A Data-Driven Approach for Predicting the Elastic Properties of Inorganic Materials

CS 229 Final Project

Camila Cendra

Department of Materials Science and Engineering, Stanford University

ccendra@stanford.edu

1. Introduction

Rational design of application-specific materials is the ultimate goal of modern materials science and engineering[1]. Both materials discovery and materials' properties prediction *in silico* are very active areas of research; currently, large data sets of materials properties obtained from first-principle computational methods are being developed[2,3]. Nevertheless, first-principle calculations require tremendous amounts of computational resources[4]. Machine Learning (ML) methods can provide a way to tackle this computational bottleneck: materials properties can be quickly predicted using a trained ML model, enabling accelerated materials discovery and faster and cheaper development of materials for novel applications.

One intrinsic property of substantial relevance in the screening of materials for novel applications is the material's elastic modulus, which describes the response of the material to external forces. Furthermore, it correlates with many of the material's mechanical and thermal properties[5]. The goal of this project is to aid in the rapid design and screening of new materials by bypassing the need for first-principle computational methods and, instead, use fast supervised ML algorithms to predict the elastic modulus. We use linear regression, a simple neural network, and a random forest regressor to output the predicted elastic modulus.

Our ML models work by performing a pre-processing step to generate a set of descriptive attributes as input features (X): using well-known atomic properties, we generate a list of chemical and physical descriptors. The true labels of the model during training (Y) are the elastic moduli calculated from first-principle computations. We show that simple ML algorithms can be used to predict elastic moduli with relatively high accuracy, achieving a coefficient of correlation of 0.9 and low RMSE.

2. Related Work

To date, several repositories containing materials data have been developed[2,3,6,7], including a dataset with the complete elastic properties of inorganic crystalline compounds[5]. The emergence of comprehensive databases of materials properties is enabling machine learning approaches to quickly predict properties of new materials systems[1,8]. Previous work in the field has shown that ML models can be used to predict a variety of material properties, such as the enthalpy of formation of crystalline compounds[9], bandgap energies of certain classes of crystals[10], vibrational free energies and entropies of crystalline materials[4], and mechanical properties of metallic alloys[11,12], among others.

Previous examples of ML models for materials properties are constructed from three parts: training data, a set of attributes that describe each material, and a ML algorithm to map attributes to predicted properties. Different sets of descriptive attributes have been designed, proposed[1,4], and successfully used to uniquely

describe each material in the dataset, and they are generally related to fundamental physical and chemical properties of the material of interest. For instance, Zhang et al.[10] used a series of manually crafted chemical parameters as descriptive attributes to model the band gaps of binary semiconductors from a small dataset, achieving a Pearson correlation coefficient (r) of 0.86. On similar lines, Legrain et al.[4] recently analyzed the effect of different classes of descriptors in predicting the vibrational free energies and entropies of inorganic solids, revealing that, for large databases a set of descriptors simply based on the compound's chemical formula is able to predict vibrational entropies quite accurately, whereas smaller databases require additional and more elaborate descriptors. Overall, previous work shows that a wide range of materials properties can be successfully predicted by defining a set of descriptive attributes. In this work, we use the guidelines described above to generate a set of descriptive attributes in order to predict the elastic modulus.

3. Dataset and Feature Extraction

As shown in Figure 1, two datasets were utilized to extract the features and true labels for our ML models. The first dataset is a large materials database obtained from the Materials Project[3] and it was curated by Citrine Informatics[1]. It contains first-principle calculations of ~57000 different materials and includes a stream of different relevant materials properties. An algorithm was developed to extract from this large database the chemical name (*raw* X) of materials in which the elastic modulus (*true* Y) was reported, entailing 4208 different samples.

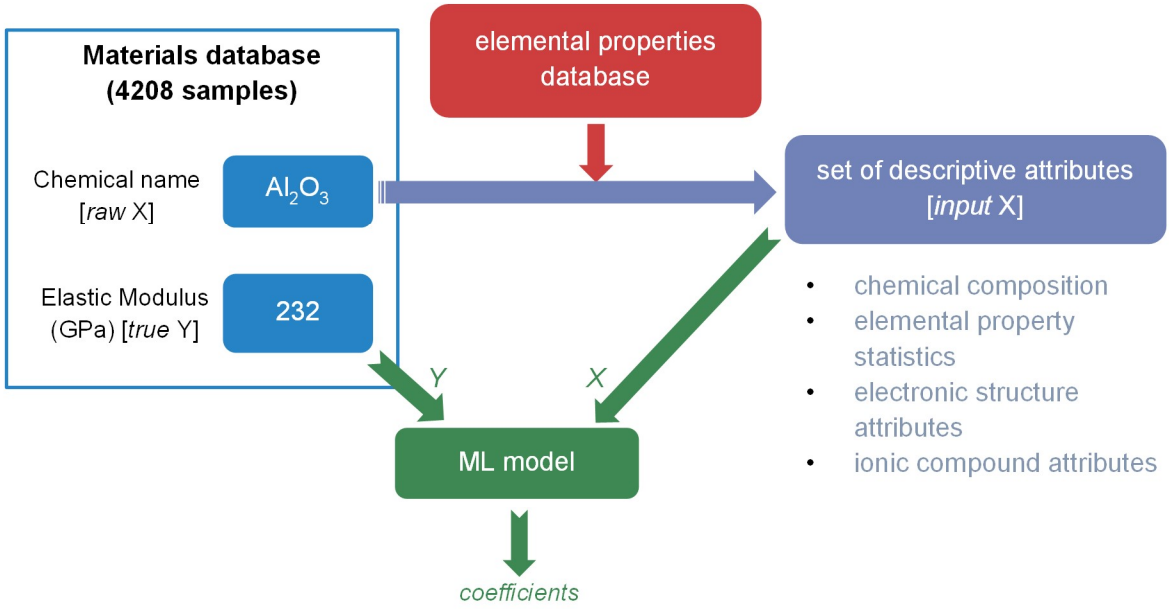

Figure 1. Schematic representation of the dataset and feature generation for training of the ML models. Two parameters are extracted from the materials database (blue box): true elastic modulus and the chemical name of the material (e.g. aluminum oxide, Al_2O_3). Using well-known elemental properties (red box) and the chemical formulation of the material, a set of descriptive attributes are generated. The ML models are then trained using as features descriptive attributes and as labels the elastic modulus for a given material in the training set.

[1] Citrine Informatics: https://citrine.io/

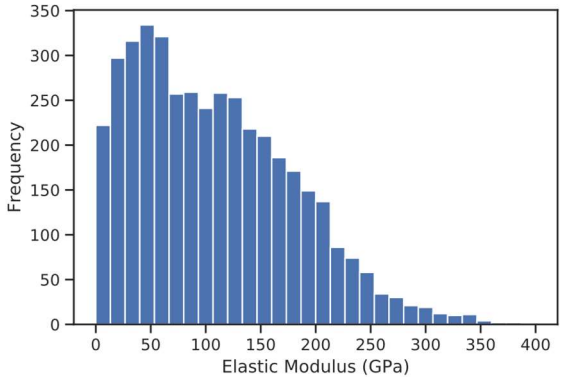

Figure 2. Elastic modulus distribution across the dataset.

The elastic modulus distribution of the dataset is shown in Figure 2. The second dataset is a periodic table of elements including commonly known chemical and physical properties, which was obtained from the web[2] and manually completed. An algorithm was developed to generate a set of descriptive attributes (*input* X) based on the chemical composition of the material and is further described below.

3.1 Descriptive Attributes

The features (X) of the ML models consist of 135 descriptive attributes, 118 of which encode the particular chemical composition under consideration and 17 of which encode heuristic quantities developed using chemical intuition[9]. For example, in aluminum oxide (Al_2O_3), only two of the 118 descriptive attributes are non-zero (Al_fraction = 0.4, O_fraction = 0.6). The remainder 17 descriptive attributes are heuristic quantities[9] obtained for the material:

- Average atomic mass
- Average column on the periodic table
- Average row on the periodic table
- Maximum difference in atomic number
- Average atomic number
- Maximum difference in atomic radii
- Average atomic radius
- Maximum difference in electronegativity
- Average electronegativity
- Average number of *s*, *p*, *d*, and *f* valence electrons (4 features)
- *s*, *p*, *d*, and *f* fraction of valence electrons (4 features)

3.3. Data Preprocessing

After generating a set of descriptive features (X) for the materials database, the data was split into a 3939/537/632 train/test/dev set. The X matrix was standardized to zero mean and unit variance using the training data.

4. Methods

We used three models available from the sklearn Python package[13]: Ridge Regression (RR), Multi-layer Perceptron (MLP) and Random Forest Regressor (RFR).

4.1 Ridge Regression

Ridge regression was used as a benchmark model and it consists of a linear model that minimizes the least squares loss while penalizing the size of the coefficients. The normal equation is as follows:

$$\hat{\beta}_{ridge} = (X^T X + \alpha I_p) X^T Y$$

where I_p is the identity matrix of size 135x135 and α is a pre-chosen penalty term (here, $\alpha = 0.5$).

4.2 Multi-layer Perceptron

MLP is a fully-connected neural network (NN) consisting of at least three layers of nodes: an input layer, a hidden layer, and an output layer. Briefly, in a NN the input layer receives features (X) which are put into linear combination and fed into neurons in the hidden layer(s), where they are

[2] Periodic table of elements: https://github.com/andrejewski/periodic-table

passed into an activation function; in our case, we decided to use a single hidden layer with a rectified linear unit (ReLU) activation function. The outcomes of the hidden layer(s) are then put into linear combination and fed into the output layer. As common in regression problems, we used an identity function for the output layer. Training of the NN is carried out through backpropagation, which allows for minimization of the square error loss function:

$$Loss(\hat{y}, y) = \frac{1}{2}\|\hat{y} - y\|_2^2 + \frac{\alpha}{2}\|W\|_2^2$$

4.3 Random Forest Regressor

RFR is a type of ensembling technique that fits decision trees on subsets of a dataset and uses averaging over the decision trees to improve accuracy of the predictions. Optimization of the number of trees and maximum depth was performed (Figure 3), revealing that the model's performance on the validation set depends on the maximum depth. For our model, we utilized 100 trees and a conservative maximum depth of 15.

4.4 Evaluation Metrics

10-fold cross validation (CV) was used to evaluate model performance. The advantage of CV is that all the instances in the dataset are tested once using a model that did not see that instance while training. The elastic moduli database was randomly divided into 10 segments. 9 segments were used to fit the model and the remaining segment was used to test the model. This procedure was repeated 10 times with different segments. The evaluation criteria employed to evaluate the predictive performance of the models are the coefficient of correlation (r) and the Root-Mean-Squared-Error ($RMSE$). They are defined as follows:

$$r = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

where $\hat{y}_i$ and y_i are the predicted and target elastic modulus, respectively, and $\bar{\hat{y}}$ and $\bar{y}$ are the mean of the predicted and target elastic modulus. n is the number of samples. The coefficient of correlation is a measure of the strength of the relationship between the predicted and the measured values, determining the accuracy of the fitting model (i.e. $r = 1$ shows a perfect positive correlation). $RMSE$ is an error measurement, with smaller error indicating a better prediction accuracy.

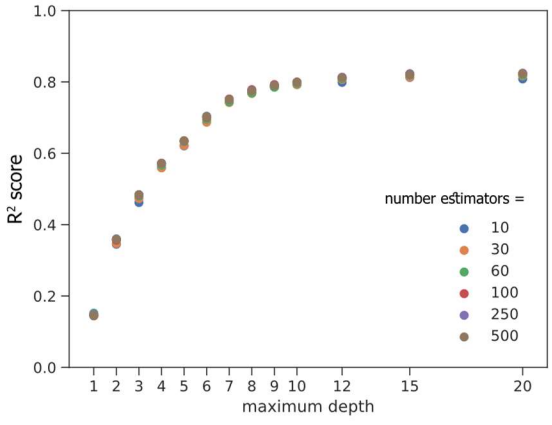

Figure 3. Parameter optimization of the random forest regressor.

5. Results and Discussion

Parity plots reflect the performance of the models (Figure 4). From RR, we can see that the predicted values are somewhat scattered, with decaying accuracy both in the test and train sets for materials with high elastic modulus. For MLP we see a tendency to overfitting, with training points very close to actual values and test points slightly scattered. In the RFR, there is a slight scattering tendency for the test points, though much reduced in comparison to MLP. This is possibly due to the low risk of overfitting random forests. Like RR, the RFR model presents

a) Ridge Regression (RR) **b)** Multi-layer Perceptron (MLP) **c)** Random Forest Regressor (RFR)

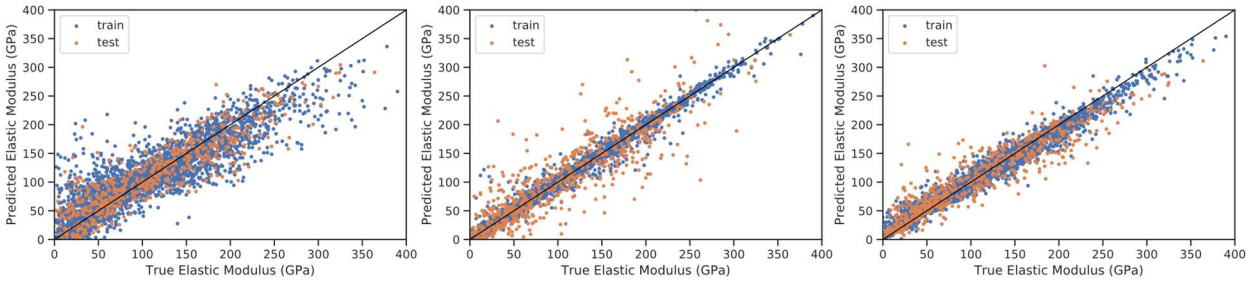

Figure 4. Parity plots for three different regressors: **a)** RR, **b)** MLP, and **c)** RFR. Blue and orange dots denote datapoints corresponding to the train and test sets, respectively. Black continuous lines are shown as a guide to the eye to indicate perfect parity.

Table 1. Evaluation metrics of the ML models.

	train set		test set		10-fold CV	
Model	***RMSE* [GPa]**	***r***	***RMSE* [GPa]**	***r***	***RMSE* [GPa]**	***r***
RR	35	0.88	42	0.84	36 ±16	0.87 ± 0.03
MLP	28	0.92	37	0.88	31 ± 4	0.90 ± 0.02
RFR	27	0.93	38	0.88	31 ± 4	0.90 ± 0.02

decreased accuracy for high elastic modulus materials, possibly due to datapoint scarcity (Figure 2).

Table 1 shows the evaluation metrics of the different models. Overall, MLP and RFR exhibit good and comparable performance, with low RMSE and high coefficients of correlation. The performance achieved in this work is in agreement with previous studies using sets of descriptive attributes to predict properties of materials using ML models[4,10,14].

6. Conclusions and Future Work

We have demonstrated promising results for predicting the elastic modulus of inorganic materials from a set of descriptive attributes which can be readily obtained for any chemical composition. Both MLP and RFR are suitable predictors given the chosen descriptive attributes.

There are many possible routes for future work. The most straightforward route consists on predicting other elastic properties using the developed set of descriptors, such as the shear modulus. Longer-term approaches can focus on implementing advanced ensembling algorithms and partitioning the dataset into groups of similar materials to boost predictive accuracy.

References

[1] L. Ward, A. Agrawal, A. Choudhary, C. Wolverton, *npj Comput. Mater.* **2016**, *2*, 1.

[2] S. Kirklin, J. E. Saal, B. Meredig, A. Thompson, J. W. Doak, M. Aykol, S. Rühl, C. Wolverton, *npj Comput. Mater.* **2015**, *1*.

[3] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S.

Cholia, D. Gunter, D. Skinner, G. Ceder, K. A. Persson, *APL Mater.* **2013**, *1*.

[4] F. Legrain, J. Carrete, A. Van Roekeghem, S. Curtarolo, N. Mingo, *Chem. Mater.* **2017**, *29*, 6220.

[5] M. De Jong, W. Chen, T. Angsten, A. Jain, R. Notestine, A. Gamst, M. Sluiter, C. K. Ande, S. Van Der Zwaag, J. J. Plata, C. Toher, S. Curtarolo, G. Ceder, K. A. Persson, M. Asta, *Sci. Data* **2015**, *2*, 1.

[6] J. E. Saal, S. Kirklin, M. Aykol, B. Meredig, C. Wolverton, *Jom* **2013**, *65*, 1501.

[7] E. Gossett, C. Toher, C. Oses, O. Isayev, F. Legrain, F. Rose, E. Zurek, J. Carrete, N. Mingo, A. Tropsha, S. Curtarolo, *Comput. Mater. Sci.* **2018**, *152*, 134.

[8] E. Gossett, C. Toher, C. Oses, O. Isayev, F. Legrain, F. Rose, E. Zurek, N. Mingo, A. Tropsha, S. Curtarolo, 1.

[9] B. Meredig, A. Agrawal, S. Kirklin, J. E. Saal, J. W. Doak, A. Thompson, K. Zhang, A. Choudhary, C. Wolverton, *Phys. Rev. B - Condens. Matter Mater. Phys.* **2014**, *89*, 1.

[10] Y. Zhang, C. Ling, *npj Comput. Mater.* **2018**, *4*, 28.

[11] S. Chatterjee, M. Murugananth, H. K. D. H. Bhadeshia, *Mater. Sci. Technol.* **2007**, *23*, 819.

[12] H. K. D. H. Bhadeshia, R. C. Dimitriu, S. Forsik, J. H. Pak, J. H. Ryu, *Mater. Sci. Technol.* **2009**, *25*, 504.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, others, *J. Mach. Learn. Res.* **2011**, *12*, 2825.

[14] M. De Jong, W. Chen, R. Notestine, K. Persson, G. Ceder, A. Jain, M. Asta, A. Gamst, *Sci. Rep.* **2016**, *6*, 1.

Developed code available at: https://github.com/ccendra/229_machineLearning