Sloppy exponentials

(Sethna,)

The problem of extracting the decay rates from a sum of exponential decays is a famously difficult inverse problem, from the early days of radioactivity to modern simulations of lattice quantum chromodynamics. In a series of exercises, we shall use our information geometry ideas to study the simplest version of this problem: the sum of N
exponential decays:
$y\boldsymbol{\Theta}(t)$=(1/N) ∑ ^N exp(−$\theta\alpha t$).

We anticipate that it will be challenging to disentangle decay rates θ which are close to one another, unless one has high-precision data over large ranges of time. All the decay curves are smoothly monotonically decreasing, and one could imagine modeling a sum of two decays with a single intermediate decay rate. You shall find in these exercises that this simple model illustrates the behavior we have found widespread in multiparameter models in physics, engineering, biology, and other fields.

In this first exercise, we presume we have perfect experimental data for the decay $d(t)$ at M points t_i equally spread for t between 0 and 10, with separation Δt=10/M. We shall be considering how well this data can be represented by other values of the parameters $\boldsymbol{\Theta}$, so our cost is:

$C(\boldsymbol{\Theta},\boldsymbol{\Theta}$[0])=∑^$M$ ($y\boldsymbol{\Theta}$(t_i)−$y\boldsymbol{\Theta}$[0](t_i))^2/2σ2 ≈ ∫(1/2)($y\boldsymbol{\Theta}(t)$−$y\boldsymbol{\Theta}$[0](t))^2 dt.
where for convenience (since our data is perfect) we set σ2=1/Δt. We shall use the continuum approximation to evaluate the Hessian at the best fit.

To start, suppose $d(t)$ has two decay rates $\boldsymbol{\Theta}$[0]=[1,2], so the data $d(t)$=(1/2)(exp(−t)+exp(−2t)).

(a) Write a function that returns $y\boldsymbol{\Theta}(t)$, and a function that computes the cost for Δt=0.01. Draw a contour plot of C in the square 0.5<$\theta\alpha$<2.5, with contours at C={2−12,2−11,…,20}. Set the number of grid points per side to 40 (so $\Delta\theta$=0.02) to see the two minima.

Remember to not loop over the times: exp({1,2,3}) = [e^1,e^2,e^3]

```
y[θ_][t_] := 1 / Length[θ] Sum[Exp[-θ[[n]] t], {n, 1, Length[θ]}]
θ0 = ...;
dt = ...;
ts = Table[t, { ...}];
Cost[θ_, θ0_, ts_] := dt Sum[(y[θ][ts[[i]]] - y[θ0][ts[[i]]]) ^2, {i, 1, Length[ts]}]
(* Too slow! Do it with vector operations. y[θ][ts]
 generates entire vector of predictions *)
Cost[θ_, θ0_, ts_] := dt Norm[y[θ][ts] - ...] ^2 / 2
CostContinuum[θ_, θ0_] := Integrate[( ...) ^2 / 2, {t, 0, ∞}]
```

```
In[ ]:= Cost[{2, 3}, θ0, ts]
CostContinuum[{2, 3}, θ0] // N
```

```
levels = Table[2^n, {n, -14, -4}];
ContourPlot[Cost[{θx, θy}, θ0, ts], {θx, 0.5, 2.5}, {θy, 0.5, 2.5},
 Contours → levels, ContourShading → None, MaxRecursion → 3]
```

The diagonal in this plot gives single exponential decays. How well does a single exponential capture the behavior at $\mathbf{\Theta}$[0]?

(b) Constraining $\theta 1$=$\theta 2$, find the point of minimum cost θmin. Where is the point on the contour plot? Compare the two curves $y\theta[0](t)$ and $y\theta\mathrm{min}(t)$, and also plot their difference.

```
toMinimize1exponent[θ_] := Cost[{ ...}, θ0, ts]
minθ = θ /. FindMinimum[ ..., {θ, 1.5}][[2]]
Plot[{y[ ...][t], y[θ0][t]}, {t, 0, 10}, PlotRange → All]
Plot[y[θ0][t] - y[ ...][t], {t, 0, 10}, PlotRange → All]
```

One can see from the contour plot that measuring the two rate constants separately would be a challenge. This is because the two exponentials have similar shapes, so increasing one decay rate and decreasing the other can almost perfectly compensate for one another.

This clearly is not a deep truth for two exponentials. But the effect is hugely magnified when we have many parameters. We can see this by computing the eigenvalues of the cost Hessian.

(c) Analytically calculate the Jacobian $J\_t\alpha$=$\partial y\_\mathbf{\Theta}(t)/\partial\theta\alpha$ in the continuum approximation. Using the Jacobian, show that the Hessian for the cost evaluated at the best fit is H_$\alpha\beta$=(2/N^2) (1/($\theta\alpha$+$\theta\beta$)^3).

```
J[t_, θα_] = D[ ... [t], θα]
```

```
H[θα_, θβ_, 2] = Integrate[ ..., {t, 0, ∞}, Assumptions → {θα > 0, θβ > 0}]
```

(d) Using your answer from part (c), write a routine to calculate the entire array $H(\mathbf{\Theta})$. Check it by examining the eigenvectors and eigenvalues for the N=2 case of part (b). What do you predict the ratio R = (long axis/short axis) to be, in terms of the two eigenvalues λstiffer and λsloppier? Are the directions roughly in line with the eigenvectors?

```
Hess = Table[H[θα, θβ, 2], {θα, θ0}, {θβ, θ0}]
Hess // N
Eigensystem[Hess] // N
...
```

```
H[θα_, θβ_, n_] = 2 / (n^2 ...)
```

(e) For a sum of seven exponentials, with $\mathbf{\Theta}$[0]=[1,2,3,…,7], construct the Hessian, and find its eigenvalues. Are they sloppy (roughly equally spaced in log)? By roughly what factor does each successive eigenvalue shrink?

```
θ7 = 1.0 Table[n, {n, 1, 7}];
Hess7 = Table[ ..., {θα, θ7}, {θβ, θ7}];
vals = Eigenvalues[Hess];
vals
...
```

This sloppiness makes it strikingly difficult to extract the parameter values from the data.

(f) Argue that the number of measurements n_measure needed to estimate a parameter scales inversely with its variance (n_measure ~ 1/σ^2). Given that the eigenvalues of the Hessian give the variance along the various eigendirections, by what factor $n$sloppy/$n$stiff is it harder to measure the parameters along the sloppy directions, for your sum of seven exponentials?

```
In[◦]:= SloppyStiffRatio = (1 / vals⟦7⟧) / (1 / vals⟦1⟧)
```

(g) Given that the diagonal elements of the inverse cost Hessian are proportional to the variance in the corresponding parameter for one sampling of the Gaussian given by the cost, what are the variances in the seven parameters $\theta 0\_\alpha$?

```
σVary7 = Table[Sqrt[Inverse[ ...]⟦ ...⟧], {n, 1, 7}]
```