Linear-time exact sampling of sum-constrained random variables

Andrea Sportiello and Frédérique Bassino
CNRS and LIPN, Université Paris Nord, Villetaneuse

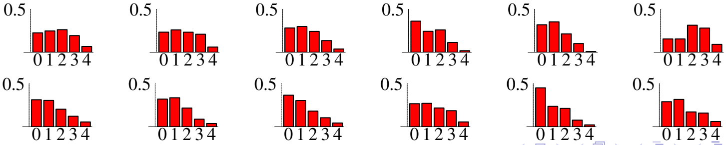
GASCom 2018, Athens, Greece
June 19th 2018

The problem

$$\Omega_n = \{\boldsymbol{x}\} \quad \boldsymbol{x} = (x_1, \ldots, x_n), \quad |\boldsymbol{x}| := \sum_i x_i, \quad x_i \in \mathbb{N}$$

$$\mu_{n,m}(\boldsymbol{x}) = \frac{1}{Z} \prod_{i=1}^{n} f_{\ i}\ (x_i) \times \delta_{|\boldsymbol{x}|,m}$$

$f_i(x)$



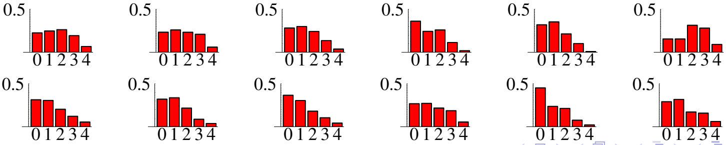
The problem

$$\Omega_n = \{\boldsymbol{x}\} \quad \boldsymbol{x} = (x_1, \dots, x_n), \quad |\boldsymbol{x}| := \sum_i x_i, \quad x_i \in \mathbb{N}$$

$$\mu_{n,m}(\boldsymbol{x}) = \frac{1}{Z} \prod_{i=1}^{n} f_i\,(x_i) \times \delta_{|\boldsymbol{x}|,m}$$

Problem: Assume that sampling from each distrib. f_i costs $\mathcal{O}(1)$. Find an algorithm that samples from the distribution $\mu_{n,m}$ in average linear time.

$f_i(x)$



The problem

$$\Omega_n = \{\boldsymbol{x}\} \quad \boldsymbol{x} = (x_1, \dots, x_n), \quad |\boldsymbol{x}| := \textstyle\sum_i x_i, \quad x_i \in \mathbb{N} \quad \longleftarrow$$

random vector
of integers

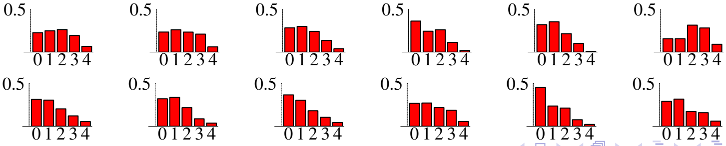
$$\mu_{n,m}(\boldsymbol{x}) = \tfrac{1}{Z} \textstyle\prod_{i=1}^{n} f_{\ i}\ (x_i) \times \delta_{|\boldsymbol{x}|,m} \quad \longleftarrow$$

NOT completely independent
(a single linear constraint)

$\uparrow$ variables are NOT
identically distributed

Problem: Assume that sampling from each distrib. f_i costs $\mathcal{O}(1)$.
Find an algorithm that samples from the distribution $\mu_{n,m}$
in average linear time.

$f_i(x)$



The problem

$$\Omega_n = \{\mathbf{x}\} \quad \mathbf{x} = (x_1, \dots, x_n), \quad |\mathbf{x}| := \sum_i x_i, \quad x_i \in \mathbb{N}$$

$\longleftarrow$ random vector of integers

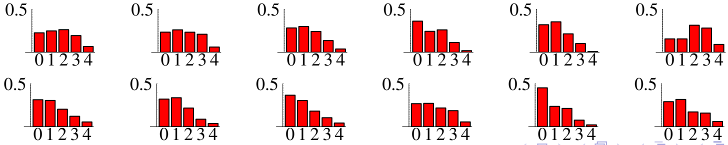
$$\mu_{n,m}(\mathbf{x}) = \frac{1}{Z} \prod_{i=1}^{n} f_i(x_i) \times \cancel{\delta_{|\mathbf{x}|,m}}$$

$\longleftarrow$ completely independent: the problem trivialises!

$\uparrow$ variables are NOT identically distributed

<u>Problem:</u> Assume that sampling from each distrib. f_i costs $\mathcal{O}(1)$. Find an algorithm that samples from the distribution $\mu_{n,m}$ in average linear time.

$f_i(x)$



The problem

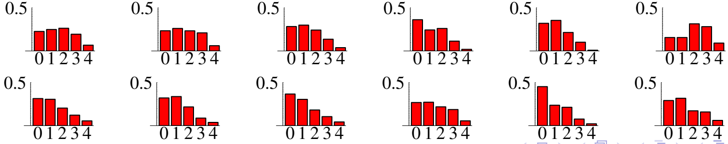
$$\Omega_n = \{\boldsymbol{x}\} \quad \boldsymbol{x} = (x_1, \dots, x_n), \quad |\boldsymbol{x}| := \textstyle\sum_i x_i, \quad x_i \in \mathbb{N} \quad \longleftarrow$$ random vector of integers

$$\mu_{n,m}(\boldsymbol{x}) = \tfrac{1}{Z} \textstyle\prod_{i=1}^{n} f_{\cancel{i}}(x_i) \times \delta_{|\boldsymbol{x}|,m} \quad \longleftarrow$$ NOT completely independent (a single linear constraint)

variables are identically distributed:
doable by using permutation symmetry [L. Devroye, 2012]

Problem: Assume that sampling from each distrib. f_i costs $\mathcal{O}(1)$. Find an algorithm that samples from the distribution $\mu_{n,m}$ in average linear time.

$f_i(x)$



The problem

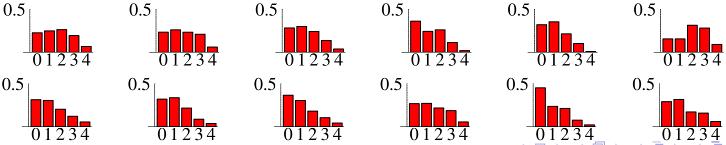
$$\Omega_n = \{\boldsymbol{x}\} \quad \boldsymbol{x} = (x_1, \ldots, x_n), \quad |\boldsymbol{x}| := \sum_i x_i, \quad x_i \in \mathbb{N} \quad \longleftarrow$$ random vector of integers

$$\mu_{n,m}(\boldsymbol{x}) = \frac{1}{Z} \prod_{i=1}^{n} f_{\;i}\;(x_i) \times \delta_{|\boldsymbol{x}|,m} \quad \longleftarrow$$ NOT completely independent (a single linear constraint)

$\uparrow$ variables are NOT identically distributed

Problem: Assume that sampling from each distrib. f_i costs $\mathcal{O}(1)$. Find an algorithm that samples from the distribution $\mu_{n,m}$ in average linear time.

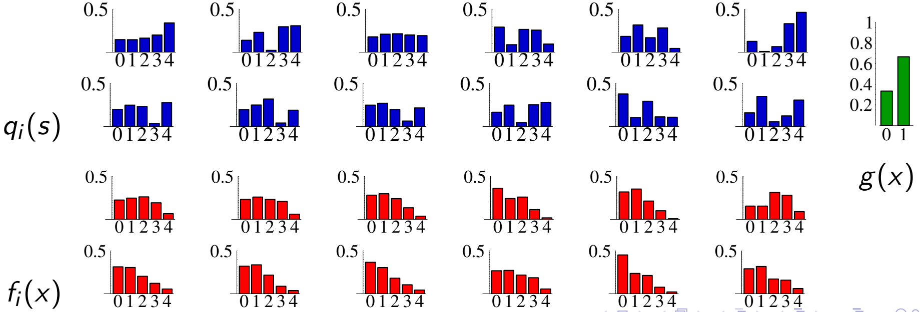
$f_i(x)$



Our solution

$$\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{N}^n, \qquad \mu_{n,m}(\boldsymbol{x}) = \frac{1}{Z} \prod_{i=1}^{n} f_i(x_i) \times \delta_{|\boldsymbol{x}|,m}$$

Our solution: **positive decomposition.** Assume that there exists $g(x) \in \{\mathrm{Bern}_b, \mathrm{Poiss}, \mathrm{Geom}_b\}$, and $\{q_i(s)\}_{1 \le i \le n; s \in \mathbb{N}}$ real positive, such that $f_i(x) = \sum_s q_i(s) g^{*s}(x)$. Then our new algorithm does it!



The new algorithm in a nutshell

Our new trick is based on the following ideas:

- Rejection algorithms have an extra factor in their complexity, on the scale of the inverse of the acceptance rate. In order to have the optimal complexity scaling, you need the average acceptance rate not to scale with the size n.
- Positive decomposition gives $f_i(x) = \sum_s q_i(s) g^{*s}(x)$.
  As a result the measure $\mu_{n,m}(\boldsymbol{x}) = \frac{1}{Z} \prod_{i=1}^{n} f_i(x_i) \times \delta_{|\boldsymbol{x}|,m}$ is a marginal of a measure in two sets of variables:
  $\mu_{n,m}(\boldsymbol{x}, \boldsymbol{s}) = \frac{1}{Z} \prod_{i=1}^{n} \left(q_i(s_i)\, g^{*s_i}(x_i) \right) \times \delta_{|\boldsymbol{x}|,m}$.
- You can first sample $\boldsymbol{s}$, with measure $\mu_1(\boldsymbol{s}) = \prod_{i=1}^{n} q_i(s_i)$, then accept this vector $\boldsymbol{s}$ with rate $a(\boldsymbol{s}) \propto g^{*|\boldsymbol{s}|}(m)$, and finally sample $\boldsymbol{x}$ with measure $\mu_2(\boldsymbol{x} \mid \boldsymbol{s}) = \prod_{i=1}^{n} g^{*s_i}(x_i)$.
- The acceptance rate is high because, although $g^{*|\boldsymbol{s}|}(m) = \mathcal{O}(n^{-\frac{1}{2}})$, we have $g^{*|\boldsymbol{s}|}(m) / \max_N (g^{*N}(m)) = \Theta(1)$.

We need something different...

Devroye uses the permutation symmetry of the random variables, and samples first the total number of x_i's equal to any given y (with a rejection scheme), then their reordering.

This is troublesome (use of float approximations for multinomial coefficients, need for extra tricks if the f_i's do not have finite support,...), and, most importantly, cannot be done here, as the variables are not identically distributed...

The obvious rejection scheme

A first algorithm is obtained by neglecting the linear global constraint. Assume (as usual) that $m = \sum_i \mathbb{E}[f_i]$ and $\sigma^2 := \sum_i \mathbb{V}\mathrm{ar}[f_i]$ are $\Theta(n)$. Up to a Lagrange multiplier, we can assume w.l.o.g. that $\mathbb{E}(|\boldsymbol{x}|) = m$.

This 'Boltzmann sampling' rejection algorithm would give:

Algorithm : Naïve rejection sampling *complexity* $\sim n^{3/2}$

begin
  repeat
    $|\boldsymbol{x}| = 0$;
    for $i \leftarrow 1$ *to* n **do** $\leftarrow$ *complexity* $\sim n$;
      $x_i \Leftarrow f_i$; $|\boldsymbol{x}|$+=x_i;
  until $|\boldsymbol{x}| = m$ $\leftarrow$ *complexity* $\sim \sqrt{n}$;
  return $(x_1, \dots, x_n)$
end

Shannon complexity bound

We have seen that the naïve algorithm has complexity $\sim n^{\frac{3}{2}}$.
This seems bad. But how bad exactly? How good can we possibly do?

Let us try to determine the intrinsic minimal complexity of this problem.

As we have seen in Olivier's talk, the time complexity is defined only up to a multiplicative constant, while for the random-bit complexity also the overall constants do matter.
Of course, the second one is a lower bound to the first one.

The intrinsic minimal random-bit complexity of an exact sampling problem is given by the Shannon entropy of the associated measure:

$$S[\mu] = -\sum_{\boldsymbol{x} \in \Omega_n} \mu(\boldsymbol{x}) \ln \mu(\boldsymbol{x})$$

Shannon complexity bound

Simple fact 1: if $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\mu_{\mathrm{iid}}(\boldsymbol{x}) = f_1(x_1) \cdots f_n(x_n)$,

$$S[\mu_{\mathrm{iid}}] = \sum_i S[f_i]$$

Simple fact 2: if also $p(s) = \mathbb{P}(|\boldsymbol{x}| = s)$ and $\mu_s(\boldsymbol{x}) = \frac{\mu_{\mathrm{iid}}(\boldsymbol{x})}{p(s)} \cdot \delta_{|\boldsymbol{x}|,s}$,

$$S[\mu_{\mathrm{iid}}] - S[p] = \sum_s p(s) S[\mu_s] = \mathbb{E}(S[\mu_s])$$

A bit more subtle: Suppose that $\mathbb{V}\mathrm{ar}[p] = \Theta(n)$, and p has the same value for mode and average, $s^* = \mathbb{E}_p(s) = \mathrm{argmax}(p(s))$. Then $\mathbb{E}(S[\mu_s]) \leq S[\mu_{s^*}]$, and in turns

$$S[\mu_{\mathrm{iid}}] - S[p] \leq S[\mu_{s^*}] \leq S[\mu_{\mathrm{iid}}]$$

so that

$$S[\mu_{s^*}] = \sum_i S[f_i] + \Theta(\ln n)$$

Shannon complexity bound

Simple fact 1: if $\boldsymbol{x} = (x_1, \dots, x_n)$ and $\mu_{\mathrm{iid}}(\boldsymbol{x}) = f_1(x_1) \cdots f_n(x_n)$,

$$S[\mu_{\mathrm{iid}}] = \sum_i S[f_i]$$

Simple fact 2: if also $p(s) = \mathbb{P}(|\boldsymbol{x}| = s)$ and $\mu_s(\boldsymbol{x}) = \frac{\mu_{\mathrm{iid}}(\boldsymbol{x})}{p(s)} \cdot \delta_{|\boldsymbol{x}|,s}$,

$$S[\mu_{\mathrm{iid}}] - S[p] = \sum_s p(s) S[\mu_s] = \mathbb{E}(S[\mu_s])$$

A bit more subtle: Suppose that $\mathbb{V}\mathrm{ar}[p] = \Theta(n)$, and p has the same value for mode and average, $s^* = \mathbb{E}_p(s) = \mathrm{argmax}(p(s))$. Then $\mathbb{E}(S[\mu_s]) \leq S[\mu_{s^*}]$, and in turns

$$S[\mu_{\mathrm{iid}}] - S[p] \leq S[\mu_{s^*}] \leq S[\mu_{\mathrm{iid}}]$$

so that

$$S[\mu_{s^*}] = \sum_i S[f_i] + \Theta(\ln n)$$

our complexity goal is linear

Do we know cases in which linearity is achievable by a conceptually-simple algorithm?

Do we know cases in which linearity is achievable
by a conceptually-simple algorithm?

Yes! You just saw this in Olivier's talk! Call this the BBHL algorithm.
The problem of uniformly sampling strings in $\{\bullet, \circ\}^n$
with $\#\{\bullet\} = k$ and $\#\{\circ\} = n - k$ is solved by the BBHL algo,
with linear time complexity and optimal random-bit complexity.

Do we know cases in which linearity is achievable by a conceptually-simple algorithm?

Yes! You just saw this in Olivier's talk! Call this the BBHL algorithm.
The problem of uniformly sampling strings in $\{\bullet, \circ\}^n$ with $\#\{\bullet\} = k$ and $\#\{\circ\} = n - k$ is solved by the BBHL algo, with linear time complexity and optimal random-bit complexity.

- BBHL$[n, m]$ solves the problem for $\mu_{n,m}(\boldsymbol{x}) = \frac{1}{Z} \prod_i \mathrm{Bern}_b(x_i) \times \delta_{|\boldsymbol{x}|,m}$, with $b = \frac{m}{n} \in]0, 1[$
- BBHL$[m + n - 1, m]$ solves the problem for $\mu_{n,m}(\boldsymbol{x}) = \frac{1}{Z} \prod_i \mathrm{Geom}_b(x_i) \times \delta_{|\boldsymbol{x}|,m}$, with $b = \frac{m}{n} \in]0, +\infty[$

Very good, but this is only for i.i.d. cases. . .

BBHL optimal shuffling: a reminder

Algorithm : BBHL shuffling algorithm

---

begin
- $a = k, \ b = n - k, \ i = 0;$
- **repeat**
  - i ++;
  - $\nu_i \longleftarrow \mathrm{Bern}_\beta;$
  - **if** $\nu_i = 1$ **then** a -- **else** b --
- **until** $a < 0$ *or* $b < 0$ — *complexity* $\sim n;$
- **if** $a < 0$ **then** $\bar{\nu} = 0$ **else** $\bar{\nu} = 1;$
- **for** $j \leftarrow i$ ***to*** n **do**
  - $\nu_j = \bar{\nu};$
  - $h \longleftarrow \mathrm{RndInt}_j;$
  - swap ν_j and ν_h — complexity $\sim \sqrt{n} \ln n;$
- **return** $\boldsymbol{\nu}$

end

The rejection paradigm

Recall the naïve rejection algorithm:

You want to do exact sampling for $\mu(\boldsymbol{x})$,
when $\mu(\boldsymbol{x}) \propto \mu_0(\boldsymbol{x})a(\boldsymbol{x})$, with $a(\boldsymbol{x}) \in [0,1]$,
supposing that you know how to sample from μ_0

Algorithm : Rejection sampling $\qquad T[\mu] \sim T[\mu_0]\,\mathbb{E}(a(\boldsymbol{x}))^{-1}$

begin
  repeat
    $\boldsymbol{x} \Leftarrow \mu_0$; $\qquad \leftarrow$ complexity $T[\mu_0]$;
    $\alpha \Leftarrow \mathrm{Bern}_{a(\boldsymbol{x})}$;
  until $\alpha = 1$ $\qquad \leftarrow$ *complexity* $\mathbb{E}(a(\boldsymbol{x}))^{-1}$;
  return $\boldsymbol{x}$
end

The rejection paradigm for decomposed measures

Now assume $\mu(\boldsymbol{x}) \propto \sum_{\boldsymbol{y}} \mu_1(\boldsymbol{y})\, \mu_2(\boldsymbol{x} \,|\, \boldsymbol{y})\, a(\boldsymbol{y})$, with $a(\boldsymbol{y}) \in [0,1]$, supposing that you know how to sample from μ_1, and $\mu_2(\cdot \,|\, \boldsymbol{y})$

Algorithm : Rejection sampling for decomposed measures

begin
  repeat
    $\boldsymbol{y} \Leftarrow \mu_1$; $\qquad \leftarrow$ sample a tentative $\boldsymbol{y}$ with μ_1;
    $\alpha \Leftarrow \mathrm{Bern}_{a(\boldsymbol{y})}$;
  until $\alpha = 1$ $\qquad \leftarrow$ *accept $\boldsymbol{y}$ with rate $a(\boldsymbol{y})$*;
  $\boldsymbol{x} \Leftarrow \mu_2(\cdot \,|\, \boldsymbol{y})$; $\qquad \leftarrow$ sample $\boldsymbol{x}$ with $\mu_2(\,\cdot\,|\boldsymbol{y})$;
  return $\boldsymbol{x}$
end

$$T = \frac{\sum_{\boldsymbol{y}} \mu_1(\boldsymbol{y})\big(T_1(\boldsymbol{y}) + a(\boldsymbol{y})\, T_2(\boldsymbol{y})\big)}{\sum_{\boldsymbol{y}} \mu_1(\boldsymbol{y}) a(\boldsymbol{y})} = \frac{\mathbb{E}(T_1 + a T_2)}{\mathbb{E}(a)} \leq \frac{T_1^{\max}}{\mathbb{E}(a)} + T_2^{\max}$$

The rejection paradigm for decomposed measures

Now assume $\mu(\boldsymbol{x}) \propto \sum_{\boldsymbol{y}} \mu_1(\boldsymbol{y})\, \mu_2(\boldsymbol{x} \,|\, \boldsymbol{y})\, a(\boldsymbol{y})$, with $a(\boldsymbol{y}) \in [0,1]$, supposing that you know how to sample from μ_1, and $\mu_2(\cdot \,|\, \boldsymbol{y})$

Algorithm : Rejection sampling for decomposed measures

begin
- **repeat**
  - $\boldsymbol{y} \Leftarrow \mu_1$; $\leftarrow$ complexity $T_1(\boldsymbol{y})$;
  - $\alpha \Leftarrow \mathrm{Bern}_{a(\boldsymbol{y})}$;
- **until** $\alpha = 1$ $\leftarrow$ *complexity* $a(\boldsymbol{y})^{-1}$;
- $\boldsymbol{x} \Leftarrow \mu_2(\cdot \,|\, \boldsymbol{y})$; $\leftarrow$ complexity $T_2(\boldsymbol{y})$;
- **return** $\boldsymbol{x}$

end

$$T = \frac{\sum_{\boldsymbol{y}} \mu_1(\boldsymbol{y})\big(T_1(\boldsymbol{y}) + a(\boldsymbol{y})\, T_2(\boldsymbol{y})\big)}{\sum_{\boldsymbol{y}} \mu_1(\boldsymbol{y}) a(\boldsymbol{y})} = \frac{\mathbb{E}(T_1 + a T_2)}{\mathbb{E}(a)} \leq \frac{T_1^{\max}}{\mathbb{E}(a)} + T_2^{\max}$$

Positive decomposition provides a decomposed measure

Positive decomposition tells that, for all i,
$f_i(x) = \sum_s q_i(s) g^{*s}(x)$, with $q_i(s) \geq 0$.

From the normalisation of the f_i's and of g,
it follows that also the $q_i(s)$ are probability distributions.

As a result the measure $\mu_{n,m}(\boldsymbol{x}) = \frac{1}{Z} \prod_{i=1}^{n} f_i(x_i) \times \delta_{|\boldsymbol{x}|,m}$
is a marginal of a measure in two sets of variables:
$\mu_{n,m}(\boldsymbol{x}, \boldsymbol{s}) = \frac{1}{Z} \prod_{i=1}^{n} \left(q_i(s_i)\, g^{*s_i}(x_i) \right) \times \delta_{|\boldsymbol{x}|,m}$.

This is exactly as in a decomposed measure, with correspondence

sample $\boldsymbol{s}$	with measure $\mu_1(\boldsymbol{s})$	$\mu_1(\boldsymbol{s}) = \prod_{i=1}^{n} q_i(s_i)$
accept $\boldsymbol{s}$	with rate $a(\boldsymbol{s})$	$a(\boldsymbol{s}) \propto g^{*\lvert\boldsymbol{s}\rvert}(m)$
sample $\boldsymbol{x}$	with measure $\mu_2(\boldsymbol{x} \mid \boldsymbol{s})$	$\mu_2(\boldsymbol{x} \mid \boldsymbol{s}) = \prod_{i=1}^{n} g^{*s_i}(x_i)$

Note: although μ_1 and μ_2 depend on the vector $\boldsymbol{s}$,
the rate a only depends on $|\boldsymbol{s}| = \sum_i s_i$.

Increasing the acceptance rate

The crucial point is that the decomposition allows to increase the acceptance rate!

In the 'ordinary' rejection scheme, you accept $\boldsymbol{x}$ iff a probabilistic event occurs (in our case, $|\boldsymbol{x}| = m$). If this probability is intrinsically small (in our case, $\Theta(n^{-1/2})$), there is nothing you can do.

In the rejection scheme for decomposed measures, the rate $a(\boldsymbol{s})$ is defined up to a multiplicative factor, as long as $\max_{\boldsymbol{s}} a(\boldsymbol{s}) \leq 1$.

Here, the obvious choice for $a(\boldsymbol{s})$ is $a(\boldsymbol{s}) = g^{*|\boldsymbol{s}|}(m)$, which is $\mathcal{O}(n^{-\frac{1}{2}})$.

However, we can push it up to $a(\boldsymbol{s}) = \dfrac{g^{*|\boldsymbol{s}|}(m)}{\max_N(g^{*N}(m))}$.

As we will see, with this choice $\mathbb{E}(a(\boldsymbol{s})) = \Theta(1)$.

How to sample from $\mathrm{Bern}_{a(\boldsymbol{s})}$

This idea is not sufficient by itself. Even if you know in advance that, after maximisation, $\mathbb{E}(a(\boldsymbol{s})) = \Theta(1)$, you still have a problem:

sampling a Bernoulli rnd var with parameter $a(\boldsymbol{s})$ is difficult
if you do not have an analytic expression for $a(\boldsymbol{s})$.

It is not compulsory to have an analytic expression for $a(\boldsymbol{s})$
(just think to how the Monte Carlo algorithm:
$x \Leftarrow \mathrm{Rnd}[0,1]$; $y \Leftarrow \mathrm{Rnd}[0,1]$; **return** $\mathrm{sign}(1 - x^2 - y^2)$
samples $\mathrm{Bern}_{\pi/4}$ without knowing $\pi\dots$)

however, it makes life easier, and in our case we have it for free
if we choose the base function $g(x)$ for positive decomposition
in the list $g(x) \in \{\mathrm{Bern}_b, \mathrm{Poiss}, \mathrm{Geom}_b\}$

How to sample from $\mathrm{Bern}_{a(\boldsymbol{s})}$

Example with Bernoulli (the other cases are similar)
(just write $a(s)$ for $a(\boldsymbol{s})$, with $s = |\boldsymbol{s}|$)

$$a(s) = \frac{g^{*s}(m)}{\max_N(g^{*N}(m))} = \frac{b^m(1-b)^{s-m}\binom{s}{m}}{\max_N\left(b^m(1-b)^{N-m}\binom{N}{m}\right)}$$

The max is realised for $N = \bar{N} := \lfloor m/b \rfloor$, thus

$$a(s) = (1-b)^{s-\bar{N}} \frac{s!(\bar{N}-m)!}{\bar{N}!(s-m)!}$$

Good news 1: This is easily evaluated to high precision
(i.e., calculating d binary digits has complexity $\ll 2^d$),
so that the average cost of $\mathrm{Bern}_{a(s)}$ is $\Theta(1)$.

Good news 2: For large m, and $b = \Theta(1)$, $a(s)$ converges to an un-normalised Gaussian centered around $\bar{N}$, and of variance $\Theta(m)$.

A rough evaluation of the complexity

Recall the basic steps in the rejection algo for our decomposed measure:

sample $\boldsymbol{s}$	with measure $\mu_1(\boldsymbol{s})$	$\mu_1(\boldsymbol{s}) = \prod_{i=1}^n q_i(s_i)$
accept $\boldsymbol{s}$	with rate $a(s)$	$a(s) = g^{*s}(m)/g^{*\bar{N}}(m)$
sample $\boldsymbol{x}$	with measure $\mu_2(\boldsymbol{x} \mid \boldsymbol{s})$	$\mu_2(\boldsymbol{x} \mid \boldsymbol{s}) = \prod_{i=1}^n g^{*s_i}(x_i)$

and that this algorithm has complexity

$$T \leq \frac{T_1^{\max}}{\mathbb{E}_{\mu_1}(a(s))} + T_2^{\max} \qquad \text{where } T_1^{\max}, T_2^{\max} = \Theta(n).$$

Under mild CLT hypotheses, the measure on $s = |\boldsymbol{s}|$ induced by $\mu_1(\boldsymbol{s})$ is a (normalised) Gaussian centered in $\bar{N}$, with variance $\sigma_1^2 n$, while $a(s)$ is an un-normalised Gaussian, centered in $\bar{N}$, with variance $\sigma_2^2 n$:

$$\mathbb{E}(a) \simeq \int \mathrm{d}x \, \frac{1}{\sqrt{2\pi\sigma_1^2 n}} \exp\left[-\frac{x^2}{2n} \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \right) \right] = \frac{\sigma_2}{\sqrt{\sigma_1^2 + \sigma_2^2}}$$

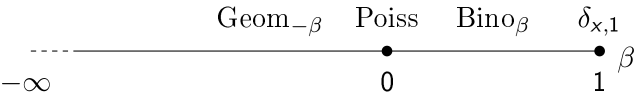
$$T \lesssim T_1^{\max} \sqrt{1 + (\sigma_1/\sigma_2)^2} + T_2^{\max} = \Theta(n)$$

The precise result

The three fundamental distributions

$$g_\beta^{*s}(r) = \begin{cases} \mathrm{Bern}_\beta^{*s}(r) = \beta^r (1-\beta)^{s-r} \binom{s}{r} & \beta \in]0,1[\\ \mathrm{Poiss}_s(r) = e^{-s} \frac{s^r}{r!} & \beta = 0 \\ \mathrm{Geom}_{-\beta}^{*s}(r) = |\beta|^r (1+|\beta|)^{-s-r} \binom{s+r-1}{r} & \beta \in]-\infty, 0[\end{cases}$$

are such that g_α^{*s} has a positive decomposition in g_β iff $\alpha \leq \beta$.

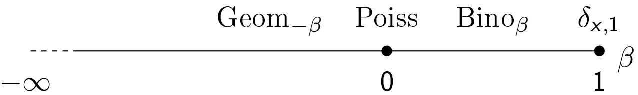


For the list of functions $\mathcal{F} = \{f_1, \dots, f_n\}$ in our measure,
call $\beta_{\min}(\mathcal{F})$ the smallest value of β
such that all the f_i's have a positive decomposition in g_β.

The precise result

Then, the largest value for $\mathbb{E}(a)$
that can be achieved within our framework is

$$\boxed{a_{\max}(\mathcal{F}) := \sqrt{1 - \beta_{\min}(\mathcal{F})} \cdot \sqrt{\frac{\sum_i \mathbb{E}[f_i]}{\sum_i \mathbb{V}\mathrm{ar}[f_i]}}}$$



Examples of application

So, we have constructed our algorithm
for the linear-time exact sampling of sum-constrained random variables,
in the case in which they are *not* equally distributed.

However, you could just think:
«who cares about not-equally-distributed variables?
After all, every time I wanted to generate walks, trees, etc.,
I always wanted equally-distributed variables. . . »

The point is: examples of this sort
may be hidden beyond some smart bijection,
starting from more customary (and symmetric) problems.

This is well illustrated by two classical examples:

- Set partitions, and Stirling numbers of the second kind
- Permutations with m cycles, and Stirling numbers of the first kind

Set partitions, and Stirling numbers of the second kind

Call $\mathcal{S}^{\mathrm{set}}_{n,m}$ the ensemble of partitions of a set with n (labelled) elements into m (unlabeled) non-empty subsets.

W.l.o.g. we can assume that the set has a total ordering.

Example, for $(n, m) = (28, 9)$, and the set

$$\{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, \alpha, \beta\}$$

consider the partition

$$\big\{\{a, g, t\}, \{b, d, m, o, \alpha\}, \{c, j, s, y\}, \{e, h, v\}, \{f, k, q\}, \{i, l, z\}, \{n, p, u\}, \{r, \beta\}, \{w, x\}\big\}$$

Call $\mathcal{S}^{\mathrm{set}}_{n,m}$ the ensemble of partitions of a set with n (labelled) elements into m (unlabeled) non-empty subsets.

W.l.o.g. we can assume that the set has a total ordering.

Example, for $(n, m) = (28, 9)$, and the set

$$\{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, \alpha, \beta\}$$

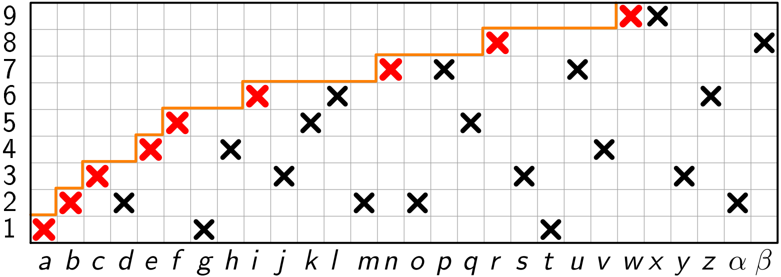
consider the partition

$$\big\{\{a, g, t\}, \{b, d, m, o, \alpha\}, \{c, j, s, y\}, \{e, h, v\}, \{f, k, q\}, \{i, l, z\}, \{n, p, u\}, \{r, \beta\}, \{w, x\}\big\}$$

Although the sets are not labeled, they are canonically ordered, e.g. by their smallest element. As a result, we have a canonical incidence matrix T, with $T_{ij} = 1$ if the element j is in subset i.

Set partitions, and Stirling numbers of the second kind

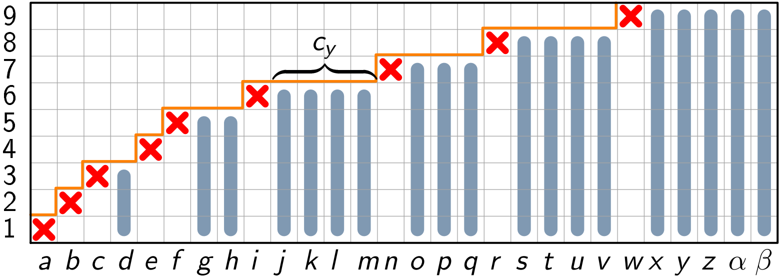
$$\{\{a,g,t\},\{b,d,m,o,\alpha\},\{c,j,s,y\},\{e,h,v\},\{f,k,q\},\{i,l,z\},\{n,p,u\},\{r,\beta\},\{w,x\}\}$$



Call backbone $B(T)$ the list of smallest elements in the subsets, here $B = \{a, b, c, e, f, i, n, r, w\}$.

Set partitions, and Stirling numbers of the second kind

$$\{\{a,g,t\},\{b,d,m,o,\alpha\},\{c,j,s,y\},\{e,h,v\},\{f,k,q\},\{i,l,z\},\{n,p,u\},\{r,\beta\},\{w,x\}\}$$



Call backbone $B(T)$ the list of smallest elements in the subsets, here $B = \{a, b, c, e, f, i, n, r, w\}$. The number of partitions T with $B(T) = B$ is the trivial product: $\prod_{y=1}^{m} y^{c_y}$, but the quantities c_y are linearly constrained: $\sum_y c_y = n - m$.

Set partitions, and Stirling numbers of the second kind

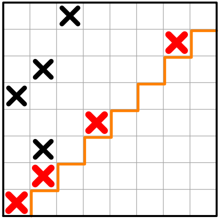
As a result, sampling uniformly set partitions in $\mathcal{S}_{n,m}$, which bijectively coincides to sampling uniformly the tableaux T, boils down to sampling the backbone B with the non-uniform measure
$$\mu_{n,m}(c_1,\dots,c_m) \propto \prod_{y=1}^{m} y^{c_y} \times \delta_{|c|,n-m}$$

This is exactly our framework! Introduce an appropriate Lagrange multiplier $\omega^{\sum_y c_y}$, in order to have $\mathbb{E}(|c|) = n-m$
(the good choice is the solution to the equation $\frac{n}{m} = -\frac{\ln(1-\omega)}{\omega}$)
The functions $f_y(c_y)$ are $\mathrm{Geom}_{b_y}(c_y)$, with $b_y = \frac{\omega y}{n-\omega y}$

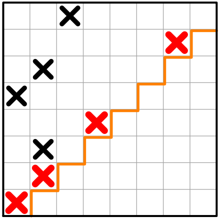
Now, Geom_a has a positive decomposition in terms of Bern_b
$$\mathrm{Geom}_a(x) = \sum_s \mathrm{Geom}_{\frac{a}{a+b}}(s)\,\mathrm{Bino}_{s,b}(x)$$

Choosing for simplicity $b=\frac{1}{2}$, our algorithm works, with an average acceptance rate $\mathbb{E}(a) = \sqrt{\frac{e^{-\theta}-1+\theta}{2(e^{\theta}-1-\theta)}}$ $\qquad (\omega = 1-e^{-\theta})$

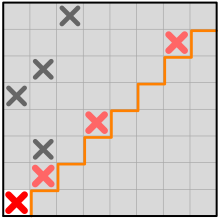
Call $\mathcal{S}^{\mathrm{cyc}}_{n,m}$ the set of permutations $\sigma \in \mathfrak{S}_n$ with m cycles.
Describe σ through the insertion table associated to its growth,
for example, for $\sigma = \big((15)(2638)(4)(7)\big)$



Call $\mathcal{S}^{\mathrm{cyc}}_{n,m}$ the set of permutations $\sigma \in \mathfrak{S}_n$ with m cycles.
Describe σ through the insertion table associated to its growth,
for example, for $\sigma = \big((15)(2638)(4)(7)\big)$

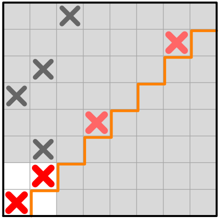


Call $\mathcal{S}^{\mathrm{cyc}}_{n,m}$ the set of permutations $\sigma \in \mathfrak{S}_n$ with m cycles.
Describe σ through the insertion table associated to its growth,
for example, for $\sigma = \big((15)(2638)(4)(7)\big)$



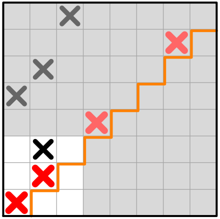
$\sigma = \big((1)\big)$

Call $\mathcal{S}^{\mathrm{cyc}}_{n,m}$ the set of permutations $\sigma \in \mathfrak{S}_n$ with m cycles.
Describe σ through the insertion table associated to its growth,
for example, for $\sigma = \big((15)(2638)(4)(7)\big)$



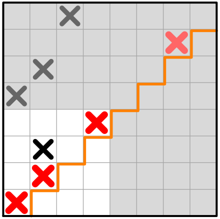
$\sigma = \big((1)(2)\big)$

Call $\mathcal{S}^{\mathrm{cyc}}_{n,m}$ the set of permutations $\sigma \in \mathfrak{S}_n$ with m cycles.
Describe σ through the insertion table associated to its growth,
for example, for $\sigma = \big((15)(2638)(4)(7)\big)$



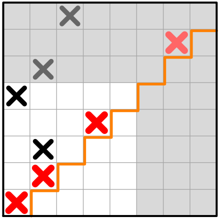
$\sigma = \big((1)(23)\big)$

Call $\mathcal{S}^{\mathrm{cyc}}_{n,m}$ the set of permutations $\sigma \in \mathfrak{S}_n$ with m cycles.
Describe σ through the insertion table associated to its growth,
for example, for $\sigma = \big((15)(2638)(4)(7)\big)$



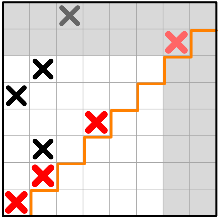
$\sigma = \big((1)(23)(4)\big)$

Call $\mathcal{S}^{\mathrm{cyc}}_{n,m}$ the set of permutations $\sigma \in \mathfrak{S}_n$ with m cycles.
Describe σ through the insertion table associated to its growth,
for example, for $\sigma = \big((15)(2638)(4)(7)\big)$



$\sigma = \big((15)(23)(4)\big)$

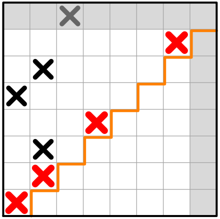
Call $\mathcal{S}^{\mathrm{cyc}}_{n,m}$ the set of permutations $\sigma \in \mathfrak{S}_n$ with m cycles.
Describe σ through the insertion table associated to its growth,
for example, for $\sigma = \big((15)(2638)(4)(7)\big)$



$\sigma = \big((15)(263)(4)\big)$

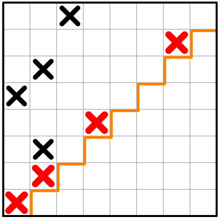
m-cycle permutations, and Stirling numbers of the 1st kind

Call $\mathcal{S}^{\mathrm{cyc}}_{n,m}$ the set of permutations $\sigma \in \mathfrak{S}_n$ with m cycles.
Describe σ through the insertion table associated to its growth,
for example, for $\sigma = \big((15)(2638)(4)(7)\big)$



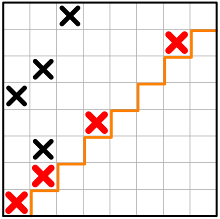
$\sigma = \big((15)(263)(4)(7)\big)$

Call $\mathcal{S}^{\mathrm{cyc}}_{n,m}$ the set of permutations $\sigma \in \mathfrak{S}_n$ with m cycles.
Describe σ through the insertion table associated to its growth,
for example, for $\sigma = \big((15)(2638)(4)(7)\big)$



$\sigma = \big((15)(2638)(4)(7)\big)$

Call $\mathcal{S}^{\mathrm{cyc}}_{n,m}$ the set of permutations $\sigma \in \mathfrak{S}_n$ with m cycles.
Describe σ through the insertion table associated to its growth,
for example, for $\sigma = ((15)(2638)(4)(7))$

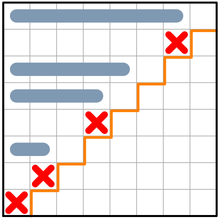


$\sigma = ((15)(2638)(4)(7))$

Call $B(\sigma) = \{0, 0, 1, 0, 1, 1, 0, 1\}$,
the indicator function of "black rows"
of $T(\sigma)$, the backbone of σ.

m-cycle permutations, and Stirling numbers of the 1st kind

Call $\mathcal{S}^{\mathrm{cyc}}_{n,m}$ the set of permutations $\sigma \in \mathfrak{S}_n$ with m cycles.
Describe σ through the insertion table associated to its growth,
for example, for $\sigma = ((15)(2638)(4)(7))$



$\sigma = ((15)(2638)(4)(7))$
Call $B(\sigma) = \{0, 0, 1, 0, 1, 1, 0, 1\}$,
the indicator function of “black rows”
of $T(\sigma)$, the backbone of σ.

The number of σ's with backbone
$B = (x_1, \dots, x_n)$ is $\prod_y (y-1)^{x_y}$,
and we must have $|\boldsymbol{x}| = m$

Again, this is exactly our framework!
just with inhomogeneous Bernoulli variables,
instead of inhomogeneous Geometric variables.

Conclusions

Our problem was the exact sampling in linear time from the measure $\mu_{n,m}(\boldsymbol{x}) = \frac{1}{Z}\prod_{i=1}^{n} f_i(x_i) \times \delta_{|\boldsymbol{x}|,m}$

We have provided a solution to this problem in the case in which all the f_i's have a positive decomposition in terms of the same function g_β

$$g_\beta(r) = \begin{cases} \mathrm{Bern}_\beta(r) & \beta \in]0,1[\\ \mathrm{Poiss}_1(r) & \beta = 0 \\ \mathrm{Geom}_{-\beta}(r) & \beta \in]-\infty, 0[\end{cases}$$

In this case, the rejection scheme has a complexity related to the average acceptance rate, which is

$$a_{\max} = \sqrt{1-\beta} \cdot \sqrt{(\mathbb{E}[f_1] + \cdots + \mathbb{E}[f_n])/(\mathbb{V}\mathrm{ar}[f_1] + \cdots + \mathbb{V}\mathrm{ar}[f_n])}$$

Applications include the uniform exact sampling of set partitions with a prescribed number of subsets (and, in turns, minimal automata), and of permutations with a prescribed number of cycles.