Linear Models For Regression, Part 1

Kim Steenstrup Pedersen

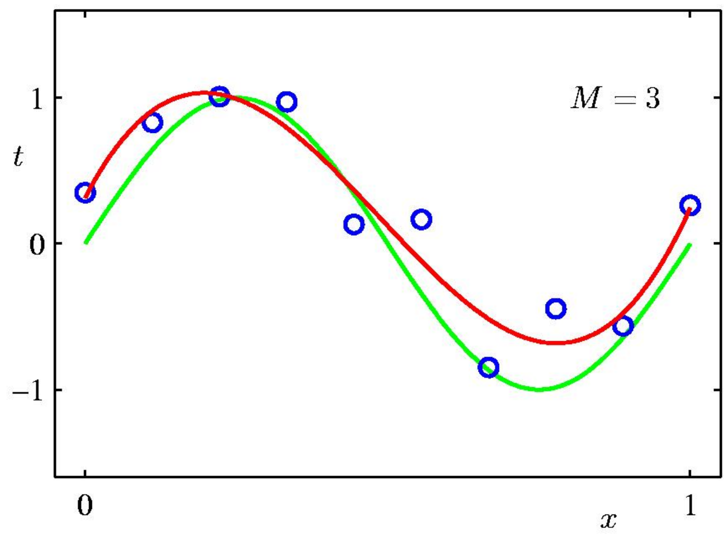
Plan for lecture

- Motivation for using regression in automated data analysis
- A curve fitting example
- Curve fitting revisited - a Bayesian probabilistic interpretation
- Linear models for regression

Regression: A supervised learning problem

- ***We have*** a data set consisting of *N* pairs ($\boldsymbol{x}_n$,$\boldsymbol{t}_n$) of observations $\mathbf{x} \in R^D$ and corresponding observed target values $\mathbf{t} \in R^K$. We assume there exist a functional relationship between them
$$\mathbf{t}(\mathbf{x}) : R^D \rightarrow R^K$$
- ***We want*** to learn a model that allows us to predict target values:
$$\mathbf{y}(\mathbf{x}) : R^D \rightarrow R^K$$
- Prediction?
  - Either interpolation or extrapolation from data using $\boldsymbol{y}(\boldsymbol{x})$.
  - We aim to model the predictive distribution $p(\mathbf{t} \mid \mathbf{x})$ and apply it for predictions of the target value for any observations.

Polynomial example:



Blue circles show (x_n,t_n), green curve the “truth” and red curve show a model y(x) with D=1 and K=1.

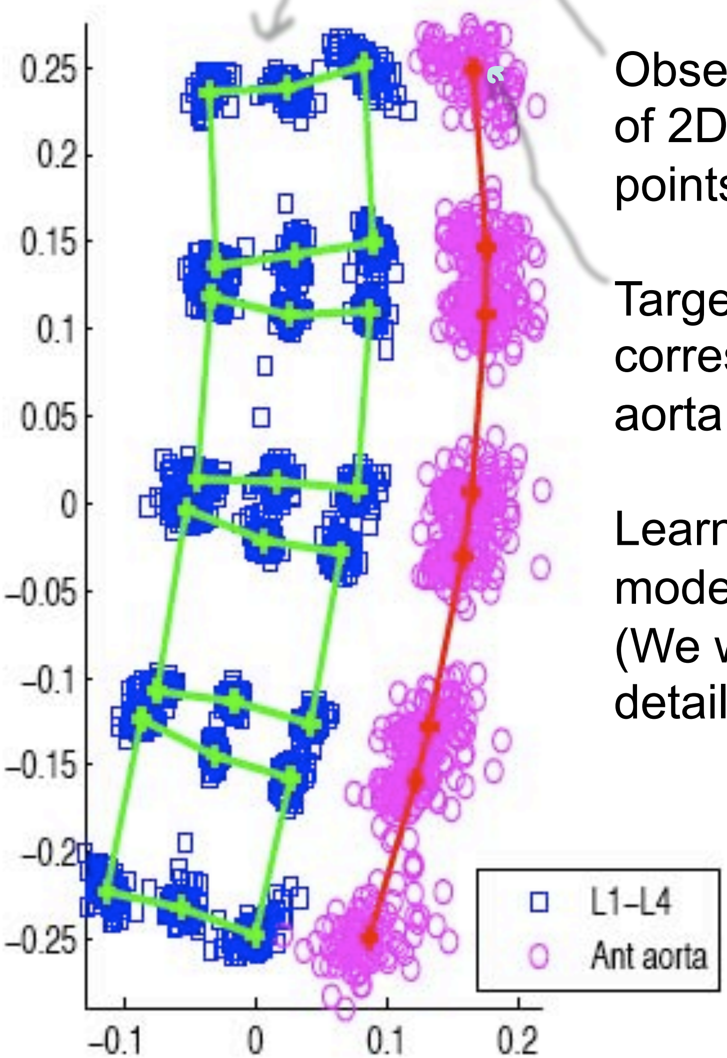
NOTE: In general, it is important that training observations represent typical samples / observations, otherwise we cannot hope to model the problem.

Example: Predicting Aorta Wall Location in X-ray Images

- Predict the location of the spinal aorta walls conditioned on the vertebra location.
- Needed for quantification of aorta calcification – aorta area vs. calcification area.
  - Hard problem because soft tissue is not visible in x-rays, but calcifications are!
- Use a shape model of vertebrae and linear regression with vertebrae locations as input and aorta wall locations as target.

(Data from Ph.D. Thesis of Lars Arne Conrad-Hansen, ITU, 2006)

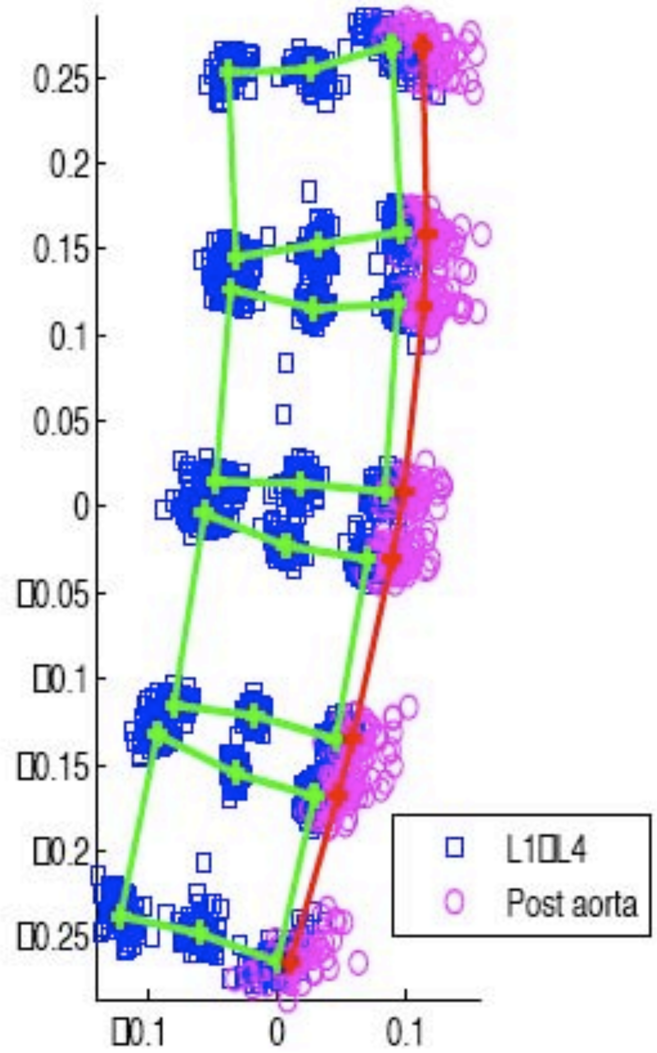
Example: Training Set and Posterior Mean Shape



Observation vector x_n consist of 2D position of 3 vertebrae points.

Target variable t_n is the corresponding 2D location of aorta point.

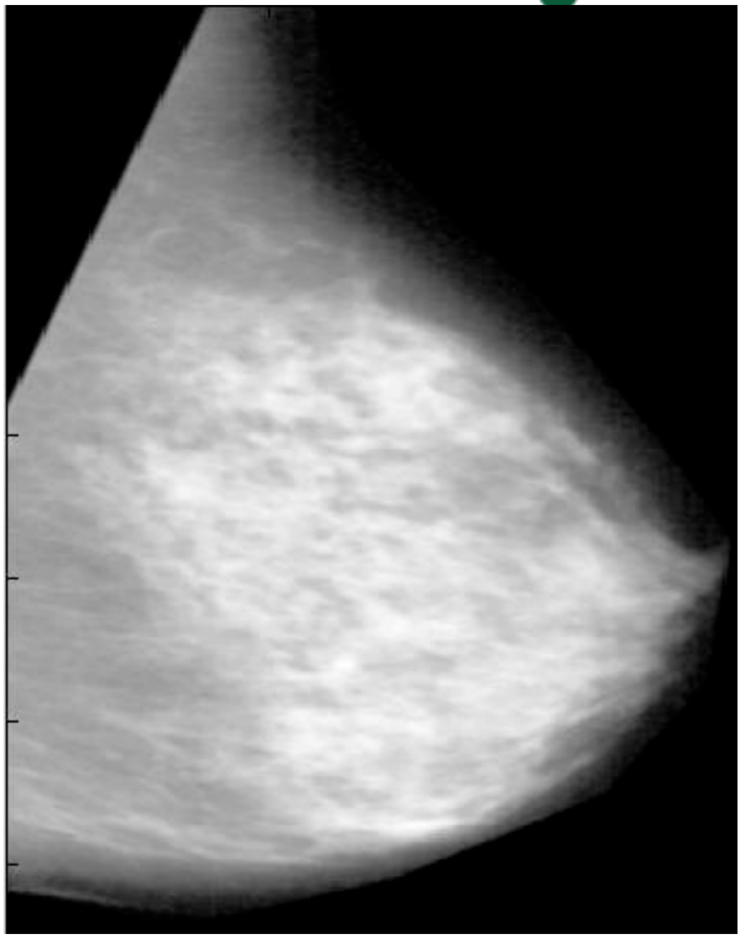
Learn separately a linear model for each aorta wall. (We will return with the details of the model later)



Example: Automated mammographic analysis

Disease scoring (regression):

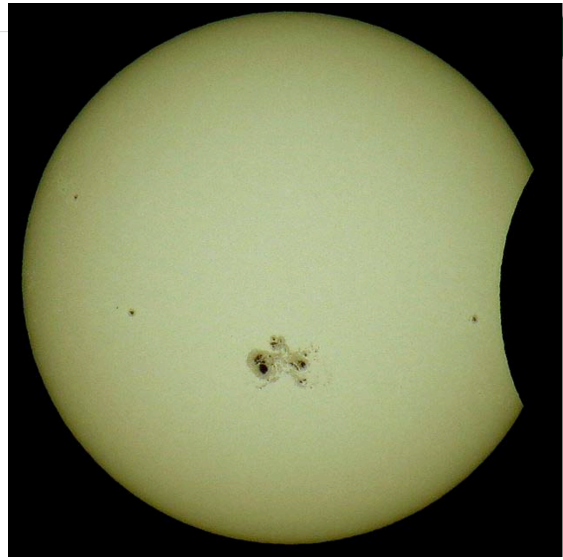
- From analysis of the mammogram can we predict risk (disease scoring) of getting breast cancer?
- Observations x_n: A vector of image measurements quantifying fatty tissue distribution.
- Target values t_n: For each mammogram we have a BI-RADS score provided by radiologist.
- Goal: Learn a regression model of the BI-RADS score providing a continuous score.

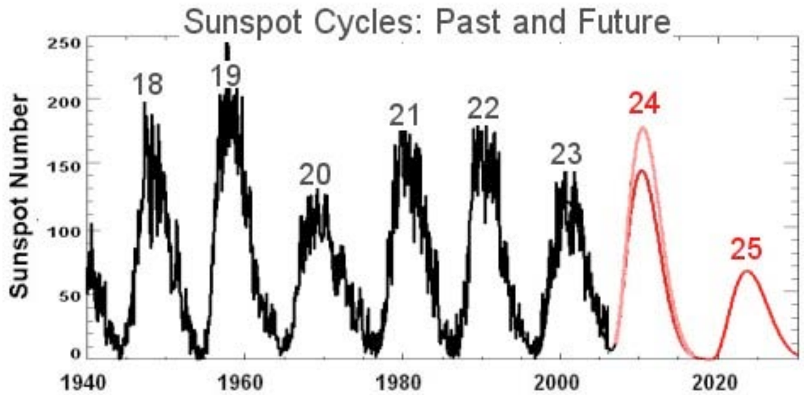


(Image from J. Raundahl Ph.D. Thesis, DIKU, 2007)

Example: Sunspots (Assignment 2)

- **Input variable:**
  - Number of sunspot in previous years
- **Target variable:**
  - Number of sunspots in following years
- **Your task:**
  - Learn a linear regression model
    $$t = \mathbf{y}(\mathbf{x})$$
    for predicting sunspot numbers
  - How to do this?
  - We learn today and Tuesday

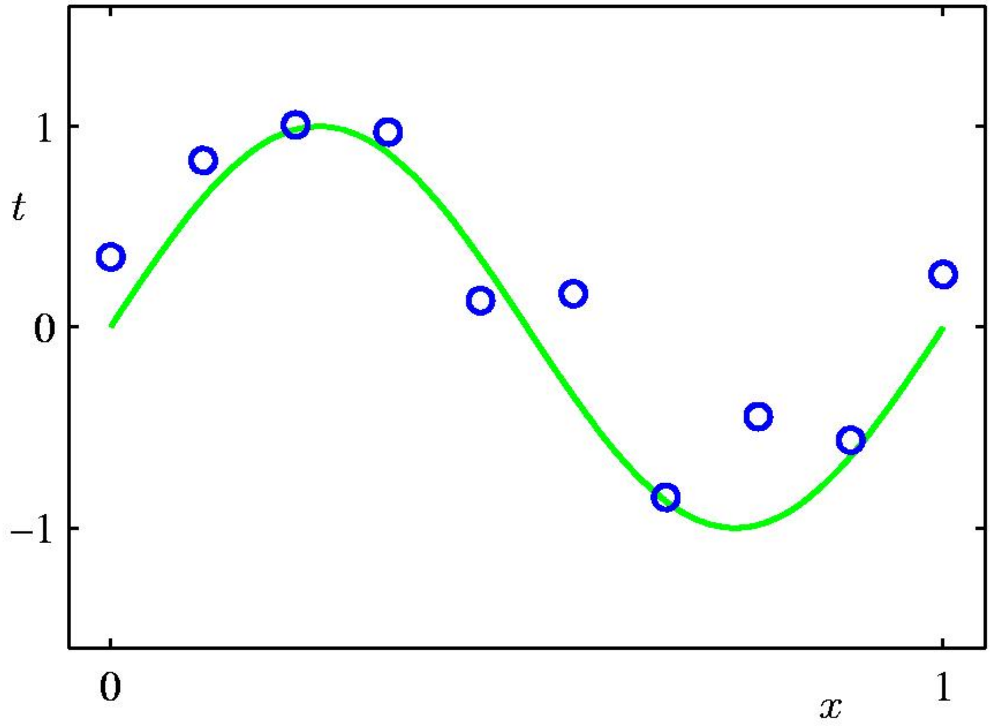




http://en.wikipedia.org/wiki/Sunspot

Lets look at the problem of polynomial curve fitting

Polynomial Curve Fitting (Regression)



Synthetic data set example:

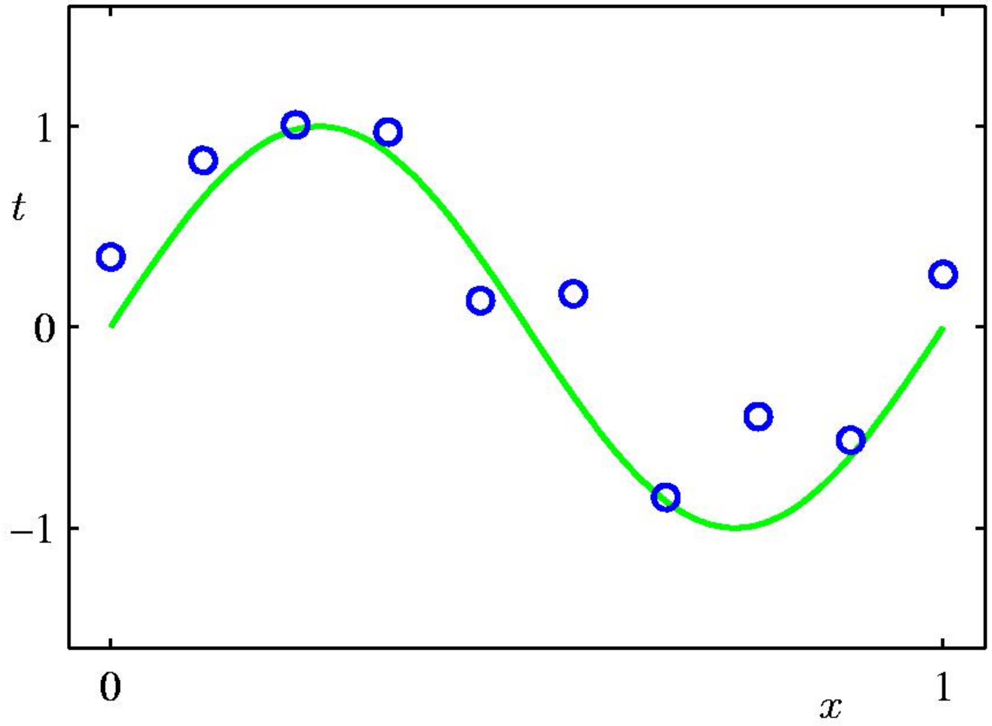
$$t(x) = Sin(2\pi x) + \chi \ , \chi \sim \mathcal{N}(\chi \mid 0, 0.3^2)$$

Training set: $X = (x_1, \ldots, x_N)^T \quad , \quad N = 10$

$$T = (t_1, \ldots, t_N)^T$$

Polynomial Curve Fitting (Regression)

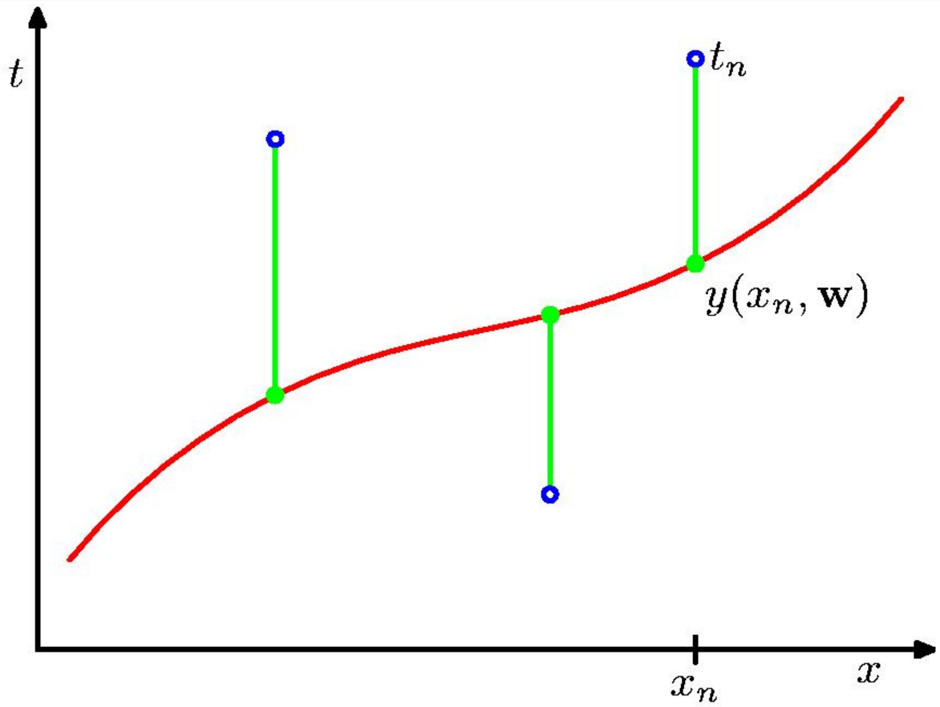
We don't know the green curve and would like to estimate a model of it



$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

This is a linear model in $\mathbf{w}=(w_0,\ldots,w_M)$, but non-linear in x.

A Solution:
Minimize the Sum-of-Squares Error Function

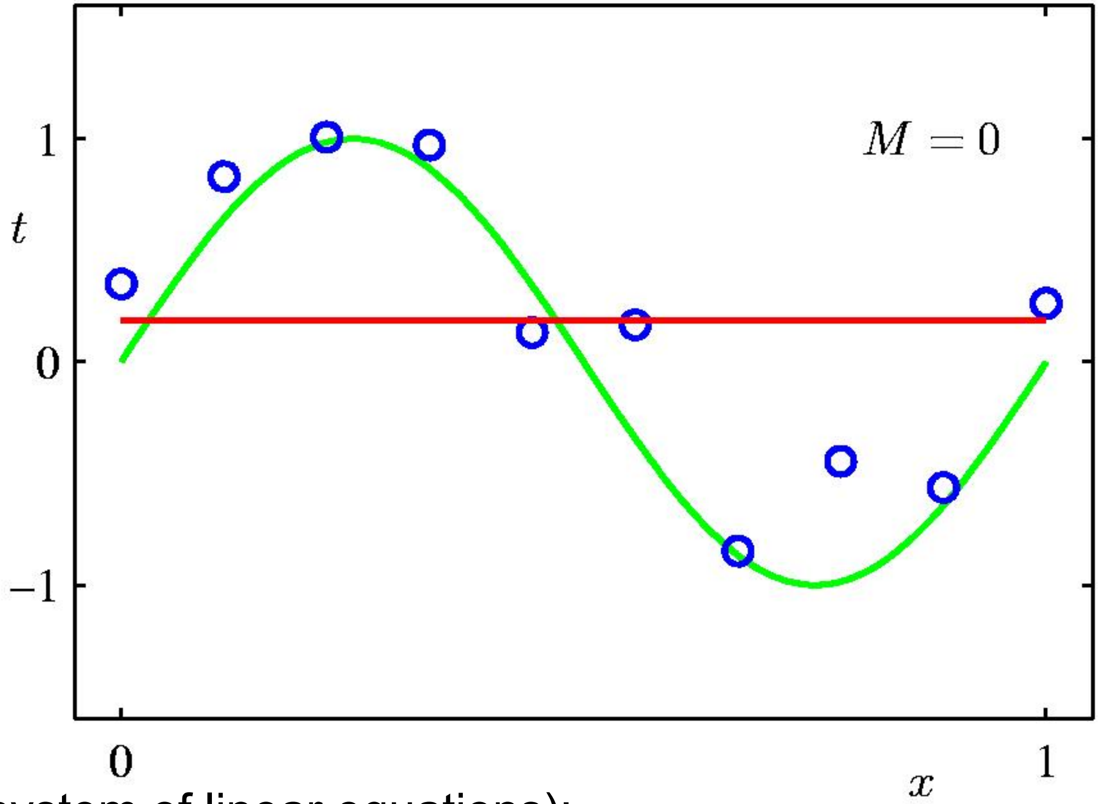


Choose **w** that minimizes the sum of squares error:

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

A unique solution exist since it is a quadratic problem.

0^{th} Order Polynomial



$$y(x,\mathbf{w}) = w_0$$

$$A_{00} = N \ , b_0 = \sum_{n=1}^{N} t_n$$

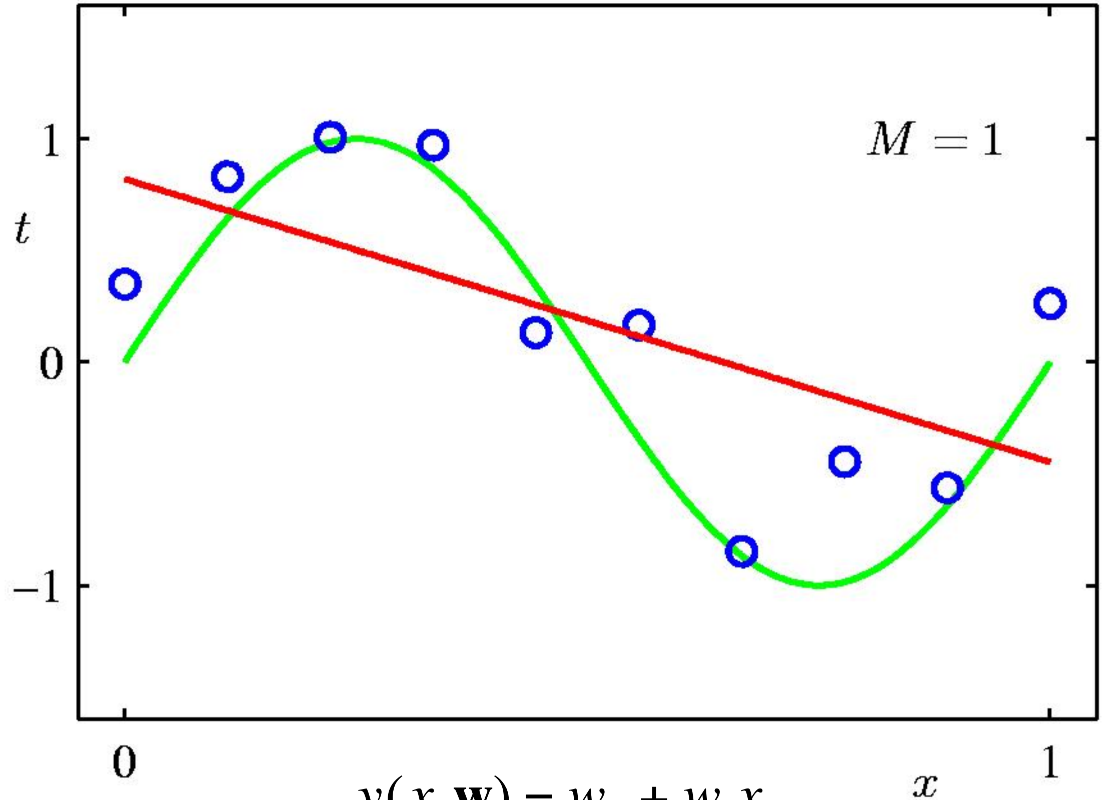
$$w_0 = \frac{1}{N}\sum_{n=1}^{N} t_n$$

Solution (system of linear equations):

$$\frac{\partial E}{\partial w_i} = 0 \ , \ i = 0,\ldots,M \quad \Rightarrow \quad \mathbf{A}\mathbf{w} = \mathbf{b} \quad \Rightarrow \quad \mathbf{w}^* = \mathbf{A}^{-1}\mathbf{b}$$

$$A_{ij} = \sum_{n=1}^{N} x_n^i x_n^j \quad , \quad b_i = \sum_{n=1}^{N} x_n^i t_n$$

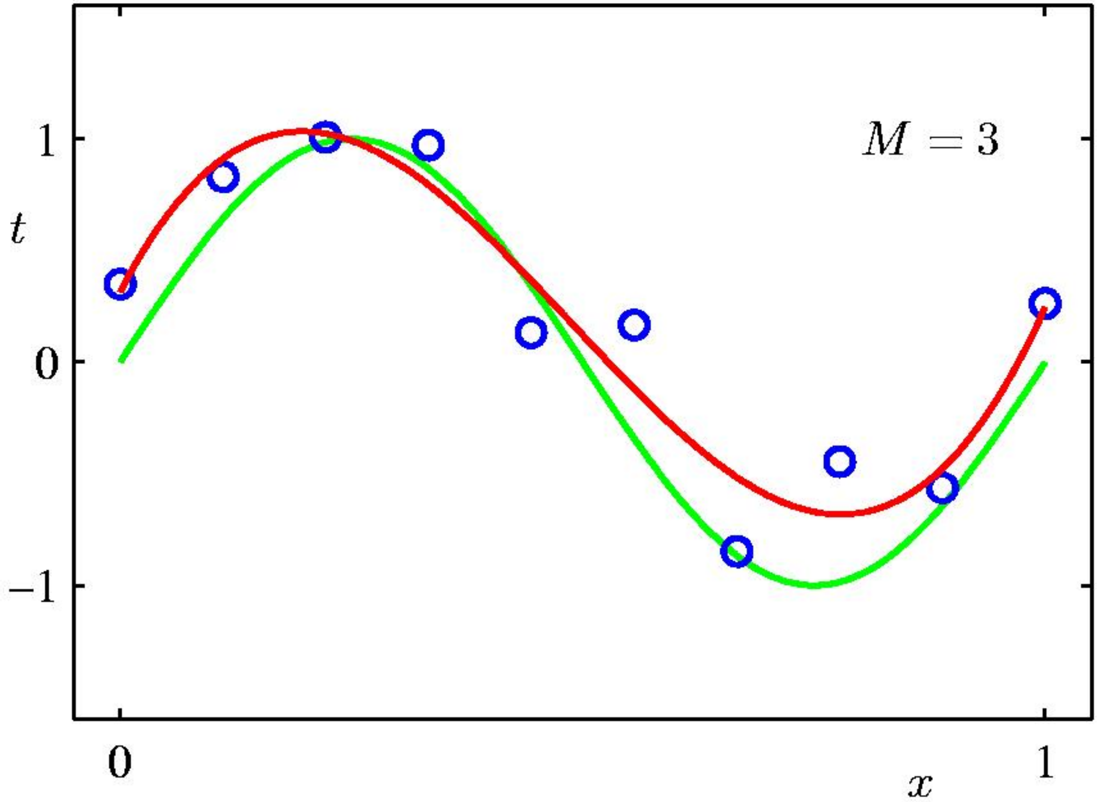
1^{st} Order Polynomial



$$y(x,\mathbf{w}) = w_0 + w_1 x$$

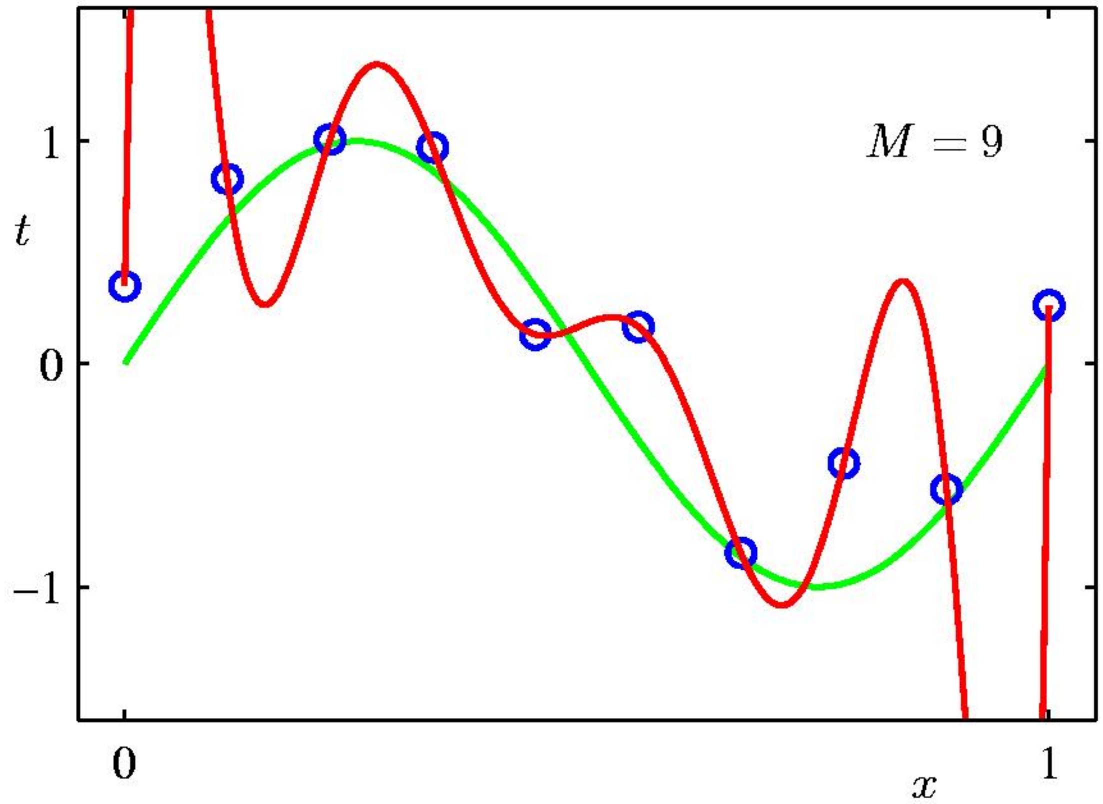
$$A_{00} = N \;,\; A_{01} = A_{10} = \sum_{n=1}^{N} x_n \;,\; A_{11} = \sum_{n=1}^{N} x_n^2 \;,\; b_0 = \sum_{n=1}^{N} t_n \;,\; b_1 = \sum_{n=1}^{N} x_n t_n$$

3^{rd} Order Polynomial
Model selection: Which model to choose?



$$y(x,\mathbf{w}) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

9^{th} Order Polynomial
Model selection: Which model to choose?



$$y(x,\mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_9 x^9$$

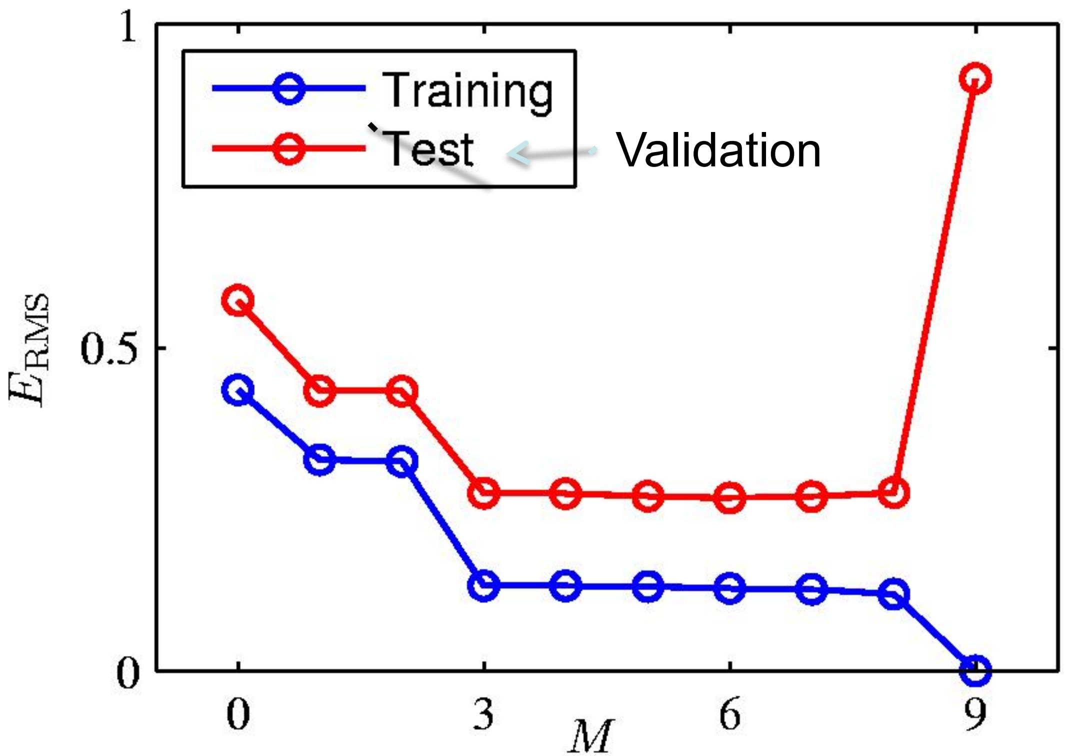
$E(\mathbf{w}^*) = 0$ Perfect fit? No, an example of *overfitting*

Central Goal of Learning: Generalization

- **Goal:** Maximize the generalization ability of the learnt model $y(x)$ so as to provide good performance on predicting the outcome of $t(x)$ for previously unseen data.

- In order to verify the generalization ability, we divide the data set, e.g. (x_n,t_n), into:
  - *Training set* for learning the model
  - *Validation set* for selecting the model (not always necessary)
  - *Test set* for evaluating the generalization ability of the learnt and selected model
- **Advice:** Avoid overfitting y(x) to the training set because it gives poor generalization ability!

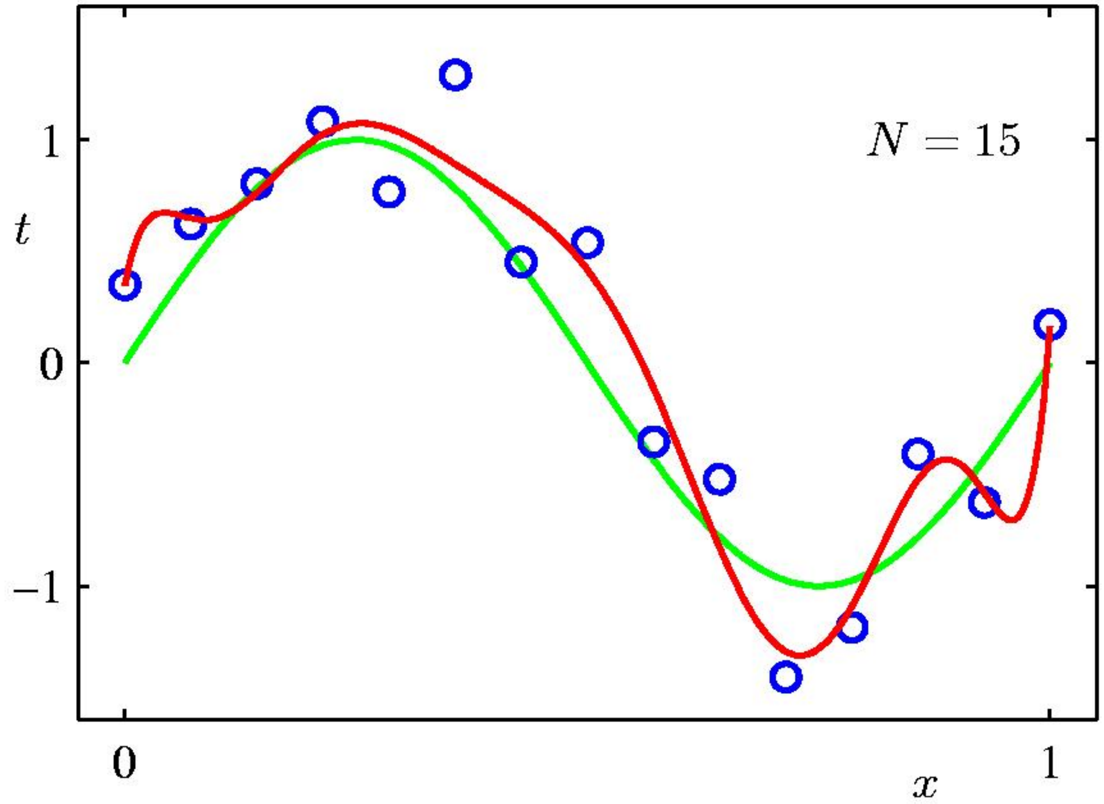
Over-fitting (lack of generalization) and model selection



Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$

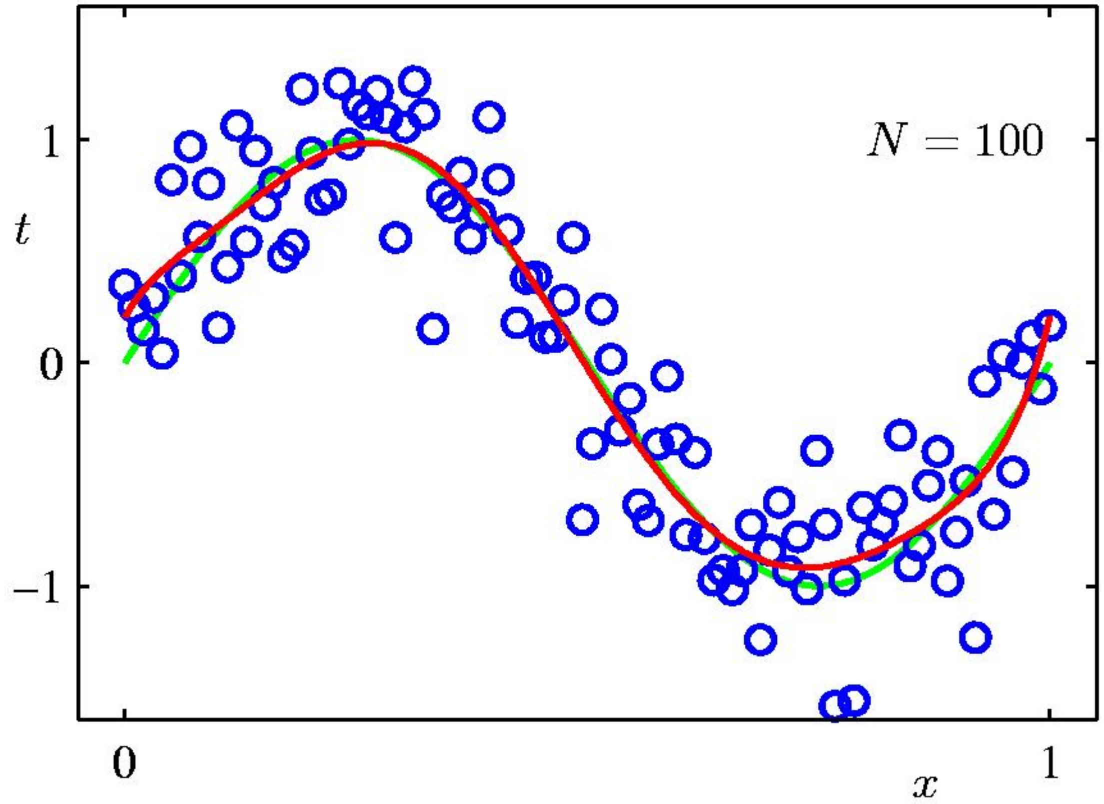
Data Set Size: $N = 15$

9th Order Polynomial



Data Set Size: $N = 100$

9th Order Polynomial



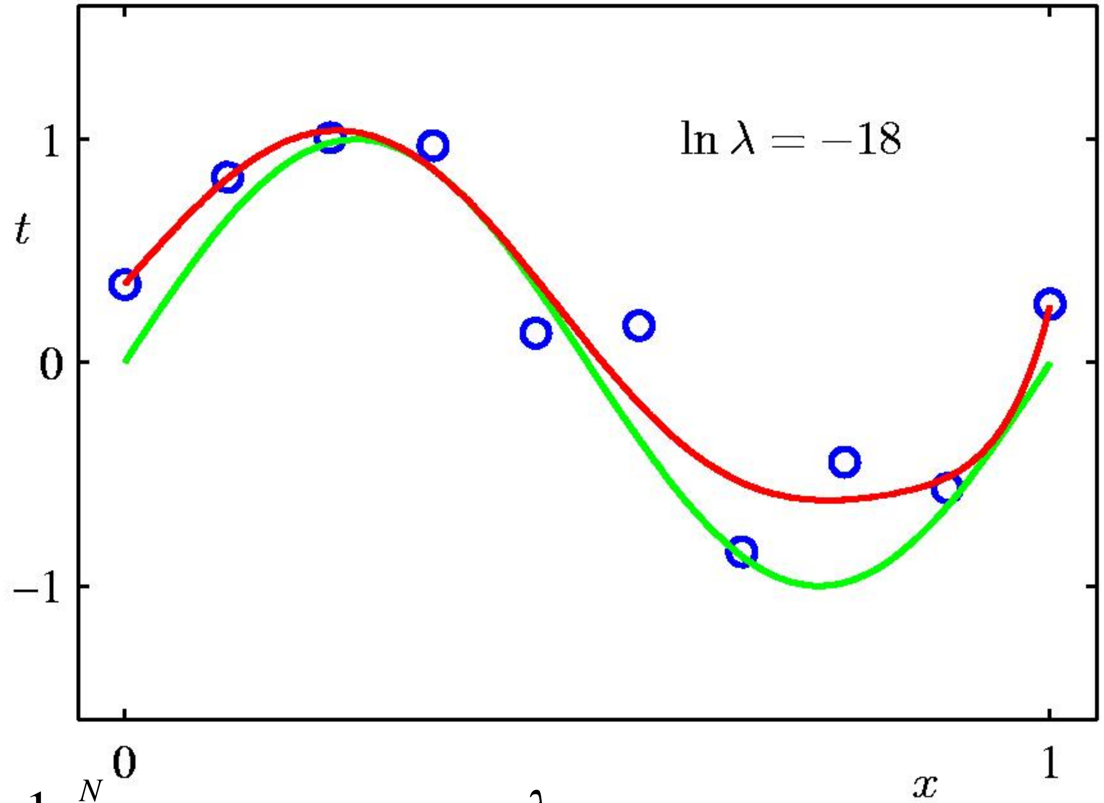
The more data *N*, the more complex models *M* we may choose, if *M*<<*N*

Polynomial Coefficients (Back to *N*=10)

	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

An Improved Approach: Regularization

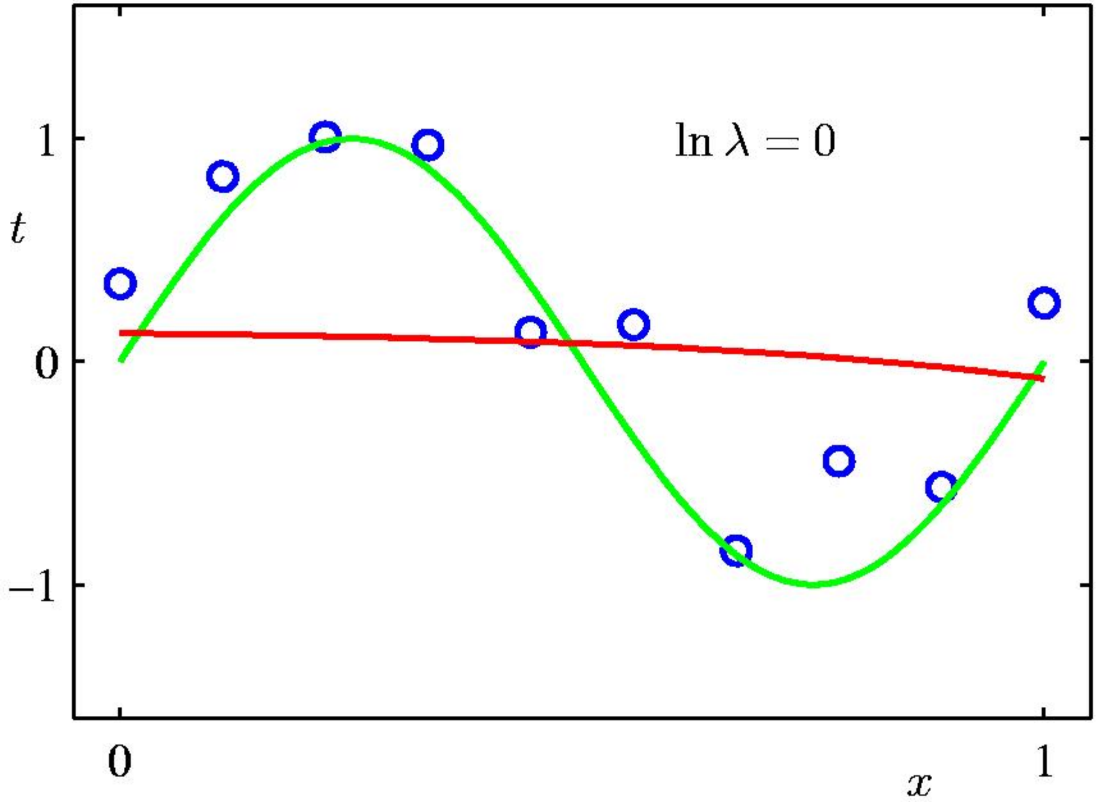
9th Order Polynomial



$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left[y(x_n,\mathbf{w}) - t_n\right]^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2 \quad , \quad \|\mathbf{w}\|^2 = w_0^2 + \cdots + w_M^2$$

Now we can to some extend do M>N

Regularization: $\ln \lambda = 0$



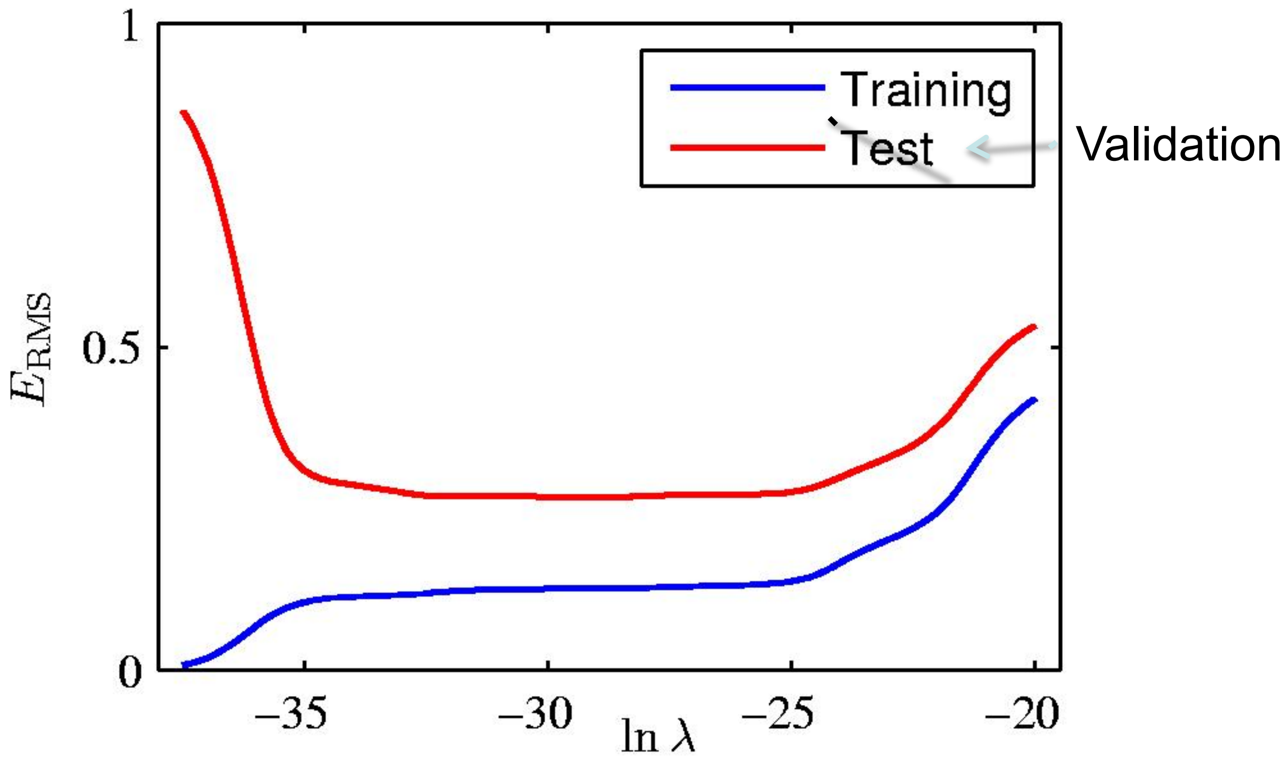
Too much regularization!

Polynomial Coefficients
***N*=10**

	$\ln\lambda = -\infty$	$\ln\lambda = -18$	$\ln\lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

Regularization: E_{RMS} vs. $\ln \lambda$

How to choose the regularization weight λ?



Lets look at the polynomial curve fitting again but from a probabilistic point of view

Bayes' theorem (Recall from lecture 2)

Assume we want to learn parameters *w* from a data set *D*.

- Bayes' theorem allows us to update our belief of uncertainty in the model parameters given observations.

$$p(w \mid D) = \frac{p(D \mid w)\, p(w)}{p(D)}$$

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

  - Our knowledge prior to the experiment is coded in the prior.
  - After the experiment our uncertainty about w has been updated and is given by the posterior distribution.

Approaches to parameter estimation (Recall from lecture 2)

- Maximum likelihood (ML) estimation
  Choose *w* that maximizes
  $$p(D \mid w) \qquad \text{(likelihood function)}$$

- Maximum a posteriori (MAP) estimation
  Choose *w* that maximizes
  $$p(w \mid D) \qquad \text{(posterior probability)}$$

A probabilistic interpretation of least squares

- Least squares is a maximum likelihood solution assuming a Gaussian noise model: $t(x) = y(x, \mathbf{w}) + \varepsilon \quad , \quad \varepsilon \sim \mathcal{N}\left(\varepsilon \mid 0, \beta^{-1}\right)$

Predictive distribution:

$$p(t \mid x, \mathbf{w}, \beta) = \mathcal{N}\left(t \mid y(x, \mathbf{w}), \beta^{-1}\right)$$

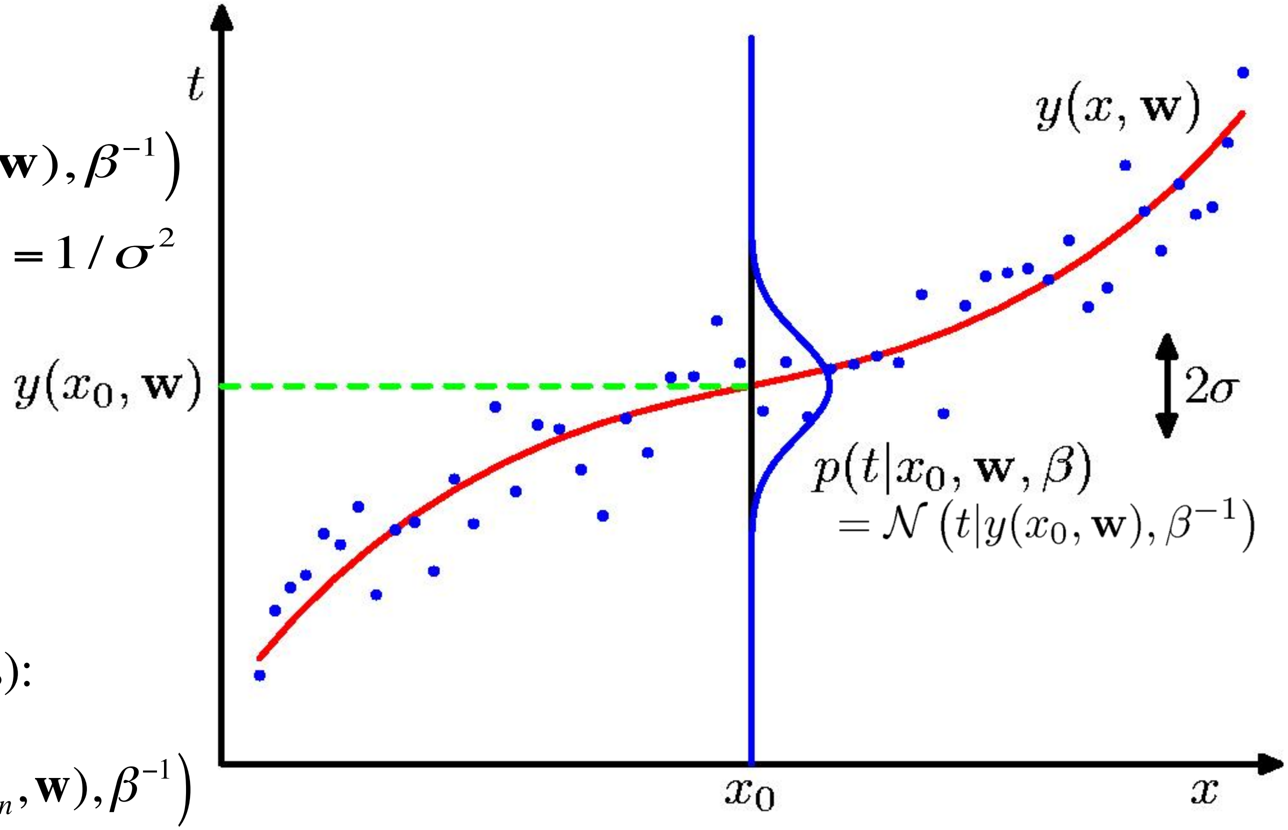
Precision and variance: $\beta = 1/\sigma^2$

$$X = (x_1, \ldots, x_N)^T$$

$$T = (t_1, \ldots, t_N)^T$$

Likelihood (i.i.d. observations):

$$p(T \mid X, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}\left(t_n \mid y(x_n, \mathbf{w}), \beta^{-1}\right)$$



A probabilistic interpretation of least squares

- Consider the log-likelihood:

$$\log p(T \mid X, \mathbf{w}, \beta) = -\frac{\beta}{2}\sum_{n=1}^{N}\left(y(x_n, \mathbf{w}) - t_n\right)^2 + \frac{N}{2}\log\beta - \frac{N}{2}\log(2\pi)$$

- From maximization to minimization: Flip the sign of log-likelihood, set β=1, and ignore the constant term

$$-\log p(T \mid X, \mathbf{w}, \beta = 1) \approx \frac{1}{2}\sum_{n=1}^{N}\left(y(x_n, \mathbf{w}) - t_n\right)^2 = E(\mathbf{w}) \quad \text{(Sum-of-squares error)}$$

- Maximum likelihood estimation for linear models for both parameters and noise variance (point estimate):

$$\mathbf{w}_{\mathrm{ML}} = \mathbf{A}^{-1}\mathbf{b} \quad , \qquad \sigma^2_{\mathrm{ML}} = \frac{1}{\beta_{\mathrm{ML}}} = \frac{1}{N}\sum_{n=1}^{N}\left(y(x_n, \mathbf{w}_{\mathrm{ML}}) - t_n\right)^2$$

A probabilistic interpretation of least squares

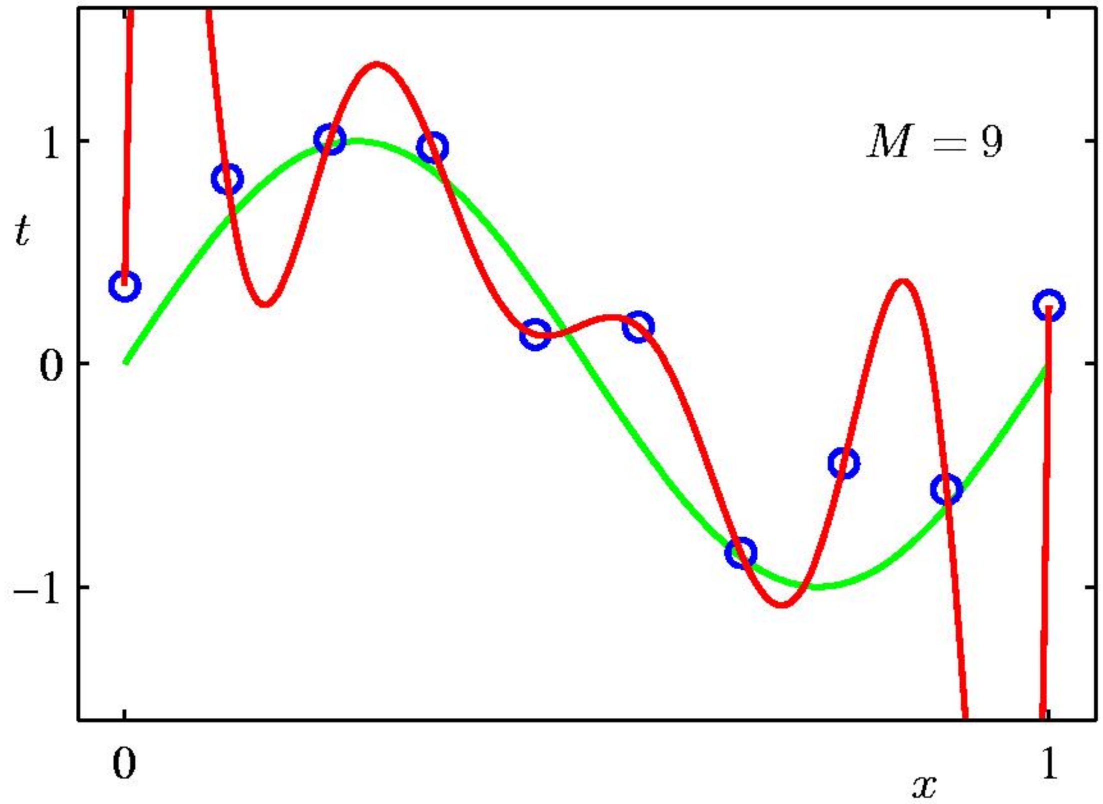
- The least squares method is a maximum likelihood (ML) solution.

- With the ML interpretation we get both a point estimate but also the predictive distribution for new *t*(*x*):

$$p(t \mid x, \mathbf{w}_{\mathrm{ML}}, \beta_{\mathrm{ML}}) = \mathcal{N}\left(t \mid y(x, \mathbf{w}_{\mathrm{ML}}), \beta_{\mathrm{ML}}^{-1}\right)$$

- Overfitting is generally a potential problem of maximum likelihood solutions.

Overfitting: 9th Order Polynomial

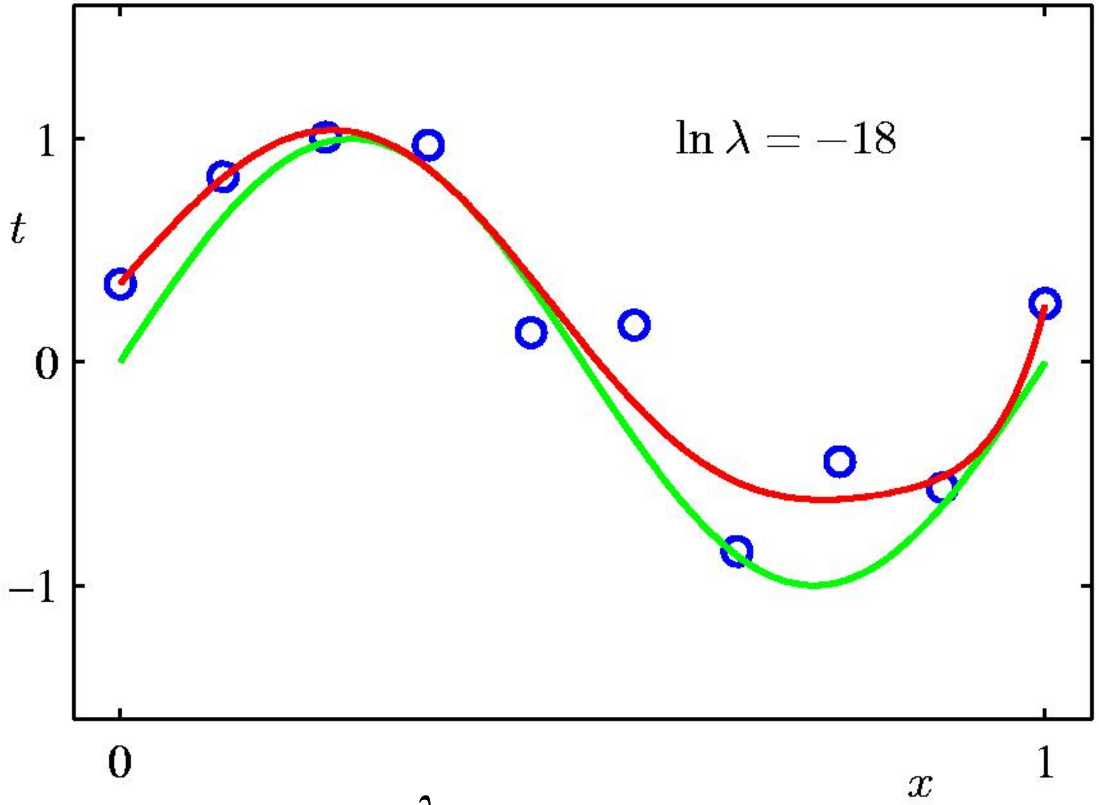


Perfect fit?

$$E(\mathbf{w}^*) = 0$$

The more data *N*, the more complex models *M* we may choose, if *M*<<*N*

Regularization: $\ln \lambda = -18 \quad , \quad M = 9$



$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left[y(x_n,\mathbf{w}) - t_n\right]^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2 \quad , \quad \|\mathbf{w}\|^2 = w_0^2 + \cdots + w_M^2$$

Now we can to some extend do *M*>*N*, but how to choose λ?

The Bayesian interpretation of regularization

- Lets introduce a (conjugated) prior on parameters – a multivariate isotropic Gaussian (remember that $\|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w}$):

$$p(\mathbf{w} \mid \alpha) = \mathcal{N}\left(\mathbf{w} \mid \mathbf{0}, \alpha^{-1}\mathbf{I}\right) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} \exp\left[-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}\right]$$

- The precision α is called a *hyper-parameter*.
- Bayes theorem using the Gaussian likelihood $p(T \mid X, \mathbf{w}, \beta)$:

$$p(\mathbf{w} \mid X, T, \alpha, \beta) \propto p(T \mid X, \mathbf{w}, \beta)\, p(\mathbf{w} \mid \alpha)$$

- MAP solution equivalent to minimization of $-\log p(\mathbf{w} \mid X, T, \alpha, \beta)$
- Leads to regularized least squares with $\lambda = \alpha/\beta$

$$-\log p(\mathbf{w} \mid X, T, \alpha, \beta) \approx \frac{\beta}{2}\sum_{n=1}^{N}\left(y(x_n, \mathbf{w}) - t_n\right)^2 + \frac{\alpha}{2}\mathbf{w}^T\mathbf{w}$$

Lets try to generalize the concept of linear models for regression

Linear basis function models

- Training data set: $X = \{x_1, \ldots, x_N\}$

$$T = \{t_1, \ldots, t_N\}$$

- The (M-1)'th order polynomial model is linear in the *M* model parameters:

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_{M-1} x^{M-1} = w_0 + \sum_{j=1}^{M-1} w_j x^j$$

- Generalize this model using (non-linear) basis functions:

$$y(x, \mathbf{w}) = w_0 + w_1 \phi_1(x) + w_2 \phi_2(x) + \cdots + w_{M-1} \phi_{M-1}(x) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(x)$$

- In vector notation using $\phi_0(x) = 1$:

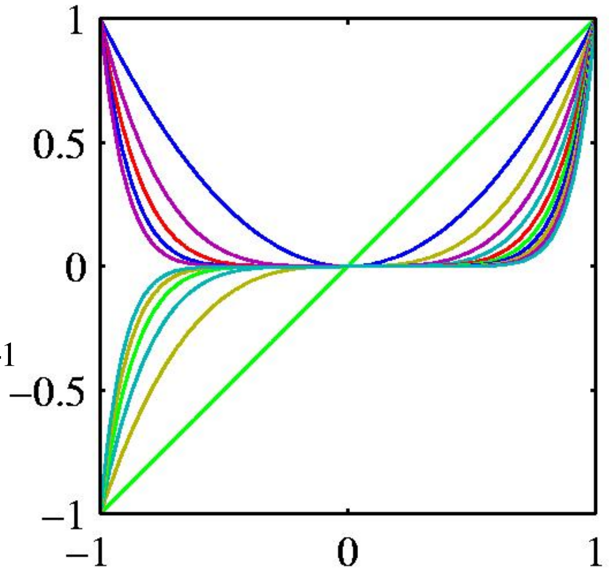
$$y(x, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(x) = \mathbf{w}^T \overline{\phi}(x)$$

$$\mathbf{w} = \left(w_0, \ldots, w_{M-1}\right)^T \;,\; \overline{\phi}(x) = \left(\phi_0(x), \ldots, \phi_{M-1}(x)\right)^T$$

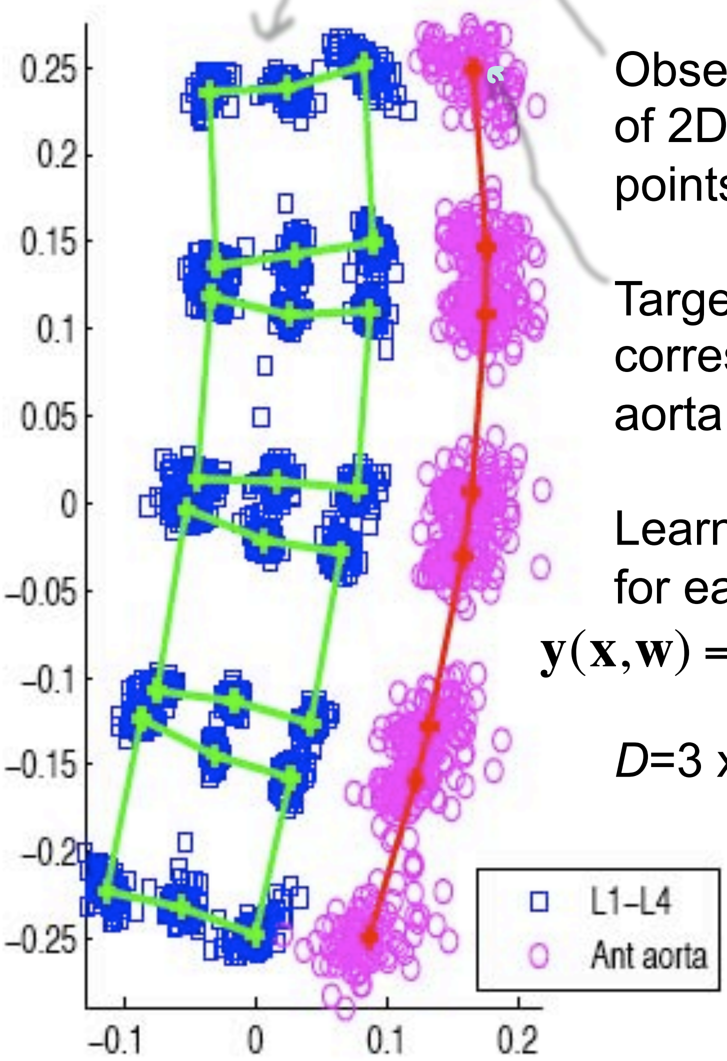
Examples of basis functions

- Simple *D*-dim. linear model: Assume $\mathbf{x} = (x_1, \ldots, x_D)^T$
  Basis functions:
  $$\phi_j(\mathbf{x}) = x_j \quad , \quad \overline{\phi}(\mathbf{x}) = (1, x_1, \ldots, x_D)^T$$
  Regression model:
  $$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \overline{\phi}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_D x_D$$

- Polynomial model (monomial basis):
  Basis functions:
  $$\phi_j(x) = x^j \quad , \quad \overline{\phi}(x) = (1, x, x^2, \ldots, x^{M-1})^T$$
  Regression model:
  $$y(x, \mathbf{w}) = \mathbf{w}^T \overline{\phi}(\mathrm{x}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_{M-1} x^{M-1}$$



Example: Training Set and Posterior Mean Shape



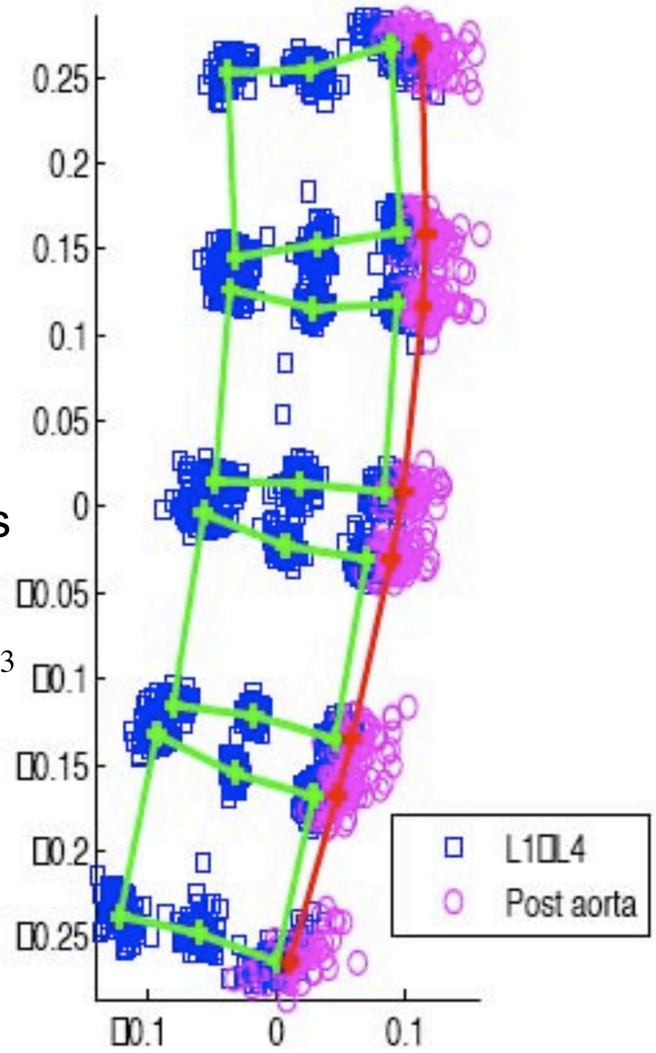
Observation vector $\boldsymbol{x}_n$ consist of 2D position of 3 vertebrae points.

Target variable $\boldsymbol{t}_n$ is the corresponding 2D location of aorta point.

Learn separately linear models for each aorta wall.

$$\mathbf{y}(\mathbf{x},\mathbf{w}) = w_0 + w_1\mathbf{x}_1 + w_2\mathbf{x}_2 + w_3\mathbf{x}_3$$

D=3 x 2 and K=2 (dim. of $\mathbf{y}$)

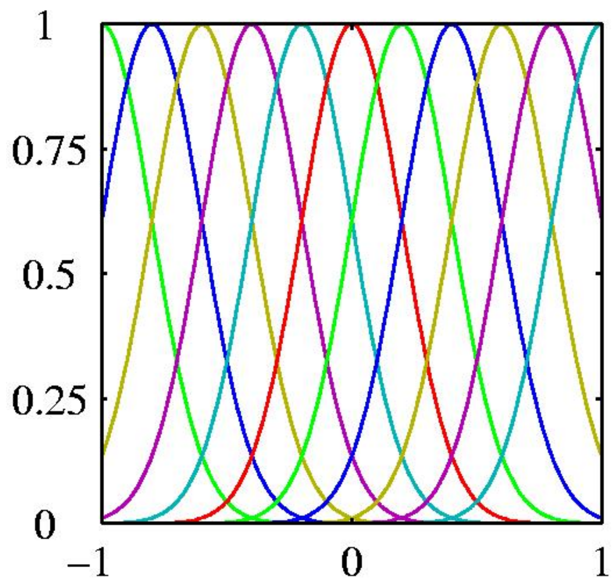


More examples of basis functions

- Gaussian basis function:
  Basis functions:
  $$\phi_j(x) = \exp\left[-\frac{(x-x_j)^2}{2s^2}\right]$$
  Regression model:

$$y(x,\mathbf{w}) = w_0 + w_1 \exp\left[-\frac{(x-x_1)^2}{2s^2}\right] + \cdots + w_{M-1} \exp\left[-\frac{(x-x_{M-1})^2}{2s^2}\right]$$

x_j position of basis function and s scale



- Other basis functions:
  - Sigmoid
  - Fourier
  - Wavelets
  - Splines (piecewise polynomial), …

Basis functions: Global versus local effect

- Polynomials fits data globally: Change a parameter and it has effect globally by changing the whole curve.
- The Gaussian basis fits data locally: Changing a parameter changes the basis weight locally and only changes the curve locally. The Gaussians are localized, but has infinite support (will cause very small changes far away).
- Splines (piecewise polynomials) fits data locally: Changing a parameter only affects the curve locally (in the region of the local polynomial).
- Wavelets fits data locally: Wavelets are localized in space/time and frequency.

Curse of Dimensionality (Again)

D-dimensional polynomial curve fitting, *M* = 3

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D} \sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D} \sum_{j=1}^{D} \sum_{k=1}^{D} w_{ijk} x_i x_j x_k$$

In general: Number of free model parameters grows polynomially D^M with dimensionality *D*, hence the data set size N should grow polynomially to keep same precision on parameter estimates.

How do we in general learn these linear models for regression?

Topic of next lecture

Summary

- Formulation of the regression problem
- Generalization: Training, validation and test data sets
- Over-fitting: Model complexity vs. amount of training data.
- Least squares and maximum likelihood solutions are equivalent under the Gaussian noise model.
- Regularization can be interpreted as using priors on the model parameters. The MAP solution using Gaussian likelihood and isotropic Gaussian prior is equivalent to the regularized least squares solution.
- Using priors and the MAP solution allows us to handle M>N. The problem of over-fitting is reduced.
- Linear models for regression from basis functions

Literature

- Curve fitting – the probabilistic interpretation: Sec. 1.1, 1.2.5 – 1.2.6
- Linear models for regression: Sec. 3.1 (pages 137 – 140)
- Curse of dimensionality: Sec. 1.4