6.867 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
  - population minimizer
- Statistical regression models
  - model formulation, motivation
  - maximum likelihood estimation

Linear regression

- Linear regression functions,

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

  combined with the squared loss, are convenient because they are *linear in the parameters*.

Linear regression

- Linear regression functions,

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

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

combined with the squared loss, are convenient because they are *linear in the parameters*.

  - we get closed form estimates of the parameters

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

where, for example, $\mathbf{y} = [y_1, \ldots, y_n]^T$.

Linear regression

- Linear regression functions,

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

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

combined with the squared loss, are convenient because they are *linear in the parameters*.

– we get closed form estimates of the parameters

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

where, for example, $\mathbf{y} = [y_1, \ldots, y_n]^T$.

– the resulting prediction errors $\epsilon_i = y_i - f(\mathbf{x}_i; \hat{\mathbf{w}})$ are uncorrelated with any linear function of the inputs $\mathbf{x}$.

Linear regression

- Linear regression functions,

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

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

  combined with the squared loss, are convenient because they are *linear in the parameters*.

  – we get closed form estimates of the parameters

  $$\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

  where, for example, $\mathbf{y} = [y_1, \ldots, y_n]^T$.

  – the resulting prediction errors $\epsilon_i = y_i - f(\mathbf{x}_i; \hat{\mathbf{w}})$ are uncorrelated with any linear function of the inputs $\mathbf{x}$.

  – we can easily extend these to non-linear functions of the inputs while still keeping them linear in the parameters

Beyond linear regression

- Example extension: m^{th} order polynomial regression where $f : \mathcal{R} \rightarrow \mathcal{R}$ is given by

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

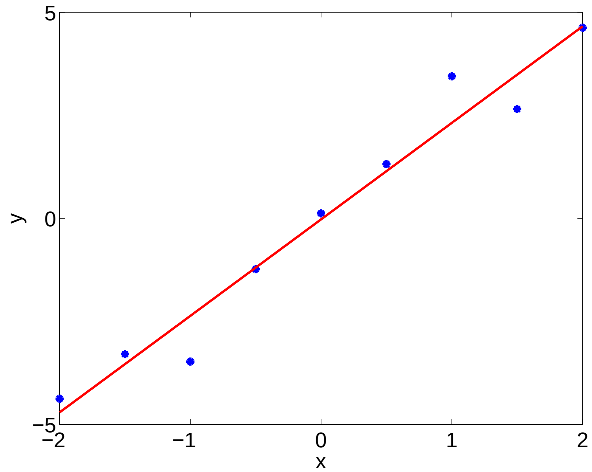
  - linear in the parameters, non-linear in the inputs
  - solution as before

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

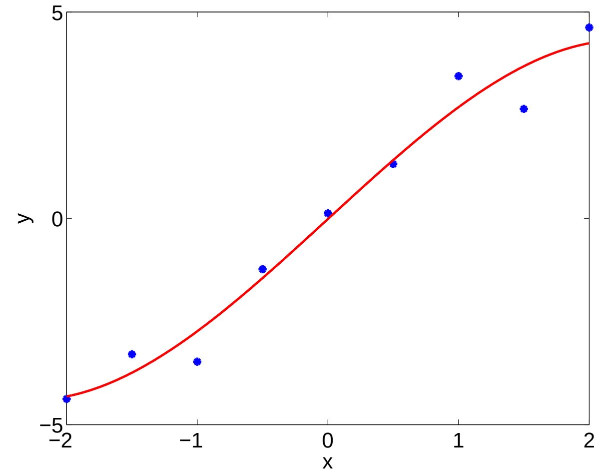
where

$$\hat{\mathbf{w}} = \begin{bmatrix} \hat{w}_0 \\ \hat{w}_1 \\ \ldots \\ \hat{w}_m \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^m \\ 1 & x_2 & x_2^2 & \ldots & x_2^m \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_n & x_n^2 & \ldots & x_n^m \end{bmatrix}$$

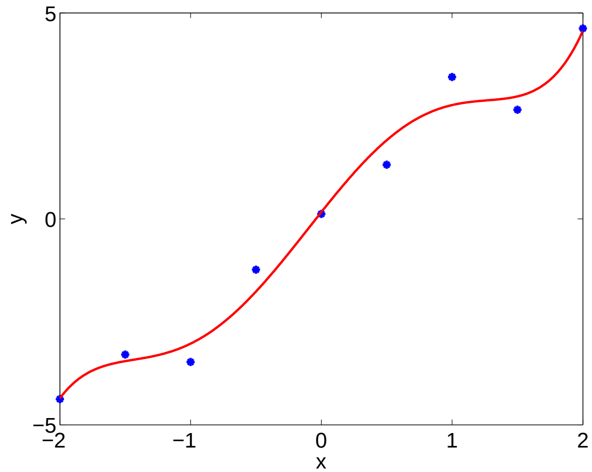
Polynomial regression



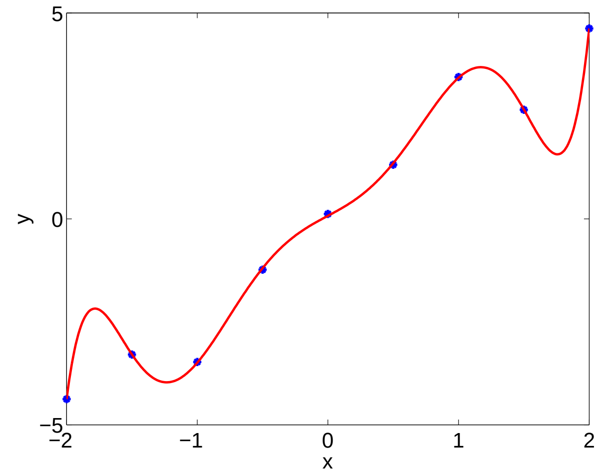
degree $= 1$



degree $= 3$



degree $= 5$



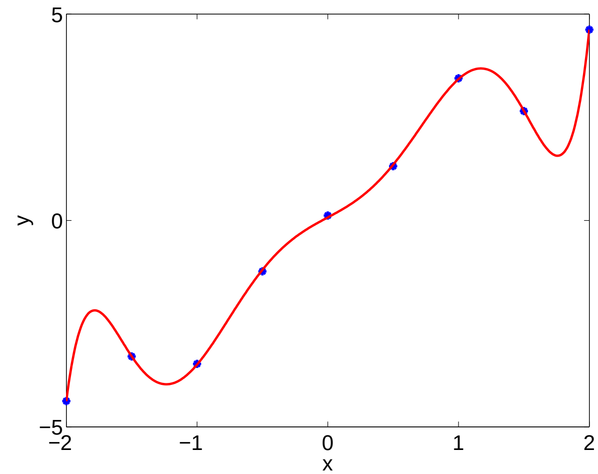
degree $= 7$

Complexity and overfitting

- With limited training examples our polynomial regression model may achieve zero training error but nevertless has a large test (generalization) error

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

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



- We suffer from *over-fitting* when the training error no longer bears any relation to the generalization error

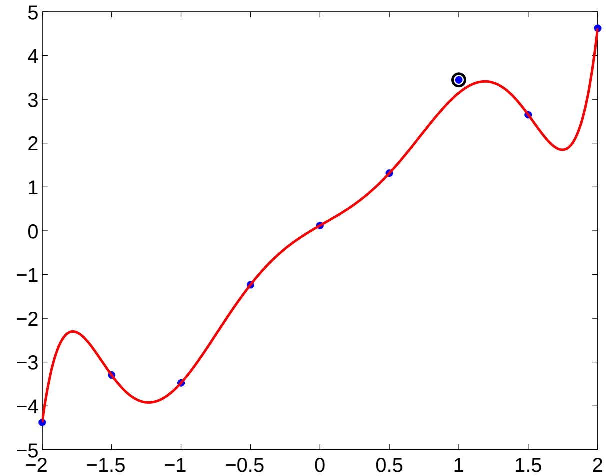
Avoiding over-fitting: cross-validation

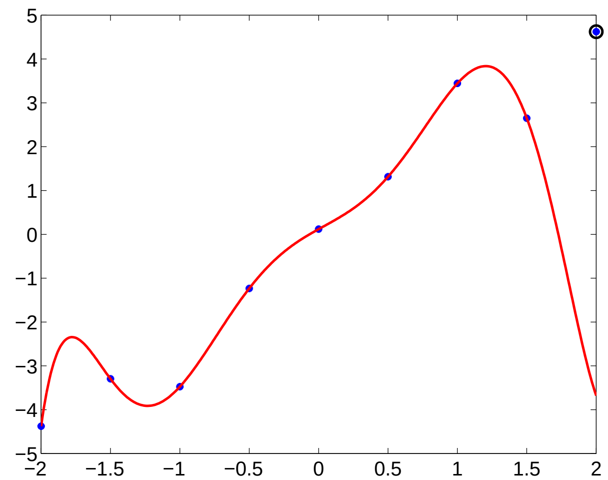
- *Cross-validation* allows us to estimate the generalization error based on training examples alone

Leave-one-out cross-validation treats each training example in turn as a test example:

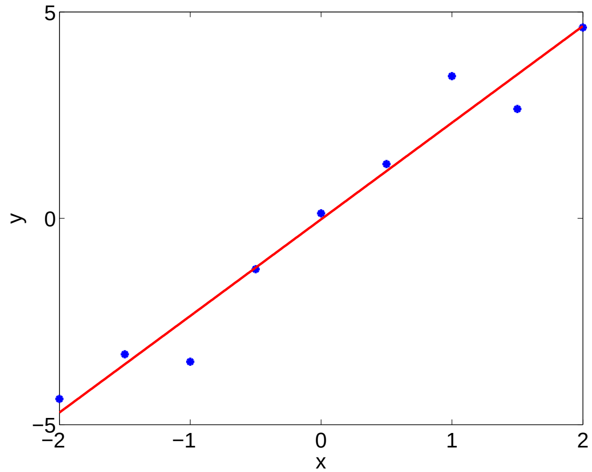
$$\mathrm{CV} = \frac{1}{n}\sum_{i=1}^{n}\Big(y_i - f(x_i;\hat{\mathbf{w}}^{-i})\Big)^2$$

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

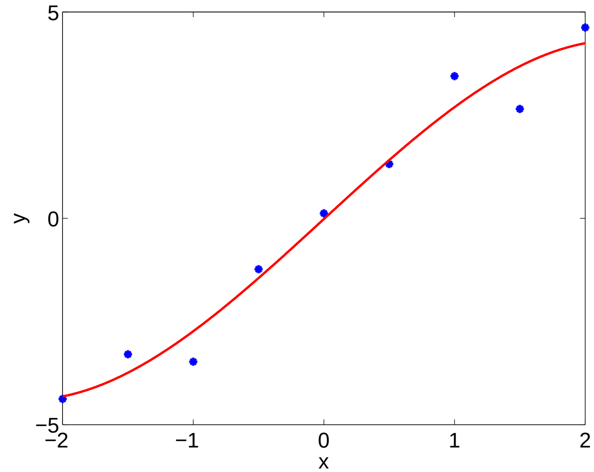




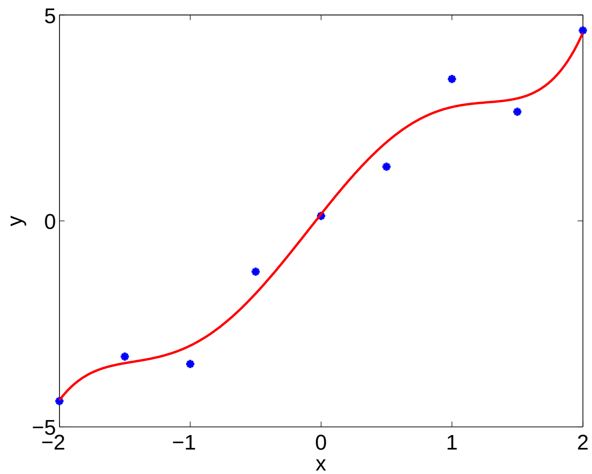
Polynomial regression: example cont'd



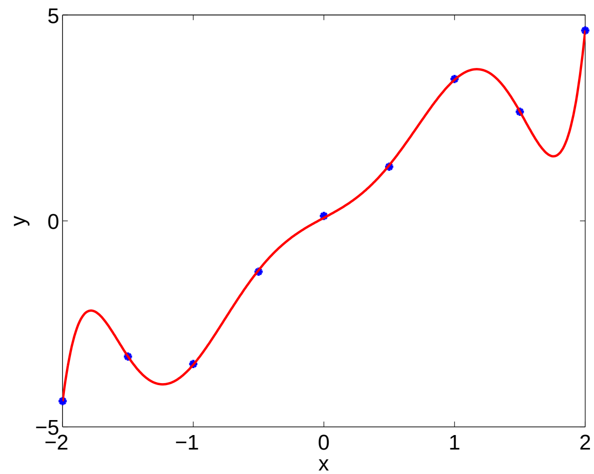
degree = 1, CV = 0.6



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 $\phi_i(\mathbf{x}) : \mathcal{R}^d \rightarrow \mathcal{R}$, and

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

- Examples:

  If $\phi_i(x) = x^i$, $i = 1, \ldots, m$, then

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

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

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

- Examples:

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

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

Additive models cont'd

- We can also make predictions by gauging the similarity of examples to "prototypes".

  For example, our additive regression function could be

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

  where the basis functions are "radial basis functions"

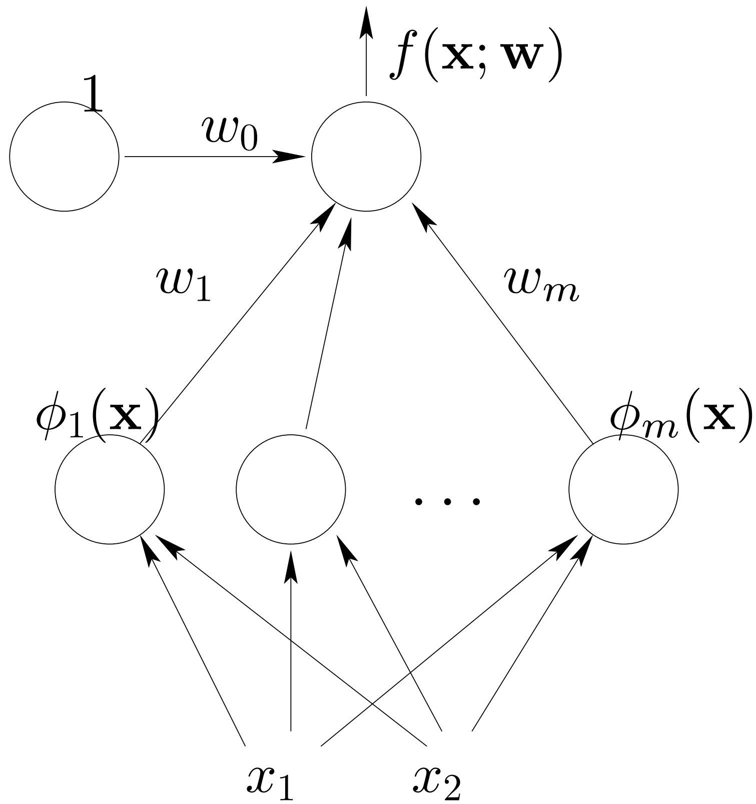
  $$\phi_k(\mathbf{x}) \;\; = \;\; \exp\{\, -\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{x}_k\|^2\,\}$$

  measuring the similarity to the prototypes; σ^2 controls how quickly the basis function vanishes as a function of the distance to the prototype.

  (training examples themselves could serve as prototypes)

Additive models cont'd

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



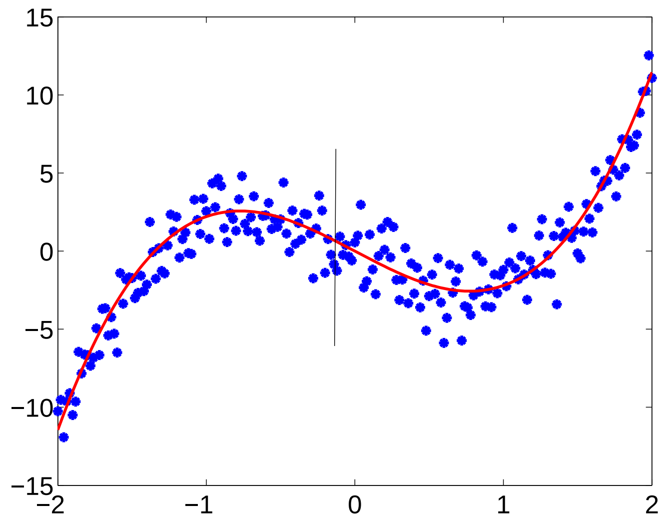
- In *neural networks* the basis functions themselves have adjustable parameters (cf. prototypes)

Squared loss and population minimizer

- What do we get if we have unlimited training examples (the whole population) and no constraints on the regression function?

$$\text{minimize} \quad E_{(x,y)\sim P}\,(y - f(x))^2$$

with respect to an unconstrained function $f : \mathcal{R} \to \mathcal{R}$



Squared loss and population minimizer

- To minimize

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

we can focus on each x separately since $f(x)$ can be chosen independently for each different x. For any particular 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 the conditional expectation

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

Topics

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

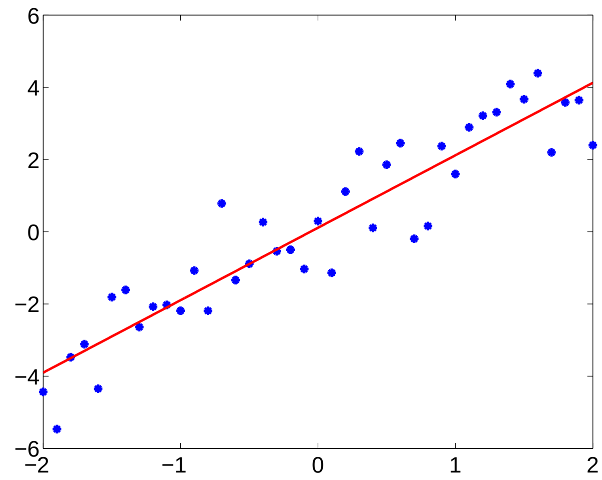
Statistical view of linear regression

- 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

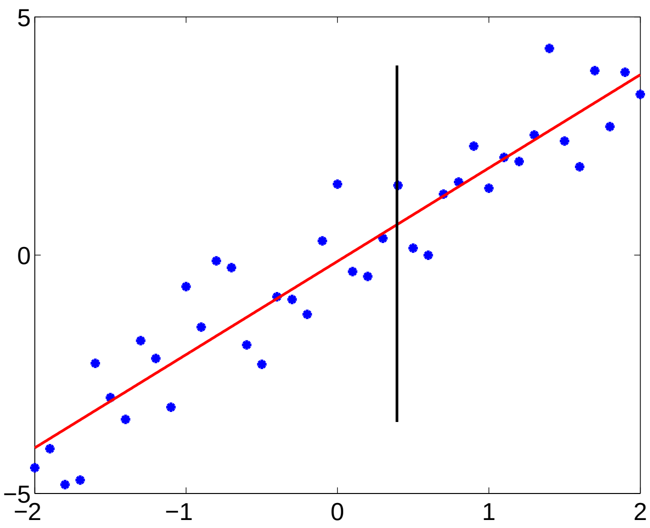


Statistical view of linear regression

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

$$\begin{aligned} E\{\, y \,|\, \mathbf{x} \,\} &= E\{\, f(\mathbf{x};\mathbf{w}) + \epsilon \,|\, \mathbf{x} \,\} \\ &= f(\mathbf{x};\mathbf{w}) \end{aligned}$$

where $E\{\, y \,|\, \mathbf{x} \,\}$ is the conditional expectation of y given $\mathbf{x}$, evaluated according to the model (not according to the underlying distribution P)



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:

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

(we model the uncertainty in the predictions, not just the mean)

- Loss function? Estimation?

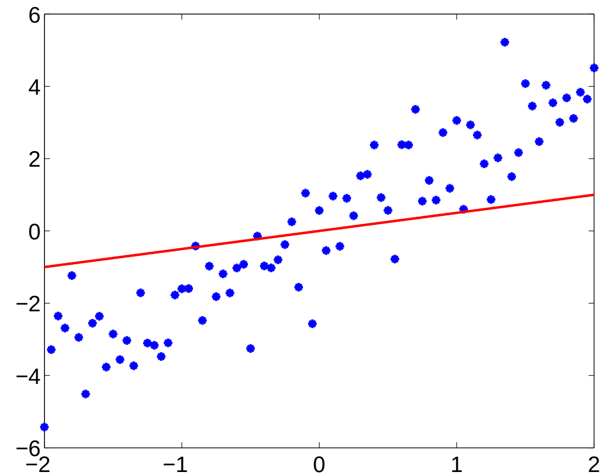
Maximum likelihood estimation

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

$$L(D_n; \mathbf{w}, \sigma^2) = \prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \mathbf{w}, \sigma^2)$$

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 cont'd

Likelihood of the observed outputs:

$$L(D; \mathbf{w}, \sigma^2) = \prod_{i=1}^{n} P(y_i|\mathbf{x}_i, \mathbf{w}, \sigma^2)$$

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

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

Maximum likelihood estimation cont'd

- Maximizing log-likelihood is equivalent to minimizing empirical loss when the loss is defined according to

$$\text{Loss}(y_i, f(\mathbf{x}_i; \mathbf{w})) = -\log P(y_i | \mathbf{x}_i, \mathbf{w}, \sigma^2)$$

  Loss defined as the negative log-probability is known as the *log-loss*.

Maximum likelihood estimation cont'd

- The log-likelihood of observations

$$\log L(D; \mathbf{w}, \sigma^2) = \sum_{i=1}^{n} \log P(y_i | \mathbf{x}_i, \mathbf{w}, \sigma^2)$$

  is a generic fitting criterion and can be used to estimate the noise variance σ^2 as well.

- Let $\hat{\mathbf{w}}$ be the maximum likelihood (here least squares) setting of the parameters. What is the maximum likelihood estimate of σ^2, obtained by solving

$$\frac{\partial}{\partial \sigma^2} \log L(D; \mathbf{w}, \sigma^2) = 0 \quad ?$$

Maximum likelihood estimation cont'd

- The log-likelihood of observations

$$\log L(D; \mathbf{w}, \sigma^2) = \sum_{i=1}^{n} \log P(y_i | \mathbf{x}_i, \mathbf{w}, \sigma^2)$$

  is a generic fitting criterion and can be used to estimate the noise variance σ^2 as well.

- Let $\hat{\mathbf{w}}$ be the maximum likelihood (here least squares) setting of the parameters. The maximum likelihood estimate of the noise variance σ^2 is

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i; \hat{\mathbf{w}}))^2$$

  i.e., the mean squared prediction error.