Solutions, Homework 11

1. Prove the formula for the determinant of the Vandermonde matrix in the special case of 2×2 and 3×3:

   NOTE: This Vandermonde matrix is built in a different way than Matlab- in this case, start with the constant column (in Matlab, the constant column is last).

   SOLUTION: For a 2×2 matrix, the determinant of $\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \end{bmatrix}$ is simply $x_2 - x_1$

   For the 3×3 matrix, the solution is:

$$\begin{vmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{vmatrix} = \begin{vmatrix} 1 & x_1 & x_1^2 \\ 0 & x_2 - x_1 & x_2^2 - x_1^2 \\ 0 & x_3 - x_1 & x_3^2 - x_1^2 \end{vmatrix} = (x_2 - x_1) \begin{vmatrix} 1 & x_1 & x_1^2 \\ 0 & 1 & x_2 + x_1 \\ 0 & x_3 - x_1 & x_3^2 - x_1^2 \end{vmatrix} =$$

$$(x_2 - x_1)(x_3 - x_1) \begin{vmatrix} 1 & x_1 & x_1^2 \\ 0 & 1 & x_2 + x_1 \\ 0 & 1 & x_3 + x_1 \end{vmatrix} = (x_2 - x_1)(x_3 - x_1) \begin{vmatrix} 1 & x_1 & x_1^2 \\ 0 & 1 & x_2 + x_1 \\ 0 & 0 & x_3 - x_2 \end{vmatrix} =$$

$$(x_2 - x_1)(x_3 - x_1)(x_3 - x_2) \begin{vmatrix} 1 & x_1 & x_1^2 \\ 0 & 1 & x_2 + x_1 \\ 0 & 0 & 1 \end{vmatrix} = (x_2 - x_1)(x_3 - x_1)(x_3 - x_2)$$

2. Write a Matlab script for classifying the iris data. Here is the script file using the parameters specified online:

```
%Script file to do the RBF by hand (classification of the iris data)

load IrisData

dd=randperm(150);  %Use these random numbers for the index of the centers
Centers=X(dd(1:10),:);  %Use 10 centers (as stated in the problem)

Xtrain=X(dd(11:90),:);  %Use 80 points for training
Ytrain=Y(dd(11:90),:);  % Keep X, Y together!

U=edm(Centers,Xtrain);  %U is 10 x 80, number of centers x number pts
Phi=rbf1(U,1,3);        %Use the Gaussian

% Although not specified in the HW, we might include a bias term by adding
% a row of 1's:
Phi=[Phi; ones(1,80)];

%Solve W_hat * Phi_hat = Y for W_hat by using the pseudo-inverse
%
%  We can build the pseudo-inverse by hand if we wanted:

[U1,S1,V1]=svd(Phi,'econ');
```

```
iS1=diag(1./diag(S1));  %Computes the inverse of Sigma

W=Ytrain'*(V1*(iS1*U1'));

% Test our RBF using all the data (easier than just the test set)
U=edm(Centers,X);
Phi=rbf1(U,1,3);
Phi=[Phi;ones(1,150)];
Yout=W*Phi;

% Confusion matrix construction:
Confuse=zeros(3,3);
for j=1:150
    yout=Yout(:,j)';
    Class(j,:)=(yout==max(yout));
    [val,idx1]=max(Class(j,:));

    %Compare this class to the actual class:
    [val,idx2]=max(Y(j,:));
    Confuse(idx2,idx1)=Confuse(idx2,idx1)+1;
end
% Divide by the number of training points in each class (50 each):
Confuse=Confuse/50;
```