Gaussian Approximation Potentials: A Brief Tutorial Introduction

Albert P. Bartók* and Gábor Csányi

We present a swift walk-through of our recent work that uses machine learning to fit interatomic potentials based on quantum mechanical data. We describe our Gaussian approximation potentials (GAP) framework, discuss a variety of descriptors, how to train the model on total energies and derivatives, and the simultaneous use of multiple models of different complexity. We also show a small example using QUIP, the software sandbox implementation of GAP that is available for noncommercial use. © 2015 Wiley Periodicals, Inc.

DOI: 10.1002/qua.24927

Introduction

Molecular scale simulation is a mature field with a wide range of electronic structure methods that approximate the solution of the Schrödinger equation in a systematic fashion. For larger scale computations, empirical interatomic potentials are used, which are nowadays fit to data generated by electronic structure models. Together these play a significant role in understanding processes on the microscopic level, complementing experiment, and theory. Computer simulations are regularly used to interpret experimental results and to predict properties of materials.

The power of atomistic simulations would be enormously enhanced if the interatomic potentials used to simulate materials were not limited by their simple empirical functional forms but accurately approached the Born–Oppenheimer potential energy surface, similarly to the case of small molecules for which quantum chemists have been fitting accurate potential energy surfaces for decades. The challenge in the materials field is that rather than fitting the total energy of a fixed number of atoms, the task is to find a unique local functional that describes the energy of a single atom or bond given its neighbor environment. This local energy function must naturally allow for bond forming and bond breaking, that is, the change in the number and identity of the atoms comprising the neighbor environment.

A number of groups—many of them contributing to the present volume—have started research programmes to address this problem using advances in the synthetic understanding that recently emerged in statistics and machine learning.[1–6] These fast-growing fields are concerned with classification, regression and probability density estimation on large and noisy datasets, and also with finding suitable variable transformations that allow increased performance in these tasks. There are a number of closely related computational frameworks that are widely used, including artificial neural networks, stochastic processes (e.g., Gaussian processes), and regularized nonparametric optimization. In this tutorial introduction, we focus on a particular exposition that allows a succinct presentation of the formalism and how it can be brought to bear on the problem of fitting potential energy surfaces for materials based on data computed by electronic structure methods. For detailed derivations of the necessary fundamental results, we refer the reader to the machine learning and statistics literature.[7,8]

Methodology

The hallmark of an interatomic potential is that the total energy, E, of a set of atoms is written as a sum of range-separated terms,

$$E = \sum_{\alpha} \sum_{i \in \alpha} \varepsilon_i^{\alpha} + \text{long range contributions} \quad (1)$$

where ε_i^{α} are local energy functionals with compact support within a radius r_{cut}, and by "long range contributions" we mean electrostatics including polarizability, van der Waals interactions, and so forth. This is an uncontrolled approximation, as there is nothing about the Schrödinger equation that tells us *a priori* that its solutions can be written in this form: the level of accuracy and its applicability in any particular situation has to be tested by numerical experiments. The index α denotes the type of contribution: the arguments of a local energy term may be any suitable descriptors, for example, atom-pair distances, bond angles, or indeed the complete atomic environment, and the index i counts the instances of these terms in a particular configuration, for example, all bonds for a pair term, all angles for a three-body angle-dependent term, or all atoms for an atom-centered term. We can think of descriptors as functions that transform the Cartesian coordinates of the atoms in the neighborhood of a given atom.

A. P. Bartók, G. Csányi
Engineering Laboratory, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, United Kingdom
E-mail: apbartok@gmail.com
Contract grant sponsor: Leverhulme Early Career Fellowship (to A.P.B.).
Contract grant sponsor: Isaac Newton Trust (to A.P.B.).

Table 1. Definition of central concepts used in fitting accurate potentials for materials.

Neighbourhood	Set of nearby atoms whose positions constitute the input to the local energy function evaluated for a given atom.
Descriptors	Transformation of the positions of atoms in the neighborhood, obeying the desired symmetries of the energy function. Also, called features.
Kernel	Similarity measure between two neighborhoods, equivalent to the covariance of the corresponding two local energy values.

In this article, we will only discuss the local energy contribution, although it is clear that for many materials in which atoms acquire significant partial charges or have easily polarizable electrons it must be complemented by electrostatic and dispersion interactions. These long range terms can either remain completely empirical, but may also include parameters that are fitted to data using approaches similar to what are used for the local term.

Gaussian process regression

We first consider the case of a single type of local energy functional. Using a set of arbitrary basis functions $\{\phi_h\}_{h=1}^{H}$ that take as their arguments any descriptor $\mathbf{d}_i$ of the neighbor environment of atom i, we write the atomic energy ε_i as

$$\varepsilon_i = \varepsilon(\mathbf{d}_i, \mathbf{w}) = \sum_h w_h \phi_h(\mathbf{d}_i), \tag{2}$$

where $\mathbf{w}$ is a vector of weights w_h corresponding to the basis functions, to be determined by the fit. If the prior probability distribution of the weights is chosen to be Gaussian with zero mean, that is, $P(\mathbf{w}) = \text{Normal}(\mathbf{w}; \mathbf{0}, \sigma_w \mathbf{I})$, the covariance of two atomic energies is

$$\begin{aligned}\langle \varepsilon_i \varepsilon_j \rangle &= \left\langle \sum_{hh'} w_h w_{h'} \phi_h(\mathbf{d}_i) \phi_{h'}(\mathbf{d}_j) \right\rangle = \sum_{hh'} \langle w_h w_{h'} \rangle \phi_h(\mathbf{d}_i) \phi_{h'}(\mathbf{d}_j) \\ &= \sigma_w^2 \sum_h \phi_h(\mathbf{d}_i) \phi_h(\mathbf{d}_j)\end{aligned} \tag{3}$$

where we exploited that $\langle w_h w_{h'} \rangle = \delta_{hh'} \sigma_w^2$. The inner product of the basis functions in the last expression defines the kernel or covariance function

$$C(\mathbf{d}_i, \mathbf{d}_j) \equiv \sum_h \phi_h(\mathbf{d}_i) \phi_h(\mathbf{d}_j). \tag{4}$$

Kernel functions in this application are to be understood as similarity measures between two atomic neighbor environments. Every basis set induces a corresponding kernel, and as seen below, only the kernel is required for regression, we never need to construct a basis set in the space of descriptors explicitly. General requirements on kernel functions are in the literature.[7,9]

Our goal is to predict the energy of an arbitrary atomic configuration, based on a dataset of previous calculations. For any set of microscopic observations $\mathbf{t}$—which could be the local atomic energies or the total energies of all atoms in a set of configurations—the covariance matrix is defined as $\mathbf{C} \equiv \langle \mathbf{t}\mathbf{t}^\top \rangle$, and its elements can be computed using the previously defined covariance function. The prior probability of observing $\mathbf{t}$ is also Gaussian,

$$P(\mathbf{t}) = \text{Normal}(\mathbf{t}; \mathbf{0}, \mathbf{C}) \propto \exp\left(-\frac{1}{2} \mathbf{t}^\top \mathbf{C}^{-1} \mathbf{t} \right). \tag{5}$$

The predicted value y of a new test configuration, given previous observations $\mathbf{t}$, has the probability distribution

$$P(y \mid \mathbf{t}) = \frac{P(\mathbf{t}, y)}{P(\mathbf{t})} \tag{6}$$

which is also Gaussian. We take the mean of this distribution as the prediction, which can be expressed[7] as

$$\bar{y} = \mathbf{k}^\top \mathbf{C}^{-1} \mathbf{t} \tag{7}$$

where $\mathbf{k}$ is the covariance vector of function values: $\mathbf{k} \equiv \langle y\, \mathbf{t} \rangle$. This shows the real power of the Gaussian process approach: the original basis functions we started with and their corresponding unknown weights are never required explicitly, the predictions only depend on the kernel function C and the previous observations $\mathbf{t}$.

It can be shown[10] that a two-layer neural network with infinite number of hidden nodes and hyperbolic tangent switching function is equivalent to a Gaussian process with

$$C(\mathbf{d}_i, \mathbf{d}_j) \propto V - |\mathbf{d}_i - \mathbf{d}_j|^2. \tag{8}$$

Extra layers in neural networks with more than two layers can be regarded as performing a nonlinear transformation on the input coordinates, before the output layers carry out the regression task.

Yet another equivalent approach for fitting functions is kernel ridge regression, where the unknown function is expanded as a linear combination of radial basis functions,*

$$f(\mathbf{d}) = \sum_i \alpha_i C(\mathbf{d}, \mathbf{d}_i), \tag{9}$$

and the weights α are optimized by minimizing the cost function

$$L = \sum_i (t_i - f(\mathbf{d}_i))^2 + \lambda ||\alpha||^2. \tag{10}$$

If we define the norm as

$$||\alpha||^2 = \alpha^\top \mathbf{C}\, \alpha, \tag{11}$$

the predictions of kernel regression are also equivalent to those of the Gaussian process. The kernel here has the dual

*We note that kernel ridge regression is not limited to radial basis functions, any positive definite kernel may be used.[9]

role of defining both the basis functions and the norm of the weights in the loss function.

Total energies. Atomic energies are unavailable in quantum mechanical calculations, which only provide the total energy and its derivatives. From these, we have to predict the local energies. It is straightforward to modify Eq. (3) to express the covariance of the total energies of two set of atoms, *N* and *M*,

$$\begin{aligned}\langle E_N E_M\rangle &= \left\langle \sum_{i\in N}\varepsilon(\mathbf{d}_i)\sum_{j\in M}\varepsilon(\mathbf{d}_j)\right\rangle = \left\langle \sum_{i\in N}\sum_{j\in M}\sum_{hh'} w_h w_{h'}\phi_h(\mathbf{d}_i)\phi_{h'}(\mathbf{d}_j)\right\rangle \\ &= \sum_{i\in N}\sum_{j\in M}\sum_{hh'}\langle w_h w_{h'}\rangle\phi_h(\mathbf{d}_i)\phi_{h'}(\mathbf{d}_j) = \sigma_w^2\sum_{i\in N}\sum_{j\in M}\sum_{h}\phi_h(\mathbf{d}_i)\phi_h(\mathbf{d}_j) \\ &= \sigma_w^2\sum_{i\in N}\sum_{j\in M}C(\mathbf{d}_i,\mathbf{d}_j)\end{aligned} \tag{12}$$

Derivatives. The total quantum mechanical energy of a configuration depends on the relative positions of the atoms and, in case of condensed systems, also the lattice parameters. Denoting a general coordinate by ξ, the partial derivative of the total energy is related to the force as

$$f_{k\alpha} = -\frac{\partial E}{\partial r_{k\alpha}} = -\frac{\partial E}{\partial \xi} \text{ if } \xi \equiv r_{k\alpha} \tag{13}$$

or to the viral stress as

$$v_{\alpha\beta} = \frac{\partial E}{\partial h_{\alpha\beta}} = \frac{\partial E}{\partial \xi} \text{ if } \xi \equiv h_{\alpha\beta} \tag{14}$$

where $r_{k\alpha}$ is the αth component of the Cartesian coordinates of atom k and $h_{\alpha\beta}$ is an element of the deformation matrix **H** of the lattice vectors. Differentiating Eq. (12) with respect to an arbitrary coordinate ξ_k of configuration *N* results in

$$\left\langle \frac{\partial E_N}{\partial \xi_k} E_M\right\rangle = \frac{\partial\langle E_N E_M\rangle}{\partial \xi_k} = \sigma_w^2\sum_{i\in N}\sum_{j\in M}\nabla_{\mathbf{d}_i}C(\mathbf{d}_i,\mathbf{d}_j)\cdot\frac{\partial \mathbf{d}_i}{\partial \xi_k}. \tag{15}$$

If ξ_k is the *x*, *y*, or *z* component of the position of atom *k*, $\frac{\partial \mathbf{d}_i}{\partial \xi_k}$ becomes exactly zero if the pair distance $|\mathbf{r}_i - \mathbf{r}_k|$ is beyond the cutoff of the environment, so the first sum need not be done over all atoms in the configuration. Similarly, the covariance of two derivative quantities may be written as

$$\left\langle \frac{\partial E_N}{\partial \xi_k}\frac{\partial E_M}{\partial \chi_l}\right\rangle = \frac{\partial^2\langle E_N E_M\rangle}{\partial \xi_k \partial \chi_l} = \sigma_w^2\sum_{i\in N}\sum_{j\in M}\frac{\partial \mathbf{d}_i^\top}{\partial \xi_k}(\nabla_{\mathbf{d}_i}C(\mathbf{d}_i,\mathbf{d}_j)\nabla_{\mathbf{d}_j}^\top)\frac{\partial \mathbf{d}_j}{\partial \chi_l}, \tag{16}$$

where the elements of the Jacobian are

$$(\nabla_{\mathbf{d}_i}C(\mathbf{d}_i,\mathbf{d}_j)\nabla_{\mathbf{d}_j}^\top)_{\alpha\beta} = \frac{\partial^2 C(\mathbf{d}_i,\mathbf{d}_j)}{\partial d_{i\alpha}\partial d_{j\beta}} \tag{17}$$

The local energy ε is still predicted using Eq. (7), but the elements of **y** are total energies or derivative quantities, and the elements of the covariance matrix **C** are, therefore, computed by Eqs. (12), (15), or (16). The elements of **k** are the covariance between the local energy that we wish to predict and the data that we have available, $\langle \varepsilon E\rangle$ or $\langle \varepsilon\, \partial E/\partial \xi\rangle$ as appropriate.

Multiple models. Interactions in some atomistic systems might be partitioned using a many-body type expansion—indeed, many traditional interatomic potentials are based on a few low-order contributions, such as two- and three-body energies.[11]. We now describe how such models can be fitted using Gaussian process regression. For example, truncating the local part of Eq. (1) at three-body contributions, the total energy is approximated as

$$E = \sum_{p\,\in\,\text{pairs}}\varepsilon_p^{(2)} + \sum_{t\,\in\,\text{triplets}}\varepsilon_t^{(3)} \tag{18}$$

where $\varepsilon^{(2)}$ and $\varepsilon^{(3)}$ are general two- and three-body energy functions, respectively, and pairs and triplets in this context may refer to atoms as well as entire molecules. Two independent Gaussian processes are used,

$$\varepsilon^{(2)}(\cdot\cdot) = \sum_h w_h^{(2)}\phi_h^{(2)}(\cdot\cdot) \tag{19}$$

$$\varepsilon^{(3)}(\therefore) = \sum_h w_h^{(3)}\phi_h^{(3)}(\therefore) \tag{20}$$

where $\cdot\cdot$ and $\therefore$ denotes generic geometric descriptors of pairs and triplets (in case of molecules, the descriptors need to describe the whole dimer and trimer configuration). The prior distributions of the two weight vectors are independent Gaussians, so the covariance of the total energy of two configurations *N* and *M* may be written as

$$\langle E_N E_M\rangle = \sigma_{w^{(2)}}^2\sum_{p\in\text{pairs}_N}\sum_{q\in\text{pairs}_M}C^{(2)}(p,q) + \sigma_{w^{(3)}}^2\sum_{t\in\text{triplets}_N}\sum_{u\in\text{triplets}_M}C^{(3)}(t,u), \tag{21}$$

where we applied the same kernel trick as above and exploited that $\langle w_h^{(2)} w_{h'}^{(3)}\rangle = 0$ for any h and h'. As the two-body terms can, in principle, be included in the three-body terms, splitting them appropriately might require setting the variances of the two terms carefully. For example, if 80% of the total interaction energy is expected to be due to pair interactions, this information can be built into the prior using the ratio $\sigma_{w^{(2)}} : \sigma_{w^{(3)}} = 4 : 1$.

Compact support. The local energy terms need to have compact support to be computationally efficient, and this is typically achieved using an explicit spatial cutoff function. In machine learning models for materials, the cutoff may be built into descriptors,[12] so only neighbors within a predefined radial distance of the central atom are considered. Alternatively, cutoffs may be implemented in the kernels. Consider the pair energy model

$$\varepsilon^{(2)}(\cdot\cdot) = f_{\text{cut}}(\cdot\cdot)\sum_h w_h^{(2)}\phi_h^{(2)}(\cdot\cdot) \tag{22}$$

where f_{cut} is defined such that it goes smoothly to zero as a function of the geometric attributes of the pair (e.g., as the

distance between them approaches a limit, in case of a pair of atoms). The resulting covariance function is

$$\langle \varepsilon^{(2)}(p)\varepsilon^{(2)}(q)\rangle = \sigma^2_{w^{(2)}} C^{(2)}(p,q) f_{\text{cut}}(p) f_{\text{cut}}(q). \tag{23}$$

In our implementation, we use

$$f_{\text{cut}}(r) = \begin{cases} 1 & \text{for} \quad r \le r_{\text{cut}} - d \\ \left[\cos\left(\pi \dfrac{r - r_{\text{cut}} + d}{d}\right) + 1\right]/2 & \text{for} \quad r_{\text{cut}} - d < r \le r_{\text{cut}} \\ 0 & \text{for} \quad r > r_{\text{cut}} \end{cases} \tag{24}$$

as the cutoff function, where d is a parameter that determines the width of the cutoff region. There is some freedom in choosing a numerical value for d, but two criteria has to be considered: the covariance should change smoothly when two atoms become connected, and minimizing the spurious effect of the cutoff transition on the derivatives. We typically use $d = 1$ Å, as this is regarded the length scale of atomic interactions.

Data noise. A configuration's total quantum mechanical energy and its derivatives can be extrapolated to exact numerical values, provided the various convergence parameters of the applied quantum mechanical method are used appropriately. However, we should still regard these as noisy observations when trying to fit a model, for the following reasons:

i. the separation into a sum of local contributions is an approximation,
ii. our model is additive over various contributions with unknown individual ratios,
iii. our model employs a finite cutoff,
iv. the quantum mechanical calculations may not be fully converged.†

Thus, we modify our model in Eq. (1) to include a Gaussian noise v_E, $P(v_E) = \text{Normal}(v_E; 0, \sigma_E)$

$$E = \sum_\alpha \sum_i \varepsilon_i^\alpha + v_E \tag{25}$$

and similarly, derivative quantities are modeled as

$$\frac{\partial E}{\partial \xi_k} = \frac{\partial \sum_\alpha \sum_i \varepsilon_i^\alpha}{\partial \xi_k} + v_\xi \tag{26}$$

where $P(v_\xi) = \text{Normal}(v_\xi; 0, \sigma_\xi)$. As a consequence, Eq. (12) is modified to give the covariance of observed total energies

$$\langle E_N E_M \rangle = \sigma_w^2 \sum_{i\in N} \sum_{j \in M} C(\mathbf{d}_i, \mathbf{d}_j) + \sigma_E^2 \delta_{NM}, \tag{27}$$

and the covariance of observed derivatives becomes

$$\frac{\partial^2 \langle E_N E_M \rangle}{\partial \xi_k \partial \chi_l} = \sigma_w^2 \sum_{i\in N} \sum_{j\in M} \frac{\partial \mathbf{d}_i^\top}{\partial \xi_k} (\nabla_{\mathbf{d}_i} C(\mathbf{d}_i, \mathbf{d}_j) \nabla_{\mathbf{d}_j}^\top) \frac{\partial \mathbf{d}_j}{\partial \chi_l} + \sigma_\xi^2 \delta_{NM} \delta_{\xi_k \chi_l} \tag{28}$$

Sparsification. It is easy to see that computing covariance matrices and vectors can become quite expensive, especially if derivative quantities are also included. This, combined with the assumption that atomic neighborhood environments are often repetitious, leads to the idea that sparse Gaussian processes might be applied. Sparsity is a central concept in machine learning, and sparse Gaussian processes are described in detail by, for example, Quinonero-Candela and Rasmussen,[13] or Snelson and Ghahramani.[14] In our adaptation of sparsification, we use representative atomic neighborhood environments, or pairs and triplets, etc. The model is built using all observations in the dataset and it can be regarded as a projection onto a subset of data points, the sparse representation.

Let us consider a set of configurations, each of which contains an arbitrary number of atoms and the corresponding set of total energies, derivatives or both. The observables are collected in the vector $\mathbf{t}$. We select a set of environments, the sparse set S, and compute the covariance matrices:

$$(\mathbf{C}_{SS})_{ss'} = \langle \varepsilon_s \varepsilon_{s'} \rangle, \text{where } s, s' \in S, \tag{29}$$

$$(\mathbf{C}_{ST})_{st} = \langle \varepsilon_s E_t \rangle, \tag{30}$$

where $s \in S$ and t is an index of total energies in $\mathbf{t}$, and

$$(\mathbf{C}_{ST})_{s\tau} = \left\langle \varepsilon_s \frac{\partial E_t}{\partial \xi_k} \right\rangle, \tag{31}$$

where $s \in S$ and τ denotes derivative observables in $\mathbf{t}$. The predicted value at an arbitrary atomic neighborhood environment $\mathbf{d}_*$ can be calculated from

$$\varepsilon_*(\mathbf{d}_*) = \mathbf{k}_*^\top (\mathbf{C}_{SS} + \mathbf{C}_{ST} \Lambda_{TT}^{-1} \mathbf{C}_{TS})^{-1} \mathbf{C}_{ST} \Lambda_{TT}^{-1} \mathbf{t}, \tag{32}$$

where $(\mathbf{k}_*)_s = \langle \varepsilon_* \varepsilon_s \rangle$, and Λ_{TT} is a diagonal matrix, where each diagonal element is σ_E^2 or σ_ξ^2, depending on the type of observable. As configurations may contain different numbers of atoms, we scale σ_E^2 accordingly. Note that the part multiplying $\mathbf{k}_*$ from the right is precomputed at the training stage, so only $\mathbf{k}_*$ needs to be computed for each prediction, and this scales linearly with the number of sparse points (and not with the total number of original data points!). Derivatives of ε_* are readily available analytically, using the appropriate covariance functions. In practice, we found that the sparse covariance matrix $\mathbf{C}_{SS}$ should be regularized by adding a small positive constant σ_{jitter} to the diagonal values. The numerical value of the constant should be as small as possible, without compromising the positive definiteness of $\mathbf{C}_{SS}$. Normally σ_{jitter} is 6–9 orders of magnitude less than the diagonal elements.

†This could be an advantage, as we do not need fully converged quantum mechanical data.

Descriptors

The success of applying machine learning techniques to fit potential energy surfaces depends to a large extent on representing the atomic environments appropriately. Transformations of atomic positions to which the local energy is invariant, that is, rotation and inversion of an environment about its center, and permutation of identical atoms should be explicitly built in. We presented a detailed study on representing chemical environments elsewhere[12] in which we focussed on atom-centeed neighborhood environments. Here, we describe a few other types of descriptors.

Pairs and triplets. Pairs of atoms are simply described by the distance between them, but in case of triplets the distances need to be symmetrized. If atoms j and k form a triplet with i as the central atom, a possible descriptor can be the vector

$$[r_{ik}+r_{ij}, (r_{ik}-r_{ij})^2, r_{jk}]. \tag{33}$$

As we mentioned earlier, the covariance function must be augmented by a cutoff function. We use $f_{\mathrm{cut}}(r_{ij})$ for the pair terms, and in case of triplets we use $f_{\mathrm{cut}}(r_{ij})f_{\mathrm{cut}}(r_{ik})$.

Water dimers. It is clear that our approach to symmetrize distances in case of three-body descriptors will be overly complicated if we attempt to apply it on more than a couple of atoms. For example, the potential energy surface of water can be modeled very accurately using a many-body expansion of interactions between water molecules.[15,16] The two-body term in the expansion necessitates a descriptor for the water-water dimer, for which we used the pairwise distances between the constituent atoms, 15 in total. However, this descriptor in this form is not invariant to permuting atoms of the same element. If exchange of hydrogen atoms between different molecules is not permitted, the following permutations $\hat{P}$ that operate on the order of atoms must be taken in account:

i. swaps of water molecules in the dimer (2)
ii. exchange of hydrogen atoms within each molecule (2×2),

so 8 in total. Instead of modifying the descriptor, we enforced permutation symmetry at the level of the kernel function. If we take an arbitrary kernel function, $C(\mathbf{d}, \mathbf{d}')$, that takes vector arguments, we can generate a permutational invariant kernel as

$$C'(\mathbf{d}, \mathbf{d}') = \sum_{\hat{P}} C(\mathbf{d}, \hat{P}\mathbf{d}'), \tag{34}$$

which must be normalized[9]:

$$C''(\mathbf{d}, \mathbf{d}') = \frac{C'(\mathbf{d}, \mathbf{d}')}{\sqrt{C'(\mathbf{d}, \mathbf{d})}\sqrt{C'(\mathbf{d}', \mathbf{d}')}}. \tag{35}$$

We used the squared exponential as our starting kernel in the case of water molecules.

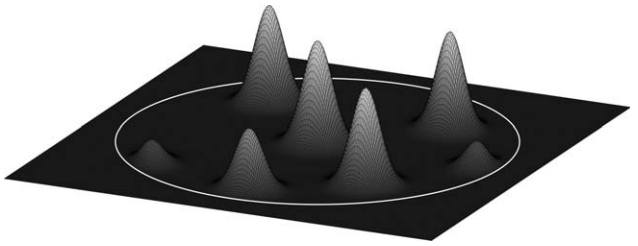

Figure 1. A two-dimensional illustration of the atomic neighbor density function used in SOAP. The white circle represents the radial cutoff distance.

SOAP. We note that our previously introduced[12] kernel based on "Smooth Overlap of Atomic Positions" may be interpreted from the function-space view we used throughout this manuscript. We represent the atomic neighborhood of atom i by the neighborhood density function (for illustration, see Fig. 1)

$$\rho_i(\mathbf{r}) \equiv \sum_j^{\text{neigh.}} \exp\left(-\frac{|\mathbf{r}-\mathbf{r}_{ij}|^2}{2\sigma_{\text{atom}}^2}\right), \tag{36}$$

and ε_i, the atomic energy of atom i can then be regarded as a functional of ρ_i

$$\varepsilon_i = \varepsilon[\rho_i] = \int w(\mathbf{r})\rho_i(\mathbf{r})d\mathbf{r} \tag{37}$$

where the prior distribution of the weights is Gaussian, so

$$\langle w(\mathbf{r})w(\mathbf{r}')\rangle = \delta(\mathbf{r}-\mathbf{r}')\sigma_w^2, \tag{38}$$

resulting in the covariance of two atomic energies

$$C(\rho_i, \rho_j) = \langle \varepsilon_i \varepsilon_j \rangle = \left\langle \int w(\mathbf{r})\rho_i(\mathbf{r})w(\mathbf{r}')\rho_j(\mathbf{r}')\, \mathrm{d}\mathbf{r}\, \mathrm{d}\mathbf{r}' \right\rangle = \sigma_w^2 \int \rho_i(\mathbf{r})\rho_j(\mathbf{r})\, d\mathbf{r}. \tag{39}$$

It is useful to note here that if C is a valid kernel, then $|C|^p$ is also valid.[9] This covariance function $|C|^p$ is not invariant to rotations, but we may convert it in a similar fashion to what we did in the case of the water dimer:

$$C'(\rho_i, \rho_j) = \int |C(\rho_i, \hat{R}\rho_j)|^p \mathrm{d}\hat{R}, \tag{40}$$

which must then be normalized, so the final result is

$$C''(\rho_i, \rho_j) = \frac{C'(\rho_i, \rho_j)}{\sqrt{C'(\rho_i, \rho_i)}\sqrt{C'(\rho_j, \rho_j)}}. \tag{41}$$

In practice, we evaluate the SOAP kernel numerically by first expanding Eq. (36) in a basis set

$$\rho_i(\mathbf{r}) = \sum_{nlm} c_{nlm}^{(i)} g_n(r) Y_{lm}(\hat{\mathbf{r}}), \tag{42}$$

where $c_{nlm}^{(i)}$ are the expansion coefficients corresponding to atom i, $\{g_n(r)\}$ is an arbitrary set of orthonormal radial basis

Table 2. Overview of the basic command line arguments of teach_sparse.

at_file	Contains the database configurations, in concatenated XYZ files
descriptor_str	Parameters of descriptor(s)
default_sigma	The assumed standard deviation of the errors, $\{\sigma_{\text{energy}}\ \sigma_{\text{force}}\ \sigma_{\text{viral}}\}$
sparse_jitter	Regularization of the sparse covariance matrix, σ_{jitter}
e0	Baseline of atomic energies

functions, and $Y_{lm}(\hat{\mathbf{r}})$ are the spherical harmonics. We form descriptors from the coefficients by computing the power spectrum elements

$$p^{(i)}_{nn'l} \equiv \frac{1}{\sqrt{2l+1}} \sum_m c^{(i)}_{nlm} (c^{(i)}_{n'lm})^*, \tag{43}$$

and the rotationally invariant covariance of atoms i and j is given by

$$C'(\rho_i, \rho_j) = \sum_{n,n',l} p^{(i)}_{nn'l} p^{(j)}_{nn'l}, \tag{44}$$

which we normalize according to Eq. (41). The normalization step is equivalent to normalizing the vector elements $p^{(i)}_{nn'l}$, so C'' is, in fact, a dot-product kernel of vectors $\mathbf{p}^{(i)}/|\mathbf{p}^{(i)}|$ and $\mathbf{p}^{(j)}/|\mathbf{p}^{(j)}|$. Note that it is often useful to raise C'' to a power $\zeta > 1$, in order to sharpen the difference between atomic environments. To see the details of the above results, we refer the reader to our earlier work.[12,17]

Software

The ideas presented in Methodology section are implemented in the QUIP package, which can be downloaded from the git repository at https://github.com/libAtoms/QUIP. Code related to Gaussian approximation potentials (GAP) prediction can be obtained under a non-commercial licence from http://www.libatoms.org/gap/gap_download.html. Users who wish to use the training code should contact the corresponding author.

QUIP is a molecular simulation sandbox written in object-oriented FORTRAN95/2003, with interfaces to python (compatible with ASE), and various other simulation packages, such as LAMMPS, CP2K, CASTEP, and others. QUIP is essentially a collection of objects and interfaces that contain and manipulate atomic configurations and interatomic potentials.

The GAP implementation provides over 20 different descriptors, which can be used with two types of covariance functions, the squared exponential

$$C(\mathbf{d}_i, \mathbf{d}_j) = \delta^2 \exp\left(-\frac{1}{2} \sum_\alpha \left(\frac{d_{i\alpha} - d_{j\alpha}}{\theta_\alpha} \right)^2 \right) \tag{45}$$

and the polynomial kernel

$$C(\mathbf{d}_i, \mathbf{d}_j) = \delta^2 (\mathbf{d}_i \cdot \mathbf{d}_j + \sigma_0^2)^\zeta. \tag{46}$$

To demonstrate the training, we provide a simple example, where we train GAP to reproduce the well-known Stillinger–Weber (SW) potential[11] for silicon. We used a database of Si_n silicon clusters, $n = 7 \ldots 13$, sampled from a 2000 K molecular dynamics simulation. We used 600 configurations in total, with total energies and forces calculated using the SW potential. We used a combination of two- and three-body interactions, with a cutoff of 4.1 Å. We used the command line

```
teach_sparse at_file=data_Si_SW.xyz descriptor_str={ \
distance_2b cutoff=4.1 n_sparseX=250 covariance_type=ard_
se theta_fac=0.5: \
angle_3b cutoff=4.1 n_sparseX=500 covariance_type=ard_
se theta_fac=0.5} \
default_sigma={0.001 0.05 0.01} sparse_jitter=1.0e-8 e0 = 0.0
```

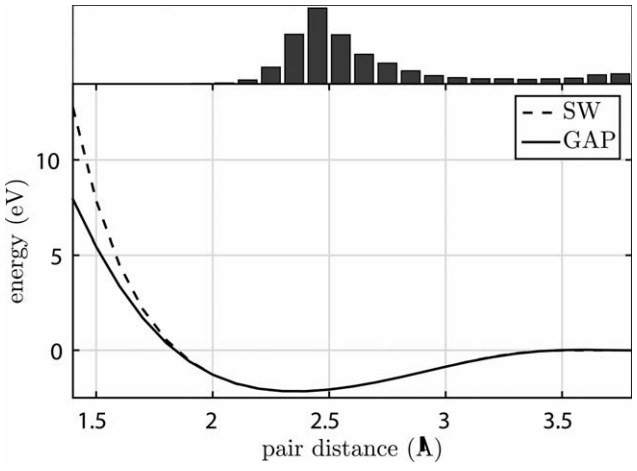

Figure 2. The two-body term in the SW potential for silicon. The dashed line represents the analytical answer and the solid line shows the GAP fit. The histogram above corresponds to the input data to the fit.

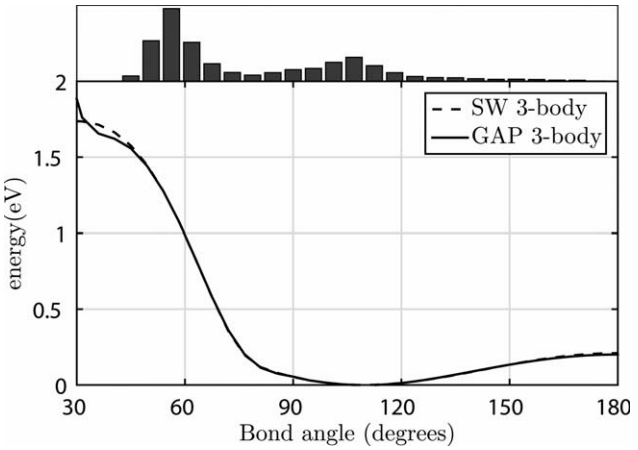

Figure 3. The three-body term in the SW potential for silicon. We fixed the two neighbors at 2.5 and 2.8 Å and varied the bond angle and plotted the sum of all three interactions between the three atoms. The dashed line represents the analytical answer and the solid line shows the GAP fit. The histogram above represents the input data to the fit.

Table 2 summarizes the command line arguments used in this example. The command line argument descriptor_str contains the parameters of the descriptor, which depend on the type. We define the number of sparse points n_sparseX and the type of covariance function. theta_fac is the simplest way to control the θ_α length-scale parameters in the squared exponential covariance function: the range of the descriptor values in each dimension α is scaled by the constant theta_fac. More descriptors can be concatenated, separated by the: symbol, resulting in the fitting of a model that is the sum terms each based on one descriptor.

It is possible to specify an existing QUIP potential as a baseline, so energies/forces/virials are subtracted from the target values before fitting. These are added back automatically when the potential is called. Naturally, the baseline can be another GAP, resulting in hierarchical models with arbitrary level of recursion.

In the above example, it is possible to check whether the GAP model was able to recover both terms of the target model, as they are available analytically. We emphasize that the fitting uses total energies and forces only, so the machine learning algorithm has to infer the separate two- and three-body terms from this convoluted data. To show the quality of the fit, we plot the pair potential in Figure 2 and the angle term in Figure 3 for both the original SW and the fitted model. The agreement is rather good in both cases, except at the edges of the range where there was no input data.

The potential file generated by teach_sparse can be used to compute the total energies and similar quantities of arbitrary configurations. For example,

```
eval at_file=data_Si_SW.xyz param_file=gp.xml init_args={IP GAP} e f
```

will compute the total energies (e) of the configurations stored in data_Si_SW.xyz as well as the atomic forces (f) using the fitted GAP model from the file gp.xml. Another use of eval is to compute and print the descriptor vectors for any descriptor type implemented in QUIP. For example,

```
eval at_file=data_Si_SW.xyz descriptor_str={angle_3b cutoff=4.1}
```

prints all three-body descriptors, in this case the descriptors defined in Eq. (33).

Acknowledgment

The authors would like to thank our referees for their comments during the revision process.

Keywords: interatomic potentials · machine learning · Gaussian process · *ab initio* · atomic environments

How to cite this article: Bartók A. P., Csányi G. *J. Quantum Chem.* **2014**, *115*, 1051–1057. DOI: 10.1002/qua.24927

[1] J. Behler, M. Parrinello, *Phys. Rev. Lett.* **2007**, *98*, 146401.
[2] J. Behler, *Phys. Chem. Chem. Phys.* **2011**, *13*, 17930.
[3] M. Rupp, A. Tkatchenko, K.-R. Müller, O. A. von Lilienfeld, *Phys. Rev. Lett.* **2012**, *108*, 058301.
[4] A. Brown, B. Braams, K. Christoffel, Z. Jin, J. Bowman, *J. Chem. Phys.* **2003**, *119*, 8790.
[5] K. Hansen, G. Montavon, F. Biegler, S. Fazli, M. Rupp, M. Scheffler, O. A. von Lilienfeld, A. Tkatchenko, K.-R. Müller, *J. Chem. Theory Comput.* **2013**, *9*, 3404.
[6] C. M. Handley, P. L. A. Popelier, *J. Phys. Chem. A* **2010**, *114*, 3371.
[7] D. Mackay, *Information Theory, Inference, and Learning Algorithms*; Cambridge University Press: Cambridge, UK, **2003**.
[8] B. Scholkopf, A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*; MIT Press: Cambridge, MA, **2001**, ISBN 0262194759.
[9] C. E. Rasmussen, C. K. I. Williams, *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, **2006**.
[10] R. M. Neal, *Bayesian Learning for Neural Networks*; Ph.D. Thesis, University of Toronto: Toronto, Canada, **1995**.
[11] F. H. Stillinger, T. A. Weber, *Phys. Rev. B* **1985**, *31*, 5262.
[12] A. P. Bartók, R. Kondor, G. Csányi, *Phys. Rev. B* **2013**, *87*, 184115.
[13] J. Q. Quinonero-Candela, C. E. Rasmussen, *J. Mach. Learn. Res.* **2005**, *6*, 1939.
[14] E. Snelson, Z. Ghahramani, In *Advances in Neural Information Processing Systems 18*; Y. Weiss, B. Schölkopf, J. Platt, Eds.; MIT Press: Cambridge, MA, USA, **2006**; p. 1257.
[15] M. J. Gillan, D. Alfè, A. P. Bartók, G. Csányi, *J. Chem. Phys.* **2013**, *139*, 244504.
[16] G. R. Medders, V. Babin, F. Paesani, *J. Chem. Theory Comput.* **2014**, *10*, 2906.
[17] W. J. Szlachta, A. P. Bartók, G. Csányi, *Phys. Rev. B* **2014**, *90*, 104108.

Received: 4 February 2015
Revised: 12 March 2015
Accepted: 20 March 2015
Published online 27 April 2015