Predictive modelling and generalisation

Goal: identify relationship between predictor (input) and target (output)

Let $\underline{x}$: predictor, y: target (here scalar)

Regression: y real-valued

Classification: y class label

$\underline{x}, y$ assumed to be random variables with joint distribution $p(\underline{x}, y)$

Ideally: find $p(y|\underline{x})$, i.e. conditional distrib.

Often: prediction function $h(\underline{x})$ yielding single $\hat{y}$ $\rightsquigarrow$ loss function $\mathcal{L}(\hat{y}, y)$ required

Quality of h: "Prediction loss"

$$J(h) = \mathbb{E}_{\hat{y},y}[\mathcal{L}(\hat{y}, y)] = \mathbb{E}_{\underline{x},y}[\mathcal{L}(h(\underline{x}), y)]$$

where $\mathbb{E}_{x,y}$ is expectation w.r.t. $p(\underline{x}, y)$

$\rightsquigarrow$ Restated goal: minimise $J(h)$ h

Training loss

Prediction loss optimisation problem typically not solvable directly

$\rightsquigarrow$ approximation required: $J(h) \approx \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(h(\underline{x}_i), y_i)$

where $(\underline{x}_i, y_i) \overset{iid.}{\sim} p(\underline{x}, y)$

i.e. "training data" $D^{train} = \{(\underline{x}_1, y_1), \ldots, (\underline{x}_n, y_n)\}$

Should be representative of data for which we do predictions

Typically: prediction function h parametrised:

$h(\underline{x}) = h_{\underline{\lambda}}(\underline{x}; \underline{\theta})$, where

$\underline{\lambda}$: hyperparameters (might indicate model family)

$\underline{\theta}$: tuning parameters (often optimised using gradient descent)

$\rightsquigarrow$ loss function $\mathcal{L}$ should be differentiable (often proxy loss L used)

$\rightsquigarrow$ "Training loss":

$$J_{\underline{\lambda}}(\underline{\theta}) = \frac{1}{n} \sum_{i=1}^{n} L(h_{\underline{\lambda}}(\underline{x}_i; \underline{\theta}), y_i)$$

Minimisation of training loss:

Often in two steps:

1. For a given $\underline{\lambda}$:

$$\hat{\underline{\theta}}_{\underline{\lambda}} = \arg\min_{\theta} J_{\underline{\lambda}}(\underline{\theta})$$

$$\hat{h}_{\underline{\lambda}}(\underline{x}) = h_{\underline{\lambda}}(\underline{x}, \hat{\underline{\theta}}_{\underline{\lambda}})$$

Minimal value: $J^*_{\underline{\lambda}} = \min_{\underline{\theta}} J_{\underline{\lambda}}(\underline{\theta}) = J_{\underline{\lambda}}(\hat{\underline{\theta}}_{\underline{\lambda}})$

2. minimise $J^*_{\underline{\lambda}}$ "hyperparameter selection"

$\underline{\lambda}$

or "model selection" if $\underline{\lambda}$ indexes model families

Randomness from D^{train}:

$\Rightarrow$ $J_{\underline{\lambda}}(\underline{\theta})$, $\hat{h}_{\underline{\lambda}}$, $J^*_{\underline{\lambda}}$ are all stochastic

Example:

Distributions of predictor and target variables:

$$p(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right)$$

$$p(y|x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(y - g(x))^2\right)$$

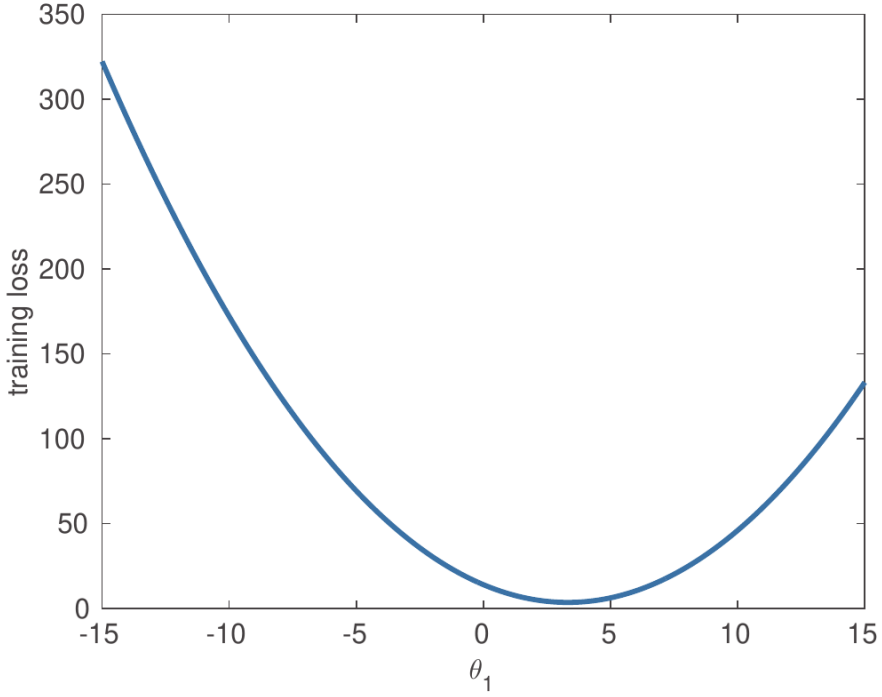
where $g(x) = \frac{1}{4}x + \frac{3}{4}x^2 + x^3$

"Ground-truth" - generally not known

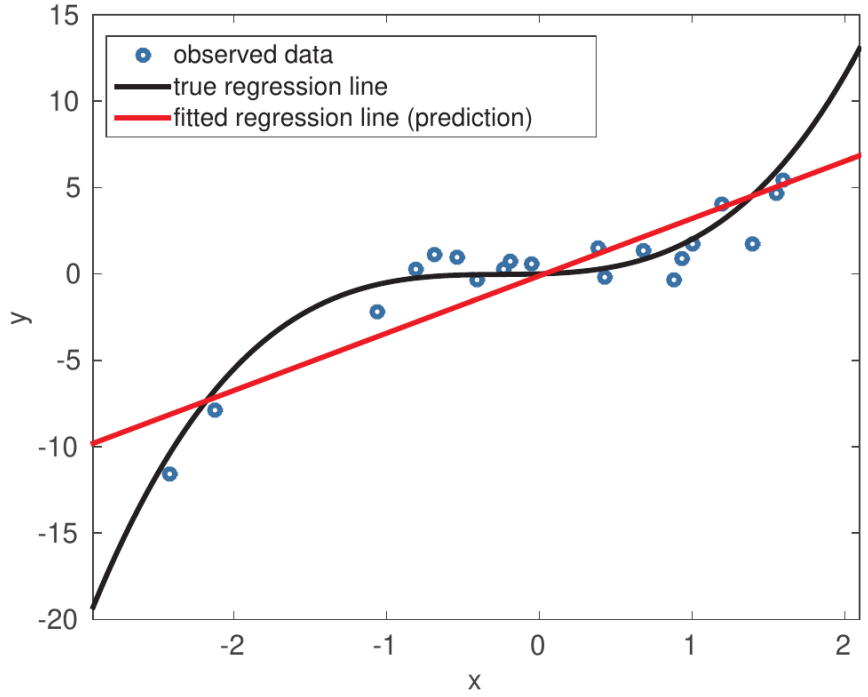
Note that $g(x) = \mathbb{E}[y|x]$

minimises expected $\mathcal{L}(\hat{y}, y) = (\hat{y} - y)^2$

Nonlinear Regression



(a) Training loss function



(b) Fitted prediction model

1) Linear regression (linear prediction model)

$$h_\theta(x; \underline{\theta}) = \theta_0 + \theta_1 x$$

$$\underline{\theta} = (\theta_0, \theta_1)^T$$

$\rightsquigarrow$ training loss: $J_1(\underline{\theta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \theta_0 - \theta_1 x_i)^2$

Optimal $\hat{\theta}_0$ for given θ_1:

$$\hat{\theta}_0 = \bar{y} - \theta_1 \bar{x}$$

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \qquad \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$\rightsquigarrow$ restated training loss:

$$J_1(\theta_1) = \frac{1}{n}\sum_{i=1}^{n}\left((y_i - \bar{y}) - \theta_1 (x_i - \bar{x})\right)^2$$

Solution: $\hat{\theta}_1 = \underset{\theta_1}{\operatorname{argmin}}\, J_1(\theta_1)$

with prediction model:

$$\hat{h}_1(x) = \hat{\theta}_0 + \hat{\theta}_1 x = \bar{y} + \hat{\theta}_1 (x - \bar{x})$$

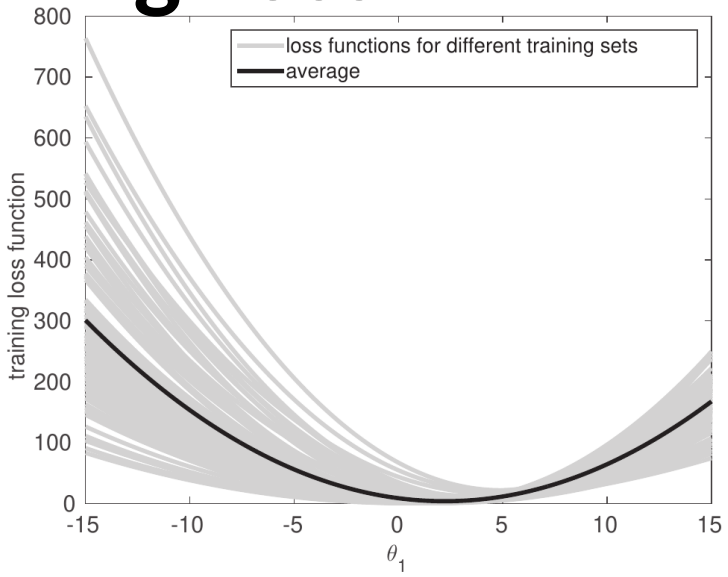
2) Polynomial regression:

$$h_\lambda(x; \underline{\theta}) = \sum_{k=0}^{\lambda} \theta_k x^k$$

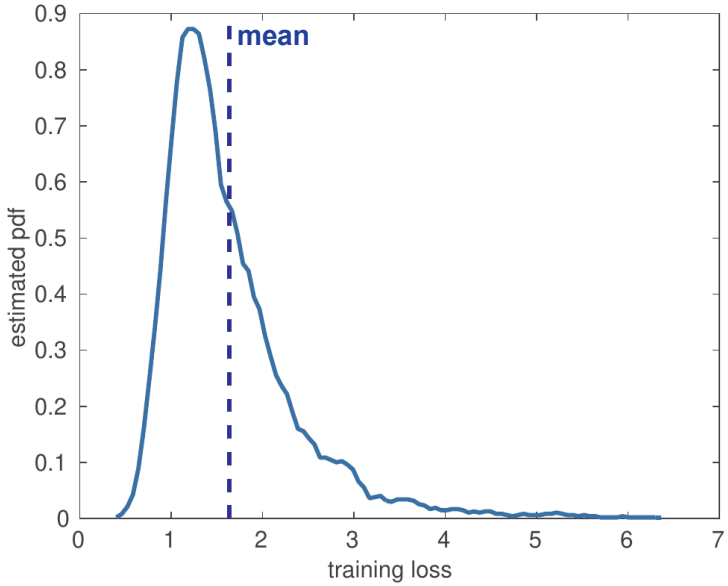
$$\underline{\theta} = (\theta_0, \ldots, \theta_\lambda)^T$$

- "Polynomials of degree λ": one prediction model for each λ
- Complexity, # parameters increase with increasing λ
- Variance and mean of training loss decrease for increasing λ

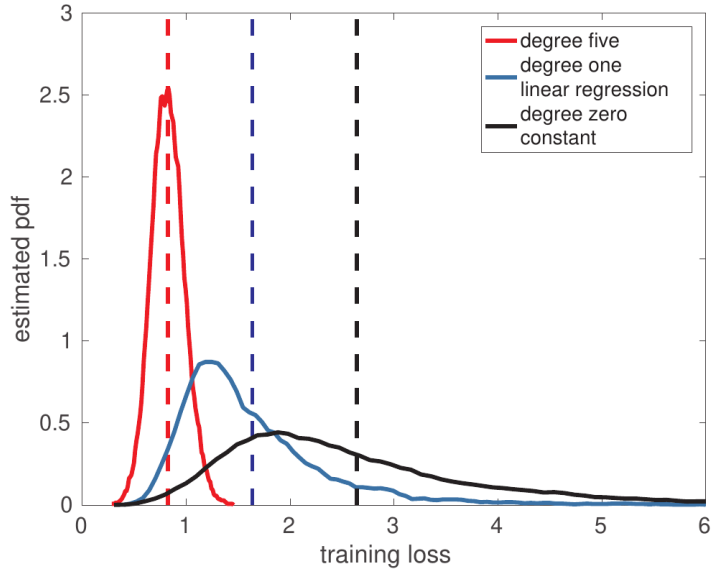
Training Loss



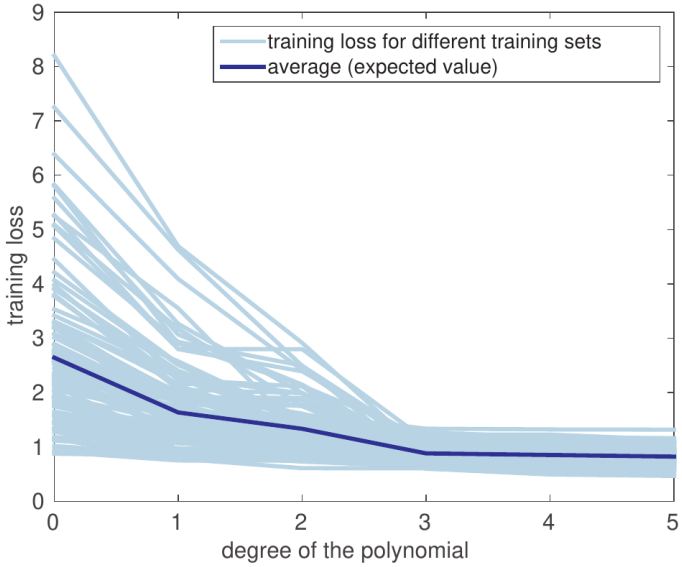
(a) Training loss functions



(b) Training loss



(a) Distribution of the training loss



(b) Training loss for different models

Dimensionality reduction preprocessing

Example: Given 100-dimensional predictor $\underline{X}$ carrying information about target y.

To simplify prediction, we are reducing the dimensionality of $\underline{X}$ from 100 to 10. We use PCA and nonlinear PCA and find that prediction is better using nonlinear PCA with a particular Φ.

Which of the following statements are correct?

A) Z_{PCA} carries more information about y than X does.

B) $Z_{NL\text{-}PCA}$ carries more information about y than X does.

C) $Z_{NL\text{-}PCA}$ carries more information about y than Z_{PCA} does.