Computational Materials Science 125 (2016) 123–135

Contents lists available at ScienceDirect

Computational Materials Science

journal homepage: www.elsevier.com/locate/commatsci

Critical assessment of regression-based machine learning methods for polymer dielectrics

Arun Mannodi-Kanakkithodi [a], Ghanshyam Pilania [b], Rampi Ramprasad [a,*]

[a] Department of Materials Science and Engineering, Institute of Materials Science, University of Connecticut, 97 North Eagleville Road, Storrs, CT 06269, USA
[b] Materials Science and Technology Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

ARTICLE INFO

Article history:
Received 6 June 2016
Received in revised form 19 August 2016
Accepted 25 August 2016

Keywords:
Materials informatics
Density functional theory
Regression

ABSTRACT

The design of new and improved materials for different applications of interest is boosted by combining computations or experiments with machine learning techniques. Materials scientists seek to use learning algorithms that can easily and efficiently be applied to their data in order to obtain quantitative property prediction models. Here, we utilize a first principles generated dataset of the electronic and dielectric properties of a chemical space of polymers to test different kinds of regression algorithms used by the machine learning community today. We explore several possibilities for the hyper-parameters that go into such learning schemes, and establish optimal strategies and parameters for high-fidelity polymer dielectrics property prediction models.

Published by Elsevier B.V.

1. Introduction

Materials science has greatly benefited in recent times from the application of machine learning techniques to available or newly generated materials data [1–7]. Whereas accurate computations and careful experimental measurements are the standard treatments for new materials discovery, machine learning can accelerate the process significantly, and open up opportunities for exploring complex chemical spaces efficiently. Given any robust dataset of materials properties, learning approaches involve making correlations and mappings between crucial but easily accessible features of materials on the one hand, and the properties of interest on the other. Relationships formed between features and properties can then be exploited for making qualitative, semi-quantitative or quantitative predictions on unseen materials.

In a recent study of designing dielectric polymers for energy storage applications [1,8], we applied machine learning techniques on computational data to develop property prediction models. With respect to high energy density capactitors, polymers suitable to be used as dielectrics should show a high dielectric constant and a large band gap, amongst other crucial features [9,10]. We used density functional theory (DFT) computations to generate dielectric constant (divided into two components, the electronic and the ionic contributions) and band gap data for a selected chemical space of organic polymers. These polymers were built by simple linear combinations of chemical units opted out of a pool of 7 basic blocks: CH_2, NH, CO, C_6H_4, C_4H_2S, CS, and O. For each polymer, DFT helps determine the ground state crystalline arrangement, for which the properties are then computed using known formalisms. Validation of the computed properties against experimental measurements [9] for known polymers makes this a reliable methodology.

Once the property data was generated for 284 polymers (all this data is presented in Refs. [1,8]), it was possible to perform machine learning via an intermediate polymer fingerprinting step. The fingerprint is mapped to the properties—the band gap (in eV), the electronic dielectric constant, and the ionic dielectric constant—to develop an efficient prediction model, that will give as output the properties of any new polymer by converting it into its fingerprint. Once a reasonably accurate prediction model is trained, one can instantly predict the dielectric constants and band gaps of any new polymers that were not considered during computations, thus providing an accelerated materials design route.

Apart from the availability of robust, uniformly generated data, there are a number of other essential factors in the machine learning process that need to be taken care of for optimal learning. These include defining a suitable fingerprint, choosing a learning algorithm, and determining the necessary subset of the data that is needed for training the learning model [5]. The fingerprints we chose and tested in Ref. [1] were chemo-structural in nature, that is, they quantified the types and combinations of different constituent blocks in the polymer. Three fingerprints were used: a count of the different types of building blocks in the polymer,

* Corresponding author.
E-mail address: rampi.ramprasad@uconn.edu (R. Ramprasad).

http://dx.doi.org/10.1016/j.commatsci.2016.08.039
0927-0256/Published by Elsevier B.V.

called fingerprint M_I, a count of the types of block pairs (fingerprint M_{II}), and a count of the types of block triplets (fingerprint M_{III}). The fingerprints were normalized and generalized for any number of blocks in the polymer repeat unit, and used to train a regression model for the three properties of interest.

Whereas all three fingerprints were tested in Ref. [1], the learning algorithm used was Kernel Ridge Regression (KRR) [11]—a nonlinear regression technique that works on the principle of similarity. Euclidean distances between fingerprints were used to quantify the similarity. A distance kernel goes into the definition of the property here, for which a Gaussian kernel was used. Around 90% of the entire polymer dataset was used to train the KRR model, and predictions were made on the remaining points as a test of the performances. Mean absolute errors (MAE) in prediction of less than 10% with respect to the DFT values were seen, which is satisfactory for a statistical model and the best performance that could be obtained using the current optimal learning parameters. The optimal fingerprint used here was M_{III}, with M_{II} and M_I discarded owing to larger prediction errors.

Although we obtained learning models as described above to predict polymer properties with reasonable accuracies, a detailed study of all the different possible machine learning (or regression) parameters is due. Such a study can be very valuable in terms of truly testing the capabilities of our machine learning philosophy for the given polymer dataset, and indeed, improving the performances. In Table 1, we try to capture all these different parameters, mentioning the specific choices that we used in Ref. [1] as well as the other possible options explored here. Whereas the fingerprint choices were already rigorously tested, each of the other parameters provide room for further testing, and thus possible performance improvement.

In this paper, we take the same polymer dataset and analyze the machine learning prediction performances for different regression algorithms, different distance kernel choices, different training set sizes and different error definitions. Possible alternative algorithms to KRR include, but are not limited to: Linear Regression (LR), Support Vector Regression (SVR), Gaussian Process Regression (GPR) and SVR with AdaBoost. Whereas we used KRR with a Gaussian kernel in Ref. [1], Linear, Laplacian or Polynomial kernels can be used as alternatives in any kernel-based regression algorithm. Further, the training set size can be varied systematically to study the prediction errors. The prediction errors can be quantified in different ways, such as mean absolute error (MAE), root mean square error (RMSE) and error based on the coefficient of determination ($1 - R^2$).

In the following sections, we present our results and discussions based on the analysis of all these machine learning parameters. We attempt to compare them critically with each other, and comment on the best possible combination of parameters that must be used given the present polymer dataset.

Table 1
A comparison of various choices of machine learning parameters used in Ref. [1] and explored here. The acronyms used stand for: Kernel Ridge Regression (KRR), Support Vector Regression (SVR), Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and goodness of fit (R^2).

Machine learning parameters	Choices used in Ref. [1]	Choices explored here
Fingerprint	M_I, M_{II}, M_{III}	M_{III}
Regression algorithm	KRR	KRR, SVR, AdaBoost
Type of kernel	Gaussian	Gaussian, Laplacian, linear, polynomial
Training set size	90% of Data	Learning curves
Error definition	MAE	RMSE, $1 - R^2$

2. Kernel Ridge Regression (KRR)

In this section, we delve deeper into KRR, the algorithm that formed the basis of all machine learning prediction models in Ref. [1]. KRR is a similarity based regression algorithm that inherently takes the nonlinearity of the system into account. The 'similarity' between any two data points is defined using some standard mathematical measure of distance, such as a Euclidean distance. For any two polymers i and j having fingerprints $\vec{\mathbf{x}}_i$ and $\vec{\mathbf{x}}_j$ respectively (where $\vec{\mathbf{x}}_i$ is an m dimensional vector with components $x_i^1, x_i^2, x_i^3 \ldots x_i^m$), the Euclidean distance between them will be defined as:

$$d(\vec{\mathbf{x}}_i, \vec{\mathbf{x}}_j) = \|\vec{\mathbf{x}}_i - \vec{\mathbf{x}}_j\|_2 = \sqrt{\left(x_i^1 - x_j^1\right)^2 + \left(x_i^2 - x_j^2\right)^2 + \ldots + \left(x_i^m - x_j^m\right)^2}. \quad (1)$$

The smaller (larger) is this distance, the more similar (dissimilar) the two polymers are. Now, KRR involves defining the property of interest (the output) as a function of such a distance measure, so that the property of any polymer can be estimated by taking its distances from all the other polymers. Mathematically, the predicted property of polymer j, denoted by $P(j)$, will be defined as follows:

$$P_{pred}(j) = \sum_{i=1}^{n} \alpha_i \mathcal{K}(\vec{\mathbf{x}}_i, \vec{\mathbf{x}}_j). \quad (2)$$

The summation is performed over the entire training set size n, and $\mathcal{K}(\vec{\mathbf{x}}_i, \vec{\mathbf{x}}_j)$ is the *kernel function* that is defined in terms of $d(\vec{\mathbf{x}}_i, \vec{\mathbf{x}}_j)$, the distance between polymer i (in the training set) and polymer j. The purpose of the kernel function is to transform the points (the polymers) from the fingerprint space to a higher dimensional space, thus making nonlinear mapping possible [12]. The two crucial parameters that need to be optimized here are the kernel coefficients α_i and the parameters that go into the kernel definition—such as the Gaussian width for a Gaussian kernel. Training of a KRR model essentially involves an iterative minimization of prediction errors leading to the optimal parameter choices.

In practice, as mentioned in the Introduction, the total available dataset is divided into two parts—the training dataset and the test dataset. When training the model using the former, an important step that must be carried out is *cross-validation*, wherein the training set itself is divided into a number of subsets. One of the subsets is used as a temporary test set while training is performed on the remaining subsets, and this procedure is repeated for each of the subsets. The optimal regression parameters are obtained corresponding to minimum average prediction errors on the temporary test sets; subsequently, the error computed over the entire training set with these parameters is referred to as the 'cross-validation error', or sometimes the cross-validated 'training error'. The purpose of cross-validation is to avoid overfitting in the data and to make the model more generalizable—that is, to ensure that the model predictions would work reasonably for points outside the training dataset.

Mathematically, the training process involves a minimization of the following expression:

$$\underset{\alpha_1,\ldots,\alpha_n}{\arg\min} \sum_{i=1}^{n} (P_{pred}(i) - P_{actual}(i))^2 + \lambda \sum_{i=1}^{n} \|\alpha_i\|_2^2. \quad (3)$$

where $P_{pred}(i)$ is the KRR model predicted property value of polymer i as defined in Eq. (2) and $P_{actual}(i)$ is its actual property value; $(P_{pred}(i) - P_{actual}(i))$ is thus a measure of the prediction error. However, the second term in the expression involves the regularization parameter λ. Regularization [2] is an important step that is again aimed at preventing overfitting, and involves adding extra