distribuții normale.

$$b_j(o) = \sum_{k=1}^M c_{j,k} \mathcal{N}(o, \mu_{j,k}, \sigma_{j,k}^2) \quad (43)$$

- Parametrii modelului:  $c_{j,k}, \mu_{j,k}, \sigma_{j,k}^2 \quad 1 \leq j \leq N, 1 \leq k \leq M$

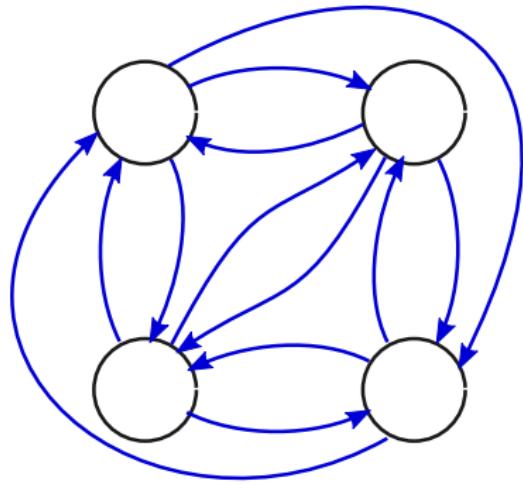


# Modele Markov Complet

## Modele Markov Complet

Un lanț Markov se numește **complet** dacă din orice stare se poate ajunge în oricare altă stare direct.

Matricea de tranziție nu are nicio valoare nulă.

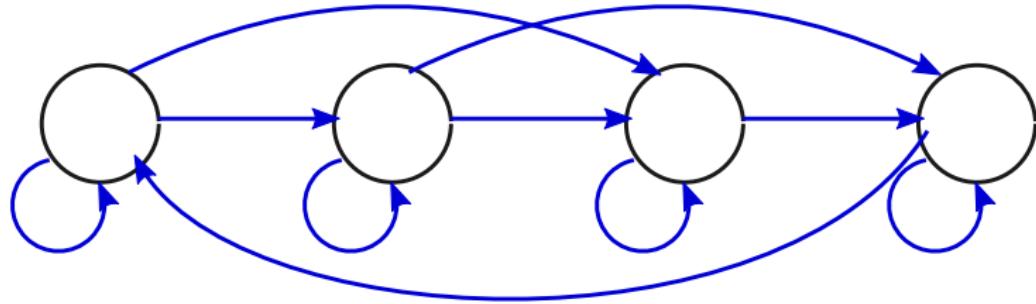


# Modele Markov Ergodice

## Modele Markov Ergodice

Un lanț Markov se numește **ergodic** dacă din orice stare se poate ajunge în oricare altă stare (nu neapărat într-o singură mutare).

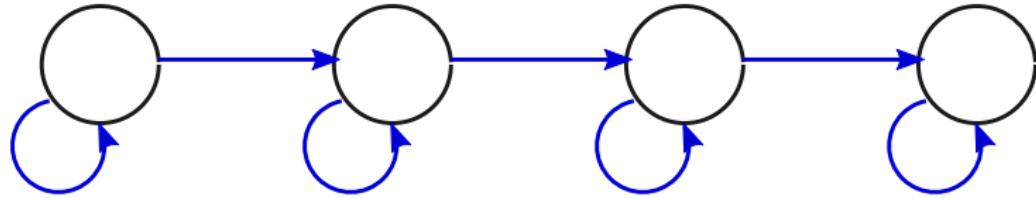
Lanțurile Markov ergodice se mai numesc și *ireductibile*.



# Modelul Bakis

## Modele Markov Bakis (stânga → dreapta)

Un model **Bakis** este unul în care nu este permisă tranziția dintr-o stare către o altă stare cu un indice mai mic.





# Alte tipuri de MMA

- modele Markov cu memorie mixtă [SJ99]
  - scalează mai bine cu secvențe de lungime mare
  - utile în: modelarea vizitării paginilor web
- modele Markov ascunse de durată variabilă [Rab89]
  - utile în: recunoașterea cuvintelor scrise cursiv de mână [CKS95]
- modele Markov ascunse factoriale [GJS97]
  - mai multe variabile de stare
  - putere de modelare mai mare
  - algoritmi exacti intractabili
  - utile în: modelarea suprapunerii unui semnal generat de mai multe voci
- modele Markov ascunse ierarhice [FST98]
  - utile în: recunoașterea scrisului de mână cursiv



# Cuprins

- 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
- 4 Demo: Recunoașterea Simbolurilor
- 5 Tipuri de MMA
- 6 Discuții și Concluzii



# Recapitulare

- Probleme cu secvențe temporale:
  - rețele bayesiene dinamice
  - modele Markov ascunse



# Recapitulare

- Probleme cu secvențe temporale:
  - rețele bayesiene dinamice
  - modele Markov ascunse
- Cele trei probleme fundamentale:
  - problema evaluării
  - problema interpretării
  - problema estimării



# Recapitulare

- Probleme cu secvențe temporale:
  - rețele bayesiene dinamice
  - modele Markov ascunse
- Cele trei probleme fundamentale:
  - problema evaluării - **Algoritmul Forward-Backward**
  - problema interpretării - **Algoritmul Viterbi**
  - problema estimării - **Algoritmul Baum-Welch**

## Referințe I

- [CKS95] M.Y. Chen, A. Kundu, and S.N. Srihari. Variable duration hidden markov model and morphological segmentation for handwritten word recognition. *Image Processing, IEEE Transactions on*, 4(12):1675–1688, 1995.
- [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.
- [FST98] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine learning*, 32(1):41–62, 1998.
- [GJS97] Zoubin Ghahramani, Michael I. Jordan, and Padhraic Smyth. Factorial hidden markov models. In *Machine Learning*. MIT Press, 1997.

## Referințe II

- [KF09] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
- [Mit97] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.
- [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.
- [SJ99] Lawrence K. Saul and Michael I. Jordan. Mixed memory markov models: Decomposing complex stochastic processes as mixtures of simpler ones. *Mach. Learn.*, 37(1):75–87, October 1999.
- [Vit67] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260–269, 1967.

## Referințe III

- [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.

# Întrebări



## Întrebări?

La revedere!



Mulțumim pentru participare!