Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

An SVD and Derivative Kernel Approach to Learning from Geometric Data

Eric Wong
School of Computer Science
Carnegie Mellon University
ericwong0@cmu.edu

J. Zico Kolter
School of Computer Science
Carnegie Mellon University
zkolter@cs.cmu.edu

Abstract

Motivated by problems such as molecular energy prediction, we derive an (improper) kernel between geometric inputs, that is able to capture the relevant rotational and translation invariances in geometric data. Since many physical simulations based upon geometric data produce derivatives of the output quantity with respect to the input positions, we derive an approach that incorporates derivative information into our kernel learning. We further show how to exploit the low rank structure of the resulting kernel matrices to speed up learning. Finally, we evaluated the method in the context of molecular energy prediction, showing good performance for modeling previously unseen molecular configurations. Integrating the approach into a Bayesian optimization, we show substantial improvement over the state of the art in molecular energy optimization.

Introduction

This paper focuses on learning from geometric data, where each input to the learning problem consists of a set of points, typically in two or three dimensional space. While many physical problems take this form, we focus here on the task of molecular energy prediction from input data describing the 3D configuration of atoms. This problem has been studied before in the machine learning literature, though in a slightly different context (Rupp et al. 2012; Montavon et al. 2012), and has numerous applications ranging from energy materials to drug design. The basic task here is, given a set of atoms and their respective 3D positions, to predict the energy of the resulting configuration; since molecules in nature will typically settle in minimum energy configurations, in finding minima of these functions we can assist in the design of new materials. Physically-based simulation to determine these configurations, notably methods based upon density functional theory (DFT) approaches, are a very mature technology, but are a very computationally intensive process. The canonical problem we consider in this paper, therefore, is the task of learning a faster surrogate function for this expensive process, using machine learning techniques. This can be used simply as a means of predicting the energy of new configurations, or within a Bayesian optimization procedure for finding the minimum energy configuration for a molecule.

A main contribution of this paper is the development of a (improper) kernel approach to dealing with this data, based upon the singular value decomposition. Geometric learning problems of the type we consider typically exhibit certain invariances, such as translational and rotational invariance (i.e., rotating or translating all the atoms on a molecule do not actually change any qualitative information about the molecule, since it remains the same molecule), and we derive a kernel formulation that respects these invariances. Furthermore, because many physical simulations that serve as the input to such geometric methods (notably those based upon adjoint methods) can also provide *derivatives* of the output quantity with respect to all point locations, we derive an approach to including derivative information into our kernel learning. Finally, we highlight the method on examples from the molecular energy prediction task, showing good performance in modeling previously unseen configurations. Incorporating the method into a Bayesian optimization framework, we show significant improvement over existing state of the art in molecular energy optimization.

Related work

This work builds upon a number of different approaches in kernel methods, computer vision, Bayesian optimization, and applications to material science. Kernel methods have long been applied to spatial data in general (see, e.g. (Pozdnoukhov 2009)), which shares a high-level similarity to the work we present here in that both methods use 2D or 3D coordinates as input to a learning system. However, the work presented here differs substantially from most work in spatial learning, as the goal there is typically to predict a quantity at a *single* spatial point (and thus the notions of translational and rotational invariance are specifically inapplicable). In contrast, the work here looks at developing a kernel that measures similarity between configurations of *multiple* points, while respecting certain invariances in the data.

The basic singular value decomposition (SVD) approach that we use, for finding optimal translation and rotations between molecules, has been previously studied for some time (Horn, Hilden, and Negahdaripour 1988), and has been used in the vision community as a subroutine for fast geometric transforms (Lu, Hager, and Mjolsness 2000). However, the main contributions over this basic approach in the current work are: 1) the introduction of a kernel based upon this SVD-based distance metric, and 2) the derivation of functional derivatives of this kernel in a compact form, which is of particular importance for the domain of molecular energy prediction. This last item also exploits work in kernel methods for incorporating derivative information in Gaussian processes (Solak et al. 2003); we use exactly this approach in order to incorporate observations of the molecular forces (derivatives) into our model, though the challenging aspect here is computing the proper kernel derivatives, which is one of the primary contributions of this paper.

Our work also uses recent approaches to Bayesian optimization (Brochu, Cora, and De Freitas 2010). From the core algorithmic perspective, we are mainly using existing approaches from the literature, specifically the lower confidence bound approach (Cox and John 1997) that is widely used in the machine learning community. However, most Bayesian optimization approaches have been applied using relatively simple forms of kernels such as the exponential or Matern kernels (Snoek, Larochelle, and Adams 2012). In contrast, the use of the geometric kernel that we derive here lets us use Bayesian optimization as a drop in replacement for existing molecular energy optimizers, based upon traditional optimization approaches such as gradient descent or LBFGS. In this context, the work here can be viewed as integrating Bayesian optimization in this approach to numerical scientific computing.

Finally, in the domain of material science, our approach builds upon a number of past works. Both neural networks and Gaussian processes have been used to predict the potential energy of a single molecule (Behler and Parrinello 2007; Bartók et al. 2010), but these used either black box or non-invariant kernel approaches, and do not integrate derivative observations. More recent approaches learn across the entire chemical compound space by using data from multiple molecules. These approaches represent molecules by their so-called Coulomb matrix and their corresponding vectors of eigenvalues (Rupp et al. 2012; Montavon et al. 2013). The major difference presented here is the use of the singular value decomposition to define the distance between two inputs and the incorporation of the available molecular energy derivative information.

An SVD kernel for geometric data

In this section we present the main algorithmic contribution of this paper, an improper kernel for geometric input data. In the sequel, we will use this kernel in the context of Gaussian process estimation and Bayesian optimization, but the key contribution here relates to the derivation of the kernel itself and its corresponding derivatives that can enable us to incorporate derivative information into the kernel.

Let $X \in \mathbb{R}^{3\times n}$ represent molecules of n atoms, where each column of the matrix denotes the 3D coordinate of the corresponding atom. Since translating or rotating a molecule results in the an unchanged molecule, a measure of similarity between two molecules should be invariant to these properties. To construct a Gaussian process model for such data, we define the distance between two molecules $X \in \mathbb{R}^{3\times n}$ and $Z \in \mathbb{R}^{3\times n}$ as the minimum distance over all rotations and translations:

$$d(X,Z) = \min_{R\in\mathbb{R}^{3\times 3}, t\in\mathbb{R}^3, R^TR=I_3} ||X-(RZ+t1^T)||_F^2 \quad (1)$$

Although this is a non-convex problem, it can be solved efficiently using the singular value decomposition. This is a well-known property (see, e.g. (Horn, Hilden, and Negahdaripour 1988)), but we include a brief derivation for completeness.

Theorem 1. *The solution to the minimization in* (1) *is given by*

$$t^\star = (X - R^\star Z)1/n, \quad R^\star = VU^T \quad (2)$$

where $USV^T = XBB^TZ^T$ is a singular value decomposition and B is the centering matrix defined as $B = I_n - 11^T/n$. Furthermore, the distance takes the closed form

$$d(X,Z) = ||XB||_F^2 + ||YB||_F^2 - 2\operatorname{Tr} S \quad (3)$$

where S is the diagonal matrix of singular values as above.

Proof. Taking the gradient of (1) with respect to t

$$\nabla_t ||X - RZ - t1^T||_F^2 = (X - RZ - t1^T)1 \quad (4)$$

gives the the first part of the solution

$$t^\star = (X - RZ)1/n. \quad (5)$$

Substituting this back into the objective transforms (1) into the equivalent optimization problem

$$d(X,Z) = \min_{R^TR=I_3} ||XB - RZB||_F^2 \quad (6)$$

and note that

$$\begin{aligned} &||XB - RZB||_F^2 \\ &= ||XB||_F^2 + ||RZB||_F^2 - 2\operatorname{Tr} B^TZ^TR^TXB \quad (7)\\ &= ||XB||_F^2 + ||ZB||_F^2 - 2\operatorname{Tr} R^TXBB^TZ^T. \end{aligned}$$

We next take the singular value decomposition $USV^T = XBB^TZ^T$

$$\operatorname{Tr} R^TUSV^T = \operatorname{Tr} V^TR^TUS = \operatorname{Tr} \tilde{R}S \quad (8)$$

where $\tilde{R}^T\tilde{R} = I$. Since S is diagonal, this problem is optimized with $\tilde{R}^\star = I$, so that $R^\star = UV^T$. The final form of the distance follows from the fact that

$$\operatorname{Tr} VU^TUSV^T = \operatorname{Tr} U^TUSV^TV = \operatorname{Tr} S. \quad (9)$$

□

We use this distance in a corresponding exponential kernel given by

$$K(X,Z) = \exp\left\{-\frac{1}{2}\gamma(||XB||_F^2 + ||ZB||_F^2 - 2\operatorname{Tr} S)\right\}. \quad (10)$$

Note that this is not a proper (semidefinite) kernel, due to the fact that this distance function is not a true metric (it does not obey the triangle inequality); thus the kernel matrix above may have negative eigenvalues. In this case, we can simply treat the kernel as a set of features describing the input (i.e, using the matrix K^TK to compute predictions); we can also exploit the fact that, for large enough γ, the kernel *will* be guaranteed to be positive definite, as all non-diagonal entries will fall off exponentially. In practice, we observe that this is rarely a problem, and all values of γ chosen by cross validation, for instance, results in positive definite kernels.

Kernel derivatives

A common theme in many learning methods that use geometric data, specifically those where the geometric data is itself generated through some simulation process, also produce *derivatives* of the output quantity with respect to all the input coordinates. For instance, the general class of methods known as *adjoint methods* produce these derivatives using approximately twice as much computation as it takes to compute the output originally. Such methods have therefore become extremely widely used in simulation and optimization, and we want to enable our kernel approach to use such information. As shown in previous work (Solak et al. 2003), a Gaussian process can incorporate derivative observations by defining an extended kernel matrix that also models the covariance between the inputs and their derivatives, and between the derivatives themselves. In particular, since the covariance is a bilinear operator:

$$\begin{aligned} \operatorname{Cov}\left(y^{(s)}, y^{(t)}\right) &= K\left(X^{(s)}, X^{(t)}\right) \\ \operatorname{Cov}\left(\frac{\partial y^{(s)}}{\partial X^{(s)}_{ij}}, y^{(t)}\right) &= \frac{\partial K(X^{(s)}, X^{(t)})}{\partial X^{(s)}_{ij}} \\ \operatorname{Cov}\left(\frac{\partial y^{(s)}}{\partial X^{(s)}_{ij}}, \frac{\partial y^{(t)}}{\partial X^{(t)}_{k\ell}}\right) &= \frac{\partial^2 K(X^{(s)}, X^{(t)})}{\partial X^{(s)}_{ij}\partial X^{(t)}_{k\ell}}. \end{aligned} \tag{11}$$

In the remainder of this section, we will derive a closed form expression for these derivatives for the SVD kernel. For convenience, it is useful to combine all the kernel derivatives between two molecules $X, Z \in \mathbb{R}^{3\times n}$ in the block form:

$$\begin{bmatrix} K(X,Z) & \frac{\partial K(X,Z)}{\partial Z_{11}} & \cdots & \frac{\partial K(X,Z)}{\partial Z_{3n}} \\ \frac{\partial K(X,Z)}{\partial X_{11}} & \frac{\partial^2 K(X,Z)}{\partial X_{11}\partial Z_{11}} & \cdots & \frac{\partial^2 K(X,Z)}{\partial X_{11}\partial Z_{3n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial K(X,Z)}{\partial X_{3n}} & \frac{\partial^2 K(X,Z)}{\partial X_{3n}\partial Z_{11}} & \cdots & \frac{\partial^2 K(X,Z)}{\partial X_{3n}\partial Z_{3n}} \end{bmatrix} \tag{12}$$

Theorem 2. *For the rotationally invariant distance kernel in* (10)*, the block of derivatives in (12) has the following form:*

$$\bar{K}(X,Z) = \exp\left\{-\frac{1}{2}\gamma d(X,Z)\right\} \begin{bmatrix} \bar{K}_{11} & \bar{K}_{12} \\ \bar{K}_{21} & \bar{K}_{22} \end{bmatrix} \tag{13}$$

where

$$\begin{aligned} \bar{K}_{11} =& 1 \\ \bar{K}_{12} =& -\gamma \operatorname{vec}(Z - VU^T X)^T \bar{B}^T \\ \bar{K}_{21} =& -\gamma \bar{B} \operatorname{vec}(X - UV^T Z) \\ \bar{K}_{22} =& \gamma \bar{B}\left((I_n \otimes UV^T)\right. \\ & \left. -(Z^T V \otimes U)\mathcal{A}(U^T X \otimes V^T)\right)\bar{B}^T \\ & + \bar{K}_{21}\bar{K}_{12} \end{aligned} \tag{14}$$

with $\otimes$ as the Kronecker product, $\bar{B} = (B \otimes I_3)$, and

$$\mathcal{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\hat{s}_{12}} & 0 & \frac{-1}{s_{12}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{s_{13}} & 0 & 0 & 0 & \frac{-1}{s_{13}} & 0 & 0 \\ 0 & \frac{-1}{s_{12}} & 0 & \frac{1}{s_{12}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{s_{23}} & 0 & \frac{-1}{s_{23}} & 0 \\ 0 & 0 & \frac{-1}{s_{13}} & 0 & 0 & 0 & \frac{1}{s_{13}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-1}{s_{23}} & 0 & \frac{1}{s_{23}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{15}$$

where $s_{ij} = S_{ii} + S_{jj}$.

Proof. To derive these expressions, we determine the derivatives of the singular value decomposition using a methodology similar to the work of (Papadopoulo and Lourakis 2000), beginning with the following identity:

$$USV^T = XBBZ^T \tag{16}$$

We take the derivative of both sides with respect to some input coordinate $X_{i,j}$, multiply on the left and right by U^T and V respectively, and take the trace. Since $U^T\left(\frac{\partial U}{\partial X_{ij}}\right)$ and $\left(\frac{\partial V^T}{\partial X_{ij}}\right)V$ are antisymmetric, we get:

$$\operatorname{tr}\left(\frac{\partial S}{\partial X_{ij}}\right) = \operatorname{tr}(J^{ij} BZ^T VU^T) \tag{17}$$

with $(J^{ij})_{kl} = \delta_{ik}\delta_{jl}$. Let b_j be the jth column of B. Substituting the value in equation 17 for the derivative of the singular values, the first derivative of the distance metric with respect to X_{ij} can be expressed as the following:

$$\frac{\partial d(X,Z)}{\partial X_{ij}} = 2(b_j \otimes e_i)^T \operatorname{vec}(X - UV^T Z) \tag{18}$$

By varying i and j, we can thus construct a row of all the first derivatives of X:

$$\frac{\partial d(X,Z)}{\partial \operatorname{vec}(X)} = 2(B \otimes I_3)^T \operatorname{vec}(X - UV^T Z) \tag{19}$$

We substitute into the derivative of the kernel to get:

$$\frac{\partial K(X,Z)}{\partial \operatorname{vec}(X)} = \exp\left\{-\frac{1}{2}\gamma d(X,Z)\right\}\bar{K}_{21} \tag{20}$$

An almost identical process results in the respective expression for the derivative with respect to the second input:

$$\frac{\partial K(X,Z)}{\partial \operatorname{vec}(Z)} = \exp\left\{-\frac{1}{2}\gamma d(X,Z)\right\}\bar{K}_{12} \tag{21}$$

For the second derivatives, we begin by directly taking the second derivative of the kernel:

$$\frac{\partial^2 K(X,Z)}{\partial Z_{kl}\partial X_{ij}} = K(X,Z)\,(\alpha - \beta) \tag{22}$$

where

$$\begin{aligned} \alpha &= \frac{1}{4}\gamma^2\left(\frac{\partial d(X,Z)}{\partial Z_{kl}}\right)\left(\frac{\partial d(X,Z)}{\partial X_{ij}}\right) \\ \beta &= \frac{1}{2}\gamma\frac{\partial^2 d(X,Z)}{\partial Z_{kl}\partial X_{ij}} \end{aligned} \tag{23}$$

The first part is expressed from the first derivative results:

$$\alpha = [\bar{K}_{21}]_{(ij)}[\bar{K}_{12}]_{(kl)} \tag{24}$$

where the subscripts $(ij), (kl)$ refer to the indices of the derivative with respect to X_{ij}, Z_{kl}. For the second part, taking the second derivative of the distance metric shows that we need the second derivative of the singular values:

$$\frac{\partial^2 d(X,Z)}{\partial Z_{kl} \partial X_{ij}} = -2 \operatorname{tr}\left(\frac{\partial^2 S}{\partial Z_{kl} \partial X_{ij}}\right) \tag{25}$$

To get this quantity, we first take the second derivative of the identity $USV^T = XBB^TZ^T$ and multiply on the left and right by U^T and V respectively. Taking the trace of the resulting equation, we get

$$\begin{aligned}\operatorname{tr}\left(\frac{\partial^2 S}{\partial Z_{kl} \partial X_{ij}}\right) &= \operatorname{tr}\left(U^T J^{ij} B J^{lk} V\right) \\ &\quad - \operatorname{tr}\left(\left(\Omega_U^{ij} + \Omega_V^{ij}\right) S \left(\Omega_U^{kl} + \Omega_V^{kl}\right)\right)\end{aligned} \tag{26}$$

where

$$\begin{aligned}\Omega_U^{ij} &= U^T \frac{\partial U}{\partial X_{ij}} \quad \Omega_U^{kl} = U^T \frac{\partial U}{\partial Z_{kl}} \\ \Omega_V^{ij} &= \frac{\partial V^T}{\partial X_{ij}} V \quad \Omega_V^{kl} = \frac{\partial V}{\partial Z_{kl}} V\end{aligned} \tag{27}$$

and $(J^{ij})_{kl} = \delta_{ik}\delta_{jl}$. For the latter term, we can solve for an explicit form of Ω_U^{ij} and Ω_V^{ij} by solving the system of equations given by the off-diagonal entries of the first derivative of equation 16. Using these, we can compute

$$\Omega_U^{ij} + \Omega_V^{ij} = \begin{bmatrix} 0 & \frac{W_{12}^{ij} - W_{21}^{ij}}{s_1 + s_2} & \frac{W_{13}^{ij} - W_{31}^{ij}}{s_1 + s_3} \\ -\frac{W_{12}^{ij} - W_{21}^{ij}}{s_1 + s_2} & 0 & \frac{W_{23}^{ij} - W_{32}^{ij}}{s_2 + s_3} \\ -\frac{W_{13}^{ij} - W_{31}^{ij}}{s_1 + s_3} & -\frac{W_{23}^{ij} - W_{32}^{ij}}{s_2 + s_3} & 0 \end{bmatrix} \tag{28}$$

where $W^{ij} = U^T J^{ij} B Z^T V$. We can get an identical expression for $\Omega_U^{kl} + \Omega_V^{kl}$, using $W^{kl} = U^T X B J^{lk} V$ in place of W^{ij}. Substituting these into the trace in equation 26 and using $\mathcal{A}$, we derive the following expression:

$$\begin{aligned}&\operatorname{tr}\left(\left(\Omega_U^{ij} + \Omega_V^{ij}\right) S \left(\Omega_U^{kl} + \Omega_V^{kl}\right)\right) \\ &\quad = \frac{1}{s_1 + s_2}\left(-W_{12}^{ij}W_{12}^{kl} + W_{12}^{ij}W_{21}^{kl} - W_{21}^{ij}W_{21}^{kl} + W_{21}^{ij}W_{12}^{kl}\right) \\ &\quad + \frac{1}{s_1 + s_3}\left(-W_{13}^{ij}W_{13}^{kl} + W_{13}^{ij}W_{31}^{kl} - W_{31}^{ij}W_{31}^{kl} + W_{31}^{ij}W_{13}^{kl}\right) \\ &\quad + \frac{1}{s_2 + s_3}\left(-W_{23}^{ij}W_{23}^{kl} + W_{23}^{ij}W_{32}^{kl} - W_{32}^{ij}W_{32}^{kl} + W_{32}^{ij}W_{23}^{kl}\right) \\ &\quad = \operatorname{vec}\left(W^{ij}\right)^T \mathcal{A} \operatorname{vec}\left(W^{kl^T}\right)\end{aligned} \tag{29}$$

We substitute into equation 26 to get an expression for the trace of the second derivative of the singular values. After rearranging, we get the final form of β:

$$\begin{aligned}\beta = &- \gamma(b_j \otimes e_i)^T (I \otimes UV^T)(b_l \otimes e_k) \\ &+ \gamma(b_j \otimes e_i)^T (Z^T V \otimes U)\mathcal{A}(U^T X \otimes V^T)(b_l \otimes e_k)\end{aligned} \tag{30}$$

By varying the indices i, j, k, l we construct a block of all the second derivatives of the kernel, resulting in the desired form for $\bar{K}_{22}$ This completes the expression for the second derivative of the kernel with respect to X_{ij} and Z_{kl}:

$$\frac{\partial^2 K(X,Z)}{\partial \operatorname{vec}(X) \partial \operatorname{vec}(Z)^T} = K(X,Z)\bar{K}_{22} \tag{31}$$

□

Low rank structure

One potential issue with the methodology proposed here is that the size of the kernel matrix, especially with derivatives, can grow very quickly: for m molecules with n atoms each, the full K matrix will be $m(3n+1) \times m(3n+1)$. Even though the goal of our approach is to use relatively few molecules for training, this can quickly grow infeasible for large n: even multiplying a vector by the full K matrix will naively have cost $O(m^2n^2)$.

However, there is also substantial structure in the full K matrix that lets us reduce this time substantially. Namely, each 22 block of the collection of derivatives is the sum of a rank-one matrix, and two matrices

$$\bar{B}(I_n \otimes UV^T)\bar{B}^T = B \otimes UV^T \tag{32}$$

and

$$\begin{aligned}&\bar{B}(Z^T V \otimes U)\mathcal{A}(U^T X \otimes V^T)\bar{B}^T \\ &\quad = (BZ^T V \otimes U)\mathcal{A}(BU^T X \otimes V^T)\end{aligned} \tag{33}$$

Multiplying a vector by this product just involves centering and multiplication by 3x3 matrices, which together are $O(n)$ operators. Thus, the cost of a matrix-vector product can easily be brought down to $O(m^2n)$. Combined with conjugate gradient approaches, these methods can be used to bring the cost of the SVD kernel with derivatives to approximately the same level as the standard SVD kernel.

Application to Molecular Energy Prediction

We now come to the main applied focus of this paper: using the above kernel approach to model the free energy of molecules. As described above, the basic task is, given a configuration of atoms in 3D space (the 3D coordinate of each atom along with it's type), we want to predict the molecular energy of the atom. This is a task with numerous applications in energy, physics, and biology, and such methods are widely studied in chemical engineering (see, e.g. (Sholl and Steckel 2011)). Existing methods, such as those based upon density functional theory (DFT), are capable of computing these quantities to reasonably high accuracy, but the methods are computationally intensive (they amount to solving a large partial differential equation to compute the energy). The goal of our overall approach, similar to the work of (Rupp et al. 2012), is to use a machine learning approach to "replicate" the results of an expensive simulation procedure with a much faster surrogate function. Unlike this past work, however, we explicitly incorporate energy derivatives (forces), which are automatically generated by DFT methods when computing the energy. We also focus upon the task of *optimizing* the molecular configuration to find the configuration with minimum energy; this is a canonical task

Kernel	RMSE
Mean predictor	2.4154
Eigenspectrum representation	0.4284
SVD kernel	0.7182
SVD kernel with derivatives	0.6239

Table 1: Prediction error on water

Kernel	RMSE
Mean predictor	0.7418
Eigenspectrum representation	0.7369
SVD kernel	0.7609
SVD kernel with derivatives	0.2276

Table 2: Prediction error on glycerol

in molecular modeling, and we show Bayesian optimization methods, based upon our approach, can substantially outperform existing state-of-the-art solvers used in the chemical engineering community, such as LBFGS.

To begin, we will evaluate the performance of our approach just on predicting the energy of previously unseen configurations of a molecule. In this section and the next, all computations were carried out using the GPAW numerical code (Mortensen, Hansen, and Jacobsen 2005), a grid-based implementation of the DFT calculator. We also use the Python Atomic Simulation Environment (ASE) (Bahn and Jacobsen 2002) to set up the computations and later to perform the molecular optimization. As mentioned above obtaining the potential energies is extremely computationally expensive using these tools, so the algorithm must be able to generalize from a comparatively small subset of the molecular space. To get accurate energy results, the time to perform a DFT calculation can take anywhere from a few hours for a simple molecule to days for complex compounds.

We applied a Gaussian process using the SVD kernel to model the energies of water and glycerol molecules. For both molecules, we generated 100 data points by adding noise to the original coordinates. Denote $\mathcal{X}$ as the set of all our inputs (3D positions of the atoms) and y the vector of corresponding energies. We form a prediction on new examples $\mathcal{X}'$ by the standard Gaussian process equations

$$\begin{aligned} \mu(\mathcal{X}') &= k(\mathcal{X}', \mathcal{X})(K(\mathcal{X}, \mathcal{X}) + \lambda I)^{-1} y \\ \sigma(\mathcal{X}') &= K(\mathcal{X}', \mathcal{X}') - \\ &\quad K(\mathcal{X}', \mathcal{X})(K(\mathcal{X}, \mathcal{X}) + \lambda I)^{-1} K(\mathcal{X}, \mathcal{X}') \end{aligned} \tag{34}$$

The kernel hyperparameters, namely exponential parameter γ and regularization parameter λ, were chosen by a grid-search with an inner 4-fold cross validation and optimized for the root mean squared error. We introduce an additional regularization parameter λ_{deriv} to account for the difference in magnitude between the energies and derivatives.

In the case of the simple molecule water, the SVD kernel is able to capture much of the structure of the potential energy function, with the derivative information giving a small improvement in performance. Since water only has 3 atoms, the SVD kernel is already able to model the molecule quite well without the derivatives. Furthermore, the eigenvalue method from (Rupp et al. 2012) performs marginally better in this application, likely because it exploits additional information about the energy of single atoms (such information could be included in our setting as well, though we do not pursue this approach here).

In contrast, on the larger glycerol molecule with 14 atoms, none of the methods except the SVD kernel with derivatives are able to accurately model the system, and all other methods perform virtually identical to simple mean predictions. This is also born out in the learning curves in Figure 1. The SVD kernel with derivatives is able to achieve low error using approximately 40 simulations, whereas the SVD-only approach (and similar approaches), never achieve good performance (the fluctuations in the learning performance for the SVD kernel in Figure 1 are a product of random folds, and are all higher than simply predicting the mean energy in the data set).

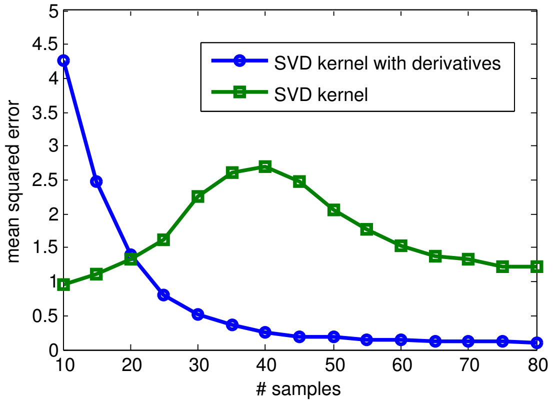

Figure 1: Learning curves for the SVD kernel on glycerol with and without derivatives. The presence of kernel derivatives results in faster convergence to a lower error.

Bayesian optimization

Finally, the ultimate goal of a fast DFT approximation is not just to predict the energy, but to find the minimum energy configuration. Since the target function has a high cost, we take a Bayesian optimization approach to minimize the potential energy of a molecule. We use a Gaussian process model with the SVD derivative kernel as an approximation of the target energy function, and we use the lower confidence bound (LCB) as our acquisition function to sample new points:

$$\text{LCB}(\mathcal{X}) = \mu(\mathcal{X}) - \kappa\sigma(\mathcal{X}) \tag{35}$$

where μ and σ are the mean and variance from the Gaussian process. We optimized this surrogate function itself using LBFGS, though importantly, this optimization procedure is very fast, as we can compute the GP predictions very quickly.

For the testing environment, we optimize three molecules whose atoms have had noise added to their initial coordinates. We treat the DFT energy calculation as the function

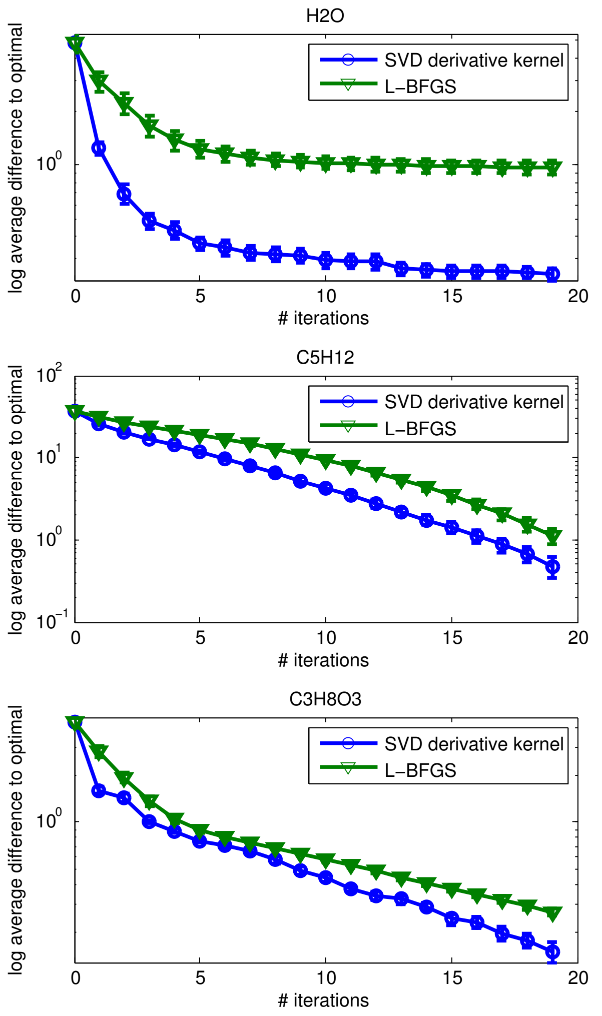

Figure 2: The average minimum found over all 40 runs per iteration for various molecules, which corresponds to the number of function samples. Results for the SVD kernel converge faster and closer than LBFGS to the optimal value.

to be minimized, and provide the negative forces as the gradient function. Each algorithm begins with only the initial starting set of coordinates, and the optimization is repeated for 40 randomly perturbed starting configurations.

We compare our optimization method against the LBFGS quasi-Newton method (directly performing LBFGS using the derivatives provided by the DFT). We use an implementation made specifically for optimizing molecular structures from the Python ASE package, which is a state-of-the-art method for performing molecular structure optimization. Because the time used by the DFT solver far exceeds any of the computation time used either by LBFGS or our Bayesian optimization, we report results in terms of the number of (expensive) simulation iterations required.

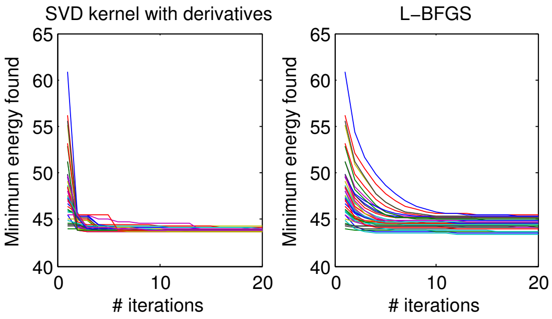

Figure 3: A plot of all 40 individual optimizations for each algorithm on water. The SVD kernel converges rapidly in almost all cases regardless of starting position. The LBFGS method depends on the starting configuration and has greater variance in convergence rate and final energy value.

We see that the SVD kernel with derivatives within the Bayesian optimization framework offers a significant improvement over the LBFGS method. On average, the SVD kernel with derivatives uses fewer iterations to get closer to the optimal value, converging more rapidly than LBFGS in Figure 2 across multiple molecules. Another benefit that the SVD method has is its ability to converge quickly to the same optimal value regardless of the starting position. For example, after 5 iterations, nearly all optimizations of water using the SVD method are close to the optimal value as seen in Figure 3. In contrast, the LBFGS method depends significantly on the starting configuration. Some LBFGS optimizations converge slower than others and reach a variety of final energy values.

Conclusion

When using physical systems or geometric data, there are certain properties of translational and rotational invariance and an availability of derivative information that we can exploit to improve the performance of learning in this space. In the case of molecular energies, predicting accurate energies is especially important since calculating the exact energy of a compound typically takes days, and scales badly as the complexity of the molecule increases.

In this paper, we have developed a kernel based on the singular value decomposition that maintains the translational and rotational invariants expected in geometric data. We incorporated derivative data into the kernel and constructed a simplified form to compute the larger covariance matrices. Our results show that the SVD kernel has good predictive ability, and can outperform other methods. Practically, our work shows that the SVD kernel can optimize molecules faster and more reliably than current methods.

References

Bahn, S. R., and Jacobsen, K. W. 2002. An object-oriented scripting interface to a legacy electronic structure code. *Computing in Science & Engineering* 4(3):56–66.

Bartók, A. P.; Payne, M. C.; Kondor, R.; and Csányi, G. 2010. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Physical review letters* 104(13):136403.

Behler, J., and Parrinello, M. 2007. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters* 98(14):146401.

Brochu, E.; Cora, V. M.; and De Freitas, N. 2010. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

Cox, D. D., and John, S. 1997. Sdo: A statistical method for global optimization. *Multidisciplinary design optimization: state of the art* 315–329.

Horn, B. K.; Hilden, H. M.; and Negahdaripour, S. 1988. Closed-form solution of absolute orientation using orthonormal matrices. *JOSA A* 5(7):1127–1135.

Lu, C.-P.; Hager, G. D.; and Mjolsness, E. 2000. Fast and globally convergent pose estimation from video images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22(6):610–622.

Montavon, G.; Hansen, K.; Fazli, S.; Rupp, M.; Biegler, F.; Ziehe, A.; Tkatchenko, A.; Lilienfeld, A. V.; and Müller, K.-R. 2012. Learning invariant representations of molecules for atomization energy prediction. In *Advances in Neural Information Processing Systems*, 440–448.

Montavon, G.; Rupp, M.; Gobre, V.; Vazquez-Mayagoitia, A.; Hansen, K.; Tkatchenko, A.; Müller, K.-R.; and von Lilienfeld, O. A. 2013. Machine learning of molecular electronic properties in chemical compound space. *New Journal of Physics* 15(9):095003.

Mortensen, J. J.; Hansen, L. B.; and Jacobsen, K. W. 2005. Real-space grid implementation of the projector augmented wave method. *Physical Review B* 71(3):035109.

Papadopoulo, T., and Lourakis, M. I. 2000. Estimating the jacobian of the singular value decomposition: Theory and applications. In *Computer Vision-ECCV 2000*. Springer. 554–570.

Pozdnoukhov, A. 2009. *Machine learning for spatial environmental data: theory, applications, and software*. EPFL press.

Rupp, M.; Tkatchenko, A.; Müller, K.-R.; and von Lilienfeld, O. A. 2012. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical review letters* 108(5):058301.

Sholl, D., and Steckel, J. A. 2011. *Density functional theory: a practical introduction*. John Wiley & Sons.

Snoek, J.; Larochelle, H.; and Adams, R. P. 2012. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, 2951–2959.

Solak, E.; Murray-Smith, R.; Leithead, W. E.; Leith, D. J.; and Rasmussen, C. E. 2003. Derivative observations in gaussian process models of dynamic systems. In *Neural Information Processing Systems*. MIT Press.