

# Apprentissage automatique 2 (KINX9AB1)

Cours 2 : modèles de séquences, les HMM  
M2 IAFA, IMA

Contributeurs :  
Thomas Pellegrini  
Philippe Muller  
Contacts : prenom.nom@irit.fr



# Modèles de séquence : les modèles de Markov Cachés (*Hidden Markov Models, HMM*)

Crédit : tous les slides avec des dés et sur les modèles de langage sont  
tirés des slides de Noah Smith :  
[http://lxmls.it.pt/2021/wp-content/uploads/2021/07/Noah\\_Smith.pdf](http://lxmls.it.pt/2021/wp-content/uploads/2021/07/Noah_Smith.pdf)

# Motivation : exemples de cas d'usage

- Traitement du Langage Naturel (*Natural Language Processing*) : traitement de données textuelles, quelques exemples :
  - "Modèles de langage" : prédire le mot suivant en fonction des précédents
  - Prédiction de tags syntaxiques (nom, verbe, préposition, etc) : *Part-of-Speech tagging*, *POS tagging* (Church, 1988 ; Brants, 2000)
  - Prédiction d'entités nommées : détecter dans un texte les mots qui sont des noms propres, des lieux, etc (Bikel et al., 1999)
  - Alignement de mots dans des textes "parallèles" (Vogel et al., 1996)

- Traitement Automatique de la Parole (*Speech processing*), quelques exemples :
  - Reconnaissance automatique de la parole : *Automatic Speech Recognition, ASR* (Rabiner, 1989 ; Gales & Young, 2008)
  - Segmentation et Reconnaissance du Locuteur : *Speaker Diarization*, qui parle quand ? *Variational Bayesian HMM* (Mireia et al., **2018!!**)
  - Synthèse automatique de la parole : *Automatic Speech Synthesis*

# Motivation : modèles de langage

- La langage écrit peut être vu comme une séquence de lettres, de morphèmes, de mots, de paires de mots, de triplets de mots, etc
- On parle de  $n$ -grammes de mots avec  $n = 1, 2, 3 \dots$
- Beaucoup de séquences différentes sont possibles...
- Comment déterminer des distributions de probabilité sur les séquences possibles ?

→ Nous avons besoin d'une famille de modèles probabilistes et d'algorithmes pour construire un modèle à partir des données

# Modèle fondé sur le contexte gauche

- Chaque symbole (mots)  $O_i$  est généré de gauche à droite, conditionné par tous les mots qui précèdent

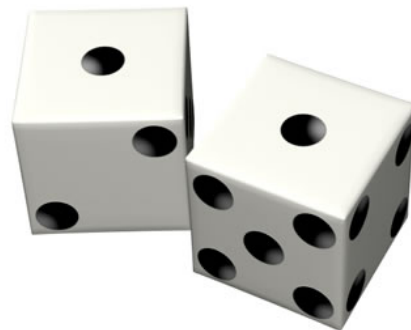
$$\begin{aligned} P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) &= \prod_{i=1}^n P(O_i | \text{start}, O_1, \dots, O_{i-1}) \\ &= \prod_{i=1}^n \gamma(O_i | \text{start}, O_1, \dots, O_{i-1}) \end{aligned}$$

- Notre modèle  $\gamma$  doit fournir la probabilité du terme à droite

# Die / Dice



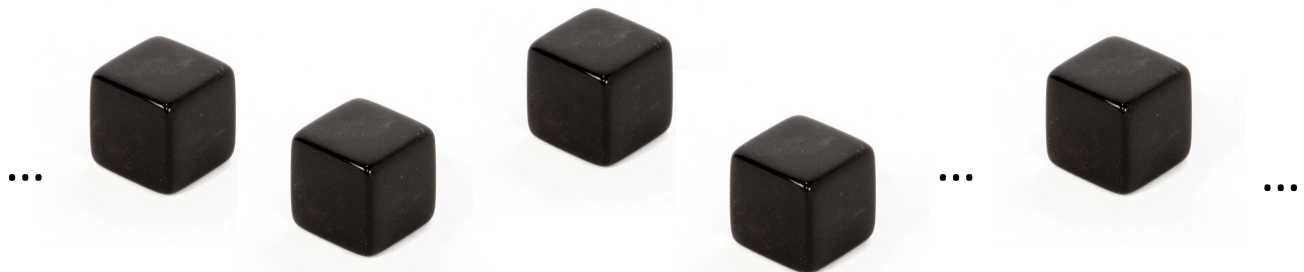
one die



two dice

start

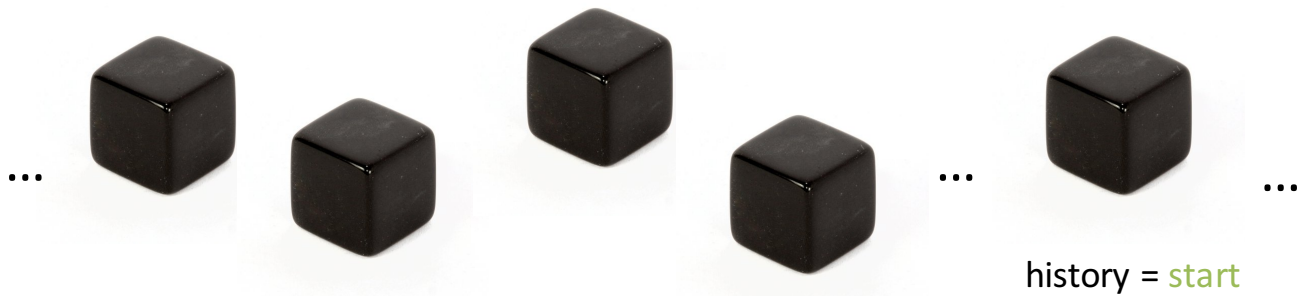
one die per history:





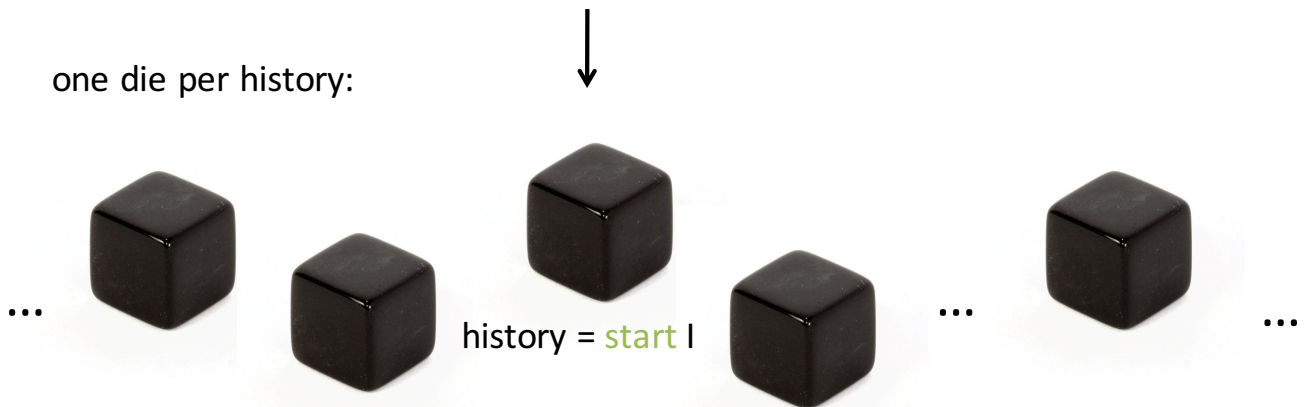


one die per history:



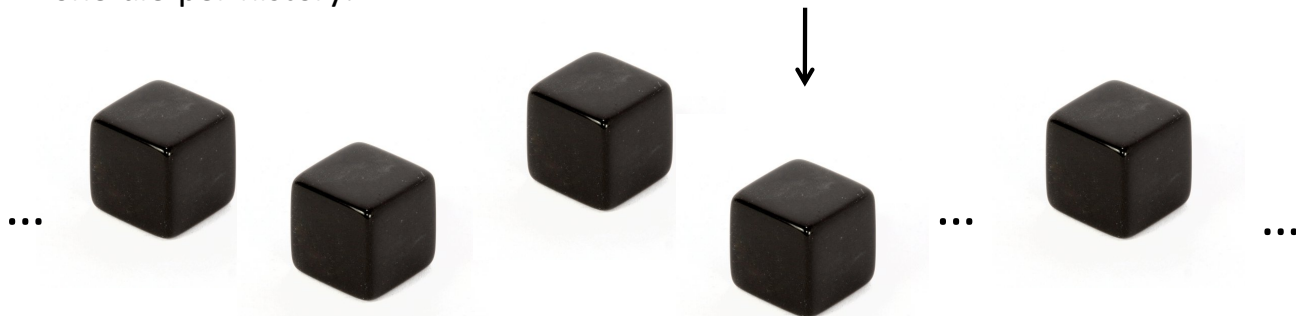
start | want

one die per history:



start I want a

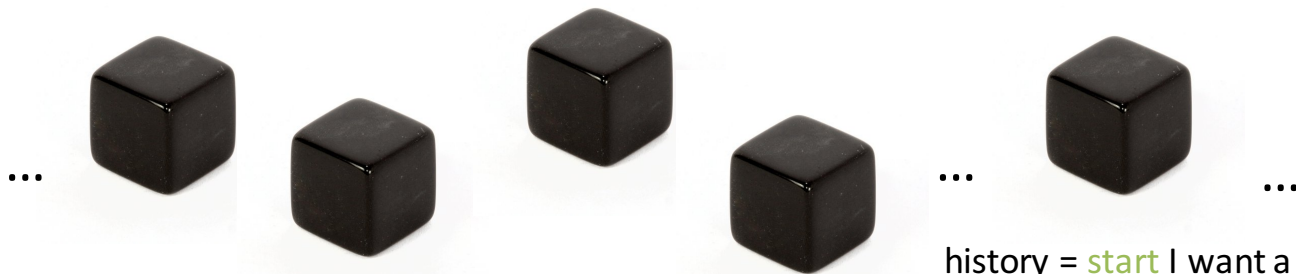
one die per history:



history = start I want

start I want a flight

one die per history:



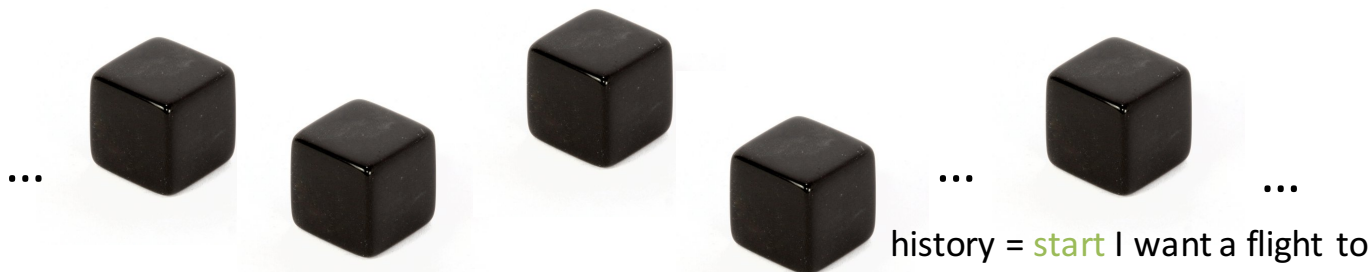
start I want a flight to

one die per history:



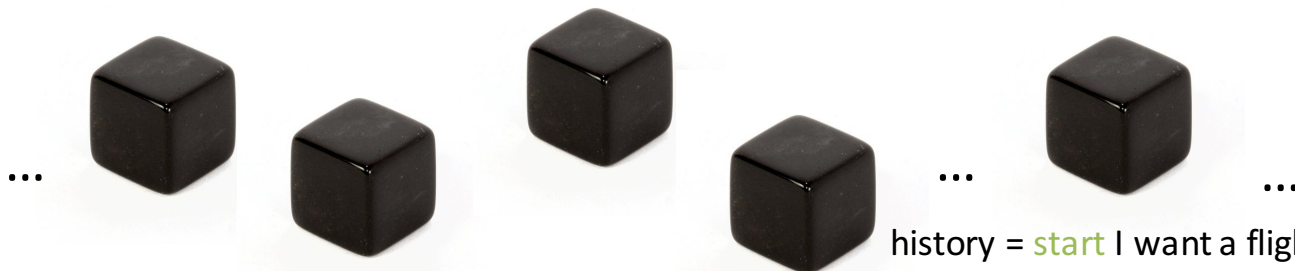
start I want a flight to Lisbon

one die per history:



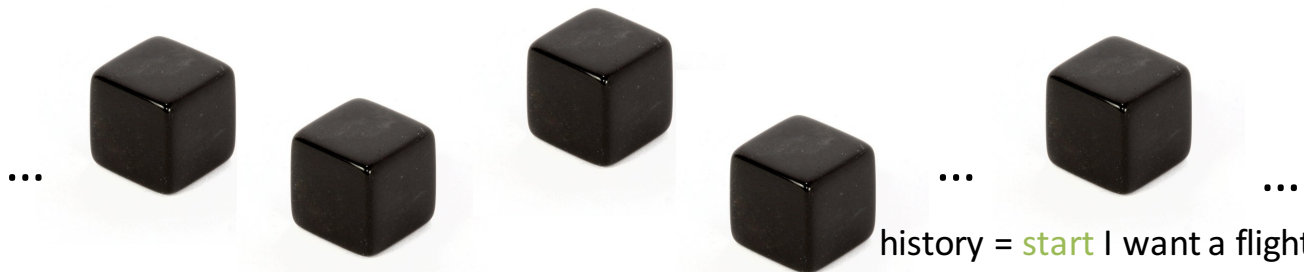
start I want a flight to Lisbon .

one die per history:



start I want a flight to Lisbon . stop

one die per history:





# Modèle fondé sur le contexte gauche

- Chaque symbole (mots)  $O_i$  est généré de gauche à droite, conditionné par tous les mots qui précèdent

$$\begin{aligned} P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) &= \prod_{i=1}^n P(O_i | \text{start}, O_1, \dots, O_{i-1}) \\ &= \prod_{i=1}^n \gamma(O_i | \text{start}, O_1, \dots, O_{i-1}) \end{aligned}$$

- Ce modèle aurait une très grande expressivité, mais...

# Modèle fondé sur le contexte gauche

- Chaque symbole (mots)  $O_i$  est généré de gauche à droite, conditionné par tous les mots qui précèdent

$$\begin{aligned} P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) &= \prod_{i=1}^n P(O_i | \text{start}, O_1, \dots, O_{i-1}) \\ &= \prod_{i=1}^n \gamma(O_i | \text{start}, O_1, \dots, O_{i-1}) \end{aligned}$$

- Ce modèle aurait une très grande expressivité, mais...
- Combien a-t-il de paramètres  $q$  ?

# Modèle fondé sur le contexte gauche

- Chaque symbole (mots)  $O_i$  est généré de gauche à droite, conditionné par tous les mots qui précèdent

$$\begin{aligned} P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) &= \prod_{i=1}^n P(O_i | \text{start}, O_1, \dots, O_{i-1}) \\ &= \prod_{i=1}^n \gamma(O_i | \text{start}, O_1, \dots, O_{i-1}) \end{aligned}$$

- Ce modèle aurait une très grande expressivité, mais...
- Combien a-t-il de paramètres  $q$  ?
- Quelle probabilité aurait une phrase pas vue pendant l'apprentissage ?

# Modèle beaucoup plus simple : "Sac de mots"

- Modèle *Bag of words* ou "Sac de mots"
- Chaque symbole (mots)  $O_i$  est indépendant de tous les autres

$$\begin{aligned} P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) &= \prod_{i=1}^n P(O_i) \\ &= \prod_{i=1}^n \gamma(O_i) \end{aligned}$$

- Ce modèle aurait une très grande expressivité, mais...
- Combien de paramètres a ce modèle ?
- Quelle probabilité aurait une phrase pas vue pendant l'apprentissage ?

start

one die:



start |

one die:



start | want

one die:



start I want a

one die:





start I want a flight

one die:



start I want a flight to

one die:



start I want a flight to Lisbon

one die:



start I want a flight to Lisbon .

one die:



start I want a flight to Lisbon . stop

one die:



# Modèle beaucoup plus simple : "Sac de mots"

- Modèle *Bag of words* ou "Sac de mots"
- Chaque symbole (mots)  $O_i$  est indépendant de tous les autres

$$\begin{aligned} P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) &= \prod_{i=1}^n P(O_i) \\ &= \prod_{i=1}^n \gamma(O_i) \end{aligned}$$

- Cette hypothèse très forte ne permet pas de bien modéliser les données

# Modèle beaucoup plus simple : "Sac de mots"

- Modèle *Bag of words* ou "Sac de mots"
- Chaque symbole (mots)  $O_i$  est indépendant de tous les autres

$$\begin{aligned} P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) &= \prod_{i=1}^n P(O_i) \\ &= \prod_{i=1}^n \gamma(O_i) \end{aligned}$$

- Cette hypothèse très forte ne permet pas de bien modéliser les données
- Combien de paramètres a ce modèle ?

# Modèle beaucoup plus simple : "Sac de mots"

- Modèle *Bag of words* ou "Sac de mots"
- Chaque symbole (mots)  $O_i$  est indépendant de tous les autres

$$\begin{aligned} P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) &= \prod_{i=1}^n P(O_i) \\ &= \prod_{i=1}^n \gamma(O_i) \end{aligned}$$

- Cette hypothèse très forte ne permet pas de bien modéliser les données
- Combien de paramètres a ce modèle ?
- Quelle probabilité aurait une phrase pas vue pendant l'apprentissage ?

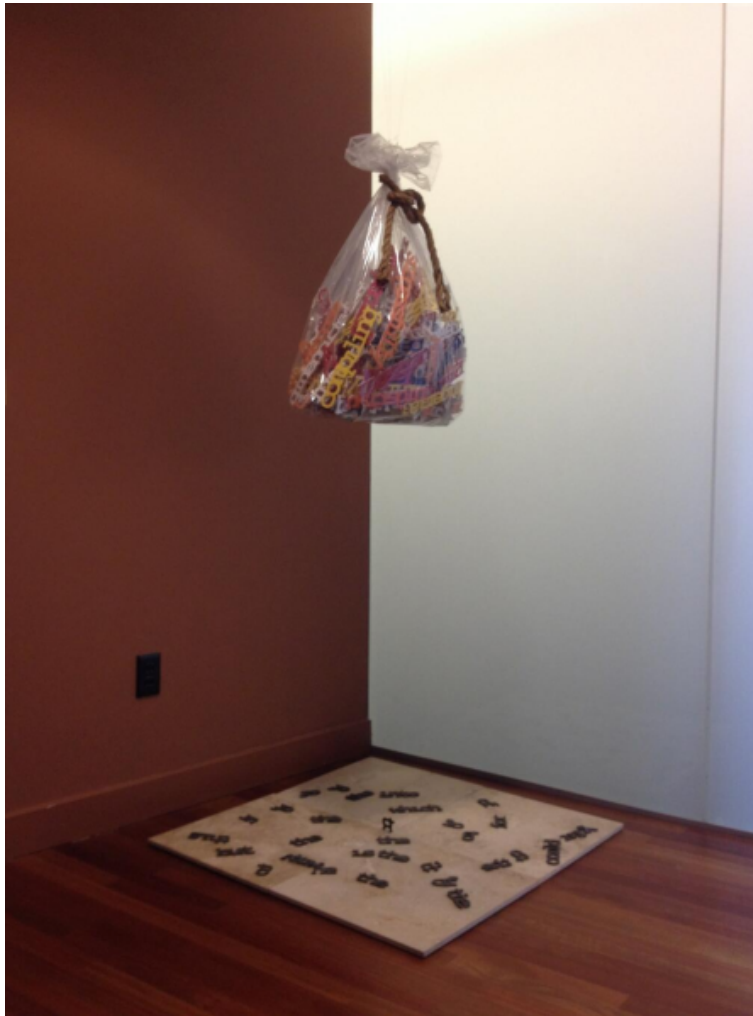


# Modèle beaucoup plus simple : "Sac de mots"

- Modèle *Bag of words* ou "Sac de mots"
- Chaque symbole (mots)  $O_i$  est indépendant de tous les autres

$$\begin{aligned} P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) &= \prod_{i=1}^n P(O_i) \\ &= \prod_{i=1}^n \gamma(O_i) \end{aligned}$$

- Cette hypothèse très forte ne permet pas de bien modéliser les données
- Combien de paramètres a ce modèle ?
- Quelle probabilité aurait une phrase pas vue pendant l'apprentissage ?
- Remarque : dans le jargon des modèles de langage, ce modèle s'appelle modèle "unigramme" (*unigram*)

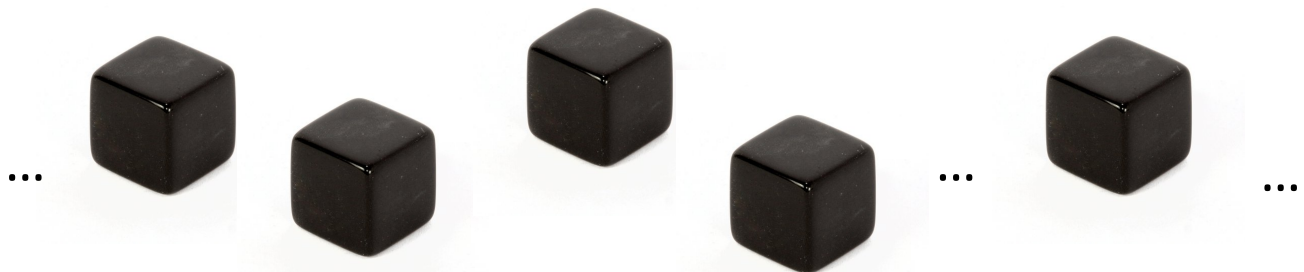


- Chaque symbole (mots)  $O_i$  est conditionné par le précédent  $O_{i-1}$  et seulement par le précédent :

$$\begin{aligned} P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) &= \prod_{i=1}^n P(O_i | O_{i-1}) \\ &= \prod_{i=1}^n \gamma(O_i | O_{i-1}) \end{aligned}$$

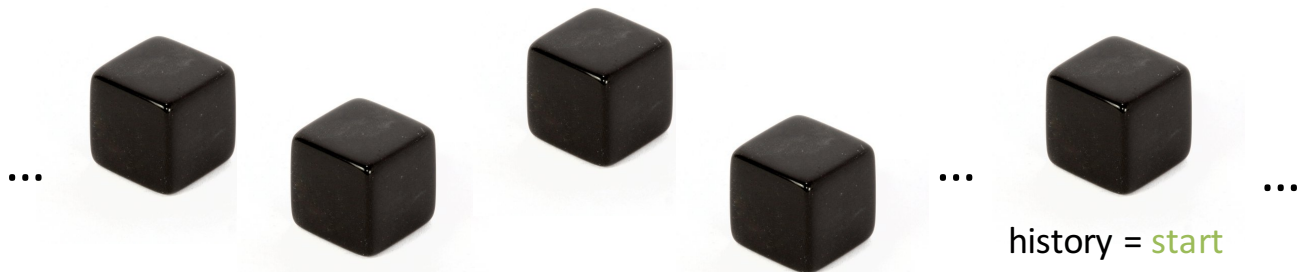
start

one die per history:



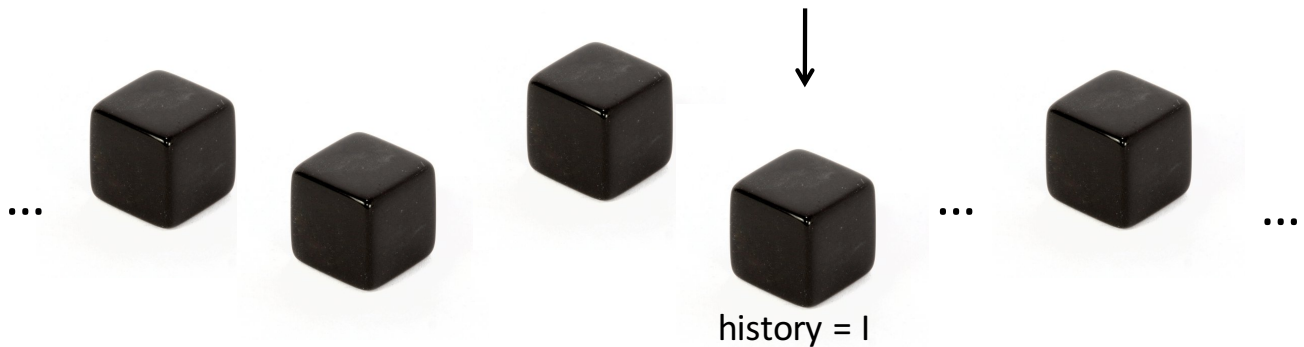


one die per history:




start I want

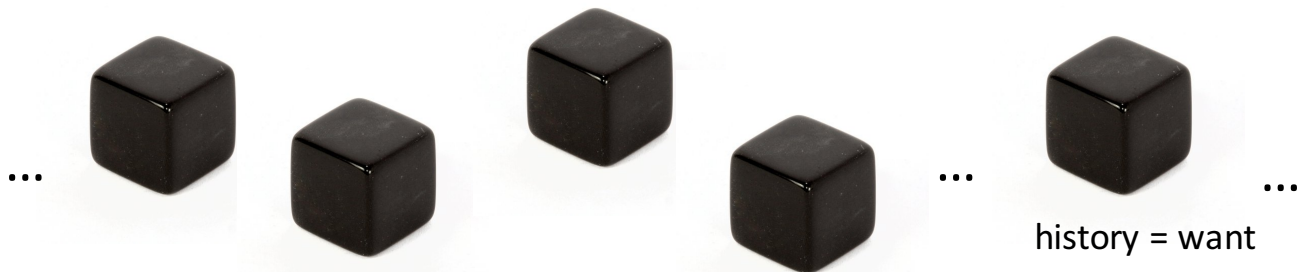
one die per history:



start I want a



one die per history:



start I want a flight

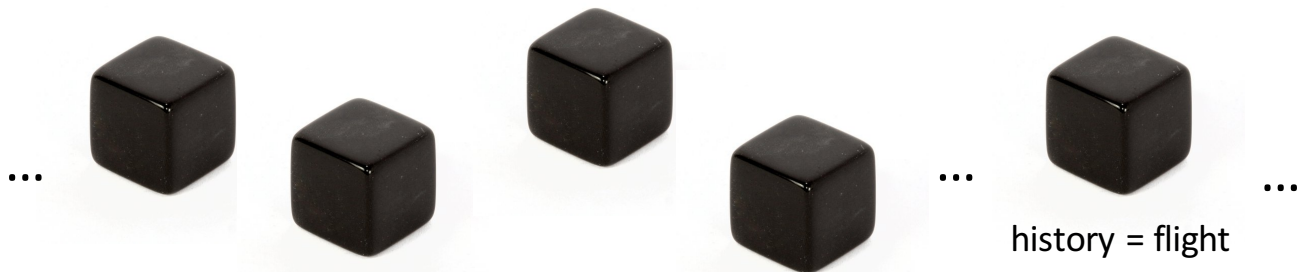
one die per history:





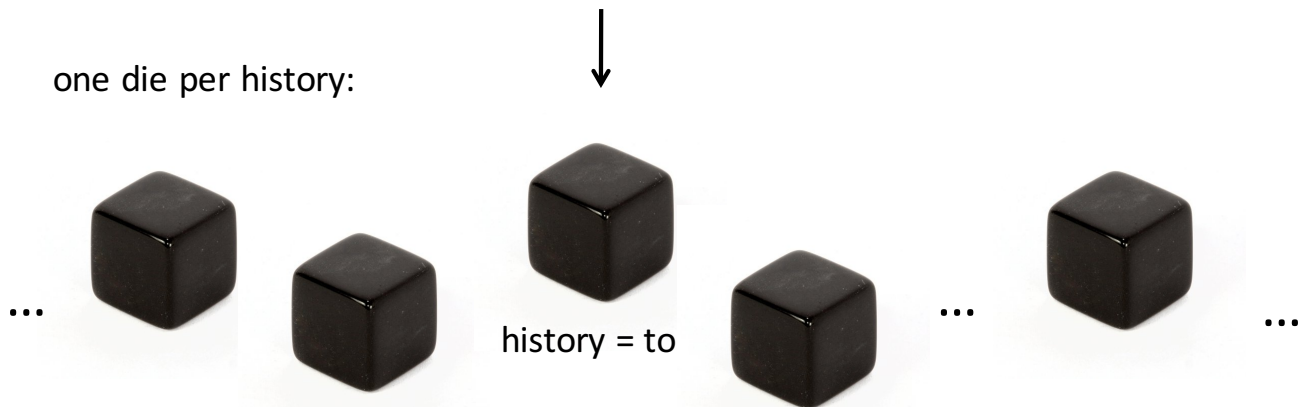
start I want a flight to

one die per history:



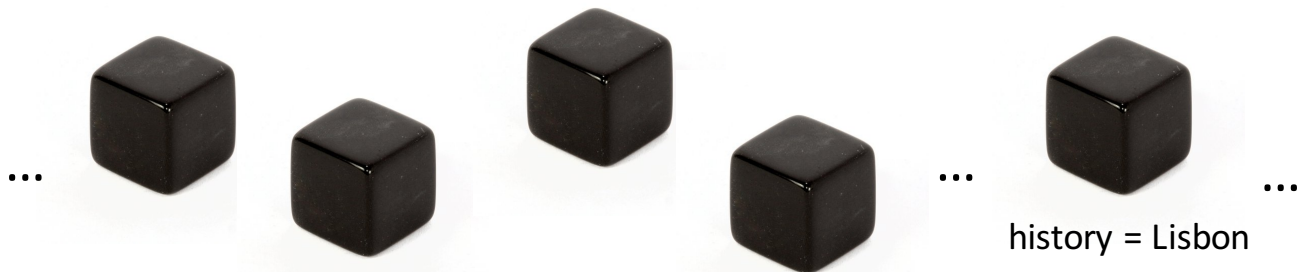
start I want a flight to Lisbon

one die per history:



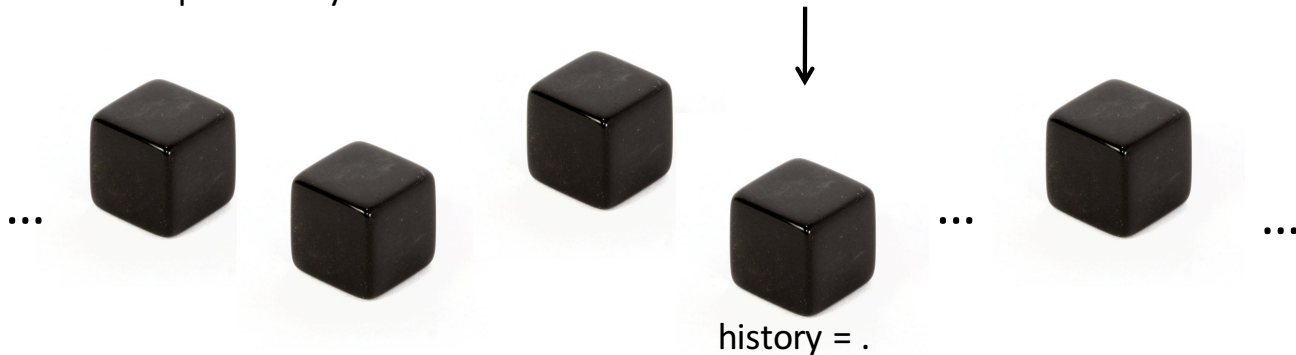
start I want a flight to Lisbon .

one die per history:



start I want a flight to Lisbon . stop

one die per history:



# Modèle entre les deux : modèle de Markov du premier ordre

- Chaque symbole (mots)  $O_i$  est conditionné par le précédent  $O_{i-1}$  et seulement par le précédent :

$$\begin{aligned} P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) &= \prod_{i=1}^n P(O_i | O_{i-1}) \\ &= \prod_{i=1}^n \gamma(O_i | O_{i-1}) \end{aligned}$$

- Cette hypothèse est souvent appelée "l'hypothèse de Markov"

# Modèle entre les deux : modèle de Markov du premier ordre

- Chaque symbole (mots)  $O_i$  est conditionné par le précédent  $O_{i-1}$  et seulement par le précédent :

$$\begin{aligned} P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) &= \prod_{i=1}^n P(O_i | O_{i-1}) \\ &= \prod_{i=1}^n \gamma(O_i | O_{i-1}) \end{aligned}$$

- Cette hypothèse est souvent appelée "l'hypothèse de Markov"
- Combien de paramètres a ce modèle ?

# Modèle entre les deux : modèle de Markov du premier ordre

- Chaque symbole (mots)  $O_i$  est conditionné par le précédent  $O_{i-1}$  et seulement par le précédent :

$$\begin{aligned} P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) &= \prod_{i=1}^n P(O_i | O_{i-1}) \\ &= \prod_{i=1}^n \gamma(O_i | O_{i-1}) \end{aligned}$$

- Cette hypothèse est souvent appelée "l'hypothèse de Markov"
- Combien de paramètres a ce modèle ?
- Quelle probabilité aurait une phrase pas vue pendant l'apprentissage ?

# Modèle entre les deux : modèle de Markov du premier ordre

- Chaque symbole (mots)  $O_i$  est conditionné par le précédent  $O_{i-1}$  et seulement par le précédent :

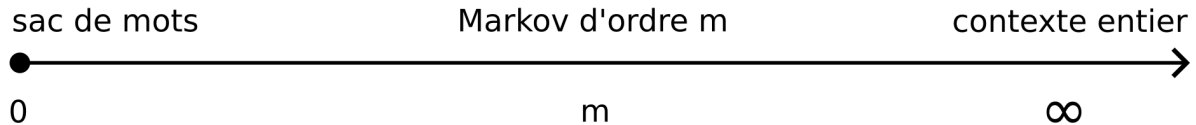
$$\begin{aligned} P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) &= \prod_{i=1}^n P(O_i | O_{i-1}) \\ &= \prod_{i=1}^n \gamma(O_i | O_{i-1}) \end{aligned}$$

- Cette hypothèse est souvent appelée "l'hypothèse de Markov"
- Combien de paramètres a ce modèle ?
- Quelle probabilité aurait une phrase pas vue pendant l'apprentissage ?
- Remarque : dans le jargon des modèles de langage, ce modèle s'appelle modèle "bigramme" (*bigram*)



# Modèle de Markov d'ordre $m$

$$P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) = \prod_{i=1}^n P(O_i | O_{i-m}, \dots, O_{i-1})$$



peu de paramètres

pouvoir expressif riche

hypothèses d'indépendance fortes

# Example

- Unigram model estimated on 2.8M words of American political blog text.

this trying our putting and funny  
and among it herring it obama  
but certainly foreign my  
c on byron again but from i  
i so and i chuck yeah the as but but republicans if  
this stay oh so or it mccain bush npr this with what  
and they right i while because obama

# Example

- Bigram model estimated on 2.8M words of American political blog text.

the lack of the senator mccain hadn t keep this story  
backwards  
while showering praise of the kind of gop weakness  
it was mistaken for american economist anywhere in the  
white house press hounded the absence of those he s as  
a wide variety of this election day after the candidate  
b richardson was polled ri in hempstead moderated by  
the convention that he had zero wall street journal  
argues sounds like you may be the primary  
but even close the bill told c e to take the obama on  
the public schools and romney  
fred flinstone s see how a lick skillet road it s  
little sexist remarks

# Example

- Trigram model estimated on 2.8M words of American political blog text.

as i can pin them all none of them want to bet that  
any of the might be  
conservatism unleashed into the privacy rule book and  
when told about what paul  
fans organized another massive fundraising initiative  
yesterday and i don t know what the rams supposedly  
want ooh  
but she did but still victory dinner  
alone among republicans there are probably best not  
all of the fundamentalist community  
asked for an independent maverick now for  
crystallizing in one especially embarrassing

# Example

- 5-gram model estimated on 2.8M words of American political blog text.

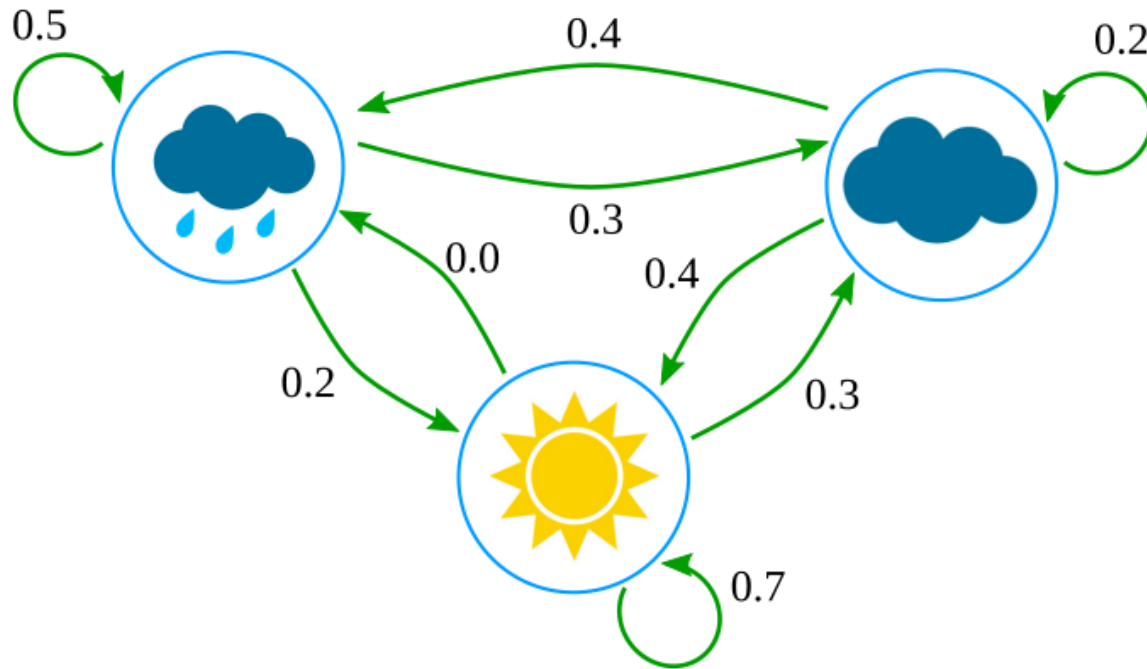
he realizes fully how shallow and insincere  
conservative behavior has been he realizes that there  
is little way to change the situation  
this recent arianna huffington item about mccain  
issuing heartfelt denials of things that were actually  
true or for that matter about the shia sunni split and  
which side iran was on would get confused about this  
any more than someone with any knowledge of us politics  
would get confused about whether neo confederates were  
likely to be supporting the socialist workers party  
at the end of the world and i m not especially  
discouraged now that newsweek shows obama leading by  
three now

# Example

- 100-gram model estimated on 2.8M words of American political blog text.

and it would be the work of many hands to catalogue all the ridiculous pronouncements made by this man since his long train of predictions about the middle east has been gaudily disastrously stupefyingly misinformed just the buffoon it seems for the new york times to award with a guest column for if you object to the nyt rewarding failure in quite this way then you re intolerant according to the times editorial page editor andrew rosenthal rosenthal doesn t seem to recognize that his choice of adjectives to describe kristol serious respected are in fact precisely what is at issue for those whom he dismisses as having a fear of opposing views

# Revenons au premier ordre : autre exemple classique



# Autre exemple classique de chaîne de Markov du premier ordre

Nous allons définir :

- les hypothèses simplificatrices du modèle
- des états et leurs indices arbitraires
- la matrice de transition  $A$
- la distribution stationnaire des états : le vecteur  $\pi$

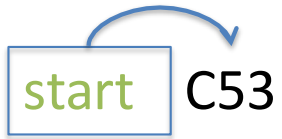


- Les modèles de séquences fondés sur des classes, de Brown *et al* (1990) :

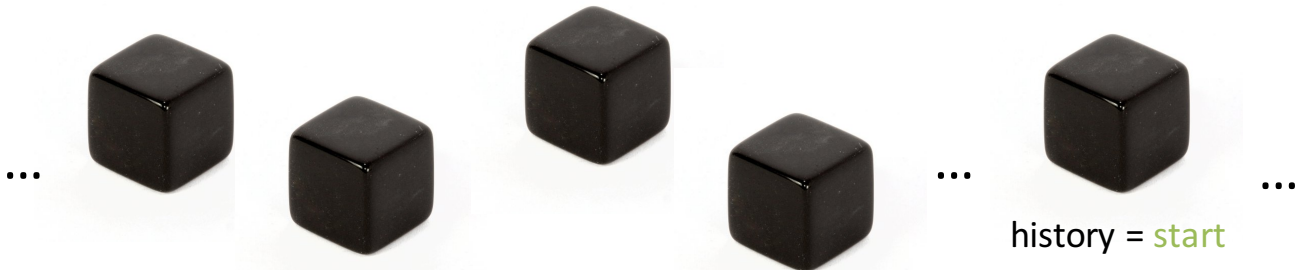
$$P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) = \prod_{i=1}^n \eta(O_i | \text{cl}(O_i)) \gamma(\text{cl}_i | \text{cl}_{i-1})$$

- $\text{cl}$  est une fonction qui associe à chaque mot  $O_i$  une classe, avec un nombre de classes possibles beaucoup plus petit que le nombre de mots différents :
  - Chaque est associé à une seule classe, connue à l'avance
  - Les classes sont déterminées par un algorithme de clustering

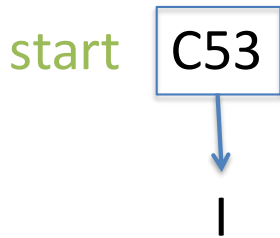
start



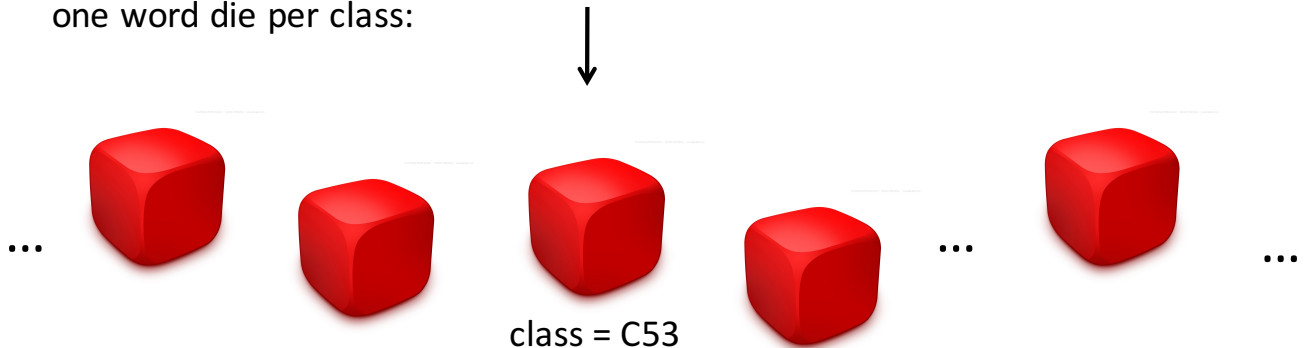
one “next class” die per class:



Each word appears on  
only one of the word dice.



one word die per class:

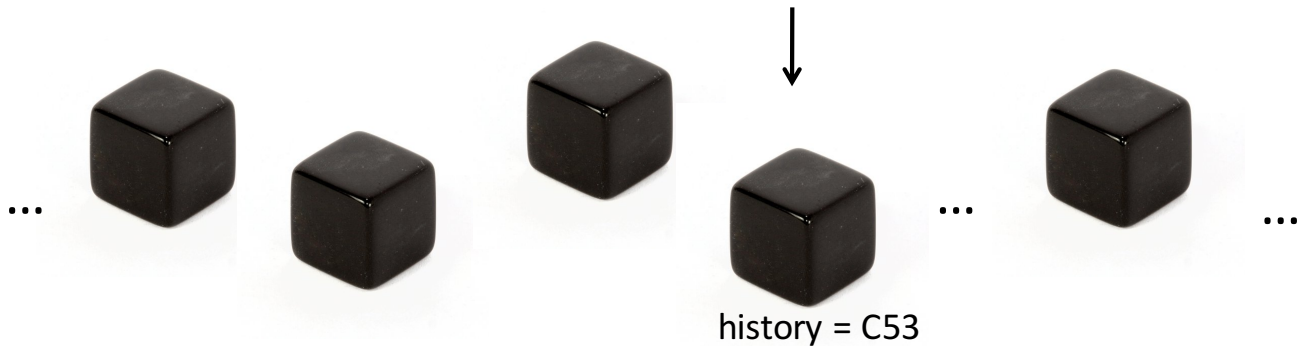


start C53 C23



|

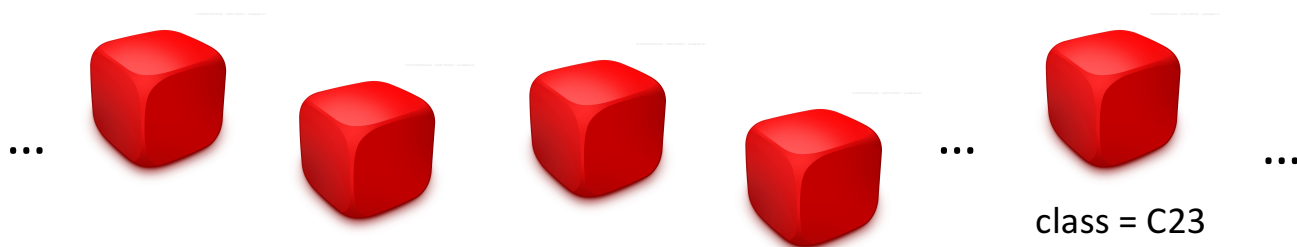
one “next class” die per class:



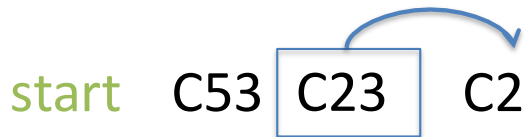
start C53 C23

I want

one word die per class:



start C53 C23 C2



I want

one “next class” die per class: ↓

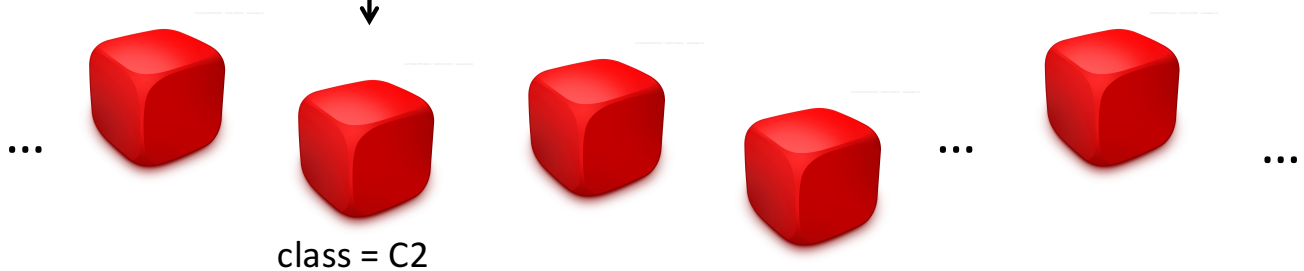


start C53 C23 C2



I want a

one word die per class:



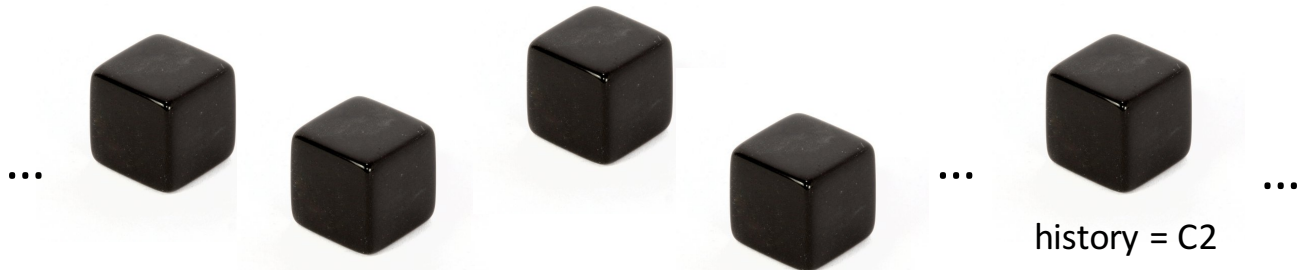


start C53 C23 C2 C5



I want a

one “next class” die per class:

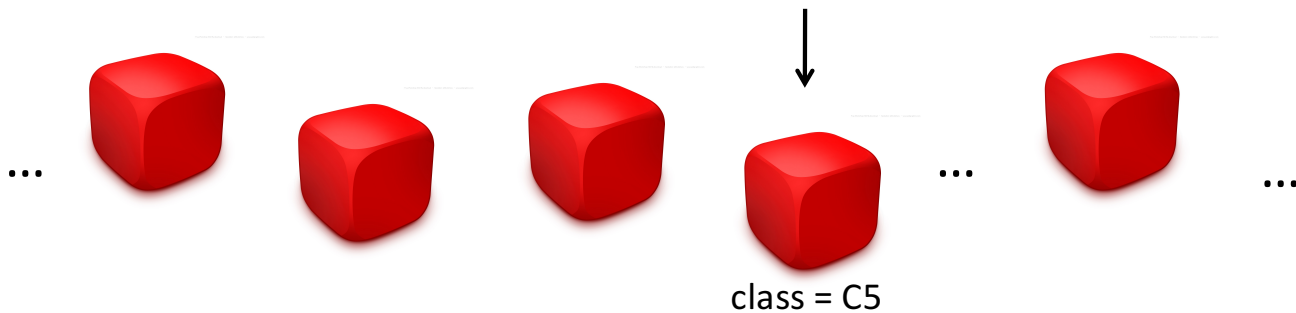


start C53 C23 C2 C5



I want a flight

one word die per class:



- Les modèles de séquences fondés sur des classes, de Brown *et al* (1990) :

$$P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) = \prod_{i=1}^n \eta(O_i \mid \text{cl}(O_i)) \gamma(\text{cl}_i \mid \text{cl}_{i-1})$$

- Hypothèses d'indépendance ?
- Nombre de paramètres du modèle ?
- Pouvoir de généralisation ?

# Les modèles de Markov Cachés

## *Hidden Markov Models, HMM*

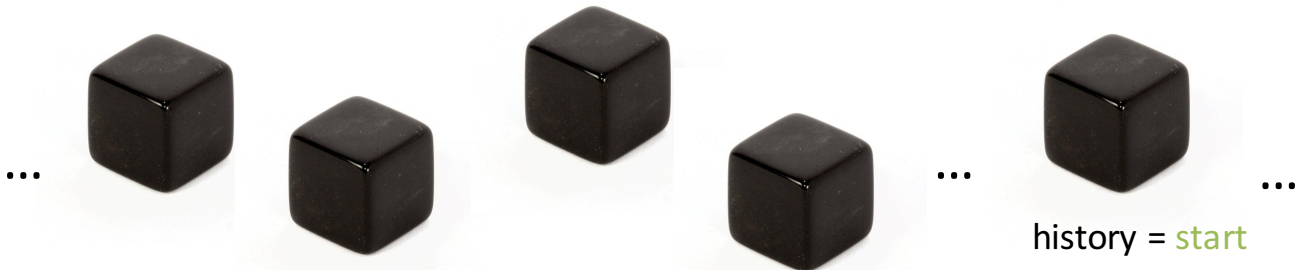
- Modèles sur des séquences de symboles, mais il y a de l'information associée à chaque symbole qui manque : leur "état"
- On fait l'hypothèse que le nombre d'états possible est fini

$$P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) = \prod_{i=1}^{n+1} \eta(O_i | S_i) \gamma(S_i | S_{i-1})$$

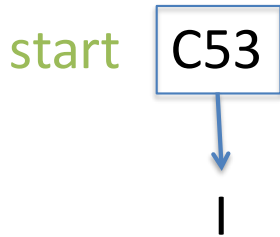
- Modèle joint sur des séquences de symboles **observables** et **d'états dits cachés**/latents/inconnus

start C53

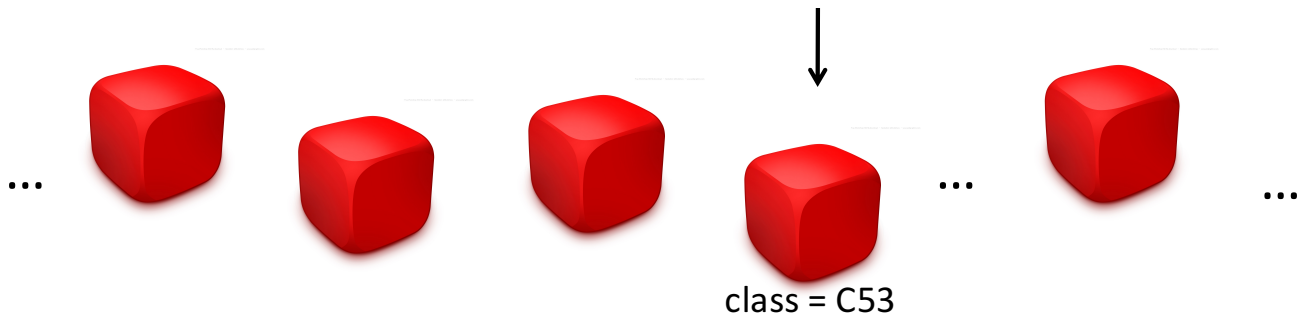
one “next class” die per class:



The only change to the class-based model is that now, the different dice can *share words*!



one word die per class:

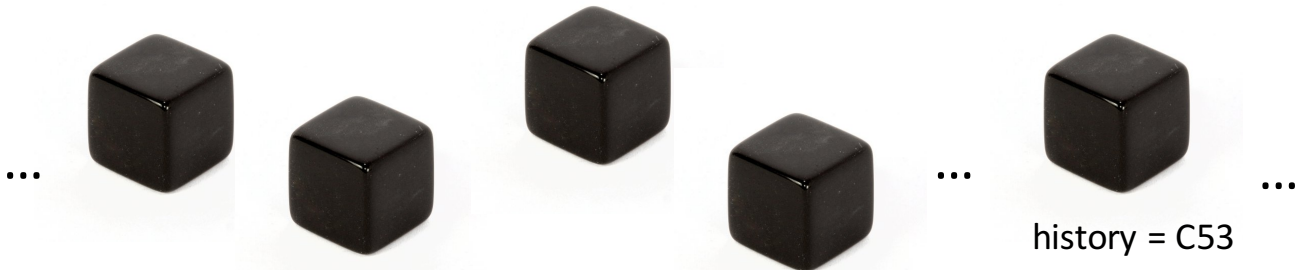


start C53 C23




|

one “next class” die per class:



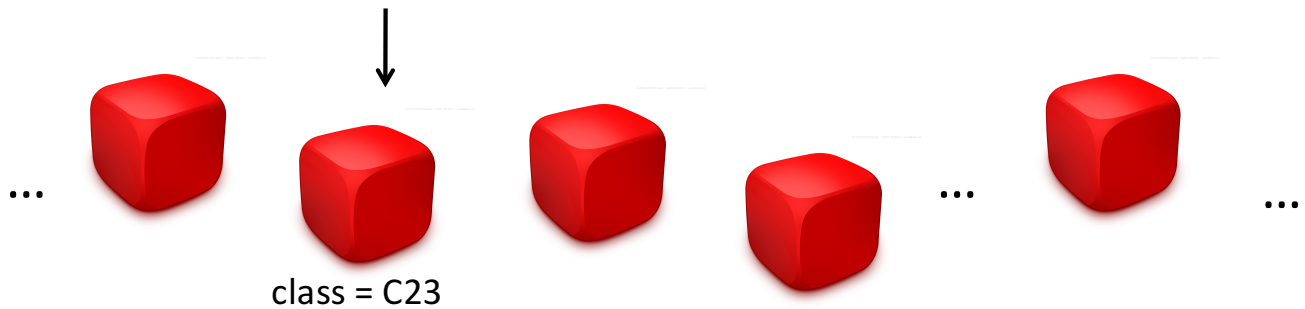


start C53 C23

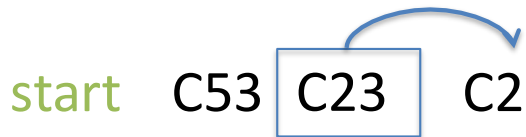


I want

one word die per class:

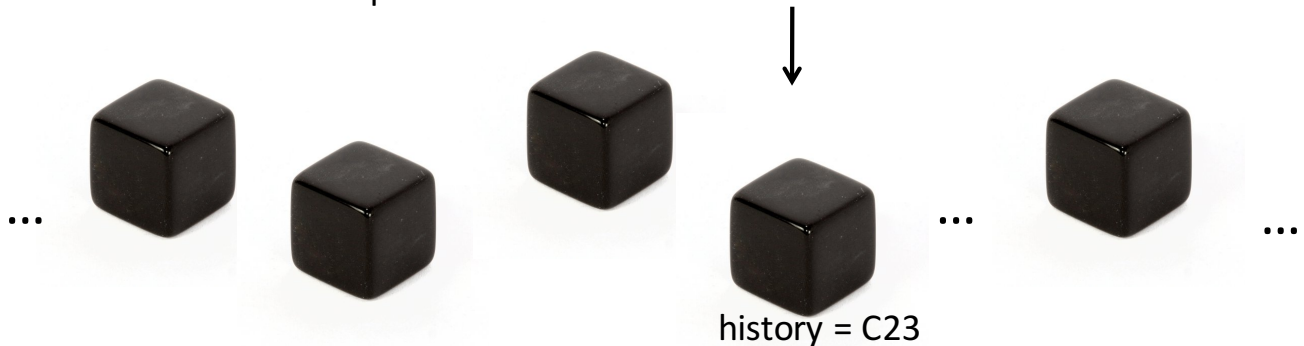


start C53 C23 C2



I want

one “next class” die per class:

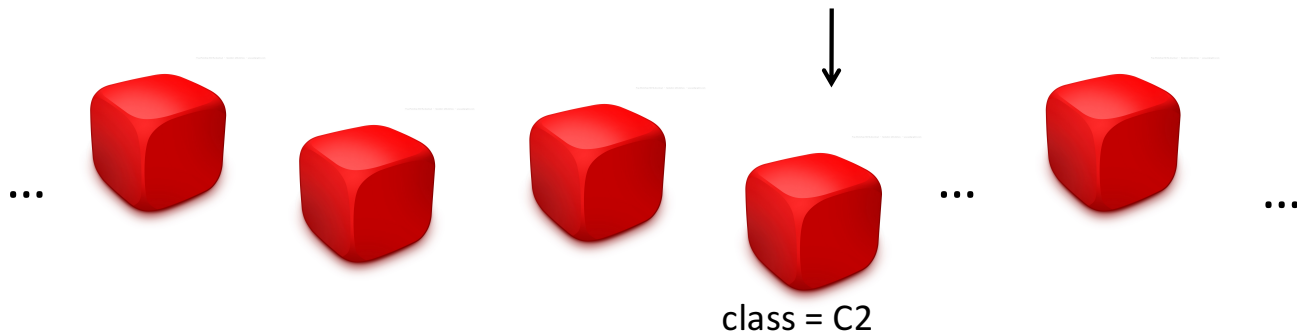


start C53 C23 C2



I want a

one word die per class:

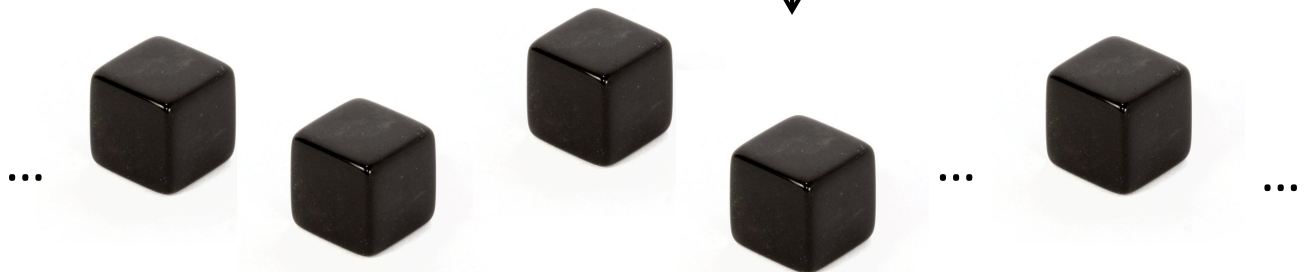


start C53 C23 C2 C5



I want a

one “next class” die per class:



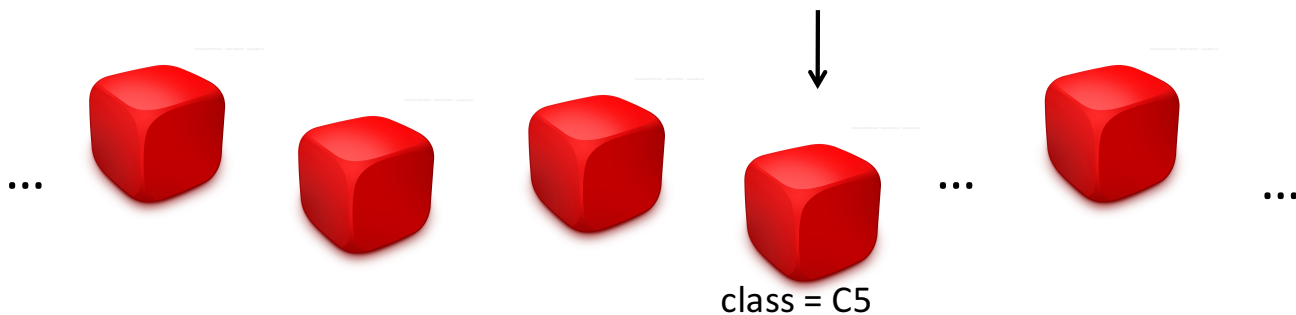
history = C2

start C53 C23 C2 C5



I want a flight

one word die per class:



# Deux façons équivalentes de générer des séquences

- Méthode 1 : transition, émission, transition, émission, etc



- Méthode 2 :

- Générer la séquence de transition entre états. Il s'agit d'une chaîne de Markov sur les états (vus comme des classes)
- Ensuite remplacer chaque état/classe par un mot

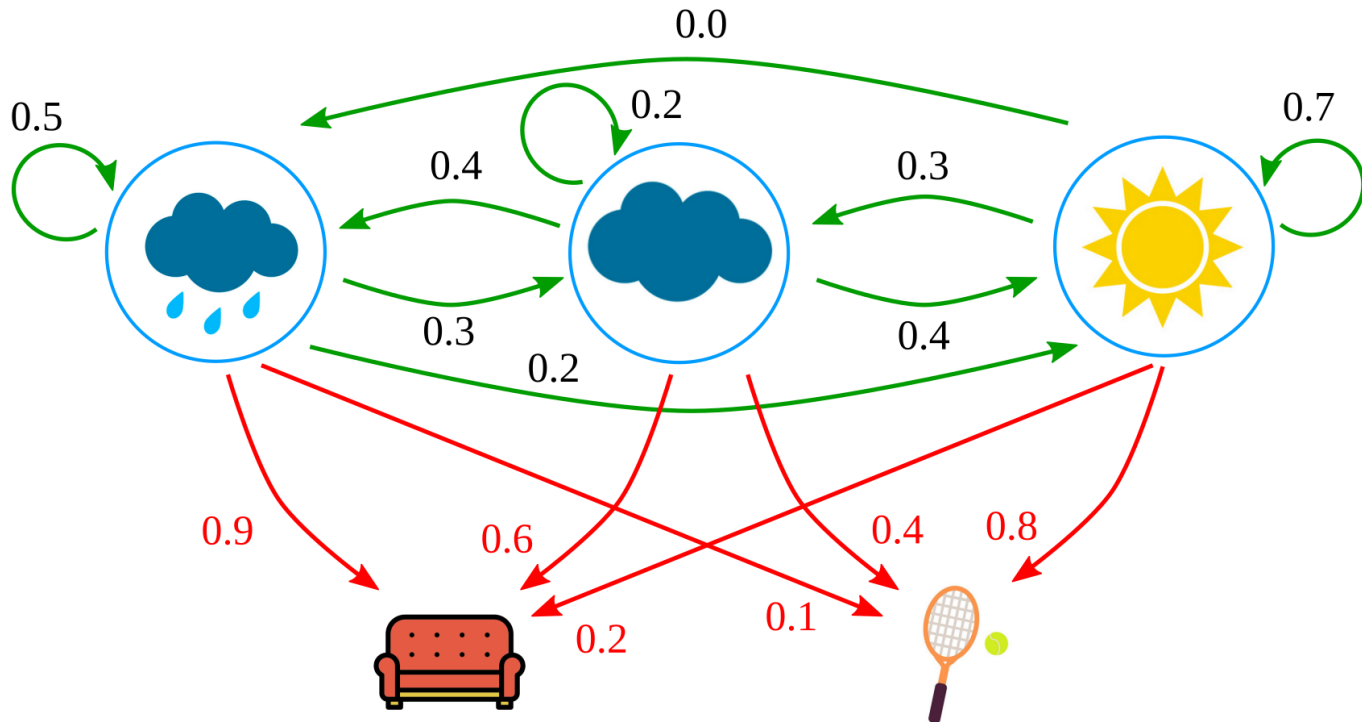


- tout comme les chaînes de Markov, on peut généraliser le modèle du premier ordre et prendre en compte un contexte plus grand :

$$P(\text{start}, O_1, O_2, \dots, O_n, \text{stop}) = \prod_{i=1}^{n+1} \eta(O_i | S_i) \gamma(S_i | S_{i-m}, \dots, S_{i-1})$$

- Nombre de paramètres ?
- Avantage : plus longue "mémoire"
- Dans ce cours, on ne va considérer que le premier ordre

# Revenons à la météo



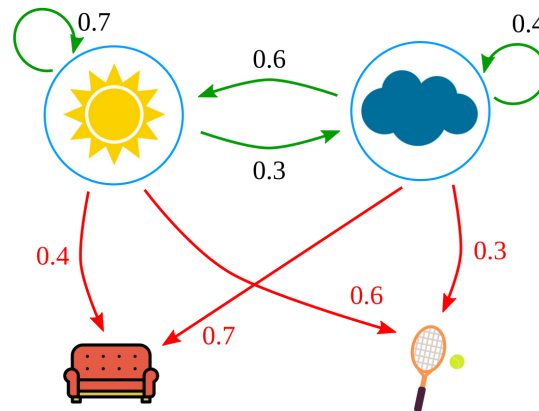


Nous allons :

- définir la matrice d'émissions  $B$
- définir les trois problèmes classiques du tutoriel de Rabiner (1989) :
  - ① l'évaluation
  - ② l'inférence ou décodage
  - ③ l'apprentissage des paramètres des HMM
- Pour chacun de ces problèmes, des algorithmes ingénieux ont été proposés et peuvent toujours être utilisés dans le cadre des réseaux de neurones profonds

# Problème 1 : l'évaluation

Considérons le modèle plus simple à deux états et deux observations suivant :



- Comment calculer la probabilité de la séquence de longueur 3 : *Tennis, Tennis, Canapé* ?
  - La complexité est exponentielle par rapport à la taille de la séquence...
- Algorithmes de programmation dynamique Forward (et Backward)

## Problème 2 : l'inférence ou le décodage

I	suspect	the	present	forecast	is	pessimistic
CD	JJ	DT	JJ	NN	NNS	JJ
NN	NN	JJ	NN	VB	VBZ	
NNP	VB	NN	RB	VBD		
PRP	VBP	NNP	VB	VBN		
		VBP	VBP	VBP		
4	4	5	5	5	2	1

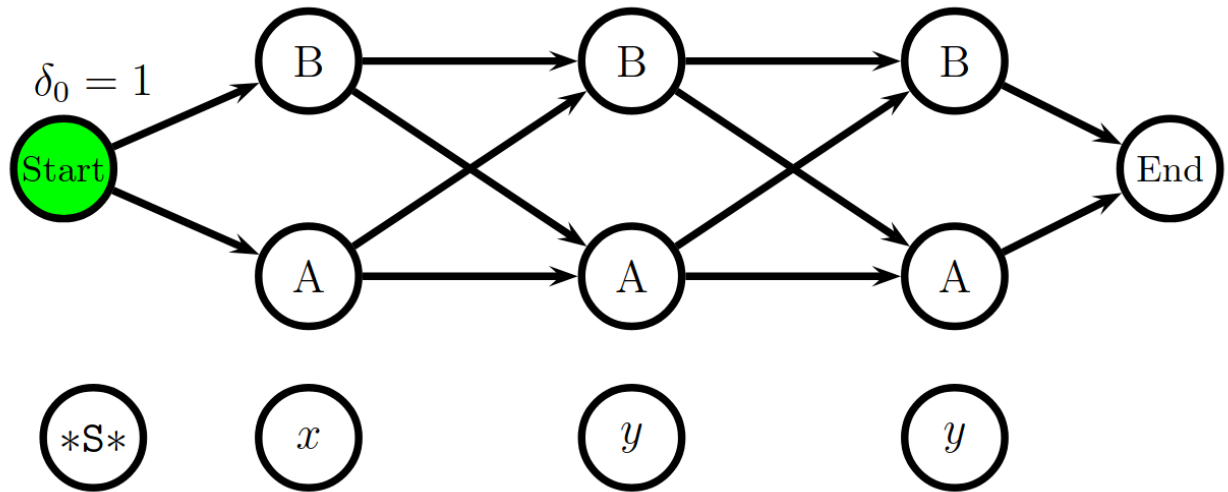
- Part Of Speech tagging
- Combien de séquences d'états possibles dans cet exemple ?

## Problème 2 : l'inférence ou le décodage

	time	flies	like	an	arrow	.
DT				10e-15	6e-21	
IN			8e-13		1e-19	
JJ			6e-14		2e-16	
NN	2e-4				3e-16	
NNP					1e-16	
VB	2e-7		1e-14		1e-19	
VBP			8e-16		4e-19	
VBZ		2e-9			3e-18	
.					1e-21	3e-17
,				4e-20	5e-22	

- Part Of Speech tagging
- Exemple de décodage Viterbi

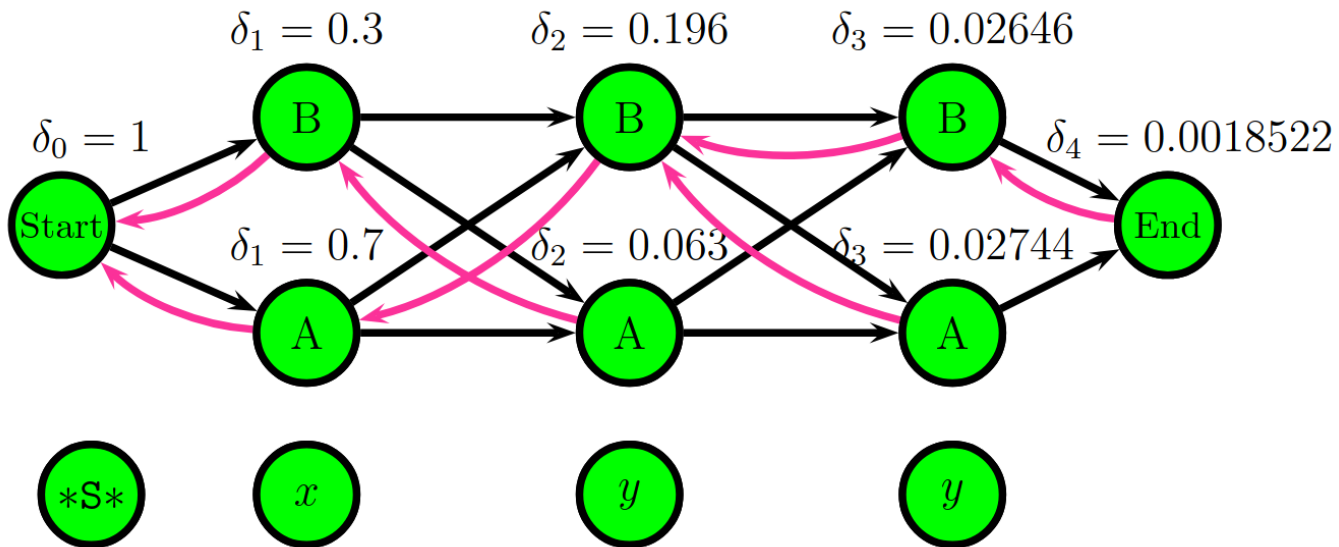
## Problème 2 : l'inférence ou le décodage



- Exemple de décodage Viterbi : séquence  $x \ x \ y$  et deux états possibles A et B
- Représentation sous forme de **treillis**

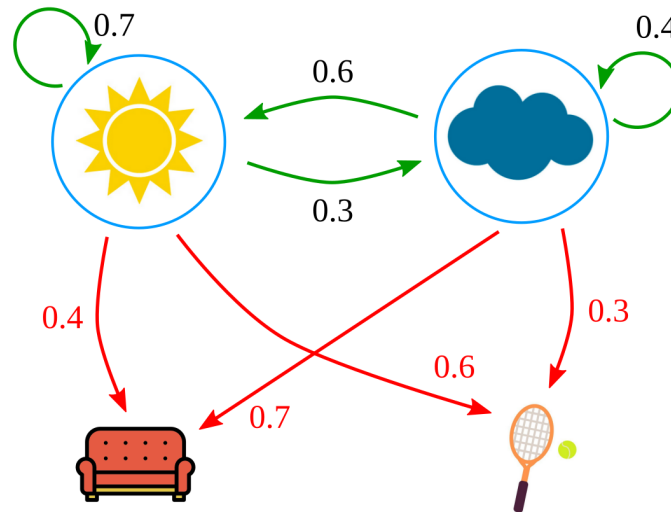
source : [https://www.davidsbatista.net/assets/documents/posts/2017-11-11-hmm\\_viterbi\\_mini\\_example.pdf](https://www.davidsbatista.net/assets/documents/posts/2017-11-11-hmm_viterbi_mini_example.pdf)

## Problème 2 : l'inférence ou le décodage



- Exemple de décodage Viterbi : séquence  $x\ y\ y$  et deux états possibles A et B
- Résultat :
  - Séquence Viterbi : ABB
  - $P(\text{ABB}, xyy) = 0.00118522$

## Problème 2 : l'inférence ou le décodage



### Exercice

- Appliquer Viterbi pour trouver la séquence météo (séquence d'états) la plus vraisemblable pour la séquence *Tennis, Tennis, Canapé* ?

# Éviter les underflows : domaine log

- On voit bien qu'au long d'une séquence, les probabilités deviennent de plus en plus petites
- Cela peut causer des underflows et de l'instabilité numérique
- La solution : plutôt que multiplier des probabilités, sommer des log-probabilités. Si  $a$  et  $b$  sont des log-probabilités, pour multiplier deux probabilités, on a :

$$\log(\exp a \times \exp b) = a + b$$

- Pour sommer deux probabilités, dans le domaine log il faut coder une fonction logsum :

$$\log(\exp a + \exp b) = a + \log(1 + \exp(b - a))$$

- Voir la fonction logsumexp du TD 1



# Comparaison Viterbi et forward

	maximisations et multiplications	additions et multiplications
de gauche à droite	Viterbi	Forward
de droite à gauche	?	Backward

- Il n'existe pas de Viterbi backward, mais il n'y a pas de raison à cela

# Comparaison Viterbi et forward

	maximisations et multiplications	additions et multiplications
de gauche à droite	Viterbi	Forward
de droite à gauche	?	Backward

- Il n'existe pas de Viterbi backward, mais il n'y a pas de raison à cela
- Viterbi maximise et forward somme, mais les mêmes calculs sont faits

# Comparaison Viterbi et forward

	maximisations et multiplications	additions et multiplications
de gauche à droite	Viterbi	Forward
de droite à gauche	?	Backward

- Il n'existe pas de Viterbi backward, mais il n'y a pas de raison à cela
- Viterbi maximise et forward somme, mais les mêmes calculs sont faits
- Viterbi et forward sont les mêmes algorithmes abstraits, instanciés dans deux différents semi-anneaux (*semi-rings*)

# Comparaison Viterbi et forward

	maximisations et multiplications	additions et multiplications
de gauche à droite	Viterbi	Forward
de droite à gauche	?	Backward

- Il n'existe pas de Viterbi backward, mais il n'y a pas de raison à cela
- Viterbi maximise et forward somme, mais les mêmes calculs sont faits
- Viterbi et forward sont les mêmes algorithmes abstraits, instanciés dans deux différents semi-anneaux (*semi-rings*)
- Un semi-anneau est un ensemble de valeurs et quelques opérations qui respectent certaines propriétés

	Réel	Viterbi
Support	réels non-négatifs	réels non-négatifs
"zéro"	0	0
"un"	1	1
"plus"	+	max
"fois"	x	x

- Booléen : utilisé pour déterminer si le HMM peut produire une séquence donnée ou non
- Comptage : déterminer combien il y a de labels valides dans une séquence
- réel domaine log : utilise les log-probas pour éviter les underflows
- k-best : déterminer k hypothèses au lieu d'une seule
- coût minimal (*min-cost*) : utiliser la distance d'édition et les algorithmes de Dijkstra

# Décodage *a posteriori* ou *Posterior decoding*

## Décodage *a posteriori* : une alternative au décodage Viterbi

Principe : calculer la probabilité de tous les états à un temps  $t$  donné, en ayant "vu" la séquence d'observations (d'où le nom de *a posteriori*) :  $P(S^t = S_i | O)$

- Pour cela, on calcule d'abord la probabilité jointe  $P(S^t = S_i, O)$

# Décodage *a posteriori* ou *Posterior decoding*

## Décodage *a posteriori* : une alternative au décodage Viterbi

Principe : calculer la probabilité de tous les états à un temps  $t$  donné, en ayant "vu" la séquence d'observations (d'où le nom de *a posteriori*) :  $P(S^t = S_i | O)$

- Pour cela, on calcule d'abord la probabilité jointe  $P(S^t = S_i, O)$
- **Exercice** : montrer que  $P(S^t = S_i, O) = \alpha^t(S_i) \beta^t(S_i)$



# Décodage *a posteriori* ou *Posterior decoding*

## Décodage *a posteriori* : une alternative au décodage Viterbi

Principe : calculer la probabilité de tous les états à un temps  $t$  donné, en ayant "vu" la séquence d'observations (d'où le nom de *a posteriori*) :  $P(S^t = S_i | O)$

- Pour cela, on calcule d'abord la probabilité jointe  $P(S^t = S_i, O)$
- **Exercice** : montrer que  $P(S^t = S_i, O) = \alpha^t(S_i) \beta^t(S_i)$
- Puis

$$P(S^t = S_i | O) = \frac{P(S^t = S_i, O)}{P(O)} \quad (1)$$

$$= \frac{P(S^t = S_i, O)}{\sum_{S_j} P(S^t = S_j, O)} \quad (2)$$

$$= \frac{\alpha^t(S_i) \beta^t(S_i)}{\sum_{S_j} \alpha^t(S_j) \beta^t(S_j)} \quad (3)$$

# Décodage *a posteriori* ou *Posterior decoding*

## Décodage *a posteriori* : une alternative au décodage Viterbi

Principe : calculer la probabilité de tous les états à un temps  $t$  donné, en ayant "vu" la séquence d'observations (d'où le nom de *a posteriori*) :  $P(S^t = S_i | O)$

- Pour cela, on calcule d'abord la probabilité jointe

$$P(S^t = S_i, O) = \alpha^t(S_i) \beta^t(S_i)$$

- Puis

$$P(S^t = S_i | O) = \frac{\alpha^t(S_i) \beta^t(S_i)}{\sum_{S_j} \alpha^t(S_j) \beta^t(S_j)}$$

- Puis on trouve l'état le plus probable :

$$S_*^t = \arg \max_{j=0 \dots N-1} P(S^t = S_j | O) \quad (4)$$

$$= \arg \max_{j=0 \dots N-1} \alpha^t(S_j) \beta^t(S_j) \quad (5)$$

→ Il faut donc appliquer les algorithmes forward et backward

Quand utiliser un décodage Viterbi ou bien un décodage Posterior ?

- En général, ils ne donnent pas les mêmes séquences d'états
- Le décodage *a posteriori* peut donner des séquences qui ont une probabilité nulle
- Pas de meilleur algorithme

Distinguer deux cas :

- Supervisé : on a des exemples de séquences d'observations avec les séquences d'états. Dans ce cas, on compte les fréquences des événements
- Non-supervisé : on n'a que des exemples de séquences d'observations. Dans ce cas, on applique un algorithme d'Estimation-Maximisation (*Expectation Maximization* ou *EM*), qui remplace les fréquences par des probabilités (fréquences dites soft)

## Exercice

- 1 Soit trois tirages aléatoires :  $[1, 2, x]$ , faits à l'aide d'une loi normale  $\mathcal{N}(1, 1)$ .  $x$  est inconnu. Quelle est sa valeur la plus probable ?

## Exercice

- ① Soit trois tirages aléatoires :  $[1, 2, x]$ , faits à l'aide d'une loi normale  $\mathcal{N}(1, 1)$ .  $x$  est inconnu. Quelle est sa valeur la plus probable ?
- ② Soit trois tirages aléatoires :  $[1, 2, 3]$ , faits à l'aide d'une loi normale  $\mathcal{N}(\mu, 1)$ .  $\mu$  est inconnue. Quelle est sa valeur la plus probable ?

$$MU = (1 + 2 + X) / 3$$

### Exercice

- ❶ Soit trois tirages aléatoires :  $[1, 2, x]$ , faits à l'aide d'une loi normale  $\mathcal{N}(1, 1)$ .  $x$  est inconnu. Quelle est sa valeur la plus probable ?
- ❷ Soit trois tirages aléatoires :  $[1, 2, 3]$ , faits à l'aide d'une loi normale  $\mathcal{N}(\mu, 1)$ .  $\mu$  est inconnue. Quelle est sa valeur la plus probable ?
- ❸ Soit trois tirages aléatoires :  $[1, 2, x]$ , faits à l'aide d'une loi normale  $\mathcal{N}(\mu, 1)$ .  $x$  et  $\mu$  sont inconnus. Quelles sont leurs valeurs la plus probable ?

$$\mathcal{N}(\mu, 1)$$

# Problème 3 non-supervisé : digression EM, intuition

EM est un algorithme très général, utilisé au-delà des HMM quand :

- Des informations manquent sur les données (typiquement les labels),
- Les paramètres (parfois "latents") d'un modèle sont à estimer



# Problème 3 non-supervisé : digression EM, intuition

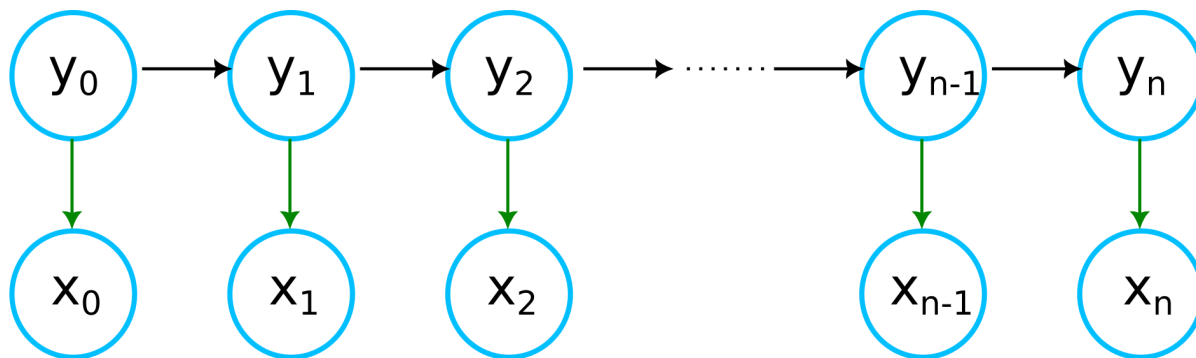
EM est un algorithme très général, utilisé au-delà des HMM quand :

- Des informations manquent sur les données (typiquement les labels),
- Les paramètres (parfois "latents") d'un modèle sont à estimer

Autres cas d'utilisation : les algorithmes de clustering

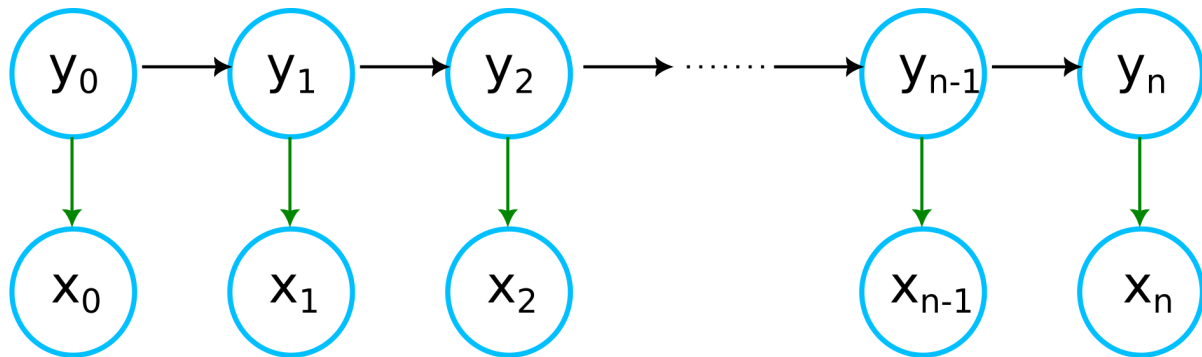
- La méthode k-means peut être vu comme un EM à décisions *hard*
- Le cas plus célèbre : clustering par mélange de Gaussiennes (clustering GMM)
  - On ne sait pas à quels clusters appartiennent les points (ici décisions *soft*)
  - On ne connaît pas les moyennes, matrices de variance-covariance et poids des Gaussiennes

# Représentation modèle graphe des HMM (*graphical model*)



- Nous avons appelé  $O$  et  $S$  les observations et les états cachés respectivement
- Mais le plus souvent, ce sont plutôt les lettres  $X$  et  $Y$  qui sont utilisées
- Exemple POS tagging :  $X$  est une séquence de mots, et  $Y$  une séquence de tags

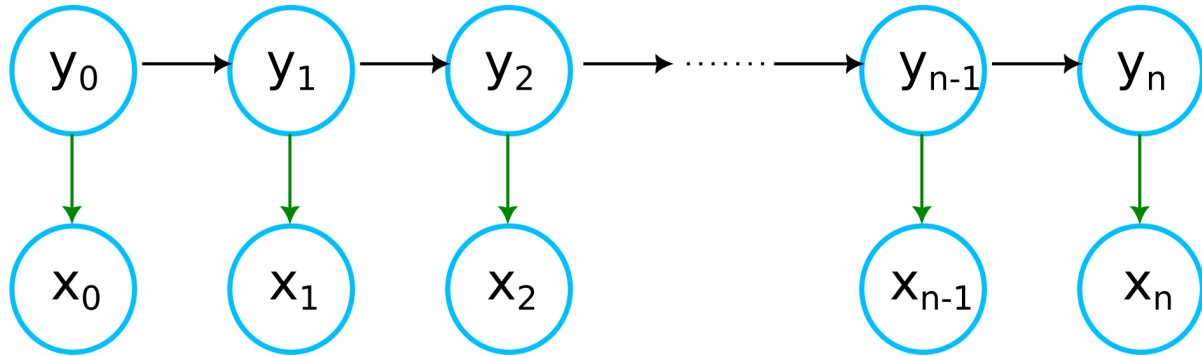
# Représentation modèle graphe des HMM (*graphical model*)



- Chaque noeud est une variable aléatoire.
- Une flèche qui arrive dans une VA indique quelle est ou quelles sont les autres VA qui la conditionnent.
- Les flèches noires correspondent aux probabilités de transition, les vertes aux probabilités d'émission.

Elle nous dit :

$$p(\mathbf{x}) = \prod_i p(x_i | \text{parents}(x_i))$$



- Les HMM capturent des dépendances seulement entre chaque état et l'observation qu'il émet
- Ils ne peuvent pas modéliser des interactions multiples
- Ils ne capturent pas non plus de dépendance à long terme (sauf à augmenter leur ordre)

- Représentation de graphe ? Modèle de séquence dirigé
- Hypothèses ? Les observations (mots ou symboles) sont indépendantes, étant donné leur état
- Inférence/décodage ? Algorithmes Viterbi et forward-backward
- Apprentissage des paramètres ? Vraisemblance maximale en supervisé, algorithme EM en non-supervisé (appelé Baum-Welch pour les HMM)