Machine learning for materials science
fitting interatomic potentials

Gábor Csányi
Engineering Laboratory

UNIVERSITY OF CAMBRIDGE
Outline

- Function fitting in many dimensions: Gaussian process regression
- Descriptors, the SOAP kernel
- Examples: water, tungsten, iron, carbon, organic molecules
Machine learning: function fitting
Machine learning: function fitting

• Complicated function in high dimensional space
  - Given {domain,range} example pairs: regression
  - Given just {domain} examples: (probability) density estimation
Machine learning: function fitting

• Complicated function in high dimensional space
  - Given \{\text{domain,range}\} example pairs: regression
  - Given just \{\text{domain}\} examples: (probability) density estimation

• Function can be really unknown
Machine learning: function fitting

- Complicated function in high dimensional space
  - Given \{domain, range\} example pairs: regression
  - Given just \{domain\} examples: (probability) density estimation

- Function can be really unknown

images \rightarrow face ?

material composition \rightarrow superconductor ?
Machine learning: function fitting

• Complicated function in high dimensional space
  - Given \{domain, range\} example pairs: regression
  - Given just \{domain\} examples: (probability) density estimation

• Function can be really unknown

  images $\rightarrow$ face ?

  material composition $\rightarrow$ superconductor ?

• Or just very expensive ("surrogate modelling")
Machine learning: function fitting

- Complicated function in high dimensional space
  - Given \{domain, range\} example pairs: regression
  - Given just \{domain\} examples: (probability) density estimation

- Function can be really unknown

- Or just very expensive ("surrogate modelling")

\[
\begin{array}{ccc}
\text{images} & \rightarrow & \text{face} \\
\text{material composition} & \rightarrow & \text{superconductor} \\
\text{atomic coordinates} & \rightarrow & \text{DFT total energy}
\end{array}
\]
Proliferation of ideas for functional forms

- Pair potentials: Lennard-Jones, RDF-derived, etc.
- Three-body terms: Stillinger-Weber, MEAM, etc.
- Embedded Atom (no angular dependence)
- Bond Order Potential (BOP)
  - Tight-binding-derived density of states,
  - attractive term with pair-potential repulsion
- ReaxFF: kitchen-sink with hundreds of parameters

\[
\varepsilon_i = \frac{1}{2} \sum_j V_2(|r_{ij}|) + \sum_{jk} k (\theta_{ijk} - \theta_0)^2
\]
\[
\varepsilon_i = \Phi \left( \sum_j \rho(|r_{ij}|) \right)
\]
Proliferation of ideas for functional forms

- Pair potentials: Lennard-Jones, RDF-derived, etc.
- Three-body terms: Stillinger-Weber, MEAM, etc.
- Embedded Atom (no angular dependence)
- Bond Order Potential (BOP)
  - Tight-binding-derived density of states, attractive term with pair-potential repulsion
- ReaxFF: kitchen-sink with hundreds of parameters

\[ \varepsilon_i = \frac{1}{2} \sum_j V_2(|r_{ij}|) + \sum_{jk} k(\theta_{ijk} - \theta_0)^2 \]

\[ \varepsilon_i = \Phi \left( \sum_j \rho(|r_{ij}|) \right) \]
Proliferation of ideas for functional forms

- Pair potentials: Lennard-Jones, RDF-derived, etc.
- Three-body terms: Stillinger-Weber, MEAM, etc.
- Embedded Atom (no angular dependence)
- Bond Order Potential (BOP)
  - Tight-binding-derived density of states, attractive term with pair-potential repulsion
- ReaxFF: kitchen-sink with hundreds of parameters

These are not the correct functions.
Limited accuracy, not systematic
Proliferation of ideas for functional forms

- Pair potentials: Lennard-Jones, RDF-derived, etc.
- Three-body terms: Stillinger-Weber, MEAM, etc.
- Embedded Atom (no angular dependence)
- Bond Order Potential (BOP)
  Tight-binding-derived density of states, attractive term with pair-potential repulsion
- ReaxFF: kitchen-sink with hundreds of parameters

These are not the correct functions. Limited accuracy, not systematic

\[ \varepsilon_i = \frac{1}{2} \sum_j V_2(|r_{ij}|) + \sum_{jk} k(\theta_{ijk} - \theta_0)^2 \]

\[ \varepsilon_i = \Phi \left( \sum_j \rho(|r_{ij}|) \right) \]

Representation is implicit

GOAL: potentials based on quantum mechanics
Interatomic potentials for molecular dynamics

Transferability
biomolecular force fields
(biochemistry)

AMBER
CHARMM
AMOEBA
GROMACS
OPLS
...
Interatomic potentials for molecular dynamics

- Transferability
- Biomolecular force fields (biochemistry)

- Reactive
- Solid state materials (physics & materials)

- AMBER
- CHARMM
- AMOEBA
- GROMACS
- OPLS
- Tersoff
- Brenner
- EAM
- BOP
- ...
Interatomic potentials for molecular dynamics

Transferability
biomolecular
force fields
(biochemistry)

AMBER
CHARMM
AMOEBA
GROMACS
OPLS
...

Accuracy
small molecules
in gas phase
(quantum chemistry)

Bowman
Szalewicz
Paesani
...

Reactive
solid state
materials
(physics & materials)

Tersoff
Brenner
EAM
BOP
...
Interatomic potentials for molecular dynamics

Transferability
biomolecular force fields
(biochemistry)

Accuracy
small molecules
in gas phase
(quantum chemistry)

Reactive
solid state materials
(physics & materials)

Bowman
Szalewicz
Paesani
...

Tersoff
Brenner
EAM
BOP
...

AMBER
CHARMM
AMOEBA
GROMACS
OPLS
...

...
Focus on short range interactions

\[ E_{\text{tot}} = \sum_{\text{atoms } i} \varepsilon(r_1 - r_i, r_2 - r_i, \ldots) + \left[ \text{Long range terms} \right] \]

( + [ "best" existing potential ] )

Finite range atomic/molecular energy function
Focus on short range interactions

\[ E_{\text{tot}} = \sum_{\text{atoms } i} \varepsilon(r_1 - r_i, r_2 - r_i, \ldots) + [ \text{ Long range terms } ] + \left( + [ \text{"best" existing potential } ] \right) \]

Finite range atomic/molecular energy function

Goals:

- Faithful reproduction of Born-Oppenheimer PES
- Systematic convergence to BO-PES – cost tradeoff
What makes a potential

Ingredients → Desirable properties
What makes a potential

Ingredients

• Representation of atomic neighbourhood

• Interpolation of functions

• Database of configurations

Desirable properties
What makes a potential

Ingredients

• **Representation** of atomic neighbourhood

• **Interpolation** of functions

• **Database** of configurations

Desirable properties

smoothness, faithfulness, continuity
What makes a potential

**Ingredients**

- **Representation of atomic neighbourhood**
- **Interpolation of functions**
- **Database of configurations**

**Desirable properties**

- smoothness,
- faithfulness,
- continuity
- flexible but smooth functional form, few sensible parameters
What makes a potential

<table>
<thead>
<tr>
<th>Ingredients</th>
<th>Desirable properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Representation of atomic neighbourhood</td>
<td>smoothness, faithfulness, continuity</td>
</tr>
<tr>
<td>• Interpolation of functions</td>
<td>flexible but smooth functional form, few sensible parameters</td>
</tr>
<tr>
<td>• Database of configurations</td>
<td>predictive power non-domain specific</td>
</tr>
</tbody>
</table>
What makes a potential

Ingredients

- Representation of atomic neighbourhood
- Interpolation of functions
- Database of configurations

Desirable properties

- smoothness, faithfulness, continuity
- flexible but smooth functional form, few sensible parameters
- predictive power non-domain specific
Fitting functions with parameters

\[ E = \sum_{j < j'} V_2(\alpha, \beta; d_{jj'}) \]
Fitting functions with parameters

\[ E = \sum_{j < j'} V_2(\alpha, \beta; d_{jj'}) \]

2-body potential

Fit to database of configurations

\[ E^n = \sum_{j < j'} V_2(\alpha, \beta; d_{jj'}^n) \]
Fitting functions with parameters

\[ E = \sum_{j<j'} V_2(\alpha, \beta; d_{jj'}) \]

2-body potential

parameters

Fit to database of configurations

\[ E^n = \sum_{j<j'} V_2(\alpha, \beta; d_{jj'}^n) \]

\[ \alpha^*, \beta^* = \arg \min_{\alpha, \beta} \sum_n \left[ E^n - \sum_{j<j'} V_2(\alpha, \beta; d_{jj'}^n) \right]^2 + \lambda \| (\alpha, \beta) \| \]
Fitting functions with parameters

\[ E = \sum_{j < j'} V_2(\alpha, \beta; d_{jj'}) \]

2-body potential

\[ \alpha^*, \beta^* = \arg \min_{\alpha, \beta} \sum_n \left[ E^n - \sum_{j < j'} V_2(\alpha, \beta; d_{jj'}^n) \right]^2 + \lambda \| (\alpha, \beta) \| \]

- \( V_2 \) nonlinear, but physically intuitive
- low dimensional space, easy to get enough data
- Good transferability ("extrapolation")

Fit to database of configurations

\[ E^n = \sum_{j < j'} V_2(\alpha, \beta; d_{jj'}^n) \]
Function fitting with kernels

Fit a function $f(x)$ based on observations $\mathbf{y} \equiv \{y_i\}$ at $\{x_i\}$

$$f(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x)$$

e.g. $k(x, x') = \sigma_w^2 e^{-|x-x'|^2 / 2\sigma^2}$
Function fitting with kernels

Fit a function \( f(x) \) based on observations \( y \equiv \{y_i\} \) at \( \{x_i\} \)

\[
f(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x)
\]

\[
y_j = \sum_{i=1}^{N} \alpha_i k(x_i, x_j)
\]

e.g. \( k(x, x') = \sigma_w^2 e^{-|x-x'|^2/2\sigma^2} \)
Function fitting with kernels

Fit a function $f(x)$ based on observations $y \equiv \{y_i\}$ at $\{x_i\}$

$$f(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x)$$

e.g. $k(x, x') = \sigma_w^2 e^{-|x-x'|^2 / 2\sigma^2}$

$$y_j = \sum_{i=1}^{N} \alpha_i \left( k(x_i, x_j) + \sigma_v^2 \delta_{ij} \right)$$

regularised fit:

choose $\sigma, \sigma_w, \sigma_v$
Function fitting with kernels

Fit a function $f(x)$ based on observations $\mathbf{y} \equiv \{y_i\}$ at $\{x_i\}$

$$f(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x)$$

e.g. $k(x, x') = \sigma_w^2 e^{-|x-x'|^2 / 2\sigma^2}$

regularised fit:

choose $\sigma, \sigma_w, \sigma_v$

$$y_j = \sum_{i=1}^{N} \alpha_i (k(x_i, x_j) + \sigma^2 \delta_{ij})$$

$$\mathbf{y} = (\mathbf{K} + \sigma^2 \mathbf{I}) \mathbf{\alpha}$$

$$\mathbf{\alpha} = \mathbf{C}^{-1} \mathbf{y}$$

$$[\mathbf{K}]_{ij} \equiv k(x_i, x_j)$$

$$\mathbf{C} \equiv \mathbf{K} + \sigma^2 \mathbf{I}$$

$$k \equiv k(x_i, x)$$
Function fitting with kernels

Fit a function $f(x)$ based on observations $\mathbf{y} \equiv \{y_i\}$ at $\{x_i\}$

$$
f(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x)
$$

e.g. $k(x, x') = \sigma_w^2 e^{-|x-x'|^2/2\sigma^2}$

regularised fit:
choose $\sigma, \sigma_w, \sigma_v$

$$
y_j = \sum_{i=1}^{N} \alpha_i (k(x_i, x_j) + \sigma_v^2 \delta_{ij})
$$

$$
\mathbf{y} = (\mathbf{K} + \sigma_v^2 \mathbf{I}) \mathbf{\alpha}
$$

$$
\mathbf{\alpha} = \mathbf{C}^{-1} \mathbf{y}
$$

$$
f(x) = \mathbf{k}^T \mathbf{C}^{-1} \mathbf{y}
$$
Function fitting with kernels

Fit a function $f(x)$ based on observations $\mathbf{y} \equiv \{y_i\}$ at $\{x_i\}$

\[
f(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x)
\]

\[
y_j = \sum_{i=1}^{N} \alpha_i \left( k(x_i, x_j) + \sigma_{\nu}^2 \delta_{ij} \right)
\]

\[
\mathbf{y} = (\mathbf{K} + \sigma_{\nu}^2 \mathbf{I}) \mathbf{\alpha}
\]

\[
\mathbf{\alpha} = \mathbf{C}^{-1} \mathbf{y}
\]

regularised fit:
choose $\sigma, \sigma_w, \sigma_{\nu}$

\[
[K]_{ij} \equiv k(x_i, x_j)
\]

\[
\mathbf{C} \equiv \mathbf{K} + \sigma_{\nu}^2 \mathbf{I}
\]

\[
k \equiv k(x_i, x)
\]

\[
f(x) = \mathbf{k}^T \mathbf{C}^{-1} \mathbf{y}
\]
Function fitting with kernels

Fit a function $f(x)$ based on observations $y \equiv \{y_i\}$ at $\{x_i\}$

$$f(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x)$$

e.g. $k(x, x') = \sigma_w^2 e^{-|x-x'|^2 / 2\sigma^2}$

$$y_j = \sum_{i=1}^{N} \alpha_i (k(x_i, x_j) + \sigma_w^2 \delta_{ij})$$

regularised fit: choose $\sigma, \sigma_w, \sigma_v$

$$y = (K + \sigma_w^2 I)\alpha$$

$$\alpha = C^{-1}y$$

$[K]_{ij} \equiv k(x_i, x_j)$

$C \equiv K + \sigma_w^2 I$

$k \equiv k(x_i, x)$

- Non-linearity is in $K$, acts as basis for a linear fit (“kernel trick”)
- High-dimensional space (can scale with data!) - flexibility
- Hard to get enough data to “fill” space - no extrapolation, poor transferability

$$f(x) = k^T C^{-1}y$$
1D example

Gaussians basis functions are wide!

\[ f(x) = k^T C^{-1} y \]
Intuitively, not every $y(x)$ is equally likely:

$$P(y) \propto e^{-\frac{1}{2} y^T C^{-1} y}$$

Function values: $(x_n, y_n)$

$$C_{ij} \propto e^{-\frac{(x_i - x_j)^2}{2\sigma^2}}$$

$$y^T C^{-1} y = [y_1 \ y_2] \begin{bmatrix} 1 & 0.01 \\ 0.01 & 1 \end{bmatrix}^{-1} \approx [1 & -0.01 \\ -0.01 & 1]$$

$$y_1 - \alpha y_2$$
Gaussian Process Regression summary

• Covariance:

\[ K(x_i, x_j) = \exp(- (x_i - x_j)^2 / 2\sigma^2) \]  

\[ f(x) = \arg \max_P P(f|\text{data}) = \sum_i \alpha_i K(x, x(i)) \]

Maximum of posterior

\[ \alpha = C^{-1}y \equiv (\sigma_w^2K + \sigma_u^2I)^{-1}y \]

\[ C_{ii'} = \sigma_w^2K(x_i, x_{i'}) + \sigma_u^2\delta_{ii'} \]

• Meaningful hyper-parameters:
  
  \( \sigma \): smoothness (x-scale) of \( f \)
  
  \( \sigma_w \): y-scale of \( f \)
  
  \( \sigma_u \): variance (noise) of input data

• This is \textit{not an optimised fit}, but a closed form estimate!
Datapoints: \( N = 2^p \)
Datapoints: $N = 2^p$
Datapoints: $N = 2^p$
Datapoints: $N = 2^p$
Datapoints: \( N = 2^p \)
Datapoints: $N = 2^p$
Datapoints: $N = 2^p$
Datapoints: $N = 2^p$
Datapoints: $N = 2^p$
Datapoints: \( N = 2^p \)
Datapoints: \( N = 2^p \)

No PBC

PBC

basis function

basis function
Datapoints: \( N = 2^p \)
Datapoints: $N = 2^p$
Datapoints: \( N = 2^p \)
Machine learning framework: kernel regression

\[ \varepsilon(q^{(i)}) = \sum_{k}^{N} \alpha_k K(q^{(i)}, q^{(k)}) \]

- Linear regression:
  \[ K_{\text{DP}}(q^{(i)}, q^{(k)}) = q^{(i)} \cdot q^{(k)} \rightarrow \varepsilon(q^{(i)}) = \sum_{j} q_j^{(i)} \sum_{k}^{N} \alpha_k q_k^{(k)} = q^{(i)} \cdot \beta \]

- Neural networks with one hidden layer
  \[ K_{\text{NN}}(q^{(i)}, q^{(k)}) = -|q^{(i)} - q^{(k)}|^2 + \text{const.} \]

- Gaussian kernel
  \[ K_{\text{SE}}(q^{(i)}, q^{(k)}) = \exp\left( -\sum_{j} \frac{(q^{(i)} - q^{(k)})^2}{2\sigma_j^2} \right) \]
Machine learning framework: kernel regression

- Linear regression:
  \[ K_{DP}(q^{(i)}, q^{(k)}) = q^{(i)} \cdot q^{(k)} \quad \Rightarrow \quad \varepsilon(q^{(i)}) = \sum_j q_j^{(i)} \sum_k \alpha_k q_k^{(k)} = q^{(i)} \cdot \beta \]

- Neural networks with one hidden layer
  \[ K_{NN}(q^{(i)}, q^{(k)}) = -|q^{(i)} - q^{(k)}|^2 + \text{const.} \]

- Gaussian kernel
  \[ K_{SE}(q^{(i)}, q^{(k)}) = \exp\left(-\sum_j \frac{(q^{(i)} - q^{(k)})^2}{2\sigma_j^2}\right) \]
Regression vs Deep Neural Networks

\[ \Phi(\Sigma) \]

Nonlinear adaptation of basis

Linear Algebra
Noise control
Few parameters

Smaller basis
Hard optimisation problem
Many empirical parameters
Example: $\text{H}_2\text{O}$ dimer

- Standard representation:
  6 atoms $\Rightarrow$ 15 interatomic distances: $x = \{r_{ij}\}$

- Symmetrize Gaussian kernel function:
  \[
  \tilde{K}(x, x') = \sum_{p \in S} K(p(x), x')
  \]
  $S$: symmetry group of molecules
Example: $\text{H}_2\text{O}$ dimer

- Standard representation:
  
  6 atoms $\Rightarrow$ 15 interatomic distances: $x = \{r_{ij}\}$

- Symmetrize Gaussian kernel function:

  $$\tilde{K}(x, x') = \sum_{p \in S} K(p(x), x')$$

  $S$: symmetry group of molecules
Example: $\text{H}_2\text{O}$ dimer

- Standard representation:
  6 atoms $\Rightarrow$ 15 interatomic distances : $x = \{r_{ij}\}$

- Symmetrize Gaussian kernel function:
  $$\tilde{K}(x, x') = \sum_{p \in S} K(p(x), x')$$
  
  $S$: symmetry group of molecules
Building a potential for materials: a representation problem

- Weyl matrix

\[
\begin{bmatrix}
    \mathbf{r}_1 \cdot \mathbf{r}_1 & \mathbf{r}_1 \cdot \mathbf{r}_2 & \cdots & \mathbf{r}_1 \cdot \mathbf{r}_N \\
    \mathbf{r}_2 \cdot \mathbf{r}_1 & \mathbf{r}_2 \cdot \mathbf{r}_2 & \cdots & \mathbf{r}_2 \cdot \mathbf{r}_N \\
    \vdots & \vdots & \ddots & \vdots \\
    \mathbf{r}_N \cdot \mathbf{r}_1 & \mathbf{r}_N \cdot \mathbf{r}_2 & \cdots & \mathbf{r}_N \cdot \mathbf{r}_N
\end{bmatrix}
\]
Building a potential for materials: a representation problem

- Weyl matrix

\[
\begin{bmatrix}
\mathbf{r}_1 \cdot \mathbf{r}_1 & \mathbf{r}_1 \cdot \mathbf{r}_2 & \cdots & \mathbf{r}_1 \cdot \mathbf{r}_N \\
\mathbf{r}_2 \cdot \mathbf{r}_1 & \mathbf{r}_2 \cdot \mathbf{r}_2 & \cdots & \mathbf{r}_2 \cdot \mathbf{r}_N \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{r}_N \cdot \mathbf{r}_1 & \mathbf{r}_N \cdot \mathbf{r}_2 & \cdots & \mathbf{r}_N \cdot \mathbf{r}_N
\end{bmatrix}
\]

- Problems:
  - not permutation invariant
  - fixed number of neighbours (no reactions!)
Drop atom ordering?

- Permutation: too costly for ~ 20 neighbours
- Sorting distances: not smooth
- Unordered list of distances? Distance histograms?
Drop atom ordering?

- Permutation: too costly for ~20 neighbours
- Sorting distances: not smooth
- Unordered list of distances? Distance histograms?

Not unique!
SOAP: Smooth Overlap of Atomic Positions

\[ \rho_i(r) = \sum_j \exp \left( -\frac{|r - r_{ij}|^2}{2\sigma^2} \right) = \sum_{nlm} c_{nlm}^{(i)} g_n(r) Y_{lm}(\hat{r}) \]
SOAP: Smooth Overlap of Atomic Positions

\[ \rho_i(r) = \sum_j \exp \left( -\frac{|r - r_{ij}|^2}{2\sigma^2} \right) = \sum_{nlm} c_{nlm}^{(i)} g_n(r) Y_{lm}(\hat{r}) \]

• Overlap integral

\[ S(\rho_i, \rho_i') = \int \rho_i(r) \rho_i'(r) dr, \]
SOAP: Smooth Overlap of Atomic Positions

\[ \rho_i(r) = \sum_j \exp \left( -|r - r_{ij}|^2 / 2\sigma^2 \right) = \sum_{nlm} c_{nlm}^{(i)} g_n(r) Y_{lm}(\hat{r}) \]

- Overlap integral
  \[ S(\rho_i, \rho_{i'}) = \int \rho_i(r) \rho_{i'}(r) dr \]

- Integrate over all 3D rotations:
  \[ k(\rho_i, \rho_{i'}) = \int \left| S(\rho_i, \hat{R}\rho_{i'}) \right|^2 d\hat{R} = \int d\hat{R} \left| \int \rho_i(r) \rho_{i'}(\hat{R}r) dr \right|^2 \]
SOAP: Smooth Overlap of Atomic Positions

\[ \rho_i(r) = \sum_j \exp \left( -|r - r_{ij}|^2 / 2\sigma^2 \right) = \sum_{nlm} c_{nlm}^{(i)} g_n(r) Y_{lm}(\hat{r}) \]

- Overlap integral
  \[ S(\rho_i, \rho_i') = \int \rho_i(r) \rho_i'(r) dr, \]

- Integrate over all 3D rotations:
  \[ k(\rho_i, \rho_i') = \int \left| S(\rho_i, \hat{R}\rho_i') \right|^2 d\hat{R} = \int d\hat{R} \left| \int \rho_i(r) \rho_i'(\hat{R}r) dr \right|^2 \]

- After LOTS of algebra: SOAP kernel
  \[ k(\rho_i, \rho_i') = \sum_{nn', l} p_{nn'}^{(i)} p_{nn'}^{(i')} = p^{(i)} \cdot p^{(i')} \]
  \[ p_{nn'l} = c_{nl}^\dagger \cdot c_{n'l} \propto \text{Steinhardt } Q_l \]
SOAP: Smooth Overlap of Atomic Positions

\[ \rho_i(r) = \sum_j \exp \left( -|r - r_{ij}|^2 / 2\sigma^2 \right) = \sum_{nlm} c_{nlm}^{(i)} g_n(r) Y_{lm}(\hat{r}) \]

- Overlap integral

\[ S(\rho_i, \rho_{i'}) = \int \rho_i(r) \rho_{i'}(r) dr, \]

- Integrate over all 3D rotations:

\[ k(\rho_i, \rho_{i'}) = \int \left| S(\rho_i, \hat{R}\rho_{i'}) \right|^2 d\hat{R} = \int d\hat{R} \int \rho_i(r) \rho_{i'}(\hat{R}r) dr \]

- After LOTS of algebra: SOAP kernel

\[ k(\rho_i, \rho_{i'}) = \sum_{n,n',l} p_{nn'l}^{(i)} p_{nn'l}^{(i')} = p^{(i)} \cdot p^{(i')} \]

\[ p_{nn'l} = c_{nl}^\dagger \cdot c_{n'l} \]

\[ K_{ij} \propto |k(\rho_i, \rho_j)|^\xi \]

\[ \propto \text{Steinhardt } Q_l \]
SOAP: Smooth Overlap of Atomic Positions

\[\rho_i(r) = \sum_j \exp \left(-\frac{|r - r_{ij}|^2}{2\sigma^2}\right) = \sum_{nlm} c_{nlm}^{(i)} g_n(r) Y_{lm}(\hat{r})\]

- Overlap integral

\[S(\rho_i, \rho_{i'}) = \int \rho_i(r) \rho_{i'}(r) dr,\]

- Integrate over all 3D rotations:

\[k(\rho_i, \rho_{i'}) = \int \left| S(\rho_i, \hat{R} \rho_{i'}) \right|^2 d\hat{R} = \int d\hat{R} \left| \int \rho_i(r) \rho_{i'}(\hat{R}r) dr \right|^2\]

- After LOTS of algebra: SOAP kernel

\[k(\rho_i, \rho_{i'}) = \sum_{n,n',l} p_{nn'l}^{(i)} p_{nn'l}^{(i')} = \mathbf{p}^{(i)} \cdot \mathbf{p}^{(i')}\]

\[p_{nn'l} = c_{nl}^\dagger \cdot c_{n'l}\]

\[K_{ij} \propto |k(\rho_i, \rho_j)|^\xi\]

\[\propto \text{Steinhardt } Q_l\]
Learning from total energies and derivatives

• Linear model allows learning from linear functions of the target, e.g. sums and derivatives

\[ \varepsilon(x_1) + \varepsilon(x_2) = \sum_{i=1}^{N} \alpha_i [K(x_1, x_i) + K(x_2, x_i)] \]

\[ \frac{\partial \varepsilon(x)}{\partial x} = \sum_{i=1}^{N} \alpha_i \frac{\partial K(x, x_i)}{\partial x} \]

• Total energies are sums, atomic forces are sums of partial derivatives

• Linear problem becomes rectangular
Using multiple kernels simultaneously

• Total energy as a body expansion converges fast

\[ E = \sum_{i,j} V_2(r_{ij}) + \sum_{i,j,k} V_3(r_{ij}, r_{ik}, r_{jk}) + \ldots \]

• Functions in lower dimensional spaces are easier to fit
Using multiple kernels simultaneously

• Total energy as a body expansion converges fast
  \[ E = \sum_{i,j}^{\text{pairs}} V_2(r_{ij}) + \sum_{i,j,k}^{\text{triplets}} V_3(r_{ij}, r_{ik}, r_{jk}) + \ldots \]

• Functions in lower dimensional spaces are easier to fit

• Gaussian process easily generalises to fitting sums
Using multiple kernels simultaneously

- Total energy as a body expansion converges fast
  \[
  E = \sum_{i,j}^{\text{pairs}} V_2(r_{ij}) + \sum_{i,j,k}^{\text{triplets}} V_3(r_{ij}, r_{ik}, r_{jk}) + \ldots
  \]

- Functions in lower dimensional spaces are easier to fit

- Gaussian process easily generalises to fitting sums

\[
E = \left(\delta^{(2b)}\right)^2 \sum_{i \in \text{pairs}} \varepsilon^{(2b)}(q_i^{(2b)}) + \left(\delta^{(3b)}\right)^2 \sum_{j \in \text{triplets}} \varepsilon^{(3b)}(q_j^{(3b)}) + \left(\delta^{(\text{MB})}\right)^2 \sum_{a \in \text{atoms}} \varepsilon^{(\text{MB})}(q_a^{(\text{MB})})
\]

\[
q^{(2b)} = |r_2 - r_1| \equiv r_{12}
\]

\[
q^{(3b)} = \left(\frac{r_{12} + r_{13}}{(r_{12} - r_{13})^2}, \frac{r_{23}}{r_{23}}\right)
\]

\[
q^{(\text{MB})} = ???
\]
Multiple chemical elements

• Straightforward generalisation of neighbour density with species index $\alpha, \beta$

$$\rho_{i}^{\alpha}(r) = \sum_{nlm} c_{nlm}^{(i,\alpha)} g_{n}(|r|) Y_{lm}(\hat{r})$$

$$p_{nn'l}^{\alpha,\beta} = c_{nl}^{\alpha \dagger} \cdot c_{n'l}^{\beta}$$

$$k(\rho_{i}, \rho_{i'}) = \sum_{\alpha,\beta} p^{(i,\alpha,\beta)} \cdot p^{(i',\alpha,\beta)}$$

• Different atomic species are treated as independent

• Possible to generalise even further:

$$k(\rho_{i}, \rho_{i'}) = \sum_{\alpha,\beta,\alpha',\beta'} p^{(i,\alpha,\beta)} \kappa_{\alpha,\beta,\alpha',\beta'} p^{(i',\alpha',\beta')}$$
Example 1: tungsten

Melting point: 3422°C
Example 1: tungsten

Melting point: 3422°C

# Building up databases for tungsten (W)

<table>
<thead>
<tr>
<th>Database:</th>
<th>Computational cost(^a) [ns/atom]</th>
<th>Elastic constants(^b) [GPa]</th>
<th>Phonon spectrum(^b) [THz]</th>
<th>Vacancy formation(^c) [eV]</th>
<th>Surface energy(^b) [eV/Å(^2)]</th>
<th>Dislocation structure(^d) [Å(^{-1})]</th>
<th>Dislocation-vacancy binding energy [eV]</th>
<th>Peierls barrier [eV/(b)]</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAP(_1): 2000 × primitive unit cell with varying lattice vectors</td>
<td>24.70</td>
<td>0.623</td>
<td>0.583</td>
<td>2.855</td>
<td>0.1452</td>
<td>0.0008</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAP(_2): GAP(_1) + 60 × 128 atom cell</td>
<td>51.05</td>
<td>0.608</td>
<td>0.146</td>
<td>1.414</td>
<td>0.1522</td>
<td>0.0006</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAP(_3): GAP(_2) + vacancy in: 400 × 53 atom cell, 20 × 127 atom cell</td>
<td>63.65</td>
<td>0.716</td>
<td>0.142</td>
<td>0.018</td>
<td>0.0941</td>
<td>0.0004</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAP(_4): GAP(_3) + (100), (110), (111), (112) surfaces 180 × 12 atom cell (110), (112) gamma surfaces 6183 × 12 atom cell</td>
<td>86.99</td>
<td>0.581</td>
<td>0.138</td>
<td>0.005</td>
<td>0.0001</td>
<td>0.0002</td>
<td>-0.960</td>
<td>0.108</td>
</tr>
<tr>
<td>GAP(_5): GAP(_4) + vacancy in: (110), (112) gamma surface 750 × 47 atom cell</td>
<td>93.86</td>
<td>0.865</td>
<td>0.126</td>
<td>0.011</td>
<td>0.0001</td>
<td>0.0002</td>
<td>-0.774</td>
<td>0.154</td>
</tr>
<tr>
<td>GAP(_6): GAP(_5) + (\frac{1}{2}(111)) dislocation quadrupole 100 × 135 atom cell</td>
<td>93.33</td>
<td>0.748</td>
<td>0.129</td>
<td>0.015</td>
<td>0.0001</td>
<td>0.0001</td>
<td>-0.794</td>
<td>0.112</td>
</tr>
</tbody>
</table>

\(^a\) Time on a single CPU core of Intel Xeon E5-2670 2.6GHz, \(^b\) RMS error, \(^c\) formation energy error, \(^d\) RMS error of Nye tensor over the 12 atoms nearest the dislocation core, cf. Figure 2.
Interatomic potentials for bcc tungsten

- Error
- 15%
- 50%

DFT reference

- Elastic const.
- C11, C12, C44

- Vacancy energy
- (100), (110), (111), (112)

- Surface energy

- BOP
- MEAM
- FS
Interatomic potentials for bcc tungsten

DFT reference

Error 15%

0

C11 C12 C44
Elastic const.

50%

Vacancy energy

(100) (110) (111) (112)
Surface energy

Database: < 128 atom cells (MD sampled)
200,000 atomic environments
10,000 basis functions

Accuracy: ~ 1 meV/atom

DFT code
CASTEP [37] (version 6.01)
Exchange-correlation functional
PBE
Pseudopotential
Ultramsot (valence 5s2 5p6 5d4 6s2)
Plane-wave energy cutoff
600 eV
Maximum k-point spacing
0.015 Å⁻¹
Electronic smearing scheme
Gaussian
Smearing width
0.1 eV
He in tungsten

(Collaboration with Duc Nguyen-Manh at CCFE)

He binding energy

- Add He-W interaction on top of W potential
- 600 training configurations of 1, 2, 3 He in W with a single vacancy

\[ E = \sum_{ij} \delta_2 E_2(\circ) + \sum_{ijk} \delta_3 E_3(\cdots) + \sum_{ijkl} \delta_4 E_4(\cdots) + \ldots + \sum_i \delta_{\text{soap}} E_s(\circlearrowright) \]

Set $\delta$s sequentially, using the previous error for the next $\delta$
He in tungsten

2 body

2+3 body

2+3+4 body

many-body

He - Vacancy binding energy

eV
Preliminary: a potential for bcc iron (Fe)

(With Nicola Marzari, Daniele Dragoni, Tom Daff)

- Database analogous to W
- Thermal expansion:

![Graph showing thermal expansion vs. temperature]
Preliminary: a potential for bcc iron (Fe)

(With Nicola Marzari, Daniele Dragoni, Tom Daff)

- Database analogous to W
- Thermal expansion:

Use noise control of Gaussian process to compensate for inadequate k-point sampling
Many properties very close to DFT

- Self-interstitial
- Peierls barrier for screw dislocation

Heat capacity

Bain path
Example 2: amorphous Carbon

- Melt-quench protocol:
  - First one with DFT, fitted with GAP, then iterate
  - Add a few randomly distorted diamond and graphite cells

<table>
<thead>
<tr>
<th>Parameter</th>
<th>2-body</th>
<th>3-body</th>
<th>SOAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta$ (eV)</td>
<td>5.0</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>$r_{\text{cut}}$ (Å)</td>
<td>3.7</td>
<td>3.0</td>
<td>3.7</td>
</tr>
<tr>
<td>$r_{\Delta}$ (Å)</td>
<td></td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>$\sigma_{\text{at}}$ (Å)</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$n_{\text{max}}$, $l_{\text{max}}$</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\zeta$</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sparsification</th>
<th>Uniform</th>
<th>Uniform</th>
<th>CUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_t$ (α-C bulk)</td>
<td>125</td>
<td>2500</td>
<td></td>
</tr>
<tr>
<td>$N_t$ (α-C surfaces)</td>
<td>50</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>$N_t$ (crystalline)</td>
<td>25</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>$N_t$ (dimer)</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$N_t$ (total)</td>
<td>15</td>
<td>200</td>
<td>4030</td>
</tr>
</tbody>
</table>
Carbon dimer

(a) Total energies

(b) Force components

(c) Cumulative distribution of energy and force errors

<table>
<thead>
<tr>
<th>State</th>
<th>RMS</th>
<th>P95</th>
<th>(GAP) Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liquid (9000 K)</td>
<td>1.27</td>
<td>6.52</td>
<td>0.19</td>
</tr>
<tr>
<td>Liquid (5000 K)</td>
<td>1.12</td>
<td>5.68</td>
<td>0.20</td>
</tr>
<tr>
<td>Quench (≈ 3500 K)</td>
<td>1.07</td>
<td>5.06</td>
<td>0.21</td>
</tr>
<tr>
<td>Amorphous (300 K)</td>
<td>0.94</td>
<td>2.23</td>
<td>0.42</td>
</tr>
<tr>
<td>Crystalline</td>
<td>0.10</td>
<td>1.32</td>
<td>0.08</td>
</tr>
</tbody>
</table>
Further properties
amorphous carbon

Young's modulus

Surface energies

(a) Young's modulus vs. density (g cm$^{-3}$)
(b) Surface energies vs. slab structure

(a) Unrelaxed surface
(b) Annealed at 1000 K
(c) Annealed at 2000 K
(d) Annealed at 3000 K

Distance from center (Å)

sp$^2$/sp$^3$ count
GAP is a “self-aware” potential

- Train a potential on bulk Si phases
- Introduce defect, colour by predicted error

\[ \hat{\sigma}^2(x) = K(x, x) - K(x, x)^T \left( K^{-1} + \sigma^2 \mathbf{I} \right) K(x, x) \]
GAP is a “self-aware” potential

• Train a potential on bulk Si phases

• Introduce defect, colour by \textit{predicted} error

\[
\hat{\sigma}^2(x) = K(x, x) - K(x, x)^T(K^{-1} + \sigma^2_\nu I)K(x, x)
\]
Same framework for molecules

Quantum chemistry structures and properties of 134 kilo molecules

Raghunathan Ramakrishnan¹, Pavlo O. Draf³,², Matthias Rupp¹ & O. Anatole von Lilienfeld⁴,⁵

Computational de novo design of new drugs and materials requires rigorous and unbiased exploration of chemical compound space. However, large uncharted territories persist due to its size scaling.

(a) MAE [kcal/mol] vs. n. train

PM7 geometry

DFT geometry

(b) Random (RND) × Farthest point (FPS)

(c) RND

(d) FPS
Same framework for molecules

- Forcefield for hydrocarbons (with Max Veit and Shell Ltd)

Saturated hydrocarbons

- 2-body GAP
- SOAP-NN GAP
- AIREBO

Unsaturated hydrocarbons

- (2+3)-body GAP
- SOAP-NN GAP
- AIREBO
Protein-ligand binding

Atom environments $\rightarrow$ entire molecules
Gaussian Approximation Potential framework

- Variety of **descriptors and kernels**: 2b, 3b, Weyl matrix, SOAP, bispectrum, etc.
- Variety of **sparsification** methods: random, k-means, CUR, uniform grids, etc.
- Multiple kernels, train from **energies, forces and stresses**
- Arbitrary **baseline** potentials added/subtracted
- Fortran with deep **python** interface (**quippy**)
  
  - [hub.docker.com/u/libatomsquip](hub.docker.com/u/libatomsquip)
  - [github.com/libAtoms/QUIP](github.com/libAtoms/QUIP)
  - [www.libatoms.org](www.libatoms.org)