

Stage

Laboratoire des Technologies Innovantes (LTI)

Kai YANG

Laboratoire des Technologies Innovantes (LTI)



Axe I : Mécanique et Couplage

- **Thème 1** : Matériaux et Efficacité Énergétique (**MEE**)
- **Thème 2** : Modélisation Mécanique et Phénomènes de Transferts (**MMPT**)

Axe II : Énergies et Systèmes

- **Thème 3** : Systèmes Intelligents (**SI**)
- **Thème 4** : Énergie Électrique et Systèmes Associés (**EESA**)

Laboratoire des Technologies Innovantes (LTI)

Intuitive

AliceTHER

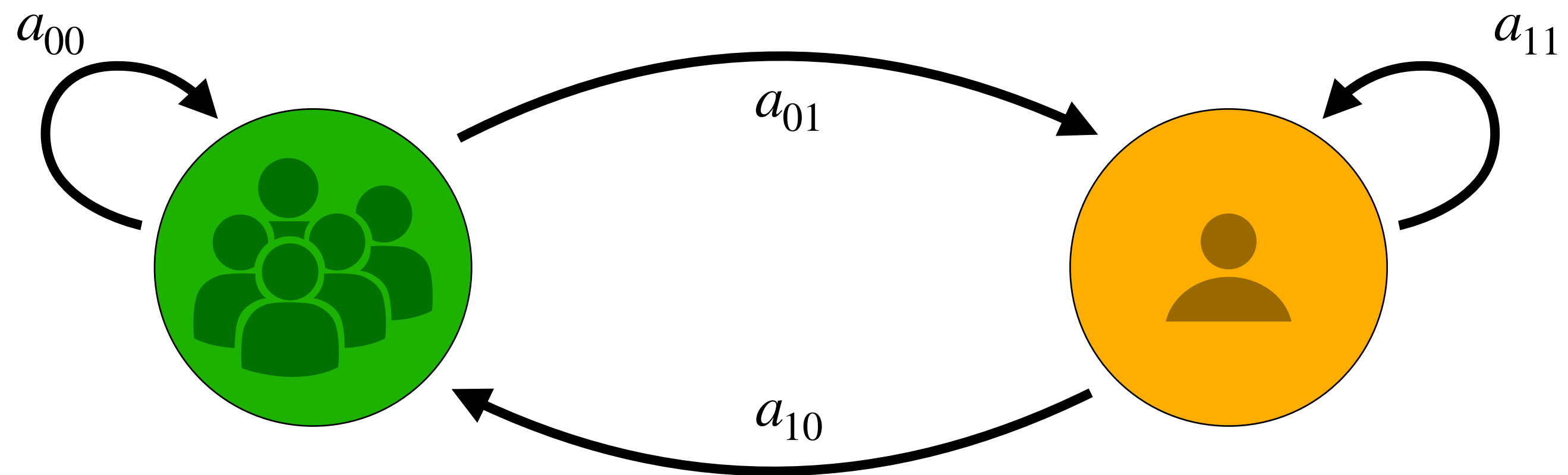
(Analyse, Intelligence et Communication des
Equipements THERmiques du bâtiment)

Axe I : Mécanique et Couplage

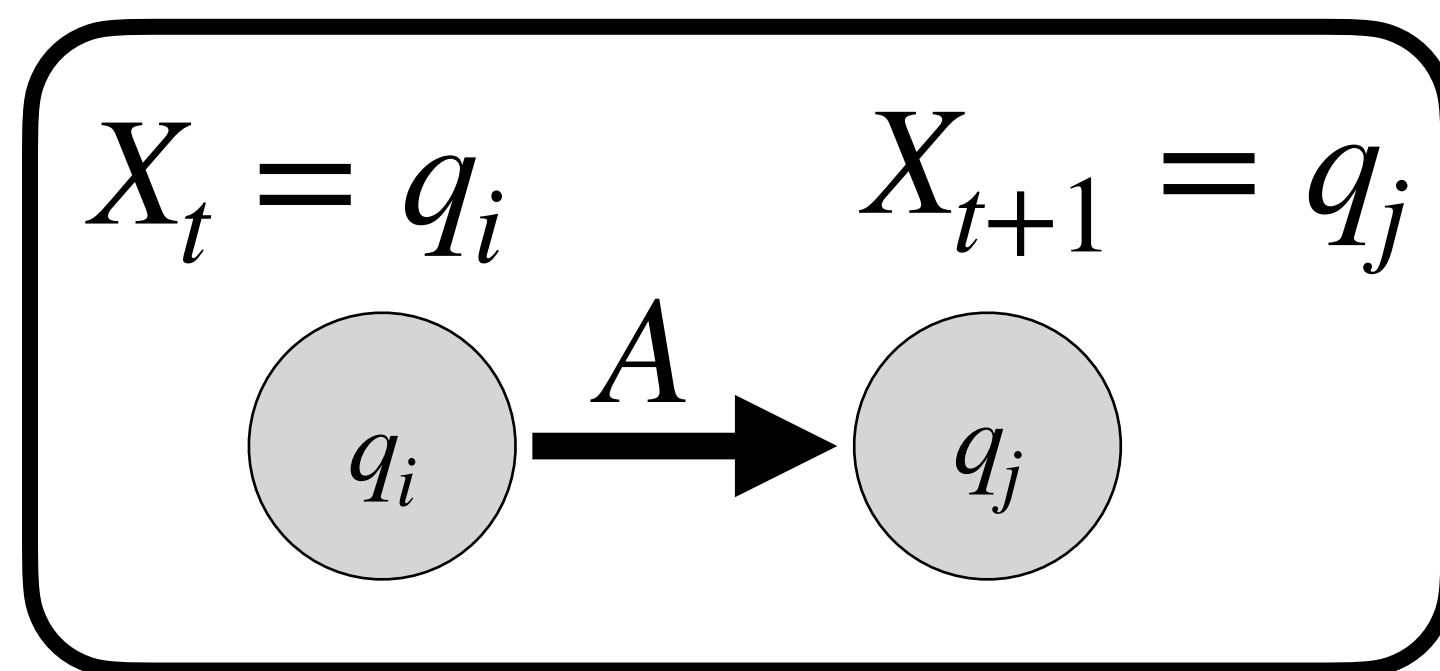
- **Thème 1** : Matériaux et Efficacité Énergétique (**MEE**)
- **Thème 2** : Modélisation Mécanique et Phénomènes de Transferts (**MMPT**)

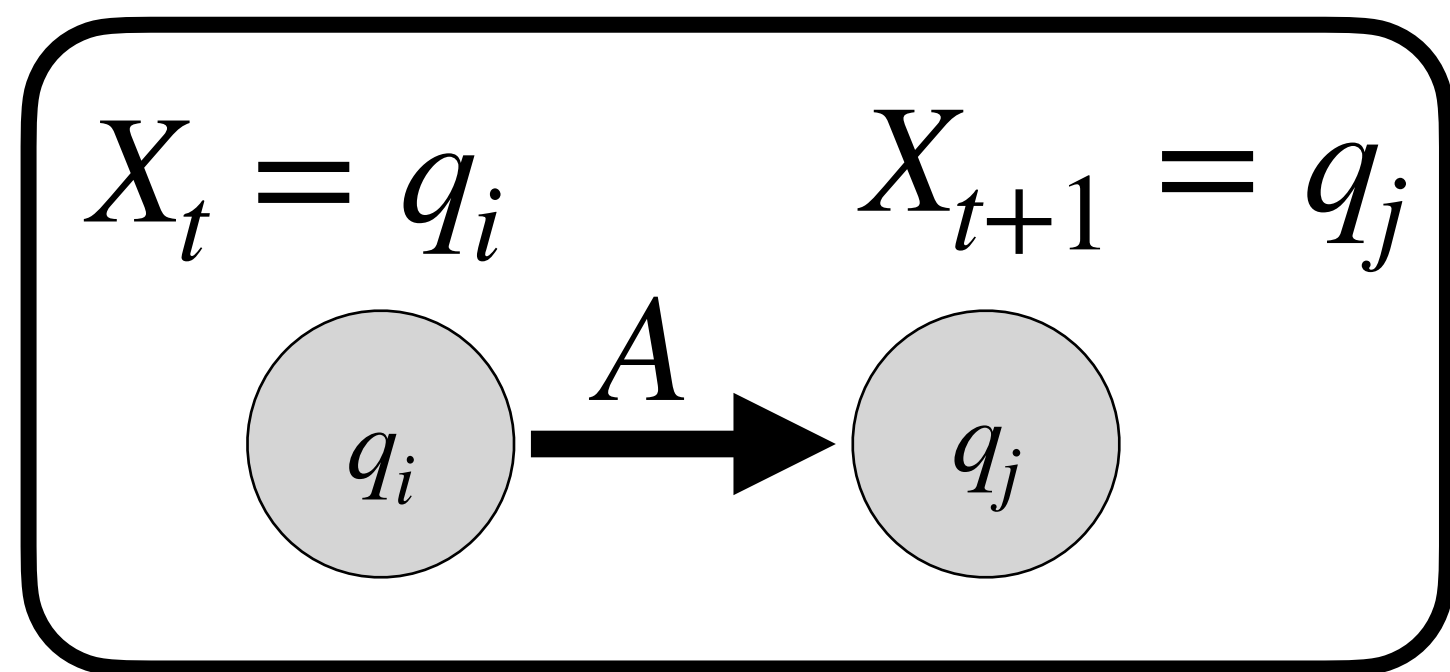
Axe II : Énergies et Systèmes

- **Thème 3** : Systèmes Intelligents (**SI**)
- **Thème 4** : Énergie Électrique et Systèmes Associés (**EESA**)

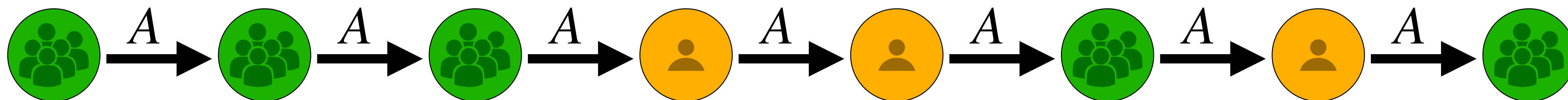


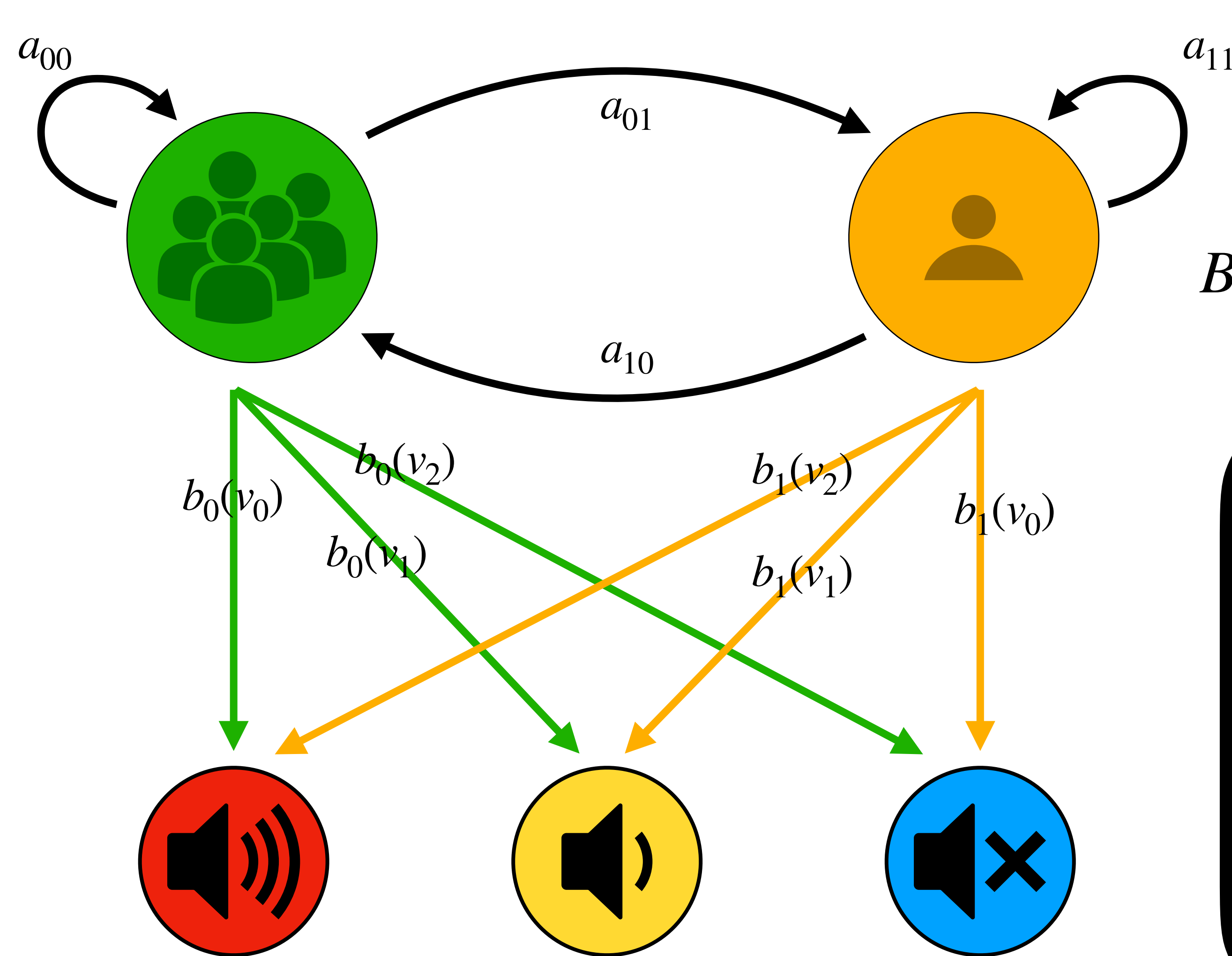
$$A = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix}$$





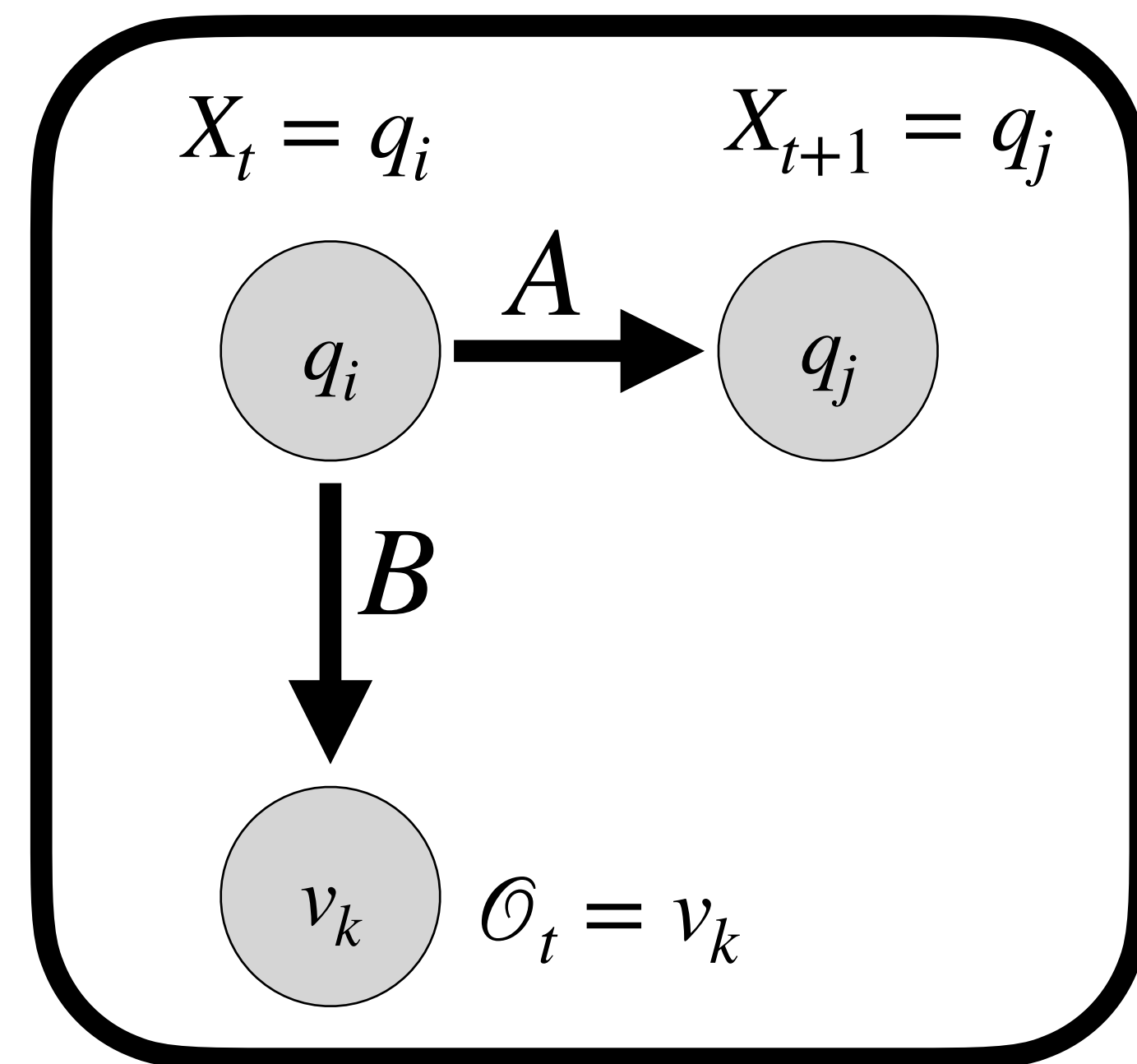
$$A = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix}$$

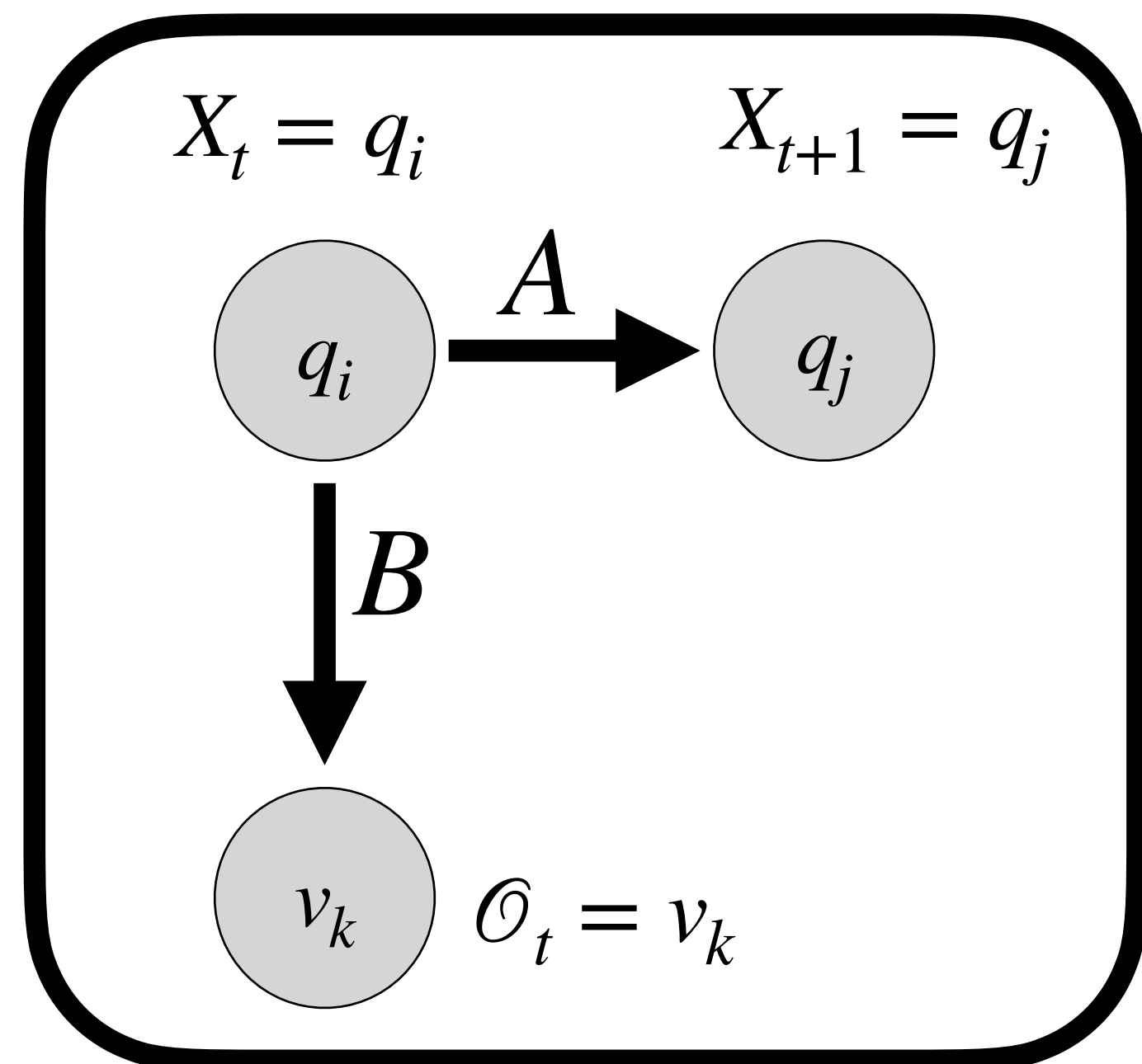




$$A = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix}$$

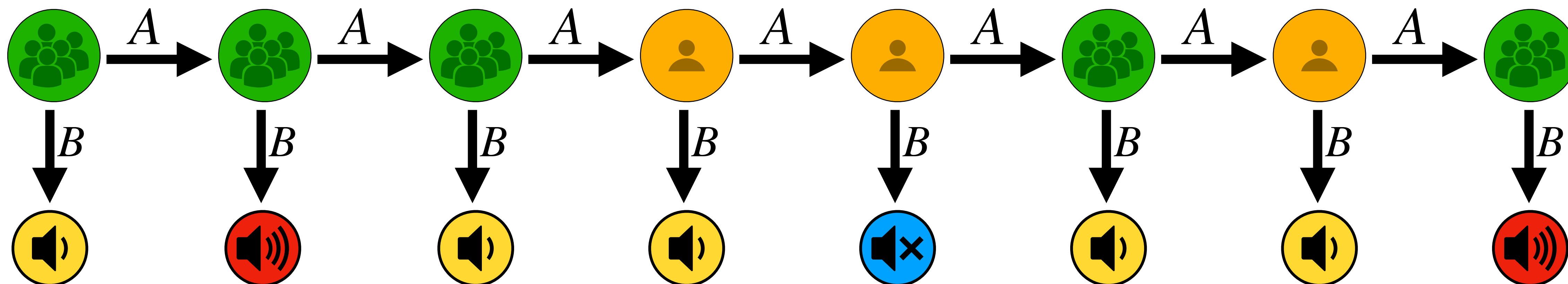
$$B = \begin{bmatrix} b_0(v_0) & b_0(v_1) & b_0(v_2) \\ b_1(v_0) & b_1(v_1) & b_1(v_2) \end{bmatrix}$$

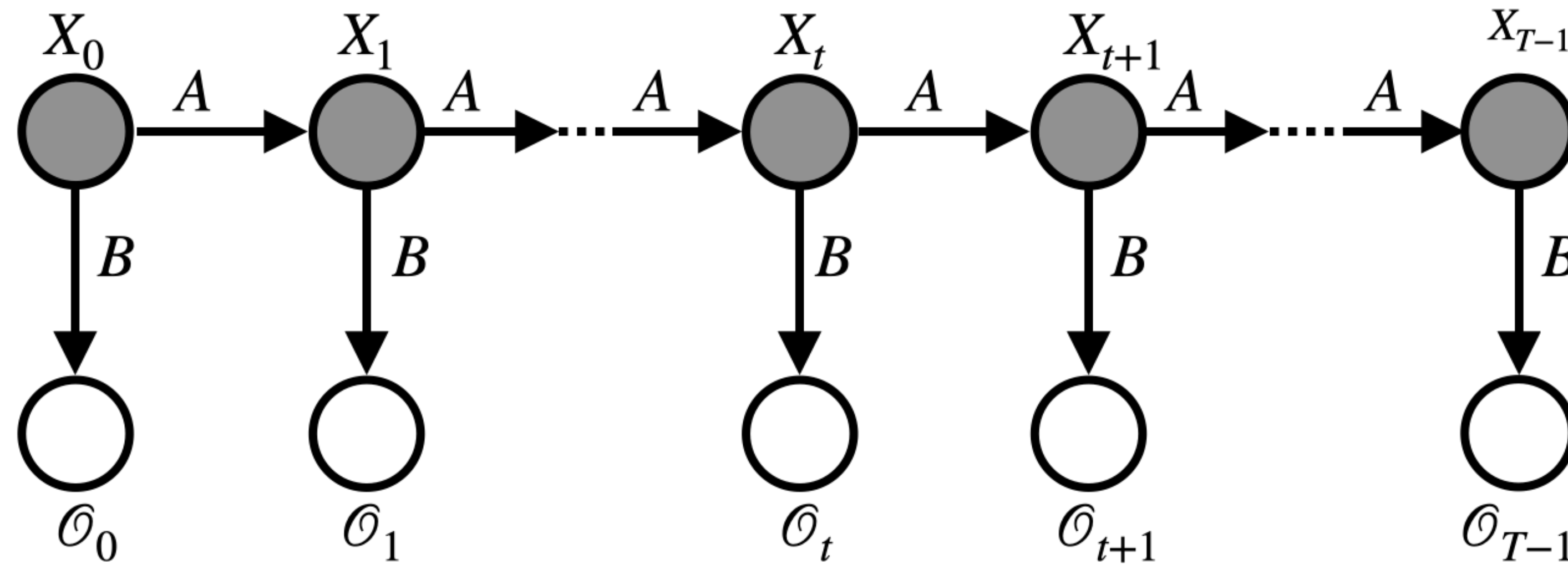




$$A = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix}$$

$$B = \begin{bmatrix} b_0(v_0) & b_0(v_1) & b_0(v_2) \\ b_1(v_0) & b_1(v_1) & b_1(v_2) \end{bmatrix}$$





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

$$\begin{cases} \pi_i = \{\pi_0, \pi_1, \dots, \pi_{N-1}\} \\ a_{ij} = P(X_{t+1} = q_j | X_t = q_i) \\ b_i(k) = P(\mathcal{O}_t = k | X_t = q_i) \end{cases}$$

Évaluation

$$P(\mathcal{O} | \lambda)$$

Forward-Backward

Apprentissage

$$\lambda_{MLE} = \arg \max_{\lambda} P(\mathcal{O} | \lambda)$$

Baum-Welch
EM

Décodeur

$$\hat{X} = \arg \max_X P(\mathcal{O} | \lambda)$$

Viterbi

Sous R

```
HMM.r
1  initHMM = function(States, Symbols, startProbs=NULL, transProbs=NULL,
2    emissionProbs=NULL)
3  > { ...
17 }
18
19  simHMM = function(hmm, length)
20 > { ...
37 }
38
39  viterbi = function(hmm, observation)
40 > { ...
89 }
90
91  forward = function(hmm, observation)
92 > { ...
122 }
123
124  backward = function(hmm, observation)
125 > { ...
155 }
156
157  baumWelch = function(hmm, observation, maxIterations=100, delta=1E-
158 > { ...
194 }
195
196  baumWelchRecursion = function(hmm, observation)
197 > { ...
253 }
```

Procedure-oriented

RArray

An array of one or two dimension, but with the name of each line or column

RDataFrame

<u>Data 1</u>	<u>Data 2</u>	<u>Data 3</u>
...
...
...

CsvHelper

Read *.csv
Write *.csv

Code avec un langage compilé (C#)

HMM

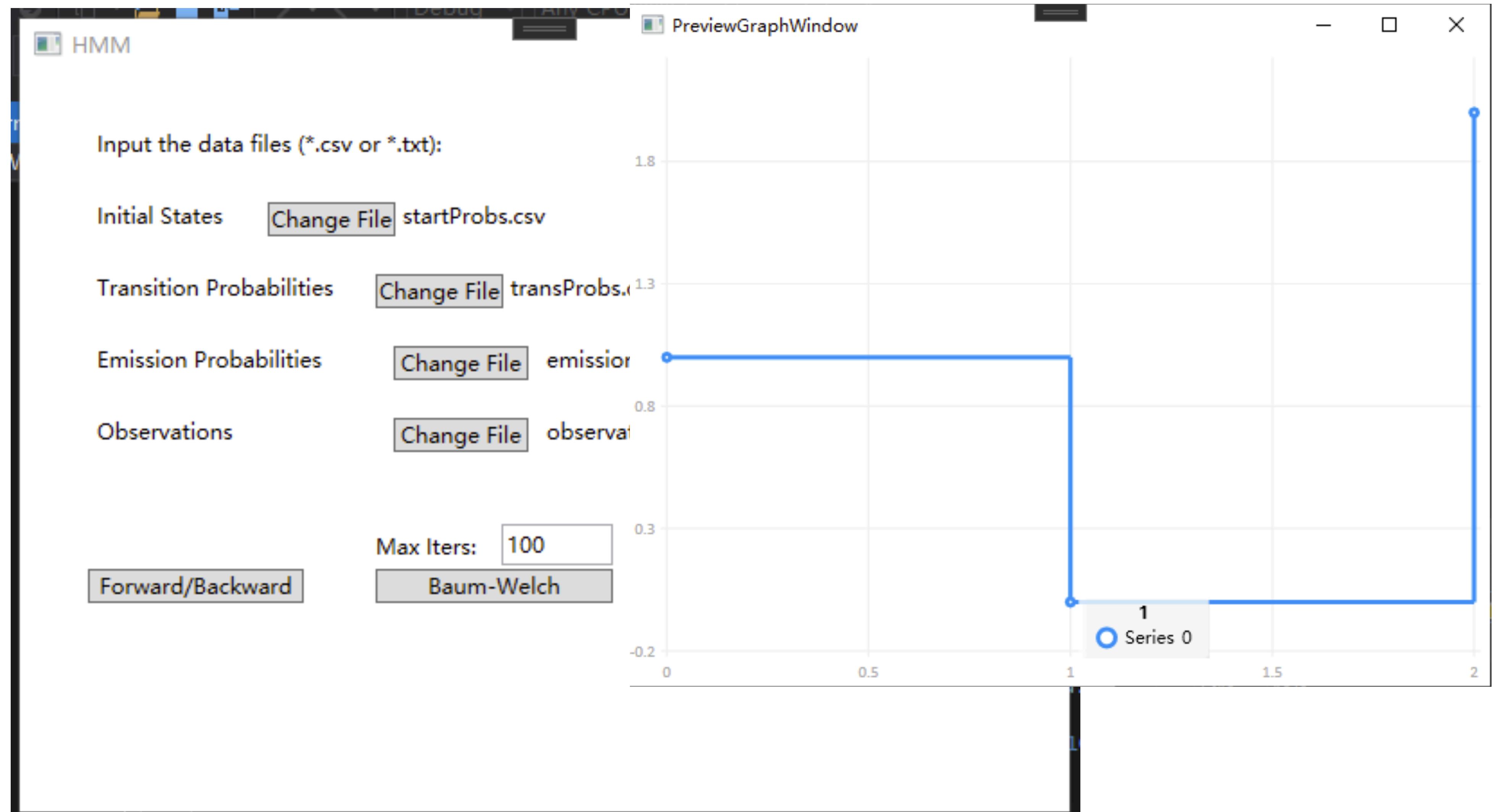
- State names
- Symbol names
- Start probabilities
- Transition probabilities
- Emission probabilities

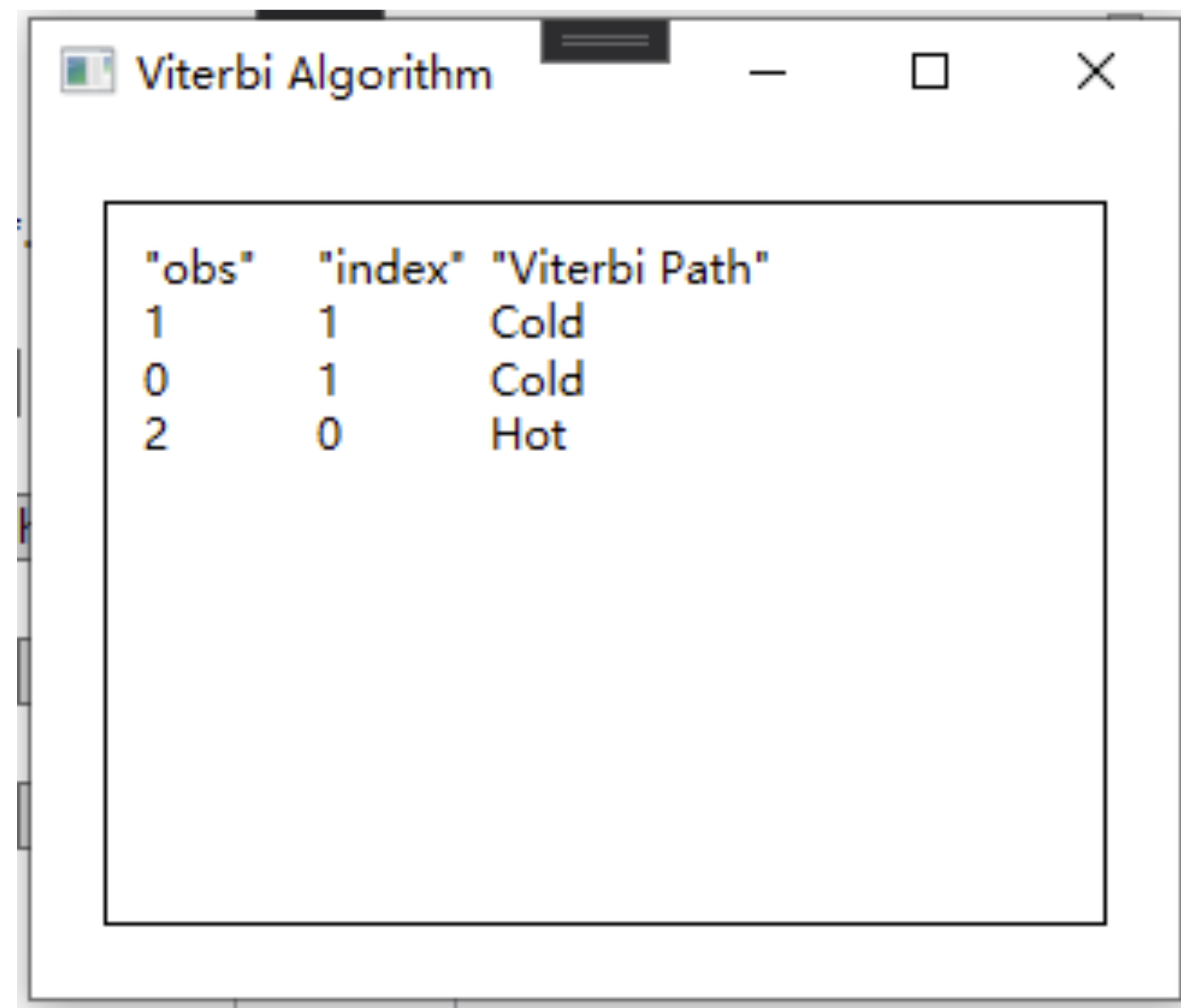
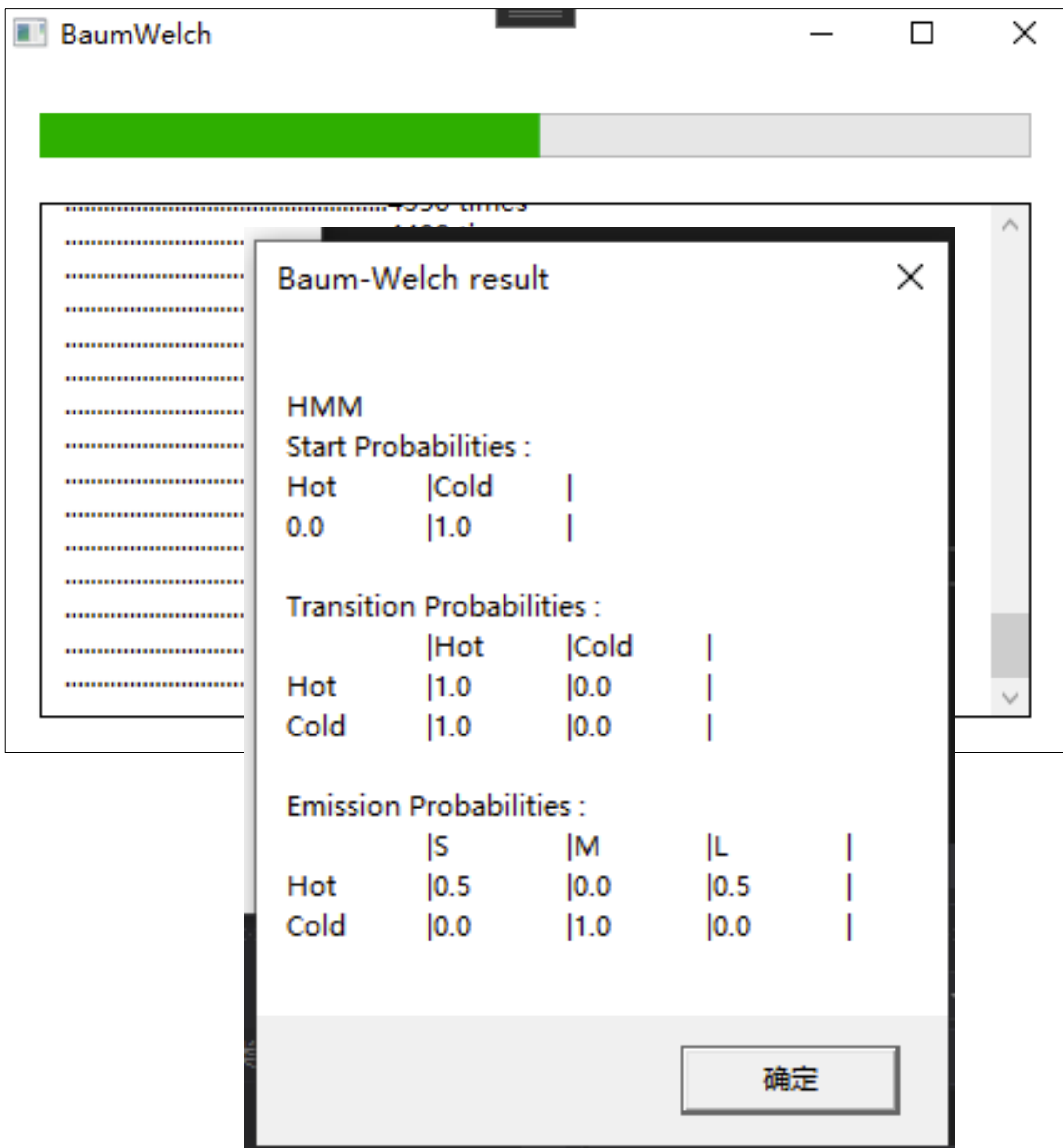
- + Constructors (for initialization)
- + Forward algorithm
- + Backword algorithm
- + Baum-Welch learning
- + Viterbi path

Program.main

Test the fonctions in
Console

Project WPF





La classe HMM

Bref

Attribut (privé)

nom	type	description
numStates	int	nombre des états cachés
numSymbols	int	nombre des états d'observation
states	string[]	nom des états cachés
symbols	string[]	nom des états d'observation
startProbs	double[]	la matrice de probabilités initiales
transProbs	double[]	la matrice de probabilités de transition des états cachés
emissionProbs	double[]	la matrice de probabilités d'émission

Propriété (publique)

nom	type	description
NumStates { get }	int	nombre des états cachés
NumSymbols { get }	int	nombre des états d'observation
States { get }	string[]	nom des états cachés
Symbols { get }	string[]	nom des états d'observation
StartProbs { get }	double[]	la matrice de probabilités initiales
TransProbs { get }	double[]	la matrice de probabilités de transition des états cachés
EmissionProbs { get }	double[]	la matrice de probabilités d'émission

Méthode publique

nom	entrée	type de retourne	description
getForwardTable	int[] observations	RArray	le résultat du tableau de alpha-pass obtenue par l'algorithme forward
getBackwordTable	int[] observations	RArray	le résultat du tableau de beta-pass obtenue par l'algorithme backward
baumWelch	int[] observations	HMM	retourner le nouveau HMM obtenue par l'algorithme Baum-Welch
baumWelch	int[] observations, int maxIterations	HMM	retourner le nouveau HMM obtenue par l'algorithme Baum-Welch
baumWelch	int[] observations, delegate doSth	HMM	retourner le nouveau HMM obtenue par l'algorithme Baum-Welch, et exécuter doSth
baumWelch	int[] observations, int maxIterations, delegate doSth	HMM	retourner le nouveau HMM obtenue par l'algorithme Baum-Welch, et exécuter doSth
baumWelch	int[] observations, int maxIterations, double delta, delegate doSth	HMM	retourner le nouveau HMM obtenue par l'algorithme Baum-Welch, et exécuter doSth
getViterbiPath	int[] observations	RArray	retourner le chemin des états cachés obtenue l'algorithme Viterbi
ToString	void	String	récrire la méthode ToString() de la classe père

Méthode privée

nom	entrée	type de retourne	description
forward	int[] observations	double[,]	le résultat logarithmique du tableau de alpha-pass obtenue par l'algorithme forward
backward	int[] observations	double[,]	le résultat logarithmique du tableau de beta-pass obtenue par l'algorithme backward
baumWelchRecursion	HMM hmm, int[] observations	HMM	une fois de mettre à jour pour l'algorithme Baum-Wemch
normalizeMatrx	double[,] matrix	double[,]	normaliser la matrice d'entrée et la retourner
updateModel	double[,] transitionMatrix, double[,] emissionMatrix	bool	mettre à jour la matrice de transition et la matrice d'émission

Détail

`id="constructor-4">HMM(string[],string[],double[],double[,],double[,])</code>`

- #### Prototype complet

```
```csharp
public HMM(string[] states, string[] symbols, double[] startProbs, double[,] transProbs,
```
```

- #### Description détaillée

Inctancier une `HMM`, et assigner la matrice initial, la matrice de transition et la mat

- #### Paramètre d'entrée

- `states` : nom d'états cachés

- `symbols` : nom d'états d'observation

- `startProbs` : matrice de probabilités des états initials

- `transProbs` : matrice de probabilités des états cachés

- `emissionProbs` : matrice de probabilités des états d'observation

- #### 📖 Exemple d'usage

```
```csharp
string[] states = new string[] { "q0", "q1" };
string[] symbols = new string[] { "v0", "v1", "v2" };
double[] startProbs = new double[] { 0.3, 0.7 };
double[,] transProbs = new double[,] {
 new double[] { 0.4, 0.6 },
 new double[] { 0.75, 0.25 }
};
double[,] emissionProbs = new double[,] {
 new double[] { 0.3, 0.3, 0.4 },
 new double[] { 0.35, 0.25, 0.4 }
};
HMM hmm = new HMM(states, symbols, startProbs, transProbs, emissionProbs);
```
```

Retourner au [\[**Constructor**\]\(#constructor\)](#)

HMM(string[],string[],double[],double[,],double[,])

• Prototype complet

```
public HMM(string[] states, string[] symbols, double[] startProbs, double[,] transProbs, double[,] emissionProbs)
```

• Description détaillée

Inctancier une `HMM`, et assigner la matrice initial, la matrice de transition et la matrice d'émission, par les paramètres d'entrée.

• Paramètre d'entrée

- `states` : nom d'états cachés
- `symbols` : nom d'états d'observation
- `startProbs` : matrice de probabilités des états initials
- `transProbs` : matrice de probabilités des états cachés
- `emissionProbs` : matrice de probabilités des états d'observation

• 📖 Exemple d'usage

```
string[] states = new string[] { "q0", "q1" };
string[] symbols = new string[] { "v0", "v1", "v2" };
double[] startProbs = new double[] { 0.3, 0.7 };
double[,] transProbs = new double[,] {
    new double[] { 0.4, 0.6 },
    new double[] { 0.75, 0.25 }
};
double[,] emissionProbs = new double[,] {
    new double[] { 0.3, 0.3, 0.4 },
    new double[] { 0.35, 0.25, 0.4 }
};
HMM hmm = new HMM(states, symbols, startProbs, transProbs, emissionProbs);
```

Retourner au [Constructor](#)

Conclusion

Une expérience

Notion
HMM

Apprentissage
automatique

Travail en
France

Gestion de la
structure du code
et projet

Traitement des
données