Projection Pursuit Regression and Other Methods to Reduce Dimensionality

Danielle Spitzig DBSPITZI@UWATERLOO.CA
20602629
Master's of Data Science and Artificial Intelligence
University of Waterloo

Abstract

This paper describes the use of projection pursuit regression (PPR), a dimensionality reduction method, and other such methods and their uses in regression. These methods are compared on a dataset of Coulomb values of different organic molecules to predict the molecule's atomization energy. All methods used were able to find useful models in terms of prediction, but the PPR model performed the best out of all other methods based on the residual plots, the mean squared errors, and the mean absolute errors derived from the test data.

Keywords: PPR, PCR, Subset Selection, Dimensionality Reduction

1. Introduction

It is becoming increasingly common for technology to produce large amounts of data. As this multivariate data becomes more common, the application of multivariate data analysis methods becomes increasingly relevant to extract useful information. In regression, one such technique to extract this useful information is through reducing the dimensions of the dataset. There are two ways this reduction can happen. Either with variable selection techniques, such as forward-stepwise selection and Lasso. Or with dimensionality reduction techniques, such as principal components regression (PCR) and projection pursuit regression (PPR).

These techniques were introduced as a way to overcome the curse of dimensionality when dealings with data in high-dimensional spaces [1]. This refers to the fact that the number of samples needed to estimate a function grows exponentially with the number of explanatory variables in a dataset. Hence, the data needed to create a reasonable estimate can quickly become too large to be feasible. Some related problems that occur with multivariate data is the fact that high-dimensional spaces are inherently sparse and there is likely to be correlations within the data [1]. This further demonstrates the need for reduction techniques to deal with these issues of high-dimensional data.

Dimensionality reduction is important in all aspects, but it has been important analytical technique in scientific fields, such as chemistry [2][3]. This is due to the fact that for many practical problems in the sciences the data is high-dimensional. The methods used in this work will be applied to a chemical, high-dimensional dataset to truly show the capabilities of such methods and to compare them.

2. Data and Preprocessing

The dataset used for this work was found on Kaggle [4] and it contains the atomization energies of molecules at their ground state. It is attributed to a paper by Dr. Himmetoglu [5], which explains that the dataset is composed of organic chemistry molecules consisting solely of carbon (C), hydrogen (H), nitrogen (N), oxygen (O), phosphorus (P), and sulphur (S) where each molecule has a range of 2 to 50 atoms. There were more criteria that each sample had to adhere to, and from a database of 75,000 molecules, 16,242 met these requirements and were used in the dataset.

There are 1278 variables in this dataset. The last column corresponds to the response to predict, the atomization energy of the molecule. Each atomization energy was calculated by simulations using the Quantum Espresso package [5]. This leaves 1277 variables, 2 of which correspond to indices and can be dropped. Hence, there are 1275 variables to be used as predictors. These predictors come from a design matrix, constructed from the eigenspectrum Coulomb matrix. The eigenspectrum refers to the upper-triangular, unrolled Coulomb matrix and the sorted eigenvalues of the matrix, where $\mathbf{Cv} = \lambda\mathbf{v}$, for each Coulomb matrix C, such that $\lambda_i \leq \lambda_{i+1}$ [6].

The Coulomb matrix for each molecule is a $n \times n$ matrix, where n is the number of atoms in the molecule. However, in order to secure consistent dimensions, the molecules that have fewer than 50 atoms are padded with zeros so that all molecules have a 50×50 matrix representation. It is important to note, and will be discussed later in this work, that these zero terms are a stand-in for no data at all, hence they should be treated as null values. So the unrolled upper-triangular part of these matrices give 1225 predictor variables, and in addition to that there are 50 eigenvalues for each molecule. This gives the 1275 predictors mentioned above [5].

The formula to calculate a Coulomb matrix is given below, where Z_i represents the nuclear charge and R_i represents the Cartesian coordinates of atom i.

$$C_{i,i} = 0.5 \cdot Z_i^{2.4} \tag{1}$$

$$C_{i,j} = \frac{Z_i Z_j}{|R_i - R_j|} \tag{2}$$

A Coulomb matrix contains one row per atom, it is symmetric, and it requires no explicit bond information. Therefore, except for molecules which have the same atoms and inter-atomic distances, the Coulomb matrix is a unique representation of a molecule [6]. In this work, the Coulomb representations of the molecule will be used to predict the calculated atomization energies.

Recall from above that the data is padded with zeros as a placeholder for null values. We did an initial search of which variables had more null values than actual values, and from this search 950 variables were found. These variables were discarded as they were found to not include enough information to warrant their addition into the dataset, and they introduced a lot of multicollinearity issues. This left us with 16,242 samples and 325 variables. This is still a sufficiently large dataset to run the reduction methods that we want to compare.

The next step was to look at the samples and see how many of them had null values as well. About 8,000 molecules still contained null values. There are a few ways to deal with null values in samples. The most common methods are an imputation strategy or removal. Due to the nature of the dataset, where non-diagonal values of the matrix would be undefined at these variables, we decided to discard these rows.

The final product after dealing with these null values was a dataset of 8,209 samples and 325 variables. After removing the null rows and columns, the dataset was standardized. This standardization was done such that each variable has a mean of 0 and a variance of 1.

The response variable was only mean-centered, and the mean-centered subset of samples used in this report is shown in the histogram in Figure 1.

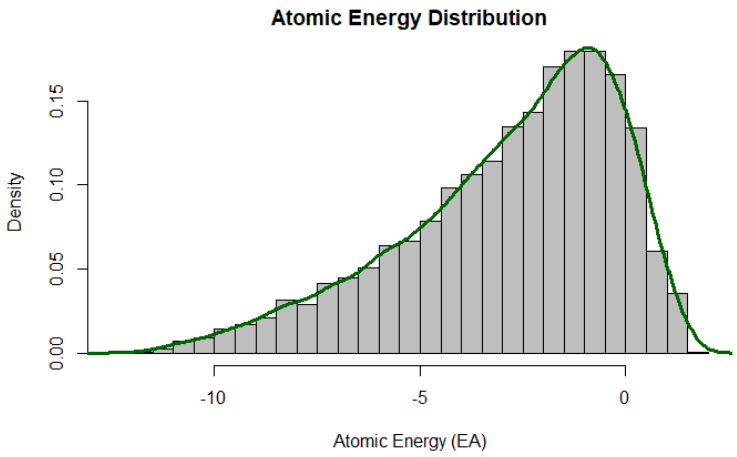


Figure 1: Distribution of the mean-centered response variable

Even after removing the null values, this is still a large multivariate dataset. There are 325 features; hence, the whole dataset cannot be visualized and instead, 5 random variables and the response variable were plotted as scatter plots. These plots are shown in Figure 2.

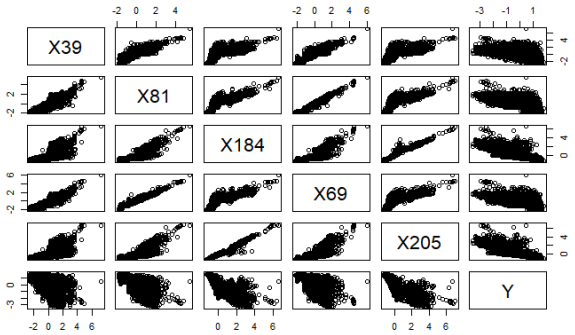


Figure 2: Plots of 5 randomly chosen scaled columns

These plots show that there appears to be multicollinearity issues. This can be seen in the correlation that some of the variables have with each other. This issue, as well as the possible issue of outliers will be further discussed in the results section of this work.

3. Dimensionality Reduction

Dimensionality reduction methods are helpful as there is often redundant information in the dataset. This is either due to multicollinearity or the variable has variance smaller than the noise present in the data [1]. Due to the redundant information there should be a way to strip the unnecessary information from the dataset producing a more compact dataset with the relevant information.

In this work forward-stepwise selection, Lasso regularization, principal components regression (PCR), and projection pursuit regression (PPR) were all utilised, with a focus on PPR.

3.1 Variable Selection

The two subset selection techniques used were forward-stepwise selection and Lasso regularization.

3.1.1 Forward Stepwise

Forward-stepwise selection works iteratively. It starts with a null model, and adds a single variable at each time step until all p variables have been chosen. Which variable is added to the model depends on either the minimizing the residual sum of squares or maximizing the R^2 value. From these p models the best model can be selected based on some criteria, such as highest adjusted R^2 value, or lowest error. The constraint with this model is that it assumes that the best model with q variables contains the best $q-1$ variables within it. Stepwise methods are considered greedy, as they produce nested results and because of this they may have higher bias than other variable selection methods [8].

3.1.2 Lasso

Lasso regularization works with a shrinkage estimator. This estimator penalizes the model whenever a variable is added. Lasso, unlike the Ridge regularization, doesn't have a closed-form solution but it does perform variable selection [8]. Lasso works by taking the minimum of a least-squares problem with an L1 penalizing factor, as shown below.

$$arg\min_{\beta}\left[\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}x_{ij}\beta_j\right)^2+\lambda\sum_{j=1}^{p}|\beta_j|\right] \tag{3}$$

Hence, with Lasso there is a trade-off between optimizing the least-squares and adding in more variables.
Variable selection approaches can be at a disadvantage with a noisy dependent variable and high-dimensional multicollinear independent variables [9]. The selected variables in these

approaches may not be the best variables to chose. This is especially prevalent in stepwise methods as they are iterative, so they include more bias [9].

3.2 PCR

PCR uses variance to take projections, where the principal components (PCs) that correspond to the highest explained variance are kept. PCR assumes that the data has been standardized to have a mean of 0 and a standard deviation of 1 [10]. Recall for ordinary least squares, the coefficients are estimated using

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

To perform PCR, we can perform a transformation such that

$$(X^T X)^{-1} = PDP^T = Z^T Z$$

where $\mathbf{D}$ is a non-negative diagonal matrix of the eigenvalues of $X^T X$, $\mathbf{P}$ is the eigenvector matrix of $(X^T X)^{-1}$, and $\mathbf{Z}$ is the matrix of principal components [10].

This method works as variables that have high multicollinearity will subsequently have small eigenvalues, so those variables can be omitted from $\mathbf{Z}$ to give a new variable denoted as $\mathbf{Z}^*$[10]. The variables left in $\mathbf{Z}^*$ are then used in place of $\mathbf{X}$ in the estimation, where:

$$\hat{A} = (Z^{*T} Z^*)^{-1} Z^{*T} Y = D^{-1} Z^{*T} Y$$

The number of PCs that should be included in $\mathbf{Z}^*$ can be determined with cross-validation [10].

3.3 PPR

The PPR used in this work was inspired by the original paper by Friedman and Stuetzle [11]. Projection pursuit (PP) is used to seek "interesting" projections of high-dimensional data into a lower dimension (often no greater than 3) [3].

It is common to interpret high-dimensional data through well-chosen projections [11], and this "interesting" projection that PP looks for is guided by optimizing an appropriate objective function. PPR uses the fraction of unexplained variance by the smooth Y versus αX, where α is the projection vector, as the aforementioned objective function to optimize.

The two basic elements of PP are the PP index and the algorithm. A PP index, $I(\alpha|X)$ is a measure of how interesting the projection by α is, where it implicitly depends on the data X. The larger the PP index is, the more interesting the projection is [3]. The algorithm optimizes the PP index to find the maximum index over all α, where the first several maxima found by the algorithm provide the most interesting projections [11].

In PPR, the regression surface is approximated through a finite sum of ridge functions such that $f^{(m)}(x) = \sum_{i=1}^{m} S_i(\alpha_I^T x)$, where S are the univariate smoothing functions, α are $p \times k$ semi-orthonormal matrices, and m is the number of ridge functions.

This approximation is constructed iteratively, where, for the next term in the model, a smoothing representation is constructed on the current residuals to get $S(\alpha X)$. As mentioned above, PPR looks to maximize variance, so the PP index that is maximized for this

linear combination is the fraction of unexplained variance [11]. This algorithm continues until the PP index of the next term is smaller than a given threshold.

In this work, the smoothing method used is called super smoother [3], this is due to the fact that the model seeks to explain response variability by a sum of smooths of the data [11]. Hence, high local variability encountered in particular steps may cause some smoothing methods to have dependence on other linear combinations. This is an unwanted side-effect, so to preserve the ability to fit the structure in future iterations the smoothing functions have to avoid accounting for fits along existing directions [11]. As a consequence of this, a variable bandwidth smoother is used in the original paper [11], and in future implementations this is changed to Friedman's super smoother [12], which is an adaptive bandwidth smoother.

PPR is a useful method as it projects the variables in the optimal direction before applying smoothing functions to the variables. This is useful as it constrains the estimate to low dimension, so it is solvable with methods such as ordinary least squares or spline methods, thus avoiding the curse of dimensionality [11]. Note that because PPR attempts to fit projections of the data, it can be difficult to interpret the fitted model as a whole. This can make the model more useful for prediction than for understanding the data.

4. Results and Discussion

The first step was to split the preprocessed dataset into a training and test set, the chosen split was 75% training and 25% test.

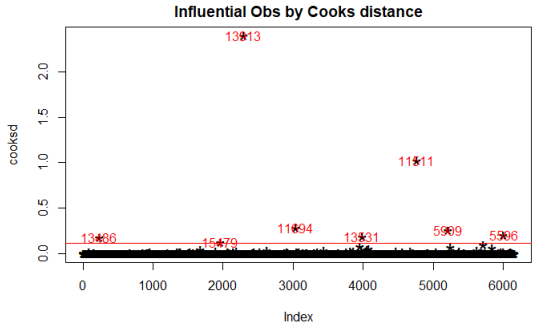
The mean squared error (MSE) and mean absolute error (MAE) was calculated on the predicted test set on all of the aforementioned methods. The linear and Huber models mentioned were run on the subset variables chosen from forward stepwise selection to observe a robust method as well. The results from these methods are given in Table 1.

Test Error Values		
Method	Mean Squared Error (MSE)	Mean Absolute Error (MAE)
Linear	0.50669	0.47976
Huber	0.5308034	0.46219
Lasso	0.41836	0.42121
PCR - 8 comp	0.76483	0.60457
PCR - 13 comp	0.58827	0.50186
PPR - 1 comp	**0.22834**	**0.30308**

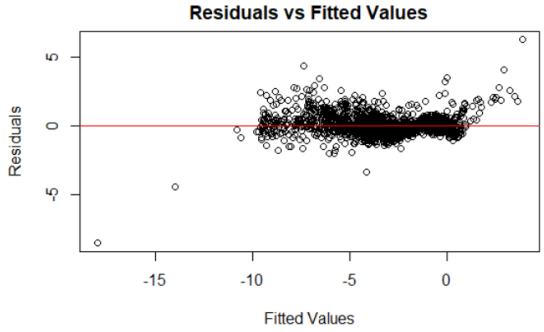
Table 1: Table of test errors associated with the regression methods used

It's interesting to note that the linear model and the robust model using Huber performed similarly. This is a possible indication of no or very few outliers in the data. One of the ways to check for outliers in multivariate data is looking at Cook's distance [13]. Cook's distance is applied to a linear model to show the influence of each sample in the

fitted response values. This is useful when there are many explanatory variables. A plot of potential outliers using Cook's distance is shown in Figure 3a.



(a) Plot of Cook's distance on a linear model.



(b) Plot of residuals against fitted values of linear model.

Figure 3: Figure 3a describes the influence of samples. Figure 4b checks the linearity assumptions of the data

All of the points with red by them are considered to have high influence. There appears to only be a few outliers, but nevertheless, some of the, seem to have a lot of influence. To further look at the outliers and other assumptions about the data, a plot of the fitted values versus the residuals is shown in Figure 4b.

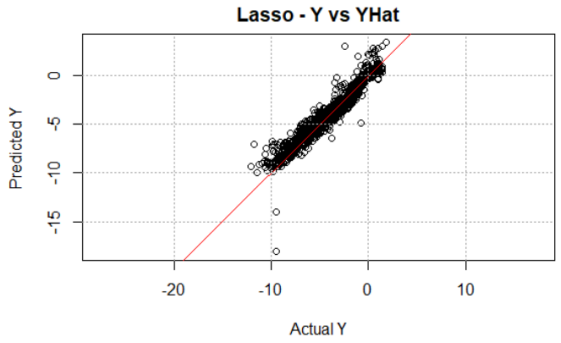
A residuals plot can show whether some key assumptions hold for linearity. In this case, the data doesn't seem to follow some of these assumptions. This plot clearly has some outliers, hence either the relationship between the variables isn't linear, or there isn't equal variance along the regression. This will be further observed after the transformation from PPR.

Let's compare the top two methods, PPR and Lasso, to look at their similarities and differences.

First, let's look at Lasso as a variable selection technique. Lasso was optimized using cross-validation on the training set, and an optimal λ was found to be approximately 0.01226. This seems like a small λ value, but of the 325 variables, Lasso dropped 269 and

another 32 variables were less than 10^{-6}. Hence, even with this small λ value Lasso was able to select a reasonable amount of variables at approximately 25.

The plot of the true response against the fitted response is shown below in Figure 4a.



(a) True response plotted against fitted response for Lasso model.

(b) Distribution of the absolute residuals for the Lasso model.

Figure 4: Figure 4a describes the Y vs $\hat{Y}$ plot for Lasso. Figure **??** describes the absolute residuals as a histogram for Lasso

It appears that the response seems to generally follow the true values; however, there does seem to be some extreme points in the model. The absolute value of the residuals was modeled in Figure **??**, to observe how close the predicted values are to the true values.

The majority of the residuals fall within the first bin, which shows that the model predicts the majority of values well, just like was shown in Figure 4a. However, there are some residuals that that are further away, showing the possibly presence of outliers in the data again.

These plots can be compared against the results from PPR. Recall that PPR projects the variables into a lower dimensional space, where it is assumed that the projections are linear. This should be shown in the residual plot of PPR in Figure 5.

There is a clear difference between the residual plots in Figure 4b and Figure 5. While this plot also has some potential outliers, the outliers don't seem as extreme like they were

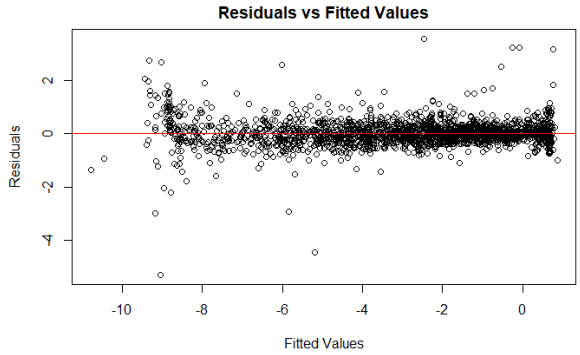


Figure 5: Plot of the residuals against the fitted values on the PPR model.

in the previous residual plot. This is mostly likely due to the transformation that PPR performs, and that this transformed data follows the linearity assumption more closely.

As above with Lasso, the plot of the true response against the fitted response is shown in Figure 6a, and the histogram of the absolute value of the residuals is modelled in Figure 6b.

There is also fewer outliers present in the PPR fitted values than there were in the Lasso fitted values. The data seems to more closely follow the true values, which just further shows that PPR model is more well-suited for this data.

The histogram of the residuals looks similar; however, the residuals are more condensed and the range of values for the residuals is smaller in PPR than it is for Lasso.

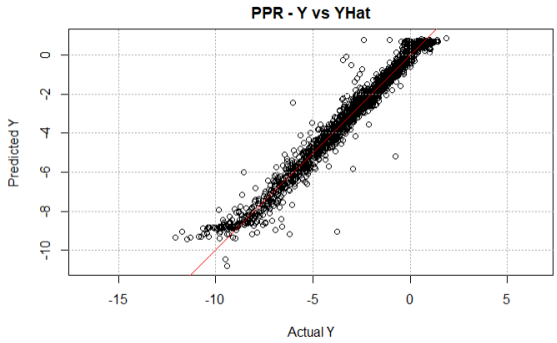
Hence, from the results above, such as the lowest MSE and MAE, and the best residual plots, it appears that PPR was a better method for dimensionality reduction with this dataset.

5. Conclusion

Dimensionality reduction techniques are an important tool to use with multivaraite data. Subset selection techniques are useful, but can run into issues when faced with multi-collinearity and noise in the response variable. Dimensionality reduction techniques are a cleaner way to deal with the issue, as they can often circumvent the curse of dimensionality through the use of projections.

6. Future Works

In the future, looking at different smoothing methods used in the PPR algorithm could be helpful. In this work only the default super smoother was utilised, but any smoothing technique could have been used. Also, it was possible to not remove as much data as was done in this work, and instead the data could have been separate into two datasets depending on the size of the molecule. This data augmentation could have resulted in a more robust model. Also, PPR is often considered a precursor to regression using neural network models; hence, in future work, we may observe the use of neural networks for multidimensional data.



(a) True response plotted against fitted response for PPR model with 1 component.

(b) Distribution of the absolute residuals for PPR model with 1 component.

Figure 6: Figure 6a Y vs $\hat{Y}$ plot for the chosen PPR model. Figure 6b describes the absolute residuals for PPR as a histogram.

Appendix

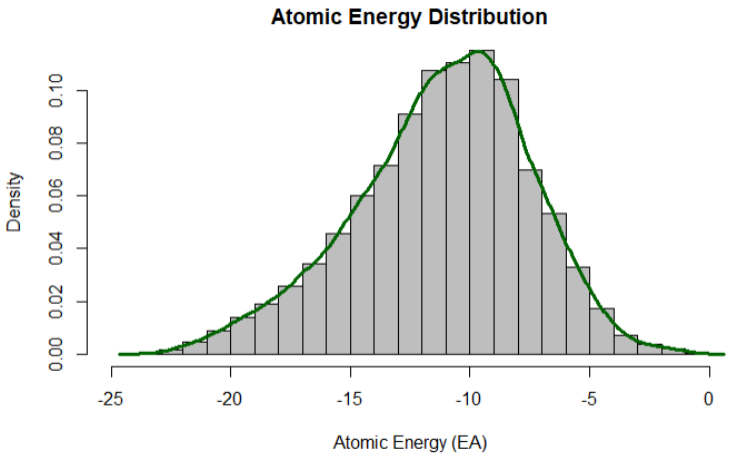


Figure 7: Distribution of atomization energy given in Ry for the original data before pre-processing

$\mathbf{C} =$

	H	H	C	C	H	H
H	0.5	0.3	2.9	1.5	0.2	0.2
H	0.3	0.5	2.9	1.5	0.2	0.2
C	2.9	2.9	36.9	14.3	1.5	1.5
C	1.5	1.5	14.3	36.9	2.9	2.9
H	0.2	0.2	1.5	2.9	0.5	0.3
H	0.2	0.2	1.5	2.9	0.3	0.5

Figure 8: The Coulomb matrix of ethene, C_2H_4

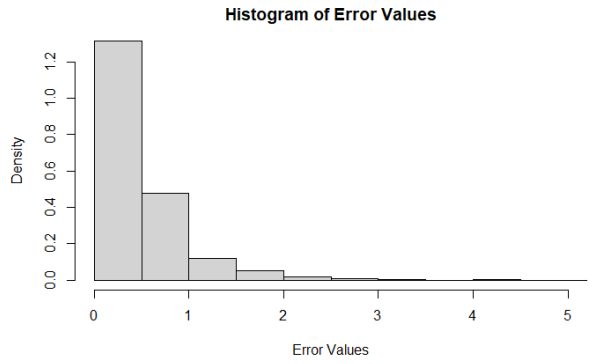


Figure 9: Distribution of the residuals from the linear model

Figure 10: Distribution of the residuals from the Huber model

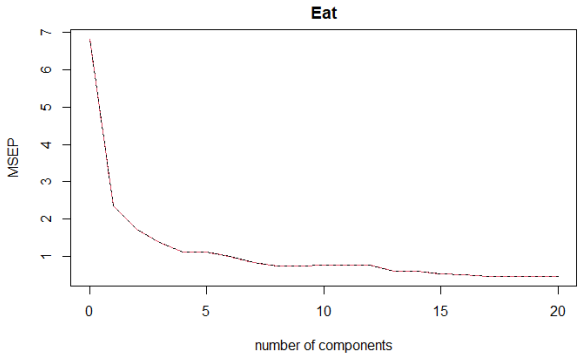


Figure 11: Validation plot to find optimal number of components for PCR

Figure 12: Distribution of the residuals from the PCR model with 8 components

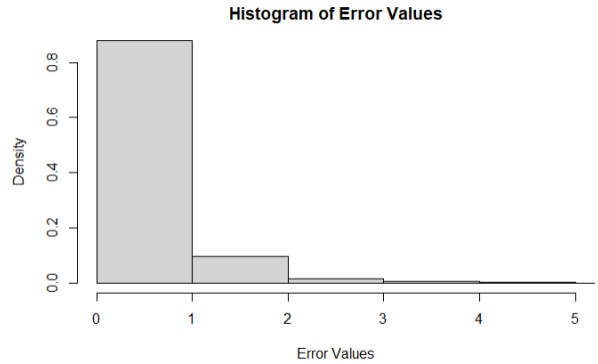


Figure 13: Distribution of the residuals from the PCR model with 13 components

References

[1] Carreira-Perpinan, M. A. A Review of Dimensionality Reduction Techniques, University of Sheffield, (1997).

[2] Ren, Y.; Liu, H.; Yao, X.; Liu, M. Prediction of ozone tropospheric degradation rate constants by projection pursuit regression, Analytica Chimica Acta, (2007).

[3] Liu, H.; Yao, X.; Lui, M.; Hu, M.; Fan, B. Prediction of gas-phase reduced ion mobility constants (K0) based on the multiple linear regression and projection pursuit regression, Talanta, (2007).

[4] Kaggle Dataset, URL: https://www.kaggle.com/burakhmmtgl/energy-molecule

[5] Himmetoglu, B. Tree-based machine learning framework for predicting ground state energies of molecules, Journal of Chemical Physics, (2016).

[6] Hansen, K.; Montavon, G.; Biegler, F.; Fazli, S.; Rupp, M.; Scheffler, M.; Anatole von Lilienfeld, O.; Tkatchenko, A.; Müller, K-R. Assessment and validation of machine learning methods for predicting molecular atomization energies, Journal of Chemical Theory and Computation, (2013).

[7] Schrier, J. Can One Hear the Shape of a Molecule (from its Coulomb Matrix Eigenvalues)?, Journal of Chemical Information and Modeling, (2020).

[8] James, G.; Witten, D.; Hastie, T.; Tibshirani, R. An Introduction to Statistical Learning with Applications in R

[9] Efron, B.; Hastie, T.; Johnstone, I.; Tibshirani, R. Least angle regression, Annals of Statistics, (2004).

[10] Chapter 340: Principal Components Regression, NCSS Statistical Software, (2020).

[11] Friedman, J.H.; Stuetzle, W. Projection pursuit regression, Journal of the American Statistical Association, (1981).

[12] Friedman, J. H. A variable span scatterplot smoother, Laboratory for Computational Statistics, Technical Report No. 5., (1984).

[13] Gao, Q.; Ahn, M.; Zhu, H. Cook's Distance Measures for Varying Coefficient Models with Functional Responses, Technometrics, (2015).