Tait Kaminski
904964310

EE133a Final Project

Problem Formulation:
So, to formalize our problem, we take our node positions as,

$$\rho_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix}$$

We also know we have 9 known points, or anchor nodes from the example figure,

$$\rho_{22} \sim \rho_{30}$$

and unknown points, or free nodes

$$\rho_1 \sim \rho_{21}$$

So, for distance,

$$f_k(\theta) = \left\| \rho_{i_k} - \rho_{j_k} \right\| - \rho_k \qquad k = 1, \ldots, L$$

Our goal is to minimize,

$$\text{minimize } \|f(\theta)\|^2 = \left\| \begin{bmatrix} f_1(\theta) \\ \vdots \\ f_k(\theta) \\ \vdots \\ f_L(\theta) \end{bmatrix} \right\|^2$$

where,

$$\theta = (\rho_1, \ldots, \rho_{21}) = \left(\begin{bmatrix} u_1 \\ v_1 \end{bmatrix}, \ldots, \begin{bmatrix} u_{21} \\ v_{21} \end{bmatrix} \right)$$

$$= \begin{bmatrix} u_1 \\ v_1 \\ \vdots \\ u_{21} \\ v_{21} \end{bmatrix}$$

Thus, for the derivative matrix, every row should follow,

$$Df(\theta) = \begin{bmatrix} \frac{\partial f_k}{\partial u_1} & \frac{\partial f_k}{\partial v_1} & \cdots & \frac{\partial f_k}{\partial u_{21}} & \frac{\partial f_k}{\partial v_{21}} \end{bmatrix}$$

So given k = 9 and N-K = 21 $\begin{bmatrix} u_i \\ v_i \end{bmatrix}$ pairs, our Derivative matrix is a 9 x 42 matrix that can be summarized with elements,

$$Df_k(\theta) = \sum_{k=1}^{L} \frac{\partial \left(\left\| \rho_{i_k} - \rho_{j_k} \right\|^2 - {\rho_k}^2 \right)}{\partial u_i}$$

$$Df_k(\theta) = \sum_{k=1}^{L} \frac{\partial \left(\left\| \rho_{i_k} - \rho_{j_k} \right\|^2 - {\rho_k}^2 \right)}{\partial v_i}$$

So, taking the partials, we get,

$$Df_k(\theta) = \frac{\partial f_k}{\partial u_i}(\theta) = \sum_{k=1}^{L} 2(u_{i_k} - u_{j_k})$$

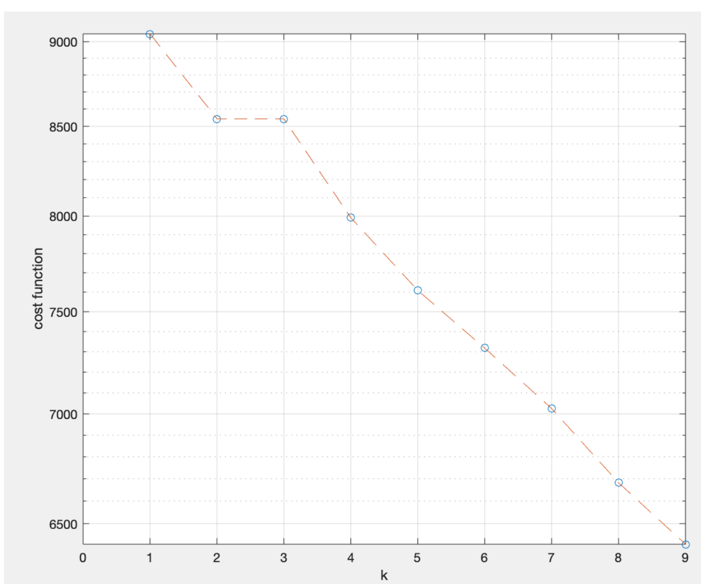
$$Df_k(\theta) = \frac{\partial f_k}{\partial v_i}(\theta) = \sum_{k=1}^{L} 2(v_{i_k} - v_{j_k})$$

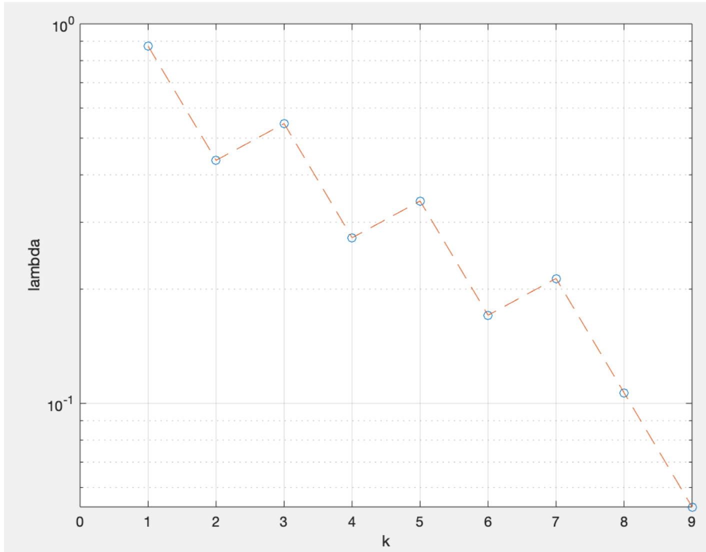
Which is clearly nonlinear and now we can see if f_k contains one free vector, we have two nonzero elements ($\frac{\partial f_1}{\partial u_1}$ and $\frac{\partial f_1}{\partial v_1}$ if $k = 1$ for example) and if f_k contains two free vectors, we have four nonzero elements ($\frac{\partial f_1}{\partial u_1}, \frac{\partial f_1}{\partial v_1}, \frac{\partial f_2}{\partial u_2}$ and $\frac{\partial f_2}{\partial v_2}$ if $k = 2$ for example). Finally, Df can be summarized as this in the case of the example given in the handout. The dimensions would change based on N.

$$Df(\theta) = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial v_1} & 0 & \cdots & \cdots & \cdots & 0 \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial v_1} & \frac{\partial f_2}{\partial u_2} & \frac{\partial f_2}{\partial v_2} & 0 & \cdots & 0 \\ \vdots & & & & & & \vdots \\ \vdots & & & \ddots & & & \vdots \\ \vdots & & & & & & \vdots \\ \vdots & & & & & & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & \frac{\partial f_9}{\partial u_9} & \frac{\partial f_9}{\partial v_9} \end{bmatrix}$$

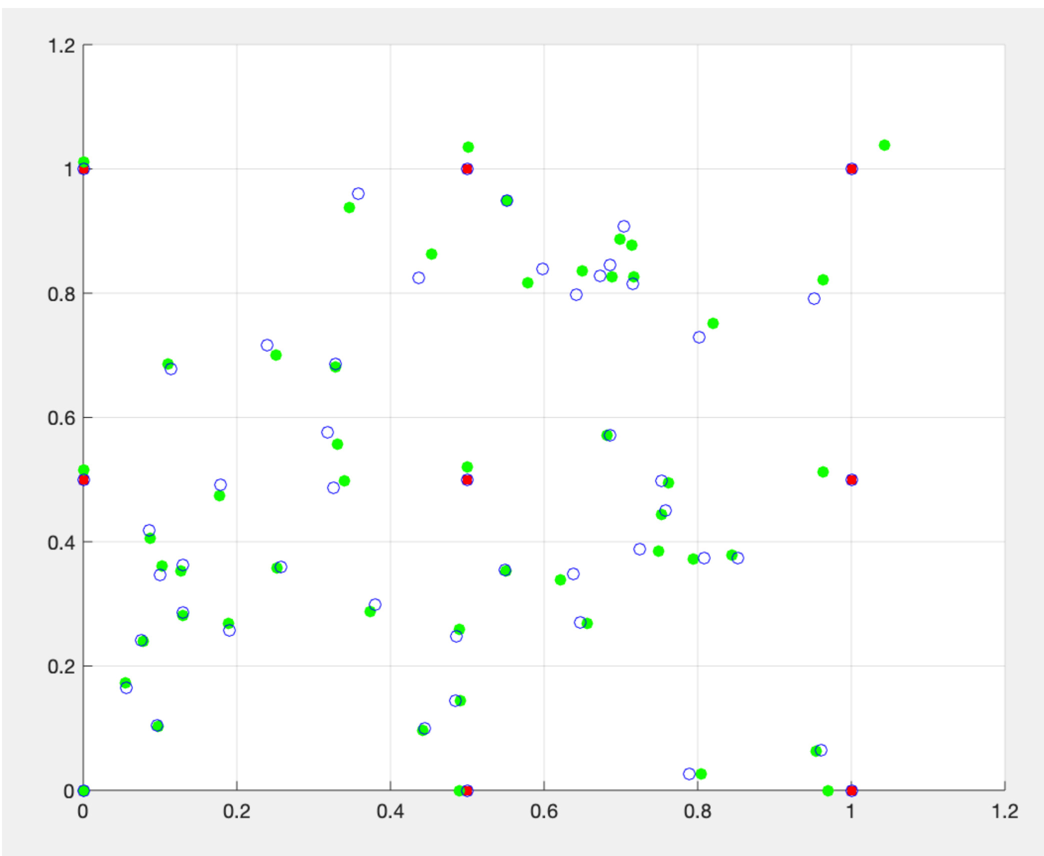
Results:
All results and code shown here is with the number of nodes N=50.

Here are the plots of the regularization parameter and the cost function versus iteration number,





And for the estimated positions, the red are the anchor points, the blue are the unknown positions, and the green are the known positions.



Code appendix:

```
function [pos_free] = network_loc(N, E, pos_anchor, rho);
N=50; R=0.4; s=0.05;
[E, pos, K] = network_loc_data(N, R);
pos_anchor = pos(N-K+1:N, :);
L = size(E,1);
d = sqrt(sum( (pos(E(:,1),:) - pos(E(:,2),:)).^2, 2));
rho = (1 + s*randn(L,1)) .* d;

lambda = 1.0;
u = rand(N-K,1);
v = rand(N-K,1);
theta = [u;v];
cost=[];

for iters = 1:100

    f = (theta.^2);
    Df = (2*theta);
    cost = [cost, norm(2*Df'.*f).^2];

    if norm(2*Df'.*f) <= 10e-5; break; end;

    newtheta = theta - [Df'; sqrt(lambda)*eye(82)] \ [f;zeros(1,1)];
    newf = newtheta.^2;

    if norm(newf) < norm(f)
        theta = newtheta;
        lambda = .8 * lambda;
    else
        lambda = 2 * lambda;
    end;
end;

%plot
semilogy([0:iters-1], cost, 'o', [0:iters-1], cost, '--');
xlabel('k');
ylabel('cost function');
grid on;
```