EPFL

ChE-609

Seyed Mohamad Moosavi, Kevin Maik Jablonka & Berend Smit

Basics of machine learning for chemistry and materials science

seyedmohamad.moosavi@epfl.ch @SMohMoosavi

Ridge regression:

Linear least squares with L2 regularisation

$$\hat{y}_{ML}(x^*) = x^* w = \begin{bmatrix} 1 & x_1^* & \dots & x_d^* \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}$$

Matrix form:

$$\hat{y}_{ML}(X) = Xw = \begin{bmatrix} 1 & x_1^1 & \dots & x_d^1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^m & \dots & x_d^m \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}$$

Loss function: $\mathscr{L} = \|\hat{y}_{ML} - y\|_2^2 + \lambda \|w\|_2^2 = \|Xw - y\|_2^2 + \lambda \|w\|_2^2$

Ridge regression: closed-form and linear kernel

Closed-form solution for ridge regression:

$$w = \arg\min_w \mathscr{L} = \arg\min_w \left(\|\hat{y}_{ML} - y\|_2^2 + \lambda \|w\|_2^2 \right)$$

$$\frac{\partial \mathscr{L}}{\partial w} = 0 \implies w = (X^T X + \lambda I)^{-1} X^T y$$

also, you can write it as (see exercise notes): $w = X^T (X X^T + \lambda I)^{-1} y$

$$a := (X X^T + \lambda I)^{-1} y \implies w = X^T a$$

$$\hat{y}_{ML}(x^*) = x^* w = \begin{bmatrix} 1 & x_1^* & \dots & x_d^* \end{bmatrix} X^T a = \begin{bmatrix} 1 & x_1^* & \dots & x_d^* \end{bmatrix} \begin{bmatrix} x_1^1 & \dots & x_1^m \\ \vdots & \ddots & \vdots \\ x_d^m & \dots & x_d^m \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix}$$

$$= \sum_{i=1}^{m} x^* x_i^T a_i = \sum_{i=1}^{m} K(x^*, x_i)\ a_i$$

Linear kernel

Kernel Ridge Regression (KRR):

It'e equivalent to previous solution, except we changed the basis:

$$\hat{y}_{ML}(X) = Ka\,, a = (K + \lambda I)^{-1} y$$

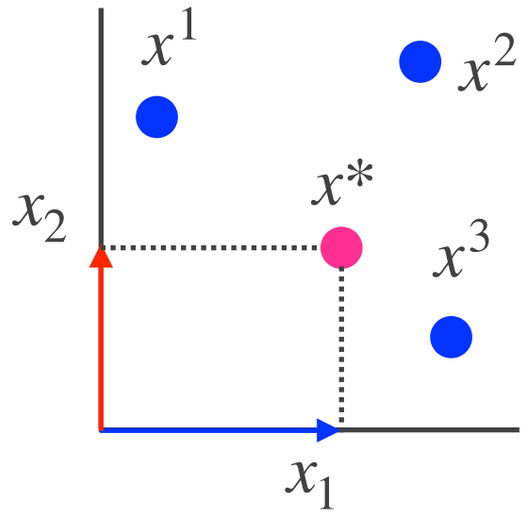
$$\hat{y}_{ML}(x^*) = \sum_{i=1}^{m} K(x^*, x_i)\ \ a_i$$

What does change of basis mean in linear algebra:

For simplicity, let's look at a case with 2 variables and 3 training data

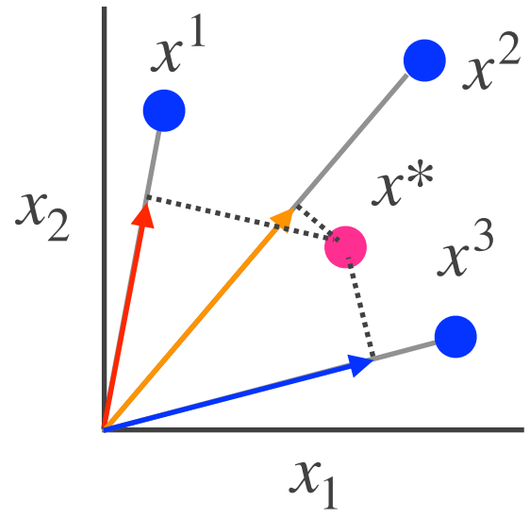
$$\hat{y}_{ML}(x^*) = x^* w = \begin{bmatrix} x_1^* & x_2^* \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = x_1^* w_1 + x_1^* w_2$$

$$= \begin{bmatrix} x_1^* & x_2^* \end{bmatrix} \begin{bmatrix} x_1 & 0 \\ 0 & x_2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$



$$\hat{y}_{ML}(x^*) = \begin{bmatrix} 1 & x_1^* & \dots & x_d^* \end{bmatrix} X^T a$$

$$= \begin{bmatrix} x_1^* & x_2^* \end{bmatrix} \begin{bmatrix} x_1^1 & x_1^2 & x_1^3 \\ x_2^1 & x_2^2 & x_1^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_3 \end{bmatrix}$$



Kernel Ridge Regression (KRR): nonlinearity

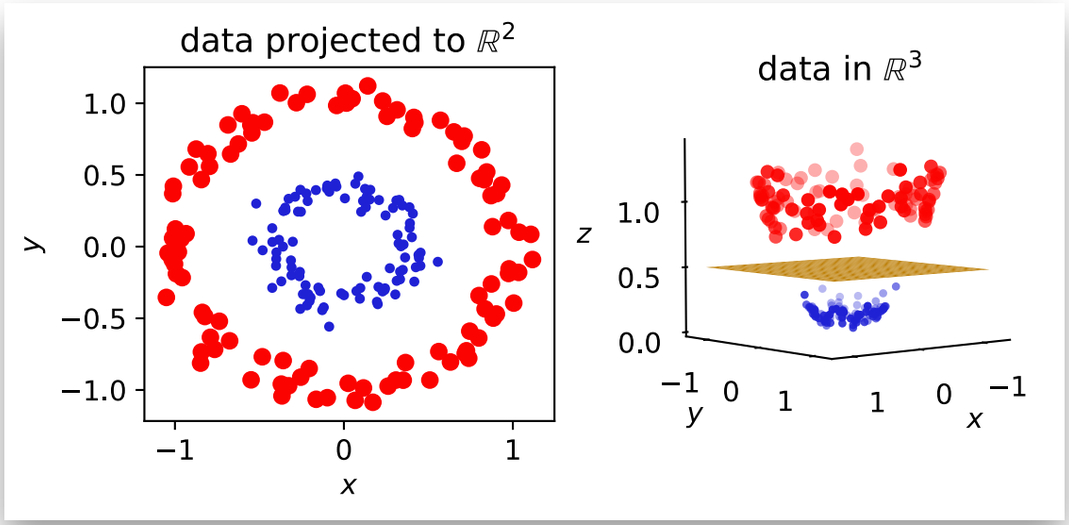
Include nonlinearity, for example a quadratic model:

$$\hat{y}_{ML}(X) = \phi(X)w$$

$$= \begin{bmatrix} 1 & x_1^1 & x_2^1 & (x_1^1)^2 & (x_2^1)^2 & x_1^1 x_2^1 \\ \vdots & \vdots & & & & \\ 1 & x_1^m & x_2^m & (x_1^m)^2 & (x_2^m)^2 & x_1^m x_2^m \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \end{bmatrix}$$

$$K(x^*, x_i) = \langle \phi(x^*), \phi(x_i) \rangle = ((x^*)^T x + 1)^2$$

This makes the data that are not linearly separable in $\mathbb{R}^2$, linearly separable in higher dimensions, still computed efficiently



Kernel Ridge Regression (KRR): kernel trick

This allows us to use non-linear kernels efficiently:

$$\hat{y}_{ML}(X) = \phi(X)w$$

$$w = (\phi(X)^T\phi(X) + \lambda I)^{-1}\phi(X)^T y$$

$$\hat{y}_{ML}(x^*) = \sum_{i=1}^{m} K(x^*, x_i) \quad a_i$$

$$K(x^*, x_i) = \langle \phi(x^*), \phi(x_i) \rangle$$

In the case of linear kernel:

$$K(x^*, x_i) = \langle x^*, x_i \rangle$$

In the case of quadratic kernel:

$$K(x^*, x_i) = \langle \phi(x^*), \phi(x_i) \rangle = ((x^*)^T x + 1)^2$$

Kernel Ridge Regression (KRR): Gaussian kernel

The Gaussian kernel is very popular in chemistry

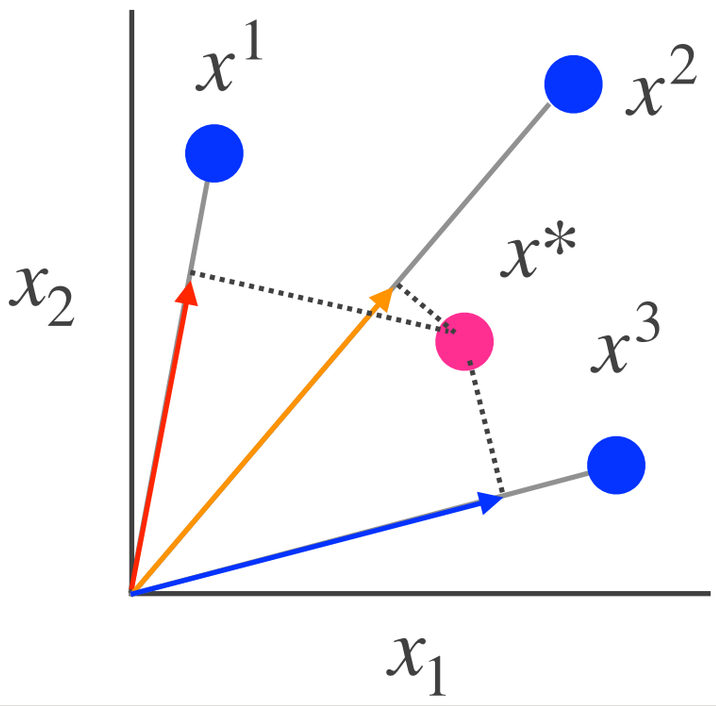
$$K(x^*, x) = \exp\left(-\gamma \|x^* - x\|_2^2\right)$$

This kernel provides an intuitive notion of similarity, i.e. distance in feature space:

$$\hat{y}_{ML}(x^*) = \sum_{i=1}^{m} a_i K(x^*, x^i)$$

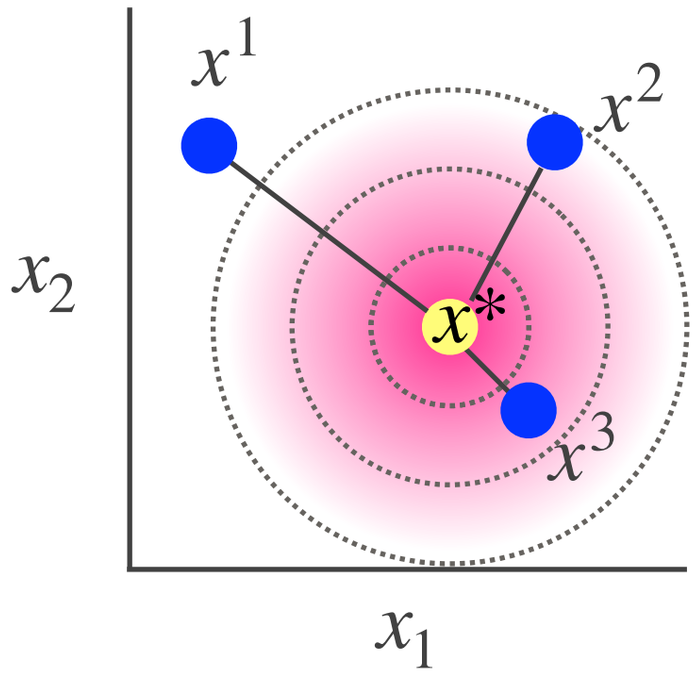
Linear basis

$K(x^*, x_i) = \langle x^*, x_i \rangle$



Radial basis

$K(x^*, x_i) = \exp\left(-\gamma \|x^* - x_i\|_2^2\right)$



Summary

- Kernel and linear models
- Kernel trick
- Similarity and Gaussian kernel
- One should do CV to obtain the hyperparameters of kernel