TECHNISCHE UNIVERSITÄT DARMSTADT

Statistical Machine Learning

Lecture 5: Regression

Marcus Rohrbach
Department of Computer Science
TU Darmstadt

Summer Term 2025

Today's Objectives

- Extending supervised learning to continuous labels $\mathbf{y}$:
  - Learning continuous mappings $\mathbf{y} = f(\mathbf{x})$
- Covering topics:
  - Linear Regression and its interpretations
  - Deriving Linear Regression from Maximum Likelihood Estimation
  - What is overfitting?
  - Bayesian Linear Regression

Outline

1. Introduction to Linear Regression
2. Maximum Likelihood Approach to Regression
3. Bias and Variance in Linear Regression
4. Bayesian Linear Regression
5. Wrap-Up

Outline

1. Introduction to Linear Regression

2. Maximum Likelihood Approach to Regression

3. Bias and Variance in Linear Regression

4. Bayesian Linear Regression

5. Wrap-Up

TECHNISCHE UNIVERSITÄT DARMSTADT

Supervised Learning Taxonomy

- The task is to learn a mapping f from input to output

$$f : I \rightarrow O, \quad \mathbf{y} = f(\mathbf{x}; \theta)$$

	Regression	Classification
Input $\mathbf{x} \in I$	(images, text, sensor measurements, ...)	
Output $\mathbf{y} \in O$	continuous set O	discrete, finite set O
Deterministic	$\mathbf{y} = f(\mathbf{x}, \theta)$	
Probabilistic	$p(\mathbf{y}\|\mathbf{x}; \theta)$	

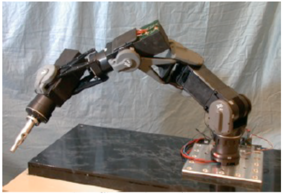
- Today: Regression for continuous labels $\mathbf{y} \in O$

TECHNISCHE UNIVERSITÄT DARMSTADT

Motivation

- You want to predict the torques y of a robot arm

$$
\begin{aligned}
y &= l\ddot{q} - \mu\dot{q} + mlg\sin(q) \\
&= \begin{bmatrix} \ddot{q} & \dot{q} & \sin(q) \end{bmatrix} \begin{bmatrix} l & -\mu & mlg \end{bmatrix}^\intercal \\
&= \phi(\mathbf{x})^\intercal \boldsymbol{\theta}
\end{aligned}
$$



- Can we do this with a data set?

$$
\mathcal{D} = \left\{ (\mathbf{x}_i, y_i) \,\middle|\, i = 1 \cdots n \right\}
$$

- A linear regression problem!

TECHNISCHE UNIVERSITÄT DARMSTADT

Least Squares Linear Regression

- Given pairs of training data points and associated function values $(\mathbf{x}_i, y_i)$

$$\mathcal{X} = \left\{\mathbf{x}_1, \ldots, \mathbf{x}_n | \mathbf{x_i} \in \mathbb{R}^d\right\}$$
$$\mathcal{Y} = \{y_1, \ldots, y_n | y_i \in \mathbb{R}\}$$

  - Note: here we only do the case $y_i \in \mathbb{R}$. In general y_i can have more than one dimension, i.e., $y_i \in \mathbb{R}^k$ for some positive integer k

- Start with linear regressor

$$\mathbf{x}_i^\mathsf{T}\mathbf{w} + w_0 = y_i \quad \forall i = 1, \ldots, n$$

  - One linear equation for each training data point/label pair
  - Exactly the same basic setup as for least-squares classification! Only the values are continuous

TECHNISCHE UNIVERSITÄT DARMSTADT

Least Squares Linear Regression

$$\mathbf{x}_i^\top \mathbf{w} + w_0 = y_i \quad \forall i = 1, \ldots, n$$

- **Step 1**: Define

$$\hat{\mathbf{x}}_i = \begin{pmatrix} \mathbf{x}_i \\ 1 \end{pmatrix} \quad \hat{\mathbf{w}} = \begin{pmatrix} \mathbf{w} \\ w_0 \end{pmatrix}$$

- **Step 2**: Rewrite

$$\hat{\mathbf{x}}_i^\top \hat{\mathbf{w}} = y_i \quad \forall i = 1, \ldots, n$$

- **Step 3**: Matrix-vector notation

$$\hat{\mathbf{X}}^\top \hat{\mathbf{w}} = \mathbf{y}$$

  - where $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_n]$ (each $\hat{\mathbf{x}}_i$ is a vector) and $\mathbf{y} = [y_1, \ldots, y_n]^\top$

TECHNISCHE UNIVERSITÄT DARMSTADT

Least Squares Linear Regression

- **Step 4**: Find the least squares solution

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \left\| \hat{\mathbf{X}}^\intercal \mathbf{w} - \mathbf{y} \right\|^2$$

$$\nabla_{\mathbf{w}} \left\| \hat{\mathbf{X}}^\intercal \mathbf{w} - \mathbf{y} \right\|^2 = \mathbf{0}$$

$$\hat{\mathbf{w}} = \left(\hat{\mathbf{X}} \hat{\mathbf{X}}^\intercal \right)^{-1} \hat{\mathbf{X}} \mathbf{y}$$

- A closed form solution!

TECHNISCHE UNIVERSITÄT DARMSTADT

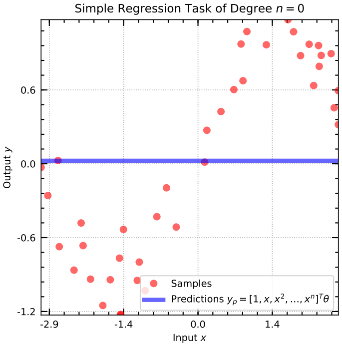
Least Squares Linear Regression

$$\hat{\mathbf{w}} = \left(\hat{\mathbf{X}}\hat{\mathbf{X}}^\intercal\right)^{-1}\hat{\mathbf{X}}\mathbf{y}$$

- Where is the costly part of this computation?
  - The inverse is a $\mathbb{R}^{D\times D}$ matrix
  - Naive inversion takes $O\left(D^3\right)$, but better methods exist
- What can we do if the input dimension D is too large?
  - Gradient descent
  - Work with fewer dimensions

Limitations

- So far $y_i = \mathbf{x}_i^T \mathbf{w} + \mathbf{w}_0$
- Can we represent arbitrary polynomials?

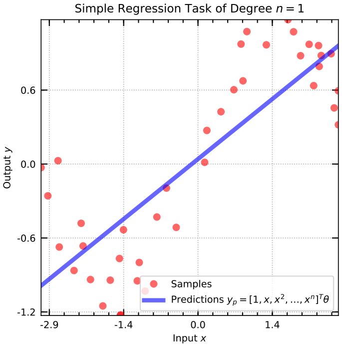


- No, only
  - constant mappings for $\mathbf{w} = 0$

TECHNISCHE UNIVERSITÄT DARMSTADT

Limitations

- So far $y_i = \mathbf{x}_i^T \mathbf{w} + \mathbf{w}_0$
- Can we represent arbitrary polynomials?
- No, only
  - constant mappings for $\mathbf{w} = 0$, or
  - linear mappings



TECHNISCHE UNIVERSITÄT DARMSTADT

Polynomial Regression

- How can we fit arbitrary polynomials using least-squares regression?
  - We introduce a feature transformation as before

$$y(\mathbf{x}) = \mathbf{w}^{\intercal}\phi(\mathbf{x}) = \sum_{i=0}^{M} w_i \phi_i(\mathbf{x})$$

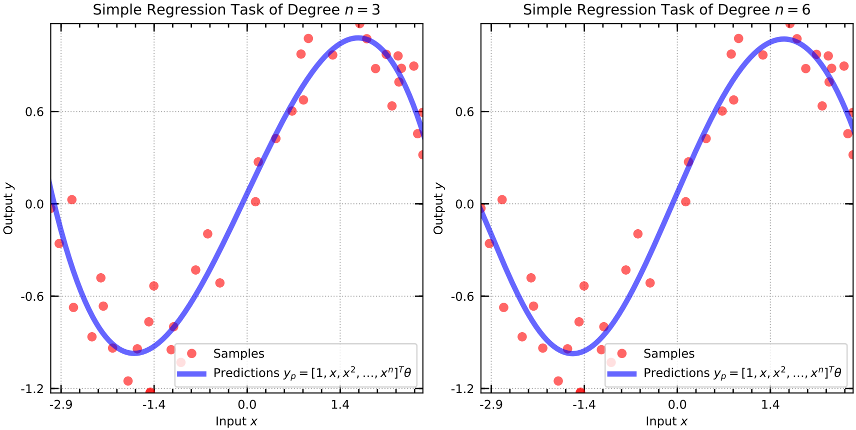
  - Assume $\phi_0(\mathbf{x}) = 1$
  - $\phi_i(.)$ are called the basis functions or features
  - Still a linear model in the parameters $\mathbf{w}$
- E.g. fitting a cubic polynomial

$$\phi(x) = \left(1, x, x^2, x^3\right)^{\intercal}$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Polynomial Regression

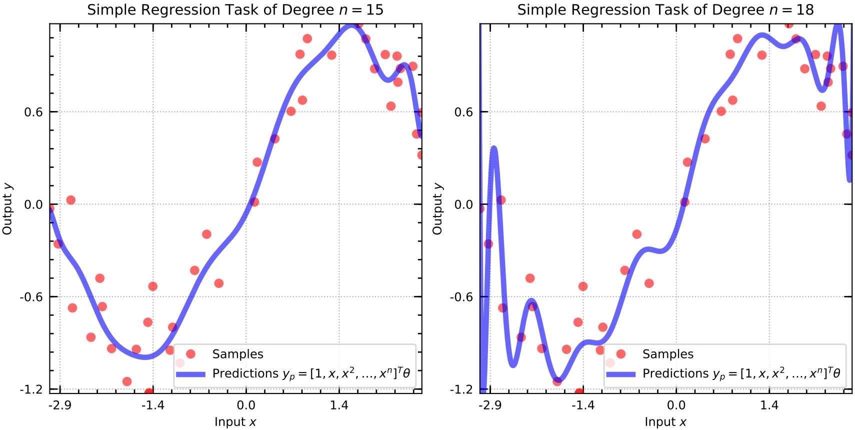
- Polynomial of degree $n = 3, 6$



- Fits data well

Polynomial Regression

- Polynomial of degree $n = 15, 18$



- Massive overfitting

TECHNISCHE UNIVERSITÄT DARMSTADT

Mechanical Interpretation

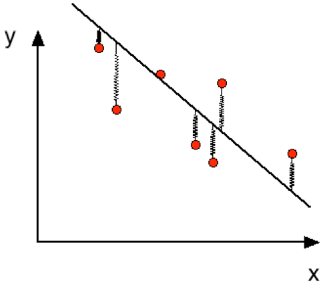
- Potential energy of spring i

$$E_{\text{pot}} = \frac{1}{2} k \Delta y_i^2,$$
$$\Delta y_i = y_i - f(\mathbf{x_i})$$

- System's total potential energy

$$E_{\text{tot}} = \frac{1}{2} k \sum_i \Delta y_i^2$$

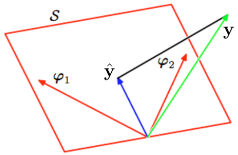
- Goal: Minimize E_{tot} by fitting $f(\mathbf{x_i})$



TECHNISCHE UNIVERSITÄT DARMSTADT

Geometric Interpretation

- Let's consider N data points $(t_i, \mathbf{x}_i)$
- The axes t_i define an N-dimensional space
- The $M < N$ features $\phi_j(\mathbf{x_i})$ span a linear sub-space of dimensionality M
- $\mathbf{y} = [f(\mathbf{x_1}), ..., f(\mathbf{x_N})]$ is a vector in the M-dimensional sub-space
- Least-squares regression finds $\mathbf{y}$ as the orthogonal projection of the data $\mathbf{t}$ into the sub-space.



Outline

1. Introduction to Linear Regression

2. Maximum Likelihood Approach to Regression

3. Bias and Variance in Linear Regression

4. Bayesian Linear Regression

5. Wrap-Up

TECHNISCHE UNIVERSITÄT DARMSTADT

Probabilistic Regression

- **Assumption 1**: Our target function values are generated by adding noise to the function estimate

$$y = f(\mathbf{x}, \mathbf{w}) + \epsilon$$

  - y - target function value; f - regression function; $\mathbf{x}$ - input value; $\mathbf{w}$ - weights or parameters; ϵ - noise

- **Assumption 2**: The noise is a random variable that is Gaussian distributed

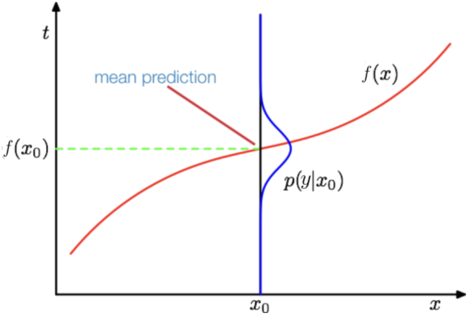
$$\epsilon \sim \mathcal{N}\left(\epsilon; 0, \sigma^2\right)$$

$$\Rightarrow \quad p\left(y \,\middle|\, \mathbf{x}; \mathbf{w}, \beta\right) = \mathcal{N}\left(y \,\middle|\, \mathbf{x}; f(\mathbf{x}, \mathbf{w}), \sigma^2\right)$$

  - $f(\mathbf{x}, \mathbf{w})$: mean; σ: standard deviation; $\beta = 1/\sigma^2$: precision

- Note that y is now a random variable with underlying probability distribution $p\left(y \,\middle|\, \mathbf{x}; \mathbf{w}, \sigma\right)$

Probabilistic Regression



TECHNISCHE UNIVERSITÄT DARMSTADT

Probabilistic Regression

- **Given**
  - Training input data points $\mathcal{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$
  - Associated function values $\mathcal{Y} = [y_1, \ldots, y_n]^\intercal$
- **Conditional likelihood** (assuming the data is i.i.d.)

$$p\left(\mathbf{y} \middle| \mathbf{X}; \mathbf{w}, \beta\right) = \prod_{i=1}^{n} \mathcal{N}\left(y_i \middle| \mathbf{x}_i; f\left(\mathbf{x}_i, \mathbf{w}\right), \sigma^2\right)$$

(with linear model)

$$= \prod_{i=1}^{n} \mathcal{N}\left(y_i \middle| \mathbf{x}_i; \mathbf{w}^\intercal \phi\left(\mathbf{x}_i\right), \sigma^2\right)$$

  - $\mathbf{w}^\intercal \phi\left(\mathbf{x}_i\right)$ is the generalized linear regression function
- Maximize the likelihood w.r.t. (with respect to) $\mathbf{w}$ and σ^2

TECHNISCHE UNIVERSITÄT DARMSTADT

Maximum Likelihood Regression

- Simplify using the **log**-likelihood

$$
\begin{aligned}
\log p\left(\mathbf{y} \middle| \mathbf{X}; \mathbf{w}, \sigma\right) &= \sum_{i=1}^{n} \log \mathcal{N}\left(y_i \middle| \mathbf{x}_i; \mathbf{w}^\mathsf{T}\phi\left(\mathbf{x}_i\right), \sigma^2\right) \\
&= \sum_{i=1}^{n} \left[\log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) - \frac{1}{2\sigma^2}\left(y_i - \mathbf{w}^\mathsf{T}\phi\left(\mathbf{x}_i\right)\right)^2\right] \\
&= -n\log\sigma - \frac{n}{2}\log\left(2\pi\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \mathbf{w}^\mathsf{T}\phi\left(\mathbf{x}_i\right)\right)^2
\end{aligned}
$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Maximum Likelihood Regression

- Gradient w.r.t. $\mathbf{w}$

$$
\begin{aligned}
\nabla_{\mathbf{w}} \log p\left(\mathbf{y} \,\middle|\, \mathbf{X}; \mathbf{w}, \sigma\right) &= \mathbf{0} \\
\Rightarrow \quad -\frac{1}{\sigma^2} \sum_{i=1}^{n} \left(y_i - \mathbf{w}^\intercal \phi\left(\mathbf{x}_i\right)\right) \phi\left(\mathbf{x}_i\right) &= \mathbf{0}
\end{aligned}
$$

- Define

$$
\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}, \quad \boldsymbol{\Phi} = \begin{bmatrix} | & & | \\ \phi\left(\mathbf{x}_1\right) & \dots & \phi\left(\mathbf{x}_n\right) \\ | & & | \end{bmatrix}
$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Maximum Likelihood Regression

$$\begin{aligned}\sum_{i=1}^{n} y_i \phi(\mathbf{x}_i) &= \left[\sum_{i=1}^{n} \phi(\mathbf{x}_i)\, \phi(\mathbf{x}_i)^\intercal\right] \mathbf{w} \\ \boldsymbol{\Phi}\mathbf{y} &= \boldsymbol{\Phi}\boldsymbol{\Phi}^\intercal \mathbf{w} \qquad \text{(Matrix notation)} \\ \mathbf{w}_{\mathrm{ML}} &= \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^\intercal\right)^{-1} \boldsymbol{\Phi}\mathbf{y}\end{aligned}$$

- The same result as in least squares regression!

Maximum Likelihood Regression

- We obtain the same **w** as with least squares regression
  - Least-squares is equivalent to assuming the targets are Gaussian distributed with fixed noise β
  - Note: The least squares method is not distribution-free!
- However, the Maximum Likelihood approach is much more powerful!
  - We can also estimate β

$$\sigma_{\mathrm{ML}}^2 = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \mathbf{w}_{\mathrm{ML}}^{\mathsf{T}} \phi\left(\mathbf{x}_i\right) \right)^2$$

  - We can gauge the uncertainty of our estimate!

TECHNISCHE UNIVERSITÄT DARMSTADT

Loss Functions in Regression

- Given a new data point $\mathbf{x}_t$, in least squares regression the function value is $y_t = \hat{\mathbf{x}}_t^\intercal \hat{\mathbf{w}}$
- But in maximum likelihood regression we have a probability distribution over the function value $p\left(y \,\middle|\, \mathbf{x}; \mathbf{w}, \sigma\right)$
- How do we actually estimate a function value y_t for a new data point $\mathbf{x}_t$?
- We need a loss function, just as in the classification case

$$\begin{aligned} L : \mathbb{R} \times \mathbb{R} \quad &\rightarrow \quad \mathbb{R}^+ \\ (y_t, f(\mathbf{x}_t)) \quad &\rightarrow \quad L(y_t, f(\mathbf{x}_t)) \end{aligned}$$

Loss Functions in Regression

TECHNISCHE UNIVERSITÄT DARMSTADT

- Minimize the expected loss

$$\mathbb{E}_{\mathbf{x},y\sim p(\mathbf{x},y)}[L] = \int\int L(y, f(\mathbf{x}))\, p(\mathbf{x}, y)\, \mathrm{d}\mathbf{x}\mathrm{d}y$$

- Simplest case: squared loss

$$L(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$$

$$\mathbb{E}_{\mathbf{x},y\sim p(\mathbf{x},y)}[L] = \int\int (y - f(\mathbf{x}))^2\, p(\mathbf{x}, y)\, \mathrm{d}\mathbf{x}\mathrm{d}y$$

$$\frac{\partial\mathbb{E}[L]}{\partial f(\mathbf{x})} = -2\int (y - f(\mathbf{x}))\, p(\mathbf{x}, y)\, \mathrm{d}y = 0$$

$$\int y p(\mathbf{x}, y)\mathrm{d}y = f(\mathbf{x})\int p(\mathbf{x}, y)\, \mathrm{d}y$$

Loss Functions in Regression

$$\int y p(\mathbf{x}, y)\, \mathrm{d}y = f(\mathbf{x}) \int p(\mathbf{x}, y)\, \mathrm{d}y$$

$$\int y p(\mathbf{x}, y)\, \mathrm{d}y = f(\mathbf{x})\, p(\mathbf{x})$$

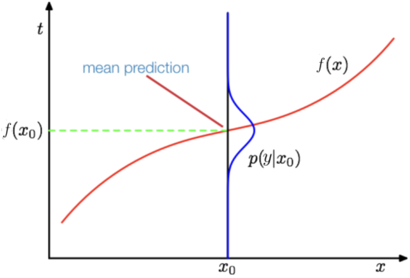
$$f(\mathbf{x}) = \int y \frac{p(\mathbf{x}, y)}{p(\mathbf{x})} \mathrm{d}y = \int y p(y \mid \mathbf{x})\, \mathrm{d}y$$

$$f(\mathbf{x}) = \mathbb{E}_{y \sim p(y|\mathbf{x})}[y] = \mathbb{E}\left[y \,\middle|\, \mathbf{x}\right]$$

- Under squared loss, the optimal regression function is the mean $\mathbb{E}\left[y \,\middle|\, \mathbf{x}\right]$ of the posterior $p(y \mid \mathbf{x})$
- It is also called mean prediction

TECHNISCHE UNIVERSITÄT DARMSTADT

Loss Functions in Regression



- For our generalized linear regression function

$$f(\mathbf{x}) = \int y \mathcal{N}\left(y \middle| \mathbf{w}^\top \phi(\mathbf{x}), \sigma^2\right) \mathrm{d}y = \mathbf{w}^\top \phi(\mathbf{x})$$

Outline

1. Introduction to Linear Regression

2. Maximum Likelihood Approach to Regression

3. Bias and Variance in Linear Regression

4. Bayesian Linear Regression

5. Wrap-Up

TECHNISCHE UNIVERSITÄT DARMSTADT

Bias and Variance in Linear Regression

- So far, we have assumed that $f(\mathbf{x}, \mathbf{w}) = \phi(\mathbf{x})^T \mathbf{w}$ depends only on the weights.
- Let's assume that we have fitted the parameters $\mathbf{w}$ of the model $f_{\mathcal{D}}(\mathbf{x})$ on data $\mathcal{D}$
  - How can we assess the quality of the model f, e.g., the choice of the polynomial degree of ϕ?
- We estimate the bias and variance of the regression model

TECHNISCHE UNIVERSITÄT DARMSTADT

Bias and Variance in Linear Regression

Notation:

- The training data $\mathcal{D}$ was generated by $y(\mathbf{x})$
- We assume the following model to predict the data $\mathcal{D}$

$$y(\mathbf{x}_q) = f(\mathbf{x}_q) + \epsilon$$

with $E\{\epsilon\} = 0$ and $\mathrm{Var}\{\epsilon\} = \sigma_\epsilon^2$

- We denote the trained function estimator on data $\mathcal{D}$ by $\hat{f}_\mathcal{D}$
- We evaluate the **Expected Squared Error** for query $\mathbf{x}_q$ estimated from all possible data sets $\mathcal{D}$

$$L_{\hat{f}}(\mathbf{x}_q) = \mathbb{E}_{\mathcal{D},\epsilon}\left[\left(y(\mathbf{x}_q) - \hat{f}_\mathcal{D}(\mathbf{x}_q)\right)^2\right]$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Bias and Variance in Linear Regression

Expected Squared Error for query $\mathbf{x}_q$ estimated from all possible data sets $\mathcal{D}$

$$
\begin{aligned}
L_{\hat{f}}(\mathbf{x}_q) &= \mathbb{E}_{\mathcal{D},\epsilon}\left[\left(y(\mathbf{x}_q) - \hat{f}_{\mathcal{D}}(\mathbf{x}_q)\right)^2\right] \\
&= \mathbb{E}_{\mathcal{D},\epsilon}\left[\left(y(\mathbf{x}_q) - f(\mathbf{x}_q) + f(\mathbf{x}_q) - \hat{f}_{\mathcal{D}}(\mathbf{x}_q)\right)^2\right] \\
&= \mathbb{E}_{\mathcal{D},\epsilon}\left[\underbrace{\left(y(\mathbf{x}_q) - f(\mathbf{x}_q)\right)^2}_{=\epsilon^2} + \left(f(\mathbf{x}_q) - \hat{f}_{\mathcal{D}}(\mathbf{x}_q)\right)^2 + 2\underbrace{\left(y(\mathbf{x}_q) - f(\mathbf{x}_q)\right)\left(f(\mathbf{x}_q) - \hat{f}_{\mathcal{D}}(\mathbf{x}_q)\right)}_{=\epsilon\left(f(\mathbf{x}_q) - \hat{f}_{\mathcal{D}}(\mathbf{x})\right)}\right] \\
&= \color{blue}{\sigma_\epsilon^2} + \color{red}{\mathbb{E}_{\mathcal{D}}\left[\left(f(\mathbf{x}_q) - \hat{f}_{\mathcal{D}}(\mathbf{x}_q)\right)^2\right]}; \qquad \mathbb{E}_{\mathcal{D},\epsilon}\left[\epsilon\left(f(\mathbf{x}_q) - \hat{f}_{\mathcal{D}}(\mathbf{x_q})\right)\right] = 0
\end{aligned}
$$

- The loss can be decomposed into the model noise and the expected squared error between the regression model and $f(\mathbf{x_q})$

TECHNISCHE UNIVERSITÄT DARMSTADT

Bias and Variance in Linear Regression

- We shed some light on the model discrepancy term

$$\mathbb{E}_{\mathcal{D}}\left[\left(f(\mathbf{x}_q) - \hat{f}_{\mathcal{D}}(\mathbf{x}_q)\right)^2\right]$$

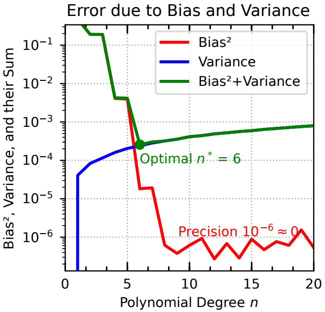
- using $\bar{\hat{f}}(\mathbf{x}_q) = \mathbb{E}_{\mathcal{D}}\left[\hat{f}_{\mathcal{D}}(\mathbf{x}_q)\right]$, we obtain

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\left[\left(f(\mathbf{x}_q) - \hat{f}_{\mathcal{D}}(\mathbf{x}_q)\right)^2\right] &= \mathbb{E}_{\mathcal{D}}\left[\left(f(\mathbf{x}_q) - \bar{\hat{f}}(\mathbf{x}_q) + \bar{\hat{f}}(\mathbf{x}_q) - \hat{f}_{\mathcal{D}}(\mathbf{x}_q)\right)^2\right] \\ &= \underbrace{\left(f(\mathbf{x}_q) - \bar{\hat{f}}(\mathbf{x}_q)\right)^2}_{=\mathrm{bias}^2\left[\hat{f}_{\mathcal{D}}(\mathbf{x}_q)\right]} + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(\bar{\hat{f}}(\mathbf{x}_q) - \hat{f}_{\mathcal{D}}(\mathbf{x}_q)\right)^2\right]}_{=\mathrm{var}\left[\hat{f}_{\mathcal{D}}(\mathbf{x}_q)\right]}\end{aligned}$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Bias-Variance Tradeoff

- (Total) **Bias** $\text{bias}^2\left[\hat{f}_{\mathcal{D}}\right] = \mathbb{E}_{\mathbf{x}_q}\left[\left(f(\mathbf{x}_q) - \mathbb{E}_{\mathcal{D}}\left[\hat{f}_{\mathcal{D}}(\mathbf{x}_q)\right]\right)^2\right]$
  - Structure error
  - Model $\hat{f}_{\mathcal{D}}(\mathbf{x}_q)$ cannot do better
- (Total) **Variance** $\text{var}\left[\hat{f}_{\mathcal{D}}(\mathbf{x}_q)\right] = \mathbb{E}_{\mathbf{x}_q,\mathcal{D}}\left[\left(\hat{f}_{\mathcal{D}}(\mathbf{x}_q) - \mathbb{E}_{\tilde{\mathcal{D}}}\left[\hat{f}_{\tilde{\mathcal{D}}}(\mathbf{x}_q)\right]\right)^2\right]$
  - Estimation error
  - Finite data sets will always have errors
- **Expected Total Error** $\propto$ Bias^2+Variance
  - You typically cannot minimize both

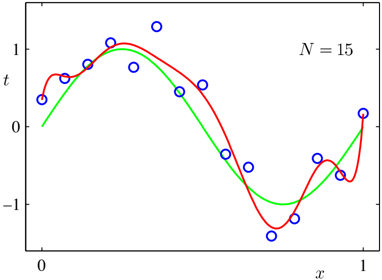


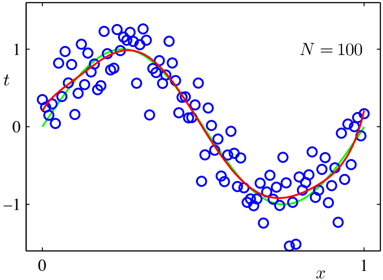
TECHNISCHE UNIVERSITÄT DARMSTADT

Are we Done with Regression?

Relatively little data leads to overfitting

Enough data leads to a good estimate





- What can we do to avoid overfitting in small data settings? Recall the density estimation class
- Use a prior distribution over the parameters $p(\mathbf{w})$

Outline

1. Introduction to Linear Regression

2. Maximum Likelihood Approach to Regression

3. Bias and Variance in Linear Regression

4. **Bayesian Linear Regression**

5. Wrap-Up

TECHNISCHE UNIVERSITÄT DARMSTADT

Bayesian Linear Regression

- We place a prior on the parameters **w** to tame the instabilities

$$p\left(\mathbf{w} \,\middle|\, \mathbf{X}, \mathbf{y}\right) \propto p\left(\mathbf{y} \,\middle|\, \mathbf{X}, \mathbf{w}\right) p\left(\mathbf{w}\right)$$

  - Parameter prior: $p\left(\mathbf{w}\right)$
  - Likelihood of targets under the data and parameters (as before): $p\left(\mathbf{y} \,\middle|\, \mathbf{X}, \mathbf{w}\right)$
  - Posterior over the parameters: $p\left(\mathbf{w} \,\middle|\, \mathbf{X}, \mathbf{y}\right)$

- Notice the VERY important difference: in this setting, you do not get a single value for the parameters anymore, but rather a probability distribution over the parameters

Bayesian Regression

- Simple idea: Put a Gaussian prior on **w**
- It will put a "soft" limit on the coefficients and thus avoid instabilities

$$\mathbf{w} \sim p_0(\mathbf{w}; \sigma_0) = \mathcal{N}\left(\mathbf{w}; \mathbf{0}, \sigma_0^2 \mathbf{I}\right)$$

  - We use a zero mean Gaussian to keep the derivation compact, but you can use another mean
- Zero mean and spherical covariance (given by the diagonal covariance matrix)
- The posterior becomes

$$\begin{aligned} p\left(\mathbf{w} \middle| \mathbf{X}, \mathbf{y}; \sigma_0, \sigma\right) &\propto p\left(\mathbf{y} \middle| \mathbf{X}, \mathbf{w}; \sigma\right) p(\mathbf{w}; \sigma_0) \\ &\propto p\left(\mathbf{y} \middle| \mathbf{X}, \mathbf{w}; \sigma\right) \mathcal{N}\left(\mathbf{w}; \mathbf{0}, \sigma_0^2 \mathbf{I}\right) \end{aligned}$$

Maximum A-Posteriori (MAP)

- First attempt to solve this problem: estimate $\mathbf{w}$ by maximizing the (log) posterior on the data $\mathbf{X}$

$$\begin{aligned}
\log p\left(\mathbf{w} \,\middle|\, \mathbf{X}, \mathbf{y}; \sigma_0, \sigma\right) &= \log p\left(\mathbf{y} \,\middle|\, \mathbf{X}, \mathbf{w}; \sigma\right) + \log \mathcal{N}\left(\mathbf{w}; \mathbf{0}, \sigma_0^2 \mathbf{I}\right) + \text{const} \\
&= \sum_{i=1}^{n} \log \mathcal{N}\left(y_i \,\middle|\, \mathbf{x}_i, \mathbf{w}; \mathbf{w}^\intercal \phi\left(\mathbf{x}_i\right) \sigma^2\right) \\
&\quad + \log \mathcal{N}\left(\mathbf{w}; \mathbf{0}, \sigma_0^2 \mathbf{I}\right) + \text{const} \\
&= -\frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(y_i - \mathbf{w}^\intercal \phi\left(\mathbf{x}_i\right)\right)^2 - \frac{1}{2\sigma_0^2} \mathbf{w}^\intercal \mathbf{w} + \text{const}
\end{aligned}$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Maximum A-Posteriori (MAP)

- Find optimal MAP parameters $\mathbf{w}_{\text{MAP}}$ by taking the gradient:

$$\nabla_{\mathbf{w}} \log p\left(\mathbf{w} \,\middle|\, \mathbf{X}, \mathbf{y}, \sigma_0, \sigma\right) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \left(y_i - \mathbf{w}^\intercal \phi\left(\mathbf{x}_i\right)\right) \phi\left(\mathbf{x}_i\right) - \frac{1}{\sigma_0^2} \mathbf{w} = \mathbf{0}$$

$$\Leftrightarrow \quad \sigma^{-2} \sum_{i=1}^{n} y_i \phi\left(\mathbf{x}_i\right) = \sigma^{-2} \left[\sum_{i=1}^{n} \phi\left(\mathbf{x}_i\right) \phi\left(\mathbf{x}_i\right)^\intercal\right] \mathbf{w} + \sigma_0^{-2} \mathbf{w}$$

$$\Leftrightarrow \quad \sigma^{-2} \sum_{i=1}^{n} y_i \phi\left(\mathbf{x}_i\right) = \left[\sigma^{-2} \sum_{i=1}^{n} \phi\left(\mathbf{x}_i\right) \phi\left(\mathbf{x}_i\right)^\intercal + \sigma_0^{-2}\right] \mathbf{w}$$

$$\Leftrightarrow \quad \sigma^{-2} \boldsymbol{\Phi} \mathbf{y} = \left(\sigma^{-2} \boldsymbol{\Phi} \boldsymbol{\Phi}^\intercal + \sigma_0^{-2} \mathbf{I}\right) \mathbf{w} \quad \text{(Matrix notation)}$$

$$\Leftrightarrow \quad \mathbf{w}_{\text{MAP}} = \left(\boldsymbol{\Phi} \boldsymbol{\Phi}^\intercal + \sigma^2 / \sigma_0^2 \mathbf{I}\right)^{-1} \boldsymbol{\Phi} \mathbf{y}$$

- What is the role of σ^2 / σ_0^2 in the expression?

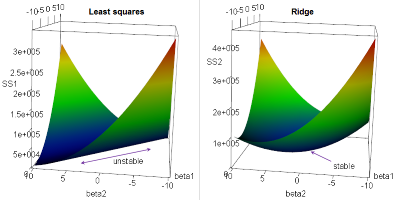
TECHNISCHE UNIVERSITÄT DARMSTADT

Maximum A-Posteriori (MAP)

$$\mathbf{w}_{\text{MAP}} = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathsf{T}} + \frac{\sigma^2}{\sigma_0^2}\mathbf{I} \right)^{-1} \boldsymbol{\Phi}\mathbf{y}$$

- The prior has the effect that it regularizes the pseudo-inverse
- Also called ridge regression

Intuition for the term "ridge", although these are not the historical reasons : If there is multi-collinearity, we get a "ridge" in the likelihood function. This in turn yields a long "valley" in the RSS. Ridge regression "fixes" the ridge. It adds a penalty that turns the ridge into a nice peak in likelihood space.



Regularized Least-squares Linear Regression as Maximum A-Posteriori (MAP)

- There is another way to look at the MAP result
- Let us add a regularization term to our objective from Least-squares Linear Regression

$$\mathbf{w} = \arg\min_{\mathbf{w}} \frac{1}{2} \left\| \boldsymbol{\Phi}^\intercal \mathbf{w} - \mathbf{y} \right\|^2 + \frac{\lambda}{2} \left\| \mathbf{w} \right\|^2$$

- Solving for **w** we get a new estimate

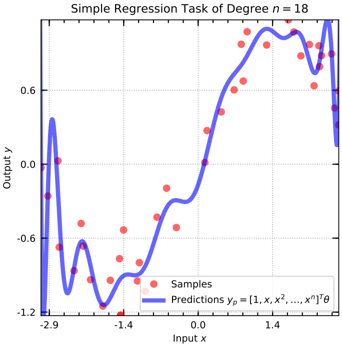
$$\mathbf{w} = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^\intercal + \lambda \mathbf{I} \right)^{-1} \boldsymbol{\Phi}\mathbf{y}$$

  - where $\lambda = \sigma^2 / \sigma_0^2$
- When you place a regularizer λ in least-squares linear regression, you are assuming the targets have Gaussian distributed noise, but also that your parameters are Gaussian distributed

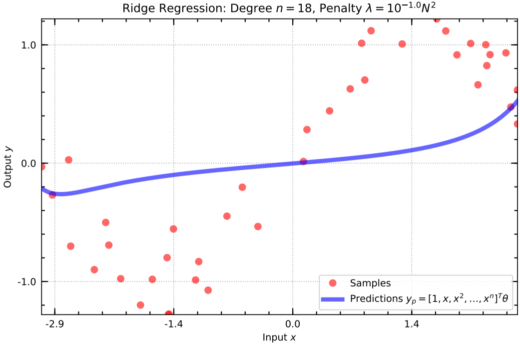
TECHNISCHE UNIVERSITÄT DARMSTADT

Bayesian Regression

- Polynomial of degree $n = 18$ with Gaussian prior on $\mathbf{w}$



Linear Regression



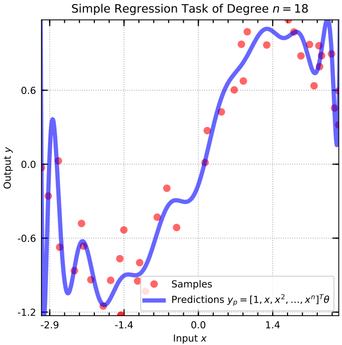
Ridge regression with $\lambda = 10^{-1}N$

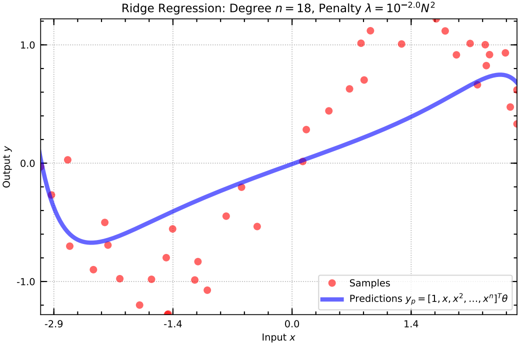
- $\lambda = \sigma^2/\sigma_0^2$ controls the complexity of the model and determines the degree of overfitting

TECHNISCHE UNIVERSITÄT DARMSTADT

Bayesian Regression

- Polynomial of degree $n = 18$ with Gaussian prior on $\mathbf{w}$





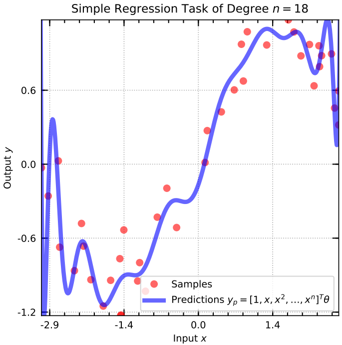
Linear Regression

Ridge regression with $\lambda = 10^{-2}N$

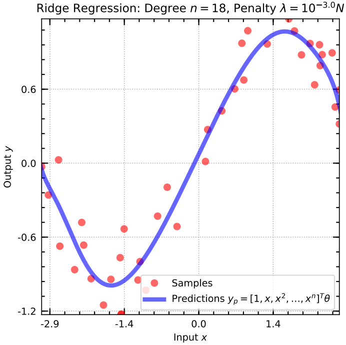
- $\lambda = \sigma^2/\sigma_0^2$ controls the complexity of the model and determines the degree of overfitting

Bayesian Regression

- Polynomial of degree $n = 18$ with Gaussian prior on $\mathbf{w}$



Linear Regression

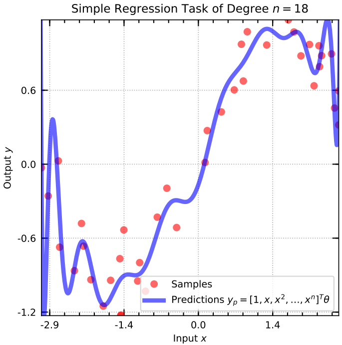


Ridge regression with $\lambda = 10^{-3}N$

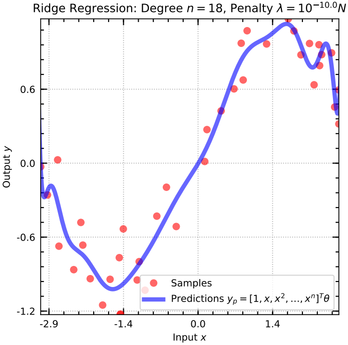
- $\lambda = \sigma^2/\sigma_0^2$ controls the complexity of the model and determines the degree of overfitting

Bayesian Regression

- Polynomial of degree $n = 18$ with Gaussian prior on $\mathbf{w}$



Linear Regression

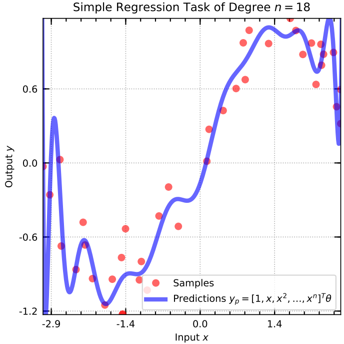


Ridge regression with $\lambda = 10^{-10}N$

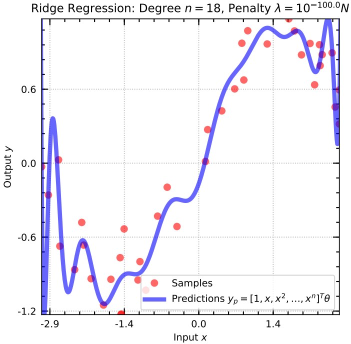
- $\lambda = \sigma^2/\sigma_0^2$ controls the complexity of the model and determines the degree of overfitting

Bayesian Regression

- Polynomial of degree $n = 18$ with Gaussian prior on $\mathbf{w}$



Linear Regression

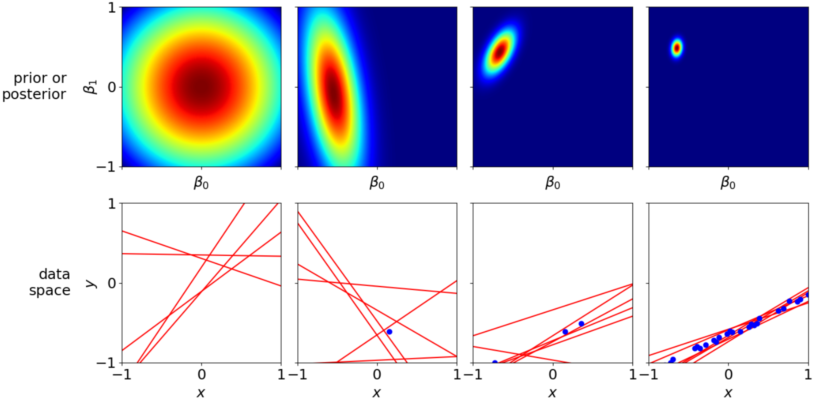


Ridge regression with $\lambda = 10^{-100}N$

- $\lambda = \sigma^2/\sigma_0^2$ controls the complexity of the model and determines the degree of overfitting

Bayesian Regression

- Posterior evolution with more data points



https://gregorygundersen.com/blog/2020/02/04/bayesian-linear-regression/

TECHNISCHE UNIVERSITÄT DARMSTADT

Full Bayesian Regression

- We can go further than MAP estimation
- Observation: We do not actually need to know **w**, all we want to do is to predict a function value based on the training data
- Idea: "Remove" **w** by marginalizing over it

$$p\left(y_t \,\middle|\, \mathbf{x}_t, \mathbf{X}, \mathbf{y}\right) = \int p\left(y_t, \mathbf{w} \,\middle|\, \mathbf{x}_t, \mathbf{X}, \mathbf{y}\right) \mathrm{d}\mathbf{w}$$

  - y_t - predicted value; $\mathbf{x}_t$ - test input; $\mathbf{X}$ - training data points; $\mathbf{y}$ - training function values

Full Bayesian Regression

$$\underbrace{p\left(y_t \,\middle|\, \mathbf{x}_t, \mathbf{X}, \mathbf{y}\right)}_{\text{predictive distribution}} = \int p\left(y_t, \mathbf{w} \,\middle|\, \mathbf{x}_t, \mathbf{X}, \mathbf{y}\right) \mathrm{d}\mathbf{w}$$

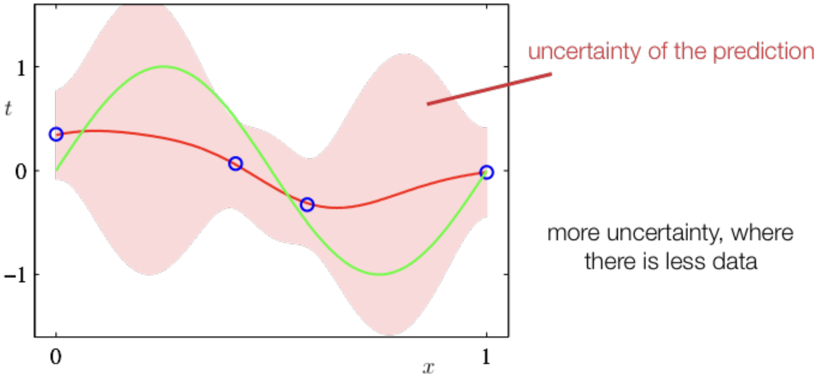
$$= \int p\left(y_t \,\middle|\, \mathbf{w}, \mathbf{x}_t, \mathbf{X}, \mathbf{y}\right) p\left(\mathbf{w} \,\middle|\, \mathbf{x}_t, \mathbf{X}, \mathbf{y}\right) \mathrm{d}\mathbf{w}$$

$$= \int \underbrace{p\left(y_t \,\middle|\, \mathbf{w}, \mathbf{x}_t\right)}_{\text{regression model}} \underbrace{p\left(\mathbf{w} \,\middle|\, \mathbf{X}, \mathbf{y}\right)}_{\text{posterior distribution}} \mathrm{d}\mathbf{w}$$

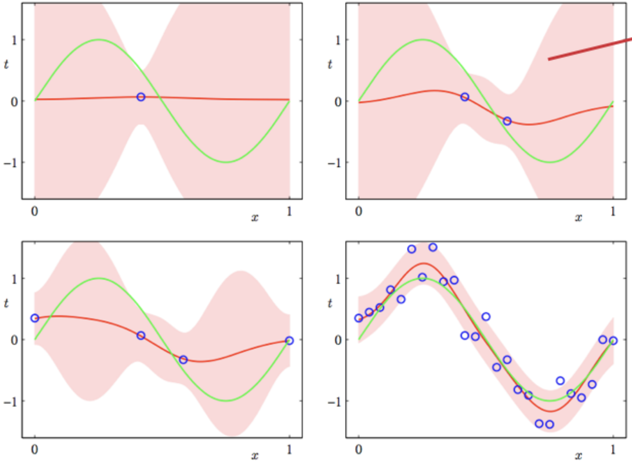
- For Gaussian distributions, this can be done in closed form, leading to so-called Gaussian Processes

Gaussian Processes - Quick Preview

- Essentially Kernelized Bayesian Ridge Regression is equivalent to Gaussian Processes. We will not cover them now, but here is a quick preview of what they can do



Gaussian Processes - Quick Preview



uncertainty of the prediction

more uncertainty, where there is less data

Outline

1. Introduction to Linear Regression

2. Maximum Likelihood Approach to Regression

3. Bias and Variance in Linear Regression

4. Bayesian Linear Regression

5. Wrap-Up

5. Wrap-Up

You know now:

- How to formulate a linear regression problem
- The different methods to perform linear regression: least-squares, maximum likelihood and maximum a-posteriori
- Derive the equations for the parameters using the different methods
- Why introducing a prior distribution over the parameters can combat overfitting
- The bias and variance tradeoff in maximum likelihood approaches

Self-Test Questions

- What is regression (in general) and linear regression (in particular)?
- What is the cost function of regression and how can I interpret it?
- What is overfitting?
- How can I derive a Maximum-Likelihood Estimator for Regression?
- Why are Bayesian methods important?
- What is MAP and how is it different to full Bayesian regression?

Homework

- Reading Assignment to dive deeper into this week's content
  - Lindholm ch. 3.1
  - Murphy ch. 7
  - Bishop ch. 3
- Reading Assignment to prepare the next lecture
  - Lindholm ch. 9
  - Bishop ch. 6.4
  - Murphy ch. 15