

# MACHINE LEARNING IN BIOINFORMATICS

## FROM LINEAR REGRESSION TO KERNEL REGRESSION

Philipp Benner

*philipp.benner@bam.de*

VP.1 - eScience

Federal Institute of Materials Research and Testing (BAM)

April 25, 2024

# LINEAR REGRESSION

- Solid understanding of linear regression allows us to understand many aspects of complex models, including neural networks
- Many models can be derived from linear regression, including polynomial, kernel, and logistic regression, as well as neural networks
- We start from a Bayesian perspective and show how to derive the linear regression model and a method for parameter estimation with a specific focus on model assumptions

- Bayes theorem:

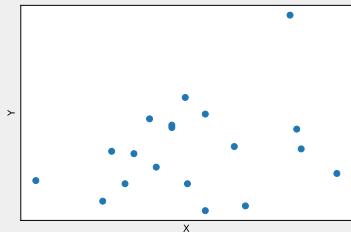
$$\text{pr}(H | X) = \frac{\text{pr}(X | H)\text{pr}(H)}{\text{pr}(X)}$$

where  $\text{pr}(H | X)$  is the posterior distribution of a hypothesis  $H$  given observed data  $X$ ,  $\text{pr}(X | H)$  the likelihood,  $\text{pr}(H)$  the prior distribution, and  $\text{pr}(X)$  the marginal likelihood

- $H$  is our hypothesis and can take many forms, e.g.
  - ▶ In case of the spam classifier we had  $H = \text{'spam'}$
  - ▶  $H$  can also refer to the parameter of a distribution, e.g. when we want to estimate the mean of a normal distribution
- In any case, probabilities depend on our model assumptions and therefore are a subjective choice

# LINEAR REGRESSION

Let **Y** be the dependent variable (response variable) and **X** the independent variable (covariate, or predictor):



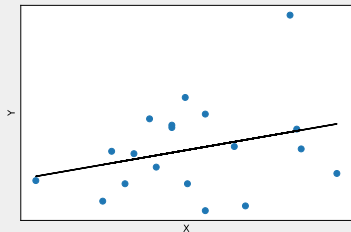
We assume the following model

$$\mathbf{Y} = f(\mathbf{X}) + \epsilon$$

where  $f$  is a linear function that models the expectation  $\mathbb{E}[Y | X]$ , and  $\epsilon$  is a noise term (e.g.  $\epsilon \sim \text{Normal}(0, \sigma^2)$ )

# LINEAR REGRESSION

Let **Y** be the dependent variable (response variable) and **X** the independent variable (covariate, or predictor):



We assume the following model

$$\mathbf{Y} = f(\mathbf{X}) + \epsilon$$

where  $f$  is a linear function that models the expectation  $\mathbb{E}[Y | X]$ , and  $\epsilon$  is a noise term (e.g.  $\epsilon \sim \text{Normal}(0, \sigma^2)$ )

# LINEAR REGRESSION

- We can also write  $\mathbf{Y} \sim \text{Normal}(f(\mathbf{X}), \sigma^2)$
- We assume no distribution for  $\mathbf{X}$
- We assume  $f$  is a linear function, i.e.

$$f(x) = ax + b$$

- How can we generate data  $(x_i, y_i)_i$  with this model?
  - ▶ For  $i = 1, \dots, n$ :
    - Select some value for  $x_i$
    - Draw  $\epsilon_i$  from  $\text{Normal}(0, \sigma^2)$
    - Compute  $y_i = f(x_i) + \epsilon_i$

# LINEAR REGRESSION - PARAMETER ESTIMATION

- In the Bayesian framework, parameters are estimated using the posterior distribution
- We want to know the probability of our hypothesis or parameters  $\theta = (a, b)$  given a set of  $n$  observations  $x = (x_i)_{i=1}^n$  and  $y = (y_i)_{i=1}^n$

- An estimate  $\hat{\theta}$  of our parameters  $\theta$  can be computed as the *maximum a posteriori (MAP) estimate*

$$\hat{\theta} = \arg \max_{\theta} \text{pr}(\theta | x, y)$$

- There are other choices, for instance the *posterior expectation*, which all have their justifications
- We use the MAP for linear regression, because it leads to a computationally simple solution



# LINEAR REGRESSION - PARAMETER ESTIMATION

- For a flat prior, the MAP is equivalent to the *maximum likelihood estimate (MLE)*, i.e.

$$\begin{aligned}\hat{\theta} &= \arg \max_{\theta} \text{pr}(\theta | \mathbf{x}, \mathbf{y}) \\ &= \arg \max_{\theta} \frac{\text{pr}(\mathbf{x}, \mathbf{y} | \theta) \text{pr}(\theta)}{\text{pr}(\mathbf{x}, \mathbf{y})} \\ &= \arg \max_{\theta} \text{pr}(\mathbf{x}, \mathbf{y} | \theta) \text{pr}(\theta) \\ &= \arg \max_{\theta} \text{pr}(\mathbf{x}, \mathbf{y} | \theta)\end{aligned}$$

assuming  $\text{pr}(\theta)$  is constant<sup>1</sup>

- This result is not specific to linear regression models

---

<sup>1</sup>A uniform prior  $\text{pr}(\theta)$  is called *improper prior* when  $\theta$  is a continuous variable, because  $\text{pr}(\theta)$  does not integrate to one

- Furthermore, we have

$$\begin{aligned}\hat{\theta} &= \arg \max_{\theta} \text{pr}(\mathbf{x}, \mathbf{y} | \theta) \\ &= \arg \max_{\theta} \text{pr}(\mathbf{y} | \mathbf{x}, \theta) \text{pr}(\mathbf{x} | \theta) \\ &= \arg \max_{\theta} \text{pr}(\mathbf{y} | \mathbf{x}, \theta)\end{aligned}$$

- In the last step we took advantage of the fact that the distribution of our covariates  $\mathbf{x}$  does not depend on the parameters  $\theta$ , which are the slope and intercept of the linear function
- In fact, we do not have to assume a particular distribution for our covariates!

# LINEAR REGRESSION - OLS

- Plugging in our normal distribution we arrive at

$$\begin{aligned}\hat{\theta} &= \arg \max_{\theta} \text{pr}(y_1 \dots y_n \mid x_1, \dots, x_n, \theta) \\ &= \arg \max_{\theta} \prod_{i=1}^n \text{pr}(y_i \mid x_i, \theta) \\ &= \arg \max_{\theta} \sum_{i=1}^n \log \text{pr}(y_i \mid x_i, \theta) \\ &= \arg \max_{\theta} \sum_{i=1}^n \log \frac{1}{\sigma \sqrt{2\pi}} \exp \left\{ -\frac{(y_i - f(x_i))^2}{2\sigma^2} \right\} \\ &= \arg \max_{\theta} \sum_{i=1}^n -(y_i - f(x_i))^2\end{aligned}$$

- The estimate

$$\begin{aligned}\hat{\theta} &= \arg \min_{\theta} \sum_{i=1}^n (y_i - f(x_i))^2 \\ &= \arg \min_{\theta} \sum_{i=1}^n (y_i - \hat{y}_i)^2\end{aligned}$$

is called the *ordinary least squares (OLS)* estimate

- It minimizes the squared error between our prediction  $\hat{y}_i$  and our observations  $y_i$
- In other words, it minimizes the squared residuals  $\epsilon_i = y_i - f(x_i)$

# LINEAR REGRESSION - GENERALIZATION

- For generalizing linear regression to multiple predictors, we first define

$$\mathbf{x} = \begin{bmatrix} 1 \\ \tilde{\mathbf{x}} \end{bmatrix}, \quad \boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$

i.e.  $\mathbf{x}$  is a vector where the first component is always 1

- This definition allows to write

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{b} + \mathbf{a}\tilde{\mathbf{x}} \\ &= \theta_1 + \theta_2\tilde{\mathbf{x}} \\ &= \begin{bmatrix} 1 \\ \tilde{\mathbf{x}} \end{bmatrix}^\top \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \\ &= \mathbf{x}^\top \boldsymbol{\theta} \end{aligned}$$

# LINEAR REGRESSION - GENERALIZATION

- Adding additional predictors is now very simple

$$x = \begin{bmatrix} 1 \\ x^{(2)} \\ \vdots \\ x^{(p)} \end{bmatrix}, \quad \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{bmatrix}$$

- The number of predictors / features is given by  $p$ , where the first predictor is  $(1, 1, \dots, 1)^\top$
- It follows that

$$\begin{aligned} f(x) &= x^\top \theta \\ &= \theta_1 + x^{(2)}\theta_2 + \dots + x^{(p)}\theta_p \end{aligned}$$

# LINEAR REGRESSION - NOTATION

- In general, we have  $n$  observations and  $p$  predictors
- For the  $i$ th observation  $(x_i, y_i)$ ,  $y_i$  is a scalar and  $x_i$  a vector

$$x_i = (1, x_i^{(2)}, \dots, x_i^{(p)})^\top$$

- We define the matrix

$$X = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(p)} \\ x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(p)} \\ \vdots & \vdots & \ddots & \vdots \\ x_n^{(1)} & x_n^{(2)} & \dots & x_n^{(p)} \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(2)} & \dots & x_1^{(p)} \\ 1 & x_2^{(2)} & \dots & x_2^{(p)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n^{(2)} & \dots & x_n^{(p)} \end{bmatrix}$$

# LINEAR REGRESSION - NOTATION

- This notation allows us to write linear regression as

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(2)} & \dots & x_1^{(p)} \\ 1 & x_2^{(2)} & \dots & x_2^{(p)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n^{(2)} & \dots & x_n^{(p)} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

- Or in matrix notation simply as

$$y = X\theta + \epsilon$$

## Data matrix $X$

For a data matrix  $X \in \mathbb{R}^{n \times p}$ , rows will always correspond to observations and columns correspond to features. The first column is the vector  $(1, 1, \dots, 1)^\top$ . We always assume that  $X$  has full rank, i.e.  $\text{rank}(X) = \min(n, p)$



If  $n > p$  and  $X^T X$  has full rank we can use **ordinary least squared (OLS)** to estimate  $\theta$ :

$$\hat{\theta} = \arg \min_{\theta} \|\epsilon\|_2^2 = \arg \min_{\theta} \|y - X\theta\|_2^2$$

Differentiation with respect to  $\theta$  and solving for the roots leads to:

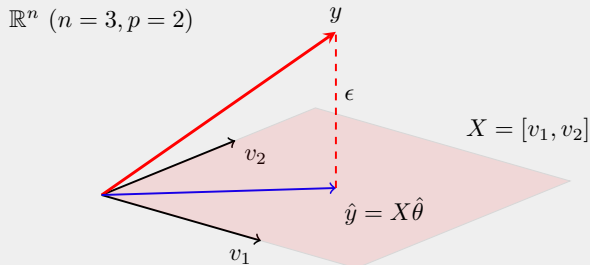
$$\begin{aligned} \Rightarrow \quad \hat{\theta} &= (X^T X)^{-1} X^T y \\ &= X^T y \quad \text{if } X^T X = I \end{aligned}$$

$X(X^T X)^{-1} X^T$  is called a projection matrix...

# LINEAR REGRESSION - OLS PROJECTION

Let  $X\theta = v_1\theta_1 + v_2\theta_2 + \dots + v_p\theta_p$ , where  $v_i$  denotes the  $i$ th column of  $X$

$$\hat{\theta} = \arg \min_{\theta} \|y - X\theta\|_2^2$$



$X(X^T X)^{-1}X^T y$  projects  $y$  onto the plane defined by the columns of  $X$

---

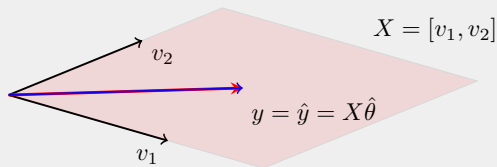
<sup>1</sup>[Hastie et al., 2009]

# LINEAR REGRESSION - OLS PROJECTION

Let  $X\theta = v_1\theta_1 + v_2\theta_2 + \dots + v_p\theta_p$ , where  $v_i$  denotes the  $i$ th column of  $X$

$$\hat{\theta} = \arg \min_{\theta} \|y - X\theta\|_2^2$$

$\mathbb{R}^n$  ( $n = 3, p = 2$ )



If  $y$  is already inside the plane, we obtain  $\epsilon = 0$

---

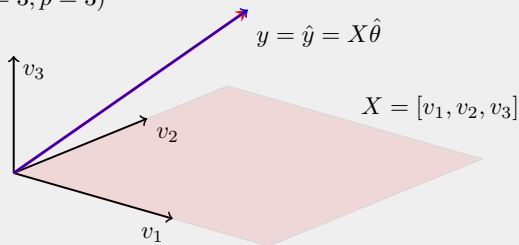
<sup>1</sup>[Hastie et al., 2009]

# LINEAR REGRESSION - OLS PROJECTION

Let  $X\theta = v_1\theta_1 + v_2\theta_2 + \dots + v_p\theta_p$ , where  $v_i$  denotes the  $i$ th column of  $X$

$$\hat{\theta} = \arg \min_{\theta} \|y - X\theta\|_2^2$$

$\mathbb{R}^n$  ( $n = 3, p = 3$ )



If  $p \geq n$  then  $\epsilon = 0$  and for  $p > n$  we have infinitely many solutions (assuming  $v_i$  are pairwise independent)

<sup>1</sup>[Hastie et al., 2009]

# LINEAR REGRESSION - UNDERDETERMINED OLS

- For  $p > n$  the OLS estimate

$$\hat{\theta} = \arg \min_{\theta} \|y - X\theta\|_2^2$$

has infinitely many solution  $\hat{\theta}$  such that  $\|y - X\hat{\theta}\|_2^2 = 0!$

# LINEAR REGRESSION - UNDERDETERMINED OLS

- For  $p > n$  the OLS estimate

$$\hat{\theta} = \arg \min_{\theta} \|y - X\theta\|_2^2$$

has infinitely many solution  $\hat{\theta}$  such that  $\|y - X\hat{\theta}\|_2^2 = 0!$

- Which one should we choose?

# LINEAR REGRESSION - UNDERDETERMINED OLS

- For  $p > n$  the OLS estimate

$$\hat{\theta} = \arg \min_{\theta} \|y - X\theta\|_2^2$$

has infinitely many solution  $\hat{\theta}$  such that  $\|y - X\hat{\theta}\|_2^2 = 0!$

- Which one should we choose?
- Remember our initial model

$$y = X\theta + \epsilon$$

and yet the estimate  $\hat{\theta}$  satisfies  $y = X\hat{\theta}$

# LINEAR REGRESSION - UNDERDETERMINED OLS

- For  $p > n$  the OLS estimate

$$\hat{\theta} = \arg \min_{\theta} \|y - X\theta\|_2^2$$

has infinitely many solution  $\hat{\theta}$  such that  $\|y - X\hat{\theta}\|_2^2 = 0!$

- Which one should we choose?
- Remember our initial model

$$y = X\theta + \epsilon$$

and yet the estimate  $\hat{\theta}$  satisfies  $y = X\hat{\theta}$

- Either  $\epsilon = 0$  or  $\hat{\theta}$  contains all the noise



# LINEAR REGRESSION - UNDERDETERMINED OLS

For instance, we could take that  $\theta$  with minimal length, i.e. the **minimum  $\ell_2$ -norm** solution<sup>2</sup>

$$\begin{aligned} & \arg \min_{\theta} \|\theta\|_2^2 \\ & \text{subject to } X\theta = y \end{aligned}$$

The solution is almost equivalent to the standard OLS solution, i.e.

$$\hat{\theta} = (X^T X)^+ X^T y$$

where  $(X^T X)^+$  Moore-Penrose pseudoinverse<sup>3</sup> of  $X^T X$ .

---

<sup>2</sup>**Common practice for training neural networks**

<sup>3</sup>The Moore-Penrose pseudoinverse of a matrix  $X$  is computed as follows: Let  $X = S\Sigma V^T$  be the singular value decomposition of  $X$ , where  $\Sigma$  is a diagonal matrix containing the singular values.  $X^+ = S\Sigma^+ V^T$  where  $\Sigma^+$  contains the reciprocal of all non-zero singular values.

## Ridge Regression

The ridge regression estimate is defined as

$$\hat{\theta}(\lambda) = \arg \min_{\theta} \|X\theta - y\|_2^2 + \lambda \|\theta\|_2^2$$

where  $\lambda$  is called the *regularization strength* or *penalty*. Note that  $\|\theta\|_2^2 = \sum_{i=2}^n \theta_i^2$ , i.e.  $\theta_1$  is not constrained

- There exists an analytical solution to the ridge estimate:

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

- In the overparameterized case, for  $\lambda > 0$  we obtain  $\|\epsilon\|_2^2 > 0$

---

<sup>3</sup>Convex optimization: [Boyd and Vandenberghe, 2004]

# LINEAR REGRESSION - RIDGE REGRESSION

- For  $\lambda \rightarrow \infty$  the estimate  $\lambda \hat{\theta}(\lambda)$  converges to the componentwise regression estimator
- For  $\lambda \rightarrow 0$  the estimate  $\hat{\theta}(\lambda)$  converges to the minimum  $\ell_2$ -norm OLS solution<sup>4</sup>
- The penalty  $\lambda \|\theta\|_2^2$  can be interpreted as a Gaussian prior
- Ridge regression is useful when  $n < p$  and  $n \geq p$

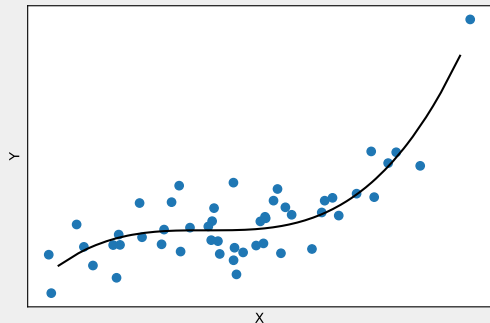
---

<sup>4</sup> $A + \lambda I$  is invertible even for very small  $\lambda$ . In numerics,  $A + \lambda I$  is also used as a trick to ensure that a matrix is positive-definite.

# **KERNEL REGRESSION**

# POLYNOMIAL REGRESSION

- How can we change linear regression to model non-linear relations between **X** and **Y**?



# REGRESSION IN FEATURE SPACE

Polynomial regression

$$\mathbf{Y} = \theta_1 + \theta_2\mathbf{X} + \theta_3\mathbf{X}^2 + \theta_4\mathbf{X}^3 + \dots + \epsilon,$$

More generally, we write

$$\mathbf{Y} = \phi(\mathbf{X})\theta + \epsilon,$$

where  $\phi : \mathbb{R}^p \rightarrow \mathbb{R}^{p'}$  is a **feature map** that maps points in  $p$ -dimensional input space into a  $p'$ -dimensional feature space, e.g.

$$\phi(\mathbf{X}) = (1, \mathbf{X}, \mathbf{X}^2, \mathbf{X}^3, \dots)$$

Basically linear (or ridge) regression in  $p'$ -dimensional feature space, but non-linear in input space

# KERNEL REGRESSION

- What if we do not know the exact set of features for our data?
- Can we simply test a large amount of possible features?
- Can we have more features than observations, i.e.  $n \leq p$ ?

Ridge regression in feature space:

$$\hat{\theta}(\lambda) = \arg \min_{\theta} \|\phi(X)\theta - \mathbf{y}\|_2^2 + \lambda \|\theta\|_2^2$$

where  $\phi$  is applied to each row of  $X$ , i.e.  $\phi(X) \in \mathbb{R}^{n \times p'}$ .

**Computationally expensive if  $p' \gg p$  and  $n \gg 1$ , assuming  $X$  is not sparse.**

Reformulate the ridge regression estimate

$$\hat{\theta}(\lambda) = \arg \min_{\theta} \|\phi(X)\theta - \mathbf{y}\|_2^2 + \lambda \|\theta\|_2^2$$

using **kernels**. Let  $\theta = \phi(X)^\top \eta$ , where  $\eta \in \mathbb{R}^n$  is a new parameter vector and  $\theta \in \text{span}(\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_n)) \subset \mathbb{R}^p$ . It follows that

$$\begin{aligned} \hat{\eta}(\lambda) &= \arg \min_{\eta} \left\| \phi(X)\phi(X)^\top \eta - \mathbf{y} \right\|_2^2 + \lambda \left\| \phi(X)^\top \eta \right\|_2^2 \\ &= \arg \min_{\eta} \left\| K\eta - \mathbf{y} \right\|_2^2 + \lambda \eta^\top K\eta \end{aligned}$$

where  $K = \phi(X)\phi(X)^\top \in \mathbb{R}^{n \times n}$  is the **kernel matrix**.



## Definition: Kernel function

A function  $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  is called a *kernel* if there exists a feature map  $\phi : \mathcal{X} \rightarrow \mathcal{F}$  such that

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$$

$K = (\kappa(\mathbf{x}_i, \mathbf{x}_j))_{\mathbf{x}_i \in \mathcal{X}, \mathbf{x}_j \in \mathcal{X}}$  is called the kernel matrix.

- $\mathcal{X}$  can be an arbitrary space, for instance DNA sequences
- $\kappa(\mathbf{x}_i, \mathbf{x}_j)$  is interpreted as a similarity measure in feature space
- Evaluating  $\kappa(\mathbf{x}_i, \mathbf{x}_j)$  does not always require to explicitly compute  $\phi(\mathbf{x})$
- Not having to map data into feature space is called the **kernel trick**

# EXAMPLE KERNELS

## ■ Linear kernel

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j, \text{ where } \phi(\mathbf{x}) = \mathbf{x}$$

## ■ Polynomial kernel

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + 1)^d$$

where  $d > 0$  is the degree. For  $\mathcal{X} = \mathbb{R}^2$  and  $d = 2$

$$\phi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)^\top$$

## ■ Radial basis function (RBF) kernel

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right)$$

where the feature space has infinite dimensions

# PREDICTIONS

Let  $x_{\text{new}}$  denote the position where we would like to compute a prediction  $\hat{y}$

- Linear Regression

$$\hat{y} = \phi(x_{\text{new}})^\top \hat{\theta}$$

- Kernel Regression

$$\hat{y} = \sum_{i=1}^n \kappa(x_i, x_{\text{new}}) \hat{\eta}_i = \phi(x_{\text{new}})^\top \phi(X)^\top \hat{\eta}$$

which requires the full training set  $X = (x_i)_i \in \mathbb{R}^{n \times p}$ , where we simply used the definition  $\theta = \phi(X)^\top \eta$  to replace  $\hat{\theta}$  in the prediction of the linear regression model

# PARAMETERS AND HYPERPARAMETERS

- We call  $\theta$  and  $\eta$  the **parameters** of a (kernel) regression model
- The parameters of a kernel function (e.g.  $\sigma^2$  for the RBF kernel) or the regularization strength  $\lambda$  are also parameters of the model, but one step further up the hierarchy
- We call the parameters of a kernel function and the regularization strength **hyperparameters**
- In a Bayesian setting, the parameters control the likelihood function, whereas the hyperparameters parametrize the prior distribution

# KERNEL REGRESSION - PROS AND CONS

## Pros:

- Computationally efficient regression for high-dimensional feature spaces for moderate data sets
- Implicit regularization, i.e. only as many parameters as data points (but equivalent to minimum  $\ell_2$ -norm solution of standard regression)

## Cons:

- Kernel matrix grows quadratically with number of samples
- $\theta \in \mathbb{R}^p \rightsquigarrow \eta \in \mathbb{R}^n$ , which creates dependencies between features
- Interpretation of parameters in feature space requires computation of  $\phi(X)^\top \eta$
- For infinite feature spaces  $\phi$  cannot be computed
- No feature selection possible ( $\ell_1$  penalty)

# **RANDOM FEATURES**

# RANDOM FEATURES

Kernel matrix grows quadratically with the number of data points, which prevents kernel methods to be applied to large data sets.

Basic idea<sup>5</sup>: Define a mapping  $\xi : \mathcal{X} \rightarrow \mathbb{R}^q$  with  $q \ll p$  such that

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) \approx \xi(\mathbf{x}_i)^\top \xi(\mathbf{x}_j)$$

Regression can then be performed in  $\mathbb{R}^q$  after explicitly mapping each data point to the reduced feature space.

How do we compute  $\xi$ ?

---

<sup>5</sup>[Rahimi et al., 2007]

## Bochner's theorem

A continuous shift-invariant kernel  $\kappa : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{C}$  with  $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \kappa(\mathbf{x}_i - \mathbf{x}_j)$  is positive definite iff there exists a non-negative measure  $\mu$  such that

$$\begin{aligned}\kappa(\mathbf{x}_i - \mathbf{x}_j) &= \int_{\mathbb{R}^d} \exp(i\omega^\top (\mathbf{x}_i - \mathbf{x}_j)) \, d\mu(\omega) \\ &= \mathbb{E}_\omega \exp(i\omega^\top (\mathbf{x}_i - \mathbf{x}_j)) = \mathbb{E}_\omega \exp(i\omega^\top \mathbf{x}_i) \exp(i\omega^\top \mathbf{x}_j)^* .\end{aligned}$$

I.e. the kernel  $\kappa$  is the (inverse) Fourier transform of  $\mu$ .

When both  $\kappa$  and  $\mu$  are real-valued then

$$\kappa(\mathbf{x}_i - \mathbf{x}_j) = \mathbb{E}_\omega \cos(\omega^\top (\mathbf{x}_i - \mathbf{x}_j))$$

---

<sup>5</sup> $x^*$  is the complex conjugate of  $x$  and remember that  $\exp(ix)^* = \exp(-ix)$



## Monte Carlo approximation

Let  $\mu$  be a distribution and  $\omega$  a random variable with distribution  $\mu$ . From the law of large numbers it follows that

$$\mathbb{E}_{\omega} f(\omega) = \int f(x) d\mu(x) \approx \frac{1}{q} \sum_{j=1}^q f(\omega_j)$$

where  $\omega_1, \dots, \omega_q$  are independent samples from  $\mu$ .

Monte Carlo approximation of the Fourier integral

$$\omega_k \stackrel{i.i.d.}{\sim} \mu$$

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) \approx \frac{1}{q} \sum_{k=1}^q \exp(i\omega_k^\top (\mathbf{x}_i - \mathbf{x}_j)) = \xi(\mathbf{x}_j)^* \xi(\mathbf{x}_i)$$

where  $\xi(\mathbf{x}) = \frac{1}{\sqrt{q}} (\exp(i\omega_1^\top \mathbf{x}), \dots, \exp(i\omega_q^\top \mathbf{x}))^\top$ .

In practice: We know the kernel  $\kappa$  and must derive the measure  $\mu$ . Afterwards, we can approximate  $\kappa$  by drawing  $q$  samples  $\omega_k$  from  $\mu$  and map  $x$  into feature space using

$$\xi(x) = \frac{1}{\sqrt{q}} \left( \exp(i\omega_1^\top x), \dots, \exp(i\omega_q^\top x) \right)^\top.$$

The measure  $\mu$  is given by the Fourier transform of  $\kappa$  with density

$$f_\mu(\omega) = \int_{\mathbb{R}^d} \exp(-i\omega^\top \delta) \kappa(\delta) d\delta, \quad \text{where } \delta = x_i - x_j$$

# RANDOM FEATURES

Example: Radial basis function (RBF) kernel (infinite dimensional feature space)

$$\kappa(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|_2^2}{2\sigma^2}\right)$$

The measure  $\mu$  is given by a spherical normal distribution ( $\Sigma = \sigma^2 I$ ) with density

$$f_\mu(\omega) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{\|\omega\|_2^2}{2\sigma^2}\right)$$

Since  $\kappa$  and  $\mu$  are real, we have

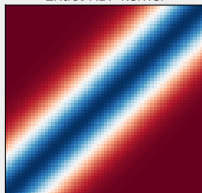
$$\xi(x) = \frac{1}{\sqrt{q}} \left( \cos(\omega_1^\top x), \sin(\omega_1^\top x), \dots, \cos(\omega_q^\top x), \sin(\omega_q^\top x) \right)^\top$$

---

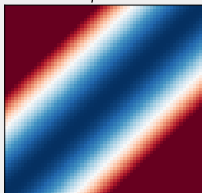
<sup>5</sup> $\cos(x_i - x_j) = \cos(x_i)\cos(x_j) + \sin(x_i)\sin(x_j)$

# RANDOM FEATURES

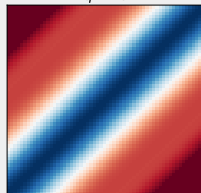
Exact RBF kernel



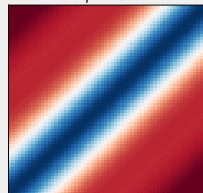
$q = 1$



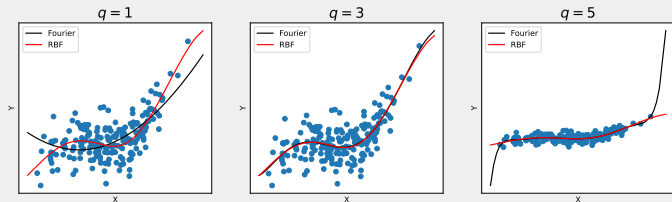
$q = 10$



$q = 100$

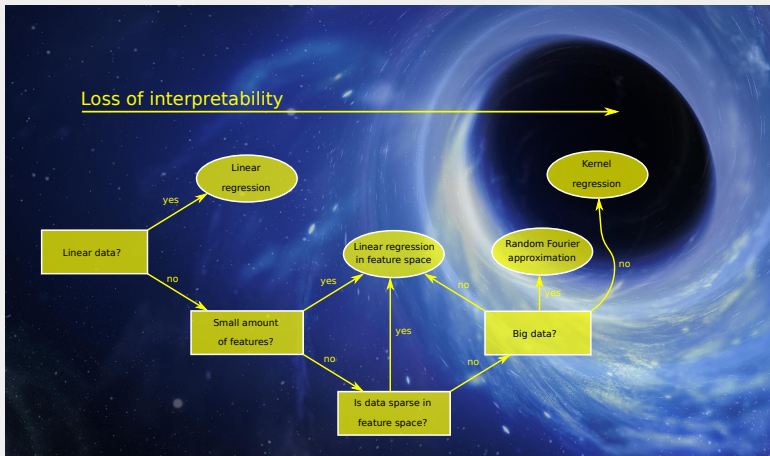


# RANDOM FEATURES






- Kernel regression is not identical to linear regression with random Fourier features
- As many parameters as random Fourier features
- Regularization must be used to prevent overfitting

# GUIDE TO KERNEL REGRESSION



<sup>5</sup>The complexity of kernel regression can be reduced by computing approximate solutions with batch gradient descent

# REFERENCES

-  BOYD, S. AND VANDENBERGHE, L. (2004).  
**CONVEX OPTIMIZATION.**  
Cambridge university press.
-  HASTIE, T., TIBSHIRANI, R., AND FRIEDMAN, J. (2009).  
**THE ELEMENTS OF STATISTICAL LEARNING: DATA MINING, INFERENCE, AND PREDICTION.**  
Springer Science & Business Media.
-  RAHIMI, A., RECHT, B., ET AL. (2007).  
**RANDOM FEATURES FOR LARGE-SCALE KERNEL MACHINES.**  
In *NIPS*, volume 3, page 5. Citeseer.