Generation zufälliger Partitionen

Stefan Volz

9. Juni 2021

Zusammenfassung

Wir bezeichnen ein k-Tupel $(c_1, ..., c_k) \in \mathbb{N}_0^k$ mit Elementen $c_i \leq r \in \mathbb{N}$ als geordnete r-beschränkte k-Partition von $n \in \mathbb{N}$, wenn $\sum_{j=1}^{k} c_j = n$. Bezeichne mit $\mathcal{P}_k^r(n)$ die Menge dieser Partitionen. Gesucht wird ein effizienter Algorithmus um gleichverteilt ein zufälliges Element aus $\mathcal{P}_k^r(n)$ zu generieren.

Inhaltsverzeichnis

1 Unbeschränkte Partitionen **2**
- 1.1 Motivation . . . 2
- 1.2 Geschlossene Form von $\psi_k(n)$ . . . 3

2 Beschränkte Partitionen **5**
- 2.1 Motivation . . . 5
- 2.2 Wahrscheinlichkeitsverteilung der beschränkten Partitionen . . . 5
- 2.3 Effiziente Bestimmung von $P(X_j = m)$ . . . 6
- 2.4 Implementierung . . . 7

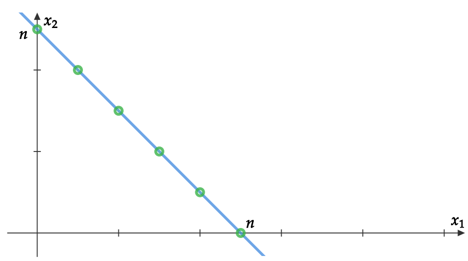
1 Unbeschränkte Partitionen

Betrachten wir zunächst die Menge der unbeschränkten Partitionen $\mathcal{P}_k^n(n) =: \mathcal{P}_k(n)$.

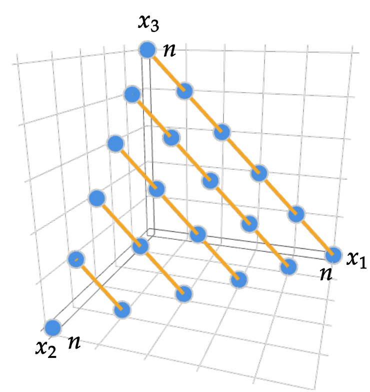
Notation 1 *Wir bezeichnen die Anzahl an unbeschränkten Partitionen mit $\psi_k(n)$ und nennen k die Dimension dieser Partitionen.*

1.1 Motivation

Betrachten wir $\mathcal{P}_k(n)$ als die Menge der Lösungen $(x_1, ..., x_k)$ der Gleichung $x_1 + ... + x_k = n$, so liegen diese geometrisch betrachtet alle auf der durch selbige Gleichung definierten Hyperebene.



(a) Beispiel für $k = 2, n = 5$



(b) Beispiel für $k = 3, n = 5$

Abbildung 1: Geometrische Veranschaulichung der unbeschränkten Partitionen

Dieses Bild legt nahe, dass sich die Anzahl an 3-dimensionalen Partitionen aus der Anzahl an 2-dimensionalen Partitionen zusammensetzt - also eine rekursive Beziehung zwischen ψ_3 und ψ_2 besteht. Setzen wir z.B. x_3 auf einen festen Wert, so folgt

$$x_1 + x_2 = n - x_3,$$

also ist (x_1, x_2) eine 2-Partition von $n - x_3$ wenn (x_1, x_2, x_3) eine 3-Partition von n ist. Da x_3 alle Werte in $\{0, ..., n\}$ annehmen kann ergibt sich somit

$$\psi_3(n) = \sum_{x_3=0}^{n} \psi_2(n - x_3) \underset{r:=n-x_3}{=} \sum_{r=0}^{n} \psi_2(r). \tag{1}$$

Für den ψ_2 finden wir mittels kombinatorischer Argumente (die Menge $\{0, ..., n\}$ hat $n + 1$ Elemente, wir wählen hieraus eines für x_2 aus und definieren somit eindeutig eine Partition. Es gibt $\binom{n+1}{1} = n+1$ Möglichkeiten diese Auswahl zu treffen.) $\psi_2(n) = n+1$. Mit Rückblick auf Gleichung 1 schreiben wir dies als

$$\psi_2(n) = n + 1 = \sum_{r=0}^{n} 1 = \sum_{r=0}^{n} \psi_1(n), \tag{2}$$

wobei wir $\psi_1(n) = 1$ als trivial wahr erkennen. Dies legt die generelle Formel

$$\psi_{k+1}(n) = \sum_{r=0}^{k} \psi_k(n), \qquad \text{mit } k \in \mathbb{N}, n \in \mathbb{N}_0 \tag{3}$$

nahe, von deren Gültigkeit wir uns auch in höheren Dimensionen leicht überzeugen (Setzen wir einen Freiheitsgrad auf $r \in \{0, ..., n\}$ fest so müssen die restlichen eine beliebige k-Partition von $n - r$ bilden.).

1.2 Geschlossene Form von $\psi_k(n)$

Berechnen wir nach obiger Formel $\psi_k(n)$ für einige k erhalten wir:

$$\begin{aligned}
\psi_1(n) &= \quad 1 \\
\psi_2(n) &= \quad (n+1) \\
\psi_3(n) &= \frac{1}{2}(n+1)(n+2)(n+3) \\
\psi_4(n) &= \frac{1}{6}(n+1)(n+2)(n+3)(n+4)
\end{aligned}$$

was die Vermutung

$$\psi_k(n) = \frac{1}{(k-1)!} \prod_{r=1}^{k-1} (n + r) \tag{4}$$

nahelegt. Da es nicht völlig trivial ist die Gültigkeit dieser Formel anhand Gleichung 3 zu beweisen, wollen wir sie mittels generierenden Funktionen herleiten. Sei daher

$$G_n(x) := \sum_{k=1}^{\infty} \psi_k(n) x^k$$

die Familie der gewöhnlichen generierenden Funktionen den Folgen $(\psi_k(n))_{k\in\mathbb{N}}$ für $n \in \mathbb{N}_0$. Multiplizieren wir Gleichung 3 beidseitig mit x^k und summieren über ihren Gültigkeitsbereich in k erhalten wir

$$\begin{gathered}
\frac{1}{x} \sum_{k=1}^{\infty} \psi_k(n) x^k - \psi_1(n) = \sum_{k=1}^{\infty} \sum_{r=0}^{n} x^k \psi_k(n) = \sum_{r=0}^{n} \sum_{k=1}^{\infty} x^k \psi_k(n) \\
\iff G_n(x) = x(\sum_{r=0}^{n} G_r(x) - \psi_1(n)).
\end{gathered}$$

Durch abspalten des letzten Terms der rechtsseitigen Summe und anschließendes Auflösen nach $G_n(x)$ erhalten wir

$$G_n(x) = \frac{x}{1-x}(\sum_{r=0}^{n-1} G_r(x) + \psi_1(n)). \tag{5}$$

Wir berechnen die G_n mittels dieser Formel und

$$G_0(x) \underset{\psi_1(n)=1}{=} \sum_{k=1}^{\infty} x^k = \frac{1}{1-x} - 1 = \frac{x}{1-x}$$

für einige n:

$$G_1(x) = \frac{x}{(1-x)^2}$$
$$G_2(x) = \frac{x}{(1-x)^3}$$
$$G_3(x) = \frac{x}{(1-x)^4}.$$

Es liegt nahe, dass

$$G_n(x) = \frac{x}{(1-x)^{n+1}}. \tag{6}$$

Sei $g(x) := \frac{1}{1-x}$. Nach der Cauchy-Produktformel gilt

$$g(x)^2 = \frac{1}{1-x} \cdot \frac{1}{1-x} = \sum_{k=0}^{\infty} \sum_{r=0}^{k} x^r x^{k-r} = \sum_{k=0}^{\infty} (k+1) x^k = Dg(x).$$

Somit gilt $D(g^n(x)) = ng^{n-1}(x)Dg(x) = ng^{n+1}(x)$ woraus wir

$$g^{n+1}(x) = \frac{1}{n} Dg^n(x) = \frac{1}{n(n-1)} D^2 g^{n-1}(x) = ... = \frac{1}{n!} D^n g(x)$$

$$\iff D^k g^n(x) = (n)^{\bar{k}} g^{n+k}(x)$$

erhalten, wobei $(n)^{\bar{k}}$ die steigende Faktorielle bezeichnet. Somit folgt für die Taylorentwicklung von g^n um 0

$$g^n(x) = \sum_{k=0}^{\infty} \frac{D^k g^n(0)}{k!} x^k = \sum_{k=0}^{\infty} \frac{(n)^{\bar{k}} g^{n+k}(0)}{k!} x^k = \sum_{k=0}^{\infty} \frac{(n)^{\bar{k}}}{k!} x^k,$$

womit wir für G_n folgende Reihendarstellung erhalten

$$G_n(x) = \sum_{k=0}^{\infty} \frac{(n+1)^{\bar{k}}}{k!} x^{k+1} = \sum_{k=1}^{\infty} \frac{(n+1)^{\overline{k-1}}}{(k-1)!} x^k.$$

Durch Koeffizientenvergleich mit der ursprünglichen Definition der generierenden Funktion erhalten wir die in Gleichung 4 bereits vermutete Darstellung

$$\psi_k(n) = \frac{(n+1)^{\overline{k-1}}}{(k-1)!} = \frac{1}{(k-1)!} \prod_{r=0}^{k-2} (n+1+r) = \frac{1}{(k-1)!} \prod_{r=1}^{k-1} (n+r). \tag{7}$$

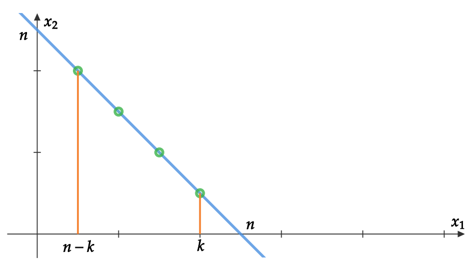
2 Beschränkte Partitionen

Wenden wir uns nun den beschränkten Partitionen zu.

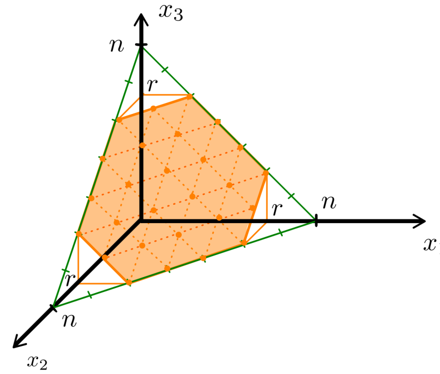
Notation 2 *Wir bezeichnen die Anzahl an r-beschränkten k-Partitionen mit* $\varphi_k^r(n)$.

2.1 Motivation

Wie auch zuvor können wir uns die beschränkten Partitionen geometrisch veranschaulichen.



(a) Beispiel für $k = 2, n = 5$



(b) Beispiel für $k = 3, n = 7$

Abbildung 2: Geometrische Veranschaulichung der beschränkten Partitionen

Wir nehmen sozusagen alle unbeschränkten Partitionen, und schneiden dann die kleineren Teile weg, bei denen einzelne Komponenten die Schranke r überschreiten würden.

2.2 Wahrscheinlichkeitsverteilung der beschränkten Partitionen

Diese Überlegungen können wir direkt in eine Gleichung übersetzen und erhalten somit

$$\varphi_k^{r-1}(n) = \psi_k(n) - k\psi_k(n-r).$$

Hierbei erklärt sich der Index $(r-1)$ dadurch, dass wir die Elemente der "Schnittgrenze" mitnehmen wollen. Nehmen wir nun an, dass alle Partitionen gleichwahrscheinlich sind, so erhalten wir ein Laplaceexperiment. Es sei $\Omega := \mathcal{P}_k^r(n)$ die Ergebnismenge, sowie $\omega := (X_1, ..., X_k) \in \Omega$ eine Zufällige Partition. Da $X_j = m$ genau dann wenn die restlichen Komponenten eine $(k-1)$-Partition bilden (also $(X_1, ..., X_{j-1}, X_{j+1}, ...X_k) \in \mathcal{P}_k^r(n-m)$), gibt es φ_{k-1}^r

Möglichkeiten für die Konfiguration von ω. Es gilt demnach

$$p_k^r(n,m) := P(X_j = m) = \frac{\varphi_{k-1}^r(n-m)}{\varphi_k^r(n)}. \tag{8}$$

2.3 Effiziente Bestimmung von $P(X_j = m)$

Da in 7 und somit auch in 8 viele Fakultäten vorkommen ist diese Darstellung nicht praktikabel umsetzbar. Wir beginnen damit die Situation erstmal noch etwas schlimmer zu machen indem wir die fallende Faktorielle mittels Fakultäten ausdrücken und erhalten

$$\varphi_k^{r-1}(n) = \frac{(n+k-1)!}{n!(k-1)!} - k\frac{(n-r+k-2)!}{(n-(r+1))!(k-1)!} = \frac{\Gamma(n+k)}{n!\Gamma(k)} - k\frac{\Gamma(n-r+k-1)}{\Gamma(n-r)\Gamma(k)}$$

Wenden wir auf die beiden Terme jeweils $\exp \circ \ln$ an, ergibt sich mit $\ln F := \ln \circ (\cdot !)$

$$\begin{aligned}
\varphi_k^{r-1}(n) &= \exp(\ln \psi_k(n)) - \exp(\ln(k\psi_k(n-r))) \\
&= \exp(\ln\Gamma(n+k) - \ln F(n) - \ln\Gamma(k)) \\
&\quad - \exp(\ln(k) + \ln\Gamma(n+k-(r+1)) - \ln\Gamma(n-r) - \ln\Gamma(k)).
\end{aligned}$$

Die Funktion $\ln\Gamma$ lässt sich über eine asymptotische Entwicklung auf Basis der verallgemeinerten Stirlingformel

$$\Gamma(x) = \sqrt{\frac{2\pi}{x}}\left(\frac{x}{e}\right)^x e^{\mu(x)} \text{ mit } x > 0, 0 < \mu(x) < \frac{1}{12x}$$

gut und effizient approximieren, es gilt

$$\ln\Gamma(x) = (x - \frac{1}{2})\ln x - x + \frac{1}{2}\ln 2\pi + \frac{1}{12x} + \frac{1}{360x^3} + \mathcal{O}(x^{-4}).$$

Da selbst eine Taylorentwicklung höherer Ordnung von $f : (x,y) \mapsto e^x - e^y$ (z.B. $T_2 f(x,y,u,v) = e^u(1 + x - u + \frac{(x-u)^2}{2}) - e^v(1 + y - v + \frac{(y-v)^2}{2})$ mit $u, v = \frac{|a-b|}{2}$: $T_2 f(x,y) = e^{\frac{|b-a|}{2}}(a-b)(1 + \max\{a,b\})$), uns für ausreichend große Argumente in praktisch unbrauchbaren Größenordnungen lässt, ersetzen wir exp in diesen Ausdrücken durch die Taylorentwicklung erster Ordnung und erhalten somit

$$\begin{aligned}
\varphi_k^{r-1}(n) &\doteq \ln\Gamma(n+k) - \ln F(n) - \ln\Gamma(k) \\
&\quad - \ln k + \ln\Gamma(n+k-(r+1)) - \ln\Gamma(n-r) - \ln\Gamma(k)
\end{aligned}$$

womit wir $P(X_j = m)$ effizient berechnen können. Hiermit können wir uns eine Zufallsverteilung generieren[1] und auf Basis dieser Zufallszahlen erzeugen.

[1] Sofern es zu teuer ist die ganze Verteilung zu erzeugen könnte man ggf. durch lückenweises Sampling und anschließende Renormalisierung einzelne Zahlen erhalten. Man erzeuge also für $\nu \in \mathbb{N}$ zunächst $M := \{(\nu m, p_k^r(n, \nu m)) | 0 \leq \nu m \leq n\}$, $\mathrm{N} := \{p | (\_, p) \in M\}$ und hieraus die Wahrscheinlichkeitsverteilung $\{(m, \frac{p}{\sum_{\rho \in N} \rho}) | (m,p) \in M\}$. Ggf. kann man nach Erzeugen einer Zahl iterativ mit einem kleineren ν eine Auswahl auf dem durch die erste Zufallszahl bestimmten Teilintervall durchführen.

2.4 Implementierung

Sofern man wirklich formal korrekt gleichverteilte vollständige Partitionen erzeugen will müsste man in jedem Schritt j als Verteilung die Wahrscheinlichkeiten $P(X_j = m | X_{j-1} = x_{j-1}, ..., X_1 = x_1), 0 \leq m \leq r$ nutzen.
Der tatsächlich implementierte Algorithmus geht wie folgt vor

```
Data: n, m, k ∈ ℕ
Result: zufällige Partition (x_1, ..., x_m) von n sodass x_j <= k
Gewichte ← Feld für n + 1 natürliche Zahlen auf 0 initialisiert
for i ← 0 to min{k, n} do
    Gewicht(i) ← p_m^k(Rest, i)
end
do
    Partition = (x_1, ..., x_m) ← Feld für m natürliche Zahlen auf 0
      initialisiert
    Rest ← n
    for i ← 1 to m − 1 do
        while True do
            r ← sample (Gewichte) #Generiere Zufallszahl nach
              gegebener Verteilung
            if r ≤ Rest then
                Rest ← max{0, Rest −r}
                x_i ← r
                break
            end
        end
    end
    x_m ← Rest
while max(Partition) ≤ k #Die generierte Partition ist gültig;
return Partition
```