Angarita, Y.

Quantum Chemistry Project
Data Analysis Report Part I

Topic: Applications of Data Analysis and Machine Learning (ML) using a published algorithm and logic in the field of Quantum Chemistry

A. Brief description of contribution

The purpose is to analyze molecular data from N = 16,242 training observations to understand the fundamentals for the regression neural network algorithm.

The goal is to fit a model that can predict ground state energies of molecules made up of six atoms: C, H, N, O, P, and S, based on p = 1,275 feature vectors.

To be specific, a given molecule in the dataset, for a total of 16,242 molecules, numbered with an index is represented by a *p*-dimensional feature vector x_i where *p* is the total number of unique entries in the Coulomb matrix (for instance, the upper triangular part of the symmetric 50x50 matrix C_{ij}, unrolled in into a 1,275-dimensional vector) or the number of eigenvalues (for example, a 50-dimensional vector of eigen values) (Himmentogu, B., 2016). The unfolded vectors from the Coulomb matrices were used in this analysis.

In terms of machine learning algorithms, a penalized (regularized) least squares fit of a linear model using ridge regression, with the model parameters obtained by batch gradient descent was performed. The tuning parameters were chosen using five-fold cross validation, and the best-fit model parameters was inferred on the training dataset conditional on an optimal tuning parameter, which is similar to the approach described by the author on the research article.

B. Data

Data for these observations are given in the attached roboBohr.csv file, with atoms of each molecule labeled on each row (rows 2 through 16,243), and input features and response given on the columns (with the first row representing a header for each column). There are six quantitative features, given by columns labeled "0", '1", "2", "3", "4" ..."1274"

C. Effect of tunning parameter on inferred regression coefficients.

A discrete grid of seven tuning parameter values $\lambda \in \{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}\}$ was considered where the tuning parameter is evaluated across a wide range of values on a log scale. For each tuning parameter value, gradient descent was used to infer the best-fit model.

D. Deliverables

Deliverable 1: Illustration of the effect of the tuning parameter on the inferred ridge regression coefficients by generating a plot using Python of 1,275 lines (one for each of the p = 1,275 features), with the y-axis as B_j,j = 1,2, ... ,1275, and the x-axis the corresponding log-scaled tuning parameter value log10 (λ) that generated the particular B_j. Both axes with labels. Without the log scaling of the tuning parameter, the plot will look distorted.

Deliverable 2: Illustration of the effect of the tuning parameter on the cross-validation error by generating a plot using Python with the y-axis as CV error, and the x- axis the corresponding log-scaled tuning parameter value $\log_{10}(\lambda)$ that generated the particular CV(5) error. Both axes with labels. Without the log scaling of the tuning parameter, the plot will look distorted.

Deliverable 3: Indication of the value of λ that generated the smallest $CV_{(5)}$ error.

Deliverable 4: Given the optimal λ, the model was retrain on the entire dataset of N = 16,242 observations and the estimates of the p = 1,275 best-fit model parameters were provided.

E. Findings

The model described above seems to work for this data because the plot generated on deliverable 2 shows the expected behavior of a curvature where, as the tuning parameter lambda increases, the CV error decreases

The tuning parameter lambda that produces the lowest CV error is 1000 for this dataset

The model score using the entire dataset suggests the trained model can make accurate predictions approximately 95.6% of the times.

F. Conclusions

The second value for 'ridge_regressor.best_score_' is the main cross-validated score of the best_estimator. The mean can be positive or negative. This served as an additional metric to describe the best predictor. Maybe displaying the mean square error may be more informative.

It seems a good approach to compare these results with those to be obtained without using statistical or ML libraries and provide a discussion as to why the results are different if applicable.

For future and, to optimize the model, it seems a good approach and/or suggestion to perform a penalized (regularized) least squares fit of a linear model using elastic net, with the model parameters obtained by coordinate descent.

G. References

B. Himmetoglu, Tree based machine learning framework for predicting ground state energies of molecules, J. Chem. Phys. 145, 134101 (2016).

Author Note:

The structure and logic of the algorithm used for this project was designed by Michael DeGiorgio, Ph.D., Associate Professor, Department of Computer and Electrical Engineering and Computer Science, and instructor of the CAP 5625 Computational Foundations of AI course.

Enrollment in the course mentioned above was parallel to that of this course. The main purpose was to improve in computational, logic design implementation, and data analysis to be applied on this and future projects. The mathematical details of the algorithm can be shared upon request.