
Linear Regression with Normal Equations Algorithm (without Lasso) with Regularization

Table of Contents

| | |
|---|---|
| Clearing and closing the figures | 1 |
| Loading data set and visualizing it | 1 |
| Creating training and testing datasets for the learning algorithm | 1 |
| Applying normal equations | 2 |
| Training accuracy of our algorithm using Normal equations | 2 |
| Cross Validation Accuracy of our algorithm | 2 |
| Testing our learnt algorithm using Normal equations | 2 |

Clearing and closing the figures

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

Loading data set and visualizing it

```
fprintf('Loading dataset...\n\n');
load('comp.mat');
X = comp(:,1:7);
y = comp(:,8);

[X, mu, sigma] = normalize(X);
% adding intercept term
X = [ones(size(X,1),1) X];

Loading dataset...
```

Creating training and testing datasets for the learning algorithm

```
% Training data
Xtrain = X(1:160,:); % disp(size(Xtrain));
ytrain = y(1:160,:); % disp(size(ytrain));

% Cross validation data
Xval = X(161:190,:); % disp(size(xval));
yval = y(161:190,:); % disp(size(yval));
```

```
% Testing Data
Xtest = X(191:end,:); % disp(size(Xtest));
ytest = y(191:end,:); % disp(size(ytest));
```

Applying normal equations

```
fprintf('Linear Regression using Normal equations with
regularization...\n');
% Acquiring parameters using normal equations
lambda = 1;
[theta] = normalEqnReg(Xtrain, ytrain, lambda);
```

Linear Regression using Normal equations with regularization....

Training accuracy of our algorithm using Normal equations

```
% Applying learnt parameters on test data
pricetr = Xtrain * theta;

% Showing algorithm accuracy
errortr = abs(pricetr - ytrain) ./ ytrain; % error between actual and
predicted
accuracytr = 100 - (mean(errortr) * 100); % percentage accuracy
obtained
fprintf('Training accuracy on training set: %f\n', accuracytr);
```

Training accuracy on training set: 67.069495

Cross Validation Accuracy of our algorithm

```
%disp(size(Xtest));
pricecv = Xval * theta;

% Showing algorithm accuracy
errorcv = abs(pricecv - yval) ./ yval;
accuracycv = 100 - (mean(errorcv) * 100);
fprintf('Cross Validation accuracy on cross validation set: %f\n',
accuracycv);
```

Cross Validation accuracy on cross validation set: 59.232415

Testing our learnt algorithm using Normal equations

```
% Applying learnt parameters on test data
pricete = Xtest * theta;
```

```
% Showing algorithm accuracy
errorte = abs(pricete - ytest) ./ ytest; % error between actual and
predicted values
accuracyte = 100 - (mean(errorte) * 100); % percentage accuracy
obtained
fprintf('Testing accuracy on test set: %f\n', accuracyte);
```

Testing accuracy on test set: 67.612659

Published with MATLAB® R2016b