The density method for permutations with prescribed descents

Philippe Marchal

CNRS/University Paris Nord

GASCom 2018, Athens, 18-20 June 2018

The set-up

σ: permutation of $\{1, 2, \ldots n\}$.
Descent set: $\mathcal{D} = \{i \in [1, n-1],\ \sigma(i) > \sigma(i+1)\}$.
Goal: enumerate and generate uniformly at random permutations with a given descent set.

The set-up

σ: permutation of $\{1, 2, \ldots n\}$.
Descent set: $\mathcal{D} = \{i \in [1, n-1],\ \sigma(i) > \sigma(i+1)\}$.
Goal: enumerate and generate uniformly at random permutations with a given descent set.

Naive counting

Inclusion-exclusion formula: number of permutations with descent set $\mathcal{D}$ is

$$\sum_{\mathcal{E}=\{i_1<i_2\ldots<i_k\}\subset\mathcal{D}^c} (-1)^{|\mathcal{D}^c|-|\mathcal{E}|} \binom{n}{n-i_k, i_k-i_{k-1}, \ldots, i_2-i_1, i_1}$$

The set-up

σ: permutation of $\{1, 2, \ldots n\}$.
Descent set: $\mathcal{D} = \{i \in [1, n-1],\ \sigma(i) > \sigma(i+1)\}$.
Goal: enumerate and generate uniformly at random permutations with a given descent set.

Naive counting

Inclusion-exclusion formula: number of permutations with descent set $\mathcal{D}$ is

$$\sum_{\mathcal{E}=\{i_1<i_2\ldots<i_k\}\subset\mathcal{D}^c} (-1)^{|\mathcal{D}^c|-|\mathcal{E}|} \binom{n}{n-i_k, i_k-i_{k-1}, \ldots, i_2-i_1, i_1}$$

Practical problem: number of computations to perform. If $n = 100$, $|\mathcal{D}| = 50 \to 2^{50}$ terms in the summation.

Generating and counting

Alternating permutations [André 1879]. Exponential generating function

$$\sum_n \frac{a_n}{n!} t^n = \sec t + \tan t$$

Other periodic patterns [Carlitz '75, Mendes et al. '10, Luck '14].

Generating and counting

Alternating permutations [André 1879]. Exponential generating function

$$\sum_n \frac{a_n}{n!} t^n = \sec t + \tan t$$

Other periodic patterns [Carlitz '75, Mendes et al. '10, Luck '14].

Naive sampling

Number of alternating permutations of length $n \sim \frac{4}{\pi}\left(\frac{2}{\pi}\right)^{n+1} n!$.

The rejection algorithm has exponential complexity $\sim \frac{\pi}{4}\left(\frac{\pi}{2}\right)^{n+1}$.

Generating and counting

Alternating permutations [André 1879]. Exponential generating function

$$\sum_n \frac{a_n}{n!} t^n = \sec t + \tan t$$

Other periodic patterns [Carlitz '75, Mendes et al. '10, Luck '14].

Naive sampling

Number of alternating permutations of length $n \sim \frac{4}{\pi} \left(\frac{2}{\pi}\right)^{n+1} n!$.
The rejection algorithm has exponential complexity $\sim \frac{\pi}{4} \left(\frac{\pi}{2}\right)^{n+1}$.

Goal

Introduce the **density method** to generate and enumerate. Alternative approaches exist [Viennot '79, Basset '16] but our method also applies in other contexts.

The order polytope

Idea: the descent set induces a poset on $\{1, \dots n\}$.
For instance, if $\sigma(1) < \sigma(2)$ and $\sigma(2) > \sigma(3)$ we equip the set $\{1, 2, 3\}$ with the order $\prec$ defined by $1 \prec 2$, $3 \prec 2$.

.

The order polytope

Idea: the descent set induces a poset on $\{1, \dots n\}$.
For instance, if $\sigma(1) < \sigma(2)$ and $\sigma(2) > \sigma(3)$ we equip the set $\{1, 2, 3\}$ with the order $\prec$ defined by $1 \prec 2$, $3 \prec 2$.
Generate a random element of the associated *order polytope*

$$\mathcal{P} = \{(Y_1, Y_2, Y_3) \in [0, 1]^3, Y_1 < Y_2, Y_3 < Y_2\}$$

Then recover the permutation from (Y_1, Y_2, Y_3) by sorting: $\sigma(i) = k$ if i is the k-th smallest element in $\{Y_1, Y_2, Y_3\}$.

The order polytope

Idea: the descent set induces a poset on $\{1, \dots n\}$.
For instance, if $\sigma(1) < \sigma(2)$ and $\sigma(2) > \sigma(3)$ we equip the set $\{1, 2, 3\}$ with the order $\prec$ defined by $1 \prec 2$, $3 \prec 2$.
Generate a random element of the associated *order polytope*

$$\mathcal{P} = \{(Y_1, Y_2, Y_3) \in [0, 1]^3, Y_1 < Y_2, Y_3 < Y_2\}$$

Then recover the permutation from (Y_1, Y_2, Y_3) by sorting: $\sigma(i) = k$ if i is the k-th smallest element in $\{Y_1, Y_2, Y_3\}$.

In the general case, we want to generate a sequence (Y_i) of reals $\in [0, 1]$ satisfying the order relation induced by the descent set.

The density method: computing densities

To construct the (Y_i), we use the notion of *density* of a random variable.
$f : \mathbb{R} \to \mathbb{R}_+$ is the density of $X \in \mathbb{R}$ if, for all reals $a \leq b$,

$$\mathbb{P}(X \in [a, b]) = \int_a^b f(x)dx$$

.

The density method: computing densities

To construct the (Y_i), we use the notion of *density* of a random variable.
$f : \mathbb{R} \to \mathbb{R}_+$ is the density of $X \in \mathbb{R}$ if, for all reals $a \leq b$,

$$\mathbb{P}(X \in [a, b]) = \int_a^b f(x)dx$$

Define by induction the sequence of functions

- $f_n(x) = 1$.
- If $i \in \mathcal{D}$, $f_i(x) = \int_0^x f_{i+1}(y)dy$.
- If $i \notin \mathcal{D}$, $f_i(x) = \int_x^1 f_{i+1}(y)dy$.

Number of computations: $O(n^2)$.

The density method: sampling in the polytope

Let $U_i, 0 \leq i \leq n$ be iid, uniform on $[0, 1]$.

- Y_1 has density proportional to f_1: Y_1 is the solution $\in [0, 1]$ of

$$U_1 \int_0^1 f_1(y)dy = \int_0^{Y_1} f_1(y)dy$$

  Polynomial equation, the solution is $Y_1 = 0$ if $U_1 = 0$ and $Y_1 = 1$ if $U_1 = 1$. In the general case, since $U_1 \in [0, 1]$, $Y_1 \in [0, 1]$.

The density method: sampling in the polytope

Let $U_i, 0 \leq i \leq n$ be iid, uniform on $[0,1]$.

- Y_1 has density proportional to f_1: Y_1 is the solution $\in [0,1]$ of
$$U_1 \int_0^1 f_1(y)dy = \int_0^{Y_1} f_1(y)dy$$
Polynomial equation, the solution is $Y_1 = 0$ if $U_1 = 0$ and $Y_1 = 1$ if $U_1 = 1$. In the general case, since $U_1 \in [0,1]$, $Y_1 \in [0,1]$.
- Y_{i+1} has conditional density proportional to f_{i+1}:
  - If $i \in \mathcal{D}$, recall that $f_i(x) = \int_0^x f_{i+1}(y)dy$. Then
  $$U_{i+1} f_i(Y_i) = \int_0^{Y_{i+1}} f_{i+1}(y)dy$$
  - If $i \notin \mathcal{D}$, $U_{i+1} f_i(Y_i) = \int_{Y_{i+1}}^1 f_{i+1}(y)dy$.

The algorithm

- Generate random variables (Y_i) as above.
- Sort the (Y_i) to generate a random permutation.

The algorithm

- Generate random variables (Y_i) as above.
- Sort the (Y_i) to generate a random permutation.

Theorem

The algorithm generates a random, uniform permutation with descent set $\mathcal{D}$ using $O(n^2)$ computations.

The algorithm

- Generate random variables (Y_i) as above.
- Sort the (Y_i) to generate a random permutation.

Theorem

The algorithm generates a random, uniform permutation with descent set $\mathcal{D}$ using $O(n^2)$ computations.

Computing the densities takes $O(n^2)$ computations, generating the (Y_i) takes $O(n^2)$ computations and sorting takes $O(n \log n)$ computations.
We have to show that the random permutation is uniform.

Sketch of the proof

The density of Y_1 is $c_1 f_1$, where

$$c_1 = \frac{1}{\int_0^1 f_1(x)dx}$$

Sketch of the proof

The density of Y_1 is $c_1 f_1$, where

$$c_1 = \frac{1}{\int_0^1 f_1(x)dx}$$

Suppose for instance that 1 is a descent.
The density of Y_2, conditional on Y_1, is $c_2 f_2 \mathbf{1}_{\{Y_2 \leq Y_1\}}$.
Since $f_1(x) = \int_0^x f_2(y)dy$,

$$c_2 = \frac{1}{f_1(Y_1)}$$

Sketch of the proof

The density of Y_1 is $c_1 f_1$, where

$$c_1 = \frac{1}{\int_0^1 f_1(x)dx}$$

Suppose for instance that 1 is a descent.
The density of Y_2, conditional on Y_1, is $c_2 f_2 \mathbf{1}_{\{Y_2 \leq Y_1\}}$.
Since $f_1(x) = \int_0^x f_2(y)dy$,

$$c_2 = \frac{1}{f_1(Y_1)}$$

By independence, the density of the sequence $(Y_1, \dots Y_n)$ is the telescopic product

$$c_1 f_1(Y_1) \frac{f_2(Y_2)}{f_1(Y_1)} \dots \frac{f_n(Y_n)}{f_{n-1}(Y_{n-1})} \times \mathbf{1}_{\mathcal{D}} = c_1 \mathbf{1}_{\mathcal{D}}$$

Counting

Theorem (Enumeration)

The number of permutations with descent set $\mathcal{D}$ is

$$n! \int_0^1 f_1(x)dx$$

Counting

Theorem (Enumeration)

The number of permutations with descent set $\mathcal{D}$ is

$$n! \int_0^1 f_1(x)dx$$

Theorem (Monotonicity result)

Let $\mathcal{B}$ be the set of local extrema of the permutation. Let $F(\mathcal{B})$ be the number of permutations with set of local extrema $\mathcal{B}$. Then $F : \mathcal{P}(\{2, 3, \ldots n-1\} \to \mathbb{N}$ is an increasing function.

In particular, we recover the fact that alternating permutations are the most likely profile [De Bruijn '70, Ehrenborg et al. '02].

Periodic descent set

Define f_n by increasing induction. Let $F(x,y) = \sum_n y^n f_n(x)$. Then

$$\sum_{n\geq 0} \frac{a_n}{n!} y^n = 1 + \int_0^1 yF(x,y)dx$$

a_n: number of permutations of length n with the desired pattern.

Periodic descent set

Define f_n by increasing induction. Let $F(x,y) = \sum_n y^n f_n(x)$. Then

$$\sum_{n\geq 0} \frac{a_n}{n!} y^n = 1 + \int_0^1 yF(x,y)dx$$

a_n: number of permutations of length n with the desired pattern.
If n is a descent (resp. an ascent), $f'_{n+1} = f_n$ (resp $f'_{n+1} = -f_n$).

$$\frac{\partial^p F(x,y)}{\partial x^p} = (-1)^k y^p F(x,y)$$

where p is the period and k the number of ascents in a period.
Solve the equation using the boundary conditions.

Further problems

- Other problems on permutations: random generation and enumeration when the sequence of ascents and descents is a word belonging to a given set. Rational language: [Basset '16]. More general questions?

Further problems

- Other problems on permutations: random generation and enumeration when the sequence of ascents and descents is a word belonging to a given set. Rational language: [Basset '16]. More general questions?
- The density method can be used for other problems of posets:
  - rectangular Young tableaux [Marchal '16]
  - urn models and corners of triangular Young tableaux [Banderier-Marchal-Wallner '18]
  - Young tableaux with local decreases (our forthcoming talk)
  - permutations with algebraic language patterns (in progress)