# Machine learns the physics of intermolecular interactions

Volodymyr Sakun
*Department of Civil Engineering, McGill University, 817 Sherbrooke St. West, Montreal, QC H3A 0C3, Canada*

Rustam Z. Khaliullin*
*Department of Chemistry, McGill University, 801 Sherbrooke St. West, Montreal, QC H3A 0B8, Canada*
(Dated: February 14, 2018)

Abstract is to be written once we finish the main text. Many journals have a 150-word limit on abstract. IMO, best abstracts contain no more than 5 sentences.

## INTRODUCTION

Machine learning shows enormous promise for modeling of materials and molecules [**?** ]. One of the many rapidly growing area of its application is the development of interatomic potentials—functions that describe the dependence of the interaction energy between atoms on their positions. In the last decade, several approaches based on artificial neural networks (ANNs) and Gaussian process regression (GPR) have been developed to approximate interatomic potentials using accurate quantum mechanical data as training data. The main advantage of carefully calibrated machine-learned potentials (MLPs) is that they combine the high accuracy of quantum mechanical description of interaction between atoms with the computational efficiency of simple human-derived analytical potentials. Being the key component for atomistic modeling, MLPs with their rare combination of accuracy and cost have enabled simulations of complex materials on unprecedented time and length scales [**? ? ?** ].

The high accuracy of MLPs is the result of a very flexible form of the model energy function, parametric or nonparametric. This flexibility allows the learning algorithm to adjust the model in order to reproduce the known energies at the training points and then *interpolate* between the points thus predicting the energies for new structures. Unfortunately, the accuracy of the existing MLPs comes with poor transferability. Machine-learning algorithms are unable to *extrapolate* the energy function to the atomistic configurations that do not resemble those in the training set. The transferability problem arises not only for configuration that lie *outside*—according to some distance measure—the training set but also may appear for configurations *inbetween* training points. A typical example of the former case is an MLP trained to reproduce the energies for a material in a certain pressure interval that fails dramatically for any pressure outside this range. The latter situation occurs when the training set is too sparse for accurate interpolation. For example, an MLP trained on configurations collected from liquid phases might not be able to predict energies of its crystal phases even though the local structures in liquid resembles those in solid state. In order to cope with the sparsity problem, training sets must include accurate quantum mechanical energies for a large number of representative atomistic configurations.

While the accuracy-transferability problem plagues many human-derived energy models it is particularly severe in the case of MLPs. For MLPs, the transferability issues summarized above stem from the complete neglect of the physics of interaction between atoms. For example, ANNs rely on unphysical sigmoid functions, whereas GPR can be viewed as a nonparametric energy representation in a set of unphysical Gaussian functions. Thus MLPs are designed as very flexible universal approximators that just *mimic* the functional shape of complex interatomic potentials.

The severity of the transferability problem has inspired numerous attempts to put the physical components back into MLPs. The resulting hybrid approach uses fixed human-derived analytical potentials, in which parameters are now obtained with machine-learning algorithms. While such a scheme can extend the transferability and range of applications of a potential, its fixed analytical form reduces the accuracy of predictions.

In this work, we propose a different approach to learning interatomic potentials that aims at improving their transferability without compromising the flexibility and accuracy. The key idea of the methodology described here is to represent the energy model by a multitude of physically appropriate trial functions and then let the learning algorithm select only those few that reproduce the training data best. A large number of the trial energy functions ensures the accuracy of the model. The physics built into the model and the tight control of overfitting improve its transferability. Furthermore, the Occam's razor principle applied to the learning process produces simple analytical energy expressions that resemble those describing fundamental physical laws. The obtained physically motivated analytical equations can properly describe interactions between atoms outside the training set. To demonstrate a feasibility of the outlined approach, we considered only intermolecular interactions, for which physically relevant energy models can be constructed with linear functions.

## METHODOLOGY

This expression also satisfies another important physical constraints: it makes energy invariant to translations and rotations of the entire molecular system.

For any system, the potential energy as a function of interatomic distances satisfies two constraints: it approaches zero when all atoms are infinitely far from each other and it becomes infinite as the distance between any two atoms approaches zero.

$$\lim_{\forall R_i \to \infty} E(\mathbf{R}) = 0 \tag{1}$$

$$\lim_{R_i \to 0} E(\mathbf{R}) = \infty \tag{2}$$

Define: Features

We chose powers of inverse distances (POIDs) as a set of [features] because their sum can easily satisfy these two very general physical constraints:

$$E(R_1, R_2, \ldots, R_N) = \sum_{k=1}^{M_1} \sum_{p=1}^{N} \frac{c_{pk}}{R_p^k} + \sum_{k,m=1}^{M_2} \sum_{p>q}^{N} \frac{c_{pkqm}}{R_p^k R_q^m} + \\ + \ldots + \sum_{k,m,n=1}^{M_G} \sum_{p>q>r}^{N} \frac{c_{pkqm\ldots rn}}{R_p^k R_q^m \ldots R_r^n} \equiv \mathbf{f} \cdot \mathbf{c} \tag{3}$$

where $N = \frac{1}{2}A(A-1)$ is the number of distances $R$ between $A$ atoms, $M_g$ is the maximum power for a term containing the product of $g$ distances, $G$ is the maximum ZZZ, $\mathbf{c}$ is the vector of tunable coefficients, $\mathbf{f}$ is the vector of features. To make training flexible/ bonds are classified as intermolecular and intramolecular. In order to account for the equivalence of atoms of the same element and to make sure that the energy is invariant to permutation of atoms in the input, the fitting coefficients for the physically equivalent features are restricted to be equal. Taking this symmetry into account allows to decrease the number of independent features in the model and training time substantially. For example, the energy of two water molecules depends on 15 distances, only five of which are unique: oxygen-oxygen, intermolecular oxygen-hydrogen, intramolecular oxygen-hydrogen, intermolecular oxygen-hydrogen, intramolecular hydrogen-hydrogen.

Despite satisfying a variety of constraints, expression (3) remains very general. The multipole expansion of intermolecular energies implies that POIDs can adequately represent a wide variety of intermolecular potentials. The flexibility of the model is very highas it includes many-body beyond simple pairwise interactions through products of POIDs.

**Systems and training set.**

Two water molecules TIP3P.

Two water molecules, MP2.

Two-molecule and three-molecule water models, with geometries obtained from AIMD.

First of all, the data must contain enough records to cover all possible arrangements that exist in real life. In other words, data must contain enough configurations with varying potential from highest in absolute value to zero. Another requirement is that data must be sparse and should have equal records for each representative bin, which is in our case average distance between centers of masses. So, first part of the algorithm takes required amount of records for each interval and that separates training set and test set. Only training set is being using for fitting procedure and only test set is being using for predicting and evaluating the accuracy of the fit and visualizing results. In order to analyze the efficiency of predicting algorithm, the number of observations can be reduced to required quantity. Some algorithms require data standardization, so before doing anything we have to perform feature scaling for training set in order to have zero mean and unit variance for each feature.

[Related to Models] For three water molecules where we have 9 atoms the total number of distances = $9^2/2$ - $9/2$ = 36. Using same set of powers ($M_1$=15 and $M_2$=7) without feature reduction we would have 540+17640=18180 features. However, reducing the number of features using the describe above technique and discarding all intramolecular distances the reduced number of features decreases to only 45+630=675 which is much more convenient and manageable for fitting.

[Related to Models] However, calculation of variance influence factors showed that there exists very strong multicollinearity between features which indicates that products must be included in the model. Similarly, to single distances model, if $M_1$=0 and $M_2$=1, set contains 22 features and new model has in overall 22+5=27 features. Another assumption that has been made that our function depends only on intermolecular distances. After performing several fitting procedures this assumption has been confirmed by results since fitting algorithm discards them almost in all cases. This fact allowed us to reduce the number of features even more. To get reasonable fit, the number of powers must be increased which will increase the number of features. For the model with $M_1$=15 and $M_2$=7 the number of features is 45+245=290 which is more than acceptable for fitting. Without performing feature reduction, we would have in total 225+2940=3165 features (this number of features is the same order of magnitude as that of ANNs-based MLPs). In case of two water molecules reasonable fit could be obtained using only single distances. However, if we have three water molecules, the full model should be used since the simplified one cannot give accurate fit.

**Training procedure.** In this model, POIDs and their products serve as *features*. To find a minimalistic and physically relevant representation of the interatomic potential designed an algorithm that finds a set of the best features while simultaneously tuning their flexible parameters (i.e. linear coefficients in our case) to produce the

best fit. The training procedure includes two stages: feature selection and fine-tuning. The second part is the improvement of the subset performed using correlation analysis (CA) and searching for alternate features.

[OLS]. Main equation. Why is OLS bad algorithm for fitting?

We explored the applicability of two methods to perform feature selection: elastic net regularization (EN) and genetic algorithm (GA). While both algorithms produce almost identical results the advantage of [the?] genetic algorithm is that it can be easily generalized to features with nonlinear fitting coefficients (e.g. exponentially decaying functions can be included in the feature list later).

*Elastic net.* Basic idea and main equation

As it was mentioned if there is a group of highly correlated variables, then the LASSO selects just one variable from a group and drop the others. This is not convenient because in our case LASSO can drop important features while keeping less important one. EN can overcome this problem since it contains two penalties: linear and quadratic. By adjusting the two regularization factors, it is possible to force EN to keep dependent features in the model. Later the unnecessary features will be removed by another algorithm. Therefore, EN allows to reduce the number of features from thousands to hundreds or even less. Next step is to reduce the number of features decreased by EN to desired value. It can be done by the backwards elimination algorithm (BE), which removes the feature Basically, BE removes the worst (less significant feature), one at a time, based on criterion provided by fitness function $F$ until the desired number of features is reached:

$F =$

Since after the reduction of number of features by EN the number of observations is much greater than number of variables, so we can use ordinary least squares (OLS) as a fitness function and the criterion would be minimum squared error (MSE). BE can be used for linear and nonlinear features; the only difference is different fitness functions.

*Genetic algorithm.* Very general description of GA. Population of chromosomes improved by iterative mutations and corssovers.

We used classic genetic algorithm to select best subset of features from the initial set. Since the number of desired selected features is relatively small and for our model does not exceeds 20 features OLS can be used as a fitness function. GA contains many parameters that can be varied in order to achieve better performance and faster convergence. Population size, chromosome size, mutation probability, elite fraction (fraction of good chromosomes), mutation fraction / crossover fraction, mutation methods: correlated or random, crossover method: random or best. Stopping criteria are elapsed time, number of iterations and tolerance.

The main difference from classic GA is that mutation function replaces randomly selected feature with alternate feature from correlation list. The procedure of obtaining lists of alternate features will described in the next two sections.

*Fine-tuning ZZZ.* The second stage of the training processes is designed to deal with strong multicollinearity between features. Since our goal is to reduce the number of POIDs in the fitting function, most common methods of dealing with linear dependent features such as principal component analysis are not applicable. We designed a procedure for eliminating/adding/replacing collinear features with two goals: compact and accurate.

focused on eliminating collinear variables in such a way that final fit will contain only one feature for each collinear group. It is not the ideal representation but it allows us having good enough fit minimize number of variables that would give a compact energy equation. To find out how a linear correlation between all our features we calculated the correlation matrix (i.e. matrix of Pearson correlation coefficients). Next step is to create a list of dependent features for each variable using threshold MinCorrelation between 0 and 1 for cut off. If correlation between two features is greater than MinCorrelation they are considered to be linearly dependent. Suppose we have only 5 features and our correlation matrix looks like in Fig. 1. If we set MinCorrelation = 0.8 than Variable X1 will have a list of dependent variables [X2, X3], X2 will have [X1, X3], X3=> [X1, X2], X4 and will have empty lists that means they are linearly independent within chosen tolerance. The closer MinCorrelation to zero - the greater list of dependent variables.

Its goal is keeping the number of predictors find the best subset of features that gives lowest MSE. BF uses list of dependent variables described above and tries to choose alternate features that are related to the original ones but gives better fit. Algorithm description. Algorithm stops if for any feature from the selected subset no better alternatives was found. Since the subset contains only 20 or less features in our case the algorithm works relatively fast. Its speed depends on MinCorrelation variable that influence the formation of alternate features lists. The closer this variable to 0 the greater lists and the lower performance.

**Final stage.** At this stage, we have few features that were previously selected by either EN or GA. In case of GA this number can be strict since it is equal to chromosome size assigned in GA. For EN this number can be equal to desired value only approximately by varying regularization strength of L1 portions of EN. Let's say we have 20 preselected features from GA. The first goal of final fit is to improve MSE by replacing selected features which can have linearly dependent alternatives and are not selected. Second goal if this procedure is to have 20 sets of different features with number of predictors decreasing from 20 to 1 in order to see how the goodness of

the fit is getting worth with decreasing number of predictors which can give a possibility to analyze the remaining features and to try to give them physical interpretation. The loop has few steps:

1. Improve the fit keeping the number of predictors (BF)
2. Save best possible fit for future analysis
3. Remove worst feature from feature set defined by fitness function: worst feature is the one that gives smallest change in score after removing (BE)
4. Check if number of predictors is greater than 2, otherwise stop

At the end of final stage, we have n sets of linear or exponential predictors with decreasing number from n to 2. As a result, we can see how the goodness of the fit depends on the number of predictors.

## RESULTS AND DISCUSSION

The last part summarizes results in report file which include n models with corresponding fitting coefficients and creates plots that represent prediction of each model using only test set. It also compares our fit with gaussian process fit accomplished using the same training set. In order to have manageable number of features we have to restrict max power to some reasonable number. Our experiments show that for single distance max power $M_1$ from (3) value 15 will be sufficient. For features that contain product of distances max power $M_2$ from (4) equal to 7. Increasing those number to greater value does not give any significant improvement, but augments number of features and slow down the fitting procedure. Table I.

Refer to SI for a discussion of LASSO - a special case of EN.

To results: However, in the fine-tuning algorithm makes greater improvement of the fitting. Usually, using MinCorrelation = 0.8 improvement 10% - 20% can be achieved if we use GA for obtaining initial subset. If we use EN, this improvement can be even greater since EN works faster than GA but gives worst results.

### System A: TIP3P water dimer

We would like to show that our algorithm recovers the correct analytic equation from sparse data. In order to test it and to see how well it would find a simple expression, first of all, we used equation (3) to generate data in the interval where the distance between centers of masses of two water molecules varies from 2 to 8 angstrom. We used range of powers from -1 to -15 for features with single distances and the same range for features that contain product of distances. We did not remove intramolecular distances to see how well our code can reveal the formula we used. Also, for simulation we took only a random numbers of coordinates that equally represent each small bin in chosen interval. The bin size was 0.2 angstrom. Initial number of features was 225 + 12600 = 12825. After performing dimensionality reduction genetic algorithm started with 75 + 3585 = 3660 features and 69 training points per bin (total number of training records was 2139). Initial chromosome size was and number of population was equal to 100. At the and of code execution the exact linear fit with 5 variables and original coefficients were recovered.

$$E(R_i) = c_o\left(\frac{1}{R_{O_1O_2}^{12}} - \frac{1}{R_{O_1O_2}^{6}}\right) + \frac{c_1}{R_{O_1O_2}} - \\ - c_2\left(\frac{1}{R_{O_1H_{21}}} + \frac{1}{R_{O_1H_{22}}} + \frac{1}{R_{O_2H_{11}}} + \frac{1}{R_{O_2H_{12}}}\right) + \\ + c_3\left(\frac{1}{R_{H_{11}H_{21}}} + \frac{1}{R_{H_{11}H_{22}}} + \frac{1}{R_{H_{12}H_{21}}} + \frac{1}{R_{H_{12}H_{22}}}\right) \quad (4)$$

### System B: water dimer

Use \label and \ref to keep track of numbering. All figures should be in the vector format. EPS is the best format. Font in figures must be as large the font in figure caption. Remove caption from the figure. All physical quantities must have labels. Energy units are kJ·mol$^{-1}$. 1 hartree = 2625.499638 kJ·mol$^{-1}$.

In this part, we focus on two water molecules model. First of all, we need to analyze how MSE of the function obtained from fitting procedure depends on number of predictors. We have two functions with varying number of predictors obtained by modeling equations (4) and (5).

Fig. 3 and 4 show the relationship between root of mean squared error (RMSE) and adjusted R2 vs. number of predictors modeled by equation (3). Training set covers all region. Looking at Fig. 3 and 4 we might be interested in results where we have 5 predictors. Looking at detailed description (Fig. 5 and 6) that summarizes all relevant information about the model

Data from Table 1 is nothing else but formula that describes energy for two water molecules system. According to Fig. 1 and considering that physically equivalent distances have the same fitting coefficients it can be summarized in one simple formula:

Terms with power -1 look like Coulombs law. We have only one negative coefficient in front of O-H bond, the rest are positive. The goodness of this fit is summarized in Table 2

Table 2

*Parameters that describe goodness of fit represented by (5)*

TABLE I. Systems and models

| System | A | B1 | B2 | C | D |
|---|---|---|---|---|---|
| Github nickname | set2.x | set6.x | set6.x | set4.x | |
| $G$ | 1 | 1 | 2 | 2 | |
| $M_1$ | 15 | 15 | 15 | 15 | |
| $M_2$ | 0 | 0 | 7 | 7 | |
| Chemical composition | $(H_2O)_2$ | $(H_2O)_2$ | $(H_2O)_2$ | $(H_2O)_3$ | |
| No. of unique features | | | | | |
| Ref. energy | TIP3P | MP2 | MP2 | BLYP+D/mo-TZV2P | |
| Configurations | Grid | $R_{OO}$ bins | $R_{OO}$ bins | AIMD | |

TABLE II. Table 1. Final five-feature fit for system ZZZ obtained with GA/EN.

| Distance | Power, $n$ | Coefficient, $Ha \cdot Å^n$ |
|---|---|---|
| O-O | 12 | 470.76 |
| O-H | 1 | -0.11866 |
| O-O | 1 | 0.23203 |
| H-H | 1 | 0.060300 |
| H-H | 5 | 0.022497 |

| **MSE Train** | **RMSE Train** | **R2 Train** | **R2 Adjusted Train** | **MSE Test** | **RMSE Test** | **R2 Test** | **R2 Adjusted Test** |
|---|---|---|---|---|---|---|---|
| 4.01E-07 | 0.000633 | 0.92285 | 0.922759 | 4.13E-07 | 0.000643 | 0.949011 | 0.94877 |

After performing fitting using full features set according to (4) we obtained similar graphs that represents relationship between RMSE or R2 and number of predictors. If we take a closer look at details of fit with 5 predictors we could observe that the goodness of fit if better. The algorithm chose 4 out of 5 features that contain product of distances which signifies that they are not completely independent and there exists some relationship between them.

TABLE III. Table 3. Coefficients for (4) with bond descriptions and corresponding number of distances and powers

| **Bond 1** | **Power 1** | **Bond 2** | **Power 2** | **Number of distances in feature** | **Coefficients** |
|---|---|---|---|---|---|
| H-H | -2 | | | 4 | 0.171764693 |
| O-O | -1 | H-H | -1 | 4 | 0.180766608 |
| O-O | -6 | O-H | -4 | 4 | 23.07894762 |
| O-H | -1 | H-H | -1 | 8 | -0.213562728 |
| O-H | -1 | H-H | -1 | 8 | 0.036990729 |

It should be mentioned that last two rows in table 3 represent different pairs of distances. There are 7 types of pairs of distances applicable to this model and they will be described later.

Another interesting aspect is to compare our results with Gaussian process. At the same time, we have to investigate one of our goals: can our algorithm predict in regions where there are no training points, in other words can we use it to interpolate and extrapolate. To do this we removed training points from some regions. In the following example simulation will be performed in the region where average distance between center of masses of two water molecules spans [2.4 .. 15] angstrom. However, we removed all data in region [5 .. 7] angstrom to see how interpolation works and at the edge region [9 .. 15] angstrom to observe extrapolation. It has to mentioned that we kept test points in entire region [2.4 .. 15] angstrom. In addition, predicted values of energy obtained by Gaussian process are to be compared with our method.

FIG. 1. Fig. 6. $G = 2$.

TABLE IV. Table 4. Parameters that describe goodness of fit with coefficients from Table 3

| **MSE Train** | **RMSE Train** | **R2 Train** | **R2 Adjusted Train** | **MSE Te** |
|---|---|---|---|---|
| 1.68E-07 | 0.00041 | 0.973192 | 0.97316 | 1.65E-07 |

FIG. 2. Fig. 8. *Average bin error vs. average distance between centers of masses. Two water molecules model. Single distance features from (3) with 7 predictors and Gaussian Process. Trained regions [2.4..5], [7..9] angstrom, test region [2.4..15] angstrom*

Fig. 8 and 9 clearly show that in trained region [5..7] both algorithms give pretty well prediction where error is relatively small and predicted energy almost identical with true energy. In the [2.4..5] angstrom region error is greater as well as deviation of predicted energy since on short distances the behavior of two water molecules system is much more complex and true energy function has more complex shape. However, if we examine two regions where there are no training points we can see a big difference in prediction. Our linear model is still good: predicted energy values are close to true energy and level of the error is almost the same as in adjacent trained region. Gaussian process behaves differently: error goes up and values of predicted energy also higher, especially

FIG. 3. Fig. 9. Average energy per bin vs. average distance between centers of masses. Two water molecules model. True energy plot. Single distance features from (3) with 7 predictors and Gaussian Process. Trained regions [2.4..5], [7..9] angstrom, test region [2.4..15] angstrom

et the edge region [9..15] angstrom. It can be observed from histogram on Fig. 10 that error distribution is reasonable since the most frequently occurred error values are close to zero.

FIG. 4. Fig. 10. Absolute error histogram for function with 7 predictors modeled by genetic algorithm according to (3). Two water molecules model. Trained regions [2.4..5], [7..9] angstrom, test region [2.4..15] angstrom.

Similarly, to those results Fig. 11, 12, 13 represent the model according to (4) with 6 predictors and the same other parameters. The fit is even better than if we used eq. (3), but general behavior is similar.

FIG. 5. Fig. 11. Average bin error vs. average distance between centers of masses. Two water molecules model. Single distance features from (4) with 6 predictors and Gaussian Process. Trained regions [2.4..5], [7..9] angstrom, test region [2.4..15] angstrom

Another important aspect is to examine the dependence of goodness of fit on number of training points. To do that we used model based on eq. (4) and trained regions [2.4..5], [7..9] and test region [2.4 .. 15] angstrom. However, instead of using all training points we used 10 different fits with number of training points from 10% of initial number up to 100% with the step 10%. Both linear model and Gaussian process were examined in equal conditions. The results are represented by Fig. 14 and 15. It can be clearly seen that Gaussian process requires more training points to obtain reasonable prediction. Linear model behaves better when we used reduced number of training points and gave reasonable results even at 10% of initial set. All points were chosen randomly for each small interval that represent some value measured as average distance between the centers of masses of two water molecules.

## CONCLUSIONS

In this work, we proposed a new machine learning strategy to reproduce the interaction energy between molecules based on a limited training set of accurate quantum mechanical data. Unlike existing machine learning techniques, the new methodology is designed to incorporate physically relevant information into the model while still retaining its high flexibility. In addition

FIG. 6. Fig. 12. Average energy per bin vs. average distance between centers of masses. Two water molecules model. True energy plot. Single distance features from (4) with 6 predictors and Gaussian Process. Trained regions [2.4..5], [7..9] angstrom, test region [2.4..15] angstrom

FIG. 7. Fig. 13. Absolute error histogram for function with 6 predictors modeled by genetic algorithm according to (4). Two water molecules model. Trained regions [2.4..5], [7..9] angstrom, test region [2.4..15] angstrom.

to this, the learning strategy is developed around a strict control of overfitting and thus selects only the most important components of model. It is demonstrated, using water molecules as an example, that this approach recovers fundamental physical laws that govern intermolecular interactions. As a result, the obtained models do not only reproduce quantum mechanical energies in the region spanned by the training set configurations but are also capable of extrapolating well outside this region – a remarkable achievement for machine learning in molecular modeling. Moreover, it is shown that incorporating physically motivated features into the model drastically reduces the number of reference data points in the training set without sacrificing the quality of predictions. In the future, the approach presented here will be extended to systems of strongly interacting atoms that may require incorporating more complex nonlinear physical features.

## COMPUTATIONAL METHODS

**Gaussian process regression.** To better understand the accuracy and the performance of our method it would be nice to compare its results with the ones obtained by using another machine learning algorithms such as artificial neural network or gaussian process. For this article, Gaussian process has been chosen since it can " mimic" the shape of almost any complex function. We used GaussianProcessRegressor from sklearn.gaussian_process which is the implementation of Gaussian Processes for Machine Learning by Rasmussen and Williams. During fitting the hyperparameters of the kernel are optimized by maximizing the log-marginal-likelihood based on the passed optimizer. We used two kernels RBF which is basically Radial-basis function or squared-exponential kernel and White Kernel the main use of which is to add noise level component to the signal. Those two kernels have two crucial parameters: length scale and noise level respectively. Varying those parameters, it is possible to find the local maxima of the function to achieve the best possible fit. Either optimizer provided by scipy.optimize.fmin_l_bfgs_b or user defined optimizer that must have specific signature can be used to attune kernels and therefore GP regressor to give op-

FIG. 8. Fig. 14. RMSE vs. % of usage of training points. Two water molecules. Linear model (4) with 5 to 15 predictors and Gaussian process. Trained regions [2.4..5], [7..9] angstrom, test region [2.4..15] angstrom.

FIG. 9. Fig. 15. R2 vs. % of usage of training points. Two water molecules. Linear model (4) with 5 to 15 predictors and Gaussian process. Trained regions [2.4..5], [7..9] angstrom, test region [2.4..15] angstrom.

timal results. Since local maxima is a function of two main variables, to find optimal solution we used a grid with varying scale (Fig. 2). Both optimizers gave almost identical results.

## ACKNOWLEDGMENTS

The research was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) through Discovery Grants (RGPIN-2016-0505). The authors are grateful to Compute Canada and, in particular, the McGill HPC Centre for computer time.

### LASSO

We have explored performance of least absolute shrinkage and selection operator (LASSO) method for the feature selection stage of the training algorithm. LASSO is a special case of the elastic net regularization with ZZZ coefficient being set to zero. We found that LASSO, based on both forward stepwise and least angle regression, tends to drop important features that are somewhat collinear to those already selected. As a result, the final list selected by LASSO contains too few features.

In a multiple regression model multicollinearity indicates that the collinear independent variables are somehow related. The presence of multicollinearity within the set of features causes many problems in the understanding the significance of individual independent features in our regression model. Our feature set is generated in such a way that it is impossible to avoid collinearity and multicollinearity since basically we have set of distances raised to different powers. The variance inflation factors (VIF) can prof that we have multicollinearity issues in our set. It allows a quick estimation of how much a variable is contributing to the standard error in our regression model. VIF is very large when significant multicollinearity issues exist. Multicollinearity can lead to less reliable probability values and larger confidence intervals.

| | X1 | X2 | X3 | X4 | X5 |
|---|---|---|---|---|---|
| X1 | 1.00 | 0.95 | 0.90 | 0.30 | 0.20 |
| X2 | 0.95 | 1.00 | 0.85 | 0.70 | 0.30 |
| X3 | 0.90 | 0.85 | 1.00 | 0.60 | 0.40 |
| X4 | 0.30 | 0.70 | 0.60 | 1.00 | 0.25 |
| X5 | 0.20 | 0.30 | 0.40 | 0.25 | 1.00 |

Fig. 2. *Correlation matrix example for 5 variables*

**Best fit algorithm.** Schematically the algorithm looks like following pseudocode:

```
while found:
found = False
for feature in features:
    for alternate in feature.alternates:
        Replace_temporarely(alternate, features)
        new_score = Score(features)
        if new_score < best_score:
            found = True
            Replace_permanently(alternate, features)
            best_score = new_score
            continue
```

where:

feature.alternates - a list of alternating (most correlated) features created previously in correlation analysis part

Replace_temporarely - temporarily replacement of current feature by its alternative from the list

Replace_permanently - permanent replacement of current feature by its alternative. It also means that the list feature.alternates is been replaced too by appropriate one from correlation analysis

Score - some fitness function that evaluates the goodness of the fit. For linear model we used OLS and for exponential fit we used scipy.optimize.least_squares.