Linear Black Box models

Example: Finite Impulse Response (FIR)

$$\begin{aligned} y(t) &= B(q)u(t) + e(t) \\ &= b_1 u(t-1) + \ldots + b_n u(t-n) + e(t) \end{aligned}$$

where B(q) is a polynomial in q^{-1}.

The corresponding predictor $\hat{y}(t|\theta) = B(q)u(t)$ is thus based on the following regression vector:

$$\varphi(t) = [u(t-1), u(t-2), \ldots, u(t-n)]$$

General family of linear black box models

$$\boxed{A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t)}$$

- *Box-Jenkins* (BJ) model ($A = 1$);
- *ARMAX model* ($F = D = 1$);
- *Output-Error* (OE) ($A = C = D = 1$);
- *ARX model* ($F = C = D = 1$).

General family of linear black box models

The regressors, i.e., the components of $\varphi(t, \theta)$ are in this general case given by:

- $u(t-k)$ (associated with the B polynomial);
- $y(t-k)$ (associated with the A polynomial);
- $\hat{y}_u(t-k|\theta)$ simulated outputs from past u (associated with the F polynomial);
- $\epsilon(t-k) = y(t-k) - \hat{y}(t-k|\theta)$ Prediction errors (associated with the D polynomial);
- $\epsilon_u(t-k) = y(t-k) - \hat{y}_u(t-k|\theta)$ simulation errors (associated with the D polynomial).

Nonlinear black-box structures

We have observed inputs $u(t)$ and outputs $y(t)$ from a dynamical system:

$$u^t = [u(1), u(2), \dots, u(t)],$$
$$y^t = [y(1), y(2), \dots, y(t)].$$

We are looking for a relationship between past observations $[u^{t-1}, y^{t-1}]$ and future outputs $y(t)$:

$$y(t) = g(u^{t-1}, y^{t-1}) + v(t),$$

where the additive term $v(t)$ accounts for the fact that the next output $y(t)$ will not be an exact function of past data. $v(t)$ must be very small so $g(u^{t-1}, y^{t-1})$ will be a good approximation.

Nonlinear black box models

Let us parametrize the function g with a finite dimensional parameter vector θ:

$$g(u^{t-1}, y^{t-1}, \theta).$$

g can be written as a concatenation of two mappings:

$$(u^{t-1}, y^{t-1}) \rightarrow \varphi(t) \rightarrow g(t),$$
$$g(u^{t-1}, y^{t-1}, \theta) = g(\varphi(t), \theta),$$

where

$$\varphi(t) = \varphi(u^{t-1}, y^{t-1}).$$

The general case allows parametrization of $\varphi(t)$:

$$\varphi(t) = \varphi(u^{t-1}, y^{t-1}, \eta).$$

Nonlinear black box models

$$\hat{y}(t|\theta) = g(\varphi(t), \theta),$$

where g is some nonlinear function parametrized by θ and $\varphi(t)$ are similar to regressors. Some types are:

- *NFIR* models which use $u(t-k)$ as regressors;
- *NARX* models which use $u(t-k)$ and $y(t-k)$ as regressors;
- *NOE* models which use $u(t-k)$ and $\hat{y}_u(t-k|\theta)$ as regressors;
- *NARMAX* models which use $u(t-k)$ and $y(t-k)$ and $\epsilon(t-k|\theta)$.

Possibilities of nonlinear mappings

$$g(\varphi, \theta),$$

which for any given θ goes from R^d to R^p and

$$\varphi = (\varphi_1, \dots, \varphi_d)^T.$$

It is natural to think of the parametrized function family as function expansions:

$$g(\varphi, \theta) = \sum \theta_k, g_k(\varphi),$$

where g_k are basis functions (functional basis in some cases).

Some ossibilities for g_k:

- Fourier series, Volterra series, Wavelets,
- Radial Basis Functions, B-splines, Sigmoid Neural Networks, etc.

The limitations of estimators designed with examples: The bias and variance dilemma

- estimators have intrinsic limitations due to their finite representational capacity and the use of finite training data set;
- these limitations are responsible for the *generalization errors* of the model when it is used with data which was not presented during the learning process.

- Given a data set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i), i = 1, \ldots, N\}$, where $\boldsymbol{x}$ is the independent variable and y is the response, obtained by sampling N times the set $X \times Y$ according to a probability distribution $P(\boldsymbol{x}, y)$.

- An *estimator* will be any function $h : X \to Y$ where the parameters are estimated using $\mathcal{D}$.

- The mean squared error of the estimator h is given by the functional:

$$\begin{aligned} \mathcal{I}[h] &= E[(y - h(\boldsymbol{x}))^2], \\ &= \int_{X \times Y} P(\boldsymbol{x}, y)(y - h(\boldsymbol{x}))^2 d\boldsymbol{x} dy, \end{aligned}$$

  defined as the *expected risk* .

The expected risk can be decomposed in two parts:

$$\mathcal{I}[h] = E[(h_o(\boldsymbol{x}) - h(\boldsymbol{x}))^2] + E[(y - h_o(\boldsymbol{x}))^2],$$

where $h_o(\boldsymbol{x})$ is the regression function $h_0(\boldsymbol{x}) = E[y|\boldsymbol{x}]$.

- It is possible to conclude that $h_o(\boldsymbol{x})$ minimizes the expected risk and is therefore the *best estimator*, since $h_o(\boldsymbol{x})$ is an unbiased estimate.

- The second term of the equation is the variance of y and cannot be influenced by the design of the estimator $h(\boldsymbol{x})$.

- $\boxed{\text{\textit{learning from examples} = reconstruction of the function } h_o(\boldsymbol{x}) \text{ given the set } \mathcal{D} = \{(\boldsymbol{x}_i, y_i), i = 1, \ldots, N\}}$
  where $\boldsymbol{x}$ has distribution $P(\boldsymbol{x})$
  and y is a random variable with mean $h_o(\boldsymbol{x})$
  and distribution $P(y|\boldsymbol{x})$.

- Assuming that the noise is additive, it is possible to write $y = h_o(\boldsymbol{x}) + \eta$, where η has distribution $P(y|\boldsymbol{x})$.

- In practice, $P(\boldsymbol{x}, y)$ is unknown and $\mathcal{I}[h]$ cannot be evaluated. Because only the training data set $\mathcal{D}$ is provided, the *expected risk* must be approximated by the *empirical risk*:

$$\mathcal{I}_{\text{emp}}[h] = \frac{1}{N}\sum_{i=1}^{N}(y_i - h(\boldsymbol{x}_i))^2.$$

- The problem of finding a regressor h that minimizes the empirical risk is ill-posed because an infinite number of solutions may exist. To avoid this, we must consider a family of parametric functions for the estimators.

- Non linear black box models with Q parameters can be chosen and represented by $\widehat{h}_{Q,N}(\boldsymbol{x})$. The index N means that the parameters were estimated using a set $\mathcal{D}$ with N elements.

- The error between the regression function $h_o(\boldsymbol{x})$ and the estimator $\widehat{h}_{Q,N}(\boldsymbol{x})$ will be called the generalization error:

$$E_{\mathcal{D}}[(h_o(\boldsymbol{x}) - \widehat{h}_{Q,N}(\boldsymbol{x}))^2],$$

  where $E_{\mathcal{D}}$ represents expectation over the ensemble of all possible $\mathcal{D}$.

- The generalization error can be decomposed in two parts, named bias and variance:

$$E_{\mathcal{D}}[(h_o(\boldsymbol{x}) - \widehat{h}_{Q,N}(\boldsymbol{x}))^2] = \underbrace{\left[\overline{h}_{Q,N}(\boldsymbol{x}) - h_o(\boldsymbol{x})\right]^2}_{(BIAS)^2} + \underbrace{\frac{1}{n}\sum_{j=1}^{n}\left[h^j_{Q,N}(\boldsymbol{x}) - \overline{h}_{Q,N}(\boldsymbol{x})\right]^2}_{VARIANCE}.$$

  where n is the number of estimators in the population and $\overline{h}_{Q,N}(\boldsymbol{x})$ is defined by:

$$\overline{h}_{Q,N}(\boldsymbol{x}) = \frac{1}{n}\sum_{j=1}^{n} h^j_{Q,N}(\boldsymbol{x}).$$

  Each estimator $h^j_{Q,N}(\boldsymbol{x})$ is supposed to be designed with an independent training data set $\mathcal{D}^j$.

- The bias measures the extent to which the average (over all sets) of the neuro-fuzzy models differ from the desired function $h_o(\boldsymbol{x})$, and the variance measures the extent to which the network function $h_{Q,N}(\boldsymbol{x})$ is sensitive to the particular choice of data set.

- The number of rules Q represent the power of approximation (or hypothesis complexity) of the hypothesis class if Q increases, the power of approximation increases. The bias and variance, and therefore the generalization error, depend on the complexity.

The bias/variance dilemma

- As complexity increases, bias decreases and variance increases;

- As complexity decreases, bias increases and variance decreases.

To demonstrate the importance of careful parameter selection, some neuro-fuzzy models are designed to approximate the following function:

$$g(x) = 4.26(\exp(-x) - 4\exp(-2x) + 3\exp(-3x)).$$

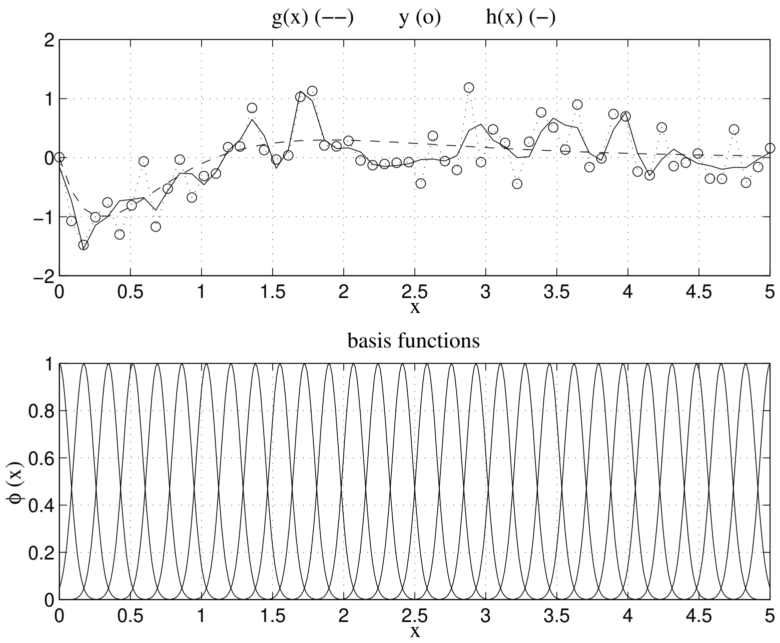
The training data is generated by sampling the following function:

$$y = g(x) + \eta,$$

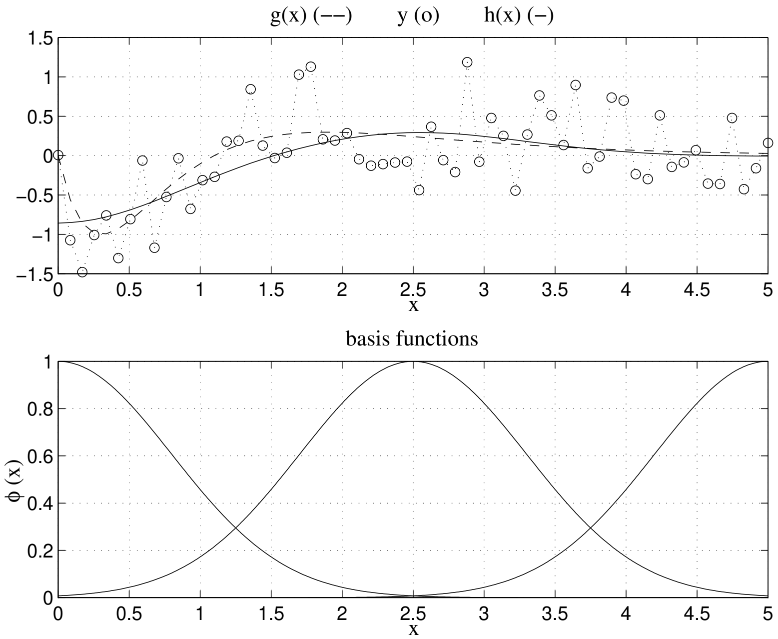
where η is an additive Gaussian noise in the output with standard deviation of $\sigma_r = 0.4^*$. Sixty uniformly distributed training data points $\mathcal{D} = \{(x_i, y_i), i = 1, \ldots, 60\}$ are generated in the interval of $X = [0.0, 5.0]$. Neuro-fuzzy models $h(x)$ with Gaussian basis functions are used. The Gaussian basis function can be represented as:

$$\phi(x) = \exp\left[-\frac{1}{2}\left(\frac{x - \overline{x}}{\sigma_w}\right)^2\right].$$

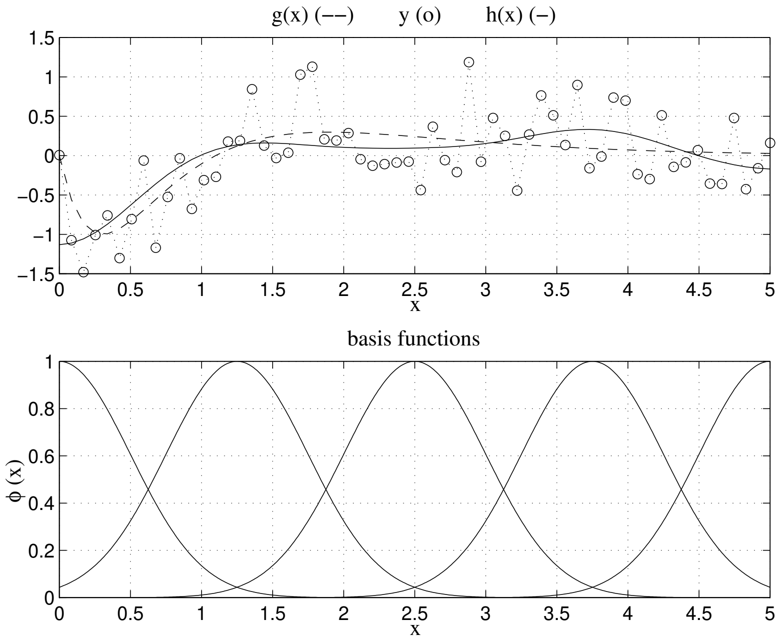
*in this example, the noise is made intentionally large to illustrate the over-fitting



estimator with 30 Gaussian basis functions and $\sigma_w = 0.07$. $g(x)$ is the true function, y is the training data and $h(x)$ is the estimator.



estimator with 3 Gaussian basis functions and $\sigma_w = 0.8$. $g(x)$ is the true function, y is the training data and $h(x)$ is the estimator.



estimator with 5 Gaussian basis functions and $\sigma_w = 0.5$. $g(x)$ is the true function, y is the training data and $h(x)$ is the estimator.

Example	number of basis functions	σ_w	$\mathcal{I}_{\text{true}}$	$\mathcal{I}_{\text{emp}}$
ex:1	30	0.07	0.092	0.08
ex:2	3	0.8	0.029	0.197
ex:3	5	0.5	0.048	0.181

Summary of mean squared error results.