Continuous MaxEnt Distributions in *Mathematica*: a “Parameter-Free” Approach

Barrie James Stokes

Discipline of Clinical Pharmacology
Faculty of Health
University of Newcastle, NSW, Australia
barrie.stokes@newcastle.edu.au

Abstract. *Mathematica* is used to develop a method of obtaining continuous MaxEnt distributions using the Lagrange Multiplier method that works directly in the discrete case. The continuous PDF (probability density function) is described by list of coordinates of the form $\{\{abs_1, ord_1\}, \{abs_2, ord_2\}, \dots, \{abs_n, ord_n\}\}$, where the abscissae abs_i are specified numerically so as to define the domain of the PDF, and the ordinates ord_i are initially general symbolic variables that are assigned numeric values via the procedure to be described. In all the applications given here, $n=51$. A set of Lagrange Multiplier equations is solved for the ord_i , making use of the fact that the *Mathematica* function `Interpolation[]`, which constructs *interpolationFunction* objects normally used for numeric data and for describing numerical solutions to differential equations, can operate on “semi-symbolic” data.

INTRODUCTION

The original problem addressed was to determine a PDF on a known domain having three prescribed properties, a given mean, and given 2.5% and 97.5% points, i.e., fitting a Frequentist description of an uncertain quantity (a hazard ratio arising from a survival study) given as $\{\text{mean}, 95\%CI\} = \{0.491, \{0.290, 0.831\}\}$.

The first attempt sought a Beta PDF complying with the three constraints. When this proved impossible, a ‘5-parameter’ Beta distribution was defined in *Mathematica*. This is a standard Beta, shifted and scaled to have domain $\{k, m\}$ and incorporating a power transformation of the independent variable with parameter λ . A solution was obtained using the *Mathematica* procedure `FindMinimum[]`, as shown in Section 1.

Later it was realised that the resulting PDF, although elegant in appearance, might not be the MaxEnt solution, constrained as it is to belong to the Beta family. It is straightforward to find a MaxEnt solution to a discrete problem in *Mathematica*, as for example Jaynes’ famous mean=4.5 die problem, using the Lagrange Multiplier method, and is implemented in Section 2.

The idea that a similar approach could be applied to a continuous MaxEnt problem was then explored. It was found that the facilities offered by the *Mathematica* function `Interpolation[]`, in particular the ability to apply it to data combining numeric abscissae with symbolic ordinates, provided a means of formulating the Lagrange Multiplier method for continuous distributions.

Section 3 shows the result of applying this approach to finding the MaxEnt PDF with the normalisation and a variance constraint. The result is gratifyingly familiar.

The internal *Mathematica* representation of the 53 equations for the 51 ordinates and the 2 multipliers, which are solved simultaneously using the *Mathematica* procedure **FindRoot[]**, has over 2,000,000 elements. *Mathematica* 7.0.0 solved these equations to 20 significant figures in around 13 seconds on a 2 GHz Asus W2 laptop running Windows XP. For reassurance, the solution is compared graphically with the known analytic solution.

In Section 4 the approach is applied to the original {0.491, {0.290, 0.831}} problem and the entropies of the ‘5-parameter’ Beta and the ‘51-point’ solutions are compared.

1. THE ‘5-PARAMETER’ BETA SOLUTION

The parameters of the ‘5-parameter’ Beta PDF defined above can be determined under the constraint that the mean is 0.491, and the 2.5% and 97.5% percentiles are 0.290 and 0.831 respectively. The *Mathematica* function **FindMinimum[]** is used with an objective function which is the square root of the sum of 3 quadratic error terms.

In *Mathematica* the ‘5-parameter’ Beta PDF was defined via the following input statement:

$$\text{pdfBetaExtended}[\alpha_?\text{NumericQ}, \beta_?\text{NumericQ}, k_?\text{NumericQ}, m_?\text{NumericQ}, \lambda_?\text{NumericQ}, x_?\text{NumericQ}] := \frac{\text{PDF}\left[\text{BetaDistribution}[\alpha, \beta], \left(\frac{x-k}{m-k}\right)^\lambda\right]}{\frac{(m-k) \text{Beta}\left[1, \alpha + \frac{1}{\lambda} - 1, \beta\right]}{\lambda \text{Beta}[\alpha, \beta]}}$$

Figure 1 illustrates the solution found by **FindMinimum[]**.

FIGURE 1. The ‘5-parameter’ Beta distribution solution for the “0.490 (0.290, 0.831)” problem.

2. JAYNES’ ORIGINAL ANALYSIS OF A BIASED DIE

The maximum entropy problem discussed in [1] asks for the 6 maximally noncommittal probabilities for the faces of a die, given that the sum of the probabilities is 1, and that

the long term average score of the die in question is not 3.5 (as it is for a die with $p_1=p_2=p_3=p_4=p_5=p_6=\frac{1}{6}$), but 4.5.

The MaxEnt approach to this problem, according to which the maximally noncommittal assignment of prior probabilities is found by maximising the entropy of the discrete distribution, subject to the known constraint(s), is implemented in *Mathematica*. In order to employ the Lagrange Multiplier method to apply MaxEnt in this case, we first define the entropy of the die face probabilities as the objective function using the *Mathematica* command

$$\text{objective} = - \sum_{i=1}^6 p_i \text{Log}[p_i]$$

The following code prepares the problem, and finds the MaxEnt probabilities via **FindRoot[]**:

```
constraint1 = p1 + p2 + p3 + p4 + p5 + p6 - 1;
(* the normalisation constraint *)
constraint2 = 1 p1 + 2 p2 + 3 p3 + 4 p4 + 5 p5 + 6 p6 - 4.5;
(* the mean constraint *)
g = objective + λ1 constraint1 + λ2 constraint2;
(* the Lagrange multiplier function *)
equations1 = Table[∂pig == 0, {i, 1, 6}]; (* differentiate wrt the pi *)
equations2 = Table[∂λng == 0, {n, 1, 2}]; (* differentiate wrt the λi *)
equations = Join[equations1, equations2]; (* combine the equations *)
soln1 = FindRoot[equations, {p1, 0.5}, {p2, 0.5}, {p3, 0.5}, {p4, 0.5},
  {p5, 0.5}, {p6, 0.5}, {λ1, 0.5}, {λ2, 0.5}]
```

These commands yield the following *Mathematica* output, giving the solution in the form of a list of replacement rules:

$$\{p_1 \rightarrow 0.0543532, p_2 \rightarrow 0.0787715, p_3 \rightarrow 0.11416, p_4 \rightarrow 0.165447, \\ p_5 \rightarrow 0.239774, p_6 \rightarrow 0.347494, \lambda_1 \rightarrow -2.2833, \lambda_2 \rightarrow 0.371049\}.$$

This is exactly the solution given by Jaynes in [1], shown in Figure 2. Its entropy is 1.61358. The maximum possible entropy in this problem is that of the uniform distribution, $p_i = 1/6$, with the well-known value $\text{Log}[6]$, or 1.79176.

FIGURE 2. MaxEnt solution of biased die problem (entropy = 1.61358).

3. CONTINUOUS MAXENT PROBLEM WITH A VARIANCE CONSTRAINT

The approach is tested on the problem where there is a single variance constraint. The domain of our MaxEnt PDF is defined as $\{-4, 4\}$, and a 51-point solution is constructed.

```
xMin, xMax = {-4, 4}; nPoints = 51;
(* set domain and number of points *)

abscissae = Table[ x, {x, xMin, xMax,  $\frac{xMax - xMin}{nPoints - 1}$  } ];
(* construct abscissae list (numeric) *)

ordinateNames = Table[ ordi, {i, 1, nPoints} ];
(* construct ordinate list (symbolic) *)

data = Transpose[{abscissae, ordinateNames}];
(* construct data list of coordinates *)

data//Short
```

In the following shortened form of output, $\langle\langle 46 \rangle\rangle$ denotes 46 omitted terms:

$$\left\{ \{-4, \text{ord}_1\}, \left\{-\frac{96}{25}, \text{ord}_2\right\}, \left\{-\frac{92}{25}, \text{ord}_3\right\}, \langle\langle 46 \rangle\rangle, \left\{\frac{96}{25}, \text{ord}_{50}\right\}, \{4, \text{ord}_{51}\} \right\}.$$

Constructing an *InterpolatingFunction* object with the "semi-symbolic" data above as input

```
f = Interpolation[data, InterpolationOrder  $\rightarrow$  1];
(* create linear InterpolatingFunction object *)
```

a general expression for the area under the PDF by direct integration of $f[]$ is obtained via:

```
symbolicArea = Integrate[f[x], {x, xMin, xMax}]/Simplify;
(* symbolic expression for the area under the PDF *)
symbolicArea//Short
```

The following shortened form of the output omits 43 terms:

$$\frac{2}{25} (\text{ord}_1 + 2\text{ord}_2 + 2\text{ord}_3 + 2\text{ord}_4 + \langle\langle 43 \rangle\rangle + 2\text{ord}_{48} + 2\text{ord}_{49} + 2\text{ord}_{50} + \text{ord}_{51})$$

Next, define a symbolic representation via another *InterpolatingFunction* object of the function whose integral gives a symbolic expression for the mean of the required PDF using the commands:

```
meanF = Interpolation[
  Transpose [ { {abscissae, Table [ x × f[x], { x, xMin, xMax,
    xMax - xMin
    nPoints - 1 } ] } } ],
  InterpolationOrder → 1];
symbolicMean = Integrate[meanF[x], {x, xMin, xMax}]/Simplify;
(* symbolic expression for the mean of the PDF *)
symbolicMean//Short
```

The final integration step yields:

$$-\frac{8}{625} (25\text{ord}_1 + 48\text{ord}_2 + 46\text{ord}_3 + 44\text{ord}_4 + \langle\langle 71 \rangle\rangle).$$

The same procedure, via a further *InterpolatingFunction* object, gives a symbolic expression for the variance of the required PDF:

```
varianceF =
  Interpolation[Transpose [ { {abscissae, Table [(x - symbolicMean)2 ×
    f[x], { x, xMin, xMax,
    xMax - xMin
    nPoints - 1 } ] } } ], InterpolationOrder → 1];
symbolicVariance = Integrate[varianceF[x], {x, xMin, xMax}];
(* symbolic expression for the variance of the PDF *)
symbolicVariance//Short
```

This shortened output indicates the complexity of the resulting general expression:

$$\begin{aligned} & \frac{4}{25} \left(\frac{1}{2} \left(\text{ord}_1 \left(-4 + \frac{8}{625} (25 \text{ord}_1 + \langle\langle 74 \rangle\rangle) \right) \right)^2 \right. \\ & - \text{ord}_2 \left(-\frac{96}{25} + \frac{8}{625} (25 \langle\langle 3 \rangle\rangle_1 + \langle\langle 74 \rangle\rangle) \right)^2 \left. + \text{ord}_2 \left(-\frac{96}{25} + \frac{8}{\langle\langle 3 \rangle\rangle} \langle\langle 1 \rangle\rangle \right)^2 \right) \\ & + \langle\langle 48 \rangle\rangle + \frac{4}{25} (\text{ord}_{50} (\langle\langle 1 \rangle\rangle \langle\langle 1 \rangle\rangle^2 + \langle\langle 1 \rangle\rangle \langle\langle 1 \rangle\rangle)). \end{aligned}$$

Finally, define the symbolic function whose integral is the entropy of the PDF. The code block:

```
entropyF = Interpolation[Transpose[
  { abscissae, Table [ -f[x] × Log[f[x]], { x, xMin, xMax,  $\frac{xMax - xMin}{nPoints - 1}$  } ] } ],
  InterpolationOrder → 1];
symbolicEntropy = Integrate[entropyF[x], {x, xMin, xMax}];
(* symbolic expression for the entropy of the PDF *)
symbolicEntropy//Short
```

produces

$$\begin{aligned} & \frac{4}{25} \left(-\text{Log}[\text{ord}_2] \text{ord}_2 + \frac{1}{2} (-\text{Log}[\text{ord}_1] \text{ord}_1 + \text{Log}[\text{ord}_2] \text{ord}_2) \right) + \langle\langle 48 \rangle\rangle \\ & + \frac{4}{25} (\langle\langle 1 \rangle\rangle). \end{aligned}$$

When all the preparatory steps for the Lagrange multiplier process are combined with the following constraints and Lagrange multiplier function

```
constraint1 = symbolicArea - 1; constraint2 = symbolicVariance - 1;
g = symbolicEntropy + Total[Table[ $\lambda_i$ constrainti, {i, 1, 2}]];
```

proceeding as above in the discrete case—with WorkingPrecision set to 20—gives the solution for the 51 ordinates (shown in shortened form) as:

```
{ord1 → 0.00013495732087505838921, ord2 → 0.00025251300676883008242,
  ⟨⟨48⟩⟩, ord51 → 0.00013495732087505838921}.
```

It is well known that the exact solution on $\{-\infty, \infty\}$ is the Gaussian $N[0, 1]$. The theoretical solution on $\{-4, 4\}$ is taken as the truncated Gaussian with both the standard deviation and normalising constant adjusted to give unit area and variance. The “parameter-free” MaxEnt solution and its error are plotted in Figure 3.

FIGURE 3. MaxEnt problem with a variance constraint. Interpolated MaxEnt solution with solution ordinates (left) , MaxEnt solution error (right).

Executing the input

```
solutionEntropy = Integrate[entropyF[x]/.replaceRuleSolution, {x, xMin, xMax}]
```

gives the MaxEnt solution entropy as 1.4188726118530280474. The entropy of the truncated Gaussian on $\{-4, 4\}$ with unit area and variance, evaluated by direct integration with WorkingPrecision set to 20, is 1.4188748996793988384, so the percentage error of the “parameter-free” entropy is -0.000161242%. Expressed as log relative error, the number of correct decimal digits is 5.79.

Similar “parameter-free” MaxEnt PDFs have also been obtained for the general Uniform distribution on $\{a, b\}$ using only the normalisation constraint, the general truncated Exponential on $\{a, b\}$ using a non-central mean constraint, i.e., $E[x] \neq \frac{a+b}{2}$ in addition to the normalisation constraint, and the Gamma distribution on $\{0, b\}$ using $E[x]$ and $E[\ln(x)]$ constraints in addition to the normalisation constraint.

4. CONTINUOUS MAXENT PROBLEM WITH A MEAN CONSTRAINT AND TWO TAIL AREA CONSTRAINTS

Now the approach is tested on the original problem where the mean is constrained to be 0.491, the 2.5% point is 0.290, and the 97.5% point is 0.831. The domain for the MaxEnt PDF is defined as $\{0, 1.5\}$, and again a 51-point solution is required.

```
{xMin,xMax} = {0.0,1.5}; nPoints = 51;
(* set domain and number of points *)
```

Constructing the appropriate interpolation data using these values

```
abscissae = Table [x, {x, xMin, xMax,  $\frac{xMax - xMin}{nPoints - 1}$  } ] ;
(* construct abscissae list (numeric) *)
data = Transpose[{abscissae, ordinateNames}];
(* construct data list of coordinates *)
data//Short
```

produces the following output list:

```
{ {0., ord1 }, {0.03, ord2 }, {0.06, ord3 }, {<46>}, {1.47, ord50 }, {1.5, ord51 } }.
```

The required *InterpolationFunction* objects, and symbolic expressions for the area under the PDF and for the mean are generated as before. The constraint involving the 2.5% point requires a new symbolic expression, constructed via

```
symbolicLeftTail = Integrate[f[x], {x, xMin, 0.290}]/Simplify
```

The result of this calculation is the cumulative distribution function (CDF) evaluated at 0.290:

```
0.015ord1 + 0.03ord2 + 0.03ord3 + 0.03ord4 + 0.03ord5 + 0.03ord6 + 0.03ord7
+ 0.03ord8 + 0.03ord9 + 0.0283333ord10 + 0.00666667ord11.
```

Similarly for the constraint involving the 97.5% point, the integration:

```
symbolicRightTail = Integrate[f[x], {x, 0.831, xMax}]/Simplify
```

produces a symbolic expression for the CDF at 0.831:

```
0.00135ord28 + 0.02265ord29 + 0.03ord30 + 0.03ord31 + 0.03ord32 +
0.03ord33 + 0.03ord34 + 0.03ord35 + 0.03ord36 + 0.03ord37 + 0.03ord38 +
0.03ord39 + 0.03ord40 + 0.03ord41 + 0.03ord42 + 0.03ord43 + 0.03ord44
+ 0.03ord45 + 0.03ord46 + 0.03ord47 + 0.03ord48 + 0.03ord49 + 0.03ord50
+ 0.015ord51.
```


After a symbolic expression for the entropy is generated, all the required setup steps for the Lagrange Multiplier formulation are combined:

```

objective = symbolicEntropy;
nConstraints = 4;
constraint1 = symbolicArea - 1;
constraint2 = symbolicMean - 0.491;
constraint3 = symbolicLeftTail - 0.025;
constraint4 = symbolicRightTail - 0.025;
g = objective + Total[Table[ $\lambda_i$  constrainti, {i, 1, nConstraints}]];
equations1 = Table[ $\partial_{ord_i} g == 0$ , {i, 1, nPoints}];
equations2 = Table[ $\partial_{\lambda_n} g == 0$ , {n, 1, nConstraints}];
equations = Join[equations1, equations2];

```

With suitable starting values for the ord_i and the λ_i , **FindRoot[]** is employed once again:

```

ordinateStartingValues = Table[{ordi, RandomReal[{0.00001, 0.25}], 0.00001, 10},
  {i, 1, nPoints}];
 $\lambda$ StartingValues = Table[{ $\lambda_i$ , 1}, {i, 1, nConstraints}];
soln2 = FindRoot[equations,
  Evaluate[ordinateStartingValues/.{List[x____] → x}],
  Evaluate[ $\lambda$ StartingValues/.{List[x____] → x}], MaxIterations → 10000];

```

Extracting just the replacement rules for the ordinates:

```
replaceRuleSolution = Take[soln2, nPoints];
```

the entropy of the solution is found with

```
symbolicEntropy /. replaceRuleSolution
```

and gives -0.566244.

Figure 4 illustrates the MaxEnt solution with a mean constraint and two tail area constraints, together with its corresponding entropy function.

The entropy of the '5-parameter' Beta solution shown in Figure 1, determined via direct numerical integration, is slightly lower than that of the 51-point MaxEnt solution, as expected:

```

entropy =
NIntegrate[-pdfBetaExtended[ $\alpha, \beta, k, m, \lambda, x$ ] ×
Log[pdfBetaExtended[ $\alpha, \beta, k, m, \lambda, x$ ], {x, k, m}]/.solutionParameters

```

returns a value of -0.633924.

FIGURE 4. MaxEnt problem with a mean constraint and two tail area constraints. Interpolated MaxEnt solution with solution ordinates (left), corresponding entropy function (right).

CONCLUSION

The approach to finding univariate MaxEnt PDFs developed here appears to have some potential, given that it reproduces with high precision some known MaxEnt PDFs. *Mathematica* is a convenient computational environment in which to generate the required symbolic constraint equations and apply the Lagrange Multiplier method. Future work includes exploring the advantage—if any—of using quadratic or higher order interpolation, and the use of other constraint types than those presented here.

REFERENCES

1. E. T. Jaynes, "Where Do We Stand on Maximum Entropy?", in *The Maximum Entropy Formalism*, edited by R. D. Levine and M. Tribus, MIT Press, 1979.