Table of Contents


```
% Austin Welch
% EC503 HW7.2
% Ordinary Least Squares (OLS) versus Robust Linear Regression
```

load data

```
clear; clc;
rng default; % For reproducibility
load('linear_data.mat');
```

(a)

Implement ordinary least squares (OLS) linear regression for input (xdata) and output (ydata) provided in the linear_data.mat file. Do not use MATLAB's robustfit or regress. (i) Will the input data matrix (xdata) yield a unique solution? Why or why not? (ii) If h_OLS(x) = x^T*W_OLS + b_OLS, report the values of w_OLS and b_OLS and the resulting mean-squared error (MSE) and mean absolute error (MAE)

```
fprintf('Part (a):\n\n');
```

```
Part (a):
```

Normal equation:

$\theta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ (n << 10000 so normal equations should outperform gradient descent)

```
xData = xData';
yData = yData';
xDataExt = [xData; ones(1,length(xData))];
theta_ols = inv(xDataExt*xDataExt')*xDataExt*yData';
w_ols = theta_ols(1);
b_ols = theta_ols(2);

% uniqueness of solution
fprintf(['There should be a unique solution to the input data matrix
\n',...
    'xdata because (X*X^T) is non-singular/invertible/full-rank.\n
\n']);
```

```
% parameters
fprintf('w_ols: %0.4f\n', w_ols); % slope
fprintf('b_ols: %0.4f\n\n', b_ols); % intercept

% decision rule
h_ols = xData*w_ols+b_ols;

% quantify error
MSE_ols = sum((yData-h_ols).^2)/length(yData);
MAE_ols = sum(abs(yData-h_ols))/length(yData);
fprintf('MSE_ols: %0.4f\n', MSE_ols);
fprintf('MAE_ols: %0.4f\n\n', MAE_ols);
```

```
There should be a unique solution to the input data matrix
xdata because (X*X^T) is non-singular/invertible/full-rank.

w_ols: 0.0234
b_ols: 2.9738

MSE_ols: 0.2588
MAE_ols: 0.3175
```

visualize OLS regression line

```
%{
figure(1);
scatter(xData,yData);
hold on;
plot(xData,h_ols)
%}
% to check: above line and 'lsline' function are collinear
%lsline
```

(b)

Use MATLAB's robustfit function to implement robust linear regression for the input and output in linear_data.mat. There are two outliers in the dataset which skew the OLS regression model because it is not robust to outliers.

```
fprintf('Part (b):\n\n');

% loss functions (const, coefficient) with default tuning constants
cauchy = robustfit(xData,yData,'cauchy'); % w = 1 ./ (1 + r.^2)
fair = robustfit(xData,yData,'fair'); % w = 1 ./ (1 + abs(r))
huber = robustfit(xData,yData,'huber'); % w = 1 ./ max(1, abs(r))
talwar = robustfit(xData,yData,'talwar'); % w = 1 * (abs(r)<1)
ols = robustfit(xData,yData,'ols'); % same as (a), repeated for
 consistency

% estimates
hCauchy = xData*cauchy(2) + cauchy(1);
hFair = xData*fair(2) + fair(1);
```

```
hHuber = xData*huber(2) + huber(1);
hTalwar = xData*talwar(2) + talwar(1);
hOls = xData*ols(2) + ols(1);

% MSEs
MSEcauchy = mse(yData,hCauchy);
MSEfair = mse(yData,hFair);
MSEhuber = mse(yData,hHuber);
MSEtalwar = mse(yData,hTalwar);
MSEols = mse(yData,hOls);

% MAEs
MAEcauchy = mae(yData,hCauchy);
MAEfair = mae(yData,hFair);
MAEhuber = mae(yData,hHuber);
MAEtalwar = mae(yData,hTalwar);
MAEols = mae(yData,hOls);

% Compare robust errors to OLS error
Cauchy = [MSEcauchy; MAEcauchy];
Fair = [MSEfair; MAEfair];
Huber = [MSEhuber; MAEhuber];
Talwar = [MSEtalwar; MAEtalwar];
OLS = [MSEols; MAEols];
T = table(Cauchy, Fair, Huber, Talwar, OLS, 'RowNames',
 {'MSE', 'MAE'});
disp(T);
fprintf('OLS has lowest MSE, but highest MAE\n\n');

% report values of wHuber and bHuber
fprintf('w_huber: %0.4f\n', hHuber(2));
fprintf('b_huber: %0.4f\n\n', hHuber(1));

% plot ols and robust methods to compare
figure(2);
scatter(xData,yData,'filled'); grid on; hold on
plot(xData, hOls, 'k', 'LineWidth', 1);
plot(xData, hCauchy, 'b', 'LineWidth', 1);
plot(xData, hFair, 'g', 'LineWidth', 1);
plot(xData, hHuber, 'r', 'LineWidth', 1);
plot(xData, hTalwar, 'y', 'LineWidth', 1);
legend({'Data','OLS', 'Cauchy', 'Fair', 'Huber', 'Talwar'},'Box','off')
set(legend,'position',[0.15  0.12  0.1286  0.3])
set(gca,'fontsize',7)
title('Comparison of OLS to robust linear regression loss
 functions', ...
    'fontsize',7);
xlabel('X'); ylabel('Y');

% observations
fprintf(['The Cauchy loss function has the lowest MAE. All of the
\n', ...
    'robust loss functions are much less sensitive to the outliers
\n',...
```

```
    'compared to ordinary least squares regression.\n\n']);
```

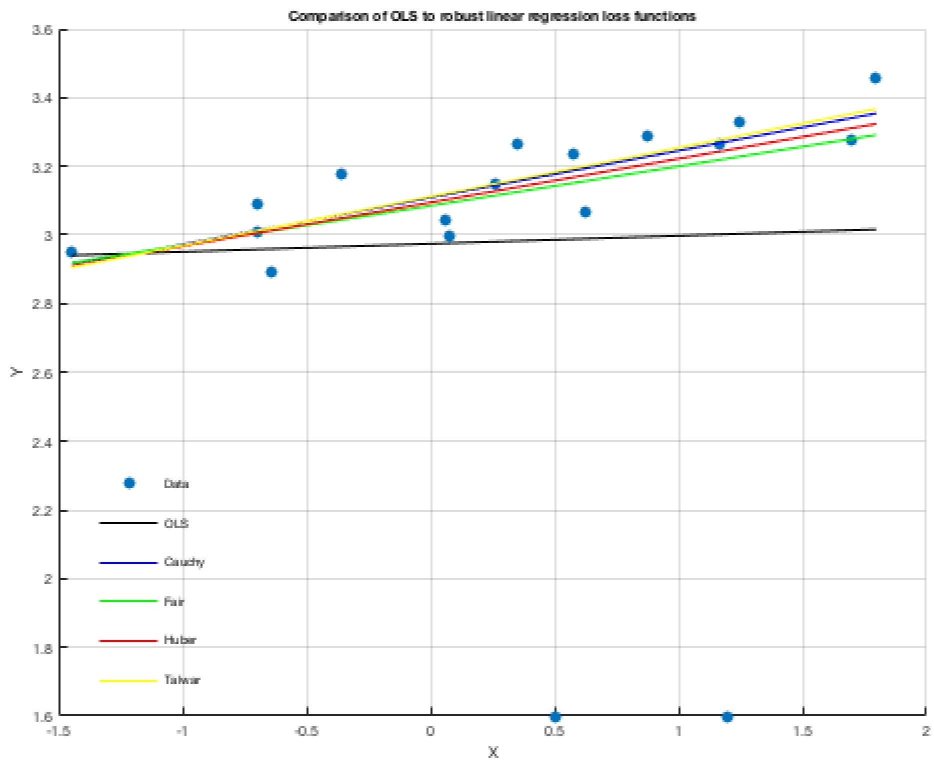
Part (b):

	Cauchy	Fair	Huber	Talwar	OLS
MSE	0.29955	0.28608	0.29218	0.30242	0.25884
MAE	0.24268	0.24683	0.24509	0.24349	0.31747

OLS has lowest MSE, but highest MAE

w_huber: 3.1743
b_huber: 3.2429

The Cauchy loss function has the lowest MAE. All of the robust loss functions are much less sensitive to the outliers compared to ordinary least squares regression.



Published with MATLAB® R2017a