Machine Learning

Training versus test error

Sebastian Engelke

Master in Business Analytics

UNIVERSITÉ
DE GENÈVE

Training versus test error

Training Data

- Fit our model $\widehat{f}$ on the training data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, the n measurements of (X, Y).
- Compute the `training error` as

$$\begin{aligned} \mathrm{MSE}_{\mathrm{Tr}} &= \mathrm{RSS}/n \\ &= \frac{1}{n}\sum_{i=1}^{n}\{y_i - \widehat{f}(x_i)\}^2. \end{aligned}$$

- Since higher model complexity always reduces this error, training error is overly `optimistic`!

Training versus test error

Training Data

- Fit our model $\widehat{f}$ on the training data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, the n measurements of (X, Y).
- Compute the `training error` as

$$\begin{aligned} \mathrm{MSE}_{\mathrm{Tr}} &= \mathrm{RSS}/n \\ &= \frac{1}{n}\sum_{i=1}^{n}\{y_i - \widehat{f}(x_i)\}^2. \end{aligned}$$

- Since higher model complexity always reduces this error, training error is overly `optimistic`!

Test Data

- The `test data` are m new samples $\{(\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_m, \tilde{y}_m)\}$ of (X, Y).
- We test our model $\widehat{f}$ fitted to the training data by predicting $\widehat{\tilde{y}}_i = \widehat{f}(\tilde{x}_i)$ at all $\tilde{x}_i$.
- The `test/generalization error` is

$$\mathrm{MSE}_{\mathrm{Te}} = \frac{1}{m}\sum_{i=1}^{m}\{\tilde{y}_i - \widehat{f}(\tilde{x}_i)\}^2.$$

- The test error is an estimate of the `expected prediction error`

$$\mathrm{Err}_{\widehat{f}} = \mathrm{E}[\{Y - \widehat{f}(X)\}^2],$$

Training versus test error

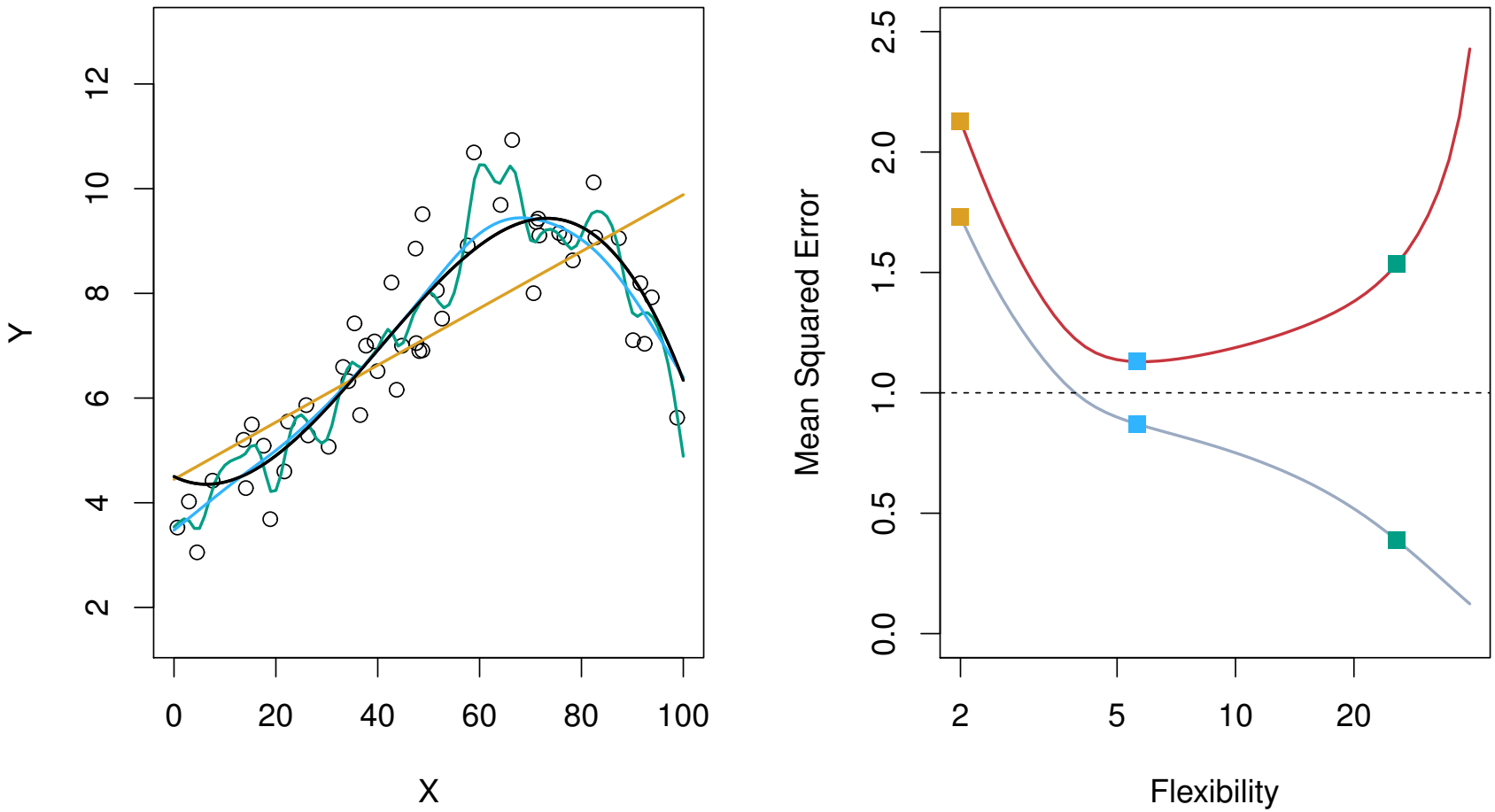

Figure: Left: Data simulated from f, the black line, and three estimates, the linear fit (orange) and higher order polynomials (blue and green). Right: Training error $\mathrm{MSE}_{\mathrm{Tr}}$ (grey) and test error $\mathrm{MSE}_{\mathrm{Te}}$ (red) for the three fits.

The Bias–Variance decomposition

- As usually, we assume $Y = f(X) + \epsilon$, with $\mathrm{E}(\epsilon) = 0$ and $\mathrm{Var}(\epsilon) = \sigma_\epsilon^2$.
- For a regression fit $\widehat{f}$, we compute the expected prediction error at an input point $X = x_0$:

$$\begin{aligned}
\mathrm{Err}_{\widehat{f}}(x_0) &= \mathrm{E}[(Y - \widehat{f}(X))^2 \mid X = x_0] \\
&\overset{\text{Exercise}}{=} \sigma_\epsilon^2 + [\mathrm{E}\widehat{f}(x_0) - f(x_0)]^2 + \mathrm{E}[\widehat{f}(x_0) - \mathrm{E}\widehat{f}(x_0)]^2 \\
&= \sigma_\epsilon^2 + \mathrm{Bias}^2(\widehat{f}(x_0)) + \mathrm{Var}(\widehat{f}(x_0)) \\
&= \texttt{Irreducible Error} + \texttt{Bias}^2 + \texttt{Variance}.
\end{aligned}$$

The Bias–Variance decomposition

- As usually, we assume $Y = f(X) + \epsilon$, with $\mathrm{E}(\epsilon) = 0$ and $\mathrm{Var}(\epsilon) = \sigma_\epsilon^2$.
- For a regression fit $\widehat{f}$, we compute the expected prediction error at an input point $X = x_0$:

$$\begin{aligned}
\mathrm{Err}_{\widehat{f}}(x_0) &= \mathrm{E}[(Y - \widehat{f}(X))^2 \mid X = x_0] \\
&\overset{\text{Exercise}}{=} \sigma_\epsilon^2 + [\mathrm{E}\widehat{f}(x_0) - f(x_0)]^2 + \mathrm{E}[\widehat{f}(x_0) - \mathrm{E}\widehat{f}(x_0)]^2 \\
&= \sigma_\epsilon^2 + \mathrm{Bias}^2(\widehat{f}(x_0)) + \mathrm{Var}(\widehat{f}(x_0)) \\
&= \texttt{Irreducible Error} + \texttt{Bias}^2 + \texttt{Variance}.
\end{aligned}$$

- The `Irreducible Error` is the error due to the noise variable ϵ and cannot be improved.
- The `Bias` is the difference between the average prediction $\mathrm{E}\widehat{f}(x_0)$ and the truth at x_0.
- The `Variance` is the variability of the prediction $\widehat{f}(x_0)$ when $\widehat{f}$ is fitted for different data sets.

The Bias–Variance decomposition

Example: The `k-Nearest-Neighbors` with tuning parameter $k \in \{1, 2, \dots\}$ gives

$$\widehat{f}_k(x_0) = \frac{1}{k} \sum_{x_i \in N_k(x_0)} y_i.$$

The Bias–Variance decomposition

Example: The `k-Nearest-Neighbors` with tuning parameter $k \in \{1, 2, \dots\}$ gives

$$\widehat{f}_k(x_0) = \frac{1}{k} \sum_{x_i \in N_k(x_0)} y_i.$$

We obtain from the above:

$$\begin{aligned} \mathrm{Err}_{\widehat{f}}(x_0) &= \mathrm{E}[(Y - \widehat{f}_k(x_0))^2 \mid X_0 = x_0] \\ &= \sigma_\epsilon^2 + \left[f(x_0) - \frac{1}{k} \sum_{x_i \in N_k(x_0)} f(x_i) \right]^2 + \frac{\sigma_\epsilon^2}{k}. \end{aligned}$$

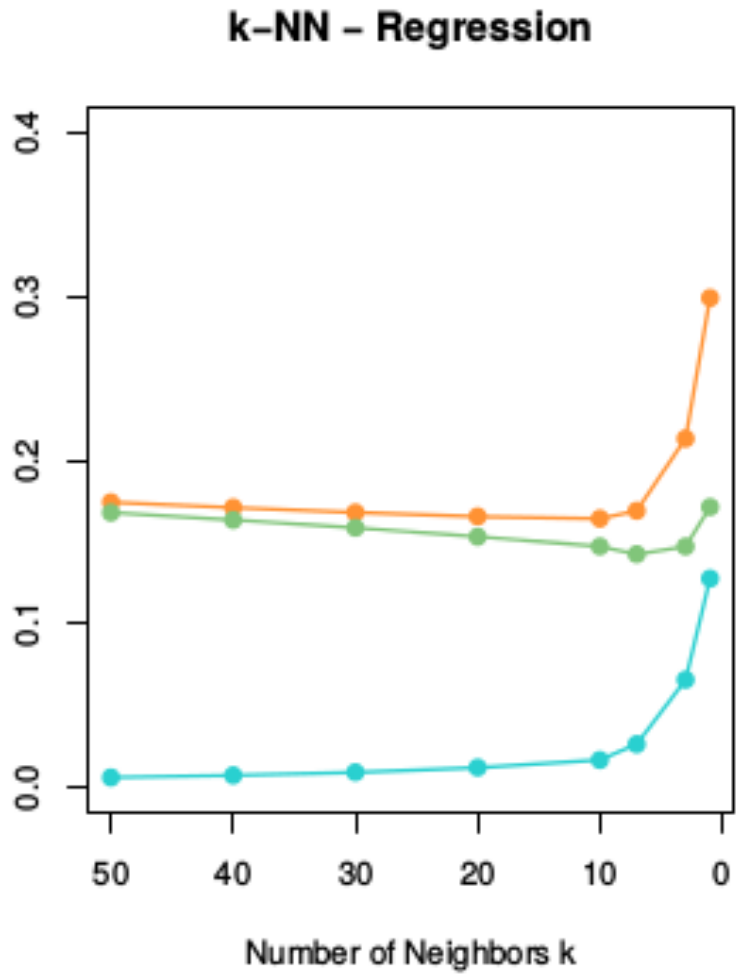

Figure: Expected prediction error (orange), squared bias (green) and variance (blue).

Bias–Variance trade off

Simple parametric methods such as `linear models`

- have low flexibility (`high bias`) but need less data to fit;
- are stable for different data sets (`low variance`);
- fit well if the data satisfies the model assumptions;
- are good for inference and `interpretation`.

Bias–Variance trade off

Simple parametric methods such as `linear models`

- have low flexibility (`high bias`) but need less data to fit;
- are stable for different data sets (`low variance`);
- fit well if the data satisfies the model assumptions;
- are good for inference and `interpretation`.

Complex (possibly non-parametric) methods such as `k-Nearest-Neighbors`

- can be very flexible (`low bias`) but need a of lot data;
- are prone to `overfitting` and vary strongly depending on the data set (`high variance`);
- are problematic in high dimensions (large p), so-called `curse of dimensionality`;
- are potentially better for `prediction`.

Training versus test error