

Repetitive experiments: probability and frequency

← Back to Chapters

Formulas 9.19 to 9.22

We are trying to solve a recurrence relation 9.19

$$M(n, G) = \sum_{j=1}^m M(n-1, G-g_j)$$

subject to the boundary condition

$$M(0, G) = \delta(G, 0).$$

Divinely inspired (or, rather, inspired by the fact that this is a kind of discretized version of a linear first order partial differential equation; but see below for more comments on this) we pretend that there exists some $h(\lambda)$ such that $M(0, -G)$ is a Laplace transform of $h(\lambda)$, i.e.

$$\delta(G, 0) = M(0, G) = \int d\lambda \exp\{\lambda G\} h(\lambda)$$

Of course this prompts the question of whether such an h actually exists. We ignore this for now (but see below).

Then, we compute $M(n, G)$ by induction on n . Let's start with $n = 1$ from $n = 0$:

$$\begin{aligned} M(1, G) &= \sum_{j=1}^m M(0, G-g_j) = \\ &= \sum_{j=1}^m \int d\lambda \exp\{\lambda(G-g_j)\} h(\lambda) = \\ &= \int d\lambda \left(\sum_{j=1}^m \exp\{-\lambda g_j\} \right) \exp\{\lambda(G-g_i)\} h(\lambda) = \\ &= \int d\lambda Z(\lambda) \exp\{\lambda G\} h(\lambda) \end{aligned}$$

where $Z(\lambda) = \sum_{j=1}^m \exp\{-\lambda g_j\}$.

Of course we then have in the same way

$$M(n, G) = \int d\lambda Z^n(\lambda) \exp\{\lambda G\} h(\lambda)$$

aka 9.22.

Remark 1, inspiration: This is an idea as old as Fourier. IF the initial condition was $M(0, G) = \exp\{\lambda_0 G\}$ for some fixed λ then the fact that for such exponential functions shifting input G to $G - g_j$ simply multiplies the result by $\exp\{-\lambda_0 g_j\}$, we would have by induction $M(n, G) = Z^n(\lambda_0) M(0, G)$. In other words, the function $\exp\{\lambda_0 G\}$ is an eigenfunction of the operator $M(\cdot) \rightarrow \sum_{j=1}^m M(\cdot - g_j)$ with eigenvalue $Z(\lambda_0)$, so repeated application of this operator to increase n just keeps multiplying the function by $Z(\lambda_0)$. This is the content of the formulas 9.20 and 9.21: setting $n = 0$ in 9.20 gives the boundary value $M(0, G) = \exp\{\lambda_0 G\}$, and then the eigenvalue $\exp\{\alpha\}$ has to be equal to $Z(\lambda_0)$.

Now the equation 9.19 is linear. So if we can make the initial condition $M(0, n)$ a sum of exponentials, the solution is the sum of corresponding solutions. This is what we did above.

This is analogous to how any exponential function $\exp\{\lambda_0 x\}$ being boundary condition to heat equation (at $t = 0$) leads to $F(t, x) = \exp\{\lambda_0^2 t\} \exp\{\lambda_0 x\}$ solution. In that setting, if one can write $F(0, x)$ as sum of exponentials then the solution follows. The heat equation is second order, so one *can* use sines and cosines as boundary conditions instead, and then decompose the actual boundary values into those (aka Fouries analysis). We have a first order difference equation, so we use exponentials.

Remark 2, finding h : I don't think there is such an $h(\lambda)$ (but have not proved this).

Remark 3: I don't think any of this matters. As far as I can tell, we are only going to be using 9.27, and not 9.22. That said, Jaynes's derivation of 9.27 is suspect (and the talk of "inverse Laplace transform" seems doubly suspect; this would be right if we were doing Fourier transforms, but not for Laplace). So we better derive 9.27. We do this after an unnecessary detour through the 1D linear recurrences.

Bonus: 1D case for no good reason (maybe for motivation). The 9.19 is a 2-variable version of recurrence relation, more particularly of a constant-recursive sequence,

$$M(n) = \sum_{k=1}^d c_k M(n - k)$$

or putting $c_0 = -1$:

$$\sum_{k=0}^d c_k M(n - k) = 0$$

The answer is that if $P(x) = \sum_{k=0}^d c_k x^k$ has **distinct** roots μ_j , then $M(n)$ has the form $\sum_{j=0}^d k_j \mu_j^n$. This is perhaps reminiscent of 9.20 (bearing in mind $\mu_j^n = \exp\{\alpha_j n\}$ where $\alpha_j = \log \mu_j$, with the caveat that μ_j are complex and could be 0).

One common approach to such equations is to study a *generating function* of the $M(n)$, i.e.

$$f_0(x) := \sum_{n \geq 0} M(n) x^n$$

One sees that – modulo the edge effects – shifting the sequence to $M(n-k)$ just multiplies f_0 by x^k :

$$f_k(x) := \sum_{n \geq k} M(n-k) x^n = \sum_{n-k \geq 0} M(n-k) x^{n-k} x^k = x^k f_0(x)$$

Then multiplying this by c_k and summing over $k = 1, \dots, d$ one sees that all the terms of degree $\geq d$ cancel, so that

$$\sum_{k=0}^d c_k x^k f_0(x) = Q(x)$$

where $Q(x)$ is a polynomial of degree at most $d-1$ consisting of all the terms that don't cancel. Note that the $Q(x)$ will vary depending on the initial values $M(0), M(1), \dots, M(d-1)$. Then

$$f_0(x) = \frac{Q(x)}{\sum_{k=0}^d c_k x^k}$$

Then one can factorize the polynomial $P(x) = \sum_{k=0}^d c_k x^k$ and use partial fraction expansion of $\frac{Q(x)}{P(x)}$ to get a formula for the coefficients $M(n)$.

We want to make the following remarks about this procedure:

- 1) Upon substitution $x = \frac{1}{z}$ the generating function becomes Z-transform, a discrete time analog of Laplace transform (think $z = \exp(s)$ and $\sum_{n \geq 0} M(n) z^{-n}$ as $\int_{t \geq 0} M(t) (\exp s)^{-t}$). The “shifting the signal multiplies the transform” property that we used is the analogue of the corresponding property of Laplace transform. In fact the very recurrence we are solving is a discrete time analogue of an ODE. So we are using the “Laplace transform method for ODEs”, just in discrete time.
- 2) An advantage of working with generating functions is that they are *formal power series*, meaning we don't worry about convergence.
- 3) From this perspective what we did is instead of solving for $M(n)$ directly, we solved for its Z/Laplace transform and then deduced what $M(n)$ must be.

Getting 9.23 (and then 9.27).

The main point is to get 9.23. That is:

$$\sum_G \exp\{-\lambda G\} M(n, G) = \sum_{(n_1, \dots, n_m)} W(n_1, \dots, n_m) \exp\{-\lambda \sum n_j g_j\}$$

with the sum over tuples of non-negative integers (n_1, \dots, n_m) with $\sum n_j = n$ and

$$W(n_1, \dots, n_m) = \frac{n!}{n_1! \cdot \dots \cdot n_m!}.$$

This can be proven by induction on n ; we will do so below. But right now, we argue why it's true directly: both sides simply sum over all possible sequences $\vec{o} = (o_1, o_2, \dots, o_n)$ of length n of outcomes $o_t \in \{1, \dots, m\}$ the value

$$V(\vec{o}) = \exp\{-\lambda \sum_{t=1}^n g_{o_t}\}.$$

The left hand side groups the outcome sequences \vec{o} by their “total amount generated” i.e. by $G = \sum g_{o_t}$. The number of sequences with this G is by definition $M(n, G)$, and each one contributes $\exp\{-\lambda \sum_{t=1}^n g_{o_t}\}$, hence the left hand side.

The right hand side groups the outcome sequences by frequencies n_j of the outcome $j \in \{1, \dots, m\}$. For a fixed tuple (n_1, \dots, n_m) of frequencies there are $W(n_1, \dots, n_m)$ sequences \vec{o} , each generating $\exp\{-\lambda \sum n_j g_j\}$, hence the right hand side.

Thus 9.23 follows. The passage from 9.23 to 9.27 is reasonably well explained in Jaynes.

Bonus: induction proof of 9.23 Base: The $n = 0$ case is simply $1 = 1$, which is true.

Step: Using 9.19 (i.e. $M(n, G) = \sum_{j=1}^m M(n-1, G-g_j)$) we have:

$$\sum_G \exp\{-\lambda G\} M(n, G) = \sum_G \exp\{-\lambda G\} \left(\sum_{j=1}^m M(n-1, G-g_j) \right)$$

We massage this a bit to make it fit the formulas we have in the induction hypothesis (note that we use that all the sums are over finite sets to exchange order of summation as we see fit):

$$\begin{aligned}
&= \sum_G \sum_{j=1}^m \exp\{-\lambda g_j\} \exp\{-\lambda(G - g_j)\} M(n-1, G - g_j) \\
&= \sum_{j=1}^m \exp\{-\lambda g_j\} \sum_G \exp\{-\lambda(G - g_j)\} M(n-1, G - g_j)
\end{aligned}$$

Now by induction hypothesis (we don't say this separately, but the summands where one of the elements of the tuple becomes negative are excluded from the corresponding sum):

$$\begin{aligned}
&= \sum_{j=1}^m \exp\{-\lambda g_j\} \sum_{(n_1, \dots, n_j-1, \dots, n_m)} W(n_1, \dots, n_j-1, \dots, n_m) \exp\{-\lambda(\sum n_j g_j - g_j)\} \\
&= \sum_{j=1}^m \sum_{(n_1, \dots, n_j-1, \dots, n_m)} W(n_1, \dots, n_j-1, \dots, n_m) \exp\{-\lambda(\sum n_j g_j)\}
\end{aligned}$$

Now since $W(n_1, \dots, n_j-1, \dots, n_m) = \frac{n_j}{n} W(n_1, \dots, n_j, \dots, n_m)$ we get:

$$\begin{aligned}
&= \sum_{j=1}^m \sum_{(n_1, \dots, n_j-1, \dots, n_m)} \frac{n_j}{n} W(n_1, \dots, n_j, \dots, n_m) \exp\{-\lambda(\sum n_j g_j)\} \\
&= \sum_{(n_1, \dots, n_j, \dots, n_m)} W(n_1, \dots, n_j, \dots, n_m) \exp\{-\lambda(\sum n_j g_j)\}
\end{aligned}$$

A version of Exercise 9.1

For more on the “missing species” problem see Wikipedia, the paper of Fisher and Section 6.2 of the Computer Age Statistical Inference by Efron and Hastie.

We follow the above references answer a different, albeit related, question: namely, how many new species we may hope to find if we sample more i.e. collect more specimens.

According to Fisher,

$$S = -\alpha \ln(1-x), \quad N = \alpha \frac{x}{1-x}$$

where α is some kind of “average Poisson rate” parameter and is independent of size of sample, while $x = \frac{\sigma}{\sigma+1}$ where $\sigma = \frac{x}{1-x}$ the scale parameter in the prior Gamma distribution over rates of Poissons, and grows linearly with sample size/sample time.

Then

$$S = \alpha \ln(1 + \sigma), N = \alpha \sigma, \frac{S}{N} = \frac{\ln(1+\sigma)}{\sigma}.$$

For $N = 122$, $S = 19$ this gives $\frac{\ln(1+t)}{t} = \frac{19}{122}$

and

$$\sigma = 19.34583762707454276246963164708576325676864924349274639666$$

This can be expected to be reasonably accurate up to about doubling the sample size/ σ .

After double the sample size i.e. doubling σ , the expected number of species S becomes

$$S_{new} = S_{init} \frac{\ln(1 + 2\sigma)}{\ln(1 + \sigma)} = 23.21425768843744049107893270266974263904162495024301691901$$

So after collecting 122 more samples we expect to discover about 4 new species.

Exercise 9.2

Presumably we are to rederive formula 3.89 (and no 3.77).

Given k linear functions $\mathcal{G}_1, \dots, \mathcal{G}_k$ of n_j with $\mathcal{G}_l(n_1, \dots, n_m) = \sum g_j^l n_j$ we introduce $M(n, G_1, \dots, G_k)$ as the number of tuples $\vec{n} = (n_1, \dots, n_m)$ such that $\mathcal{G}_l(\vec{n}) = G_l$.

Recursion 9.19 becomes

$$M(n, G_1, \dots, G_k) = \sum_{j=1}^m M(n - 1, G_1 - g_j^1, \dots, G_k - g_j^k)$$

Ansatz 9.20 becomes

$$M(n, G_1, \dots, G_k) = \exp\{\alpha n + \sum_l \lambda_l G_l\}$$

and condition 9.21

$$\exp\{\alpha\} = Z(\lambda_1, \dots, \lambda_k) = \sum_{j=1}^m \exp\left\{\sum_{l=1}^k -\lambda_l g_j^l\right\}$$

To shorten notation we write $\vec{G} = (G_1, \dots, G_k)$ and $\vec{\lambda} = (\lambda_1, \dots, \lambda_k)$. Then general solution 9.22 becomes

$$H(n, \vec{G}) = \int d\vec{\lambda} \ Z^n(\vec{\lambda}) \exp\{\vec{\lambda} \cdot \vec{G}\} h(\vec{\lambda})$$

Then with initial condition $M(0, \vec{G}) = \delta(\vec{G}, \vec{0})$ we have as before

$$Z^n(\vec{\lambda}) = \sum_{\vec{G}} M(n, \vec{G}) \exp\{-\vec{\lambda} \cdot \vec{G}\}$$

Now we apply this to derive multinomial distribution. Partition outcomes n_j into k sets S_1, \dots, S_k of size s_1, \dots, s_k and let $\mathcal{G}_l(n_j) = \begin{cases} 1 & \text{if } n_j \in S_l \\ 0 & \text{else} \end{cases}$

Then $M(n, c_1, \dots, c_k)$ is the number of ways to get a tuple \vec{n} with $\sum_{j \in S_l} n_j = c_l$ and we can compute

$$\begin{aligned} Z(\vec{\lambda}) &= \sum_l s_l \exp\{-\lambda_l\} = \sum_l s_l x_l \\ Z^n(\vec{\lambda}) &= \sum_{(c_1, \dots, c_k)} \binom{n}{c_1, \dots, c_k} \prod_l (s_l x_l)^{c_l} \end{aligned}$$

and so

$$M(n, c_1, \dots, c_k) = \binom{n}{c_1, \dots, c_k} \prod_l s_l^{c_l}$$

Probability of getting sequence of this type is $\frac{M(n, c_1, \dots, c_k)}{n^n}$ and denoting $\frac{s_l}{n} = f_l$ we have

$$P(c_1, \dots, c_k) = \binom{n}{c_1, \dots, c_k} \prod_l f_l^{c_l}$$

in agreement with 3.89.

Comments on Section 9.7

“If A is linear in the n_j , then it is the same as our G in (9.17)” – I think Jaynes means that “for a given linear function \mathcal{G} of n_j s, let A be the proposition ‘ \mathcal{G} takes value G ’”. In that case, indeed, $M(n, G) = M(n, A)$ by definition.

“the notion of entropy inherent in probability theory independently of the work of Shannon” – in the work of Shannon, the notion of entropy is derived from combinatorics of sequences, much like here.

Exercise 9.3 (the first one)

$$f_k'' = \frac{n_k - \delta_{jk} - \delta_{tk}}{n - 2}$$

$$\delta f_k = f_k'' - f_k = f_k'' - \frac{n_k}{n} = \frac{2f_k - \delta_{jk} - \delta_{tk}}{n - 2}$$

so

$$\delta H = \sum_k (-1 - \ln(f_k)) \frac{2f_k - \delta_{jk} - \delta_{tk}}{n - 2}$$

$$\frac{2H + \ln(f_j) + \ln(f_t)}{n - 2}$$

and

$$H'' = \frac{nH + \ln(f_j) + \ln(f_t)}{n - 2}$$

$$M(n - 2, G - g_j - g_t) = \exp\{(n - 2)H''\} = f_j f_t \exp\{nH\}$$

so

$$p(r_i = j, r_s = t | GnI_0) = f_j f_t$$

The trials are still pairwise independent (and by similar analysis l -wise independent as long as $k \ll n$).

Formula 9.78

Assuming not all g_j are the same, strict convexity of $\ln Z$ implies that the derivative $-\frac{\partial \ln Z}{\partial \lambda}$ attains each value in its range once. However, it only attains values $\min g_j < \lambda < \max g_j$. Averages \bar{G} outside this range - with exception of the endpoints $\min g_j$ and $\max g_j$ - are impossible (the endpoints correspond to $\lambda = \pm\infty$ and would require those u_i with non-extremal g_i to be set to zero, while not restricting the others; this is a rather special case, not directly covered by Jaynes' analysis in this section). We better condition on an achievable average.