Robust Regression: Campbell's Method & Monte Carlo Simulation

A PROJECT PRESENTATION BY :

SIDHARTH KUMAR 16BM6P45

RAHUL KUSHWAHA 16BM6JP30

GAURAV KUNAL JAISWAL *13AE30022*

Introduction

Problem : Weights of weighted regression

Objective: To find the outliers and determine the weights for the regression.

Approach : Campbell’s Robust Covariance Matrix and Monte Carlo Simulations

Methodology

Campbell’s Robust Covariance Matrix

- Weighted mean vector and covariance matrix.
- Mahanalobis distance as a measure of deviation from the center.
- Update the weight in each iteration.
- Use Moore –Penrose inverse incase matrix is not invertible.

Detect outliers

- Using the weights

Use weights to do weighted least square regression.

Verification using Monte Carlo Simulations

Campbell’s Method

Initialize weights to 1

Define b1 = 2, b2 = 1.25, m = number of features, n = number of data points

Define d_0 = sqrt (m) + b_1 / sqrt(2)

Repeat (till 1000 steps or cosine similarity between (ω_{new}, ω_{old}) = 1)

{

$$\overline{\boldsymbol{x}} = \sum_{i=1}^{n} \omega_i x_i \,/\, \sum_{i=1}^{n} \omega_i$$

$$S = \sum_{i=1}^{n} \omega_i^2 (x_i - \overline{x})' \,(x_i - \overline{x}) / [\sum_{i=1}^{n} \omega_i^2 - 1]$$

$$d_i = \{(x_i - \overline{x}) S^{-1} (x_i - \overline{x})'\}^{\;1/2}$$

$$\omega_i = \frac{\omega(d_i)}{d_i}; i = 1, n\text{: } \omega(d_i) = d_i \text{ if } d_i < d_0 \text{ else } \omega(d_i) = d_0 \exp[-\frac{0.5(d_i - d_0)^2}{b_2^2}]$$

}

Application

Data set generation

- $y = =8.7+15.5*X_1+0.1*X_2+4.6*X_3+0.3*X_4+11.5*X_5+ \varepsilon$
- Outliers data points injected = $9, 23, 35$

Perform ordinary least square regression

Perform Campbell's Robust Regression method

- Find outliers
- Regression coefficients

Results

OLS Coefficients :

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 197.1820    44.8822   4.393 6.94e-05 ***
x1m           5.0026     1.3421   3.727 0.000549 ***
x2m           0.8435     2.1933   0.385 0.702401    
x3m           1.4887     1.2849   1.159 0.252844    
x4m          -3.4928     1.3232  -2.640 0.011436 *  
x5m          11.6659     2.5865   4.510 4.77e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 17.34 on 44 degrees of freedom
Multiple R-squared:  0.5661,	Adjusted R-squared:  0.5168 
F-statistic: 11.48 on 5 and 44 DF,  p-value: 3.973e-07
```

Robust Regressions Coefficients:

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  8.92021    2.78368   3.204  0.00252 ** 
x1m         15.48101    0.09324 166.027  < 2e-16 ***
x2m          0.10102    0.09248   1.092  0.28062    
x3m          4.59987    0.10845  42.416  < 2e-16 ***
x4m          0.29719    0.09444   3.147  0.00296 ** 
x5m         11.50014    0.10909 105.421  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7275 on 44 degrees of freedom
Multiple R-squared:  0.9992,	Adjusted R-squared:  0.9991 
F-statistic: 1.143e+04 on 5 and 44 DF,  p-value: < 2.2e-16
```

Outliers points detected: 9, 23, 35

- Corresponding Weights after 1000 iterations : 7.623219e-04, 2.591972e-12, 2.011003e-16

Monte Carlo Simulations

Artificially generated 40 points using **$Y = 80-16*X_1 +12*X_2 -2*X_3 + \varepsilon$**

- Range of (X_1, X_2 , X_3) belongs to (-10, 50)

Experiment 1:

- Added one quantum of random size between (-10,-5) and (5, 10) to equi-probably randomly chosen point of every variable.
- Repeat 200 times and find mean and standard deviations of estimates of coefficients .
- Repeat with 2, 3, 5 perturbation quanta.

Experiment 2:

- Repeat Experiment 1 with only change in quantum of random size between (-25, -20) and (20, 25)

Experiment 3:

- Repeat Experiment 1 with only change in quantum of random size between (-100, -50) and (50, 100)

Observations

NO	Perturbation	B_0	B_1	B_2	B_3	$S(B_0)$	$S(B_1)$	$S(B_2)$	$S(B_3)$	RMSE0	RMSE1	RMSE2	RMSE3
	Experiment 1												
1	(-10,-5) and (5,10)	79.5030211	-15.99235	11.99437	-1.988995	0.66909	0.01393	0.01309	0.01418	0.68861	0.01413	0.01347	0.01418
2	(-10,-5) and (5,10)	79.8475179	-15.9852	11.983762	-1.99504	0.628443	0.019682	0.018661	0.019324	0.625149	0.019367	0.017902	0.019904
5	(-10,-5) and (5,10)	79.6339857	-16.00155	12.001344	-1.993572	1.2055	0.03225	0.03771	0.03096	1.20492	0.30297	0.03722	0.03143
10	(-10,-5) and (5,10)	79.6384088	-16.01002	12.028212	-1.997948	1.287027	0.037646	0.041067	0.041088	1.333755	0.038867	0.049739	0.041036
	Experiment 2												
1	(-25,-20) and (20,25)	80.5519401	-16.01249	11.997156	-1.996616	1.294308	0.038599	0.040683	0.03864	1.4041	0.040477	0.04068	0.038692
2	(-25,-20) and (20,25)	80.122677	-16.00809	11.99709	-2.002984	1.743259	0.049281	0.044269	0.055807	1.743217	0.049819	0.044254	0.055747
5	(-25,-20) and (20,25)	80.3706982	-16.02217	11.99226	-1.990942	4.62569	0.109905	0.091367	0.111869	4.628979	0.111849	0.091466	0.111956
10	(-25,-20) and (20,25)	79.0139706	-15.98989	12.037518	-1.998332	4.656686	0.12377	0.138297	0.108376	4.748532	0.123873	0.142962	0.108118
	Experiment 3												
1	(-100, -50) and (50, 100)	82.4074588	-16.00272	12.016773	-1.999521	15.1667	0.374025	0.402444	0.357036	15.31909	0.373099	0.401787	0.356142
2	(-100, -50) and (50, 100)	85.2187083	-16.02119	12.058956	-1.996796	17.18047	0.588564	0.613068	0.570512	17.91445	0.587473	0.614368	0.569093
5	(-100, -50) and (50, 100)	92.3952885	-15.9214	11.97248	-2.00558	33.93847	0.958865	1.018982	1.0412	36.0514	0.959689	1.016804	1.038609
10	(-100, -50) and (50, 100)	102.592314	-16.07592	12.059889	-2.068453	42.39692	1.18309	1.189815	1.061588	47.9471	1.182568	1.188347	1.061141

Inferences

- For small perturbations Campbell's robust estimator perform well.
- Even on increasing the size of perturbation robust estimator does fairly well but becomes biased.
- Also upon increasing number of perturbation to 10, we corrupt 35% of the data set. Considering size of (X_1, X_2 , X_3) is between (-10, 50), a perturbation of (-100, -50) and (50, 100) is large.

Conclusion

- Effect of outliers while doing OLS
- Introduced the Campbell’s robust method (Iterative method)
- Properties of Campbell’s Estimator using Monte Carlo Simulation.