
Exercise 3-1, c): KNN Regression

Felix Wittich, 28.06.2022

```
clear; close all; clc;
```

1c) Leave-One-Out Cross-Validation with KNN regression

```
rng(5); % Fix random number generator for reproducible results

% Generate non-linear data
N = 50;
noise_level = 1;
[x,y] = generate_nonlin_data_1D(N,noise_level); % Generate non-linear data

% For-loop to implement LOOCV for the MSE with kNN regression of the
% generated data set
MSE_LOOCV = zeros(10,1);
for j = 1:10 % Loop over different k

    MSE = zeros(50,1);
    for i = 1:N % Loop over all data points
        ytrain = y([1:i-1 i+1:end]); % training set in each iteration
        xtrain = x([1:i-1 i+1:end]); % training set in each iteration
        ytest = knn_regression(x(i),ytrain,xtrain,j);
        MSE(i) = mean(y(i)-ytest)^2; % calculate MSE in each iteration
    end

    MSE_LOOCV(j) = mean(MSE); % calculate mean MSE

end

% Plot MSE_LOOCV for different k values
figure
plot([1:10],MSE_LOOCV)
xlabel('k')
ylabel('MSE_{LOOCV}')

% Implementation of the basic kNN algorithm
function y_pred = knn_regression(x_train,y_train,x0,k)

x = x_train;
y = y_train;

% Calculate pairwise euclidean distance
dist = sum(sqrt((x - x0).^2),2);
% dist = pdist([repmat(x0,size(x,1),1) x],'euclidean');

% Sort the distances in ascending order (use the in-built function 'sort')
```

```
[sdist,ind] = sort(dist);

% Find the k closest output values
ynn = y(ind(1:k),:);

% Calculate the mean of the k closest output values
y_pred = (1/k)*(sum(ynn));

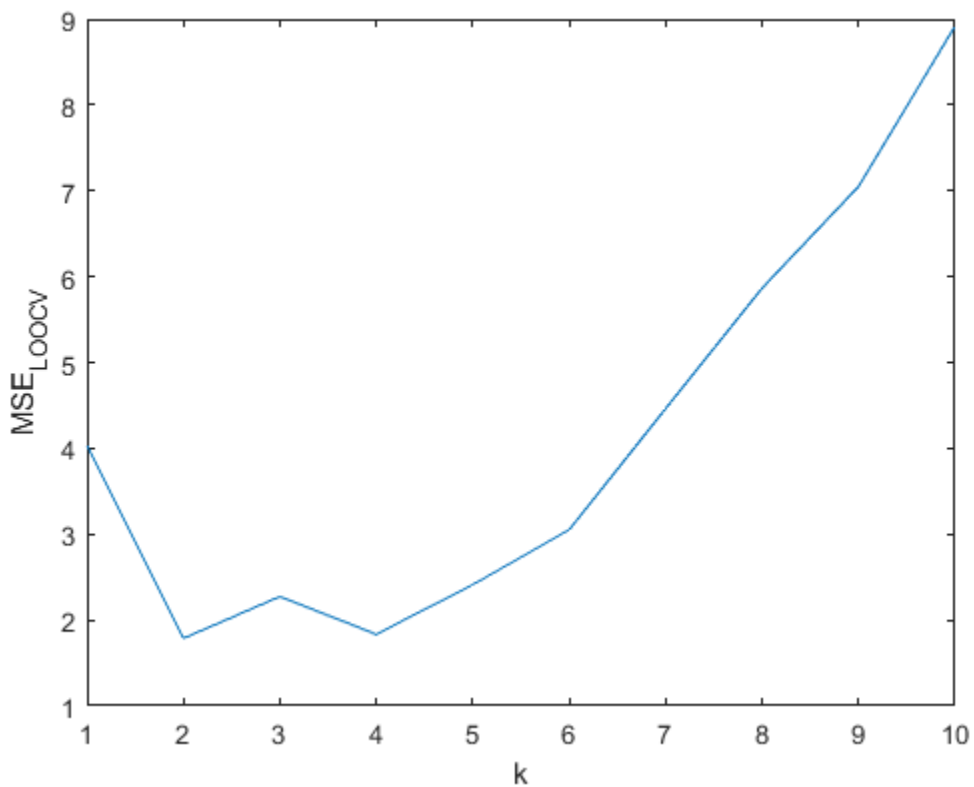
end

% Data generating function
function [x,y] = generate_nonlin_data_1D(N,nl)

% 1D Nonlinear test function (http://www.sfu.ca/~ssurjano/forretal08.html)

x = linspace(0,1,N)';
y = ((6*x-2).^2).*sin(12*x-4) + nl*randn(N,1);

end
```



Published with MATLAB® R2021b