Machine Learning in Bioinformatics

From Linear Regression to Kernel Regression

Philipp Benner
philipp.benner@bam.de

S.3 - eScience
Federal Institute for Materials Research and Testing (BAM)

June 25, 2023

Linear Regression

Motivation

- 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

- Bayes theorem:

$$\mathrm{pr}(H \mid X) = \frac{\mathrm{pr}(X \mid H)\mathrm{pr}(H)}{\mathrm{pr}(X)}$$

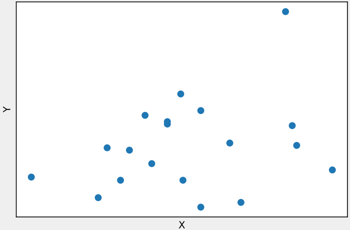
  where $\mathrm{pr}(H \mid X)$ is the posterior distribution of a hypothesis H given observed data X, $\mathrm{pr}(X \mid H)$ the likelihood, $\mathrm{pr}(H)$ the prior distribution, and $\mathrm{pr}(X)$ the marginal likelihood

- H is our hypothesis and can take many forms, e.g.
  - In case of the spam classifier we had $H = '\mathrm{spam}'$
  - 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):



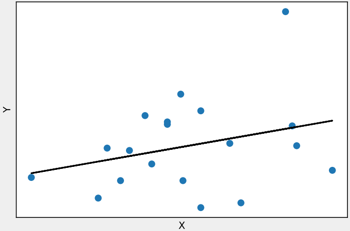
We assume the following model

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

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

Linear Regression

- We can also write $\mathbf{Y} \sim \mathrm{Normal}(f(\mathbf{X}), \sigma^2)$
- 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 $i = 1, \ldots, n$:
    - Select some value for x_i
    - Draw ϵ_i from $\mathrm{Normal}(0, \sigma^2)$
    - 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 $y = (x_i)_{i=1}^n$

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

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

- 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.

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

  assuming $\mathrm{pr}(\theta)$ is constant[1]

- This result is not specific to linear regression models

[1] A uniform prior $\mathrm{pr}(\theta)$ is called *improper prior* when θ is a continuous variable, because $\mathrm{pr}(\theta)$ does not integrate to one

Linear Regression - Parameter estimation

- Furthermore, we have

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

- In the last step we took advantage of the fact that the distribution of our covariates x does not depend on the parameters θ, which are the slope and intercept of the linear function

- In fact, we do not have do assume a particular distribution for our covariates!

Linear Regression - OLS

- Plugging in our normal distribution we arrive at

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

Linear Regression - OLS

- 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

$$x = \begin{bmatrix} 1 \\ \tilde{x} \end{bmatrix}, \qquad \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$

  i.e. x is a vector where the first component is always 1

- This definition allows to write

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

Linear Regression - Generalization

- Adding additional predictors is now very simple

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

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

- It follows that

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

Linear Regression - Notation

- In general, we have n observations and p predictors
- For the ith 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

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(2)} & \dots & x_1^{(p)} \\ 1 & x_2^{(2)} & \dots & x_2^{(p)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n^{(2)} & \dots & x_n^{(p)} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

- 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. $\mathrm{rank}(X) = \min(n, p)$

Linear Regression - OLS

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

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

Differentiation with respect to θ and solving for the roots leads to:

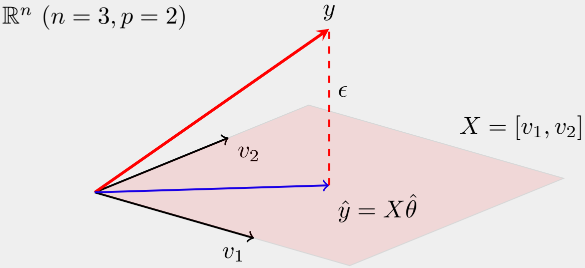
$$\begin{aligned} \Rightarrow \qquad \hat{\theta} &= (X^\top X)^{-1} X^\top y \\ &= X^\top y \qquad\qquad \text{if } X^\top X = I \end{aligned}$$

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

Linear Regression - OLS Projection

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

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



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

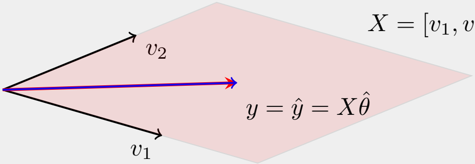
[1] [Hastie et al., 2009]

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 ith column of X

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

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



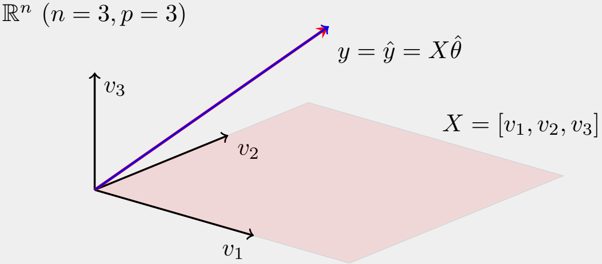
If y is already inside the plane, we obtain $\epsilon = 0$

[1] [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 ith column of X

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


$\mathbb{R}^n$ $(n = 3, p = 3)$
$y = \hat{y} = X\hat{\theta}$
v_3
$X = [v_1, v_2, v_3]$
v_2
v_1

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

[1] [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 $\left\|y - X\hat{\theta}\right\|_2^2 = 0$!

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 $\left\|y - X\hat{\theta}\right\|_2^2 = 0$!

- 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 $\left\|y - X\hat{\theta}\right\|_2^2 = 0$!

- 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 $\left\|y - X\hat{\theta}\right\|_2^2 = 0$!

- 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 θ with minimal length, i.e. the minimum ℓ_2-norm solution[2]

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

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

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

where $(X^\top X)^+$ Moore-Penrose pseudoinverse[3] of $X^\top X$.

[2] Common practice for training neural networks

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

Linear Regression - Ridge Regression

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 λ is called the *regularization strength* or *penalty*. Note that $\|\theta\|_2^2 = \sum_{i=2}^n \theta_i^2$, i.e. θ_1 is not constrained

- There exists an analytical solution to the ridge estimate:

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

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

[3] Convex optimization: [Boyd and Vandenberghe, 2004]

Linear Regression - Ridge Regression

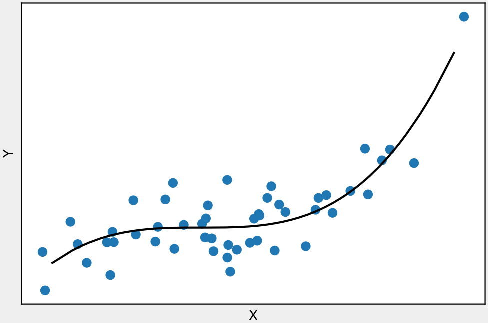
- For $\lambda \to \infty$ the estimate $\lambda\hat{\theta}(\lambda)$ converges to the componentwise regression estimator
- For $\lambda \to 0$ the estimate $\hat{\theta}(\lambda)$ converges to the minimum ℓ_2-norm OLS solution[4]
- The penalty $\lambda \|\theta\|_2^2$ can be interpreted as a Gaussian prior
- Ridge regression is useful when $n < p$ and $n \geq p$

[4] $A + \lambda I$ is invertible even for very small λ. 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

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

More generally, we write

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

where $\phi : \mathbb{R}^p \to \mathbb{R}^{p'}$ 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 ϕ 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.

Kernel Regression

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 $\theta \in \operatorname{span}(\phi(x_1), \ldots, \phi(x_n)) \subset \mathbb{R}^p$. It follows that

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

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

Kernel Regression

Definition: Kernel function

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

$$\kappa(x_i, x_j) = \phi(x_i)^\top \phi(x_j)$$

$K = (\kappa(x_i, x_j))_{x_i \in \mathcal{X}, x_j \in \mathcal{X}}$ is called the kernel matrix.

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

Example Kernels

- Linear kernel

$$\kappa(x_i, x_j) = x_i^\top x_j, \text{ where } \phi(x) = x$$

- Polynomial kernel

$$\kappa(x_i, x_j) = (x_i^\top x_j + 1)^d$$

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

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

- Radial basis function (RBF) kernel

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

where the feature space has infinite dimensions

Predictions

Let x_{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 θ and η the parameters of a (kernel) regression model
- The parameters of a kernel function (e.g. σ^2 for the RBF kernel) or the regularization strength λ 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 ℓ_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 ϕ cannot be computed
- No feature selection possible (ℓ_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[5]: Define a mapping $\xi : \mathcal{X} \to \mathbb{R}^q$ with $q \ll p$ such that

$$\kappa(x_i, x_j) = \phi(x_i)^\top \phi(x_j) \approx \xi(x_i)^\top \xi(x_j)$$

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

How do we compute ξ?

[5] [Rahimi et al., 2007]

Random Features

Bochner's theorem

A continuous shift-invariant kernel $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{C}$ with $\kappa(x_i, x_j) = \kappa(x_i - x_j)$ is positive definite iff there exists a non-negative measure μ such that

$$\kappa(x_i - x_j) = \int_{\mathbb{R}^d} \exp\left(i\omega^\top (x_i - x_j)\right) \mathrm{d}\mu(\omega)$$
$$= \mathbb{E}_\omega \exp(i\omega^\top (x_i - x_j)) = \mathbb{E}_\omega \exp(i\omega^\top x_i) \exp(i\omega^\top x_j)^* \,.$$

I.e. the kernel κ is the (inverse) Fourier transform of μ.

When both κ and μ are real-valued then

$$\kappa(x_i - x_j) = \mathbb{E}_\omega \cos(\omega^\top (x_i - x_j))$$

[5] x^* is the complex conjugate of x and remember that $exp(ix)^* = exp(-ix)$

Monte Carlo approximation

Let μ be a distribution and ω a random variable with distribution μ. From the law of large numbers it follows that

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

where $\omega_1, \ldots \omega_q$ are independent samples from μ.

Monte Carlo approximation of the Fourier integral

$$\begin{aligned} \omega_k &\overset{i.i.d.}{\sim} \mu \\ \kappa(x_i, x_j) &\approx \tfrac{1}{q}\textstyle\sum_{k=1}^{q} \exp(i\omega_k^\top(x_i - x_j)) = \xi(x_j)^*\xi(x_i) \end{aligned}$$

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

Random Features

In practice: We know the kernel κ and must derive the measure μ. Afterwards, we can approximate κ by drawing q samples ω_k from μ and map x into feature space using

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

The measure μ is given by the Fourier transform of κ with density

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

Random Features

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

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

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

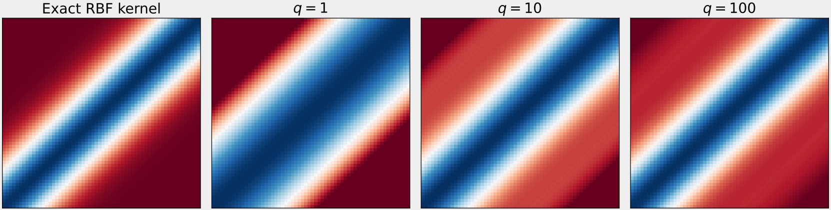
$$f_\mu(\omega) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{\left\|\omega\right\|_2^2}{2\sigma^2}\right)$$

Since κ and μ are real, we have

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

[5] $\cos(x_i - x_j) = \cos(x_i)\cos(x_j) + \sin(x_i)\sin(x_j)$

Random Features



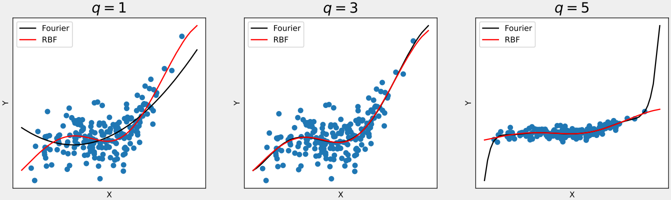
Exact RBF kernel

$q = 1$

$q = 10$

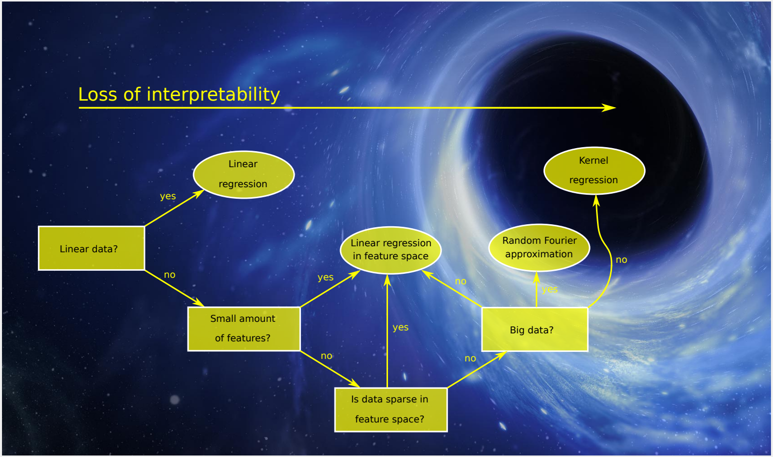
$q = 100$

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


Loss of interpretability
Linear regression
yes
Linear data?
no
Small amount of features?
yes
Linear regression in feature space
no
yes
Is data sparse in feature space?
no
Big data?
no
yes
Random Fourier approximation
no
Kernel regression

[5] 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.