# Machine learning: lecture 3

Tommi S. Jaakkola
MIT AI Lab
*tommi@ai.mit.edu*

# Topics

- Linear regression
  - overfitting, cross-validation
- Additive models
  - polynomial regression, other basis functions
- Statistical view of regression
  - noise model
  - likelihood, maximum likelihood estimation
  - limitations

# Review: generalization

- The "generalization" error

$$E_{(x,y)\sim P}\left\{(y-\hat{w}_0-\hat{w}_1x)^2\right\}$$

is a sum of two terms:

1. error of the best predictor in the class

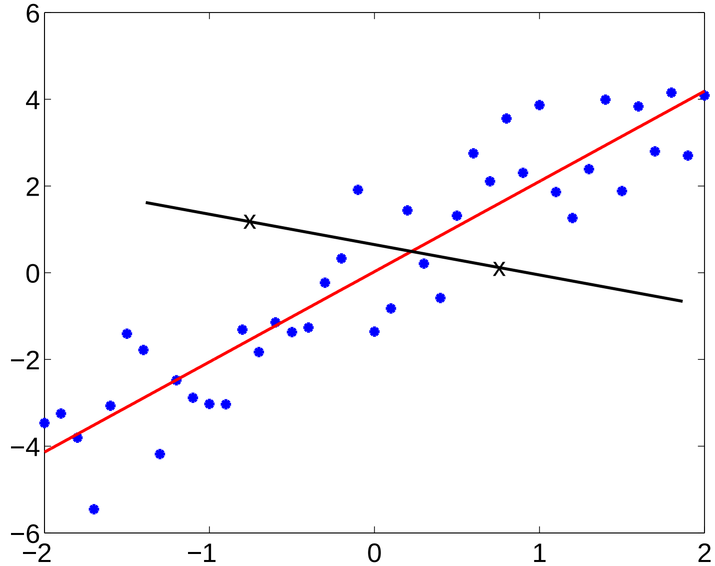
$$\begin{aligned} & E_{(x,y)\sim P}\left\{(y-w_0^*-w_1^*x)^2\right\} \\ = & \min_{w_0,w_1} E_{(x,y)\sim P}\left\{(y-w_0-w_1x)^2\right\} \end{aligned}$$

2. and how well we approximate the best linear predictor based on a limited training set

$$E_{(x,y)\sim P}\left\{\Big((w_0^*+w_1^*x)-(\hat{w}_0+\hat{w}_1x)\Big)^2\right\}$$

# Overfitting

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




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

# Cross-validation

- *Cross-validation* allows us to estimate generalization error on the basis of only the training set

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

$$\text{CV} = \frac{1}{n} \sum_{i=1}^{n} \big( y_i - (\hat{w}_0^{-i} + \hat{w}_1^{-i} x_i) \big)^2$$

  where $(\hat{w}_0^{-i}, \hat{w}_1^{-i})$ are least squares estimates computed without the $i^{th}$ training example.

# Extensions of linear regression: additive models

- Our previous results generalize to models that are linear in the parameters $\mathbf{w}$, not necessarily in the inputs $\mathbf{x}$

1. Simple linear prediction $f : \mathcal{R} \rightarrow \mathcal{R}$

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

2. $m^{th}$ order polynomial prediction $f : \mathcal{R} \rightarrow \mathcal{R}$

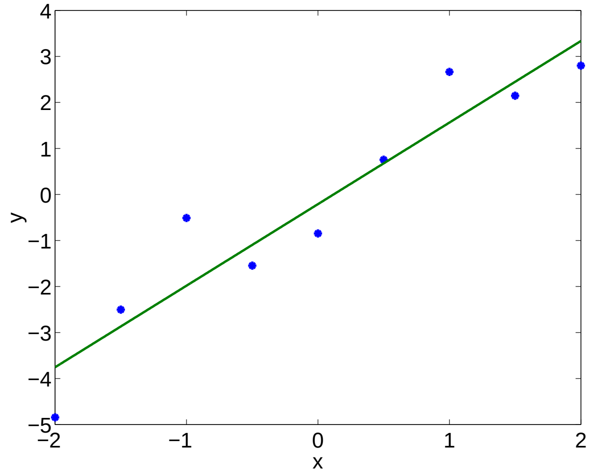
$$f(x; \mathbf{w}) = w_0 + w_1 x + \ldots + w_{m-1} x^{m-1} + w_m x^m$$

3. Multi-dimensional linear prediction $f : \mathcal{R}^d \rightarrow R$

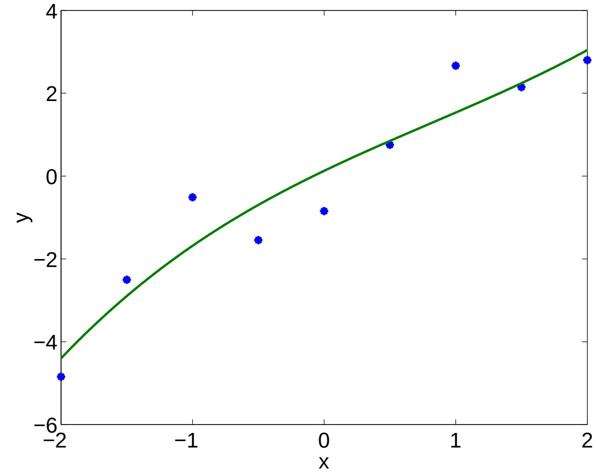
$$f(\mathbf{x}; \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_{d-1} x_{d-1} + w_d x_d$$

where $\mathbf{x} = [x_1 \ldots x_{d-1} \; x_d]^T$, $d = dim(\mathbf{x})$

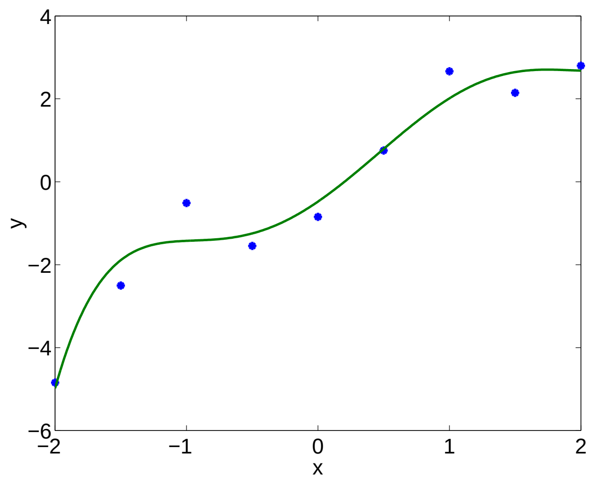
# Polynomial regression: example



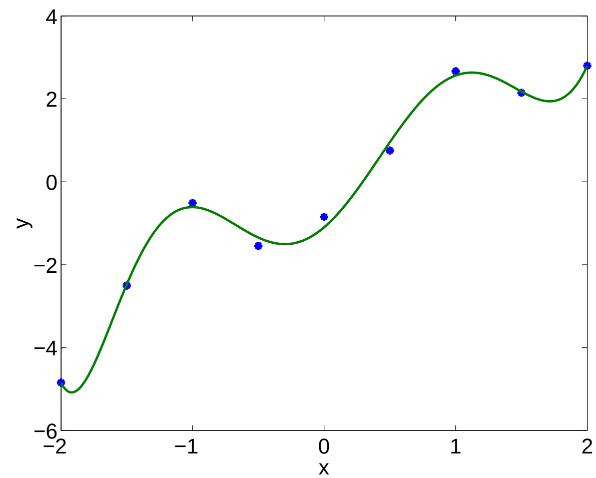

degree $= 1$




degree $= 3$

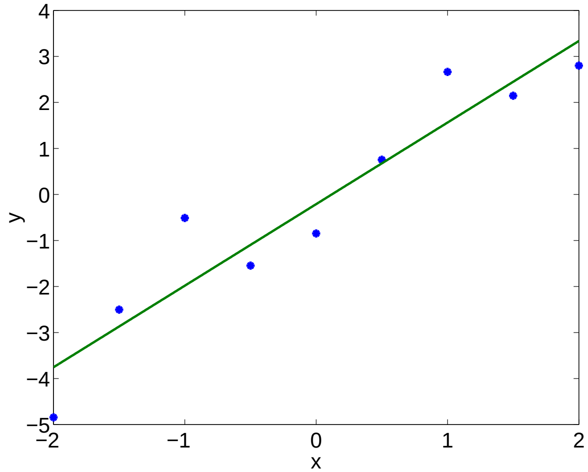



degree $= 5$

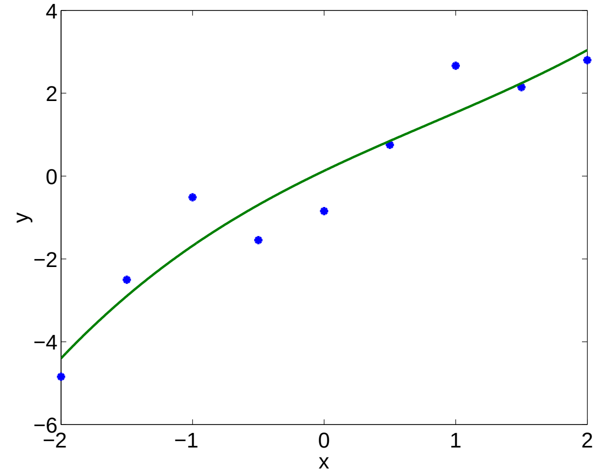



degree $= 7$

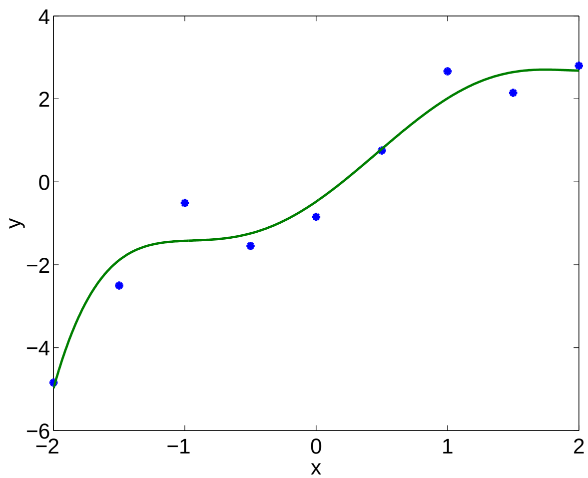
# Polynomial regression: example cont'd



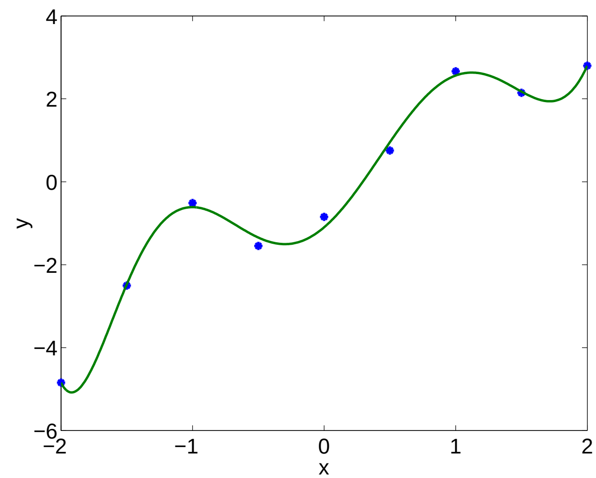

degree = 1, CV = 1.1




degree = 3, CV = 2.6




degree = 5, CV = 44.2




degree = 7, CV = 482.0

# Additive models cont'd

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

$$f(\mathbf{x}; \mathbf{w}) = w_0 + w_1 \phi_1(\mathbf{x}) + \ldots + w_{m-1} \phi_{m-1}(\mathbf{x}) + w_m \phi_m(\mathbf{x})$$

- 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

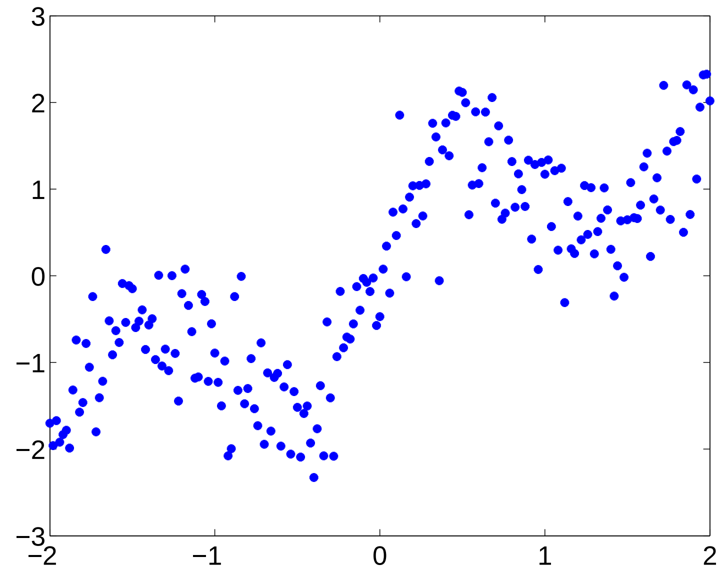
$$f(\mathbf{x}; \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_{d-1} x_{d-1} + w_d x_d$$

# Additive models cont'd

- Example: it is often useful to find "prototypical" input vectors $\mu_1, \ldots, \mu_m$ that exemplify different "contexts" for prediction

  We can define basis functions (one for each prototype) that measure how close the the input vector $\mathbf{x}$ is to the prototype

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



# Additive models cont'd

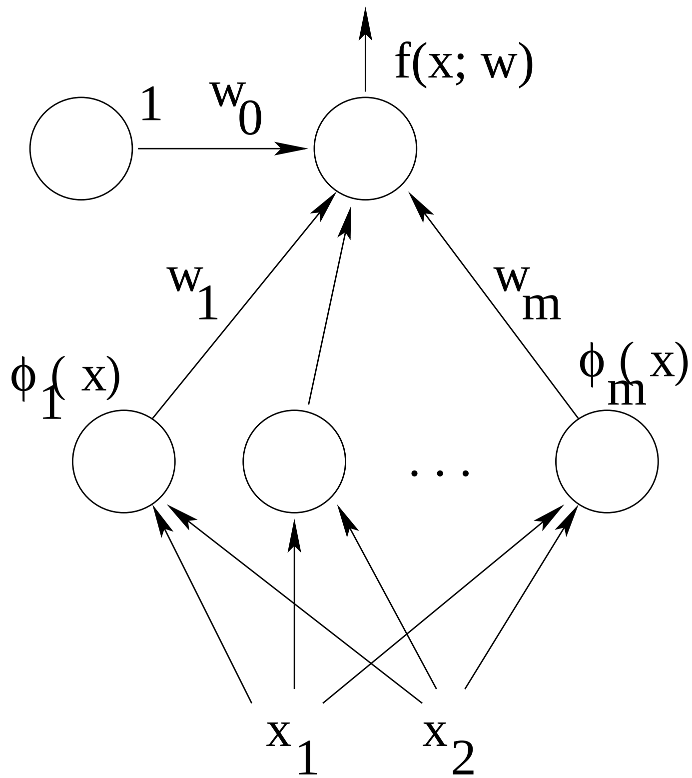
- The basis functions can capture various (e.g., qualitative) properties of the inputs.

  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

- Graphical representation of additive models (cf. neural networks):



# Statistical view of linear regression

- A statistical regression model

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

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

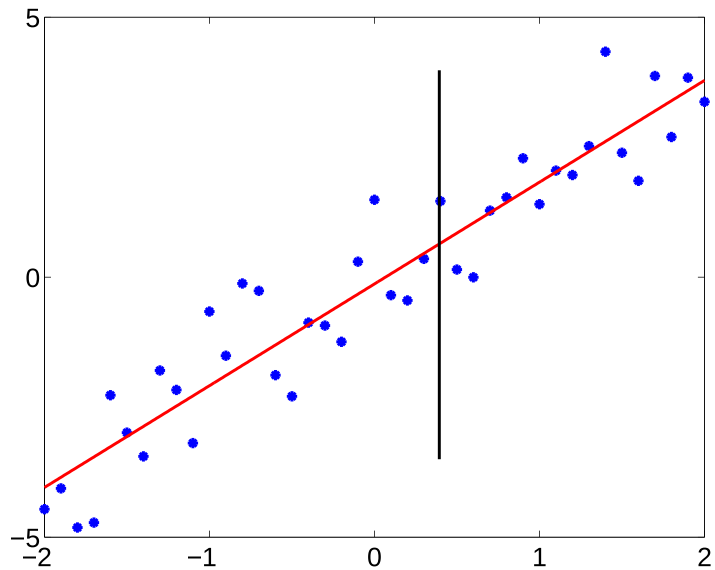
- Whatever we cannot capture with our chosen family of functions will be *interpreted* as noise

# Statistical view of linear regression

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

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

The expectation is taken with respect to $P$ that governs the underlying (and typically unknown) relation between $x$ and $y$.



# 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 $\sigma^2$:

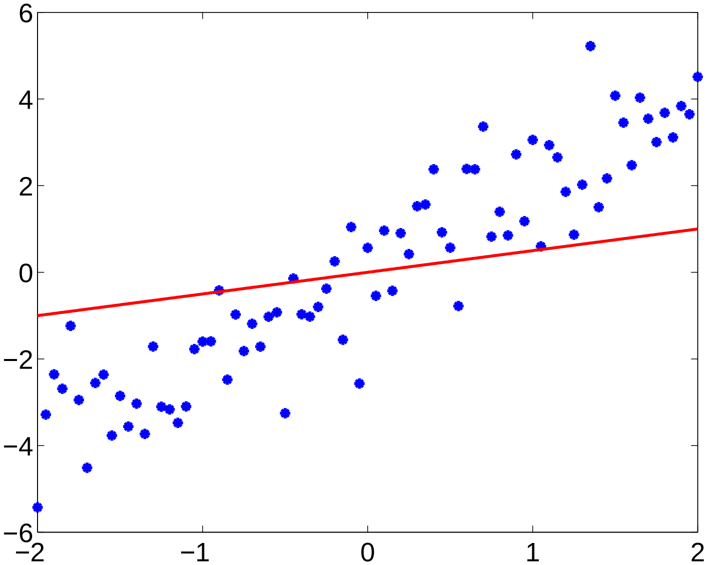
$$P(y|\mathbf{x}, \mathbf{w}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{ -\frac{1}{2\sigma^2}(y - f(\mathbf{x}; \mathbf{w}))^2 \}$$

- As a result we can also measure the uncertainty in the predictions (through variance $\sigma^2$), not just the mean
- Loss function? Estimation?

# Maximum likelihood estimation

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

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




Why is this a bad fit according to the likelihood criterion?

# Maximum likelihood estimation

Likelihood of the observed outputs:

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

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

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

# Maximum likelihood estimation cont'd

- The noise distribution and the loss-function are intricately related

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

# Maximum likelihood estimation cont'd

- The likelihood of the observed outputs

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

provides a general measure of how the model fits the data. On the basis of this measure, we can estimate the noise variance $\sigma^2$ as well as the weights $\mathbf{w}$.

Can we find a rationale for what the "optimal" noise variance should be?

# Maximum likelihood estimation cont'd

- To estimate the parameters $\mathbf{w}$ and $\sigma^2$ quantitatively, we maximize the log-likelihood with respect to all the parameters

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}} l(D; \mathbf{w}, \sigma^2) &= 0 \\ \frac{\partial}{\partial \sigma^2} l(D; \mathbf{w}, \sigma^2) &= 0 \end{aligned}$$

  The resulting noise variance $\hat{\sigma}^2$ is given by

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

  where $\hat{\mathbf{w}}$ is the *same* ML estimate of $\mathbf{w}$ as before.

  Interpretation: this is the mean squared prediction error (on the training set) of the best linear predictor.

# Brief derivation

Consider the log-likelihood evaluated at $\hat{\mathbf{w}}$

$$l(D;\hat{\mathbf{w}},\sigma^2) \;=\; \left(-\frac{1}{2\sigma^2}\right)\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i;\hat{\mathbf{w}}))^2 - \frac{n}{2}\log(2\pi\sigma^2)$$

(need to justify first that we can simply substitute in the ML solution $\hat{\mathbf{w}}$ rather than perform joint maximization)

Now,

$$\frac{\partial}{\partial\sigma^2} l(D;\hat{\mathbf{w}},\sigma^2) \;=\; \left(\frac{1}{2\sigma^4}\right)\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i;\hat{\mathbf{w}}))^2 - \frac{n}{2\sigma^2} = 0$$

and we get the solution by multiplying both sides by $2\sigma^4/n$.

# Cross-validation and log-likelihood

Leave-one-out cross-validated log-likelihood:

$$\text{CV} = \sum_{i=1}^{n} \log P(y_i | \mathbf{x}_i, \hat{\mathbf{w}}^{-i}, (\hat{\sigma}^2)^{-i})$$

where $\hat{\mathbf{w}}^{-i}$ and $(\hat{\sigma}^2)^{-i}$ are maximum likelihood estimates computed without the $i^{th}$ training example $(\mathbf{x}_i, y_i)$.

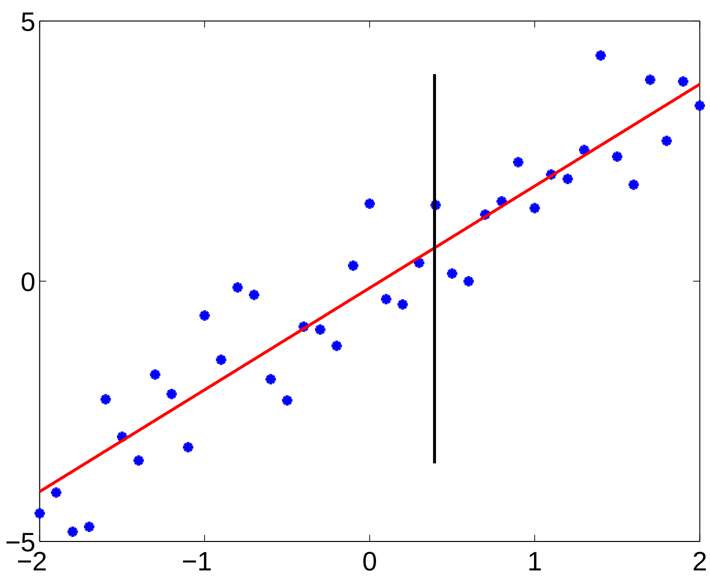
# Some limitations

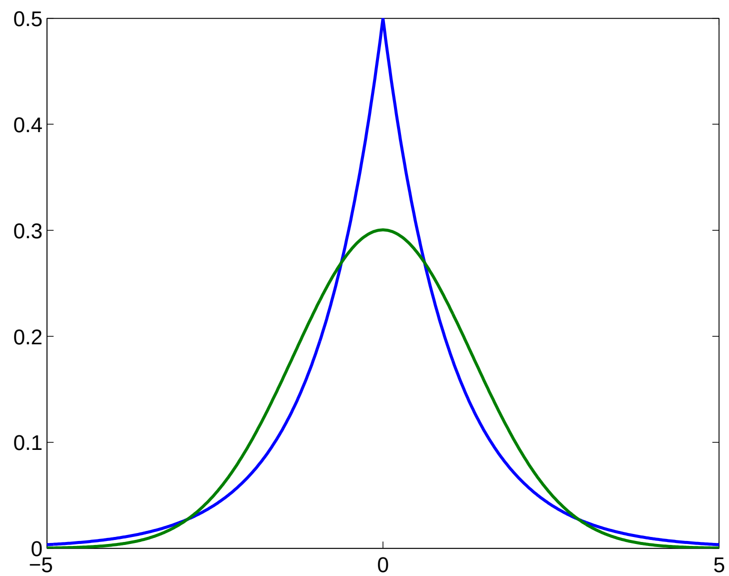
- The simple statistical model

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

is not always appropriate or useful.

Example: noise may not be Gaussian



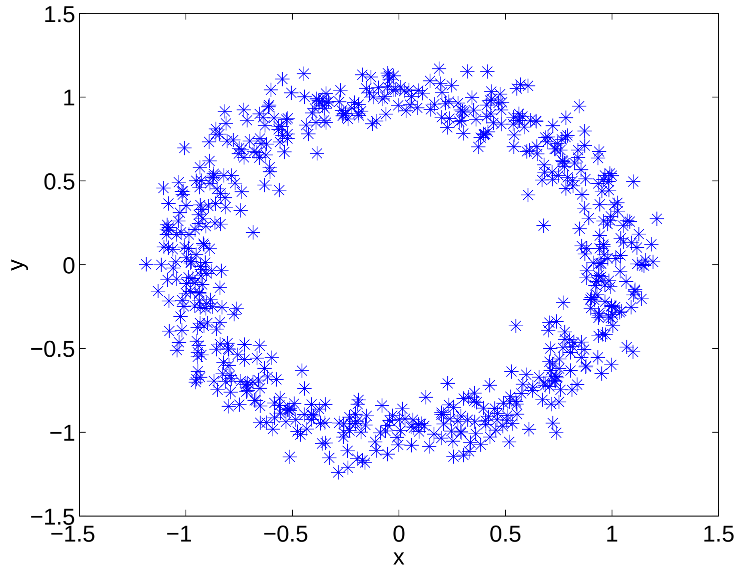



# Limitations cont'd

- It may not even be possible (or at all useful) to model the data with

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

no matter how flexible the function class $f(\cdot; \mathbf{w})$, $\mathbf{w} \in \mathcal{W}$ is. Example:




(note: this is NOT a limitation conditional models $P(y|\mathbf{x}, \mathbf{w})$ more generally)