Simulation à mémoire finie de lois de probabilités

Philippe Duchon (LaBRI)

Aléa 2017

24 mars 2017

Simulation exacte de lois de probabilités

- On se donne une loi de probabilités μ (sur $\mathbb{R}$), le problème est de produire des réalisations de cette loi.

Simulation exacte de lois de probabilités

- On se donne une loi de probabilités μ (sur $\mathbb{R}$), le problème est de produire des réalisations de cette loi.
- Dans le modèle classique "arithmétique des réels", on suppose qu'on a la possibilité de tirer des uniformes sur $[0, 1]$ (le modèle des `random()` de la plupart des langages de programmation) ; **[Devroye 1986]** donne des algorithmes exacts pour la plupart des lois classiques.

Simulation exacte de lois de probabilités

- On se donne une loi de probabilités μ (sur $\mathbb{R}$), le problème est de produire des réalisations de cette loi.
- Dans le modèle classique "arithmétique des réels", on suppose qu'on a la possibilité de tirer des uniformes sur $[0, 1]$ (le modèle des `random()` de la plupart des langages de programmation) ; **[Devroye 1986]** donne des algorithmes exacts pour la plupart des lois classiques.
- Production "bit à bit" à partir de "pile ou face" : **[Knuth et Yao, 1976]**

Simulation exacte de lois de probabilités

- On se donne une loi de probabilités μ (sur $\mathbb{R}$), le problème est de produire des réalisations de cette loi.
- Dans le modèle classique "arithmétique des réels", on suppose qu'on a la possibilité de tirer des uniformes sur $[0, 1]$ (le modèle des `random()` de la plupart des langages de programmation) ; **[Devroye 1986]** donne des algorithmes exacts pour la plupart des lois classiques.
- Production "bit à bit" à partir de "pile ou face" : **[Knuth et Yao, 1976]**
  - Descente dans un arbre binaire infini, écriture à la volée

Simulation exacte de lois de probabilités

- On se donne une loi de probabilités μ (sur $\mathbb{R}$), le problème est de produire des réalisations de cette loi.
- Dans le modèle classique "arithmétique des réels", on suppose qu'on a la possibilité de tirer des uniformes sur $[0, 1]$ (le modèle des `random()` de la plupart des langages de programmation) ; **[Devroye 1986]** donne des algorithmes exacts pour la plupart des lois classiques.
- Production "bit à bit" à partir de "pile ou face" : **[Knuth et Yao, 1976]**
  - Descente dans un arbre binaire infini, écriture à la volée
  - Existence d'arbres optimaux pour *toute loi sur* $\mathbb{R}$

Simulation exacte de lois de probabilités

- On se donne une loi de probabilités μ (sur $\mathbb{R}$), le problème est de produire des réalisations de cette loi.
- Dans le modèle classique "arithmétique des réels", on suppose qu'on a la possibilité de tirer des uniformes sur $[0, 1]$ (le modèle des `random()` de la plupart des langages de programmation) ; **[Devroye 1986]** donne des algorithmes exacts pour la plupart des lois classiques.
- Production "bit à bit" à partir de "pile ou face" : **[Knuth et Yao, 1976]**
  - Descente dans un arbre binaire infini, écriture à la volée
  - Existence d'arbres optimaux pour *toute loi sur* $\mathbb{R}$
  - Pas forcément très algorithmique : les arbres sont infinis et n'ont pas de représentation finitaire

. . .avec mémoire finie ?

- **La question du jour :** que peut-on simuler **avec une mémoire finie ?**

. . .avec mémoire finie ?

- **La question du jour :** que peut-on simuler **avec une mémoire finie ?**
- Knuth et Yao donnent des résultats non triviaux

. . .avec mémoire finie ?

- **La question du jour :** que peut-on simuler **avec une mémoire finie ?**
- Knuth et Yao donnent des résultats non triviaux
  - Beaucoup de lois sont simulables à mémoire finie, mais de manière non optimale.

. . .avec mémoire finie ?

- **La question du jour :** que peut-on simuler **avec une mémoire finie ?**
- Knuth et Yao donnent des résultats non triviaux
  - Beaucoup de lois sont simulables à mémoire finie, mais de manière non optimale.
  - **Théorème :** si une loi simulable à mémoire finie a une densité *analytique* sur un intervalle, alors la densité est en fait un *polynôme* sur ce même intervalle.

. . .avec mémoire finie ?

- **La question du jour :** que peut-on simuler **avec une mémoire finie ?**
- Knuth et Yao donnent des résultats non triviaux
  - Beaucoup de lois sont simulables à mémoire finie, mais de manière non optimale.
  - **Théorème :** si une loi simulable à mémoire finie a une densité *analytique* sur un intervalle, alors la densité est en fait un *polynôme* sur ce même intervalle.
  - Toutes les lois à densité $C_{k,\ell} x^k (1-x)^\ell$ (sur $[0,1]$) sont simulables.

Le modèle (lois sur $[0, 1]$)

- Mémoire finie : ensemble fini d'états S

Le modèle (lois sur $[0, 1]$)

- Mémoire finie : ensemble fini d'états S
- Pour chaque état : définition d'un ensemble fini de transitions possibles, avec une loi de probabilités sur ces transitions ; chaque transition "écrit" un mot fini (qui peut être vide) sur $\{0, 1\}$ (écriture en binaire)

Le modèle (lois sur $[0, 1]$)

- Mémoire finie : ensemble fini d'états S
- Pour chaque état : définition d'un ensemble fini de transitions possibles, avec une loi de probabilités sur ces transitions ; chaque transition "écrit" un mot fini (qui peut être vide) sur $\{0, 1\}$ (écriture en binaire)
- On espère que l'automate écrit (avec probabilité 1) un mot infini, qu'on interprète comme le développement binaire d'un réel aléatoire entre 0 et 1.

Le modèle (lois sur $[0, 1]$)

- Mémoire finie : ensemble fini d'états S
- Pour chaque état : définition d'un ensemble fini de transitions possibles, avec une loi de probabilités sur ces transitions ; chaque transition "écrit" un mot fini (qui peut être vide) sur $\{0, 1\}$ (écriture en binaire)
- On espère que l'automate écrit (avec probabilité 1) un mot infini, qu'on interprète comme le développement binaire d'un réel aléatoire entre 0 et 1.
- Restrictions sur les probabilités et les mots écrits :

Le modèle (lois sur $[0, 1]$)

- Mémoire finie : ensemble fini d'états S
- Pour chaque état : définition d'un ensemble fini de transitions possibles, avec une loi de probabilités sur ces transitions ; chaque transition "écrit" un mot fini (qui peut être vide) sur $\{0, 1\}$ (écriture en binaire)
- On espère que l'automate écrit (avec probabilité 1) un mot infini, qu'on interprète comme le développement binaire d'un réel aléatoire entre 0 et 1.
- Restrictions sur les probabilités et les mots écrits :
  - proba $1/2$, mots $\{0, 1\}^*$ (modèle Knuth-Yao, "pile ou face")

Le modèle (lois sur $[0,1]$)

- Mémoire finie : ensemble fini d'états S
- Pour chaque état : définition d'un ensemble fini de transitions possibles, avec une loi de probabilités sur ces transitions ; chaque transition "écrit" un mot fini (qui peut être vide) sur $\{0,1\}$ (écriture en binaire)
- On espère que l'automate écrit (avec probabilité 1) un mot infini, qu'on interprète comme le développement binaire d'un réel aléatoire entre 0 et 1.
- Restrictions sur les probabilités et les mots écrits :
  - proba $1/2$, mots $\{0,1\}^*$ (modèle Knuth-Yao, "pile ou face")
  - probas rationnelles, écriture de lettres seulement

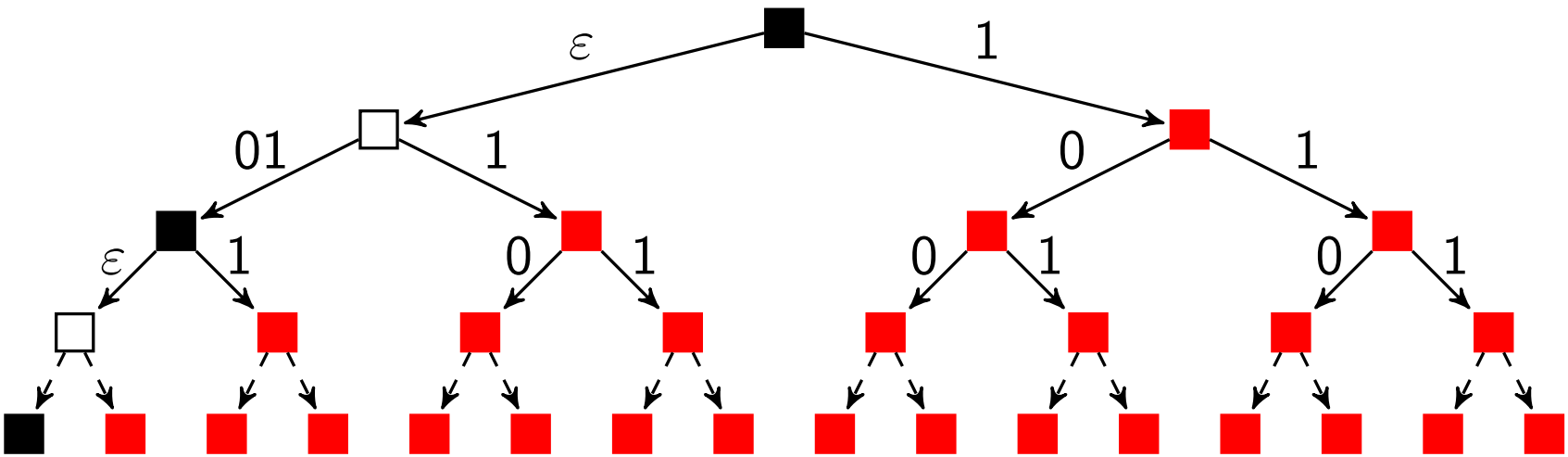
Le modèle (lois sur $[0,1]$)

- Mémoire finie : ensemble fini d'états S
- Pour chaque état : définition d'un ensemble fini de transitions possibles, avec une loi de probabilités sur ces transitions ; chaque transition "écrit" un mot fini (qui peut être vide) sur $\{0,1\}$ (écriture en binaire)
- On espère que l'automate écrit (avec probabilité 1) un mot infini, qu'on interprète comme le développement binaire d'un réel aléatoire entre 0 et 1.
- Restrictions sur les probabilités et les mots écrits :
  - proba $1/2$, mots $\{0,1\}^*$ (modèle Knuth-Yao, "pile ou face")
  - probas rationnelles, écriture de lettres seulement
  - probas rationnelles, mots $\{0,1\}^*$

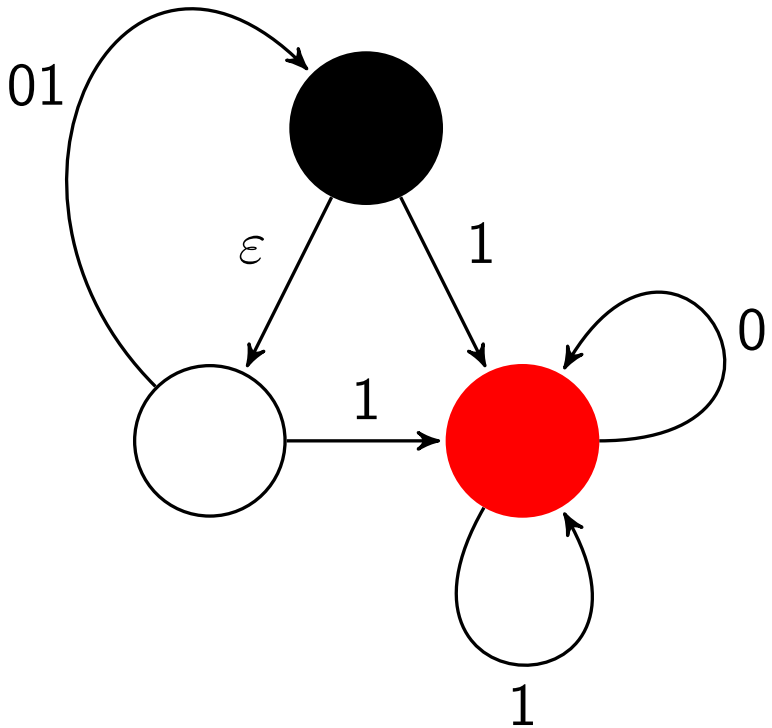
Le modèle (lois sur $[0, 1]$)

- Mémoire finie : ensemble fini d'états S
- Pour chaque état : définition d'un ensemble fini de transitions possibles, avec une loi de probabilités sur ces transitions ; chaque transition "écrit" un mot fini (qui peut être vide) sur $\{0, 1\}$ (écriture en binaire)
- On espère que l'automate écrit (avec probabilité 1) un mot infini, qu'on interprète comme le développement binaire d'un réel aléatoire entre 0 et 1.
- Restrictions sur les probabilités et les mots écrits :
  - proba $1/2$, mots $\{0, 1\}^*$ (modèle Knuth-Yao, "pile ou face")
  - probas rationnelles, écriture de lettres seulement
  - probas rationnelles, mots $\{0, 1\}^*$
  - **Lemme :** tous ces modèles sont équivalents

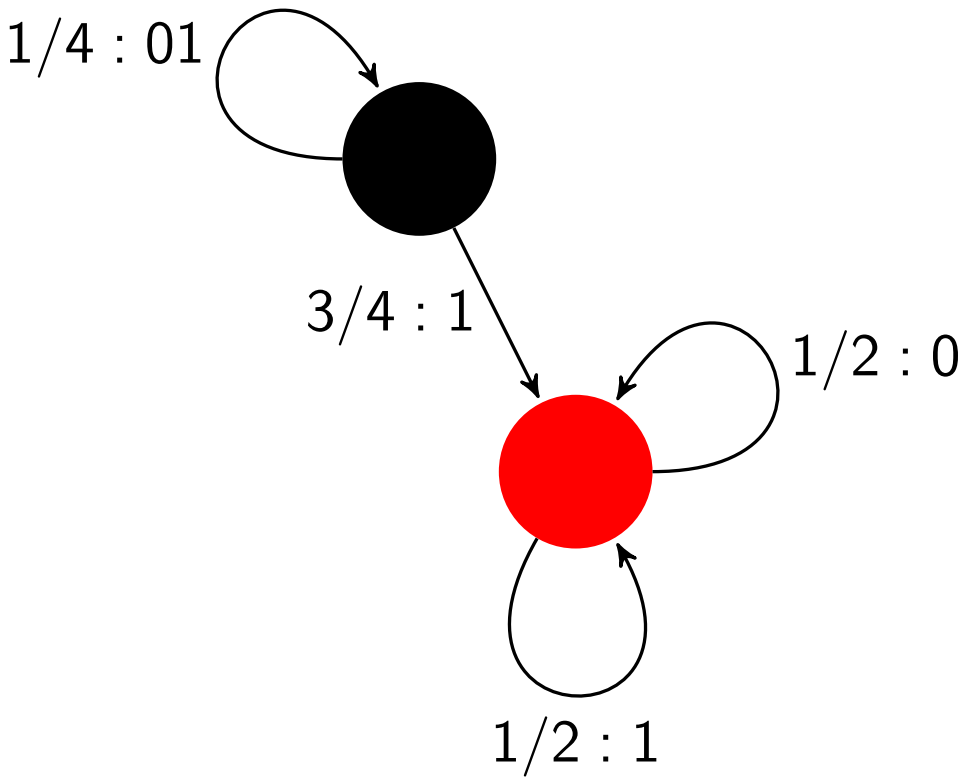
Un exemple d'arbre de Knuth et Yao



Le même exemple, en forme d'automate



Avec un état en moins



Remarques sur le modèle

- **Pour les probabilistes :** oui, c'est juste une chaîne de Markov à espace d'états fini (sauf qu'au lieu de regarder la trajectoire, chaque transition écrit un mot et on regarde la concaténation de ces mots).

Remarques sur le modèle

- **Pour les probabilistes :** oui, c'est juste une chaîne de Markov à espace d'états fini (sauf qu'au lieu de regarder la trajectoire, chaque transition écrit un mot et on regarde la concaténation de ces mots).
- **Pour les informaticiens :** ce n'est pas vraiment un automate, ni même un automate probabilisé : on ne lit pas de mot en entrée, on l'écrit.

Remarques sur le modèle

- **Pour les probabilistes :** oui, c'est juste une chaîne de Markov à espace d'états fini (sauf qu'au lieu de regarder la trajectoire, chaque transition écrit un mot et on regarde la concaténation de ces mots).
- **Pour les informaticiens :** ce n'est pas vraiment un automate, ni même un automate probabilisé : on ne lit pas de mot en entrée, on l'écrit.
- **Pour tout le monde :** une bonne partie de nos problèmes vient de ce qu'on ne regarde pas le mot produit comme un mot, mais comme un réel.

La loi uniforme (Lebesgue) sur $[0, 1]$

- Les chiffres sont indépendants, uniformes.

La loi uniforme (Lebesgue) sur $[0, 1]$

- Les chiffres sont indépendants, uniformes.
- Conséquence facile : avec probabilité 1, la densité asymptotique de 0 (ou de 1) est $1/2$.

$$\mathbf{P}\left(\lim_n \frac{1}{n}\left(\sum_{j<n} \mathbf{1}_{X_{j+1}=b}\right) = 1/2\right) = 1$$

La loi uniforme (Lebesgue) sur $[0, 1]$

- Les chiffres sont indépendants, uniformes.
- Conséquence facile : avec probabilité 1, la densité asymptotique de 0 (ou de 1) est $1/2$.

$$\mathbf{P}\left(\lim_n \frac{1}{n}\left(\sum_{j<n} \mathbf{1}_{X_{j+1}=b}\right) = 1/2\right) = 1$$

- À peine plus difficile : pour tout mot $w \in \{0,1\}^k$, la densité asymptotique de w est $1/2^k$.

$$\mathbf{P}\left(\lim_n \left(\frac{1}{n}\sum_{j<n} \mathbf{1}_{X_{j+1}\ldots X_{j+k}=b_1\ldots b_k}\right) = 1/2^k\right) = 1$$

La loi uniforme (Lebesgue) sur $[0, 1]$

- Les chiffres sont indépendants, uniformes.
- Conséquence facile : avec probabilité 1, la densité asymptotique de 0 (ou de 1) est $1/2$.

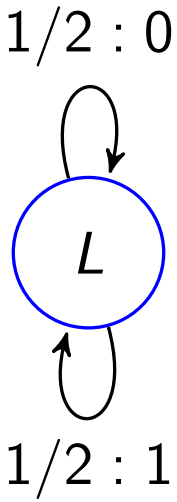
$$\mathbf{P}\left(\lim_n \frac{1}{n}\left(\sum_{j<n} \mathbf{1}_{X_{j+1}=b}\right) = 1/2\right) = 1$$

- À peine plus difficile : pour tout mot $w \in \{0,1\}^k$, la densité asymptotique de w est $1/2^k$.

$$\mathbf{P}\left(\lim_n \left(\frac{1}{n}\sum_{j<n} \mathbf{1}_{X_{j+1}\ldots X_{j+k}=b_1\ldots b_k}\right) = 1/2^k\right) = 1$$

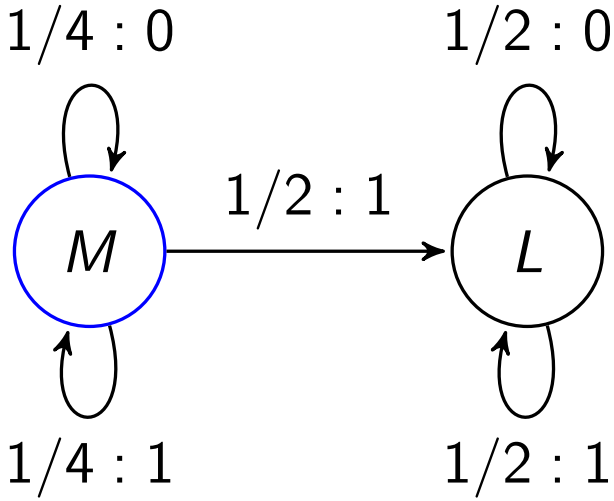
- (C'est vrai aussi bien si on regarde les occurrences en positions multiples de k, ou toutes les occurrences)

Loi uniforme sur $[0, 1]$

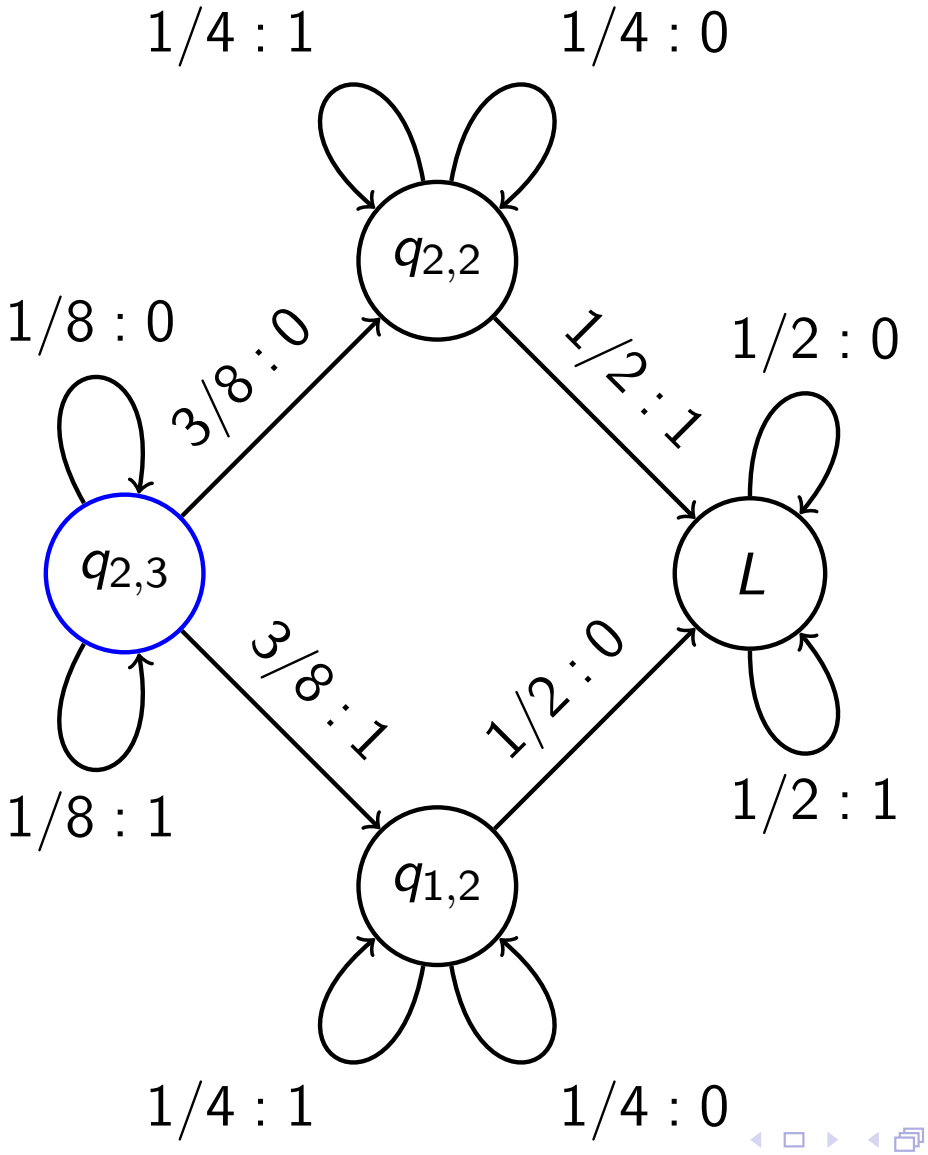


Densité 2x

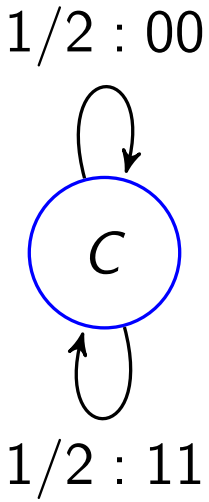
(Loi du maximum de 2 uniformes indépendantes)



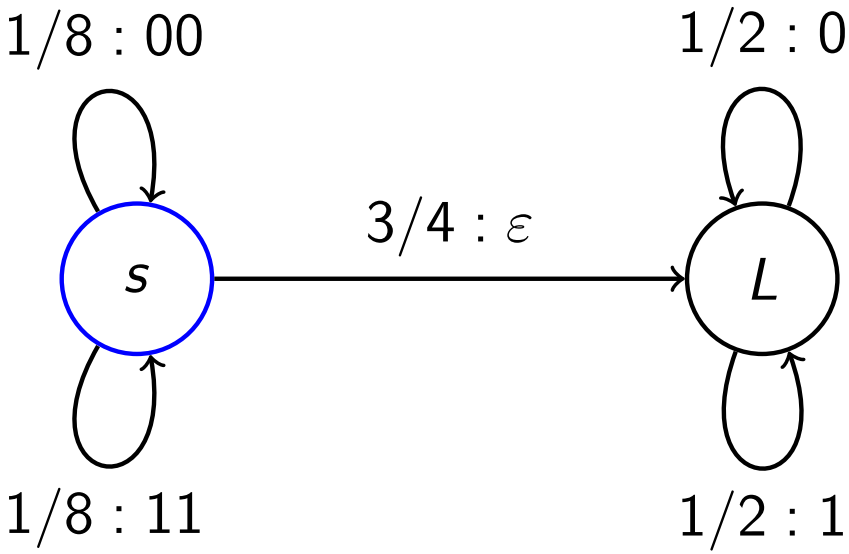
Densité $6x(1-x)$ (médiane de trois uniformes)



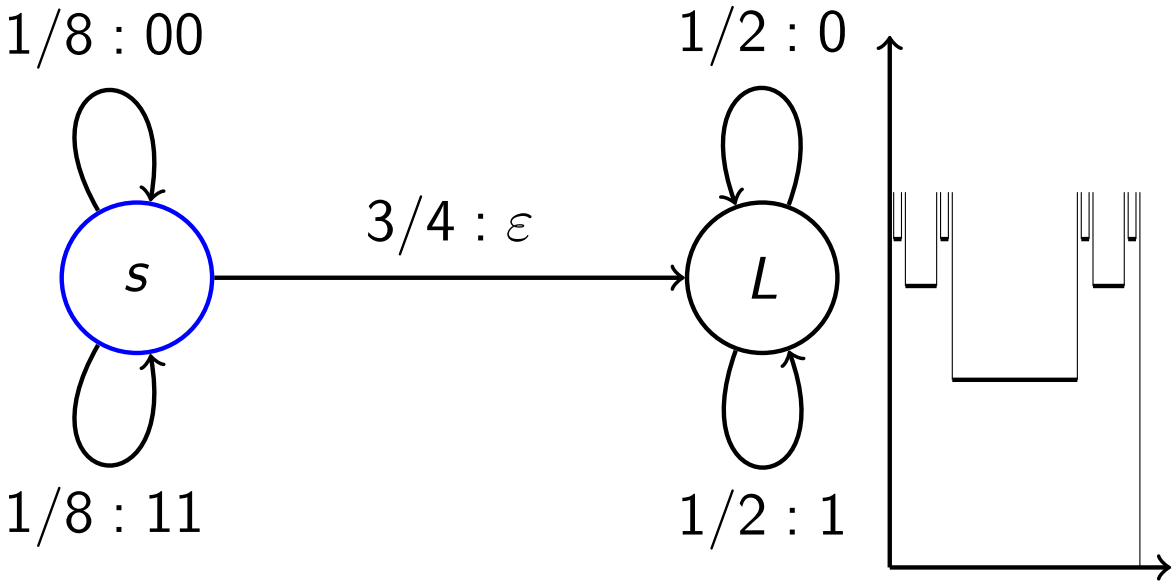
Automate bégayant



Des lois à densité “moches” (“Château de Cantor”)



Des lois à densité “moches” (“Château de Cantor”)



Classification des lois de probabilités sur $[0, 1]$

Le **théorème de représentation de Lebesgue** permet d'écrire toute loi de probabilités μ sur $[0, 1]$ comme somme de deux parties :

- une partie *absolument continue* (**à densité**) μ_c : $\mu_c([0, x]) = \int_0^x f(t)dt$
- une partie *étrangère à la mesure de Lebesgue* μ_e : mesure ayant un support dont la mesure de Lebesgue est 0

La partie étrangère peut à son tour être décomposée en

- une partie μ_a chargeant un ensemble A au plus dénombrable d'**atomes** : $\mu_a(B) = \sum_{a \in A \cap B} p(a)$
- une partie **singulière** μ_s sans atomes (fonction de répartition continue), mais qui ne charge qu'un ensemble de mesure de Lebesgue 0 (fonction dérivable presque partout, de dérivée nulle)

(On aimerait se concentrer sur la simulation de lois dont la partie singulière est nulle !)

Classification des automates

Théorème (D. 2017+)

La nature de la décomposition de la loi simulée par un automate dépend des composantes fortement connexes puits du graphe orienté des transitions :

- La partie atomique est non nulle **si et seulement si** il existe une composante puits dans laquelle, au départ de chaque état, un seul mot (infini, périodique) est émis.
- La partie absolument continue est non nulle **si et seulement si** il existe une composante puits dans laquelle, en partant d'un état aléatoire *distribué selon la loi stationnaire*, la loi simulée est **exactement** la mesure de Lebesgue.
- La partie singulière est non nulle **si et seulement si** il existe une composante puits ne satisfaisant aucune des conditions précédentes.

Ébauche de preuve (partie absolument continue)

- La séquence des états forme une chaîne de Markov à ensemble d'états fini ; sur une composante puits (récurrente), on a une unique distribution stationnaire sur les états $(p(s))_s$.

Ébauche de preuve (partie absolument continue)

- La séquence des états forme une chaîne de Markov à ensemble d'états fini ; sur une composante puits (récurrente), on a une unique distribution stationnaire sur les états $(p(s))_s$.
- Si la loi des mots produits au départ de l'état s est μ_s, la loi pour la chaîne stationnaire est $\mu = \sum_s p(s)\mu_s$.

Ébauche de preuve (partie absolument continue)

- La séquence des états forme une chaîne de Markov à ensemble d'états fini ; sur une composante puits (récurrente), on a une unique distribution stationnaire sur les états $(p(s))_s$.
- Si la loi des mots produits au départ de l'état s est μ_s, la loi pour la chaîne stationnaire est $\mu = \sum_s p(s)\mu_s$.
- Pour la chaîne stationnaire : pour tout mot $w \in \{0,1\}^k$, la densité asymptotique de w dans le mot infini émis est égale (avec proba. 1) à la probabilité que le mot émis commence par w.

Ébauche de preuve (partie absolument continue)

- La séquence des états forme une chaîne de Markov à ensemble d'états fini ; sur une composante puits (récurrente), on a une unique distribution stationnaire sur les états $(p(s))_s$.
- Si la loi des mots produits au départ de l'état s est μ_s, la loi pour la chaîne stationnaire est $\mu = \sum_s p(s)\mu_s$.
- Pour la chaîne stationnaire : pour tout mot $w \in \{0,1\}^k$, la densité asymptotique de w dans le mot infini émis est égale (avec proba. 1) à la probabilité que le mot émis commence par w.
- Si *chacune de ces probabilités vaut* $(1/2)^k$, alors le mot émis par la chaîne stationnaire suit la loi uniforme ($\mu = L$) ; sinon, μ est étrangère à la mesure de Lebesgue.

Ébauche de preuve (partie absolument continue)

- La séquence des états forme une chaîne de Markov à ensemble d'états fini ; sur une composante puits (récurrente), on a une unique distribution stationnaire sur les états $(p(s))_s$.
- Si la loi des mots produits au départ de l'état s est μ_s, la loi pour la chaîne stationnaire est $\mu = \sum_s p(s)\mu_s$.
- Pour la chaîne stationnaire : pour tout mot $w \in \{0,1\}^k$, la densité asymptotique de w dans le mot infini émis est égale (avec proba. 1) à la probabilité que le mot émis commence par w.
- Si *chacune de ces probabilités vaut* $(1/2)^k$, alors le mot émis par la chaîne stationnaire suit la loi uniforme ($\mu = L$) ; sinon, μ est étrangère à la mesure de Lebesgue.
- La mesure μ est une combinaison convexe des mesures μ_s : si μ est absolument continue, alors chaque μ_s l'est aussi (au final, si une des μ_s est absolument continue alors toutes le sont).

Commentaires

- La classification des lois simulées est effective (on sait tester algorithmiquement les deux conditions).

Commentaires

- La classification des lois simulées est effective (on sait tester algorithmiquement les deux conditions).
- *A priori*, les automates qui simulent des lois à densité sont “rares”.

Commentaires

- La classification des lois simulées est effective (on sait tester algorithmiquement les deux conditions).
- *A priori*, les automates qui simulent des lois à densité sont "rares".
- Conséquence de la preuve : pour un automate fortement connexe, s'il y a une densité, elle est forcément bornée sur $[0, 1]$.

Commentaires

- La classification des lois simulées est effective (on sait tester algorithmiquement les deux conditions).
- *A priori*, les automates qui simulent des lois à densité sont “rares”.
- Conséquence de la preuve : pour un automate fortement connexe, s’il y a une densité, elle est forcément bornée sur $[0, 1]$.
- En particulier, on peut exhiber des lois simulables, à densité, mais pas simulables par automates fortement connexes.

Loi régulière, mais pas analytique...

- Au départ, on aurait aimé généraliser le théorème de Knuth et Yao, en affaiblissant l'hypothèse d'analyticité de la densité.

Loi régulière, mais pas analytique...

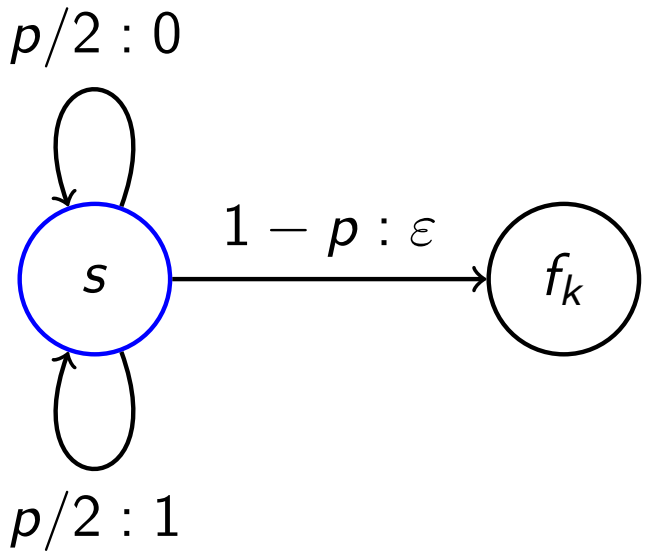
- Au départ, on aurait aimé généraliser le théorème de Knuth et Yao, en affaiblissant l'hypothèse d'analyticité de la densité.
- ...mais on sait fabriquer un automate qui simule une loi à densité k fois dérivable, mais non polynomiale !

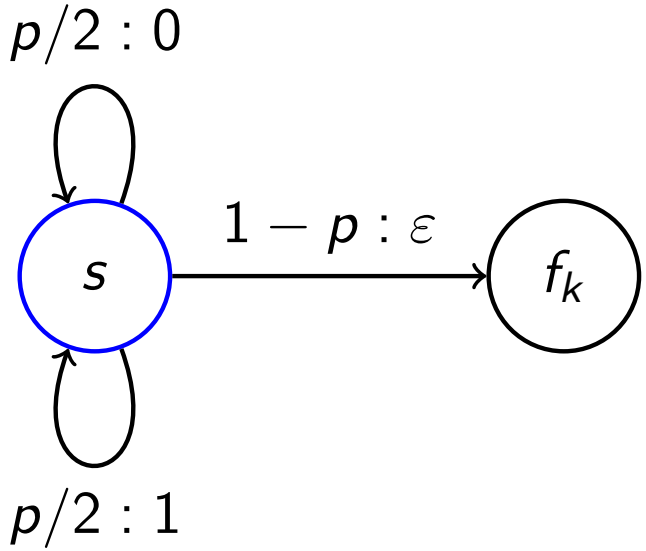
Loi régulière, mais pas analytique...

- Au départ, on aurait aimé généraliser le théorème de Knuth et Yao, en affaiblissant l'hypothèse d'analyticité de la densité.
- ...mais on sait fabriquer un automate qui simule une loi à densité k fois dérivable, mais non polynomiale !
- Pour k arbitraire : $f_k(x) = x^k(1-x)^k$ est $k-1$ fois dérivable, avec ses $k-1$ premières dérivées qui s'annulent en 0 et en 1.

Loi régulière, mais pas analytique...

- Au départ, on aurait aimé généraliser le théorème de Knuth et Yao, en affaiblissant l'hypothèse d'analyticité de la densité.
- ...mais on sait fabriquer un automate qui simule une loi à densité k fois dérivable, mais non polynomiale !
- Pour k arbitraire : $f_k(x) = x^k(1-x)^k$ est $k-1$ fois dérivable, avec ses $k-1$ premières dérivées qui s'annulent en 0 et en 1.
- Donc $f_{k,i}(x) = f(\{2^i x\})$ est $k-1$ fois dérivable sur $[0,1]$, mais pas $k+1$ fois.


$p/2:0$
$1-p:\varepsilon$
s
f_k
$p/2:1$



Densité sur $[0, 1]$ **:** $C.\sum_{i\geq 0}(p/2)^i f_k(\{2^i x\})$; si p est assez proche de 0, la série converge vers une fonction $k-1$ fois dérivable sur $[0, 1]$

Lois à densité et automates fortement connexes

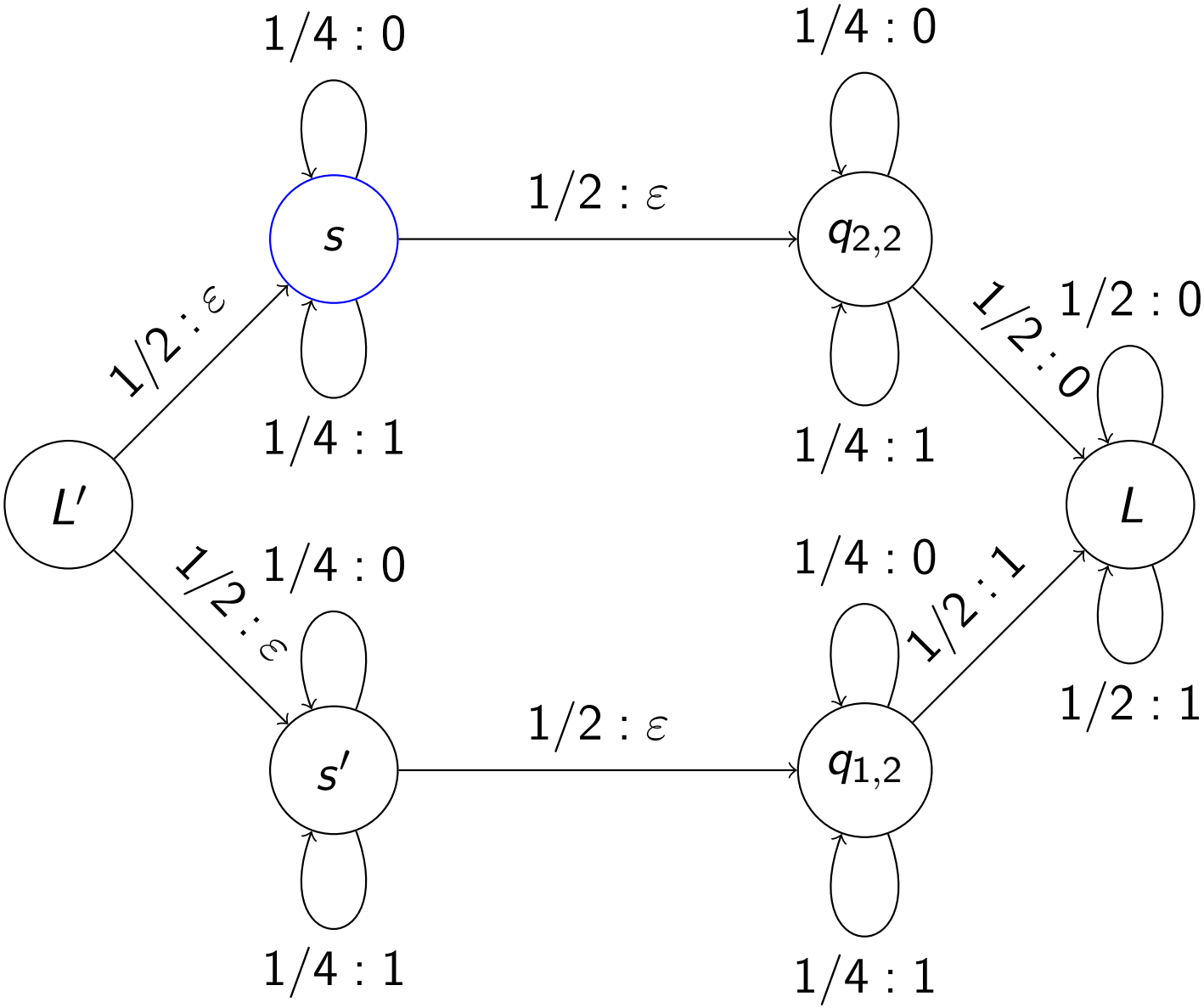
- D'après le théorème de classification, les automates qui simulent une loi à densité "cachent" la mesure de Lebesgue dans leurs composantes fortement connexes puits.

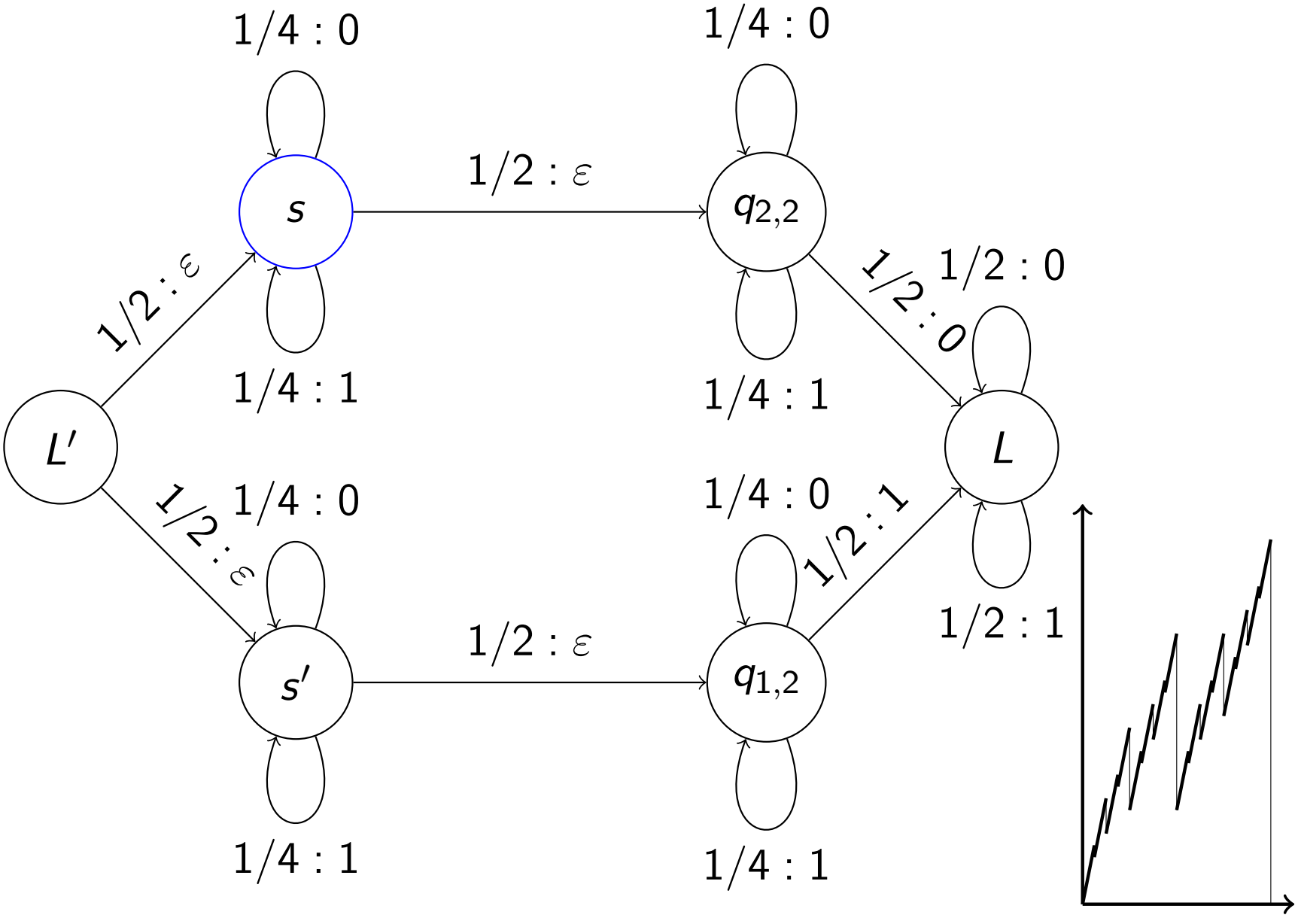
Lois à densité et automates fortement connexes

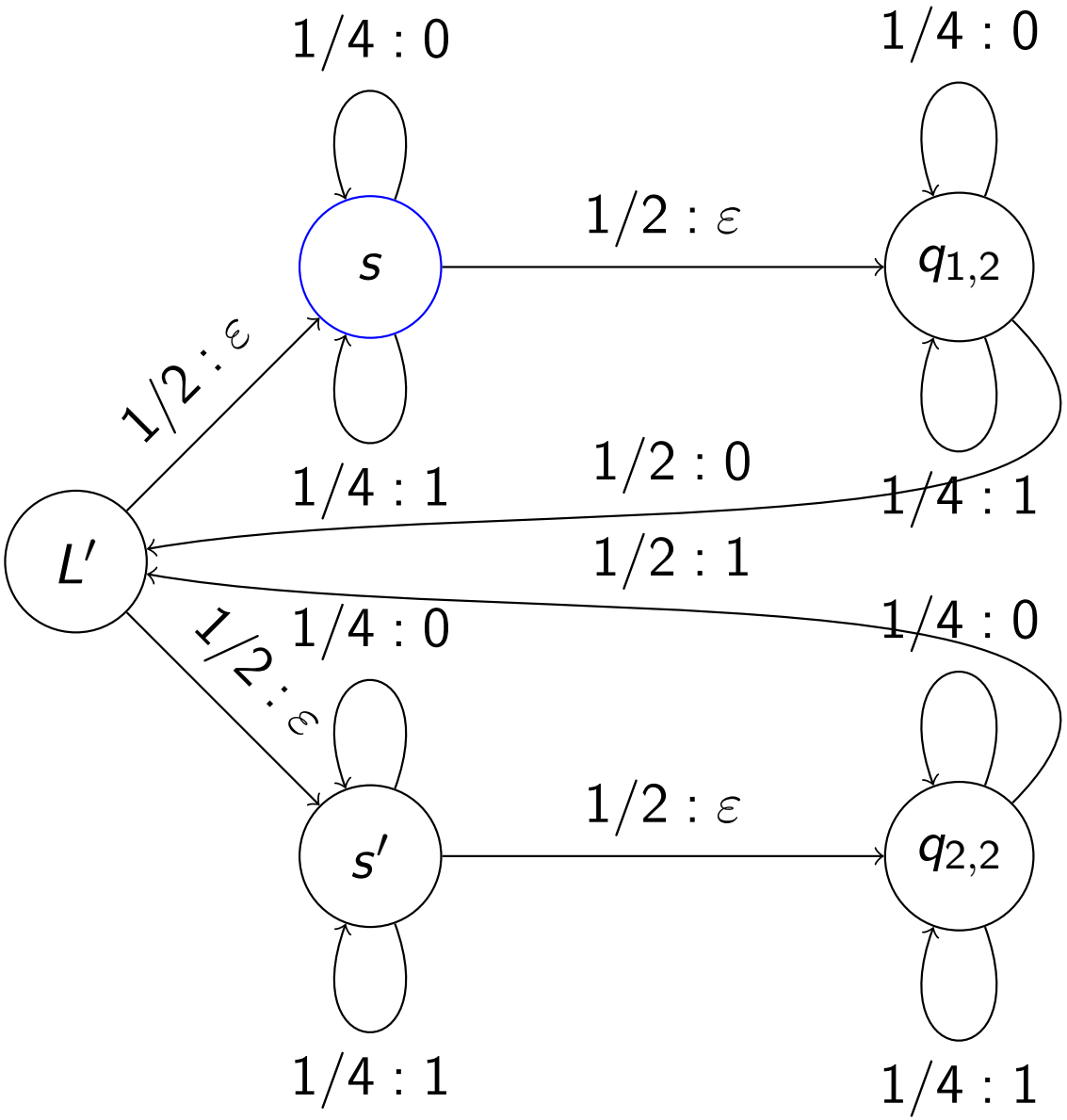
- D'après le théorème de classification, les automates qui simulent une loi à densité "cachent" la mesure de Lebesgue dans leurs composantes fortement connexes puits.
- On pourrait espérer que les lois à densité simulées par des automates fortement connexes soient "simples"

Lois à densité et automates fortement connexes

- D'après le théorème de classification, les automates qui simulent une loi à densité "cachent" la mesure de Lebesgue dans leurs composantes fortement connexes puits.
- On pourrait espérer que les lois à densité simulées par des automates fortement connexes soient "simples"
- Il n'en est rien : on peut avoir une densité qui admet un ensemble dense de discontinuités...


1/4 : 0
1/4 : 0
1/2 : ε
s
q2,2
1/2 : ε
1/2 : 0
1/2 : 0
1/4 : 1
1/4 : 1
L′
L
1/2 : ε
1/4 : 0
1/4 : 0
1/2 : 1
1/2 : ε
s′
q1,2
1/4 : 1
1/4 : 1


1/4 : 0
1/4 : 0
1/2 : ε
s
q2,2
1/2 : ε
1/2 : 0
1/2 : 0
1/4 : 1
1/4 : 1
L′
L
1/2 : ε
1/4 : 0
1/4 : 0
1/2 : 1
1/2 : ε
s′
q1,2
1/2 : 1
1/4 : 1
1/4 : 1


1/4 : 0
1/4 : 0
1/2 : ε
s
$q_{1,2}$
1/2 : ε
1/4 : 1
1/2 : 0
1/4 : 1
L'
1/2 : 1
1/4 : 0
1/4 : 0
1/2 : ε
1/2 : ε
s'
$q_{2,2}$
1/4 : 1
1/4 : 1

Et maintenant ?

- Pas de caractérisation des lois simulables, même en se limitant aux lois à densité

Et maintenant ?

- Pas de caractérisation des lois simulables, même en se limitant aux lois à densité
- Pas trop d'espoir de relâcher significativement l'hypothèse d'analyticité de Knuth et Yao (densité C^∞ ?)

Et maintenant ?

- Pas de caractérisation des lois simulables, même en se limitant aux lois à densité
- Pas trop d'espoir de relâcher significativement l'hypothèse d'analyticité de Knuth et Yao (densité C^∞ ?)
- Parmi les exemples, les lois "gentilles" sont simulables dans n'importe quelle base, pas seulement en binaire ; les cas "pathologiques" ne le sont probablement qu'en base 2^k (possibilité de théorème "à la Cobham" ? caractériser les lois simulables à la fois en binaire et en ternaire ?)

Et maintenant ?

- Pas de caractérisation des lois simulables, même en se limitant aux lois à densité
- Pas trop d'espoir de relâcher significativement l'hypothèse d'analyticité de Knuth et Yao (densité C^∞ ?)
- Parmi les exemples, les lois "gentilles" sont simulables dans n'importe quelle base, pas seulement en binaire ; les cas "pathologiques" ne le sont probablement qu'en base 2^k (possibilité de théorème "à la Cobham" ? caractériser les lois simulables à la fois en binaire et en ternaire ?)
- Version géométrique en dimension 2 : on peut au moins simuler la loi uniforme à l'intérieur d'un polygone à sommets rationnels (mais pas sur le bord) ; au-delà ?

That's all folks (rendez-vous à Aléa 2018)

That's all folks (rendez-vous à Aléa 2018)

sauf si vous avez des questions