CMSC426-HW1

UID:116353601

- Eigenvector and Eigenvalues:Eigenvectors are special vectors that maintain their direction upon linear transformation by any matrix(A). Mathematically, A*v=lambda*v, where lambda is a scalar, A.K.A. Eigenvector.As, seen by the equation, on application of transformation(A), the vector simply scales and doesn't undergo any kind of rotation. So essentially, eigenvectors can be used to find axis of rotation of a given dataset.
  A few important characteristics of Eigenvector are:
  - They are linearly independent
  - these don't change their orientation after linear transformation
- Choice of outlier rejection:
  - For dataset 1(low noise): Since the noise level was low for this particular data set, all algorithms were pretty much successful, although, RANSAC outperformed both Linear regression and regularization. For such datasets using Linear regression and regularization would be better as they are computationally inexpensive.
  - For dataset 2(moderat noise):The noise level for this data set was a little bit higher and thus linear regression does not provide an accurate fit. For such cases regularization comes to rescue. By tuning the noise rejection parameter(lambda), we get a model which fits the dataset better than linear regression. RANSAC, also provides a good fit, given we choose the inlier threshold wisely.
  - For dataset 3(High noise): For a dataset with high noise, linear regression and regularization fail miserably. RANSAC, on the other hand, gives impressively accurate results on choosing the correct parameters such as inlier threshold, inlier ration, etc.

  To summarise, RANSAC, although computationally expensive, provides robust line fitting estimation, better than less computationally expensive algorithms like linear regression and regularization, provided that the assumption for deploying RANSAC are taken care of. One such assumptions is that the dataset is large enough to perform ransac. On very small datasets, RANSAC may not provide robust estimates.

- Limitation of each algorithm:
  - Linear regression:
    Although computationally inexpensive, this method works well when the noise is low and it craps out when there is high level of noise i.e highly **sensitive to noise**.
  - Regularization:
    This solution is less sensitive to noise and computationally inexpensive. Although this method is superior to Linear regression in handling noisy data, we have to

tweak the value of lambda manually, which **works for a narrow spectrum** of cases(close to which the lambda is set for). This won't work well in a data set that has different pattern or distribution of noise. Further, this method fails if the noise isn't gaussian.

- RANSAC: This is the most robust method of all the techniques mentioned and works amazingly in very noisy data. Limitation of RANSAC is that we need **large number of points** and that it is **computationally expensive**.

Implementation:

1. Plotting eigenvector:
   Got covariance matrix for the data point. Calculate eigenvector and eigenvalues using eig function. Plot the eigenvectors multiplies with eigenvalue(magnitude) on the scatter plot.
2. Line fitting:
   a. Linear regression: Loading the data sets and implementing the linear regression equation ($\hat{\beta} = (X^T X)^{-1} X^T y$). After getting beta, just plug in the line equation. $Y = X * \hat{\beta}$.
   b. Regularization: Adding a noise rejection element lambda to linear regression equation. ($\hat{\beta} = (X^T X + \text{lambda} * I)^{-1} X^T y$). After getting beta, just plug in the line equation. $Y = X * \hat{\beta}$.
   c. RANSAC: Selecting random points and fitting a line mode. Measure the number of inliers. Repeat the algo till we have the best number of inliers.