Contents


```
% Endsem CH5440
% Ojas Phadake - CH22B007

clc;
clear all;
close all;

load VLEdata.mat

% Implementing for 1st 9 samples of liquid mol fraction and Measured Dew
% Point Pressure
```

A

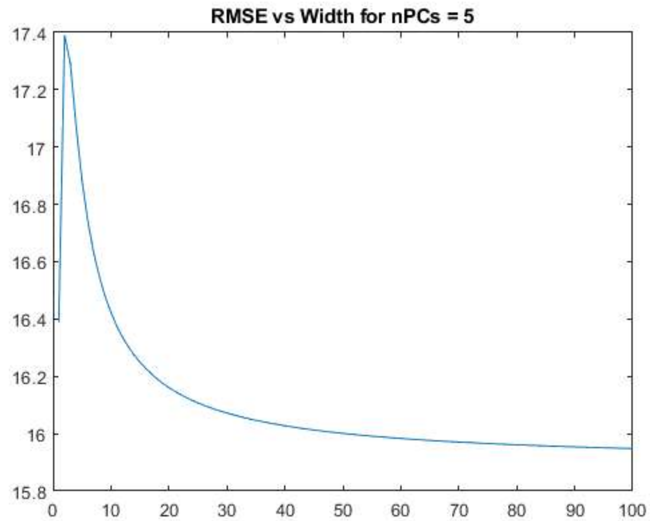
```
Pmeas= Pmeas';
xtrain = liqmf(1:9);
ptrain = Pmeas(1:9, :);

y1meas = y1meas';
ptrain2 = y1meas(1:9, :);

% The following elements have been kept for cross validation and testing
xtest = liqmf(10:13);
ptest = Pmeas(10:13, :);

% Shift and scale x data
xmean = mean(xtrain);
xstd = std(xtrain);
xs = (xtrain - xmean*ones(size(xtrain)))/xstd; % Standardized inputs
nsamples = length(xtrain);

%  Shift and scale test data exactly as we used for training data
ntest = length(xtest);
xtest = (xtest - xmean*ones(size(xtest)))/xstd;

% Predicting only the Dew point pressure:

widths = 1:1:100;
maxnPC = 10;
```

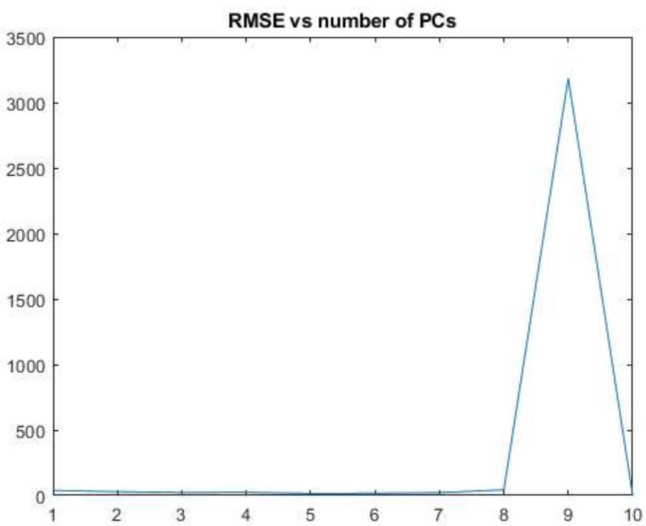
B

```
fprintf("My method of solving is to assume a certain number of PCs, eg 5 and then find the most optimal width. " + ...
    "After the most optimal width is found out, then find it wrt the best PC number and implement that. \nI agree" + ...
    "that the best method will be to find for each PC from 1 to 10 and for each width from 1 to 100 the minimum value" + ...
    "of PRESS/RMSE and then the least will give us the best answer/most optimal solution.")
PRESS = zeros(100, 1);
RMSE = zeros(100, 1);
% Here, each will have the RMSE for a given width with nPC = 5

for w=1:100
    % We must find the width which has the least PRESS/RMSE in the tested
    % Pmeas values which we will estimate and then choose the least one

    % We are essentially calculating the psat given the temperature (all)
    % and the xtest values of mol fractions

    K = zeros(nsamples,nsamples);
    width = widths(w);

    for i = 1:nsamples
        for j = i:nsamples
            diff = xs(i)-xs(j);
            K(i,j) = exp(-diff'*diff/width);  % Gaussian Kernel
            K(j,i) = K(i,j);
        end
    end

    [U D] = eig(K);


        error = zeros(1, 8);
        nfact = nsamples-5+1;
        eval = diag(D);
        lamda = eval(nfact:nsamples);
        Pc = U(:,nfact:nsamples);
```

```
        T = K*Pc*diag(lamda.^(-0.5));
        B = inv(T'*T)*T'*ptrain;

        Ktest = zeros(1,nsamples);
        for i = 1:ntest
            for j = 1:nsamples
                diff = xtest(i) - xs(j);
                Ktest(j) = exp(-diff'*diff/width);
            end

            psatest = Ktest*Pc*diag(lamda.^(-0.5))*B;
            error = error + (ptest(i, :)-psatest).^2;
        end

        PRESS(w) = sum(error)/ntest;
        RMSE(w) = sqrt(PRESS(w));

end

figure(1)
plot(RMSE)
title("RMSE vs Width for nPCs = 5")

fprintf("We see that the value of RMSE begins to gradually reduce after width = 20\n" + ...
    "Hence, we select the width = 20")

width = 20;

PRESS = zeros(maxnPC,1);
RMSE = zeros(maxnPC,1);


for p=1:maxnPC-1
        K = zeros(nsamples,nsamples);

    for i = 1:nsamples
        for j = i:nsamples
            diff = xs(i)-xs(j);
            K(i,j) = exp(-diff'*diff/width);  % Gaussian Kernel
            K(j,i) = K(i,j);
        end
    end

    [U D] = eig(K);

        error = zeros(1, 8);
        nfact = nsamples-p+1;
        eval = diag(D);
        lamda = eval(nfact:nsamples);
        Pc = U(:,nfact:nsamples);
        T = K*Pc*diag(lamda.^(-0.5));
        B = inv(T'*T)*T'*ptrain;

        Ktest = zeros(1,nsamples);
        for i = 1:ntest
            for j = 1:nsamples
                diff = xtest(i) - xs(j);
                Ktest(j) = exp(-diff'*diff/width);
            end

            psatest = Ktest*Pc*diag(lamda.^(-0.5))*B;
            error = error + (ptest(i, :)-psatest).^2;
        end

        PRESS(p) = sum(error)/ntest;
        RMSE(p) = sqrt(PRESS(p));
end

figure(2)
plot(1:1:10, RMSE)
title("RMSE vs number of PCs")

fprintf("So, we can use any number from 1 to 8 PCs when the width has been chosen as such")
```

```
My method of solving is to assume a certain number of PCs, eg 5 and then find the most optimal width. After the most optimal width is found out, then find it wrt th
I agreethat the best method will be to find for each PC from 1 to 10 and for each width from 1 to 100 the minimum valueof PRESS/RMSE and then the least will give us
Hence, we select the width = 20Warning: Matrix is close to singular or badly scaled. Results may be
inaccurate. RCOND =  6.950987e-20.
So, we can use any number from 1 to 8 PCs when the width has been chosen as such
```



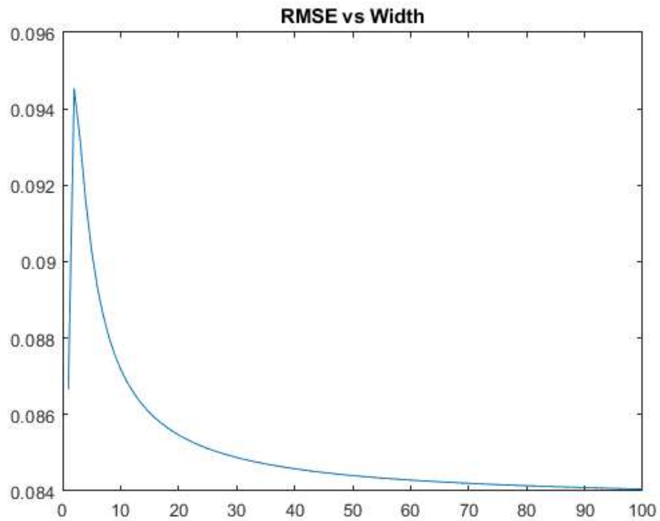


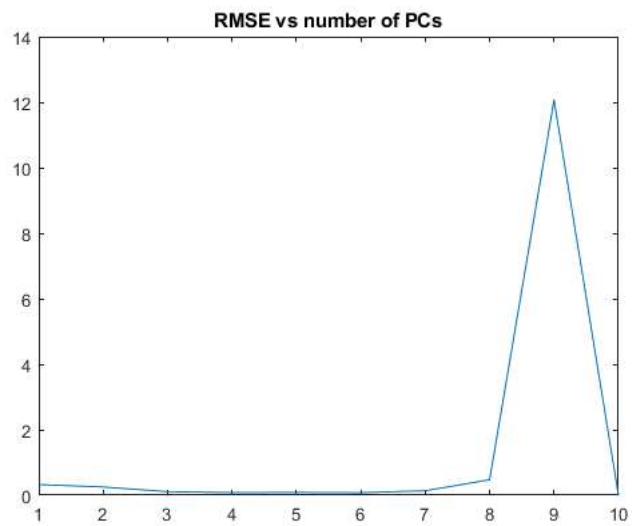
Carrying out the same for predicting vapour phase comnposition

```
ptrain = y1meas(1:9, :);
ptest = y1meas(10:13, :);

PRESS = zeros(100, 1);
RMSE = zeros(100, 1);
% Here, each will have the RMSE for a given width with nPC = 5

for w=1:100
    % We must find the width which has the least PRESS/RMSE in the tested
    % Pmeas values which we will estimate and then choose the least one
    % We will also tabulate the value for 10 PCs for each Kernel

    % We are essentially calculating the psat given the temperature (all)
    % and the xtest values of mol fractions

    K = zeros(nsamples,nsamples);
    width = widths(w);

    for i = 1:nsamples
        for j = i:nsamples
            diff = xs(i)-xs(j);
            K(i,j) = exp(-diff'*diff/width);  % Gaussian Kernel
            K(j,i) = K(i,j);
        end
    end

    [U D] = eig(K);


        error = zeros(1, 8);
        nfact = nsamples-5+1;
        eval = diag(D);
```

```
        lamda = eval(nfact:nsamples);
        Pc = U(:,nfact:nsamples);

        T = K*Pc*diag(lamda.^(-0.5));
        B = inv(T'*T)*T'*ptrain;

        Ktest = zeros(1,nsamples);
        for i = 1:ntest
            for j = 1:nsamples
                diff = xtest(i) - xs(j);
                Ktest(j) = exp(-diff'*diff/width);
            end

            psatest = Ktest*Pc*diag(lamda.^(-0.5))*B;
            error = error + (ptest(i, :)-psatest).^2;
        end

        PRESS(w) = sum(error)/ntest;
        RMSE(w) = sqrt(PRESS(w));

end

figure(3)
plot(RMSE)
title("RMSE vs Width")

fprintf("We see that the value of RMSE begins to gradually reduce after width = 20\n" + ...
    "Hence, we select the width = 20")

width = 20;

PRESS = zeros(maxnPC,1);
RMSE = zeros(maxnPC,1);


for p=1:maxnPC-1
        K = zeros(nsamples,nsamples);

    for i = 1:nsamples
        for j = i:nsamples
            diff = xs(i)-xs(j);
            K(i,j) = exp(-diff'*diff/width);  % Gaussian Kernel
            K(j,i) = K(i,j);
        end
    end

    [U D] = eig(K);

        error = zeros(1, 8);
        nfact = nsamples-p+1;
        eval = diag(D);
        lamda = eval(nfact:nsamples);
        Pc = U(:,nfact:nsamples);
        T = K*Pc*diag(lamda.^(-0.5));
        B = inv(T'*T)*T'*ptrain;

        Ktest = zeros(1,nsamples);
        for i = 1:ntest
            for j = 1:nsamples
                diff = xtest(i) - xs(j);
                Ktest(j) = exp(-diff'*diff/width);
            end

            psatest = Ktest*Pc*diag(lamda.^(-0.5))*B;
            error = error + (ptest(i, :)-psatest).^2;
        end

        PRESS(p) = sum(error)/ntest;
        RMSE(p) = sqrt(PRESS(p));
end

figure(4)
plot(1:1:10, RMSE)
title("RMSE vs number of PCs")

fprintf("So, we can use any number from 1 to 8 PCs when the width has been chosen as such")
```

```
We see that the value of RMSE begins to gradually reduce after width = 20
Hence, we select the width = 20Warning: Matrix is close to singular or badly scaled. Results may be
inaccurate. RCOND =  6.950987e-20.
So, we can use any number from 1 to 8 PCs when the width has been chosen as such
```





Published with MATLAB® R2023a