A machine learning approach to determine chemical shifts in NMR spectroscopy data

Maxime EPARS, Gianni GIUSTO, Yann MENTHA,
Machine Learning course (CS-433), EPF Lausanne, Switzerland
December 19, 2019

***Abstract*—As part of the CS-433 Machine Learning course at EPFL, this project aimed to apply a machine learning pipelines to solid-state nuclear magnetic resonance (NMR) spectroscopy data provided by Prof. Michele Ceriotti and Félix Musil from the Laboratory of Computational Science and Modeling (COSMO) at EPFL to determine the chemical shifts of atoms. For this regression task, models were not restricted to linear ones but neural networks were also implemented. The final best cross-validated model allowed to predict the chemical shielding from a complex structure with an MSE of 0.53 and a $R2$ coefficient of 0.96 (4-fold CV).**

I. Introduction

Nuclear magnetic resonance (NMR) spectroscopy is a method used to determine the chemical structure of a molecule based on the principle of the resonance frequency: any nuclear spin in a given structure depends on its chemical local environment (*i.e.* shielding). The change in resonance frequency compared to a standard reference is represented by the chemical shift. Solid-state NMR represents a specific type of this powerful method tailored to amorphous solids where, by definition, magnetic fields are orientation-dependent. Accurate methods using plane wave density functional theory (DFT) have been developed to calculate chemical shifts. However, these techniques are computationally costly and hence fail to compute the latter for large or complex molecular structures. Machine learning tools may potentially offer a new way to compute the chemical shifts with a reasonable computational cost while maintaining high accuracy.

The aim of this project is then to build a powerful model to predict chemical shieldings of hydrogen atoms from a few hundreds of crystal structures from the Cambridge Structural Database (CSD) [1] based on their local environment. Note that the chemical shift can easily be computed from the shielding value and its reference. The challenge consists therefore in determining this shield, which is precisely the target of the considered training task.

II. Models and Methods

A. Exploratory data analysis

The labels used to train the model are computed by DFT-based methods and are assumed to accurately estimate the experimental chemical shieldings. These features exploited in the model are of structural nature. These describe the local chemical environment of each atom as a spherical neighborhood density, represented by a superposition of Gaussians centered on each of the atom positions contained in this three-dimensional space [2]. This is defined as the smooth overlap of atomic positions (SOAP) kernel. Spherical neighbourhoods considered in this project have cut-off radii of 3, 5 and 7Å. In brief, the dataset is composed of 3 local environments (corresponding to the different cut-off radii), each one containing 38,514 datapoints and 14,400 features. The labels and features are all continuous variables. Visual inspection of the feature matrix (not shown here) indicates there is no significant proportion or any pattern of missing or null data, suggesting a clean dataset. The data was split into training and testing sets following a ratio of 90%/10%. Throughout the project, k-fold cross-validation based on MSE was used as the main method to assess the model-performance, as $R2$ score tends to decrease with dataset size.

B. Feature processing and engineering

For the current regression task, raw data was first cleaned and transformed to make the most of it prior building the model. To do so, the samples associated with targets classified as outliers by the interquartile range (IQR) method were removed from the training set. Data was then scaled through Min-Max in order to subtract the influence of the range difference between features. In addition, principle component analysis (PCA) was employed for some models to reduce the dimensionality of the feature space and mitigate the correlation between features variance. A number of principal components (PCs) representing 98% of the total explained variance was kept.

In parallel, we perform a feature reduction using recursive feature elimination (RFE). The method build a first model using all the available feature and progressively prune the least important ones based on weights of a ridge estimator whose regularizer parameter was previously optimized by k-fold CV. With the remaining features, we train both linear models and neural networks.

C. Linear models

As we deal with a supervised learning problem and the target is continuous, linear regression was believed to be an appropriate class of methods to predict labels. To prevent overfitting, only regularized models were implemented, namely *Ridge*, *Lasso* and *Elastic net*. Their respective regularization parameters were all optimized through grid search. Linear models were only employed on single dataset, that is with only one cutoff radius neighbourhood and not a combination of them. This provides a reference model performance to compare with neural networks.

D. Neural networks

With the data available, it is believed that neural networks may be a powerful tool to determine the actual shielding of the structure. Therefore, several multi-layer perceptron (MLP) were implemented, each outputting a prediction which is compared to the actual true value. We build shallow and deep networks to emphasize the effect of depth on the prediction. The mean-square error (MSE) is used as loss and the network is trained using the classical stochastic-gradient descent (SGD) algorithm. Either the raw data or the pre-processed (as mentioned in section II-B) data is fed to the network. We also incorporate *dropout* and *batch normalization* to reduce overfitting during the training and test several activation functions. The networks were implemented using both `PyTorch` and `Keras` frameworks and all models were trained on 150 epochs.

E. Performance estimation

To assess performance, the dataset is divided into a training and a validation set: The model construction is based on the training set, while the validation set is only used as a final estimator of the model performance. In order to compare methods, we rely mainly on 3 metrics, namely the MSE, the MAE and the coefficient of determination $R2$. As outliers are removed from the training set but not from the testing set used for assessing the model performance, MAE should be a better estimator of the model performance than MSE, since the former is a more robust metric (with respect to outiers). $R2$ provides the advantage that it is a scale-free metric, upper-bounded by 1, and may provide more information than an absolute-value metric like MAE. To obtain final unbiased performance estimation, we perform a 4-fold cross-validation for all models. Also, bias-variance decomposition is performed to evaluate the suitability of a given model to the dataset.

F. Ensemble methods

To take advantage of the 3 datasets at our disposal, we hypothesise that combining these 3 sources of information may help to get better fit. To do so, we trained 3 times the same neural network architecture individually on the train set, and predicted the label accordingly. Then, we wanted the method to perform as follows: on one hand, if the 3 predictions for the target are equally spaced, we want to take the mean as an estimator (uncertainty). If on the other hand 2 of the 3 predictions end up close to each other but the third one looks inaccurate, we want the method to give a final result as close as possible to these 2 values. We therefore developed the following formula, which acts similarly to a harmonic mean, but does not privilege smaller values as the harmonic mean does:

$$\bar{y} = \frac{\sum_{j \in S} \sum_{j \in S, i \neq j} \frac{1}{a_{ij}} y_i}{\sum_{j \in S} \sum_{i \in S, i \neq j} \frac{1}{a_{ij}}}, a_{ij} = |y_i - y_j| \qquad (1)$$

Here we have $S = 3$, one dimension for each set of values (3, 5 and 7Å). The arithmetic mean was also implemented for comparison purposes.

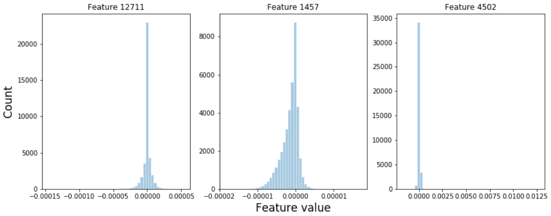

Figure 1. **Random feature distributions over a subset of the data.** Value counts of the 3 randomly selected features before data pre-processing.

III. Results

A. Data pre-processing

As shown on Figure 1, range and shape of distributions are highly feature-dependent, which strongly motivates normalization before any model implementation. Since normality is obviously not present in (all) the features, it confirms the relevance of Min-Max scaling to preprocess the data. The removal of outliers on the training set using the IQR method prior to Min-Max scaling has a little impact on the MAE (Fig. 2) but still, the variance is lower when using this method.

B. Linear models

The best linear model allows to get an MSE of 0.60 obtained through 4-fold cross-validation (Table I). It consists in a simple *Ridge* regression with parameter $\alpha = 0.003$ optimised through grid-search, with multiple other method such as PCA, KernelPCA, robust scaler and others. The 2 other methods ElasticNet and Lasso Regression were optimized as well, following the same procedure, but gave poorer results compared to *Ridge*. In addition, we display the bias-variance decomposition of the Ridge model trained on the data from the 3Å spherical neighbourhood on Figure 3 to emphasize the error evolution as a function of the sample size. The bias of the model does not decrease further when the subset size exceeds approximately 2,500 samples. Above

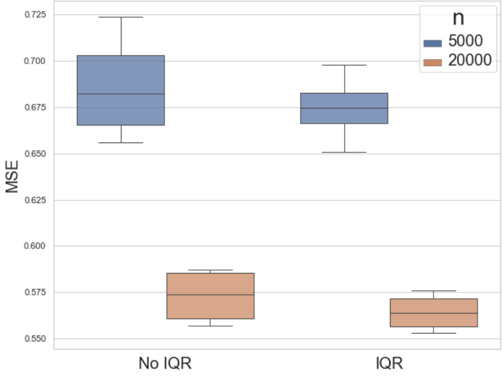

Figure 2. **Effect of outliers removal for two different sample sizes**. Results obtained using the best performing neural network. n samples were randomly selected from the original raw data and the process was repeated 4 times for statistical purposes.

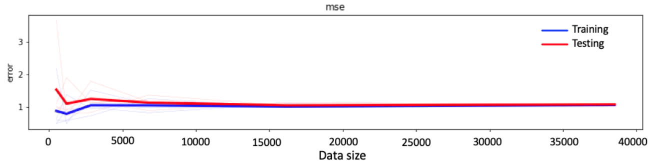

Figure 3. **Bias-variance decomposition.** Ridge model (with optimized α) trained on incremental data subset sizes with 5 different seeds for each subset. Features were standardized by Min-Max scaling, selected by their respective F1-score and further by PCA. MSE values were obtained through 4-fold cross-validation.

this value, the MSE error is still fairly important and is equivalent for both the training and testing sets. As expected, the variance is high for small data subsets, and decreases as more samples get included.

C. Neural Networks

Interesting results are achieved using a rather simple model consisting of 1 hidden layer and using leaky ReLU as activation function. Indeed, the latter obtained a MSE of 0.66 without the use of recurrent feature selection. PCA preceded by normalization proved less effective, hence either all features were fed to the network input layer, or RFE was carried out beforehand (leaving approximately 2,000 features left). With RFE and with a 4-fold cross-validation on the whole dataset gathering neighbors within a range of 5Å, we reached a mean MSE of 0.63 and 0.53 for a shallow (1 hidden layer of 100 neurons) and deep (3 hidden layers with layers of 100, 50 and 50) architecture respectively (Table I). Unfortunately, the ensemble method did not outperform the individual one, but still gave similar results on a 5-fold CV. Thus, a simpler model consisting of a single neural network was preferred, for sake of complexity. Note that surprisingly, the use of *dropout* and *batch normalization* did not lead to any model improvement.

Table I
PERFORMANCE COMPARISON BETWEEN DIFFERENT MODELS ($n = 38,514$).

	MSE	MAE	r^2
Linear model	0.60 ± 0.10	$0.51 \pm 2.1e-3$	$0.95 \pm 6.9e-3$
Shallow NN	0.63 ± 0.09	$0.52 \pm 2.0e-3$	$0.95 \pm 6.1e-3$
Deep NN	0.53 ± 0.11	0.49 ± 0.02	0.96 ± 0.01

IV. DISCUSSION

From the bias-variance decomposition, linear models show limited capacity for such a complex dataset. Indeed, there are prone to underfitting, as more data above a certain threshold (here around 2,500 samples) does not lead to any model performance improvement. This confirms the need for some method with more representational power: multi-layer perceptron in our case. Hence, non-linear models such as neural networks turned out to perform better despite requiring more training and thus computational power.

Ensemble method did not perform as well as expected: this might be due to the fact that the relevance of the information brought by each dataset was relatively unbalanced across different distance ranges. Only trial and error methods allowed to dismiss this method, which could potentially perform decently in other contexts still.

In future work, including long-range features would provide interesting insight and may increase the neural network model performance. Furthermore, the local structure of the atoms and their mutual interactions might be particularly appropriated for convolutional neural networks (CNN), although one should take the non-usual periodic structure of the atoms coordinates into account in order to do so.

REFERENCES

[1] Colin R. Groom, Ian J. Bruno, Matthew P. Lightfoot, and Suzanna C. Ward. The Cambridge Structural Database. *Acta Crystallographica Section B Structural Science, Crystal Engineering and Materials*, 72(2):171–179, April 2016.

[2] Federico M. Paruzzo, Albert Hofstetter, Félix Musil, Sandip De, Michele Ceriotti, and Lyndon Emsley. Chemical shifts in molecular solids by machine learning. *Nature Communications*, 9(1):4501, October 2018.