**EX1：**

**文本, 信件

描述已自动生成**

**EX2:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **sparsity lasso** | **sparsity ridge** | **RSS lasso** | **RSS ridge** | **norm size lasso** | **norm size ridge** |
| **lambda=0** | **0** | **0** | **32.6535** | **32.6535** | **0.95051** | **0.57572** |
| **lambda=0.1** | **0** | **0** | **32.8184** | **32.6622** | **0.785** | **0.56005** |
| **lambda=1** | **0** | **0** | **32.8155** | **32.6622** | **0.78642** | **0.56005** |
| **lambda=10** | **0.6** | **0** | **36.7525** | **33.1605** | **0.34661** | **0.46138** |
| **lambda=100** | **1** | **0** | **48.9935** | **39.3527** | **0.00012713** | **0.18981** |

**图表, 瀑布图

描述已自动生成**

**Fig 1：lambda=0**

**图表, 瀑布图

描述已自动生成**

**Fig 2：lambda=0.1**

**图表, 瀑布图

描述已自动生成**

**Fig 3: lambda=1**

**图表, 直方图, 瀑布图

描述已自动生成**

**Fig 4:lambda=10**

**图表

描述已自动生成**

**Fig 5: lambda=100**

**EX3：**

1. init
2. crime=[y X];
3. n=size(crime,1);
4. %Split the crime data for 2-fold cross validation
5. cv=cvpartition(n,'KFold',2)
6. for idx = 1:2
7. % Split each fold into TrainData and TestData
8. TrainIndex = training(cv,idx);
9. TestIndex = test(cv,idx);
10. TrainData = crime(TrainIndex,:);
11. TestData = crime(TestIndex,:);
12. %Calculate the mean R-squared from test datasets
13. Y\_Train = TrainData(:,1);
14. X\_Train = TrainData(:,2:end);
15. Y\_Test = TestData(:,1);
16. X\_Test = TestData(:,2:end);
17. [n p] = size(X\_Train);
18. %In each fold, choose a regularization parameter from {1,100,10000}
19. for lambda = [1 100 10000]
20. b=(X\_Train'\*X\_Train+lambda\*eye(p))\(X\_Train'\*Y\_Train); %ridge
21. err=sum((Y\_Test-X\_Test\*b).^2);
22. disp(['error:', num2str(err)]);
23. endfor
24. endfor

**Fold1：**

Lambda=1：error=12.4718

Lambda=100：error=11.7062

Lambda=10000：error=13.9149

**Fold2：**

Lambda=1：error=26.5896

Lambda=100：error=32.3168

Lambda=10000：error=35.0036

**E4:**

r2 = 0.5582

r2 = 0.9420

r2 = 0.9962

r2 = 0.9990

r2 = 0.9971

1. x=[600 700 800 950 1100 1300 1500]';
2. y=[253 337 395 451 495 534 573]';
3. % linear model
4. X=[ones(7,1) x];
5. beta=X\y;
6. r2=1-sum((X\*beta-y).^2)/var(y)
7. % quadratic model
8. X=[ones(7,1) x x.^2];
9. beta=X\y;
10. r2=1-sum((X\*beta-y).^2)/var(y)
11. % cubic model
12. X=[ones(7,1) x x.^2 x.^3];
13. beta=X\y;
14. r2=1-sum((X\*beta-y).^2)/var(y)
15. X=[ones(7,1) x x.^2 x.^3 x.^4];
16. beta=X\y;
17. r2=1-sum((X\*beta-y).^2)/var(y)
18. X=[ones(7,1) x x.^2 x.^3 x.^4 x.^5];
19. beta=X\y;
20. r2=1-sum((X\*beta-y).^2)/var(y)