Boltzmann Sampling and Simulation

Michèle Soria
LIP6 – UPMC

CONFERENCE IN MEMORY OF PHILIPPE FLAJOLET

Random Sampling Combinatorial structures

Combinatorial modeling → Sampling of random objects
Get information on the model by working on the random objects
- visualize and compute order of growth for quantities;
- distinguish significant properties (random vs observed);
- compare models, test software

Analytic Combinatorics → Generic Methods

- **Recursive Method**: $n \rightarrow \gamma_n$
  \[ \mathbb{P}(\gamma_n) = \frac{1}{c_n} \]
  compute enumeration sequences
- **Boltzmann Method**: $x \rightarrow \gamma$
  \[ \mathbb{P}(\gamma) \propto x^{\vert\gamma\vert} \]
  evaluate generating functions at $x$
Boltzmann Sampling (DuFlLoSc04)

Boltzmann model

\[ P_x(\gamma) = \frac{x^{\|\gamma\|} C(x)}{C} \]

uniform sampling

\[ C = \Phi(A, B) \]

\[ \Gamma C = F(\Gamma A, \Gamma B) \]
Boltzmann samplers for the random generation of combinatorial structures – DuFlLoSc04 – cited by 124

Quadratic exact-size and linear approximate-size random generation of planar graphs E Fusy - 2005 -

**Boltzmann sampling of unlabelled structures** P Flajolet, E Fusy, C Pivoteau - 2007 -

An unbiased pointing operator for unlabeled structures, with ... M Bodirsky, E Fusy, M Kang, S Vigerske - 2007 -
Degree distribution of random Apollonian network structures and Boltzmann sampling A Darrasse, M Soria - 2007 -
Properties of random graphs via boltzmann samplers K Panagiotou, A.Weiess - 2007 -
Random Graphs with Structural Constraints A Steger - 2007 -
Random generation of finitely generated subgroups of a free group F Bassino, C Nicaud, P Weil - 2008 -
Random generation of possibly incomplete deterministic automata. F Bassino, J David, C Nicaud - 2008 -
Boltzmann oracle for combinatorial systems C Pivoteau, B Salvy, M Soria - 2008 -
Random XML sampling the Boltzmann way A Darrasse -2008 -
The degree sequence of random graphs from subcritical classes N Bernasconi, K Panagiotou, - 2009 -
Boltzmann samplers for colored combinatorial objects O Bodini, A. Jacquot - 2009 -
Boltzmann Samplers for v-balanced Colored Necklaces O Bodini, A Jaquot - 2009 -
Fast and sound random generation for automated testing and benchmarking in objective Caml B Canou, A Darrasse - 2009
Uniform random sampling of planar graphs in linear time E Fusy - 2009 -
Profiles of permutations M Lugo - 2009 -
Uniform random generation of huge metamodel instances A Mougenot, A Darrasse, X Blanc, M Soria - 2009 -
Vertices of degree k in random unlabeled trees K Panagiotou, M Sinha - 2009 -
Boltzmann sampling of ordered structures O Roussel, M. Soria - 2009 -
Autour de la génération aléatoire sous modèle de Boltzmann O Bodini - 2010
Random sampling of plane partitions O Bodini, É Fusy, C Pivoteau - 2010 -
Multi-dimensional Boltzmann sampling of languages O Bodini, Y Ponty - 2010 -
Boltzmann Samplers, Pólya Theory, and Cycle Pointing M Bodirsky, E Fusy, M Kang, S Vigerske - 2010 -
Boltzmann generation for regular languages with shuffle A Darrasse, K Panagiotou, O Roussel, M Soria - 2010 -
Quadratic and linear time random generation of planar graphs E Fusy - 2010
Parametric Random Generation of Deterministic Tree Automata PC Héam, C Nicaud, S Schmitz - 2010-
Profiles of large combinatorial structures MT Lugo - 2010 -
Average Analysis of Glushkov Automata under a BST-Like Model C Nicaud, C Pivoteau, B Razet, - 2010 -
Boys-and-girls birthdays and Hadamard products O Bodini, D Gardy, O Roussel - 2011 -
Lambda terms of bounded unary height O Bodini, D Gardy, B Gittenberger - 2011 -
Boltzmann samplers for first-order differential specifications O Bodini, O Roussel, M Soria - 2011 -
Analyse d’algorithmes et génération aléatoire en théorie des langages C Nicaud - 2011 -
Enumeration and random generation of pin permutations F Bassino, M Bouvel, A Pierrot, C Pivoteau, D Rossin, 2012
Dirichlet Random Samplers for Multiplicative Structures O Bodini, J Lumbroso - 2012 -

...
"If you can specify it and compute values of GF, then you can sample it with Boltzmann method"

\[ \text{Comb. Spec. } \mathcal{C} = \Phi(A, B, \ldots) \]
\[ x < \rho ; \text{ oracle for } C(x), A(x), B(x) \]

Random Objet \( \gamma \in \mathcal{C} \);
\[ \mathbb{P}_x(\gamma) = \frac{x^{|\gamma|}}{C(x)} \]

- **Genericity**
  Combinatorial Specifications \( \rightarrow \) Equations on GF \( \rightarrow \) Sampler

- **Effectivity**
  Sampler compiled from specifications,
  Simple probabilistic algorithms, linear time

\[ A + B : \text{ flip a biased coin and sample from } A \text{ or } B \]
\[ A \times B : \text{ independently sample } \text{ from } A \text{ and } B \]
Binary trees:
\[ \mathcal{B} = \mathcal{Z} + \mathcal{B} \times \mathcal{B} \]
\[ B(z) = z + B(z)^2 = \frac{1 - \sqrt{1 - 4z}}{2} \]

Algorithm: \( \Gamma \mathcal{B}(x) \)

if Bernoulli \( \frac{x}{B(x)} \) then
    Return ■
else
    Return \( \langle \Gamma \mathcal{B}(x), \Gamma \mathcal{B}(x) \rangle \);
end if

recursive process; stops with proba 1 for \( x < \rho \)

Arbres planaires:
\[ \mathcal{T} = \mathcal{Z} \times \text{SEQ}(\mathcal{T}) \]
\[ T(z) = \frac{z}{1 - T(z)} = \frac{1 - \sqrt{1 - 4z}}{2} \]

Algorithm: \( \Gamma \mathcal{T}(x) \)

\[ k \leftarrow \text{Geom}(T(x)) ; \]
\[ \text{forest} \leftarrow \langle \Gamma \mathcal{T}(x), \ldots, \Gamma \mathcal{T}(x) \rangle_{\text{etiq}} ; \]
\[ \text{Return } \square \times \text{forest} ; \]
**SPECIFICATION OF \( C \)  \( \leadsto \)  **FUNCTIONAL EQUATION  \( C(z) \)**

Combinatorial Class \( C \), labelled objects

Exponential Generating Function (EGF)

\[
C(z) = \sum_{\gamma \in C} \frac{z^{\, |\gamma|}}{|\gamma|!} = \sum_{n \geq 0} \frac{c_n z^n}{n!}
\]

<table>
<thead>
<tr>
<th>Construction</th>
<th>Notation</th>
<th>EGF</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \varepsilon )</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>atome</td>
<td>( \mathcal{Z} )</td>
<td>( z )</td>
</tr>
<tr>
<td>Union</td>
<td>( C = A \cup B )</td>
<td>( C(z) = A(z) + B(z) )</td>
</tr>
<tr>
<td>Produit</td>
<td>( C = A \times B )</td>
<td>( C(z) = A(z) \times B(z) )</td>
</tr>
<tr>
<td>Séquence</td>
<td>( C = \text{SEQ}(A) )</td>
<td>( C(z) = (1 - A(z))^{-1} )</td>
</tr>
<tr>
<td>Ensemble</td>
<td>( C = \text{SET}(A) )</td>
<td>( C(z) = \exp(A(z)) )</td>
</tr>
<tr>
<td>Cycle</td>
<td>( C = \text{CYC}(A) )</td>
<td>( C(x) = -\log(1 - A(z)) )</td>
</tr>
</tbody>
</table>
**Specification of $\mathcal{C}$ $\leadsto$ Sampling Algorithm $\Gamma \mathcal{C}(x)$**

Combinatorial Class $\mathcal{C}$, labelled objects

Exponential Generating Function (EGF)

$$C(z) = \sum_{\gamma \in \mathcal{C}} \frac{z^{\mid \gamma \mid}}{\mid \gamma \mid !} = \sum_{n \geq 0} \frac{c_n z^n}{n!}$$

$$P_x(\gamma) = \frac{x^{\mid \gamma \mid}}{\mid \gamma \mid ! C(x)}$$

<table>
<thead>
<tr>
<th>Construction</th>
<th>Sampling Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{C} = \emptyset \text{ or } \emptyset$</td>
<td>$\Gamma \mathcal{C}(x) := \varepsilon \text{ or atom}$</td>
</tr>
<tr>
<td>$\mathcal{C} = \mathcal{A} \cup \mathcal{B}$</td>
<td>$\Gamma \mathcal{C}(x) := \text{Bernoulli} \left( \frac{A(x)}{A(x) + B(x)} \right)$; return $\Gamma \mathcal{A}(x) \text{ or } \Gamma \mathcal{B}(x)$</td>
</tr>
<tr>
<td>$\mathcal{C} = \mathcal{A} \times \mathcal{B}$</td>
<td>$\Gamma \mathcal{C}(x) := \text{return } \langle \Gamma \mathcal{A}(x); \Gamma \mathcal{B}(x) \rangle$ independent calls</td>
</tr>
<tr>
<td>$\mathcal{C} = \text{SEQ}(\mathcal{A})$</td>
<td>$\Gamma \mathcal{C}(x) := k \leftarrow \text{Geometric}(A(x)); \text{return } k \text{ independent } \Gamma \mathcal{A}(x)$</td>
</tr>
<tr>
<td>$\mathcal{C} = \text{SET}(\mathcal{A})$</td>
<td>$\Gamma \mathcal{C}(x) := k \leftarrow \text{Poisson}(A(x)); \text{return } k \text{ independent } \Gamma \mathcal{A}(x)$</td>
</tr>
<tr>
<td>$\mathcal{C} = \text{CYCLE}(\mathcal{A})$</td>
<td>$\Gamma \mathcal{C}(x) := k \leftarrow \text{Logarithmic}(A(x)); \text{return } k \text{ independent } \Gamma \mathcal{A}(x)$</td>
</tr>
</tbody>
</table>
"If you can specify it and compute values of GF, then you can sample it with Boltzmann method"

- Parameter $x$
- Computation of numerical values of generating functions: *oracle*
- Probabilistic Algorithms: discrete laws with parameters $A(x)$
Theorem (DuFlLoSc04)

The Boltzmann sampler $\Gamma C(x)$ for a class $C$, of labelled objects, recursively specified on $\varepsilon$, $\mathbb{Z}$, $\cup$, $\times$, Seq, Set, Cycle, has arithmetic complexity which is linear in the size of the result (under hypothesis oracle in $O(1)$).

(Oracle : Pivoteau-Salvy-Soria 2008, 2011)

Uniform random number in $[0,1]$ + real numbers manipulation
Theorem (FlFuPi07)

For a class $\mathcal{C}$ of non labelled objects, recursively specified on $\varepsilon$, $\mathbb{Z}$, $\cup$, $\times$, $\text{Seq}$, $\text{Set}$, $\text{Cycle}$, the Boltzmann sampler has arithmetic complexity which is linear in the size of the result (under hypothesis oracle in $O(1)$).
Size distribution of random objects $x \rightarrow \gamma \in C$

- $\mathbb{P}_x(\gamma) = \frac{x|\gamma|}{|\gamma|!C(x)}$ uniformity for all objects of a given size

- Size $N_x$ of the output is a random variable $\mathbb{P}(N_x = n) = \frac{c_n x^n}{n!C(x)}$

- Tuning $x$ allows for aiming a certain size $\mathbb{E}(N_x) = x \frac{C'(x)}{C(x)}$

- Size distribution depends on $x_{(0 < x < \rho)}$: continuum of distr. funct.
- Shape of distribution depends on the singularity type of $C(x)$

\[ C(x) = e^{(e^x - 1)} \]

\[ C(x) = \frac{1}{2 - e^x} \]

\[ C(x) = \frac{1 - \sqrt{1 - 4x}}{2} \]
Sampling with Size Control

- accept sample objects of approximate size $[n(1 - \varepsilon), n(1 + \varepsilon)]$
- rejection: if size is not in the accepted interval, then reject and sample again

**Theorem (DuFlLoSc04, FlFuPi07)**

For a class $\mathcal{C}$ of (non) labelled objects, recursively specified on $\varepsilon$, $\mathbb{Z}$, $\cup$, $\times$, Seq, Set, Cycle, the Boltzmann sampler with reject $\Gamma C(x, n, \varepsilon)$ has average arithmetic complexity in $O(n)$. 

1 trial on average

$O(1)$ trial on average

$O(n)$ trial on average
Current issues in Boltzmann sampling

- Extend the model
  - to other operators
  - to other universes

- Non uniform generation
  - multivariate, multicriteria sampling
  - sampling according to other distributions
  - biased sampling for realistic modeling

- Certification, implementation
  - exact simulation
  - discrete model
  - automatization
Discrete Boltzmann Samplers

- Arithmetic Boltzmann Samplers: evaluate GFs → approximations ... Boltzmann distribution?
- Boltzmann Samplers based on binary coin flipping?

Comb. Spec. $C = \Phi(A, B, \ldots)$

$x < \rho$; oracle for $C(x), A(x), B(x)$

Random Objet $\gamma \in C$;

$$P_x(\gamma) = \frac{x^{\mid\gamma\mid}}{C(x)}$$

Source of Random Bits

- Genericity et Effectivity
- Average boolean complexity en $O(|\gamma|)$
- Perfect Simulation of discrete random variables
  - $X$: Geometric, Poisson, Logarithmic
  - $P(X = k)$ needs $O(k)$ bits on average
Perfect Simulation of random variables

- Simulating RV : efficient algorithms operating over reals and using real source of randomness – *Luc Devroye’s treatise*
- Discrete source of randomness and finite computations?
  - **Knuth-Yao** (1976)
    random \([0,1]\) : \(U_i = \sum_{j>0} b_j 2^{-j}, \ b_j \in \{0,1\}\)
    Lazy evaluation for the generation of \((U_1, \ldots, U_N)\)
  - **Von Neumann** (1949), \(e^{-x} = \sum_{n \text{ odd}} \frac{x^{n-1}}{n-1!} - \frac{x^n}{n!}\)
    \(x = U_1 > \ldots > U_n < U_{n+1}\)
    **if** odd(n) **return** x **else** do it again
  - **Flajolet-Saheb** (1983-86) : Path length analysis of a trie

---

**Theorem (FlSa83)**

*The average number of elementary coin flips for producing \(k\) bits of the exponential variable is \(k + \gamma + o(1)\), and the g.f. of the distributions has a closed form expression.*
Exact simulation of discrete laws for Boltzmann sampling

**Algorithm : Bernoulli(\(p\))**

\[ i \leftarrow 1 + \text{Geom}(1/2); \]
Return \(B_i(p)\);

**Algorithm : Geometric(\(p\))**

\[ k \leftarrow 0; \]
do  
\[ i \leftarrow 1 + \text{Geom}(1/2); \]
if \(B_i(p) = 1\) then Return \(k\);  
else \(k \leftarrow k + 1;\)
fi ;od ;

\(\text{Bit}_i(p)\) : access with proba \(\frac{1}{2^{(i+1)}}\)

O(k+1) bits for Geom(p)= k

\(\text{Boltzmann Sampling and Simulation}\)

\[ P(VN(x, \mathcal{P}) = n) = \frac{1}{P(x)} \frac{P_n x^n}{n!} \text{ Boltz} \]

- \(U_1 < \ldots < U_n\)
  \(P_n = 1 \rightarrow P(x) = e^x \implies \text{Poisson}\)
- \(U_1 < \{U_2, \ldots, U_n\}\)
  \(P_n = (n - 1)! \rightarrow P(x) = \log \frac{1}{1-x} \implies \text{Logarithmic}\)
Buffon Machines (Flajolet-Pelletier-Soria 2011)

- Boolean Complexity Von Neuman Schema

**Theorem (FlPeSo11)**

Let \( \mathcal{P} \) class of permutations and \( x \) parameter, von Neumann Schema exactly simulates discrete v.a. \( N \), s.t. 
\[
\mathbb{P}(N = n) = \frac{1}{\mathbb{P}(\lambda)} \frac{P_n \lambda^n}{n!},
\]
with average number of flips \( O(1) \) and exponential queues.

- perfect simulation of Bernouilli laws with various parameters: \( \pi \), \( \exp(-1) \), \( \log 2 \), \( \sqrt{3} \), \( \cos(\frac{1}{4}) \), \( \zeta(5) \)

\[
C = \frac{7}{8} \zeta(3) - \frac{\pi^2}{12} \log 2 + \frac{1}{6} (\log 2)^3 = 0.53721 13193 \ldots
\]

consumes on average less than 6 coin flips, and at most 20 coin flips in 95% of the simulations.

- Work goes on: extensions to other laws, integration in Boltzmann implementations Jérémie Lumbroso, Maryse Pelletier, MS, ...