



UNIVERSITY OF
OXFORD

Machine Learning

Kernels and GAP

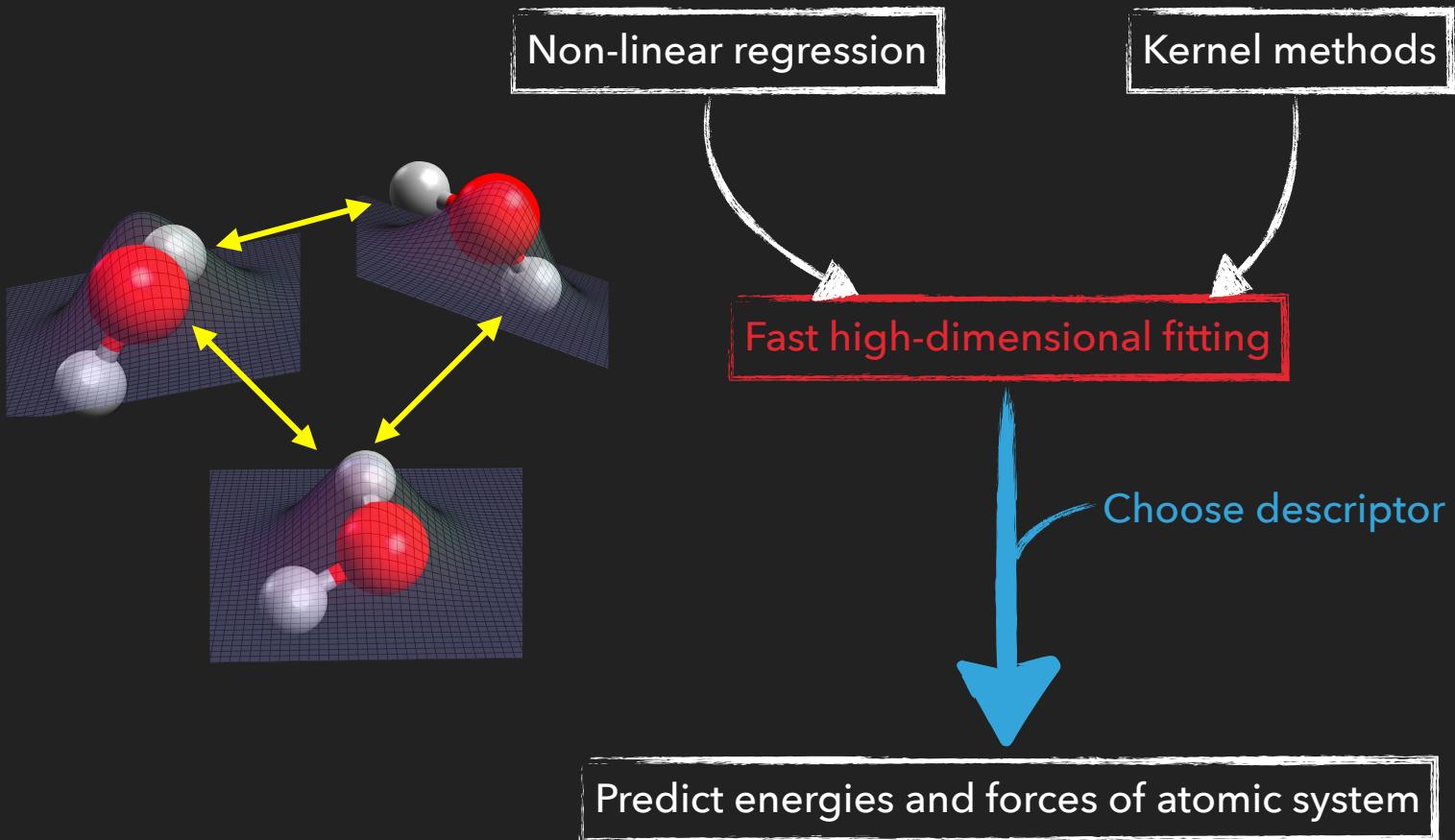
Tristan Johnston-Wood



The Aim



UNIVERSITY OF
OXFORD





Linear Regression



UNIVERSITY OF
OXFORD

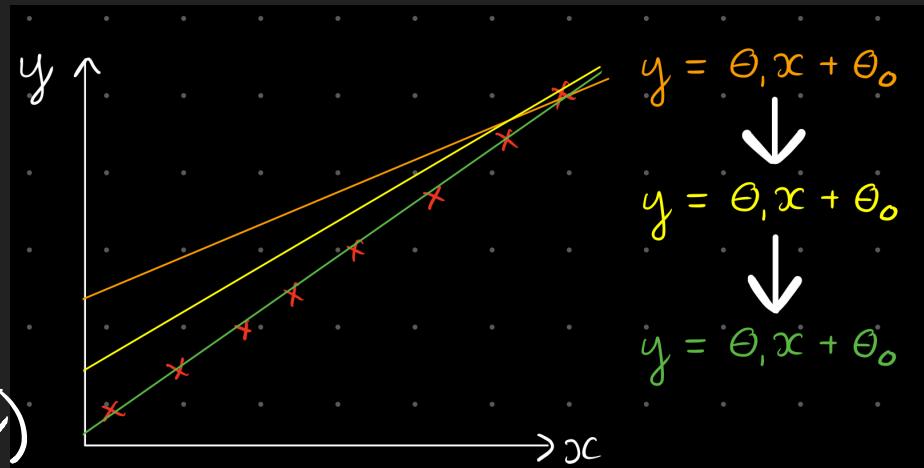
Linear regression recap

$$y = w_1 x + w_0$$



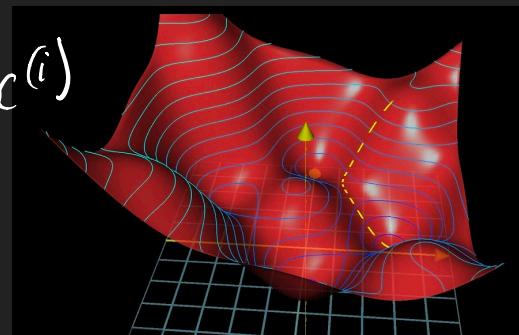
Tweak these to optimise fit

$$w_j := w_j - \alpha \underbrace{\frac{\partial}{\partial w_j} J(w)}$$



$$w_j := w_j + \alpha \sum_{i=1}^n (y^{(i)} - h(x^{(i)})) \cdot x^{(i)}$$

Gradient descent





Non-Linear Regression



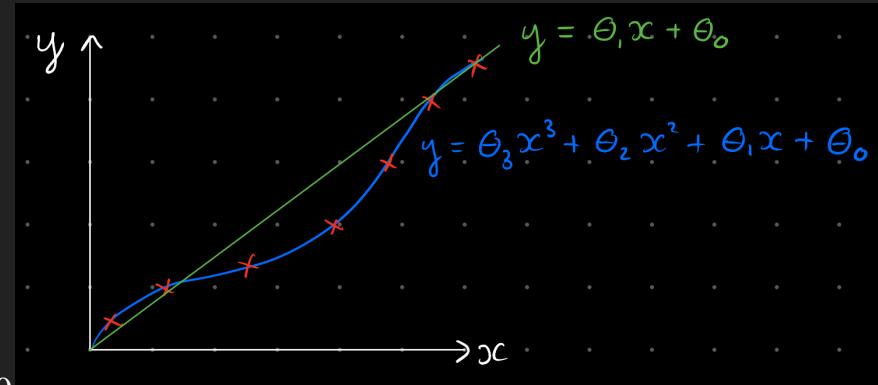
UNIVERSITY OF
OXFORD

What if a linear fit is not suitable?

$$y = w_1x + w_0$$



$$y = w_3x^3 + w_2x^2 + w_1x + w_0$$



Rewrite cubic function as linear function:

$$w_3x^3 + w_2x^2 + w_1x + w_0 = \mathbf{w}^T \phi(x)$$

$$\phi(x) = \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix} \in \mathbb{R}^4$$

$$w_j := w_j + \alpha \sum_{i=1}^n [y^{(i)} - \mathbf{w}^T \phi(x^{(i)})]$$

Gradient descent for $\phi(x)$

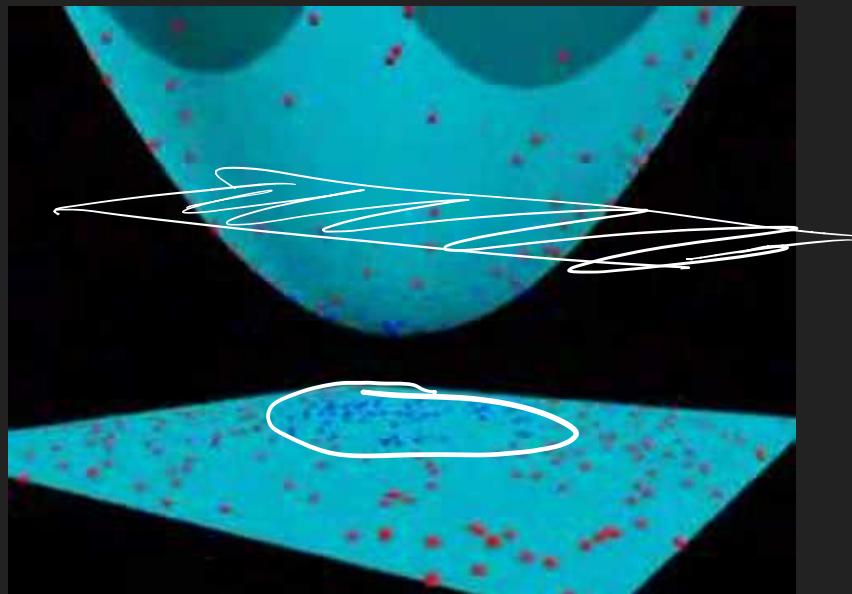


The Kernel Trick



UNIVERSITY OF
OXFORD

So we can project our function to a higher-dimensional space, $\phi(x)$.



This allows us to capture more complexity in our system.



The Kernel Trick



UNIVERSITY OF
OXFORD

When $\phi(x)$ becomes high-dimensional, gradient descent is too expensive

E.g. suppose you have a d -dimensional space and model $\phi(x)$ up to order 3:

$$\phi(x) = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_1^3 \\ x_1^2 x_2 \\ \vdots \end{bmatrix} \in \mathbb{R}^d$$

Dimension: $\phi(x) \sim \mathcal{O}(d^3)$

for $d=1000$, $\phi(x) \sim \mathcal{O}(10^9)$
This will therefore take $\mathcal{O}(10^9)$ time



The Kernel Trick



UNIVERSITY OF
OXFORD

Can we do better than

$\phi(x) \sim \mathcal{O}(d^3)$ -dimensional?

Yes, first assume (not proven):

$$\begin{aligned} w &= \sum_{i=1}^n \beta_i \phi(x^{(i)}) \\ \beta_i &:= \beta_i + \alpha \left((y^{(i)} - w^T \phi(x^{(i)})) \right) \\ &= \beta_i + \alpha \left(y^{(i)} - \underbrace{\sum_{j=1}^n \beta_j \phi(x^{(j)})^T}_{\phi(x^{(i)})} \phi(x^{(i)}) \right) \end{aligned}$$



Kernel Examples



UNIVERSITY OF
OXFORD

$$\beta_i := \beta_i + \alpha \left(y^{(i)} - \sum_{j=1}^n \beta_j \underbrace{\phi(x^{(j)})^T \phi(x^{(i)})}_{\langle \phi(x^{(j)}), \phi(x^{(i)}) \rangle} \right)$$

$$\beta_i := \beta_i + \alpha \left(y^{(i)} - \sum_{j=1}^n \beta_j \underbrace{\langle \phi(x^{(j)}), \phi(x^{(i)}) \rangle}_{K(x, z)} \right)$$

$$K(x, z) = \langle \phi(x), \phi(z) \rangle$$

So we only need to know that $\phi(x)$ exists and use the kernel, K to update our coefficients.

$$K(x, z) \sim \mathcal{O}(d)$$

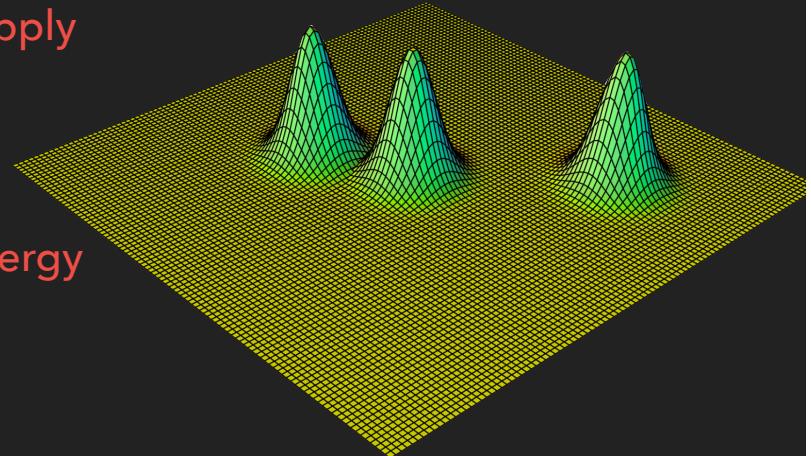


Gaussian Approximation Potential (GAP)



UNIVERSITY OF
OXFORD

That was all very abstract, so let's apply it to a (perhaps) familiar example.



Suppose we want to predict the energy of a configuration of atoms:

$$w = \sum_{i=1}^n \beta_i \phi(x^{(i)})$$

$$\epsilon_j = \sum_{i=1}^n \beta_i \phi(x^{(i)})$$

Model our atoms as densities based on atomic positions



Gaussian Approximation Potential



UNIVERSITY OF
OXFORD

$$\begin{aligned}\langle \varepsilon_j \varepsilon_i \rangle &= \left\langle \sum_{i=1}^n \beta_i \phi(x^{(i)}) \cdot \sum_{j=1}^n \beta_j \phi(x^{(j)}) \right\rangle \\ &= \sigma_\beta^2 \sum_{i,j=1}^n \underbrace{\phi(x^{(i)}) \phi(x^{(j)})}_{\text{Covariance of two atomic energies}}\end{aligned}$$

$$\langle \varepsilon_j \varepsilon_j \rangle = \sigma_\beta^2 K(x^{(i)}, x^{(j)})$$



Gaussian Approximation Potential

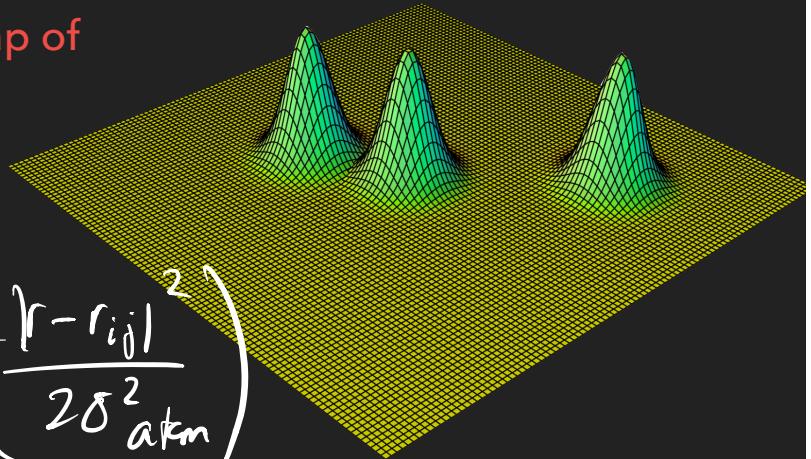


UNIVERSITY OF
OXFORD

$$\langle \varepsilon_i \varepsilon_j \rangle = \sigma_\beta^2 K(x^{(i)}, x^{(j)})$$

Kernel used in GAP: Smooth overlap of atomic positions (SOAP)

$$\rho^{(i)}(r) = \sum_j \exp\left(-\frac{|r - r_{ij}|^2}{2\delta_{\text{atm}}^2}\right)$$



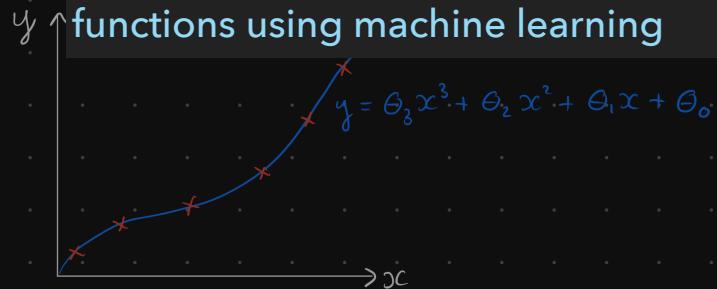


Summary



UNIVERSITY OF
OXFORD

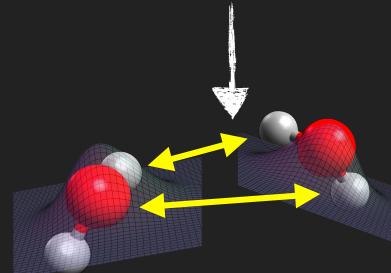
1. Want to fit arbitrarily complicated functions using machine learning



4. Equations slow to solve so use kernel trick

$$K(x, z) \triangleq \langle \phi(x), \phi(z) \rangle$$

3. Just need to calculate kernel, which takes $\mathcal{O}(d)$ time to compute



5. Can be applied to calculating energies and forces of atomic systems



ACKNOWLEDGEMENTS



UNIVERSITY OF
OXFORD



References:

- CS229 - Stanford University Machine Learning
- 3Blue1Brown - Neural Networks
- Bartók, Albert P., and Gábor Csányi. International Journal of Quantum Chemistry 115.16 (2015): 1051-1057.