UNIVERSITÉ CÔTE D'AZUR

Modélisation de l'incertitude (M2 MIAGE IA²)

Andrea G. B. Tettamanzi
Laboratoire I3S – Équipe SPARKS
andrea.tettamanzi@univ-cotedazur.fr

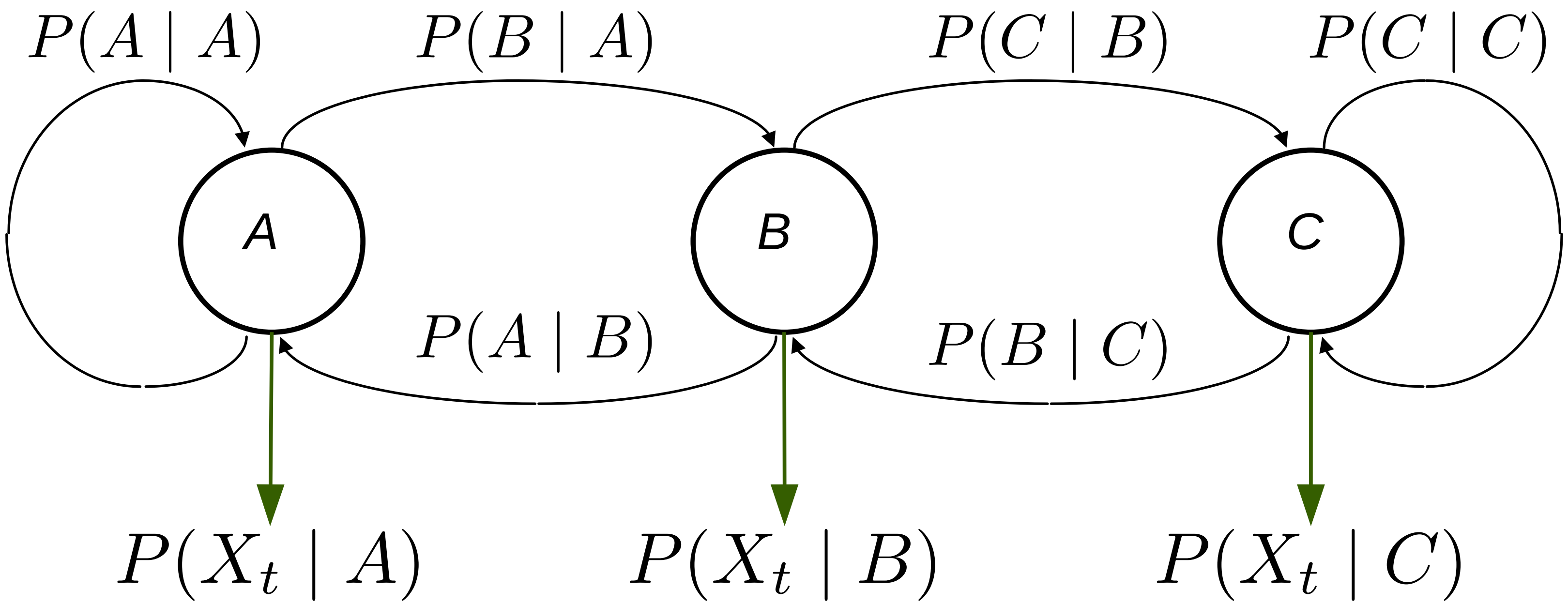
univ-cotedazur.fr

Séance 4
Modèles de Markov cachés

Dans cette séance

- Modèles de Markov cachés
- Applications des MMC
- Estimation, entraînement et décodage
- Algorithme de Viterbi
- Algorithme espérance-maximisation
- Algorithme de Baum-Welch

Modèles de Markov cachés



Processus doublement stochastique :
probabilités de transition et d'émission

Peut se mettre sous forme matricielle

Applications des MMC

- Traitement du langage
  - Reconnaissance vocale
  - Reconnaissance de l’écriture manuscrite
  - Traduction automatique
- Bio-informatique
  - Analyse de séquences d’ADN (prédiction des gènes)
- Finance
  - Prédiction d’inversion de tendance dans les séries temporelles

Problèmes dans les MMC

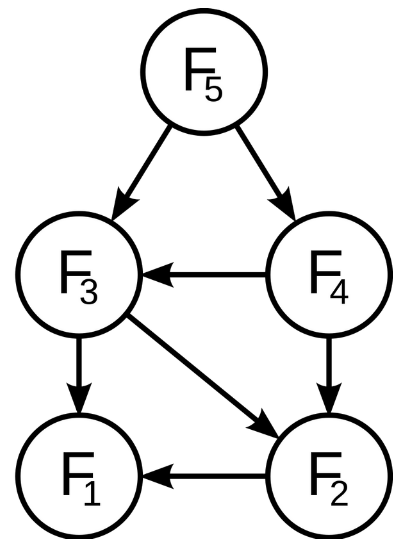
- Estimation :
  - trouver la probabilité d'une séquence d'observations étant donné le modèle
  - Algorithme de Baum-Welch (tous les chemins)
  - Algorithme de Viterbi (meilleur chemin)
- Entraînement :
  - Trouver les probabilités de transition & émission étant donné une topologie et des données
  - Algorithme espérance-maximisation
- Décodage :
  - Trouver la séquence d'états la plus probable étant donné une séquence d'observations
  - Algorithme de Viterbi

Programmation dynamique

- Méthode algorithmique pour résoudre des problèmes d'optimisation
- Introduite par Richard Bellman (années 1950)
- Idée : résoudre un problème en le décomposant en sous-problèmes, puis résoudre les sous-problèmes, des plus petits aux plus grands, en stockant les résultats intermédiaires
- Une solution optimale d'un problème s'obtient en combinant des solutions optimales à des sous-problèmes (principe d'optimalité de Bellman)

Exemple : nombres de Fibonacci

```
def fibonacci(n):
   if n==0 or n==1:
     return n
   else:
     return fibonacci(n – 1) + fibonacci(n – 2)
```




```
def fibonacci(n):
   F[0] = 0
   F[1] = 1
   for i in range(2, n):
      F[i] = F[i – 1] + F[i – 2]
   return F[n]
```

Algorithme de Viterbi

$$\max_{x_1,\ldots,x_t} P(x_1,\ldots,x_t,X_{t+1} \mid e_{1,\ldots,t}) =$$
$$\alpha P(e_{1,\ldots,t} \mid X_{t+1})$$
$$\max_{x_t}\left(P(X_{t+1} \mid x_t)\max_{x_1,\ldots,x_{t-1}} P(x_1,\ldots,x_{t-1},x_t \mid e_{1,\ldots,t})\right)$$

Le chemin (= séquence d’observation) optimal est composé de sous-chemins optimaux

Algorithme espérance-maximisation

- On initialise le modèle aléatoirement, par exemple *c* centroïdes
- On affine ce modèle itérativement an alternant deux étapes
  - Espérance : affecter chaque échantillon X_i à C_j avec

$$P(X_i \in C_j) = p(C_j \mid X_i) = \frac{p(C_j)p(X_i \mid C_j)}{p(X_i)}$$

$$p(X_i \mid C_j) = \phi(X_i; \mu_j, \sigma_j)$$

  - Maximisation : estimer les paramètres du modèle

$$\mu_k = \frac{\sum_{i=1}^{N} X_i P(X_i \in C_k)}{\sum_{j=1}^{N} P(X_i \in C_j)} \quad \sigma_k = \frac{\sum_{i=1}^{N} (X_i - \mu_k)^2 P(X_i \in C_k)}{\sum_{j=1}^{N} P(X_i \in C_j)}$$

Algorithme de Baum-Welch

- Cas particulier d'une généralisation de l'algorithme espérance-maximisation
- Déterminer le modèle qui explique le mieux une séquence d'observations donnée
- Algorithme itératif, qui permet d'estimer les paramètres du modèle qui maximisent la probabilité d'une séquence d'observables.
- Converge vers un maximum local.