Machine learning: lecture 3

Tommi S. Jaakkola
MIT CSAIL
tommi@csail.mit.edu

Topics

- Beyond linear regression models
  - Additive regression models, examples
  - generalization and cross-validation
- Statistical regression models
  - model formulation, motivation
  - maximum likelihood estimation

Review: linear regression

- A simple linear regression function is given by

$$f(x; \mathbf{w}) = w_0 + w_1 x$$

- We can set the parameters $\mathbf{w} = [w_0, w_1]$, for example, by minimizing the *empirical* or *training* error

$$\text{training error} = \frac{1}{n} \sum_{t=1}^{n} (y_t - w_0 - w_1 x_t)^2$$

- The hope here is that the resulting parameters/linear function has a low "generalization error", i.e., error on the new examples

$$\text{gen. error} = E_{(x,y) \sim P} \left(y - \hat{w}_0 - \hat{w}_1 x \right)^2$$

Review: generalization

- The "generalization" error,

$$E_{(x,y)\sim P}\,(y - \hat{w}_0 - \hat{w}_1 x)^2$$

can be written as a sum of two terms:

1. structural error (error of the best predictor in the class)

$$\begin{aligned} & E_{(x,y)\sim P}\,(y - w_0^* - w_1^* x)^2 \\ &= \min_{w_0,w_1} E_{(x,y)\sim P}\,(y - w_0 - w_1 x)^2 \end{aligned}$$

2. and the approximation error (how well we approximate the best predictor) based on a limited training set

$$E_{(x,y)\sim P}\,\big((w_0^* + w_1^* x) - (\hat{w}_0 + \hat{w}_1 x)\big)^2$$

Beyond simple linear regression

- The linear regression functions

$$f : \mathcal{R} \rightarrow \mathcal{R} \qquad f(x; \mathbf{w}) = w_0 + w_1 x, \;\; \text{or}$$

$$f : \mathcal{R}^d \rightarrow \mathcal{R} \qquad f(\mathbf{x}; \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_d x_d$$

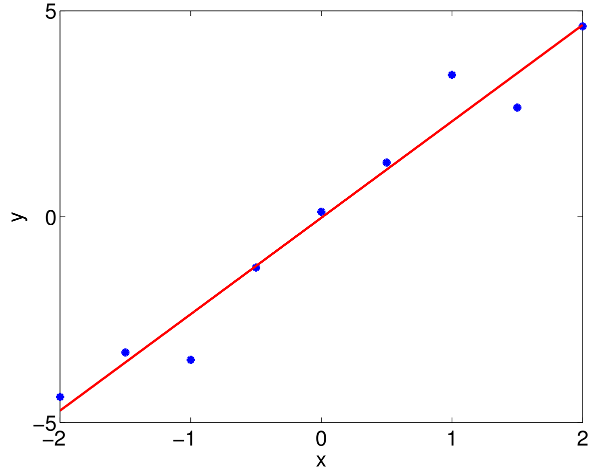
  are convenient because they are linear in the parameters, not necessarily in the input $\mathbf{x}$.

- We can easily generalize these classes of functions to be non-linear functions of the inputs $\mathbf{x}$ but still linear in the parameters $\mathbf{w}$

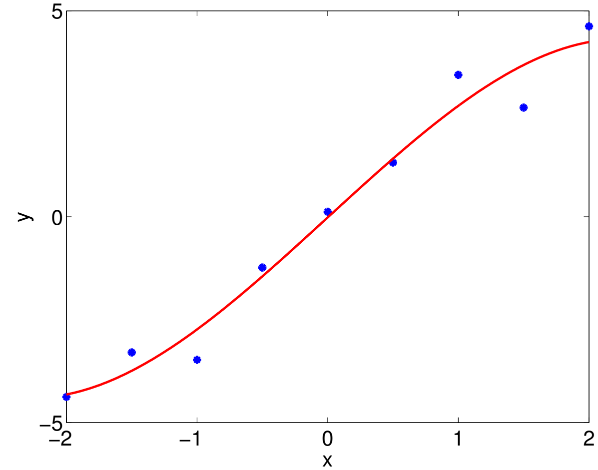
  For example: m^{th} order polynomial prediction $f : \mathcal{R} \rightarrow \mathcal{R}$

$$f(x; \mathbf{w}) = w_0 + w_1 x + \ldots + w_{m-1} x^{m-1} + w_m x^m$$

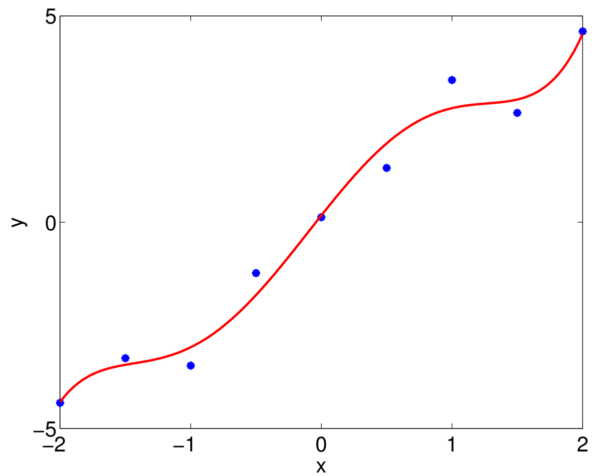
Polynomial regression: example



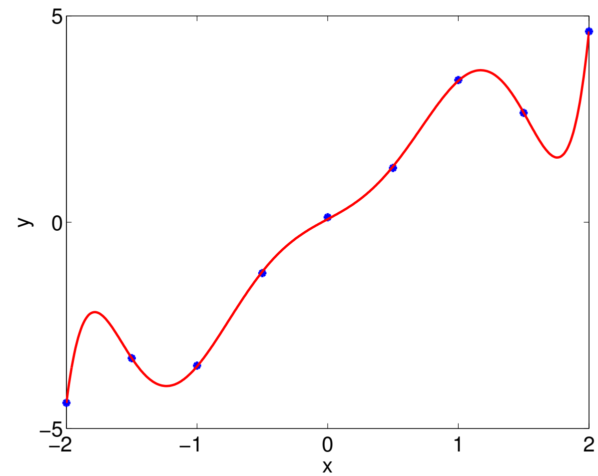
degree $= 1$



degree $= 3$



degree $= 5$



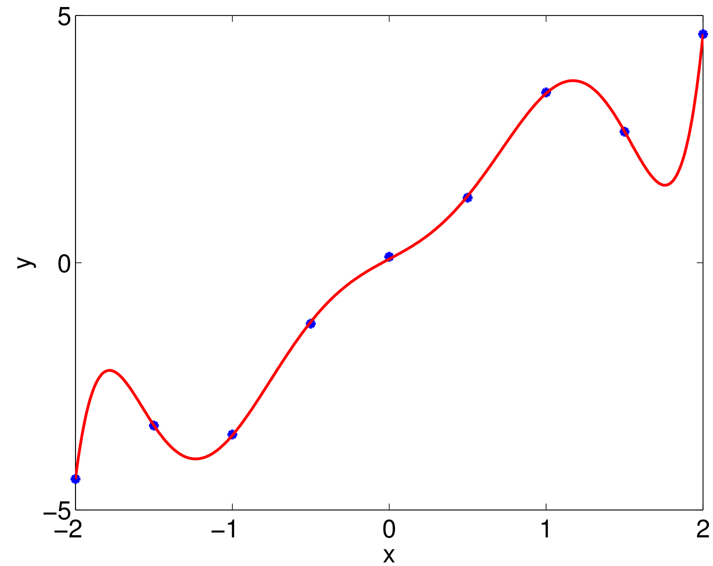
degree $= 7$

Complexity and overfitting

- With too few training examples our polynomial regression model may achieve zero training error but nevertless has a large generalization error

$$\frac{1}{n}\sum_{t=1}^{n}(y_t - f(x_t;\hat{\mathbf{w}}))^2 \approx 0$$

$$E_{(x,y)\sim P}\,(y - f(x;\hat{\mathbf{w}}))^2 \gg 0$$



- When the training error no longer bears any relation to the generalization error the function *overfits* the training data

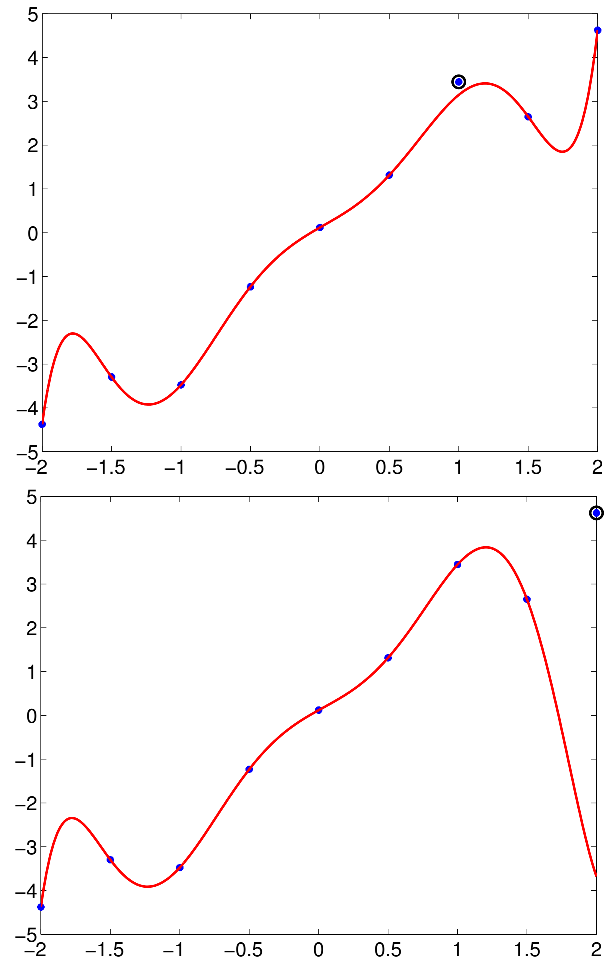
Cross-validation

- *Cross-validation* allows us to estimate the generalization error based on training examples alone

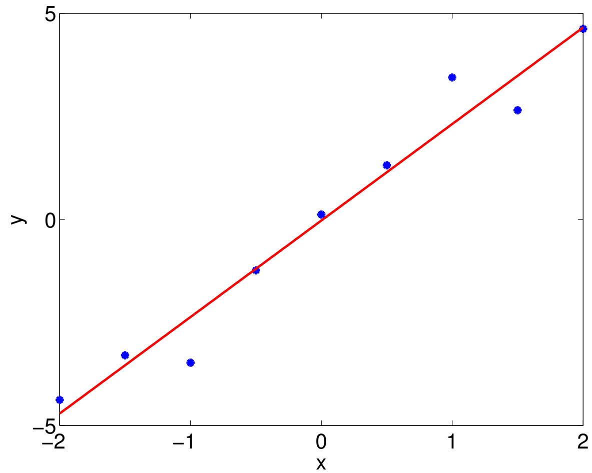
For example, the leave-one-out cross-validation error is given by

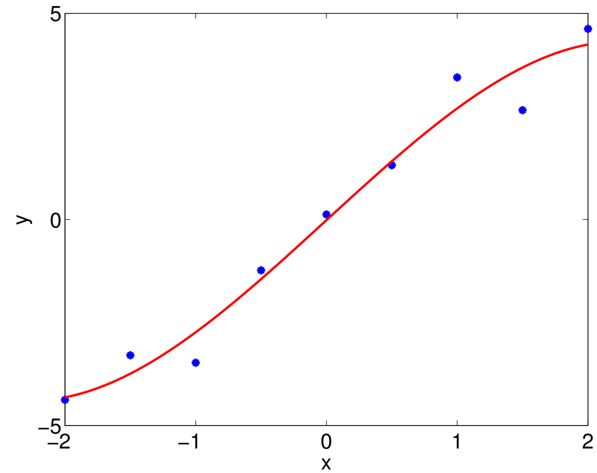
$$\text{CV} = \frac{1}{n}\sum_{t=1}^{n}\left(y_t - f(x_t;\hat{\mathbf{w}}^{-t})\right)^2$$

where $\hat{\mathbf{w}}^{-t}$ are the least squares estimates of the parameters $\mathbf{w}$ computed without the t^{th} training example.

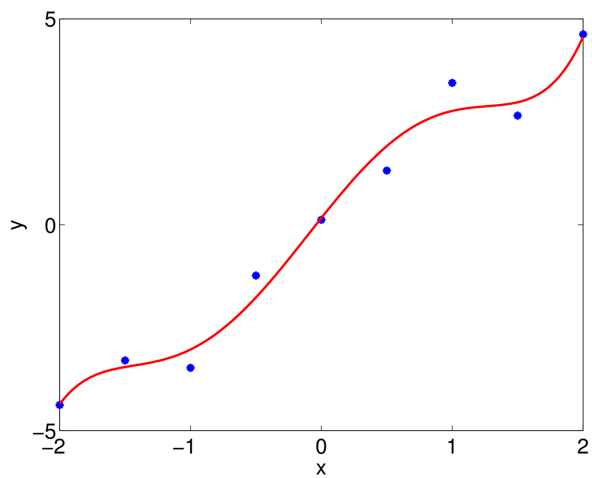


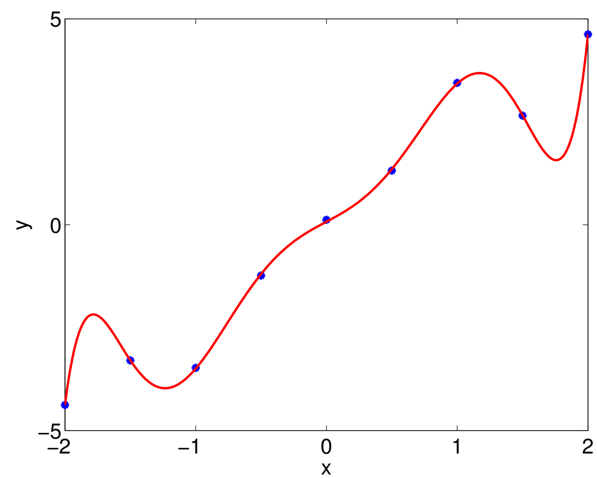
Polynomial regression: example cont'd





degree = 1, CV = 0.6 degree = 3, CV = 1.5





degree = 5, CV = 6.0 degree = 7, CV = 15.6

Additive models

- More generally, predictions can be based on a linear combination of a set of basis functions (or features) $\{\phi_1(\mathbf{x}), \ldots, \phi_m(\mathbf{x})\}$, where each $\phi_i(\mathbf{x}) : \mathcal{R}^d \to \mathcal{R}$, and

$$f(\mathbf{x}; \mathbf{w}) = w_0 + w_1\phi_1(\mathbf{x}) + \ldots + w_m\phi_m(\mathbf{x})$$

- For example:
  If $\phi_i(x) = x^i$, $i = 1, \ldots, m$, then

$$f(x; \mathbf{w}) = w_0 + w_1 x + \ldots + w_{m-1}x^{m-1} + w_m x^m$$

  If $m = d$, $\phi_i(\mathbf{x}) = x_i$, $i = 1, \ldots, d$, then

$$f(\mathbf{x}; \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_d x_d$$

Additive models cont'd

- The basis functions can capture various (e.g., qualitative) properties of the inputs.

  For example: we can try to rate companies based on text descriptions

$$
\begin{aligned}
\mathbf{x} &= \text{text document (string of words)} \\
\phi_i(\mathbf{x}) &= \begin{cases} 1 \text{ if word } i \text{ appears in the document} \\ 0 \text{ otherwise} \end{cases} \\
f(\mathbf{x};\mathbf{w}) &= w_0 + \sum_{i \in \text{words}} w_i \phi_i(\mathbf{x})
\end{aligned}
$$

Additive models cont'd

- We can also use training examples as "prototypes" and make predictions by comparing each new example to such prototypes.
- The (radial) basis functions (n of them) are now soft indicators of how close the new example is to the corresponding training example:

$$\phi_k(\mathbf{x}) \;=\; \exp\{\, -\frac{1}{2\sigma^2}\|\mathbf{x}-\mathbf{x}_k\|^2 \,\}$$

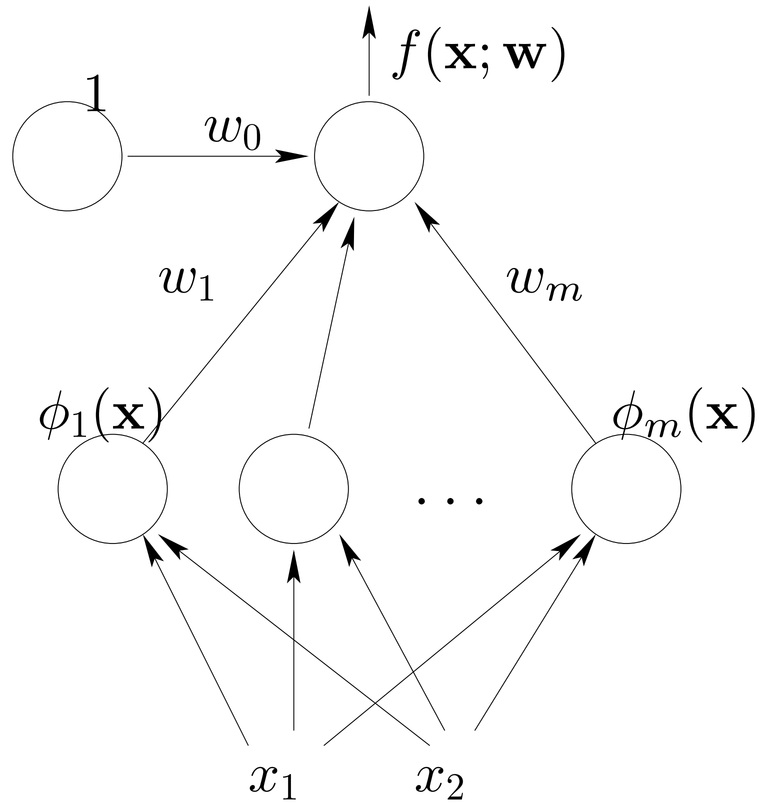
where $\mathbf{x}_k$ is the k^{th} training example and σ^2 controls how smooth the indicator is.

$$f(\mathbf{x};\mathbf{w}) = w_0 + w_1\phi_1(\mathbf{x}) + \ldots + w_n\phi_n(\mathbf{x})$$

(this class of functions depends on the training set and has many parameters; we need to *regularize* them)

Additive models: graphical view

- We can view the additive models graphically in terms of simple "units" and "weights"



- In *neural networks* the basis functions themselves have parameters and are adjustable (cf. prototypes)

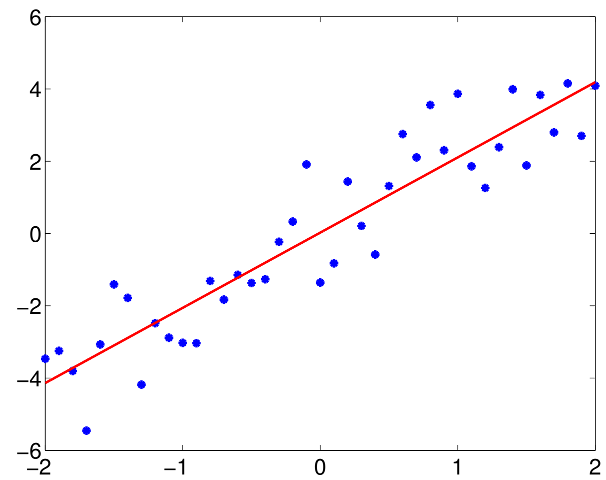
Statistical view of linear regression

- In a statistical regression model we model both the function and noise

$$\begin{aligned} \textbf{Observed output} &= \textbf{function} + \textbf{noise} \\ y &= f(\mathbf{x}; \mathbf{w}) + \epsilon \end{aligned}$$

where, e.g., $\epsilon \sim N(0, \sigma^2)$.

- Whatever we cannot capture with our chosen family of functions will be *interpreted* as noise

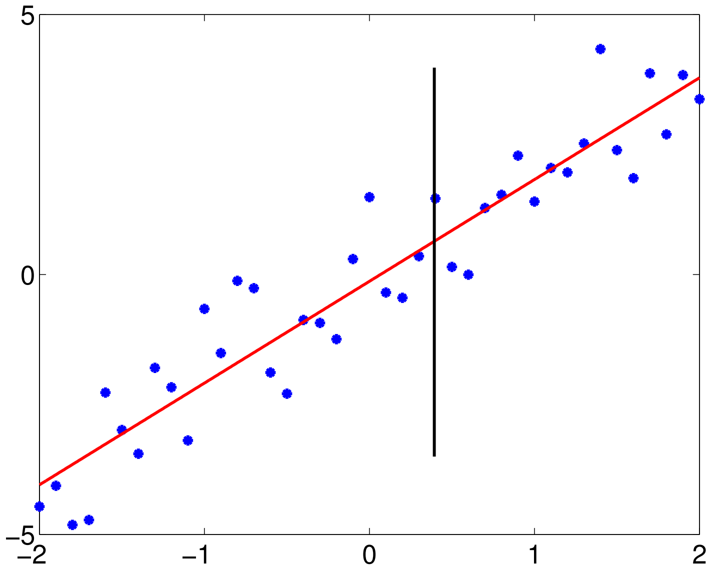


Statistical view of linear regression

- Our function $f(\mathbf{x};\mathbf{w})$ here is trying to capture the mean of the observations y given the input $\mathbf{x}$:

$$E\{\, y \,|\, \mathbf{x},\, \text{model}\} = f(\mathbf{x};\mathbf{w})$$

where $E\{\, y \,|\, \mathbf{x},\, \text{model}\}$ is the conditional expectation (mean) of y given x, evaluated according to the model.



Conditional expectation and population minimizer

- If we had no constraints on the regression function and unlimited training data in the previous regression formulation, we would minimize

$$E_{(x,y)\sim P}\,(y - f(x))^2 \;=\; E_{x\sim P_x}\left[E_{y\sim P_{y|x}}(y - f(x))^2\right]$$

where $f(x)$ can be chosen independently for each x. To find the value of $f(x)$ for each specific x, we can

$$\begin{aligned}\frac{\partial}{\partial f(x)} E_{y\sim P_{y|x}}(y - f(x))^2 &= 2E_{y\sim P_{y|x}}(y - f(x)) \\ &= 2(E\{y|x\} - f(x)) = 0\end{aligned}$$

Thus the function we are trying to approximate is

$$f^*(x) = E\{y|x\}$$

Statistical view of linear regression

- According to our statistical model

$$y = f(\mathbf{x}; \mathbf{w}) + \epsilon, \;\; \epsilon \sim N(0, \sigma^2)$$

the outputs y given $\mathbf{x}$ are normally distributed with mean $f(\mathbf{x}; \mathbf{w})$ and variance σ^2:

$$p(y|\mathbf{x}, \mathbf{w}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{\, -\frac{1}{2\sigma^2}(y - f(\mathbf{x}; \mathbf{w}))^2 \,\}$$

- As a result we can also measure the uncertainty in the predictions, not just the mean
- Loss function? Estimation?

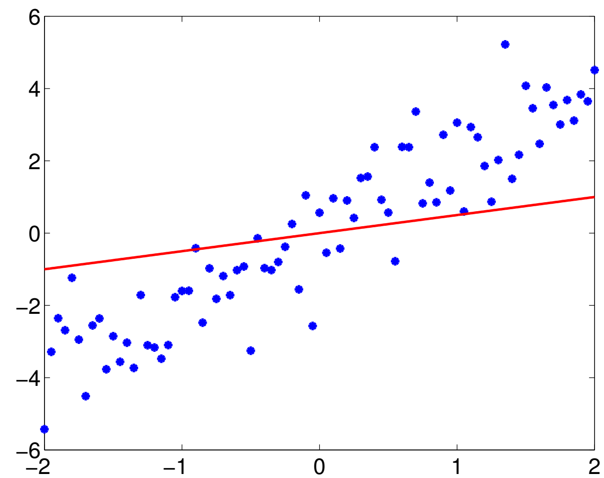
Maximum likelihood estimation

- Given observations $D_n = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ we find the parameters $\mathbf{w}$ that maximize the likelihood of the outputs

$$L(D_n; \mathbf{w}, \sigma^2) = \prod_{t=1}^{n} p(y_t | \mathbf{x}_t, \mathbf{w}, \sigma^2)$$

Example: linear function

$$p(y|\mathbf{x}, \mathbf{w}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{ -\frac{1}{2\sigma^2}(y - w_0 - w_1 x)^2 \}$$



(why is this a bad fit according to the likelihood criterion?)

Maximum likelihood estimation

Likelihood of the observed outputs:

$$L(D;\mathbf{w},\sigma^2) = \prod_{t=1}^{n} P(y_t|\mathbf{x}_t,\mathbf{w},\sigma^2)$$

- It is often easier (but equivalent) to try to maximize the log-likelihood:

$$\begin{aligned} l(D;\mathbf{w},\sigma^2) &= \log L(D;\mathbf{w},\sigma^2) = \sum_{t=1}^{n} \log P(y_t|\mathbf{x}_t,\mathbf{w},\sigma^2) \\ &= \sum_{t=1}^{n} \left(-\frac{1}{2\sigma^2}(y_t - f(\mathbf{x}_t;\mathbf{w}))^2 - \log\sqrt{2\pi\sigma^2} \right) \\ &= \left(-\frac{1}{2\sigma^2} \right) \sum_{t=1}^{n} (y_t - f(\mathbf{x}_t;\mathbf{w}))^2 + \ldots \end{aligned}$$

Maximum likelihood estimation cont'd

- Our model of the noise in the outputs and the resulting (effective) loss-function in maximum likelihood estimation are intricately related

$$\mathrm{Loss}(y, f(\mathbf{x}; \mathbf{w})) = -\log P(y|\mathbf{x}, \mathbf{w}, \sigma^2) + \text{ const.}$$

Maximum likelihood estimation cont'd

- The likelihood of observations

$$L(D; \mathbf{w}, \sigma^2) = \prod_{t=1}^{n} P(y_t | \mathbf{x}_t, \mathbf{w}, \sigma^2)$$

  is a generic fitting criterion.

- We can just as easily fit the noise variance σ^2 by maximizing the log-likelihood $l(D; \mathbf{w}, \sigma^2)$ with respect to σ^2

Maximum likelihood estimation cont'd

- The likelihood of observations

$$L(D;\mathbf{w},\sigma^2) = \prod_{t=1}^{n} P(y_t|\mathbf{x}_t,\mathbf{w},\sigma^2)$$

is a generic fitting criterion.

- We can just as easily fit the noise variance σ^2 by maximizing the log-likelihood $l(D;\mathbf{w},\sigma^2)$ with respect to σ^2

if $\hat{\mathbf{w}}$ are the maximum likelihood parameters for $f(\mathbf{x};\mathbf{w})$, then the optimal choice for σ^2 is

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{t=1}^{n}(y_t - f(\mathbf{x}_t;\hat{\mathbf{w}}))^2$$

i.e., mean squared prediction error.