Photogrammetry & Robotics Lab

Machine Learning for Robotics and Computer Vision

Regression

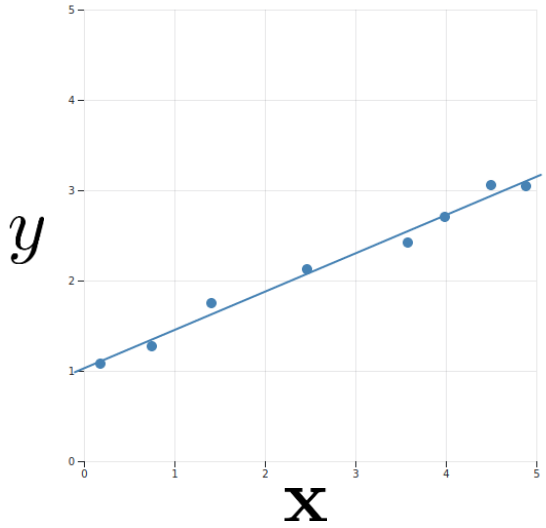
Jens Behley

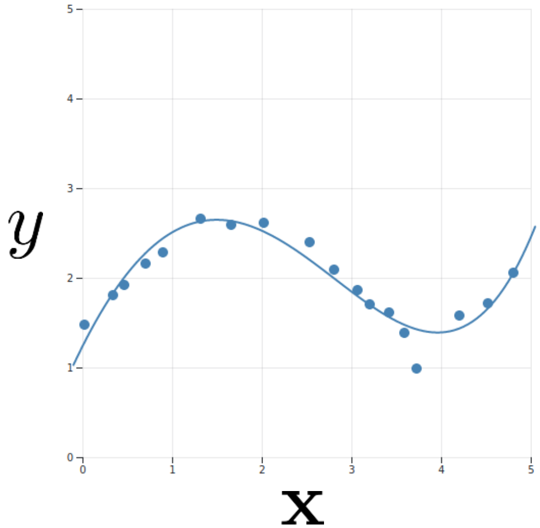
Recap: Last Lecture

- High-level overview of machine learning algorithms
- Main ingredients:
  1. **Data**
  2. **Model**
  3. **Learning**

- Discussed the importance of train, validation, and test set
- Discussed as an example **k-Nearest Neighbor** classification

Regression





- **Regression** is finding a function $f(\mathbf{x})$ that explains our targets $y \in \mathbb{R}$ for an input $\mathbf{x} \in \mathbb{R}^D$
- Assumption: we have noisy observations:

$$y_n = f(\mathbf{x}_n) + \epsilon \quad \text{with} \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

- We assume that we know σ^2 in advance.

Linear Regression

- Under this assumptions, this leads to the following probabilistic formulation

$$P(y|\mathbf{x}, \theta) = \mathcal{N}(y|f(\mathbf{x}), \sigma^2)$$

- In linear regression, we assume that parameters θ appear **linearly** in our model

$$f(\mathbf{x}) = \mathbf{x}^T\theta + \theta_0$$

- θ_0 is called **intercept** (or **bias**) that enables us to have also functions that do not pass through the origin

[MML Chap. 9.1]

Some Notation

- Training data given by

$$\mathcal{X}_{\text{train}} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n), \ldots, (\mathbf{x}_N, y_N)\}$$

with $\mathbf{x}_n = (x_1, \ldots, x_d, \ldots, x_D) \in \mathbb{R}^D$ and $y \in \mathbb{R}$

- To simplify notation, we will add $x_0 = 1$
- Which simplifies $f(\mathbf{x})$ as follows:

$$f(\mathbf{x}) = \mathbf{x}^T\theta + \theta_0 \xrightarrow{\mathbf{x} := (1, \mathbf{x})^T} f(\mathbf{x}) = \mathbf{x}^T\theta$$

- Define $\mathbf{x}_{1:N} := \mathbf{x}_1, \ldots, \mathbf{x}_N$ and $y_{1:N} := y_1, \ldots, y_N$

Probabilistic view

- Treat parameters θ as random variables
- We are interested in the posterior

$$P(\theta|\mathbf{x}_{1:N}, y_{1:N}) = \frac{P(y_{1:N}|\mathbf{x}_{1:N}, \theta)P(\theta)}{P(y_{1:N}|\mathbf{x}_{1:N})}$$

- As $P(y_{1:N}|\mathbf{x}_{1:N})$ is independent of θ, follows:

$$\underbrace{P(\theta|\mathbf{x}_{1:N}, y_{1:N})}_{\textbf{Posterior}} \propto \underbrace{P(y_{1:N}|\mathbf{x}_{1:N}, \theta)}_{\textbf{Likelihood}} \underbrace{P(\theta)}_{\textbf{Prior}}$$

- $P(y_{1:N}|\mathbf{x}_{1:N})$ is marginal likelihood/evidence

Common Parameter Estimation

- Learning is finding parameters of

$$P(\theta|\mathbf{x}_{1:N}, y_{1:N}) \propto P(y_{1:N}|\mathbf{x}_{1:N}, \theta) \quad P(\theta)$$

- Paradigms for parameter estimation:

1. Point estimate with uniform prior
   → Maximum Likelihood Estimation
2. Point estimate with given prior
   → Maximum A posteriori Estimation (MAP)
3. Determine posterior over the parameters
   → Bayesian Estimation

Maximum Likelihood Estimation

- Assuming a uniform prior, the posterior reduces to

$$P(\theta|\mathbf{x}_{1:N}, y_{1:N}) \propto P(y_{1:N}|\mathbf{x}_{1:N}, \theta)$$

- We want to find the parameters $\theta^\star$ that maximize the likelihood

$$\begin{aligned}\theta^\star &= \arg\max_\theta P(y_{1:N}|\mathbf{x}_{1:N}, \theta) \\ &= \arg\max_\theta \prod_{n=1}^{N} P(y_n|\mathbf{x}_n, \theta) \qquad \text{(i.i.d.)}\end{aligned}$$

[MML Chap. 8.1.1] [MML Chap. 9.2.1]

Negative Log-Likelihood

- As logarithm is a monotonically increasing function ➔ optimum of f ⇔optimum of log f

- With the negative log-transform, we now **minimize** accordingly:

$$\theta^\star = \arg\min_\theta - \log \prod_{n=1}^{N} P(y_n|\mathbf{x}_n, \theta)$$
$$= \arg\min_\theta \underbrace{- \sum_{n=1}^{N} \log P(y_n|\mathbf{x}_n, \theta)}_{\textbf{Negative Log-Likelihood}}$$

- Advantage: sum is numerical more stable then product of values in [0,1]!

[MML Chap. 9.2.1]

NLL for Linear Regression

- Inserting all terms for Linear Regression:

$$
\begin{aligned}
P(y_n|\mathbf{x}_n,\theta) &= \mathcal{N}(y_n|f(\mathbf{x}_n),\sigma^2) \\
&= \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{(y_n - f(\mathbf{x}_n))^2}{\sigma^2}\right) \\
&= \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{\left(y_n - \mathbf{x}_n^T\theta\right)^2}{\sigma^2}\right) \qquad (f(\mathbf{x}_n) = \mathbf{x}_n^T\theta)
\end{aligned}
$$

- It follows:

$$\log P(y_n|\mathbf{x}_n,\theta) = \text{const} - \frac{1}{2\sigma^2}\left(y_n - \mathbf{x}_n^T\theta\right)^2$$

- The NLL $\mathcal{L}(\theta)$ is then:

$$\mathcal{L}(\theta) := \frac{1}{2\sigma^2}\sum_{n=1}^{N}\left(y_n - \mathbf{x}_n^T\theta\right)^2$$

[MML Chap. 9.2.1]

Design matrix

- Let $\mathbf{X} \in \mathbb{R}^{N \times D}$ and $\mathbf{y} \in \mathbb{R}^{N}$ be defined as:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_N^T \end{pmatrix} \in \mathbb{R}^{N \times D} \qquad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} \in \mathbb{R}^{N}$$

- $\mathbf{X} \in \mathbb{R}^{N \times D}$ is called the **design matrix**
- Using these definitions, we can rewrite:

$$\mathcal{L}(\theta) := \frac{1}{2\sigma^2} \sum_{n=1}^{N} \left(y_n - \mathbf{x}_n^T \theta\right)^2$$

$$= \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\theta)^T (\mathbf{y} - \mathbf{X}\theta)$$

[MML Chap. 9.2.1]

Gradient of NLL

- Minimizing the NLL $\mathcal{L}(\theta)$ means that we have to find θ where gradient $\frac{d\mathcal{L}}{d\theta}$ is zero
- Gradient can be derived as follows:

$$\begin{aligned}
\frac{d\mathcal{L}}{d\theta} &= \frac{d}{d\theta}\left(\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\theta)^T(\mathbf{y}-\mathbf{X}\theta)\right) \\
&= \frac{1}{2\sigma^2}\frac{d}{d\theta}\left(\cancel{\mathbf{y}^T\mathbf{y}}\underline{-2\mathbf{y}^T\mathbf{X}\theta}+\underline{\theta^T\mathbf{X}^T\mathbf{X}\theta}\right) \\
&= \frac{1}{2\sigma^2}\left(\underline{-2\mathbf{y}^T\mathbf{X}}+\underline{2\theta^T\mathbf{X}^T\mathbf{X}}\right) \\
&= \frac{1}{\sigma^2}\left(-\mathbf{y}^T\mathbf{X}+\theta^T\mathbf{X}^T\mathbf{X}\right) \in \mathbb{R}^{1\times D}
\end{aligned}$$

[MML Chap. 9.2.1]

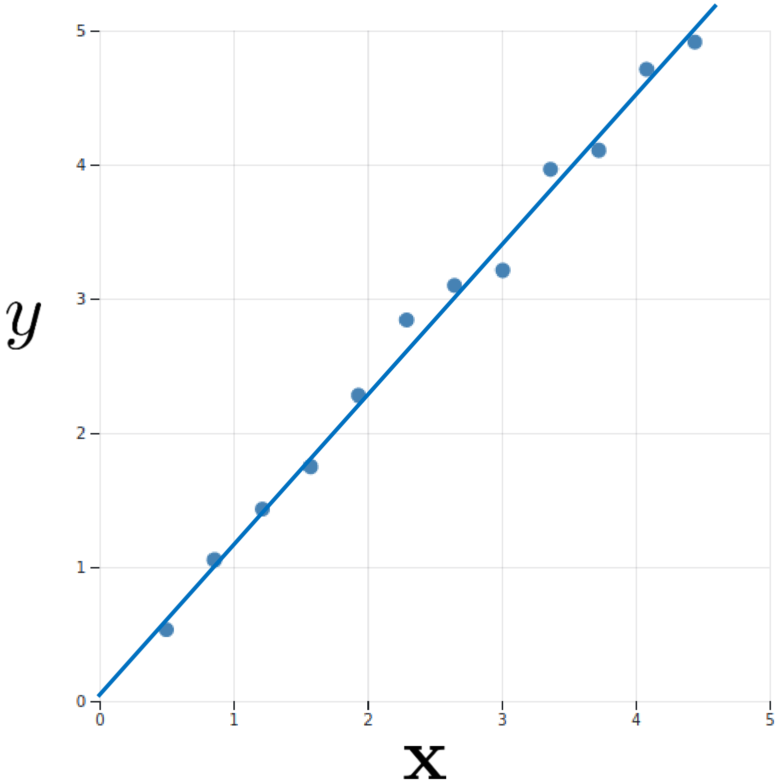
Maximum Likelihood Estimator

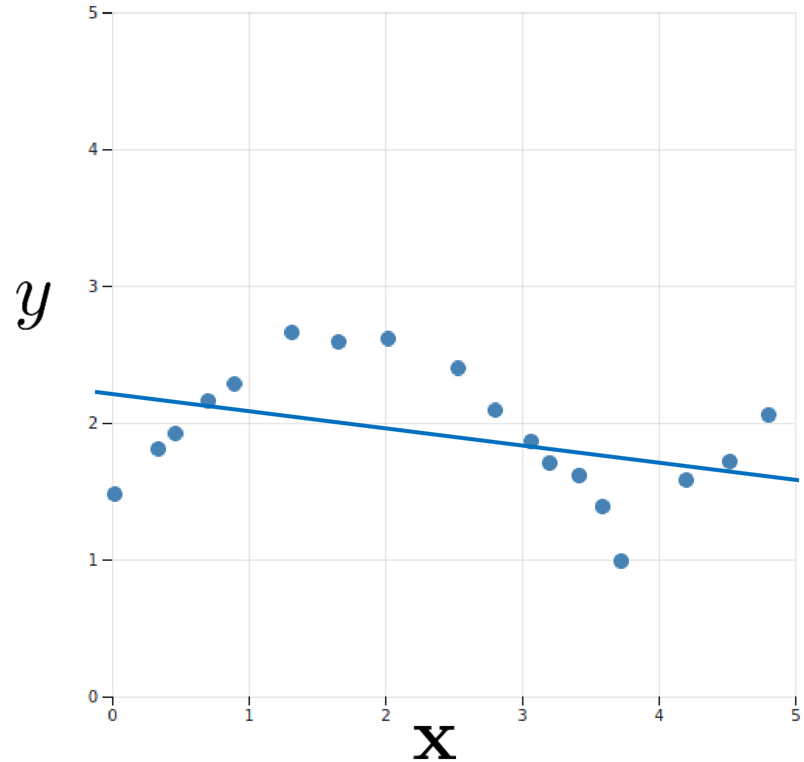
- Setting $\frac{d\mathcal{L}}{d\theta} = \mathbf{0}^T$ results in maximum likelihood parameters $\theta^\star$

$$
\begin{aligned}
\frac{d\mathcal{L}}{d\theta} = \mathbf{0}^T \quad &\Longleftrightarrow \quad \cancel{\frac{1}{\sigma^2}}\left(-\mathbf{y}^T\mathbf{X} + \theta^T\mathbf{X}^T\mathbf{X}\right) = \mathbf{0}^T \\
&\Longleftrightarrow \quad -\mathbf{y}^T\mathbf{X} + \theta^T\mathbf{X}^T\mathbf{X} = \mathbf{0}^T \\
&\Longleftrightarrow \quad \theta^T\mathbf{X}^T\mathbf{X} = \cancel{\mathbf{0}^T} + \mathbf{y}^T\mathbf{X} \\
&\Longleftrightarrow \quad \theta^T = \mathbf{y}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} \qquad\qquad (\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T \\
&\Longleftrightarrow \quad \underbrace{\theta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}}_{\text{Normal Equation}}
\end{aligned}
$$

[MML Chap. 9.2.1]

Example: Linear Fit





- Closed form solution for maximum likelihood estimate enables to fit lines

Non-linear Functions

- Linear regression is *linear in parameters*
- We can apply non-linear transformation:

$$f(\mathbf{x}) = \mathbf{x}^T\theta \longrightarrow f(\mathbf{x}) = \phi(\mathbf{x})^T\theta$$

- Let $\phi : \mathbb{R}^D \to \mathbb{R}^K$ and define $\mathbf{\Phi} \in \mathbb{R}^{N\times K}$ as

$$\mathbf{\Phi} = \begin{pmatrix} \phi(\mathbf{x}_1)^T \\ \vdots \\ \phi(\mathbf{x}_N)^T \end{pmatrix} \in \mathbb{R}^{N\times K}$$

- Everything else stays the same! Use normal equation with $\mathbf{\Phi}$ instead of $\mathbf{X}$

$$\theta = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{y}$$

[MML Chap. 9.2.1] [CVMLI Chap. 8.3]

Example: Polynomial transformation

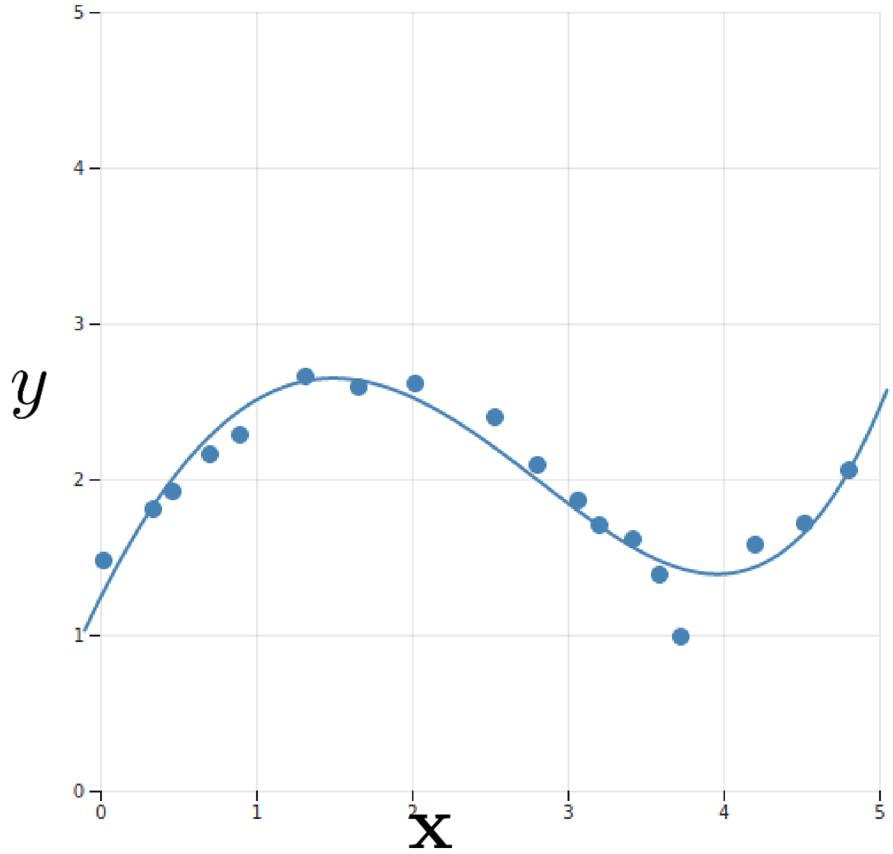
- With polynomial transformation, we can fit polynomials of degree K

$$\phi_{\text{poly}}(\mathbf{x}_n) = \begin{pmatrix} 1 \\ x_1 \\ x_1^2 \\ \vdots \\ x_1^K \\ \vdots \end{pmatrix} \in \mathbb{R}^{DK+1}$$

- With K=1 it's "vanilla" linear regression

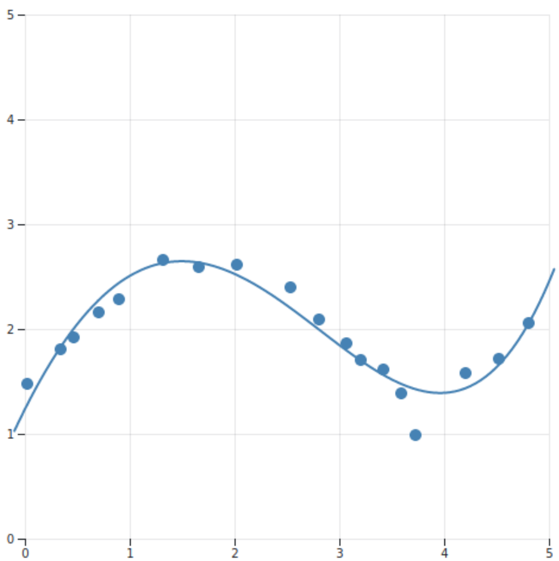
[MML Chap. 9.2.1]

Example: Polynomial Fit

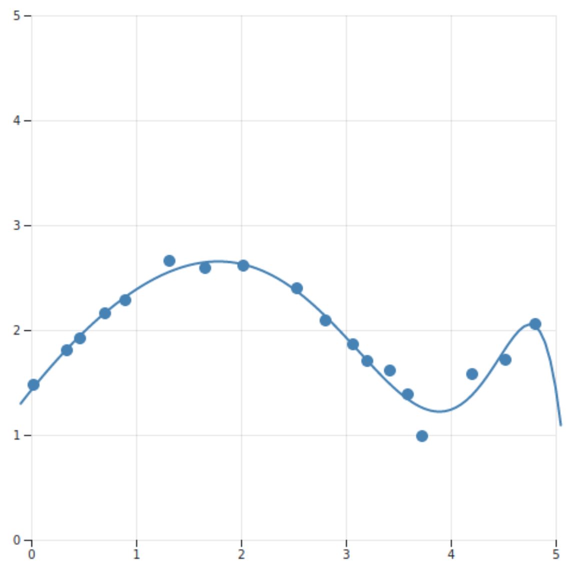


- With a polynomial of degree 3, we get a good fit. But can we do better with higher degrees?

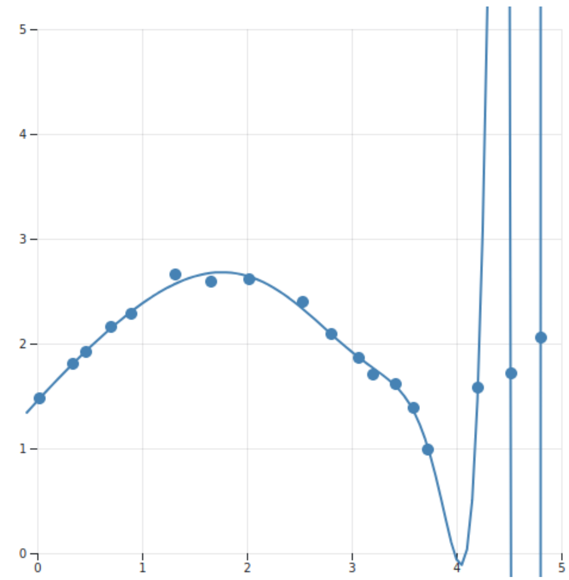
Example: Overfitting



degree 3



degree 5



degree 8

- Increasing the degree, we will get better training error
- But function will have implausible shape

Overfitting and Generalization

- Maximum Likelihood estimates overfit to training data and overconfident predictions
- High capacity models ($K > 5$) tend to fit training data too well
- Usually caused by too large parameter values

- **Solution:** Ensure that parameters do not get too large!

Common Parameter Estimation

- Learning is finding parameters of

$$P(\theta|\mathbf{x}_{1:N}, y_{1:N}) \propto P(y_{1:N}|\mathbf{x}_{1:N}, \theta) \; \boxed{P(\theta)}$$

- Paradigms for parameter estimation:

1. Point estimate with uniform prior
   → Maximum Likelihood Estimation
2. Point estimate with given prior
   → Maximum A posteriori Estimation (MAP)
3. Determine posterior over the parameters
   → Bayesian Estimation

NLL with Prior

- Assume Gaussian prior for parameters:

$$P(\theta) = \mathcal{N}(\theta | 0, b^2 \mathbf{I})$$

Identity matrix

- NLL is then

$$\mathcal{L}_{\text{MAP}}(\theta) = -\log \prod_{n=1}^{N} P(y_n | \mathbf{x}_n, \theta) \underline{- \log P(\theta)}$$

- Inserting Gaussians results in

$$\mathcal{L}_{MAP}(\theta) = \frac{1}{2\sigma^2} \sum_{n=1}^{N} \left(y_n - \phi(\mathbf{x}_n)^T \theta \right)^2 + \frac{1}{2b^2} \theta^T \theta + \text{const}$$

$$= \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{\Phi}\theta)^T (\mathbf{y} - \mathbf{\Phi}\theta) + \frac{1}{2b^2} \theta^T \theta + \text{const}$$

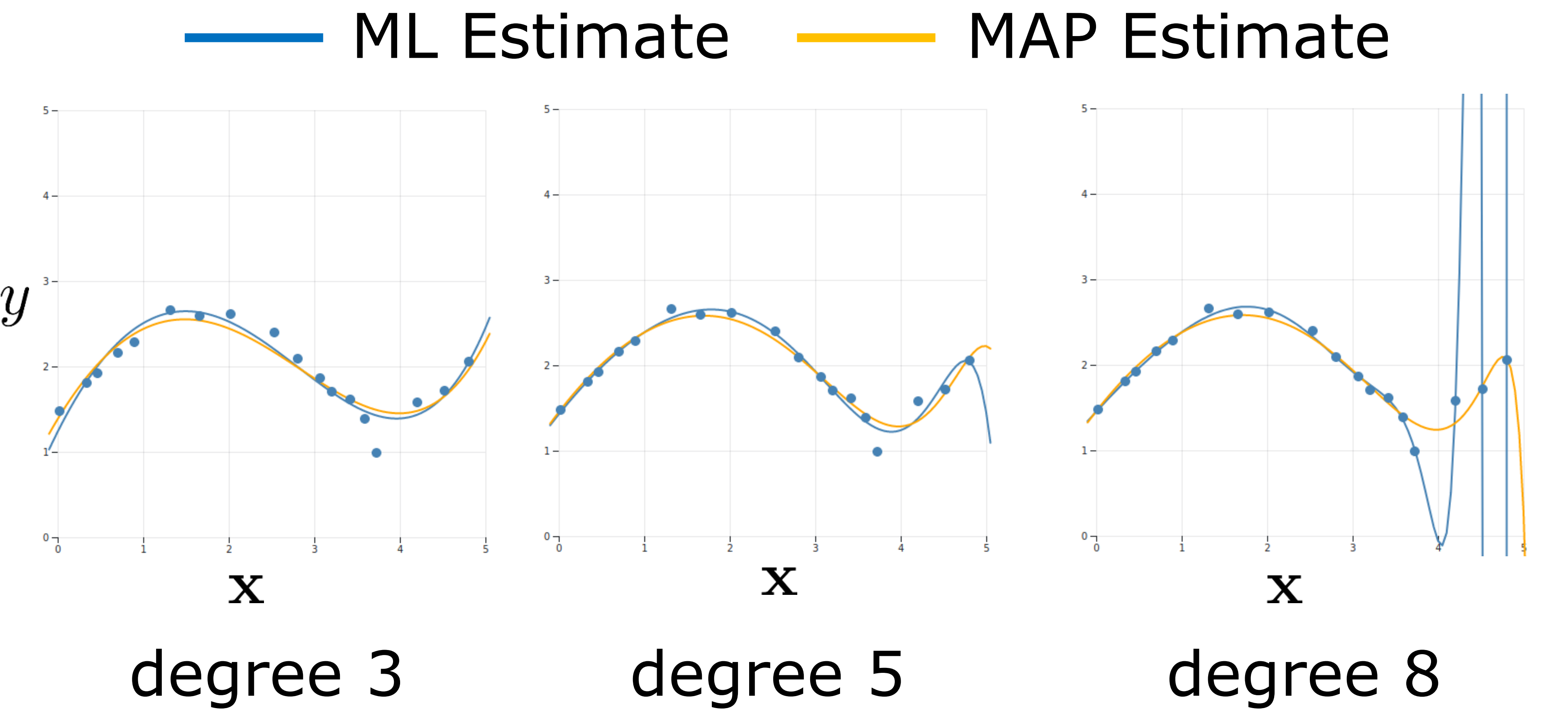
[MML Chap. 9.2.3]

Gradient of MAP NLL

- Same receipt as before:

$$
\begin{aligned}
\frac{d\mathcal{L}_{\text{MAP}}}{d\theta} = \mathbf{0}^T \iff\ & \frac{1}{\sigma^2}\left(\theta^T \boldsymbol{\Phi}^T \boldsymbol{\Phi} - \mathbf{y}^T \boldsymbol{\Phi}\right) + \frac{1}{b^2}\theta^T = \mathbf{0}^T \\
\iff\ & \frac{1}{\sigma^2}\theta^T \left(\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \frac{\sigma^2}{b^2}\mathbf{I}\right) - \frac{1}{\sigma^2}\mathbf{y}^T \boldsymbol{\Phi} = \mathbf{0}^T \\
\iff\ & \theta^T \left(\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \frac{\sigma^2}{b^2}\mathbf{I}\right) = \mathbf{y}^T \boldsymbol{\Phi} \\
\iff\ & \theta^T = \mathbf{y}^T \boldsymbol{\Phi} \left(\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \frac{\sigma^2}{b^2}\mathbf{I}\right)^{-1} \\
\iff\ & \theta = \left(\boldsymbol{\Phi}^T \boldsymbol{\Phi} \underline{+ \frac{\sigma^2}{b^2}\mathbf{I}}\right)^{-1} \boldsymbol{\Phi}^T \mathbf{y}
\end{aligned}
$$

Example: ML vs. MAP Estimate



- Smoother functions even at higher degrees!

Common Parameter Estimation

- Learning is finding parameters of

$$P(\theta|\mathbf{x}_{1:N}, y_{1:N}) \propto P(y_{1:N}|\mathbf{x}_{1:N}, \theta) \quad P(\theta)$$

- Paradigms for parameter estimation:

1. Point estimate with uniform prior
   → Maximum Likelihood Estimation
2. Point estimate with given prior
   → Maximum A posteriori Estimation (MAP)
3. Determine posterior over the parameters
   → Bayesian Estimation

Bayesian Approach

- ML and MAP estimates provide most likely parameters after observing the training data

$$\theta^\star = \arg\max_\theta P(\theta|\mathbf{x}_{1:N}, y_{1:N})$$

- **Bayesian approach**: Posterior over parameters after observing training data

$$P(\theta|\mathbf{x}_{1:N}, y_{1:N}) = \frac{P(y_{1:N}|\mathbf{x}_{1:N},\theta)P(\theta)}{P(y_{1:N}|\mathbf{x}_{1:N})}$$

$$= \frac{P(y_{1:N}|\mathbf{x}_{1:N},\theta)P(\theta)}{\int P(y_{1:N}|\mathbf{x}_{1:N},\theta)P(\theta)\ \mathrm{d}\theta}$$

[MML Chap. 9.3] [CVMLI Chap. 8.2]

Posterior Predictions

- Predictions $y_* \in \mathbb{R}$ for unseen examples $\mathbf{x}_* \in \mathbb{R}^D$ are weighted average over all possible parameters:

$$P(y_*|x_*, \mathbf{x}_{1:N}, y_{1:N}) = \int P(y_*|\mathbf{x}_*, \theta) P(\theta|\mathbf{x}_{1:N}, y_{1:N}) \, \mathrm{d}\theta$$

- **Advantage:** We take the uncertainty of the parameters into account

[MML Chap. 9.3.4]

Bayesian Linear Regression

- Stick with Gaussians for likelihood and prior

$$P(y_{1:N}|\mathbf{x}_{1:N}, \theta) = \mathcal{N}(\mathbf{y}|\mathbf{\Phi}\theta, \sigma^2\mathbf{I})$$

$$P(\theta) = \mathcal{N}(\mathbf{m}_0, \mathbf{S}_0)$$

- To derive the posterior, we can exploit that marginal is product of likelihood and prior

$$P(\theta|y_{1:N}, \mathbf{x}_1 : N) = \frac{P(y_{1:N}|\mathbf{x}_{1:N}, \theta)P(\theta)}{\int P(y_{1:N}|\mathbf{x}_{1:N}, \theta)P(\theta) \, \mathrm{d}\theta}$$

Inserting Likelihood and Prior

- We insert our choices for likelihood and prior into the numerator:

$$P(y_{1:N}|\mathbf{x}_{1:N},\theta)P(\theta) = \mathcal{N}(\underline{\mathbf{y}}|\mathbf{\Phi}\theta,\sigma^2\mathbf{I})\mathcal{N}(\underline{\theta}|\mathbf{m}_0,\mathbf{S}_0)$$

- Nearly a product of two Gaussians, but different variables
- First ensure that Likelihood in terms of θ

Gaussians: Change of variables

- Consider normal distribution of $\mathbf{x}$ with mean as linear function $\mathbf{A}\mathbf{y}+\mathbf{b}$: $\mathcal{N}(\mathbf{x}|\mathbf{A}\mathbf{y}+\mathbf{b},\Sigma)$

- Change of variable such that normal distribution in terms of $\mathbf{y}$

$$\mathcal{N}(\mathbf{x}|\mathbf{A}\mathbf{y}+\mathbf{b},\Sigma) = \eta^{-1}\mathcal{N}(\mathbf{y}|\mathbf{A}'\mathbf{x}+\mathbf{b}',\Sigma')$$

with

$$\Sigma' = (\mathbf{A}^T\Sigma^{-1}\mathbf{A})^{-1}$$

$$A' = \Sigma'\mathbf{A}^T\Sigma^{-1}$$

$$b' = -\mathbf{A}'\mathbf{b}$$

$$\eta = |\mathbf{A}|$$

[CVMLI Chap. 5.7]

Deriving the Posterior (I)

- Change variables for $\mathcal{N}(\mathbf{y}|\mathbf{\Phi}\theta, \sigma^2\mathbf{I})$ gives:

$$\mathcal{N}(\mathbf{y}|\mathbf{\Phi}\theta, \sigma^2\mathbf{I}) = \eta_1^{-1}\mathcal{N}(\theta|\mathbf{A}'\mathbf{y}, \Sigma')$$

with

$$\Sigma' = (\sigma^{-2}\mathbf{\Phi}^T\mathbf{\Phi})^{-1}$$

$$\mathbf{A}' = (\sigma^{-2}\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\sigma^{-2}\mathbf{\Phi}^T$$

$$\Sigma' = (\mathbf{A}^T\Sigma^{-1}\mathbf{A})^{-1}$$

$$A' = \Sigma'\mathbf{A}^T\Sigma^{-1}$$

- Next we want to compute

$$\mathcal{N}(\theta|\mathbf{A}'\mathbf{y}, \Sigma')\mathcal{N}(\theta|\mathbf{m}_0, \mathbf{S}_0)$$

Gaussian Product

- Product of Gaussians is again a (unnormalized) Gaussian given by:

$$\mathcal{N}(\mathbf{x}|\mathbf{a}, \mathbf{A})\mathcal{N}(\mathbf{x}|\mathbf{b}, \mathbf{B}) = \eta^{-1}\mathcal{N}(\mathbf{x}|\mathbf{c}, \mathbf{C}),$$

with

$$\mathbf{C} = \left(\mathbf{A}^{-1} + \mathbf{B}^{-1}\right)^{-1}$$

$$\mathbf{c} = \mathbf{C}\left(\mathbf{A}^{-1}\mathbf{a} + \mathbf{B}^{-1}\mathbf{b}\right)$$

$$\eta = \mathcal{N}(\mathbf{a}|\mathbf{b}, \mathbf{A} + \mathbf{B}) = \mathcal{N}(\mathbf{b}|\mathbf{a}, \mathbf{A} + \mathbf{B})$$

[MML Chap. 6.5.2] [CVMLI Chap. 5.6]

Deriving the Posterior (II)

- Using the product we can simplify further

$$\mathcal{N}\left(\mathbf{y}|\boldsymbol{\Phi}\theta, \sigma^2\mathbf{I}\right)\mathcal{N}\left(\theta|\mathbf{m}_0, \mathbf{S}_0\right)$$

$$= \underbrace{\eta_1^{-1}\mathcal{N}\left(\theta\middle|\left(\sigma^{-2}\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}\sigma^{-2}\Phi^T\mathbf{y}, \left(\sigma^{-2}\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}\right)}_{\text{change of variables}}\mathcal{N}(\theta|\mathbf{m}_0, \mathbf{S}_0)$$

$$= \eta_1^{-1}\eta_2^{-1}\mathcal{N}\left(\theta|\mathbf{m}_N, \mathbf{S}_N\right)$$

$$\boxed{\begin{aligned}\mathbf{C} &= \left(\mathbf{A}^{-1} + \mathbf{B}^{-1}\right)^{-1} \\ \mathbf{c} &= \mathbf{C}\left(\mathbf{A}^{-1}\mathbf{a} + \mathbf{B}^{-1}\mathbf{b}\right)\end{aligned}}$$

with

$$\mathbf{S}_N = \left(\sigma^{-2}\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \mathbf{S}_0^{-1}\right)^{-1}$$

$$\mathbf{m}_N = \mathbf{S}_N\left(\sigma^{-2}\boldsymbol{\Phi}^T\boldsymbol{\Phi}\left(\sigma^{-2}\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}\sigma^{-2}\boldsymbol{\Phi}^T\mathbf{y} + \mathbf{S}_0^{-1}\mathbf{m}_0\right)$$

$$= \mathbf{S}_N\left(\sigma^{-2}\boldsymbol{\Phi}^T\mathbf{y} + \mathbf{S}_0^{-1}\mathbf{m}_0\right)$$

Solving the Marginal Likelihood

- The numerator of the posterior is

$$P(y_{1:N}|\theta, \mathbf{x}_{1:N})P(\theta) = \eta_1^{-1}\eta_2^{-1}\mathcal{N}\left(\theta|\mathbf{m}_N, \mathbf{S}_N\right)$$

- The denominator is

$$\begin{aligned} P(y_{1:N}|\mathbf{x}_{1:N}) &= \int P(y_{1:N}|\theta, \mathbf{x}_{1:N})P(\theta)\ \mathrm{d}\theta \\ &= \int \eta_1^{-1}\eta_2^{-1}\mathcal{N}\left(\theta|\mathbf{m}_N, \mathbf{S}_N\right)\ \mathrm{d}\theta \\ &= \eta_1^{-1}\eta_2^{-1} \underbrace{\int \mathcal{N}\left(\theta|\mathbf{m}_N, \mathbf{S}_N\right)\ \mathrm{d}\theta}_{=1} \\ &= \eta_1^{-1}\eta_2^{-1} \end{aligned}$$

Posterior is Gaussian

- Putting everything together, gives us:

$$P(\theta|y_{1:N}, \mathbf{x}_{1:N}) = \mathcal{N}\left(\theta|\mathbf{m}_N, \mathbf{S}_N\right)$$

  with

$$\mathbf{S}_N = \left(\sigma^{-2}\mathbf{\Phi}^T\mathbf{\Phi} + \mathbf{S}_0^{-1}\right)^{-1}$$
$$\mathbf{m}_N = \mathbf{S}_N\left(\sigma^{-2}\mathbf{\Phi}^T\mathbf{y} + \mathbf{S}_0^{-1}\mathbf{m}_0\right)$$

- Posterior has the same form as the prior.

Conjugate Prior

- Prior is a **conjugate Prior** for a likelihood function if the posterior is of the *same* form/type as the prior.
- Gaussians with known Σ are self-conjugate.

- Conjugate priors lead to closed-form solutions
- Posterior is prior with updated parameters

Posterior Prediction

- Using the posterior, we can finally make predictions:

$$P(y_*|\mathbf{x}_{1:N}, y_{1:N}) = \int P(y_*|\mathbf{x}_*, \theta) P(\theta|\mathbf{x}_{1:N}, y_{1:N}) \, \mathrm{d}\theta$$

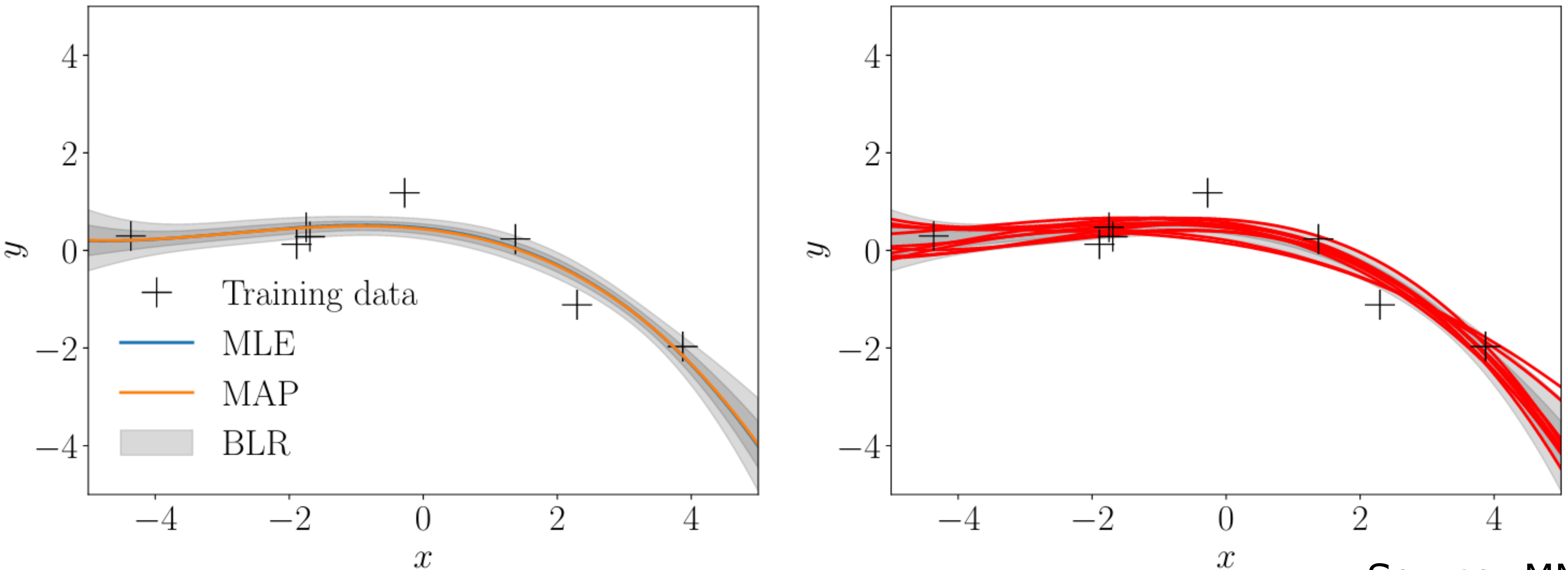
- Inserting likelihood and posterior gives:

$$= \int \mathcal{N}(y_*|\phi(\mathbf{x}_*)\theta, \sigma^2) \mathcal{N}(\theta|\mathbf{m}_N, \mathbf{S}_N) \, \mathrm{d}\theta$$

- Product of Gaussians is Gaussian again. One can derive that

$$= \mathcal{N}(y_*|\phi^T(\mathbf{x}_*)\mathbf{m}_N, \phi^T(\mathbf{x}_*)\mathbf{S}_N\phi(\mathbf{x}_*) + \sigma^2)$$

Example: Bayesian Regression

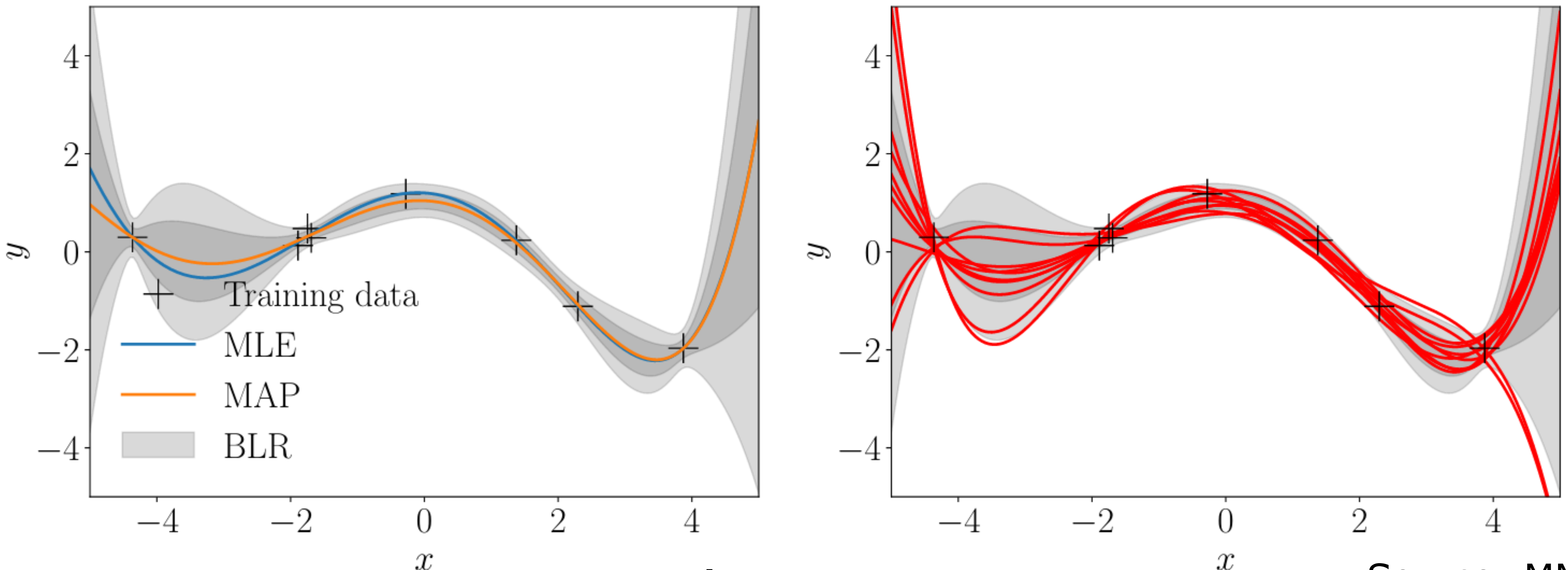


Source: MML

degree 3

- With higher degrees, we see also larger confidence intervals.

Example: Bayesian Regression

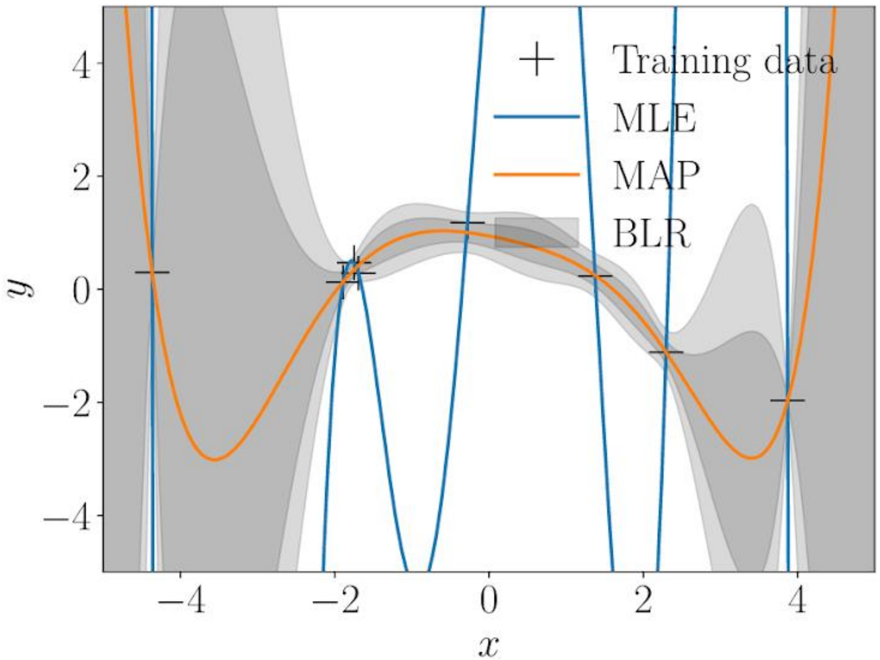


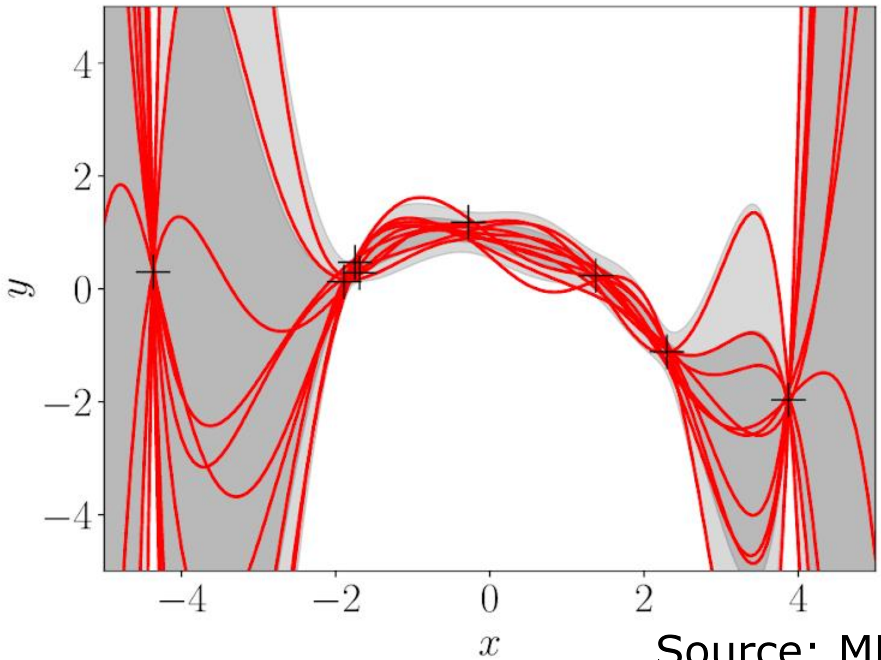
Source: MML

degree 5

- With higher degrees, we see also larger confidence intervals.

Example: Bayesian Regression





Source: MML

degree 7

- With higher degrees, we see also larger confidence intervals.

Why Bayesian Regression?

- In robotics several properties that are advantageous:

1. **Uncertainty** with variance of predictions
2. Determine ”regions of uncertainty”
3. **Incremental learning** possible: posterior can be prior for next round of learning!

Discriminative and Generative Models

- With Bayes Theorem, we can express as follows:

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} = \frac{P(\mathbf{x}|y)P(y)}{\int P(\mathbf{x}|y)P(y)\ \mathrm{d}y}$$

- Up to now, we used a discriminative model we modeled directly $P(y|\mathbf{x})$
- But we could equally model $P(\mathbf{x}, y)$ or $P(\mathbf{x}|y), P(y)$ to get P(y|x) → generative model

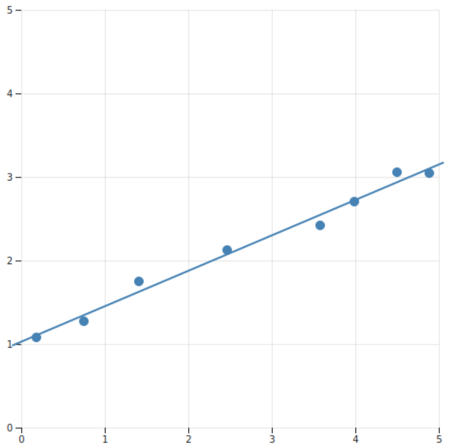
[CV-MLI Chap. 6.2]

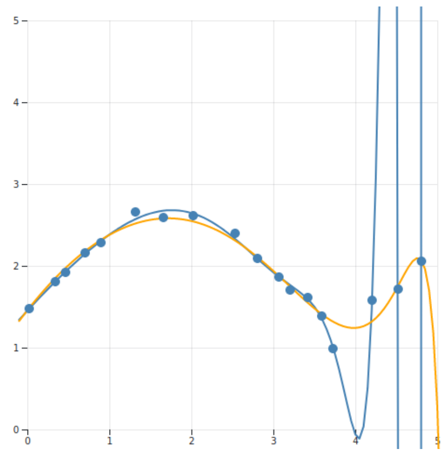
Which model to use?

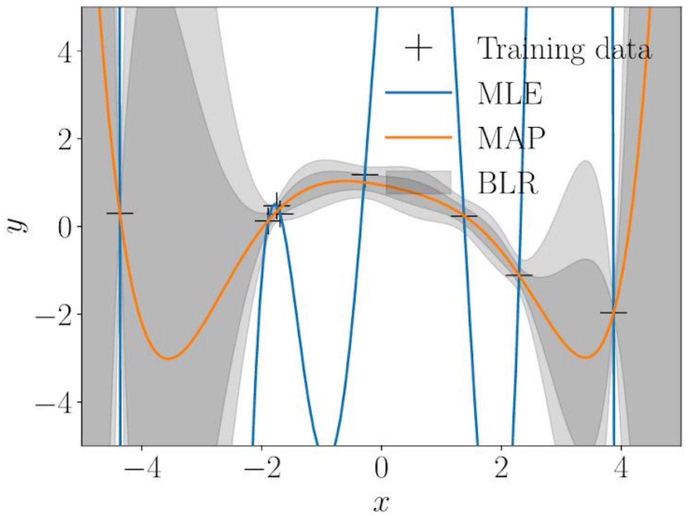
- No definite answer (depends)
- Points to consider:
  1. Inference simpler with discriminative models
  2. The data x is of higher dimension then the label; needs complex models
  3. $P(x|y)$ allows to directly model the generation process and generate data.
  4. Generative models can handle missing data.
  5. Priors can incorporate expert knowledge in the prior in generative models.

[CV-MLI Chap. 6.5]

Summary







- Linear Regression and variants
- Bayesian Linear Regression
- Discriminative vs. Generative Models

See you next week!