11.1 Are the following matrices positive definite?

(a) $A = \begin{bmatrix} -1 & 2 & 3 \\ 2 & 5 & -3 \\ 3 & -3 & 2 \end{bmatrix}$.

$\begin{bmatrix} -1 & 2 & 3 \\ 2 & 5 & -3 \\ 3 & -3 & 2 \end{bmatrix}$

A is symmetric $\langle n \times n \rangle$

not positive definite,

"every positive definite matrix A, has positive diagonal elements"

$A_{ii} = e_i^T A e_i > 0$

(b) $A = I - uu^T$ where u is an n-vector with $\|u\| < 1$.

$x^T A x > 0$ for all $x \neq 0$

$x^T A x = x^T (I - uu^T) x = (x^T I - x^T uu^T) x = x^T I x - x^T uu^T x = x^T x - x^T uu^T x$

$= \|x\|^2 - x^T uu^T x$ ← $(u^T x)^2 \leq (\|u\|^2 \|x\|^2)$ Cauchy Schwarz

$x^T A x \geq \|x\|^2 - \|u\|^2 \|x\|^2$

u & x are vectors

$= \|x\|^2 (1 - \|u\|^2) > 0$

$>0 \quad >0 \quad (x \neq 0)$

(d) $A = \begin{bmatrix} 1 & u^T \\ u & I \end{bmatrix}$ where u is an n-vector with $\|u\| < 1$.

suppose $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

$x^T A x \Rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 1 & u^T \\ u & I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Rightarrow \begin{bmatrix} x_1 & x_2^T \end{bmatrix} \begin{bmatrix} 1 & u^T \\ u & I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

$\langle 1 \times 2 \rangle \langle 2 \times 2 \rangle \langle 2 \times 1 \rangle$

$\Rightarrow \begin{bmatrix} x_1 + x_2 u & x_1 u^T + x_2 I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = (x_1 + x_2^T u)(x_1) + (x_1 u^T + x_2 I)(x_2)$

$= x_1^2 + x_2^T u x_1 + x_1 u^T x_2 + x_2 I x_2$

$= x_1^2 + x_2^T u x_1 + x_1 u^T x_2 + x_2^2$

2 of

x_1 & x_2 zero

$\|x_1\|^2 + 2 x_1 u^T x_2 + \|x_2\|^2$

$x_1^2 + 2 x_1 u^T x_2 + \|x_2\|^2 > 0$

treat as polynomial, complete the square

$x_1^2 + 2 x_1 (u^T x_2) + \|x_2\|^2 > 0 \rightarrow x_1^2 + 2 x_1 (u^T x_2) + (u^T x_2)^2 - (u^T x_2)^2 + \|x_2\|^2 > 0$

$\Rightarrow (x_1 + (u^T x_2))^2 + \|x_2\|^2 - (u^T x_2)^2 \geq 0 \Rightarrow (x_1 - u^T x_2) + \|x_2\|^2 - \|x_2\|^2 \|u\|^2 \geq 0$

Cauchy – Schwarz

if $\|u\|^2 = 1$, it would be zero, $(x_1 - u^T x_2) + \|x_2\|^2 (1 - \|u\|^2) \geq 0$

$A = R^T R$ Faster method?

11.8 For what values of the scalar a are the following matrices positive definite? To derive the conditions, factor A using a Cholesky factorization and collect the conditions on a needed for the factorization to exist.

(b) $A = \begin{bmatrix} 1 & a & 0 \\ a & 1 & a \\ 0 & a & 1 \end{bmatrix}$. $= \begin{bmatrix} R_{11} & 0 & 0 \\ R_{12} & R_{22} & 0 \\ R_{13} & R_{23} & R_{33} \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ 0 & R_{22} & R_{23} \\ 0 & 0 & R_{33} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 \\ R_{1,2:n}^T & R_{2:n,2:n}^T \end{bmatrix} \begin{bmatrix} R_{11} & R_{1,2:n} \\ 0 & R_{2:n,2:n} \end{bmatrix}$

$R_{11} = \sqrt{A_{11}} = 1 \quad R_{1,2:n} = \begin{bmatrix} R_{12} & R_{13} \end{bmatrix} = \begin{bmatrix} a & 0 \end{bmatrix} \rightsquigarrow R_{1,2:n}^T = \begin{bmatrix} R_{12} & R_{13} \end{bmatrix}^T = \begin{bmatrix} R_{12} \\ R_{13} \end{bmatrix} = \begin{bmatrix} a \\ 0 \end{bmatrix}$

$A_{2:n,2:n} - R_{1,2:n}^T R_{1,2:n} = R_{2:n,2:n}^T R_{2:n,2:n}$

$\begin{bmatrix} 1 & a \\ a & 1 \end{bmatrix} - \begin{bmatrix} a \\ 0 \end{bmatrix} \begin{bmatrix} a & 0 \end{bmatrix} = \begin{bmatrix} R_{22} & 0 \\ R_{23} & R_{33} \end{bmatrix} \begin{bmatrix} R_{22} & R_{23} \\ 0 & R_{33} \end{bmatrix} = \begin{bmatrix} R_{22}^2 & R_{22}R_{23} \\ R_{22}R_{23} & R_{23}^2 + R_{33}^2 \end{bmatrix} \rightsquigarrow \begin{bmatrix} 1 & a \\ a & 1 \end{bmatrix} - \begin{bmatrix} a^2 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1-a^2 & a \\ a & 1 \end{bmatrix}$

$\begin{bmatrix} 1-a^2 & a \\ a & 1 \end{bmatrix} = \begin{bmatrix} R_{22}^2 & R_{22}R_{23} \\ R_{22}R_{23} & R_{23}^2 + R_{33}^2 \end{bmatrix}$ $R_{22}^2 = 1 - a^2 \Rightarrow R_{22} = \sqrt{1-a^2}$

$R_{22}R_{23} = a \Rightarrow R_{23} = \frac{a}{R_{22}} = \frac{a}{\sqrt{1-a^2}} = R_{32}$

$R_{23}^2 + R_{33}^2 = 1 \Rightarrow R_{33}^2 = 1 - R_{23}^2 = 1 - \left[\frac{a}{\sqrt{1-a^2}}\right]^2 = \frac{1-a^2}{1-a^2} - \frac{a^2}{1-a^2} = \frac{1-2a^2}{1-a^2} = R_{33}^2 \rightarrow R_{33} = \frac{\sqrt{(1-2a^2)}}{\sqrt{1-a^2}}$

$\therefore \begin{bmatrix} 1 & 0 & 0 \\ a & \sqrt{1-a^2} & 0 \\ 0 & \frac{a}{\sqrt{1-a^2}} & \frac{\sqrt{1-2a^2}}{\sqrt{1-a^2}} \end{bmatrix} \begin{bmatrix} 1 & a & 0 \\ 0 & \sqrt{1-a^2} & \frac{a}{\sqrt{1-a^2}} \\ 0 & 0 & \frac{\sqrt{1-2a^2}}{\sqrt{1-a^2}} \end{bmatrix}$

diagonals must be greater than 0

$1 > 0$ ✓

$\sqrt{1-a^2} > 0 \rightarrow 1 - a^2 > 0 \Rightarrow 1 > a$

$\sqrt{1-2a^2} > 0 \Rightarrow 1 > 2a^2 \rightarrow \sqrt{\frac{1}{2}} > |a|$

(denom > 0, From above)

Is my reasoning correct?

(d) $A = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & 0 \\ a & 0 & 1 \end{bmatrix}$. $R_{11} = \sqrt{A_{11}} = 1$; $R_{1,2:n} = \begin{bmatrix} R_{12} & R_{13} \end{bmatrix}$

$R_{12} = \frac{1}{R_{11}} A_{1,2:n} = \frac{1}{1}\begin{bmatrix} 0 & a \end{bmatrix} = \begin{bmatrix} 0 & a \end{bmatrix} \quad \therefore R_{1,2:n}^T = \begin{bmatrix} 0 \\ a \end{bmatrix}$

$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 \\ a \end{bmatrix} \begin{bmatrix} 0 & a \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1-a^2 \end{bmatrix} = \begin{bmatrix} R_{22}^2 & R_{22}R_{23} \\ R_{22}R_{23} & R_{23}^2 + R_{33}^2 \end{bmatrix}$ $R_{22} = 1$, $R_{23} = R_{32} = 0$, $R_{33} = \sqrt{1-a^2}$

$A = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{12} & R_{22} & 0 \\ R_{13} & R_{23} & R_{33} \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ 0 & R_{22} & R_{23} \\ 0 & 0 & R_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & 0 & \sqrt{1-a^2} \end{bmatrix} \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & 0 \\ 0 & 0 & \sqrt{1-a^2} \end{bmatrix}$

$1 - a^2 > 0$

$1 > a^2$

$1 > |a|$

(h) $A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a \\ 1 & a & 2 \end{bmatrix}$. $R_{11} = \sqrt{A_{11}} = 1$

$R_{1,2:n} = \frac{1}{R_{11}} A_{1,2:n} = \begin{bmatrix} R_{12} & R_{13} \end{bmatrix} = \frac{1}{R_{11}}\begin{bmatrix} A_{12} & A_{13} \end{bmatrix} = \frac{1}{1}\begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix} \quad R_{1,2:n}^T = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$A_{2:n,2:n} - R_{1,2:n}^T R_{1,2:n} = R_{2:n,2:n}^T R_{2:n,2:n}$

$\begin{bmatrix} a & a \\ a & 2 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} a-1 & a-1 \\ a-1 & 1 \end{bmatrix} = \begin{bmatrix} R_{22}^2 & R_{22}R_{23} \\ R_{22}R_{23} & R_{23}^2 + R_{33}^2 \end{bmatrix}$

$R_{22} = \sqrt{a-1}$

$R_{23} = R_{32} = \sqrt{a-1}$

$R_{33} = \sqrt{2-a}$

$A = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{12} & R_{22} & 0 \\ R_{13} & R_{23} & R_{33} \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ 0 & R_{22} & R_{23} \\ 0 & 0 & R_{33} \end{bmatrix} = \begin{matrix} 1 & 0 & 0 \\ 1 & \sqrt{a-1} & 0 \\ 1 & \sqrt{a-1} & \sqrt{2-a} \end{matrix}$

$a - 1 > 0 \rightarrow a > 1$

$2 - a > 0 \quad 2 > a$

$1 < a < 2$

11.13 You are given the Cholesky factorization $A = R^T R$ of a positive definite matrix A of size $n \times n$, and an n-vector u.

(a) What is the Cholesky factorization of the $(n+1) \times (n+1)$ matrix

$$B = \begin{bmatrix} A & u \\ u^T & 1 \end{bmatrix}?$$

You can assume that B is positive definite.

(b) What is the complexity of computing the Cholesky factorization of B, if the factorization of A (*i.e.*, the matrix R) is given?

Fixed notation

a) A u u^T 1
$\langle n \times n \rangle \langle n \times 1 \rangle \langle 1 \times n \rangle \langle 1 \times 1 \rangle$

$A = A_{1:n,1:n}$, $u = V_{1:n,n+1}$, $u^T = V_{n+1,1:n}$, $1 = C_{n+1,n+1}$

$\therefore R = \begin{bmatrix} D_{1:n,1:n} & D_{1:n,n+1} \\ 0 & D_{n+1,n+1} \end{bmatrix} \rightarrow R^T = \begin{bmatrix} D^T_{1:n,1:n} & 0 \\ D^T_{1:n,n+1} & D_{n+1,n+1} \end{bmatrix}$ $\quad D^T_{n+1,n+1} = D_{n+1,n+1}$

$B = A' = R'^T R' = \begin{bmatrix} D_{1:n,1:n} & 0 \\ D^T_{1:n,n+1} & D_{n+1,n+1} \end{bmatrix} \begin{bmatrix} D_{1:n,1:n} & D_{1:n,n+1} \\ 0 & D_{n+1,n+1} \end{bmatrix} = \begin{bmatrix} A & u \\ u^T & 1 \end{bmatrix}$

$\rightarrow \begin{bmatrix} D^T_{1:n,1:n} & 0 \\ D^T_{1:n,n+1} & D_{n+1,n+1} \end{bmatrix} \begin{bmatrix} D_{1:n,1:n} & D_{1:n,n+1} \\ 0 & D_{n+1,n+1} \end{bmatrix}$

$\begin{bmatrix} D^T_{1:n,1:n} D_{1:n,1:n} & D^T_{1:n,1:n} D_{1:n,n+1} \\ D^T_{1:n,n+1} D_{1:n,1:n} & D^T_{1:n,n+1} D_{1:n,n+1} + D^2_{n+1,n+1} \end{bmatrix}$ $\quad A = D^T_{1:n,1:n} D_{1:n,1:n}$ $\quad u = D^T_{1:n,1:n} D_{1:n,n+1}$

$u^T = D^T_{1:n,n+1} D_{1:n,1:n}$ $\quad 1 = D^T_{1:n,n+1} D_{1:n,n+1} + D^2_{n+1,n+1}$

$A = D^T_{1:n,1:n} D_{1:n,1:n} = R^T R \rightarrow R = D_{1:n,1:n}$

$u = D^T_{1:n,1:n} D_{1:n,n+1} \Rightarrow U = R^T D_{1:n,n+1} \Rightarrow D_{1:n,n+1} = R^{-T} u = D^{-T}_{1:n,1:n} u$

$u^T = D^T_{1:n,n+1} D_{1:n,1:n} \Rightarrow U^T = D^T_{1:n,n+1} R \Rightarrow u^T R^{-1} = D^T_{1:n,n+1}$ $\quad [(R^{-T}u)^T = u^T R^{-1}]$

$1 = D^T_{1:n,n+1} D_{1:n,n+1} + D^2_{n+1,n+1} \rightarrow D_{n+1,n+1} = 1 - D^T_{1:n,n+1} D_{1:n,n+1}$ $\quad (R^T R)^{-1} = R^{-1} R^{-T}$

$\therefore D_{n+1,n+1} = \sqrt{1 - u^T A^{-1} u}$ $\quad u^T R^{-1} \; R^{-T} u \rightarrow u^T (R^T R)^{-1} u$, $(R^T R) = A$

$B = \begin{bmatrix} R^T & 0 \\ u^T R^{-1} & \sqrt{1 - u^T A^{-1} u} \end{bmatrix} \begin{bmatrix} R & R^{-T} u \\ 0 & \sqrt{1 - u^T A^{-1} u} \end{bmatrix}$

b) R is given, compute $U_{1:n,n+1} \rightarrow R^T x = u$ forward substitution, n^2 flops

$U_{n+1,n+1}$ is $2n$ flops $\therefore$ total is $n^2 + 2n$ flops $\sim 2n$

need help counting flops

11.19 *Multi-class classification.* In this exercise we implement the handwritten digit classification method of the lecture on Cholesky factorization on a smaller data set (of 5000 examples) than used in the lecture. The data are available in the file `mnist.mat`. The file contains four variables: `Xtrain`, `Xtest`, `labels_train`, `labels_test`.

The variable `Xtrain` is a 5000×784 matrix X_{train} containing 5000 images. Each row is a 28×28 image stored as a vector of length 784. (To display the image in row i, use `imshow(reshape(Xtrain(i,:), 28, 28)')`.) The array `labels_train` is a vector of length 5000, with elements from $\{0, 1, \ldots, 9\}$. The ith element is the digit shown in row i of X_{train}. The 5000 images in X_{train} will be used to compute the classifier.

The matrix `Xtest` and vector `labels_test` are defined similarly, and give 5000 examples that will be used to test the classifier.

- *Binary classifiers.* The multi-class classification method is based on 10 binary classifiers. Each of the binary classifiers is designed to distinguish one of the digits versus the rest. We will use the polynomial kernel of degree 3, as in the lecture.

  To compute the classifier for digit k versus the rest, we solve a linear equation

  $$(Q + \lambda I)w = y,$$

  where Q is an $N \times N$ matrix ($N = 5000$) with elements

  $$Q_{ij} = (1 + x_i^T x_j)^3, \quad i, j = 1, \ldots, N,$$

  and x_i^T is the ith row of the matrix X_{train}. The coefficient λ is a positive regularization parameter. The right-hand side y is a vector of length N, with $y_i = 1$ if the image in row i of X_{train} is an example of digit k, and $y_i = -1$ otherwise.

  In MATLAB the coefficients w for all ten binary classifiers can be computed as follows.

```
load mnist;
[N, n] = size(Xtrain);
Y = -ones(N, 10);
for j = 1:10
    I = find(labels_train == j-1);
    Y(I, j) = 1;
end;
W = ( (1 + Xtrain * Xtrain').^3 + lambda * eye(N) ) \ Y;
```

  Here we first create an $N \times 10$ matrix Y with $Y_{ij} = 1$ if image i is an example of digit $j-1$ and $Y_{ij} = -1$ otherwise. The computed matrix W has size $N \times 10$ and contains in its jth column the coefficients w of the classifier for digit $j-1$ versus the rest. The binary classifier for digit $j-1$ can be evaluated at a new image z by computing

  $$\tilde{f}^{(j)}(z) = \sum_{i=1}^{N} W_{ij}(1 + z^T x_i)^3$$

  and assigning z to class j if $\tilde{f}^{(j)}(x)$ is greater than or equal to zero.

- *Multi-class classifier.* We combine the ten binary classifiers into a multi-class classifier by taking the maximum of the ten functions $\tilde{f}^{(j)}(z)$:

  $$\hat{f}(z) = \operatorname*{argmax}_{j=1,\ldots,10} \tilde{f}^{(j)}(z) = \operatorname*{argmax}_{j=1,\ldots,10} \left(\sum_{i=1}^{N} W_{ij}(1 + z^T x_i)^3 \right).$$

  In MATLAB, if z is an image stored as a column vector of length 784, the prediction $\hat{f}(z)$ can be computed as

```
[val, prediction] = max(( 1 + z' * Xtrain' ).^3) * W);
```

  On the right-hand side we compute the maximum of a row vector of length 10. The first output argument on the left-hand side is the maximum value; the second output argument is the column index of the maximum (an integer between 1 and 10).

  The predictions for all examples in the training set can be computed using

```
[val, prediction] = max(( 1 + Xtrain * Xtrain').^3) * W, [], 2);
```

  On the right-hand side we have a matrix of size 5000×10 as the first argument of `max`. The second and third arguments are needed to indicate that we are taking the maximum over the elements in each row. The first output argument on the left-hand side is a column vector of length N with the maximum value in each row; the second output argument is a column vector of length N with the column indices of the maximum elements in each row. This vector therefore contains the class predictions for the N rows of X_{train}. The command

```
I = find( prediction - 1 ~= labels_train )
```

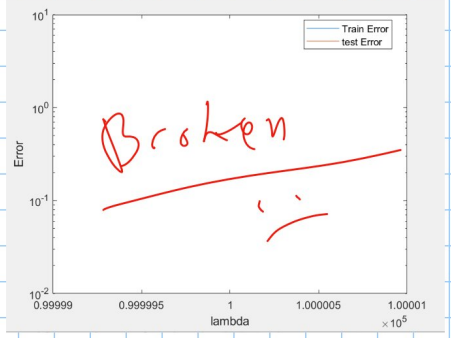
  returns the indices of the rows of X_{train} that are misclassified by the multi-class classifier.

  Similarly, we can compute the predictions for all examples in the test set using

```
[val, prediction] = max(( 1 + Xtest * Xtrain').^3) * W, [], 2);
```

Implement this method for a range of regularization parameters $\lambda = 1, 10, \ldots, 10^7$. Plot the error rate for training set and test set as a function of λ. Choose the λ that (approximately) gives the smallest error on the test set, and give the confusion matrix of the classifier for this value of λ.

```
disp(C)
     5    49   195     8    18    60   109    12     3     1
     3    59   248    10    18    45   174     9     4     1
     9    62   223     8     9    64   137    12     5     1
     7    64   195    13    14    52   141    13     1     0
     7    44   217     7     7    51   154    10     3     0
     8    39   209     7    12    40   124    14     3     0
    11    46   193     7     8    56   126    10     4     1
     1    68   206    11    12    55   148     8     3     0
     3    42   219    13     9    40   141    17     3     2
     3    54   228    18    11    58   132    12     3     1
```




```
%homework 7, Problem 4
load mnist;
[N, n] = size(Xtrain);
% lambda = [1 10, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7];
lambda = 10^5;
Y = -ones(N,10);
for j = 1:10
    I = find(labels_train == j-1);
    Y(I, j) = 1;
end;
W = ( (1 + Xtrain * Xtrain').^3 + lambda*eye(N) ) \ Y;
trainError = zeros(1,length(lambda));
testError = zeros(1,length(lambda));
for i = 1:length(lambda)
    W = ( (1 + Xtrain * Xtrain').^3 + lambda* eye(N) ) \ Y;
    [val, prediction] = max(((1+Xtrain*Xtrain') .^3)*W,[],2);
    indTrain = find(prediction - 1 ~= labels_train);
    trainError(i) = length(indTrain)*100/N;
    [val, prediction] = max(((1+Xtest*Xtrain').^3)*W, [],2);
    indTest = find(prediction -1 ~= labels_test);
    testError(i) = length(indTest)*100/N;
end;
figure;
semilogy(lambda, trainError);
hold on;
semilogy(lambda, testError);
xlabel('lambda');
ylabel('Classification Error');
legend('Train Error', 'test Error', 'location','best');
[minval, minInd] = min(testError);
W = ((1 + Xtest*Xtrain').^3 + lambda *eye(N)) \ Y;
[val, prediction] = max(((1+Xtest*Xtrain').^3) * W, [],2);
[C, order] = confusionmat(labels_test, prediction -1);

```

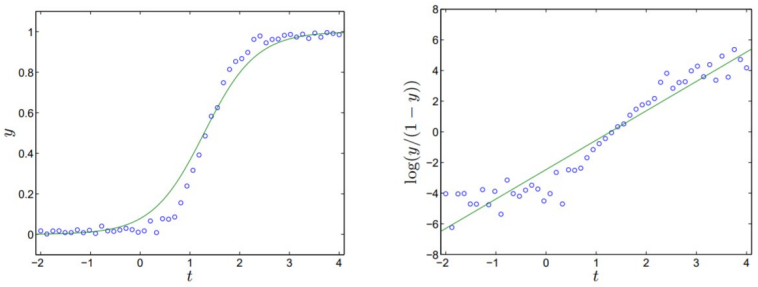
14.8 We revisit the data fitting problem of exercise 8.3. In that problem we used (linear) least squares to fit a function

$$f(t) = \frac{e^{\alpha t+\beta}}{1+e^{\alpha t+\beta}}$$

to 50 points (t_i, y_i). To formulate the problem as a least squares problem we applied a nonlinear transformation $\log(y/(1-y))$ (the inverse of the function f) to the points y_i, and minimized the function

$$\sum_{i=1}^{m}\left(\alpha t_i + \beta - \log(\frac{y_i}{1-y_i})\right)^2.$$

An example is shown in the following figure (for a different set of points than in exercise 8.3.



As can be seen in the left-hand figure, the quality of the fit is not uniform (better for y_i near 1 or 0 than at the other points). A second problem with this approach is that it requires that $0 < y_i < 1$ for all data points.

In this exercise we compute the true least squares fit on the original scale, by solving the optimization problem

$$\text{minimize} \quad \sum_{i=1}^{m}\left(\frac{e^{\alpha t_i+\beta}}{1+e^{\alpha t_i+\beta}} - y_i\right)^2. \tag{40}$$

This is a nonlinear least squares problem with two variables α, β.

Download the file `logistic_gn.m`, and execute in MATLAB as `[t, y] = logistic_gn;`. This is the set of 50 points used in the figures above. Solve the nonlinear least squares problem (40) using the Levenberg-Marquardt method. You can use as starting point the solution of the linear least squares method of exercise 8.3 (which applies because $0 < y_i < 1$ at all points), or simply take $\alpha = \beta = 0$. Terminate the iteration when $\|\nabla g(\alpha, \beta)\| \leq 10^{-6}$, where $g(\alpha, \beta)$ is the cost function in (40). Compare the solution with the result of the linear least squares method.

nonlinear least squares

minimize $\sum_{i=1}^{m} f_i(x)^2 = \|f(x)\|^2$

$(fg)' = f'g + g'f$

$(f/g)' = \frac{f'g - g'f}{g^2}$

$f = e^{\alpha t_i+\beta}$ $\quad g = 1 + e^{\alpha t_i+\beta}$

$f' = t_i e^{\alpha t_i+\beta}$ $\quad g' = t_i e^{\alpha t_i+\beta}$

$= \frac{t_i e^{\alpha t_i+\beta}}{(1+e^{\alpha t_i+\beta})^2}$

$f = e^{\alpha t_i+\beta}$ $\quad g = 1 + e^{\alpha t_i+\beta}$

$f' = e^{\alpha t_i+\beta}$ $\quad g' = e^{\alpha t_i+\beta}$

$= \frac{e^{\alpha t_i+\beta}}{(1+e^{\alpha t_i+\beta})^2}$

minimize $\sum_{i=1}^{m}\left(\frac{e^{\alpha t_i+\beta}}{1+e^{\alpha t_i+\beta}} - y_i\right) = \left\| \frac{e^{\alpha t_i+\beta}}{1+e^{\alpha t_i+\beta}} \right\|^2$ $\quad \therefore f(x) = \frac{e^{\alpha t_i+\beta}}{1+e^{\alpha t_i+\beta}} - y_i$

$\frac{\partial f}{\partial \beta} = \frac{e^{\alpha t_i+\beta}}{(1+e^{\alpha t_i+\beta})^2}$

$\frac{\partial f}{\partial \alpha} = \frac{t_i e^{\alpha t_i+\beta}}{(1+e^{\alpha t_i+\beta})^2}$

← Jacobian

$A = Df(x^{(k)}) = Df(\alpha, \beta)$

↑ Variables are α and β

$\|f(x) + Df(x)(z - x)\|^2 = \|Az - b\|^2$

$x^+ = z^* = (A^TA)^{-1}A^Tb$ newton method

```
%Homework 7, Problem 5
clear all;
[t,y] = logistic_gn;
m = length(t);

xls = [ t, ones(m,1)] \ log(y ./ (1-y));
ts = linspace(-2.1, 4.1, 1000);

x = xls;
%x = zeros(2,1);

for k = 1:100
    u = x(1)*t + x(2);
    f = exp(u) ./ (1+ exp(u) ) - y;
    A = diag(exp(u) ./ (1+exp(u)).^2) * [t, ones(m,1)];
    disp(['g = ', num2str(norm(f)^2), '. grad= ', num2str(norm(2*A'*f)) ]);
    if (norm(2*A'*f) < 1e-6)
        break;
    end;
    x = x - A \ f;
end;

ygn = exp(x(1)*ts+x(2)) ./ (1+ exp(x(1)*ts + x(2)));
yls  = exp(xls(1)*ts + xls(2)) ./ (1+ exp(xls(1)*ts + xls(2)));
plot(t,y,'o',ts, ygn, 'b-', ts, yls, 'r-');
axis([-1,4,0,1]);
```