Jia Wen Lee

ABE516x: Linear Regression using own dataset.

MATLAB was used for this task

Introduction:

I have data from LiDAR that consist of relative signal strength index (RSSI), distance and LiDAR beam angle. I know that RSSI value is affected by scan distance. RSSI value will be used as a metric for soil surface feature as such I would like RSSI values to be independent of distance. Linear regression is used to see if I can get a good model for RSSI and distance in order to correct RSSI value base on distance.

Data:

linFit.Rsquared | rssiDistData

1x102 struct with 7 fields

Fields	folder	name	index	height	polDist	rssi	angle
1	'T:\current\...	'2019-06-17...	1	50	1x173520 do...	1x173520 do...	1x173520 do...
2	'T:\current\...	'2019-06-17...	2	50	1x173520 do...	1x173520 do...	1x173520 do...
3	'T:\current\...	'2019-06-17...	3	50	1x173520 do...	1x173520 do...	1x173520 do...
4	'T:\current\...	'2019-06-17...	1	600	1x113362 do...	1x113362 do...	1x113362 do...
5	'T:\current\...	'2019-06-17...	2	600	1x113362 do...	1x113362 do...	1x113362 do...
6	'T:\current\...	'2019-06-17...	3	600	1x113363 do...	1x113363 do...	1x113363 do...
7	'T:\current\...	'2019-06-17...	1	800	1x107746 do...	1x107746 do...	1x107746 do...
8	'T:\current\...	'2019-06-17...	2	800	1x107746 do...	1x107746 do...	1x107746 do...
9	'T:\current\...	'2019-06-17...	3	800	1x107746 do...	1x107746 do...	1x107746 do...
10	'T:\current\...	'2019-06-17...	1	1000	1x116785 do...	1x116785 do...	1x116785 do...
11	'T:\current\...	'2019-06-17...	2	1000	1x116785 do...	1x116785 do...	1x116785 do...
12	'T:\current\...	'2019-06-17...	3	1000	1x116784 do...	1x116784 do...	1x116784 do...

Code:

Data was filtered to keep only LiDAR scan values that were close to perpendicular to the scan surface to remove possible effect of scan angle. Mean polar distance and RSSI of each dataset was calculated

```
%Get mean of each data set where incident angle is bewteen 89.5 and 90.5
%and remove outlier from infrared beam reflections at the edge of the test bed.
for i=1:length(rssiDistData)
    mask1=(rssiDistData(i).angle<=90.5 & rssiDistData(i).angle>=89.5);
    mask2=rssiDistData(i).polDist>1000;
    outlierMask=rssiDistData(i).polDist>3500 & rssiDistData(i).rssi>122;

    dist(i)=mean(rssiDistData(i).polDist(mask1 & mask2 & ~outlierMask));
    rssi(i)=mean(rssiDistData(i).rssi(mask1 & mask2 & ~outlierMask));
end
```

Linear model was fitted onto the data and confidence interval was calculated from the linear model.

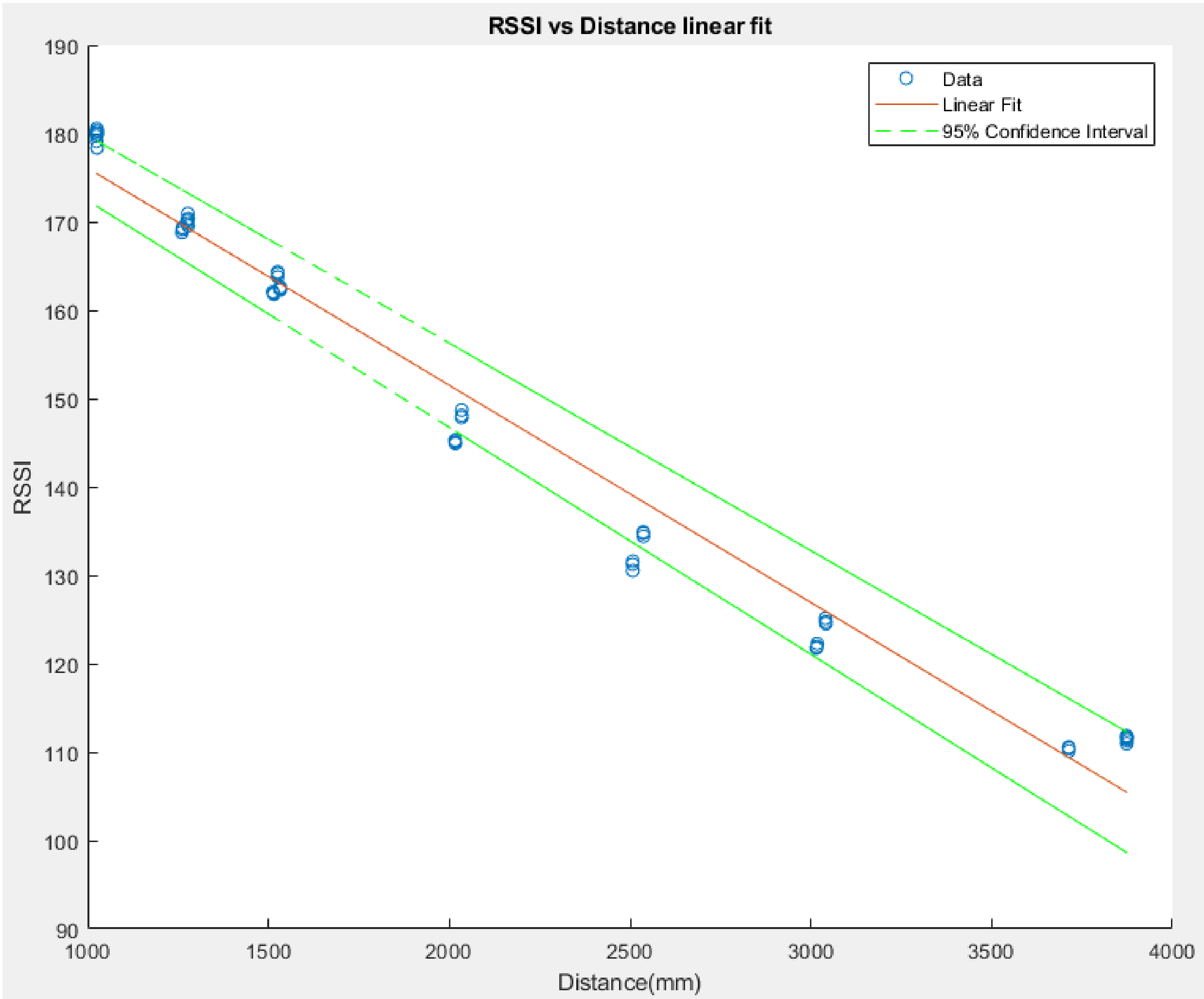
```
%Linear fit
linFit=fitlm(dist,rssi);

%Get confidence interval
CI=coefCI(linFit);
CI_L=CI(2,1)*dist+CI(1,1);
CI_H=CI(2,2)*dist+CI(1,2);
```

Figure of data, linear fit and confidence interval was plotted.

```
% Plot data, linear fit and confidence interval
figure
scatter(dist,rssi)
hold on
plot(dist,linFit.Fitted)
plot(dist,CI_L, 'g--')
plot(dist,CI_H, 'g--')
xlabel('Distance(mm)')
ylabel('RSSI')
legend({'Data','Linear Fit','95% Confidence Interval'})
title('RSSI vs Distance linear fit')
```

Result:



R-Squared Value

linFit.Rsquared | rssiDistData | linFit

linFit.Rsquared

Field	Value
Ordinary	0.9748
Adjusted	0.9743

Discussion:

R-Squared value was used to determine how well model fits the data. R-Squared value is high which means this model fits data well. Confidence interval captures quite a large portion of the data as well. Though R-Squared value was high, it can be visually seen that a second order model might work better than a linear model.