File: /General/MLAB-Text/Papers/fitexp/fitexp.tex

Fitting Multi-Exponential Models

Gary D. Knott, Ph.D.
Civilized Software, Inc.
12109 Heritage Park Circle
Silver Spring, MD 20906 USA
Tel. (301) 962-3711
Email: csi@civilized.com
URL: www.civilized.com

Fitting models of the form $f(x) = a_1 \cdot \exp(b_1 x) + a_2 \exp(b_2 x) + \cdots + a_k \exp(b_k x)$ to given data points (x_i, y_i), $i = 1, 2, \ldots, n$ is a common problem. Unfortunately, the results are often unsatisfying, even when they can be obtained. It will often be preferable to use a differential-equation-based model, e.g. a compartmental model, which has some physical basis. We may illustrate the difficulties with a particular example. Let us consider the following data due to Dr. Paul Schloerb of the University of Rochester.

x_i	y_i
1	39.814
2	32.269
3	29.431
4	27.481
5	26.086
6	25.757
7	24.932
8	23.928
9	22.415
10	22.548
11	21.900
12	20.527
13	20.695
14	20.105
15	19.516
16	19.640
17	19.346
18	18.927
19	18.857
20	17.652

We will use the model $f(x) = a_1 \exp(b_1 x) + a_2 \exp(b_2 x) + a_3$. Our goal is to estimate the values a_1, a_2, a_3, b_1, and b_2, so that $f(x_i) \approx y_i$, employing the least-squares minimization method. We may enter this data and the specified model in MLAB as follows.

```
* M = (1:20)&'kread(20)
39.814 32.269 29.431 27.481 26.086 25.757 24.932 23.928 22.415 22.548 21.9
20.527 20.695 20.105 19.516 19.64 19.346 18.927 18.857 17.652
* FUNCTION F(X)=A1*EXP(B1*X)+A2*EXP(B2*X)+A3
* CONSTRAINTS Q = {A1 > 0, A2 > 0, A3 > 0, B1 < 0, B2 < 0}
```

We often need reasonable guesses for the parameters a_1, a_2, a_3, b_1 and b_2, which are sufficiently close to the "true" values. Otherwise, we may obtain a solution which corresponds to an undesirable local minimum rather than that local minimum associated with the most physically-meaningful solution, as in the following example.

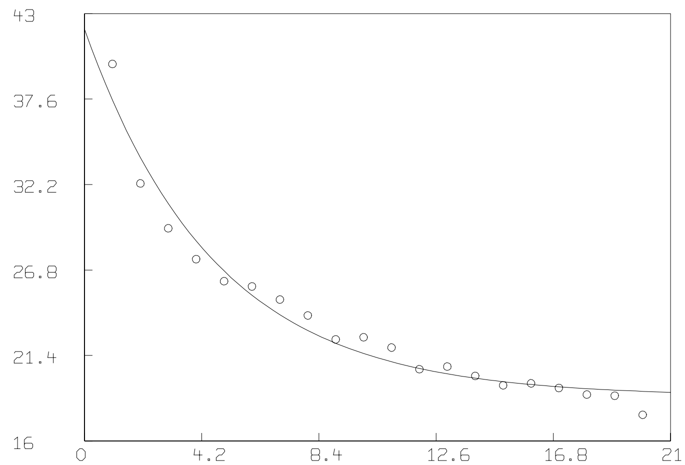
```
* A1 = 1; A2 = .5; A3 = 1; B1 = -1; B2 = -.5
* FIT(A1,A2,A3,B1,B2), F TO M ,CONSTRAINTS Q
matherr: underflow error in exp
         arg = -1.037332e+03
         return value: 0.000000e+00
final parameter values
      value               error          dependency    parameter
  3.6182537e-16      1.797693135e+308        1             A1
 2.42048135e-17      1.797693135e+308        1             A2
    23.5913          1.372938916             0             A3
   -5141744.558      1.797693135e+308        1             B1
   -176780.4082      1.797693135e+308        1             B2
5 iterations
CONVERGED
best weighted sum of squares = 5.654884e+02
weighted root mean square error = 6.139969e+00
weighted deviation fraction = 1.854011e-01
R squared = 6.031254e-15
no active constraints
```

High dependency-valuesıdependency-values near 1 implies an ill-conditioned Jacobian matrixıJacobian matrix. There may be too many parameters, a lack of data in an important range, or poorly-chosen initial parameter guesses.

It is possible to make bad initial guesses in many ways. If we start with `A1=A2=A3=1` and `B1=B2=-1` for example, then the two exponential terms in our model are identical, and, in fact, the model has degenerated into a one-exponential model. The curve-fitting process may or may not be able to re-separate the two distinct exponential components, depending upon whether there are lucky intermediate parameter values generated at the beginning of an iteration or not. Often the "wild" parameter vectors which arise for a singular Jacobian matrix with a nearly-unmagnified diagonal are "unlucky". We shall call such initial guesses degenerate.

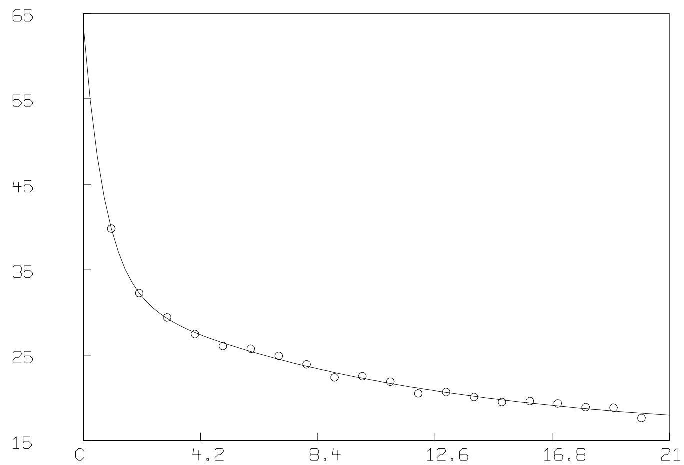
The correct approach is to make reasonable non-degenerate guesses to start with. If we fit a one-exponential term model to the data, the result may help in starting to fit the two-exponential term model. By looking at the data, we may proceed as follows.

```
* A1=20; A2=0; A3=20; B1=-.1; B2=0;
* FIT (A1,B1,A3), F TO M, CONSTRAINTS Q
final parameter values
      value              error        dependency  parameter
    23.19717945         1.240156312    0.6125079766   A1
  -0.2148403998       0.02481714336    0.8665292689   B1
     18.8176717        0.5477893054    0.8145001634   A3
4 iterations
CONVERGED
best weighted sum of squares = 1.892560e+01
weighted root mean square error = 1.055116e+00
weighted deviation fraction = 3.492313e-02
R squared = 9.665323e-01
no active constraints
* DRAW M, POINTTYPE  CIRCLE PTSIZE .01, LINETYPE NONE
* DRAW p = POINTS(F, 0:21:.25); VIEW;
```



Now we may guess introductory values for `A2` and `B2`, and proceed to fit the complete two-exponential term model.

```
* A2 = 10; B2 = -2
* FIT(A1,A2,A3,B1,B2), F TO M, CONSTRAINTS Q
final parameter values
      value            error         dependency    parameter
    17.35185308   0.5924659802    0.9172348481   A1
    30.65781056    6.440647997    0.9597145815   A2
    15.67738647    1.020820115    0.9935853439   A3
  -0.0961929828  0.01697024506    0.9926829646   B1
   -1.297370234   0.2737645734    0.9810419889   B2
17 iterations
CONVERGED
best weighted sum of squares = 2.005363e+00
weighted root mean square error = 3.656376e-01
weighted deviation fraction = 9.897291e-03
R squared = 9.964537e-01
no active constraints
* DRAW p = POINTS(F,0:21:.25); VIEW;
```



65
55
45
35
25
15
0
4.2
8.4
12.6
16.8
21