6.581J / 20.482J
Foundations of Algorithms and Computational Techniques in Systems Biology
Professor Bruce Tidor
Professor Jacob K. White

6.581/20.482J Problem Set #5

Due: 5 PM Tuesday 5/2/06

Problem 1

In the previous homework, we found the fixed points of a model by solving for x such that the time derivative $\dot{x}$ went to zero.

$$\dot{x} = F(x(p), p) = 0 \tag{1}$$

For the simple 2×2 system we were able to find an expression for the fixed point as a function of the parameters. However, for the lambda switch we only had a numerical solution. In this problem set, we will look at solving the reverse problem. Given measurements of the fixed points of the model, we want to find the parameters that are consistent with our measurements.

In class, we talked about least squares minimization, which is one choice for formulating this problem. In least squares, we are given some observation $c^T \bar{x}$, in this case the concentrations at the fixed point, and a formula for calculating the same data, $c^T x(p)$ given our model and choice of parameters p. The problem is to solve for the parameter p that minimizes the squared error defined by:

$$\Phi(x(p), p) = (c^T x(p) - c^T \bar{x})^2 \tag{2}$$

$$\hat{p} = argmin \Phi(x(p), p) \tag{3}$$

Note that c^T picks out the model variables, or combination of variables that correspond to our measurement.

1. The algorithm we discussed in class for solving the least squares problems is gradient descent.

```
1: Guess $p_0$
2: for $i = 0$ until convergence do
3:     Compute $x(p_i)$
4:     Compute $\Phi(x(p_i), p_i)$
5:     Compute $\nabla_p \Phi(p_i)$
6:     Compute $\hat{\beta} = argmin \ \ \Phi(p_i + \beta \nabla_p \Phi(p_i))$
7:     Update $p_{i+1} = p_i + \hat{\beta} \nabla_p \Phi(p_i)$
8: end for
```

   Describe briefly how the gradient descent algorithm works. When the gradient descent method converges (i.e.the updates stop changing), it usually means that the gradient with respect to the parameters is zero for the converged value of the parameters. Does that mean we have found the best set of parameters?

2. In general, we won't measure all of system variables. For example, we might only know a few of the concentrations. The matrix c^T picks out the quantities that are measured. In this experiment, the data we have is the total concentration of Cro and CI.

$$Cro_{total} = x_{Cro} + 2 \cdot x_{Cro_2} + 2 \cdot x_{Cro_2 B} \tag{4}$$

$$CI_{total} = x_{CI} + 2 \cdot x_{CI_2} + 2 \cdot x_{CI_2 B} \tag{5}$$

   Where x_{Cro} is the concentration of free *Cro*, x_{Cro_2} is the concentration of *Cro* dimer, and $x_{Cro_2 B}$ is the concentration of *Cro* dimer bound to the DNA (Likewise for the *CI* variants). Given this measurement, what is c^T? What if instead we used a micro array and only knew the concentration of mRNA for Cro and CI, what would c^T be then?

3. Step 5 of the algorithm requires us to compute the gradient of the cost function $\nabla_p \Phi(x(p), p)$, with respect to the unknown parameters p. Derive the gradient in terms of $F(x(p), p)$, the Jacobian $J_F(x(p), p)$, and

$\nabla_p F(x(p), p)$. We discussed two methods for solving this problem, the Forward Method and the Adjoint Method. Show the place in the derivation where you have choosen between these two methods. Under what circumstances will the Adjoint Method be faster than the Forward Method? Make your argument in terms of the dimensions of x, $c^T x$, and p.

4. The MATLAB file `descent.m` implements the gradient descent algorithm on a model file. The model file takes as input a particular choice of parameter p, and returns $\Phi(x(p), p)$, $\nabla_p \Phi(x(p), p)$. Fill in the missing parts of the file `lambda2.m`. using the lambda switch model given in Equation **??** below. *Hint: the first part can be implemented using newton's method code from Problem Set 5.* Using your filled `lambda2.m` and `descent.m`, find the parameters that will make the model fixed points match given data:

Experiment	Total Cro	Total CI	Cro mRNA	CI mRNA
1	33.71	0.11	1.57	0.01
2	0.05	3.74	0.02	0.479

5. Does the answer you get from Part **??** depend on your initial guess p_0? If not, why not? If so, what additional information would you need to get the correct value of p?

$$\frac{d}{dt}\begin{bmatrix} x_{mCro} \\ x_{Cro} \\ x_{Cro_2} \\ x_{Cro_2B} \\ x_{mCI} \\ x_{CI} \\ x_{CI_2} \\ x_{CI_2B} \end{bmatrix} = \begin{bmatrix} \gamma^{mCro}_{Cro_2B} x_{Cro_2B} - \gamma^{0}_{mCro} x_{mCro} \\ \gamma^{Cro}_{mCro} x_{mCro} + 2\gamma^{Cro}_{Cro_2} x_{Cro_2} - 2\gamma^{Cro_2}_{Cro} x^2_{Cro} - \gamma^{0}_{Cro} x_{Cro} \\ \gamma^{Cro_2}_{Cro} x^2_{Cro} + \gamma^{Cro_2}_{Cro_2B} x_{Cro_2B} - \gamma^{Cro}_{Cro_2} x_{Cro_2} - \gamma^{Cro_2B}_{Cro_2} x_{Cro_2}(p_{DNA_{tot}} - x_{Cro_2B} - x_{CI_2B}) \\ \gamma^{Cro_2B}_{Cro_2} x_{Cro_2}(p_{DNA_{tot}} - x_{Cro_2B} - x_{CI_2B}) - \gamma^{Cro_2}_{Cro_2B} x_{Cro_2B} \\ \gamma^{mCI}_{CI_2B} x_{CI_2B} - \gamma^{0}_{mCI} x_{mCI} \\ \gamma^{CI}_{mCI} x_{mCI} + 2\gamma^{CI}_{CI_2} x_{CI_2} - 2\gamma^{CI_2}_{CI} x^2_{CI} - \gamma^{0}_{CI} x_{CI} \\ \gamma^{CI_2}_{CI} x^2_{CI} + \gamma^{CI_2}_{CI_2B} x_{CI_2B} - \gamma^{CI}_{CI_2} x_{CI_2} - \gamma^{CI_2B}_{CI_2} x_{CI_2}(p_{DNA_{tot}} - x_{Cro_2B} - x_{CI_2B}) \\ \gamma^{CI_2B}_{CI_2} x_{CI_2}(p_{DNA_{tot}} - x_{Cro_2B} - x_{CI_2B}) - \gamma^{CI_2}_{CI_2B} x_{CI_2B} \end{bmatrix} \tag{6}$$