Regression 1

StatML
18.02.2014
Aasa Feragen
(aasa@diku.dk)

Pep talk

I.3 The Gaussian distribution and its conditional distributions

In sections 2.3.1 and 2.3.2 of [2], we considered the conditional and marginal distributions for a multivariate Gaussian. More generally, we can consider a partitioning of the components of $\mathbf{x}$ into three groups $\mathbf{x}_a, \mathbf{x}_b$, and $\mathbf{x}_c$, with a corresponding partitioning of the mean μ and of the covariance $\boldsymbol{\Sigma}$ in the form

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \\ \boldsymbol{\mu}_c \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} & \boldsymbol{\Sigma}_{ac} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} & \boldsymbol{\Sigma}_{bc} \\ \boldsymbol{\Sigma}_{ca} & \boldsymbol{\Sigma}_{cb} & \boldsymbol{\Sigma}_{cc} \end{pmatrix}$$

By use of the results of CB Sec. 2.3, find an expression for the conditional distribution $p(\mathbf{x}_a|\mathbf{x}_b)$ in which $\mathbf{x}_c$ has been marginalized out

Deliverables: Resulting expression and proof

- Why is this important?
  - Linear algebra is a fundamental ML tool
  - Exponentials and Gaussians are fundamental ML tools
  - You will see these techniques again!

Pep talk

I.3 The Gaussian distribution and its conditional distributions

In sections 2.3.1 and 2.3.2 of [2], we considered the conditional and marginal distributions for a multivariate Gaussian. More generally, we can consider a partitioning of the components of $\mathbf{x}$ into three groups $\mathbf{x}_a, \mathbf{x}_b$, and $\mathbf{x}_c$, with a corresponding partitioning of the mean μ and of the covariance $\mathbf{\Sigma}$ in the form

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \\ \boldsymbol{\mu}_c \end{pmatrix}, \quad \mathbf{\Sigma} = \begin{pmatrix} \mathbf{\Sigma}_{aa} & \mathbf{\Sigma}_{ab} & \mathbf{\Sigma}_{ac} \\ \mathbf{\Sigma}_{ba} & \mathbf{\Sigma}_{bb} & \mathbf{\Sigma}_{bc} \\ \mathbf{\Sigma}_{ca} & \mathbf{\Sigma}_{cb} & \mathbf{\Sigma}_{cc} \end{pmatrix}$$

By use of the results of CB Sec. 2.3, find an expression for the conditional distribution $p(\mathbf{x}_a|\mathbf{x}_b)$ in which $\mathbf{x}_c$ has been marginalized out

Deliverables: Resulting expression and proof

- Why is this not a disaster?
  - Friendly grading (you have to try)
  - Show that you get the point and master basic techniques
  - You can resubmit
  - Don't neglect the other exercises! If you were ok with the rest of Assignment 1, you don't need to worry.

- Why is this important?
  - Linear algebra is a fundamental ML tool
  - Exponentials and Gaussians are fundamental ML tools
  - You will see these techniques again!

Don't Panic!

I.3 The Gaussian distribution and its conditional distributions

In sections 2.3.1 and 2.3.2 of [2], we considered the conditional and marginal distributions for a multivariate Gaussian. More generally, we can consider a partitioning of the components of $\mathbf{x}$ into three groups $\mathbf{x}_a, \mathbf{x}_b$, and $\mathbf{x}_c$, with a corresponding partitioning of the mean μ and of the covariance $\boldsymbol{\Sigma}$ in the form

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \\ \boldsymbol{\mu}_c \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} & \boldsymbol{\Sigma}_{ac} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} & \boldsymbol{\Sigma}_{bc} \\ \boldsymbol{\Sigma}_{ca} & \boldsymbol{\Sigma}_{cb} & \boldsymbol{\Sigma}_{cc} \end{pmatrix}$$

By use of the results of CB Sec. 2.3, find an expression for the conditional distribution $p(\mathbf{x}_a|\mathbf{x}_b)$ in which $\mathbf{x}_c$ has been marginalized out

Deliverables: Resulting expression and proof

- This *is* hard – it's ok
- If you did the exercise – great!
- If you did part of the exercise – great!
- You will use the techniques you learned over the next weeks

- Why is this important?
  - Linear algebra is a fundamental ML tool
  - Exponentials and Gaussians are fundamental ML tools
  - You will see these techniques again!

What happens now?

- The TAs hope to grade your assignments by Thursday
- General and individual feedback at TA sessions
- Additional lecture by Christian Friday 13.30-14.15 in Aud 3 (HCØ)
- Math Q&A / help session Friday afternoon:

  14.15 - ca 16.00, A103, A104 and A105 at HCØ

Case: Automated mammographic analysis

- Image texture measurements are predictive of breast cancer
- Given 1000 images with 1000 cancer scores, can you build and evaluate a statistical model for predicting the cancer score from the image?

After today's lecture you should

- Be able to define different linear models for regression
- Be able to recognize a regression problem in practical situations
- Know common pitfalls of regression and common techniques to avoid them (regularization, experiment design)
- Understand the relationship between geometric (least squares) regression and maximum likelihood solutions to regression under a Gaussian noise model.
- Be able to deduct and implement maximum likelihood solutions to regression problems phrased trough linear models

Regression: A supervised learning problem

Input: N pairs $(\mathbf{x}_n, \mathbf{t}_n)$ of observed

input variables $\mathbf{x}_n \in \mathbb{R}^D$ and
target variables $\mathbf{t}_n \in \mathbb{R}^K$,

Assumption: There is a functional relationship

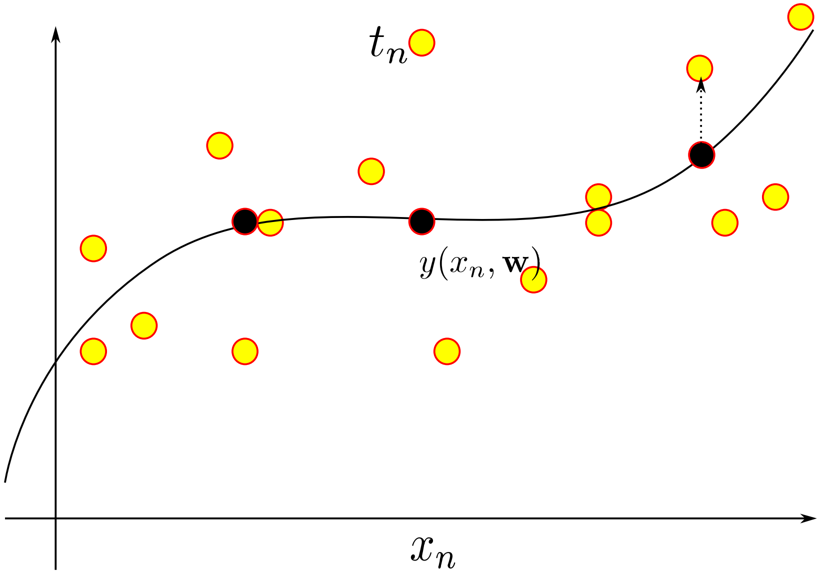
$$\mathbf{t} = \mathbf{y}(\mathbf{x})$$

where $\mathbf{y}$: $\mathbb{R}^D \to \mathbb{R}^K$

Goal: Learn the function $\mathbf{y}(\mathbf{x})$ from the N data points!

What is this good for?

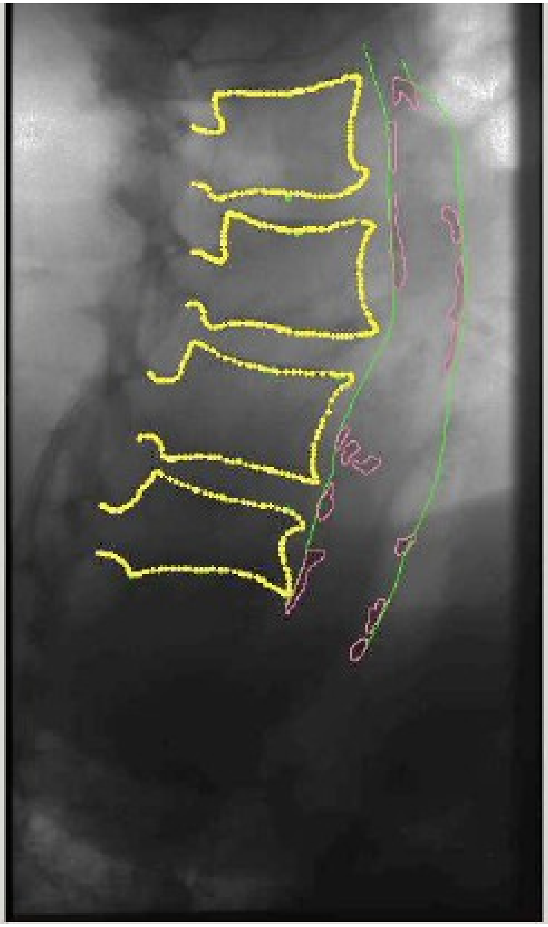
Given new observation of input variable $\mathbf{x}_0$,
predict corresponding output variable $\mathbf{y}(\mathbf{x}_0)$



Example: Predicting Aorta Wall Location in X-ray Images

Predict location of the spinal aorta walls conditioned on the vertebra location.

- Hard because soft tissue is not visible in x-rays, but calcifications are!
- Needed for quantification of aorta calcification – aorta area vs. calcification area.
- 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: Predicting Aorta Wall Location in X-ray Images

Input: N pairs $(\mathbf{x}_n, \mathbf{t}_n)$ of observed

input variables $\mathbf{x}_n \in \mathbb{R}^D$ and
target variables $\mathbf{t}_n \in \mathbb{R}^K$,

Assumption: There is a functional relationship

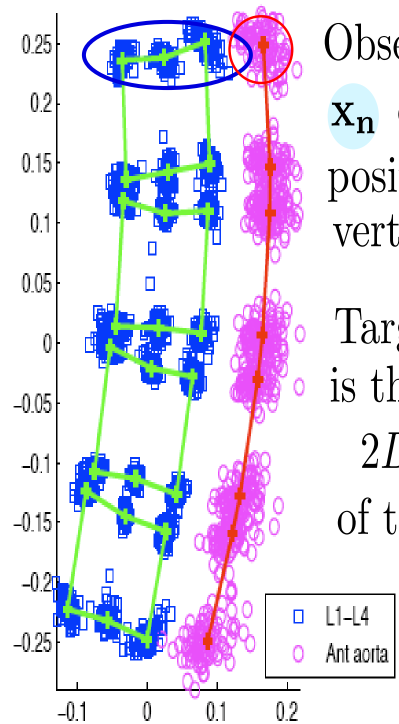
$$\mathbf{t} = \mathbf{y}(\mathbf{x})$$

where $\mathbf{y}\colon \mathbb{R}^D \to \mathbb{R}^K$

Goal: Learn the function $\mathbf{y}(\mathbf{x})$ from the N data points!

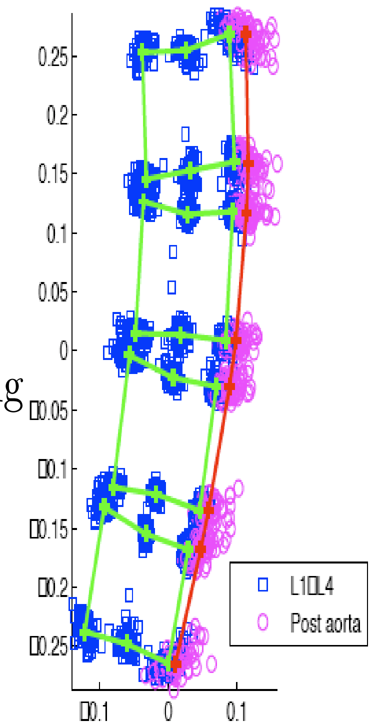
What is this good for?

Given new observation of input variable $\mathbf{x}_0$,
predict corresponding output variable $\mathbf{y}(\mathbf{x}_0)$



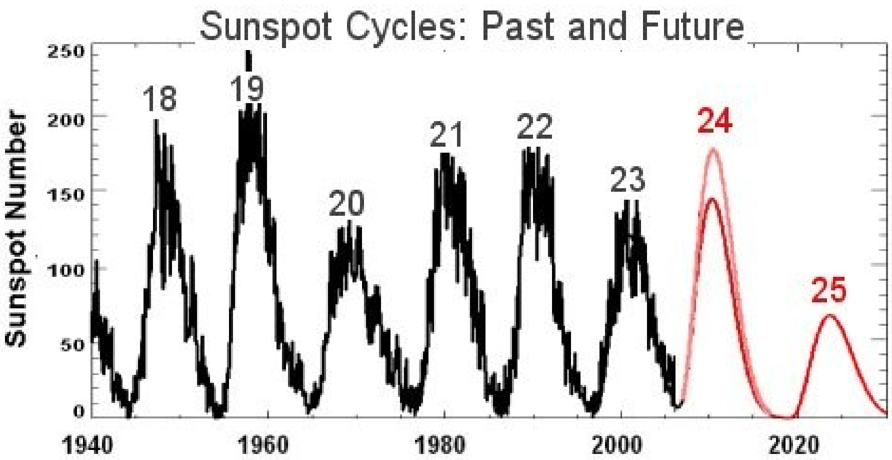
Observation vector $\mathbf{x_n}$ consists of $2D$ positions of 3 vertebrae points.

Target variable $\mathbf{t_n}$ is the corresponding $2D$ location of the aorta point.



Example: Sunspots (Assignment 2)

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



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

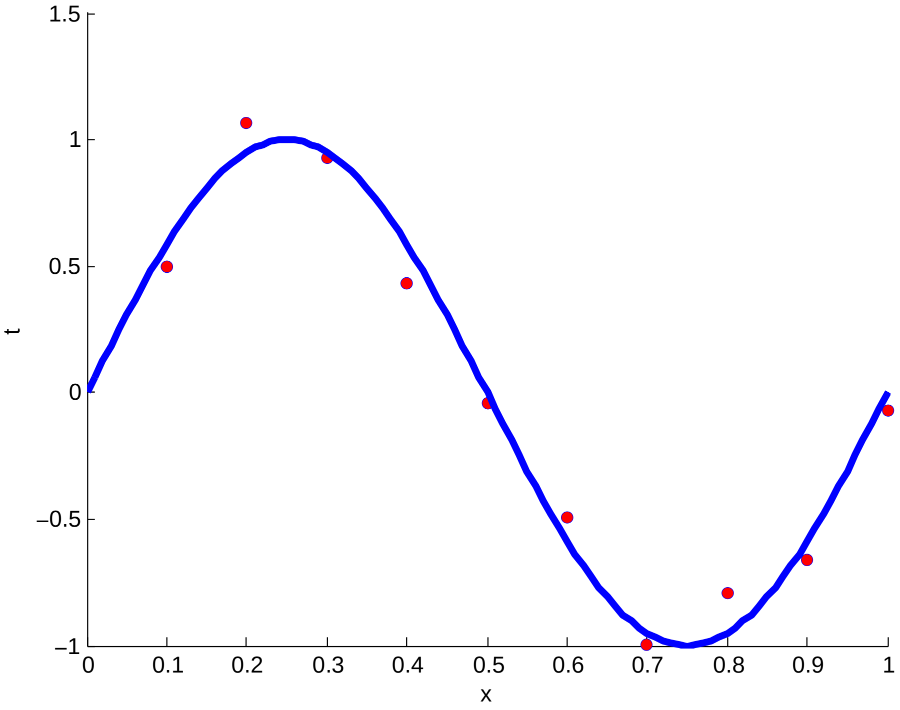
Today's running example: Polynomial curve fitting

- Synthetic data set

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

- Training set

$$\mathrm{X} = (\mathrm{x}_1, \ldots, x_N)$$
$$\mathrm{T} = (\mathrm{t}_1, \ldots, t_N)$$



Today's running example: Polynomial curve fitting

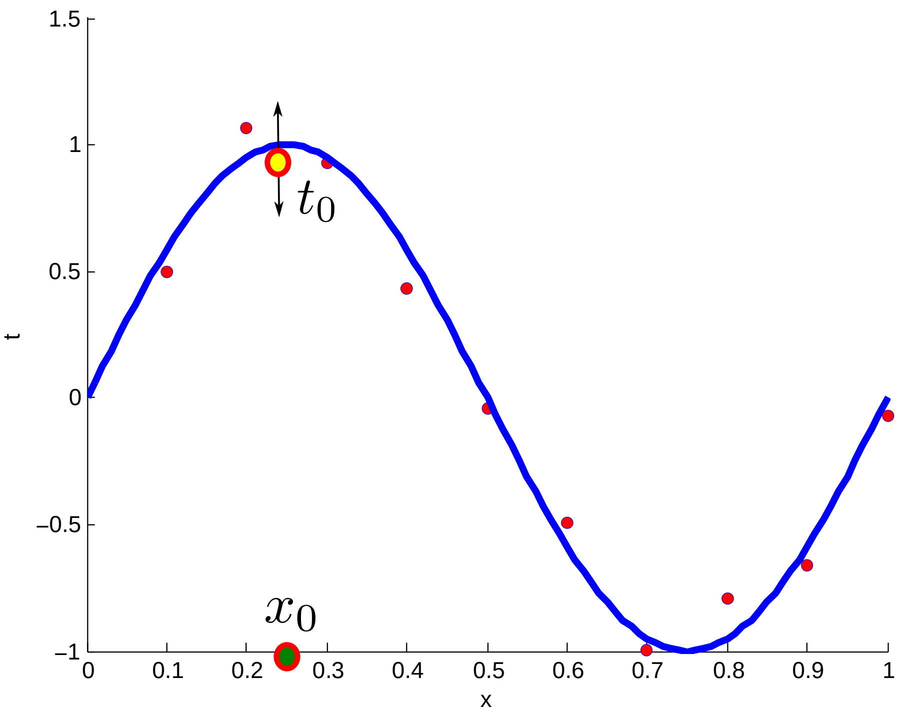
- Synthetic data set

$$\mathrm{t} = \sin(2\pi x) + \chi$$
$$\chi \sim \mathcal{N}(\mu = 0, \sigma^2 = 0.3^2)$$

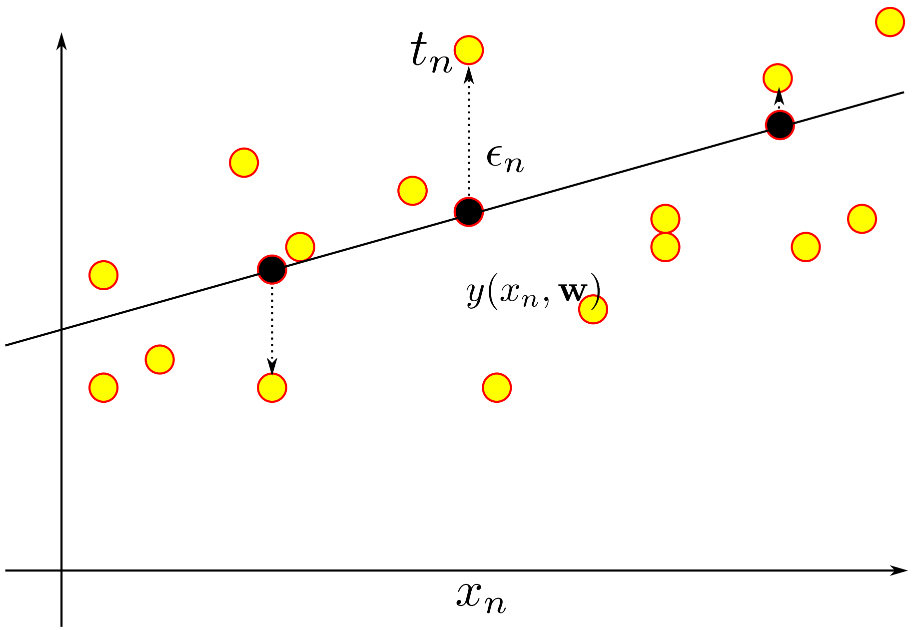
- Training set

$$\mathrm{X} = (\mathrm{x}_1, \ldots, x_N)$$
$$\mathrm{T} = (\mathrm{t}_1, \ldots, t_N)$$

- Can I learn a rule $\mathrm{t} = \mathrm{y}(\mathrm{x})$ for predicting t_0 for new x_0?



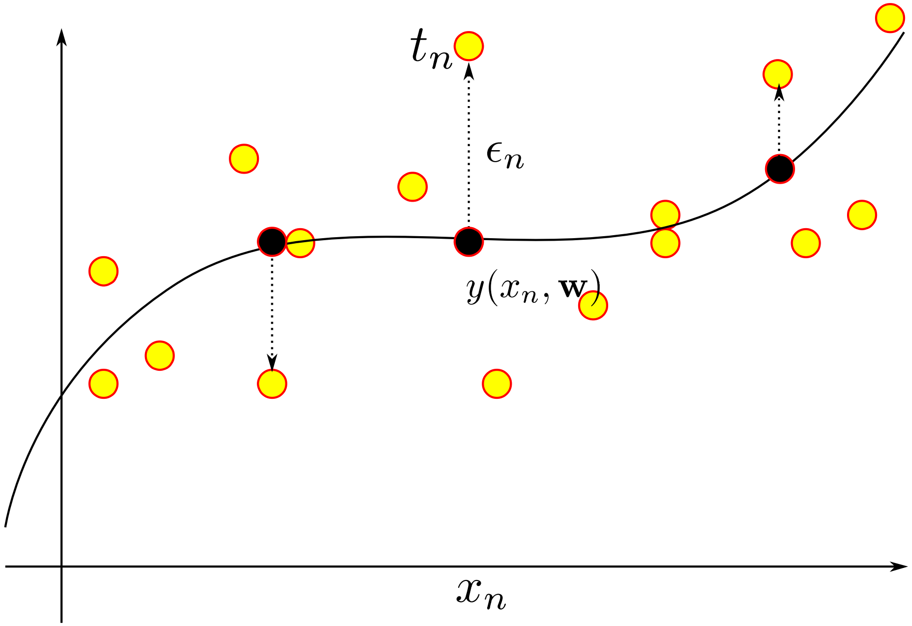
Remember from high school!



Linear regression: Find $\mathbf{w} = (w_0, w_1)$ that minimize

$$\sum_{n=1}^{N}(y(x_n, \mathbf{w}) - t_n)^2 = \sum_{n=1}^{N}(w_0 + w_1 x_n - t_n)^2$$

Least squares: Minimize sum-of-squares error



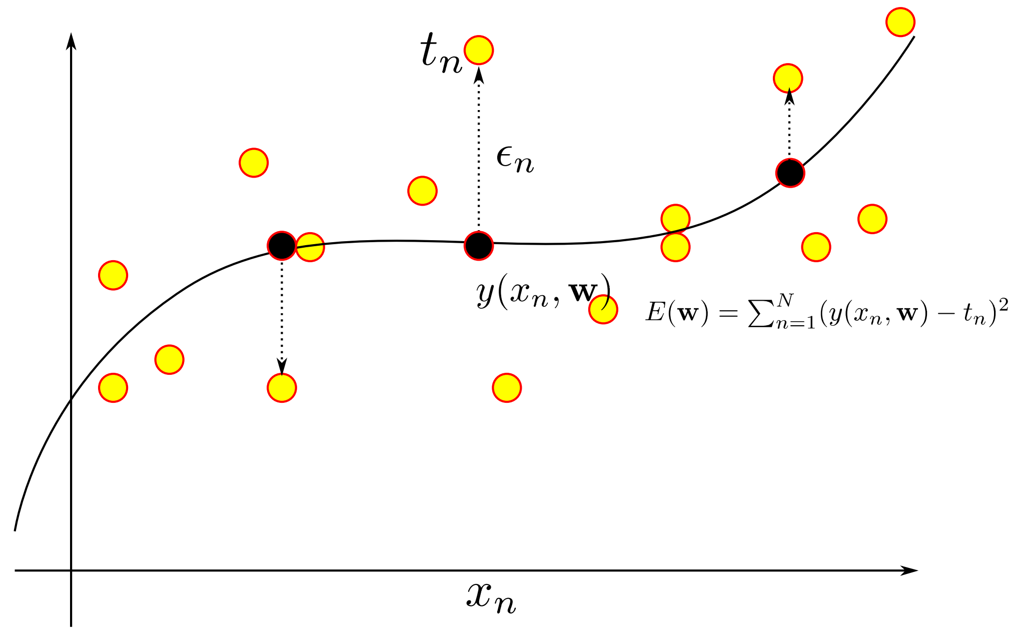
Choose parameters $\mathbf{w}$ for y that minimize

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

called the *sum of squares error*

Has a unique solution because it is a quadratic problem.

Polynomial regression

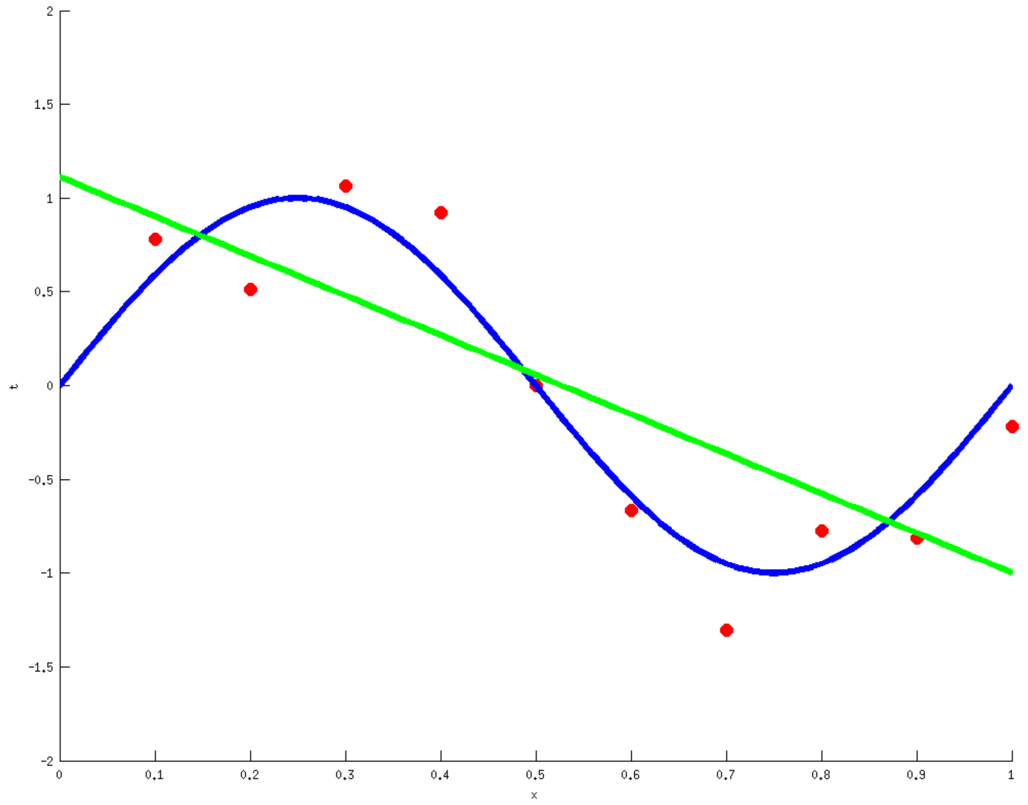


Polynomial regression: Fit a polynomial model

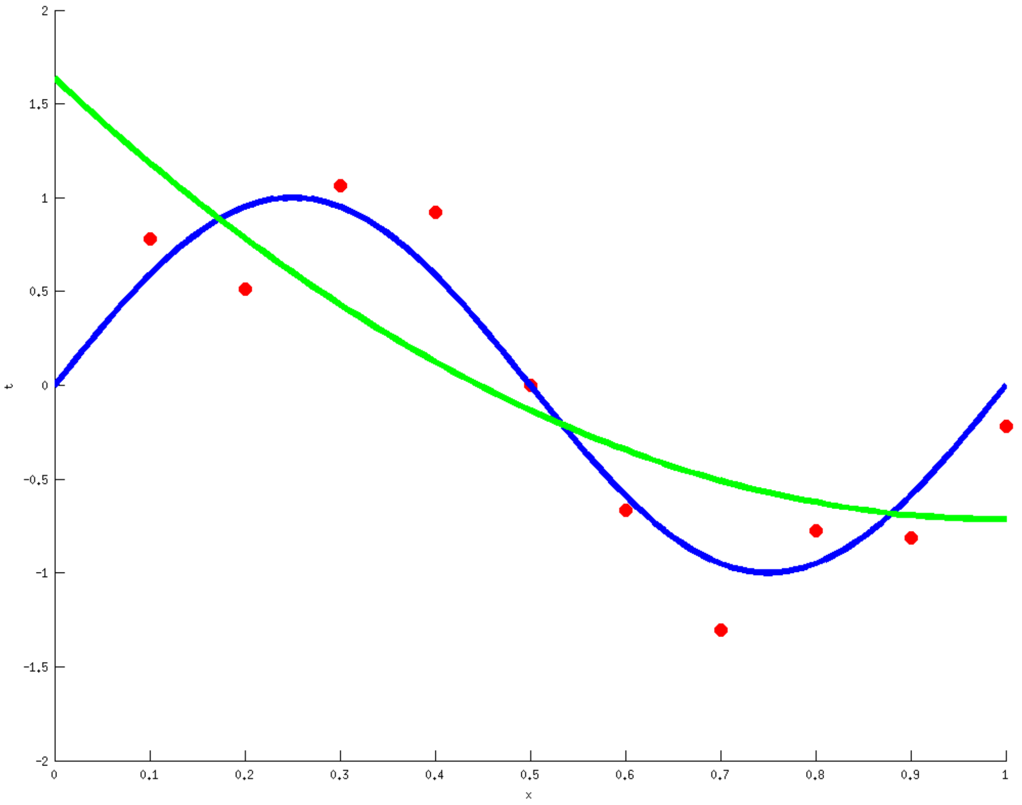
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \mathbf{w}^T \begin{pmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^M \end{pmatrix}$$

Note: Linear in $\mathbf{w}$, not in x.

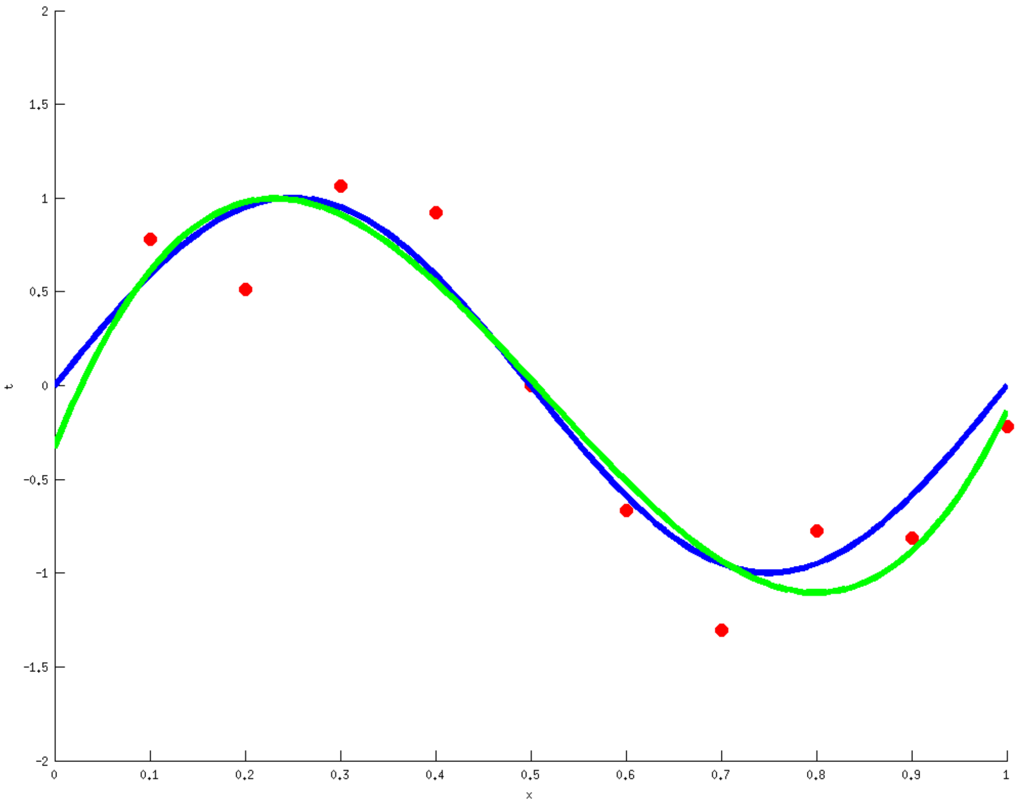
Polynomial regression: M = 1



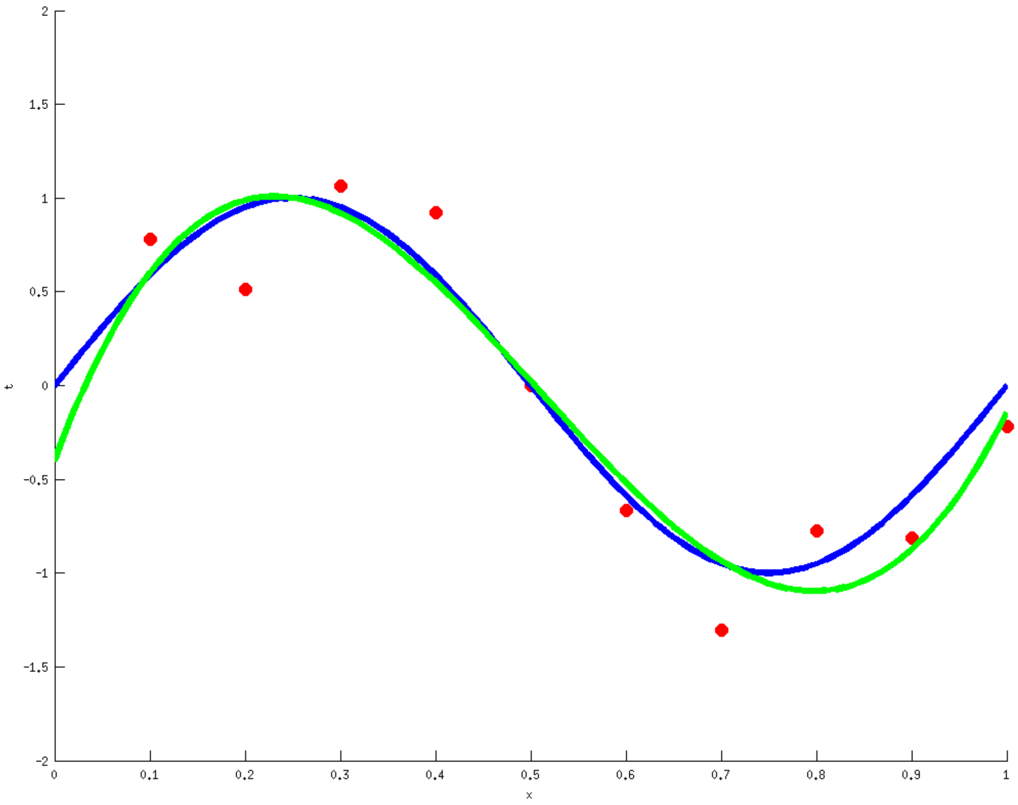
Polynomial regression: M = 2



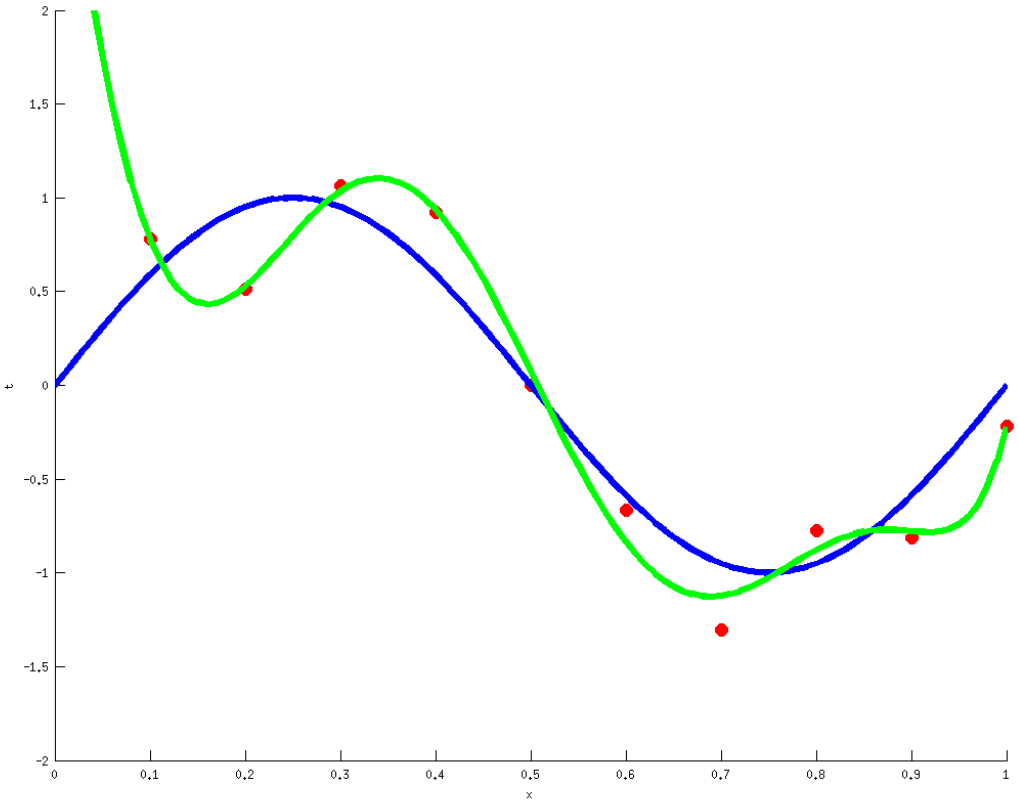
Polynomial regression: M = 3



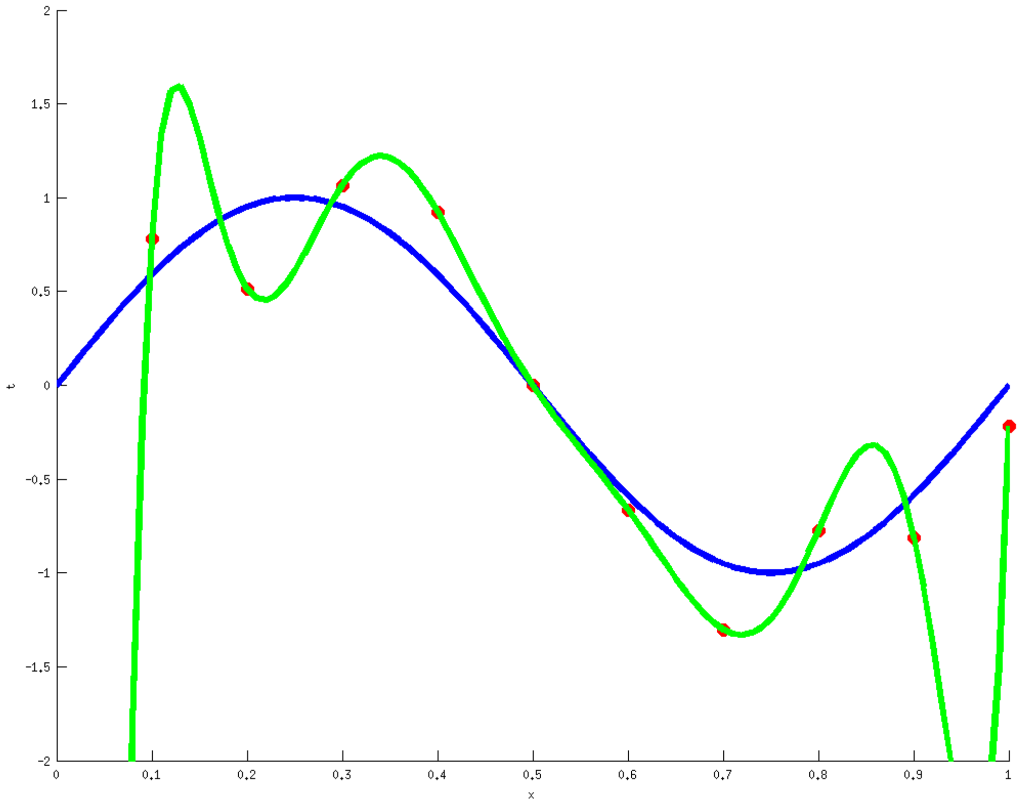
Polynomial regression: M = 4



Polynomial regression: M = 7



Polynomial regression: M = 9

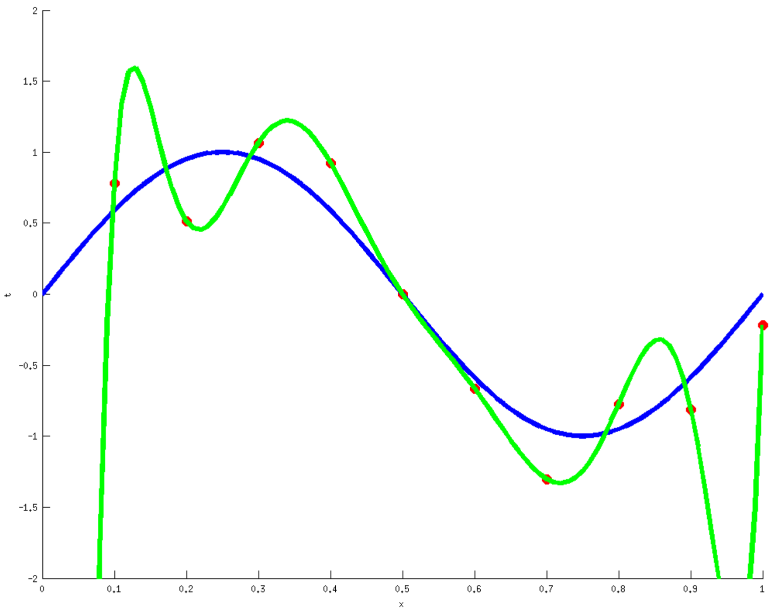


$$\mathrm{E}(\mathbf{w}^{*}) = \sum_{n=1}^{M} (y(x_n) - t_n)^2 = 0!$$

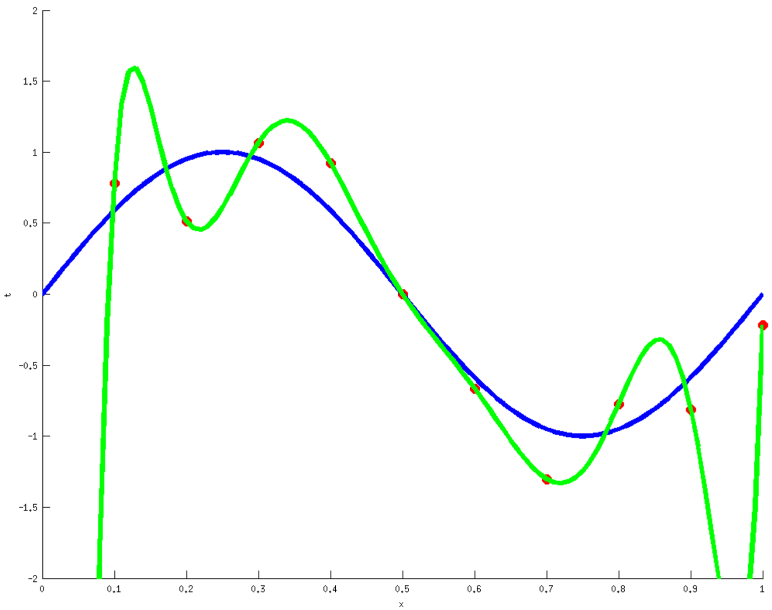
Perfect fit?

Notation: **w*** means
The optimal value of **w**

Can you see any potential problems?

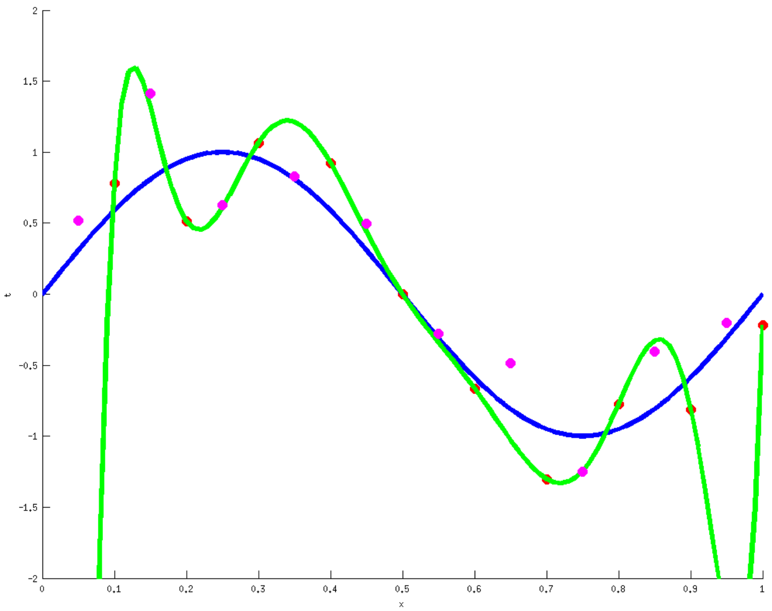


Can you see any potential problems?



- Selecting polynomial degree M?
- Measuring goodness-of-fit?
- Computational solution?

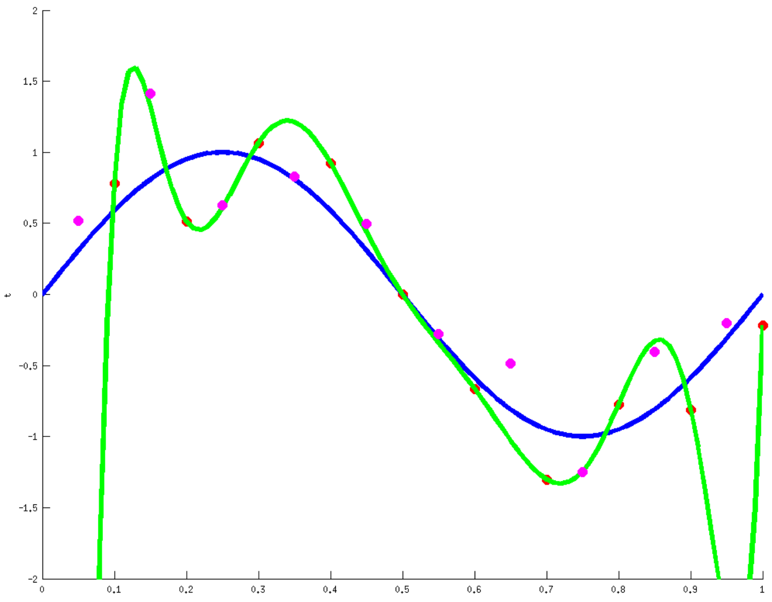
Problem 1: Measuring quality of fit Root Mean Square Error (RMS)



- Root mean square error defined as

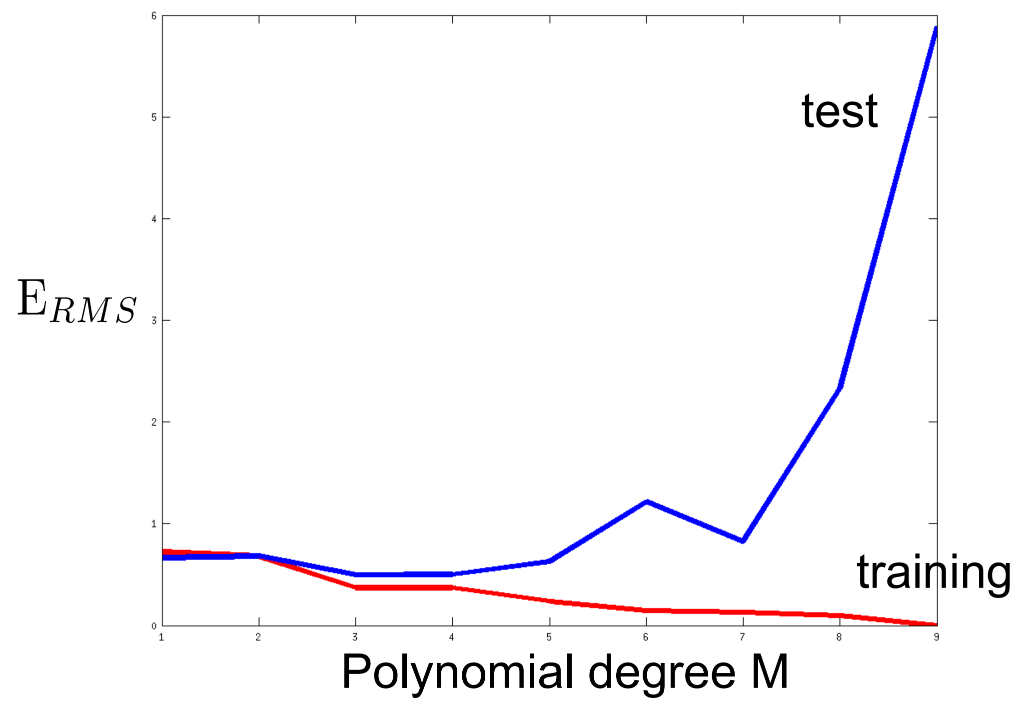
$$\mathrm{E}_{RMS} = \sqrt{2E(\mathbf{w}^*/N)}$$

Problem 1: Measuring quality of fit Root Mean Square Error (RMS)

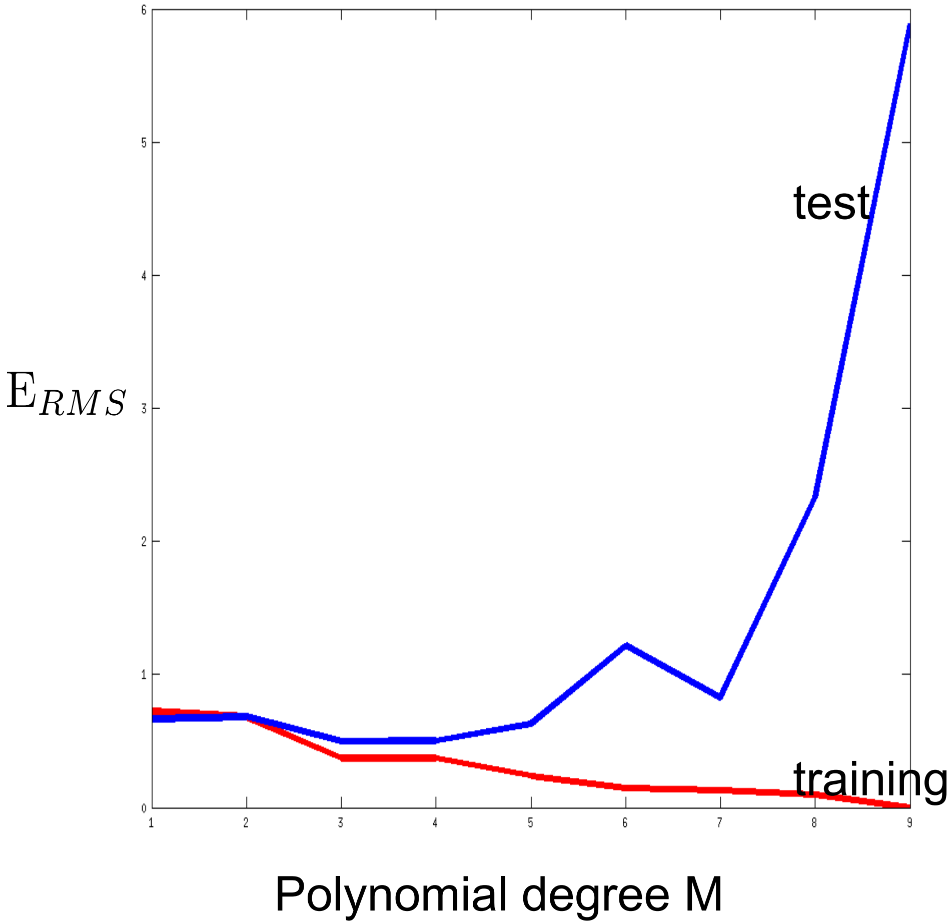


- Root mean square error defined as

$$\mathrm{E}_{RMS} = \sqrt{2E(\mathbf{w}^*/N)}$$



Problem 2: Model selection and generalization – test, validation and training sets



- How to choose an optimal degree M?
- Need to avoid overfitting and lack of generalization!

Problem 2: Model selection and generalization – test, validation and training sets

Approach 1: Split data into training, validation and test set

Training

Learn different models from training set

Validation

Choose best model based on validation set performance

Test

Report performance on test set.

Example:

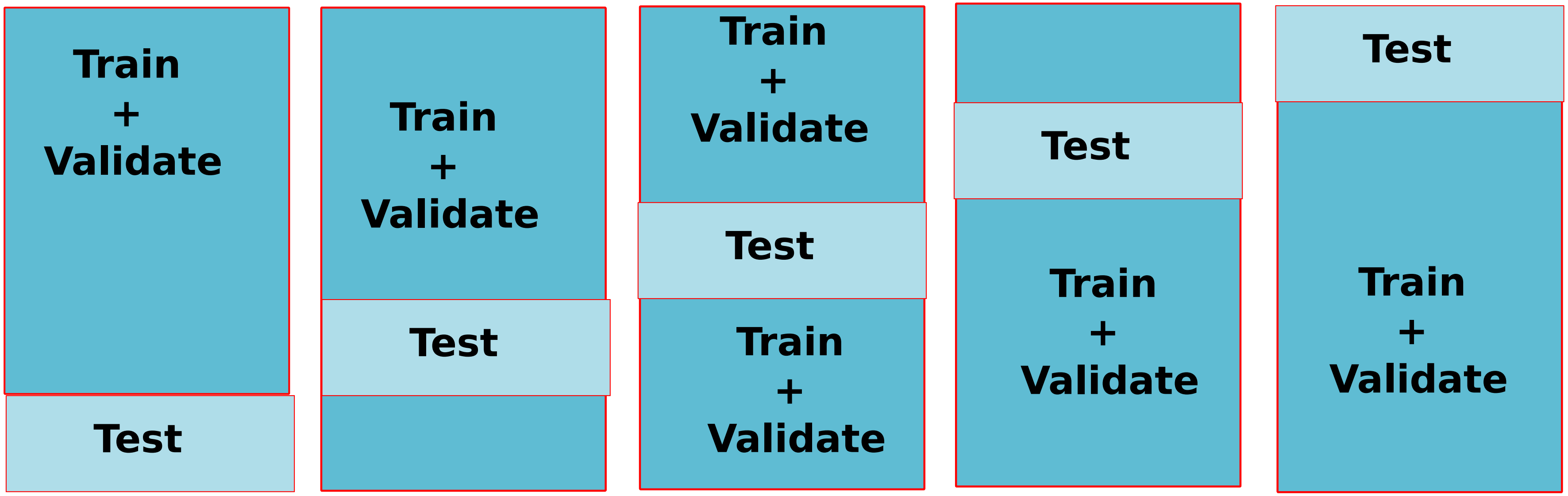
for j = 1 : M
  find optimal parameters $\mathbf{w}_j$
  on Training set
endfor
for j = 1 : M
  compute $E_{RMS}(j)$ on Validation set
end

$$j^* = \mathrm{argmin} E_{RMS}(j)$$

Learned model: $\mathbf{w}_{j^*}$
Report E_{RMS} of learned model on Test set

Problem 2: Model selection and generalization – test, validation and training sets

Approach 2: Cross-validation Loop through different partitions of the data set

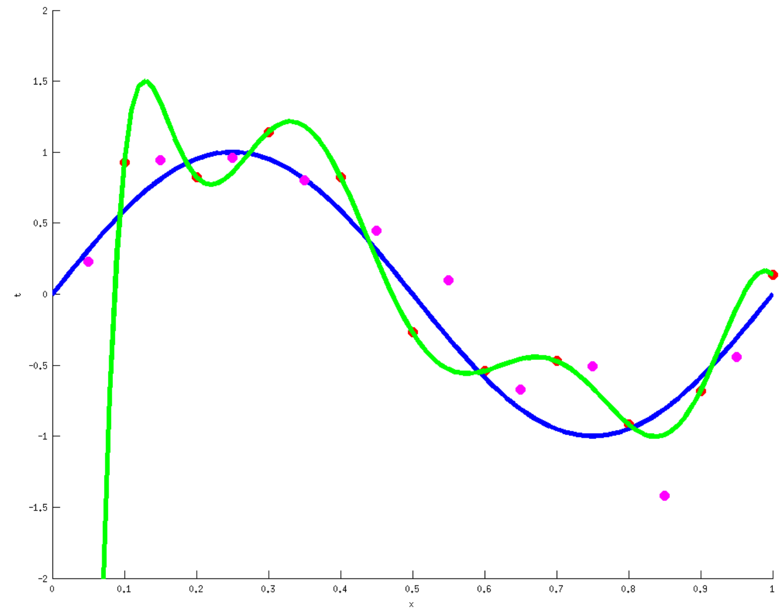


Report average and standard deviation of performance across folds

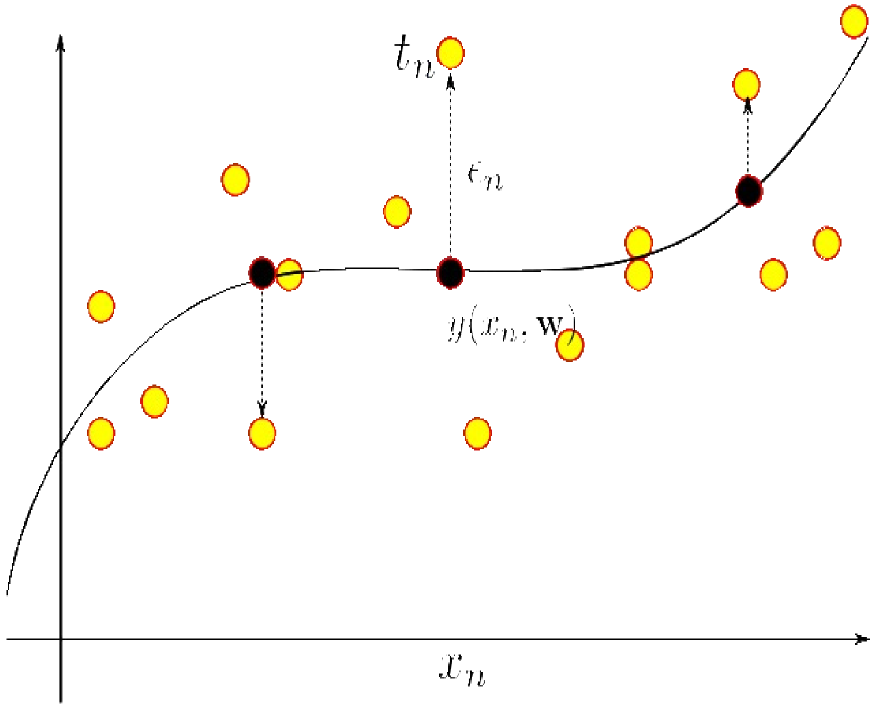
A closer look at overfitting

- The weight vector **w** gives insight into the overfitting problem

M=1	M=2	M=3		M=9
2.3797	5.0814	21.9130		-5410
1.3828	-7.9693	-31.0751		57200
	2.5007	8.7066	.....	-191210
		0.6205		315610
				-297450
				168650
				-57660
				11380
				-1160
				50



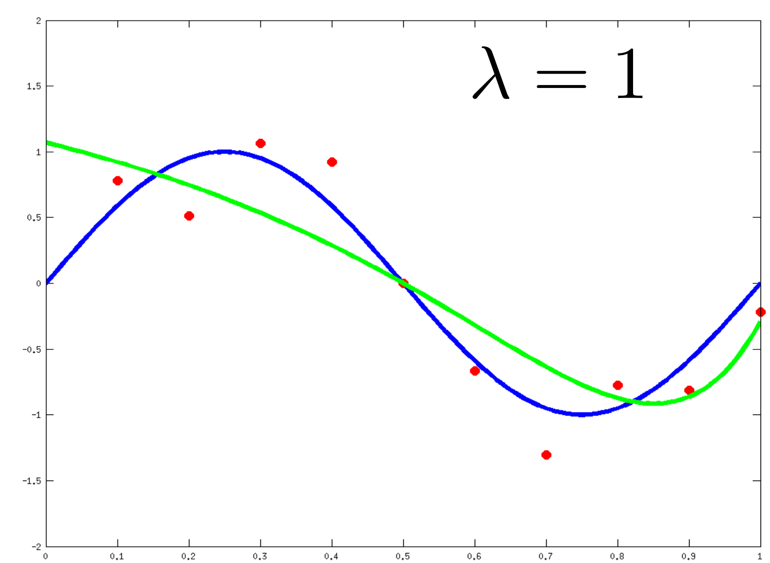
Regularization



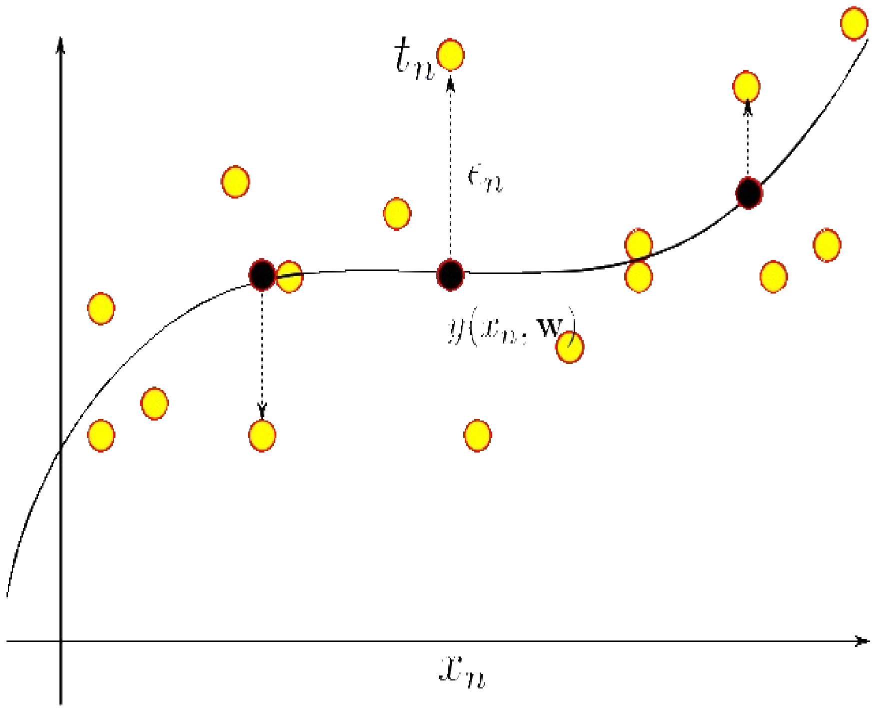
Add a **regularizing** term to the least squares loss function:

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

M=1	M=2	M=3		M=9
2.3797	5.0814	21.9130		-5410
1.3828	-7.9693	-31.0751		57200
	2.5007	8.7066	.....	-191210
		0.6205		315610
				-297450
				168650
				-57660
				11380
				-1160
				50



Regularization

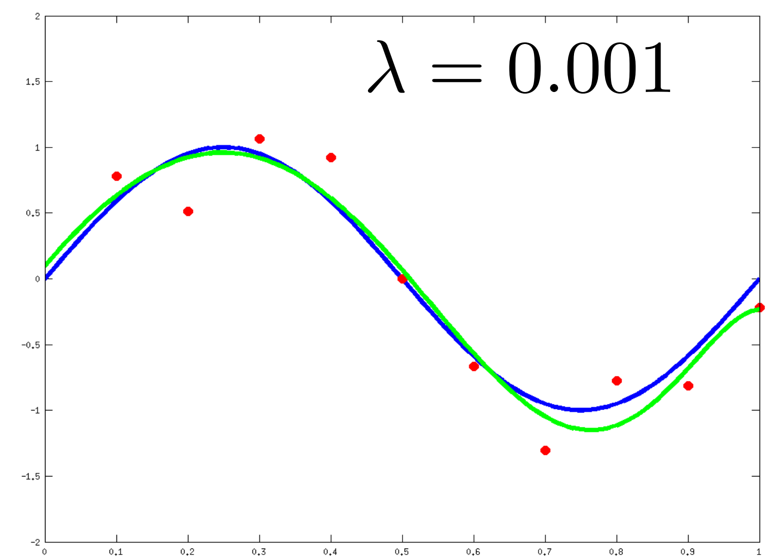


Add a **regularizing** term to the least squares loss function:

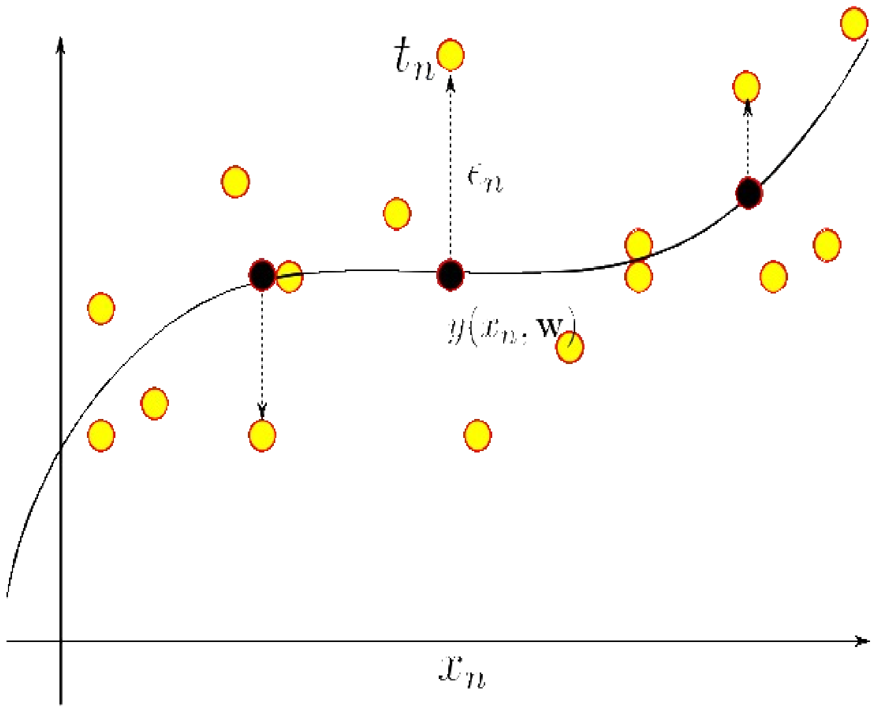
$$E(\mathbf{w}) = \sum_{n=1}^{N} (y(x_n, \mathbf{w}) - t_n)^2 + \lambda \|\mathbf{w}\|^2$$

M=1	M=2	M=3		M=9
2.3797	5.0814	21.9130		-5410
1.3828	-7.9693	-31.0751		57200
	2.5007	8.7066	.....	-191210
		0.6205		315610
				-297450
				168650
				-57660
				11380
				-1160
				50

Weights:
0.0977
6.3715
-9.1020
-11.0693
-0.2834
8.7853
10.5747
6.0683
-1.7071
-9.9701



Regularization

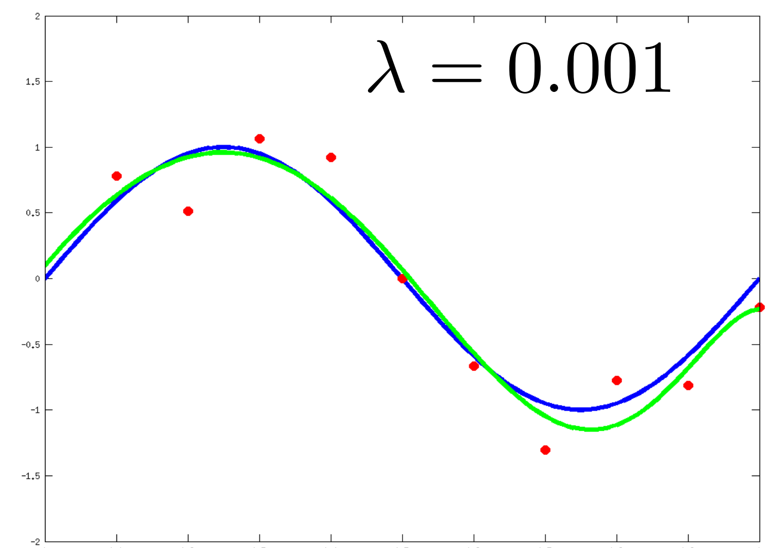


M=1	M=2	M=3		M=9
2.3797	5.0814	21.9130		-5410
1.3828	-7.9693	-31.0751		57200
	2.5007	8.7066	.....	-191210
		0.6205		315610
				-297450
				168650
				-57660
				11380
				-1160
				50

Add a **regularizing** term to the least squares loss function:

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

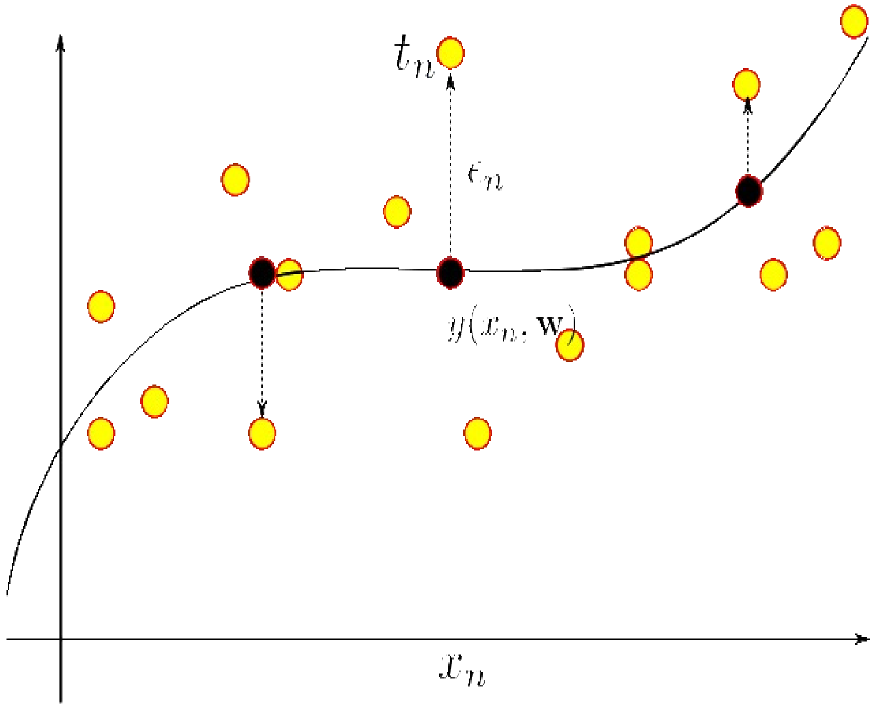
Weights:
0.0977
6.3715
-9.1020
-11.0693
-0.2834
8.7853
10.5747
6.0683
-1.7071
-9.9701



Still have to choose λ

How?

Regularization

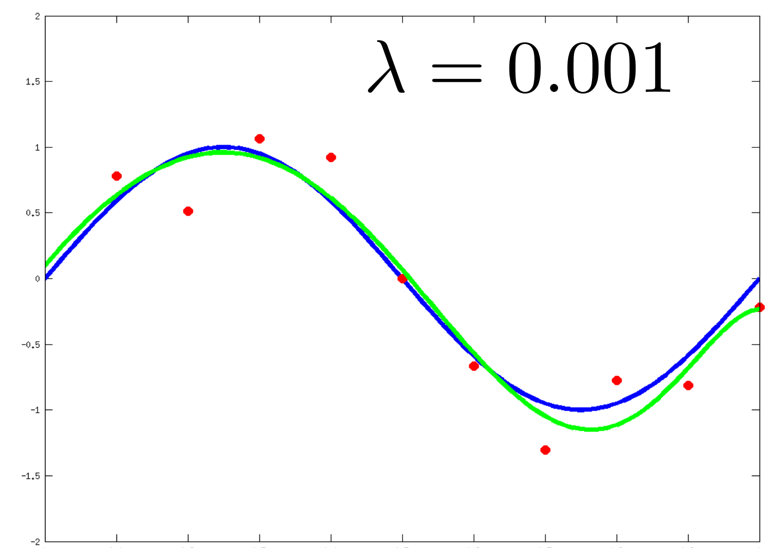


M=1	M=2	M=3		M=9
2.3797	5.0814	21.9130		-5410
1.3828	-7.9693	-31.0751		57200
	2.5007	8.7066	.....	-191210
		0.6205		315610
				-297450
				168650
				-57660
				11380
				-1160
				50

Add a **regularizing** term to the least squares loss function:

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

Weights:
0.0977
6.3715
-9.1020
-11.0693
-0.2834
8.7853
10.5747
6.0683
-1.7071
-9.9701

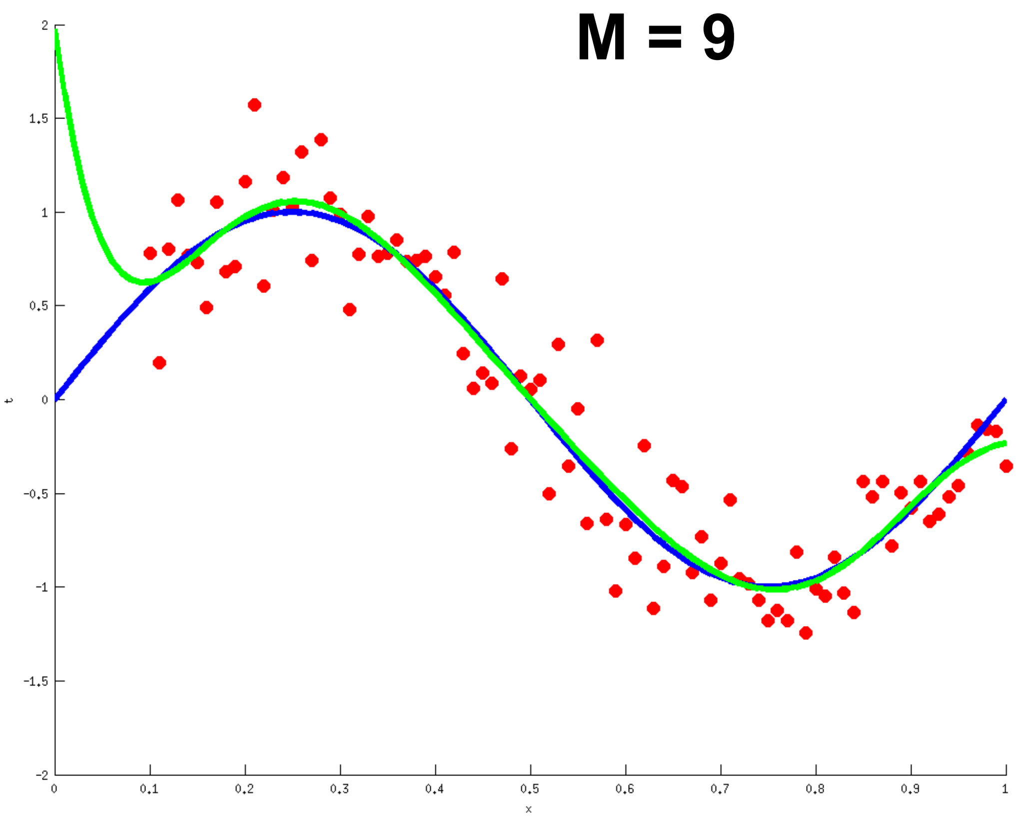


Still have to choose λ

How?

Train/Validate/Test
Cross-validation

Problem 3: Data set size

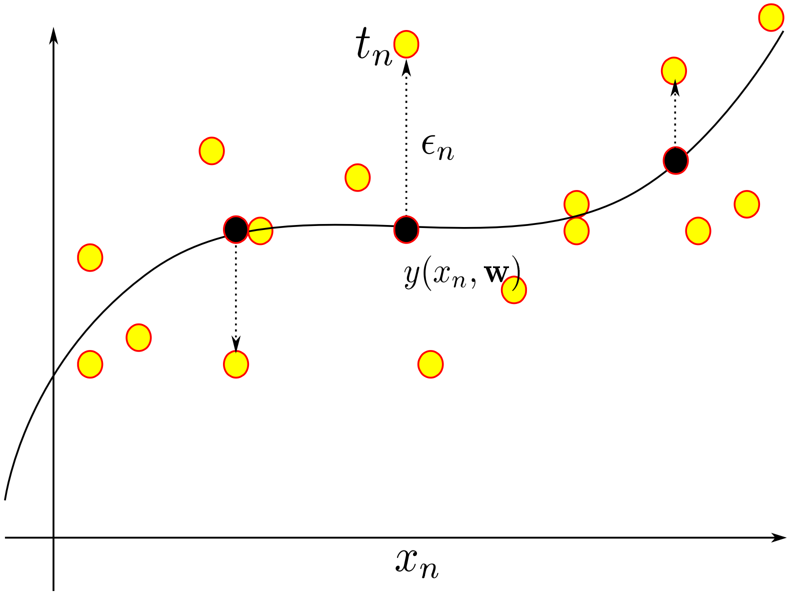


Case: Automated mammographic analysis

- Image texture measurements are predictive of breast cancer
- Can you pose “predict cancer” as a regression problem?
- What are the x and t?
- Given 1000 images with 1000 cancer scores, how would you build and evaluate a regression model?

Probabilistic view of regression

- So far, we have considered regression as a geometric curve-fitting problem



Choose parameters $\mathbf{w}$ for y that minimize

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

called the *sum of squares error*

Has a unique solution because it is a quadratic problem.

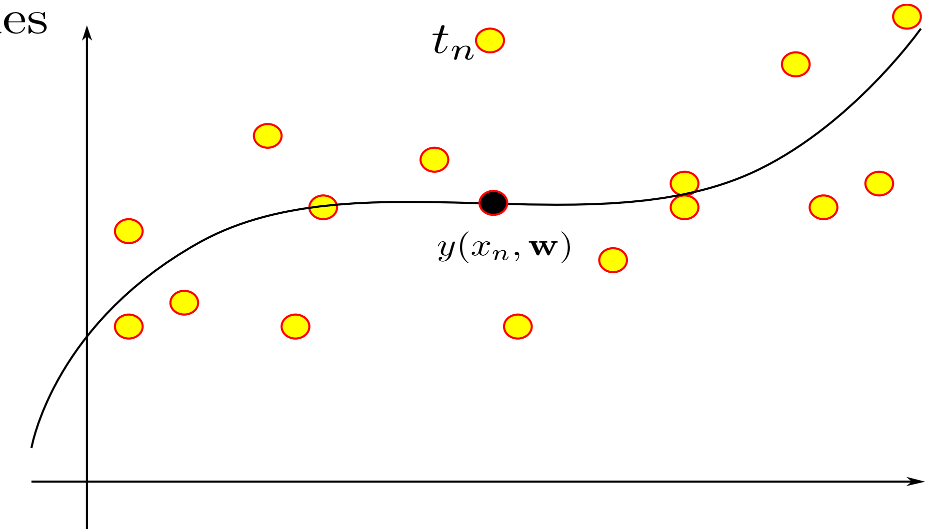
Probabilistic view of regression

N input variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

with target variables

$$\mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



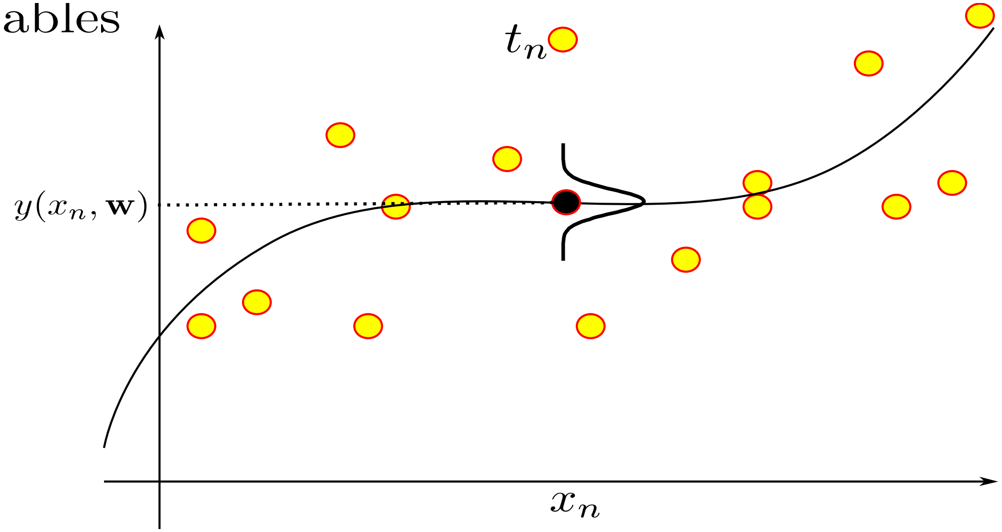
Goal: Learn the rule $y(x, \mathbf{w})$

Assume: For any input value x, the

Probabilistic view of regression

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



Goal: Learn the rule $y(x, \mathbf{w})$

Assume: For any input value x, the corresponding target value t follows a Gaussian distribution

$$\mathcal{N}(t|y(x, \mathbf{w}), \beta^{-1}) \quad \text{with mean } y(x, \mathbf{w}) \text{ and variance } \frac{1}{\beta}$$

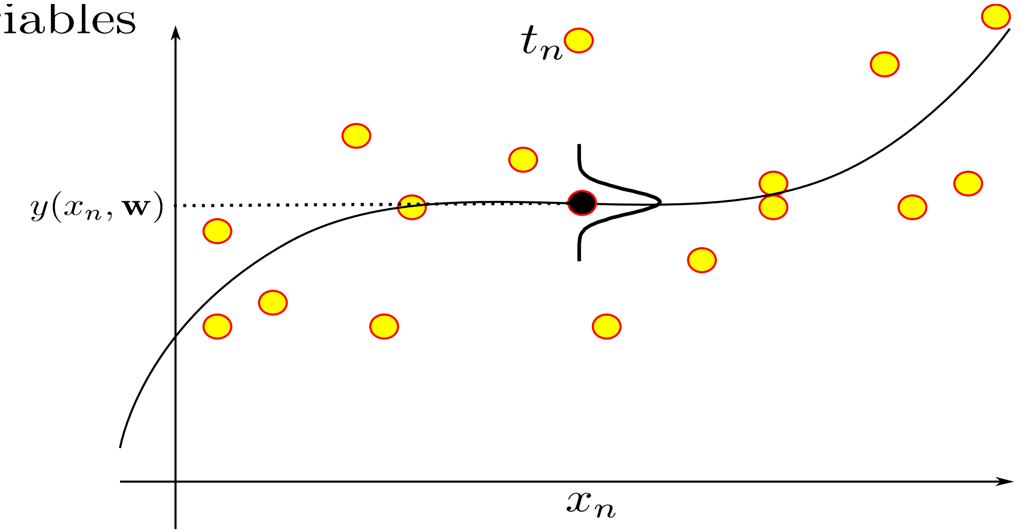
Probabilistic view of regression

N input variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

with target variables

$$\mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



Goal: Learn the rule $y(x, \mathbf{w})$

Assume: For any input value x, the corresponding target value t follows a Gaussian distribution

$$\mathcal{N}(t|y(x, \mathbf{w}), \beta^{-1}) \quad \text{with mean } y(x, \mathbf{w}) \text{ and variance } \frac{1}{\beta}$$

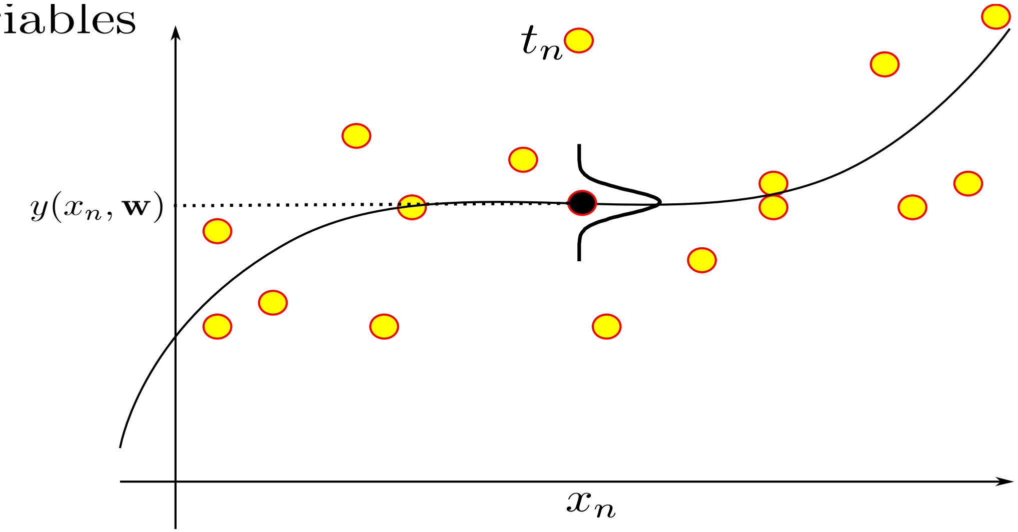
Equivalent formulation:

$$t = y(x, \mathbf{w}) + \epsilon(x) \qquad \text{where the error } \epsilon(x) \text{ follows } \mathcal{N}(0, \beta^{-1})$$

Probabilistic view of regression

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



Goal: Learn the rule $y(x, \mathbf{w})$

Assume: For any input value x, the corresponding target value t follows a Gaussian distribution

$$\mathcal{N}(t|y(x, \mathbf{w}), \beta^{-1}) \quad \text{with mean } y(x, \mathbf{w}) \text{ and variance } \tfrac{1}{\beta}$$

Equivalent formulation:

$$t = y(x, \mathbf{w}) + \epsilon(x) \qquad \text{where the error } \epsilon(x) \text{ follows } \mathcal{N}(0, \beta^{-1})$$

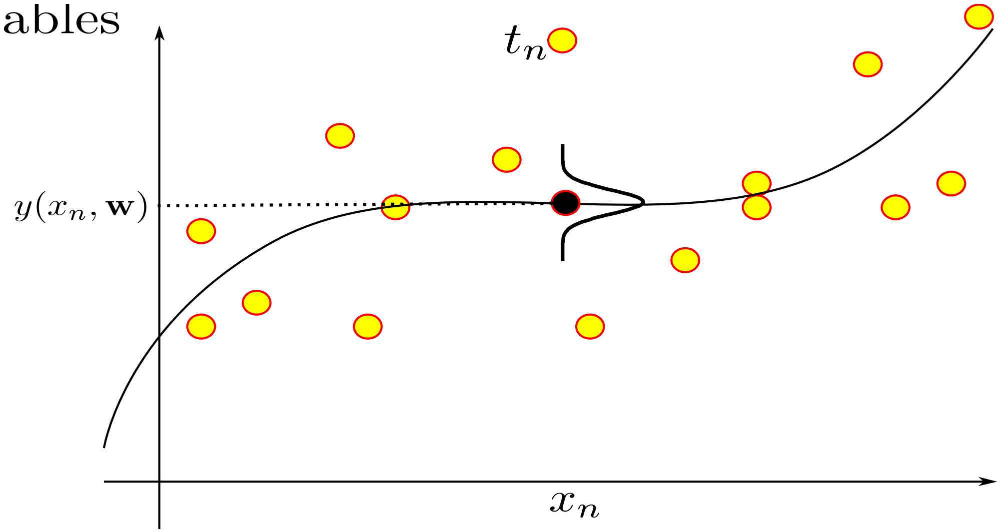
From this we can derive:

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1})$$

Probabilistic view of regression

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



Goal: Learn the rule $y(x, \mathbf{w})$

Assume: For any input value x, the corresponding target value t follows a Gaussian distribution

$$\mathcal{N}(t|y(x, \mathbf{w}), \beta^{-1}) \quad \text{with mean } y(x, \mathbf{w}) \text{ and variance } \frac{1}{\beta}$$

Equivalent formulation:

$$t = y(x, \mathbf{w}) + \epsilon(x) \qquad \text{where the error } \epsilon(x) \text{ follows } \mathcal{N}(0, \beta^{-1})$$

From this we can derive:

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1})$$

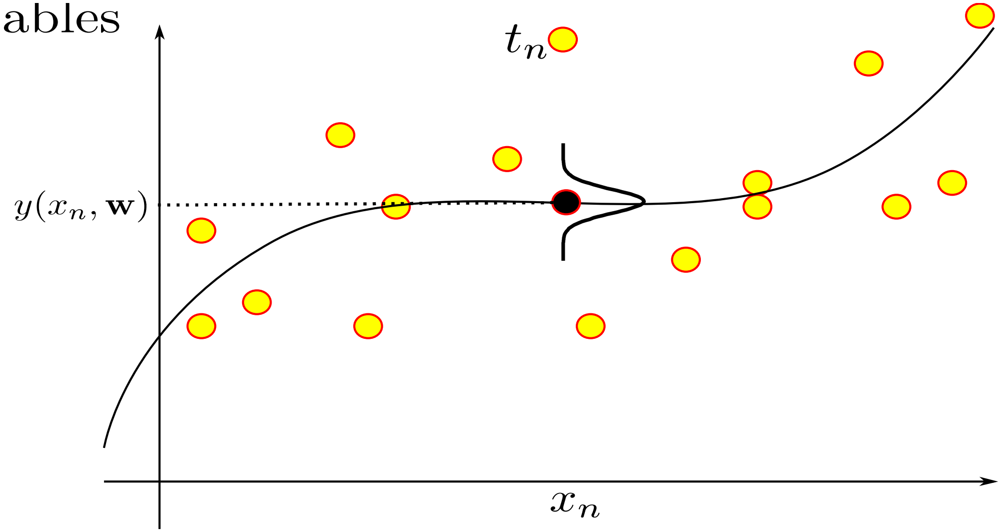
Joint probability of observing $\mathbf{t}$ given input variables $\mathbf{x}$ and model

$$t = y(x, \mathbf{w}) + \epsilon(x)$$

Probabilistic view of regression

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



Goal: Learn the rule $y(x, \mathbf{w})$

Assume: For any input value x, the corresponding target value t follows a Gaussian distribution

$$\mathcal{N}(t|y(x, \mathbf{w}), \beta^{-1}) \quad \text{with mean } y(x, \mathbf{w}) \text{ and variance } \frac{1}{\beta}$$

Equivalent formulation:

$$t = y(x, \mathbf{w}) + \epsilon(x) \qquad \text{where the error } \epsilon(x) \text{ follows } \mathcal{N}(0, \beta^{-1})$$

From this we can derive:

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1})$$

Joint probability of observing $\mathbf{t}$ given input variables $\mathbf{x}$ and model

$$t = y(x, \mathbf{w}) + \epsilon(x)$$

Likelihood of data $\mathbf{t}$ under model fixed by $\mathbf{w}, \mathbf{x}$

Recall from Lecture 2:

- Maximum Likelihood estimates

Find the model parameters

$$\mathbf{w}$$

that maximize the joint probability

$$\mathrm{p}(\mathrm{D} \mid \mathbf{w})$$

of observing the data given the model

- Maximum a posteriori estimates

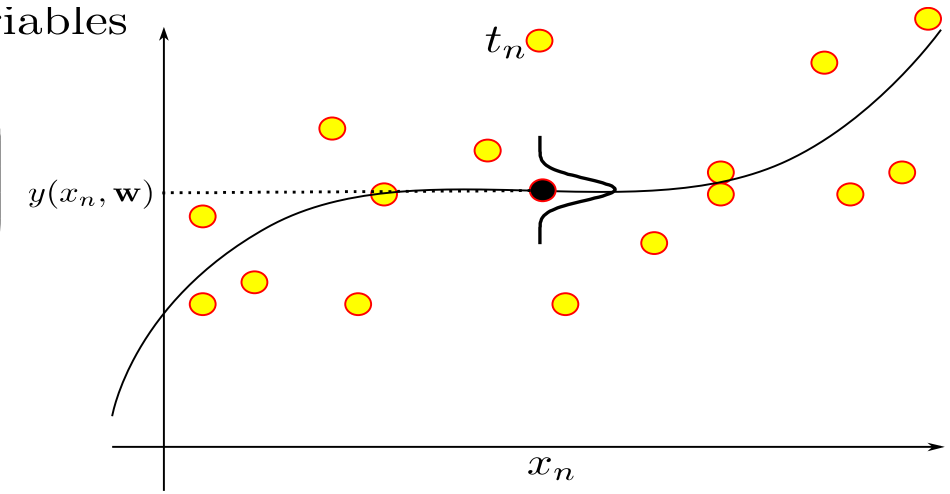
Find the most likely model parameters given the data, that is find the model parameters

$$\mathrm{p}(\mathbf{w} \mid \mathrm{D}) \propto p(D|\mathbf{w})p(\mathbf{w})$$

The Maximum Likelihood solution to the Regression Problem

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



Assume: Gaussian noise model

$$\mathcal{N}(t|y(x,\mathbf{w}), \beta^{-1})$$

Likelihood of data $\mathbf{t}$ under model fixed by $\mathbf{w}, \mathbf{x}$

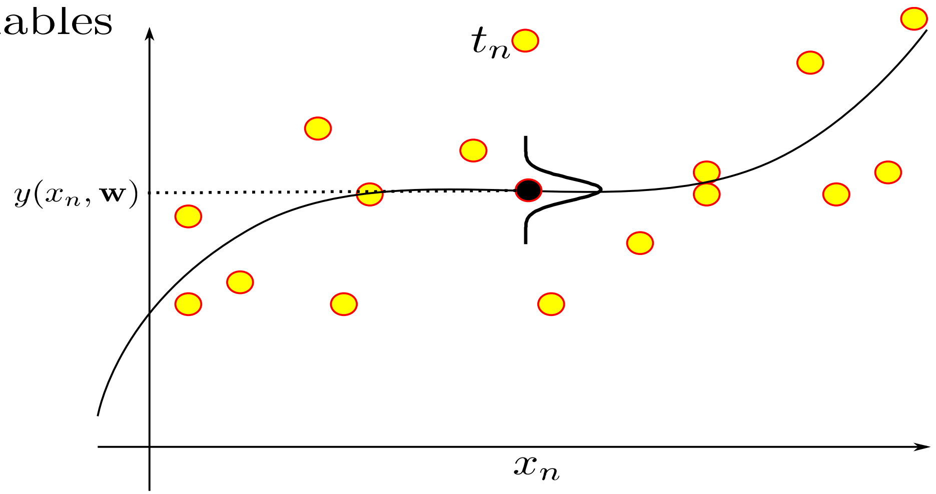
$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1}) = \prod_{n=1}^{N} \sqrt{\frac{\beta}{2\pi}} e^{-\frac{\beta}{2}(y_n(x_n,\mathbf{w})-t_n)^2}$$

$$\ln(x^a) = a \ln x$$
$$\ln(ab) = \ln a + \ln b$$

The Maximum Likelihood solution to the Regression Problem

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



Assume: Gaussian noise model

$$\mathcal{N}(t|y(x,\mathbf{w}), \beta^{-1})$$

Likelihood of data $\mathbf{t}$ under model fixed by $\mathbf{w}, \mathbf{x}$

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1}) = \prod_{n=1}^{N} \sqrt{\frac{\beta}{2\pi}} e^{-\frac{\beta}{2}(y_n(x_n,\mathbf{w})-t_n)^2}$$

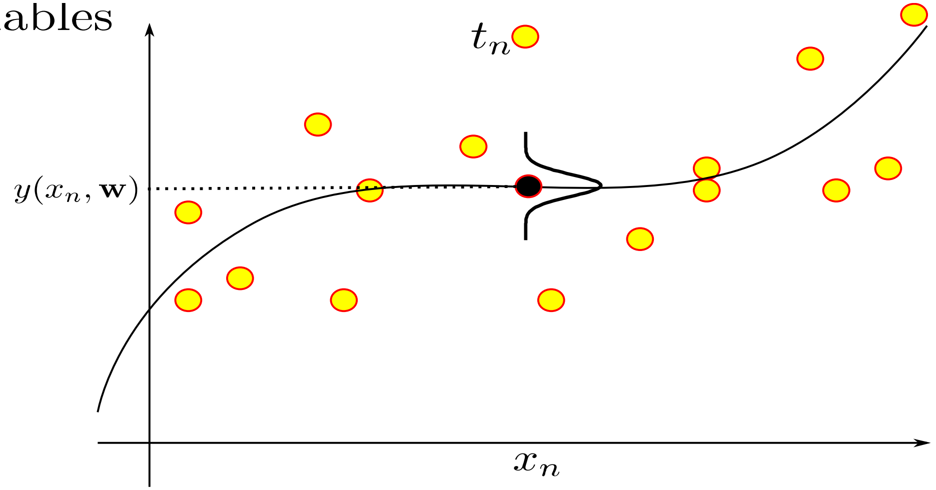
$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \ln(\sqrt{\frac{\beta}{2\pi}}^{N}) + \sum_{n=1}^{N} (-\frac{\beta}{2})(y(x_n, \mathbf{w}) - t_n)^2$$

$$\ln(x^a) = a \ln x$$
$$\ln(ab) = \ln a + \ln b$$

The Maximum Likelihood solution to the Regression Problem

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



Assume: Gaussian noise model

$$\mathcal{N}(t|y(x, \mathbf{w}), \beta^{-1})$$

Likelihood of data $\mathbf{t}$ under model fixed by $\mathbf{w}, \mathbf{x}$

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1}) = \prod_{n=1}^{N} \sqrt{\frac{\beta}{2\pi}} e^{-\frac{\beta}{2}(y_n(x_n, \mathbf{w}) - t_n)^2}$$

$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \ln(\sqrt{\frac{\beta}{2\pi}}^{N}) + \sum_{n=1}^{N} (-\frac{\beta}{2})(y(x_n, \mathbf{w}) - t_n)^2$$

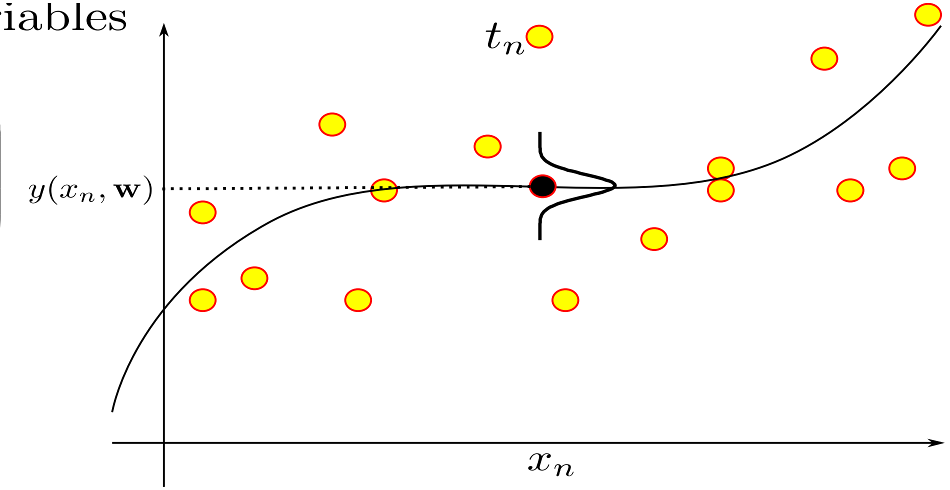
$$= \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \frac{\beta}{2} \sum_{n=1}^{N} (y(x_n, \mathbf{w}) - t_n)^2$$

$$\ln(x^a) = a \ln x$$
$$\ln(ab) = \ln a + \ln b$$

The Maximum Likelihood solution to the Regression Problem

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



Assume: Gaussian noise model

$$\mathcal{N}(t|y(x,\mathbf{w}),\beta^{-1})$$

Likelihood of data $\mathbf{t}$ under model fixed by $\mathbf{w}, \mathbf{x}$

$$p(\mathbf{t}|\mathbf{x},\mathbf{w},\beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n,\mathbf{w}),\beta^{-1}) = \prod_{n=1}^{N} \sqrt{\frac{\beta}{2\pi}} e^{-\frac{\beta}{2}(y_n(x_n,\mathbf{w})-t_n)^2}$$

$$\ln p(\mathbf{t}|\mathbf{x},\mathbf{w},\beta) = \ln(\sqrt{\tfrac{\beta}{2\pi}}^{N}) + \sum_{n=1}^{N}(-\tfrac{\beta}{2})(y(x_n,\mathbf{w}) - t_n)^2$$

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

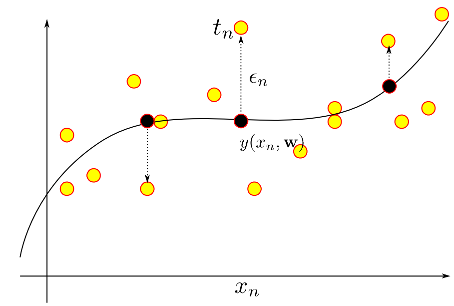
so maximizing the likelihood is equivalent to minimizing

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

$$\ln(x^a) = a\ln x$$
$$\ln(ab) = \ln a + \ln b$$

The Maximum Likelihood solution to the Regression Problem

- The **geometric least-squares** curve-fitting definition of regression is **equivalent** to the **Maximum Likelihood** solution for regression **assuming** that the noise is i.i.d. Gaussian distributed



Choose parameters $\mathbf{w}$ for y that minimize

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

called the *sum of squares error*

Has a unique solution because it is a quadratic problem.

- Maximum likelihood: Find model that maximizes probability of data given model

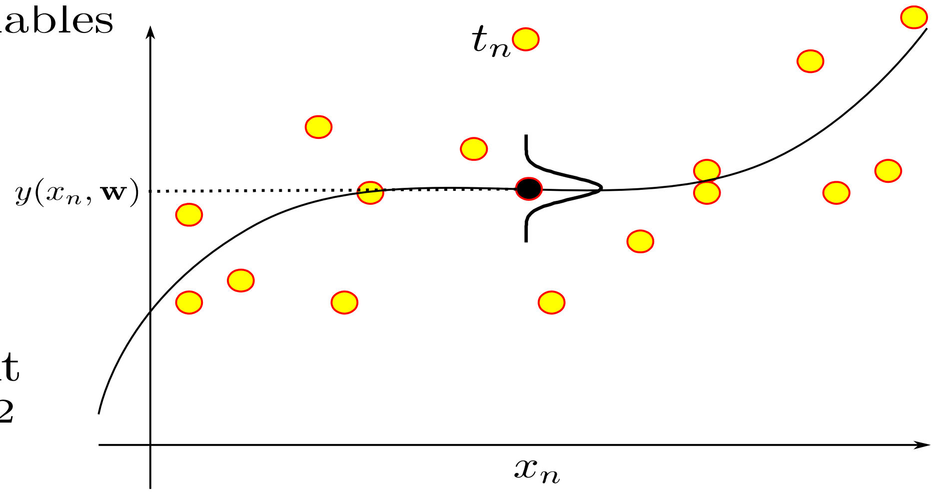
$$\mathrm{p}(\mathrm{D} \mid \mathbf{w})$$

The Maximum Likelihood solution to the Regression Problem

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$

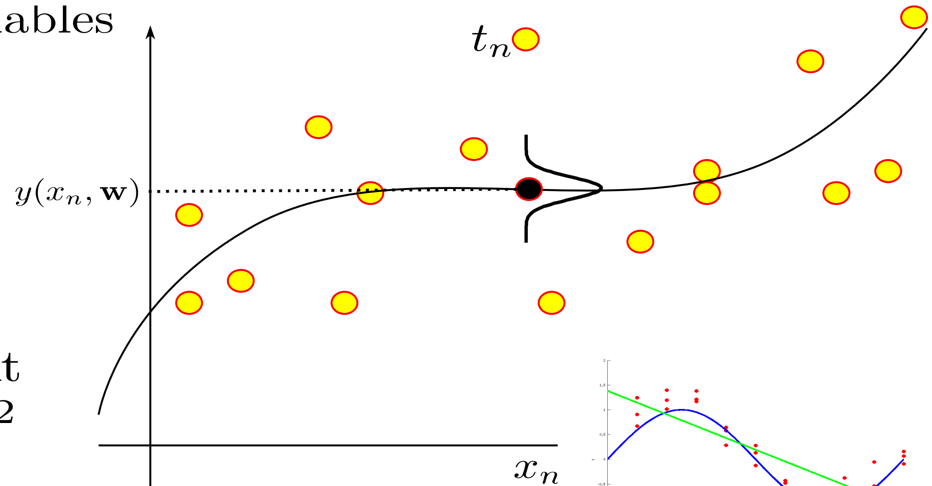
Maximizing the likelihood is equivalent to minimizing $\sum_{n=1}^{N} (y(x_n, \mathbf{w}) - t_n)^2$



The Maximum Likelihood solution to the Regression Problem

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$

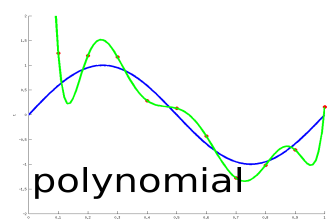


Maximizing the likelihood is equivalent to minimizing $\sum_{n=1}^{N}(y(x_n, \mathbf{w}) - t_n)^2$

So far we have considered regression models

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_D x_D$$

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



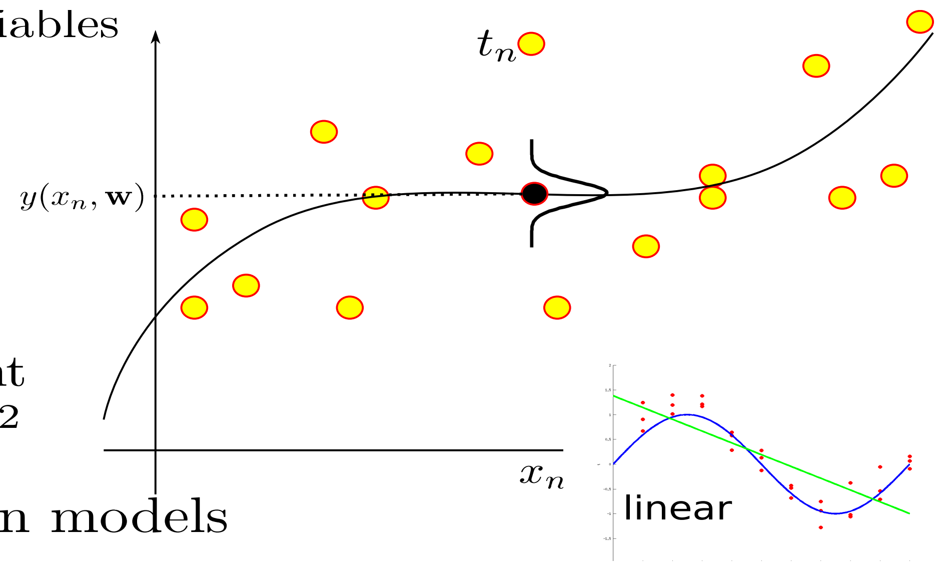
The Maximum Likelihood solution to the Regression Problem

N input variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

with target variables

$$\mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$

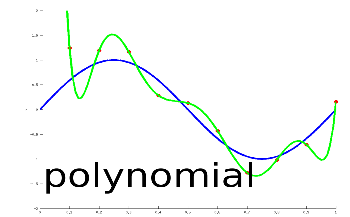


Maximizing the likelihood is equivalent to minimizing $\sum_{n=1}^{N}(y(x_n, \mathbf{w}) - t_n)^2$

So far we have considered regression models

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_D x_D$$

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



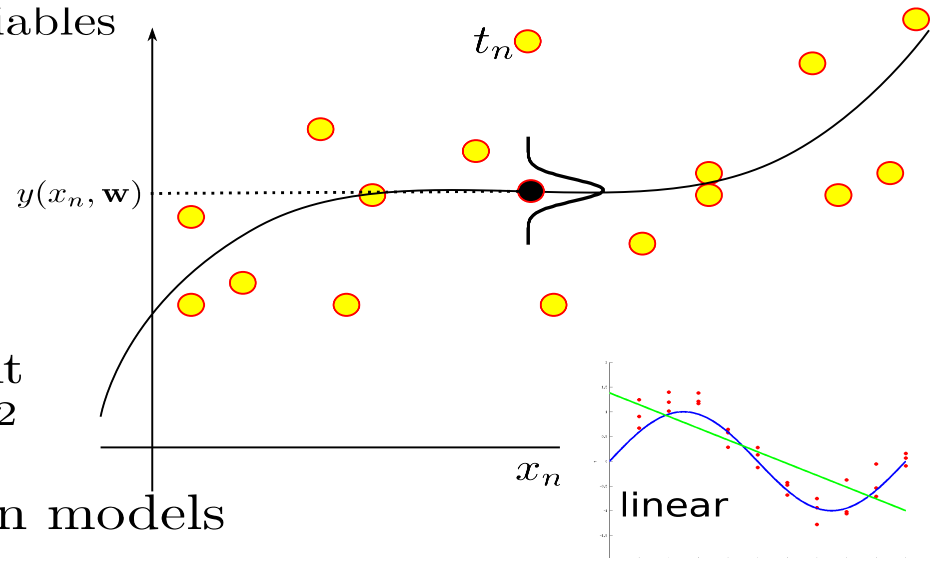
Consider the more general model

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$$

The Maximum Likelihood solution to the Regression Problem

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$

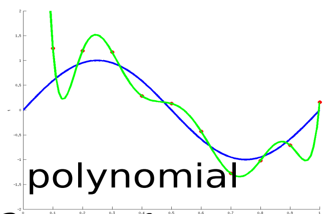


Maximizing the likelihood is equivalent to minimizing $\sum_{n=1}^{N}(y(x_n, \mathbf{w}) - t_n)^2$

So far we have considered regression models

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_D x_D$$

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



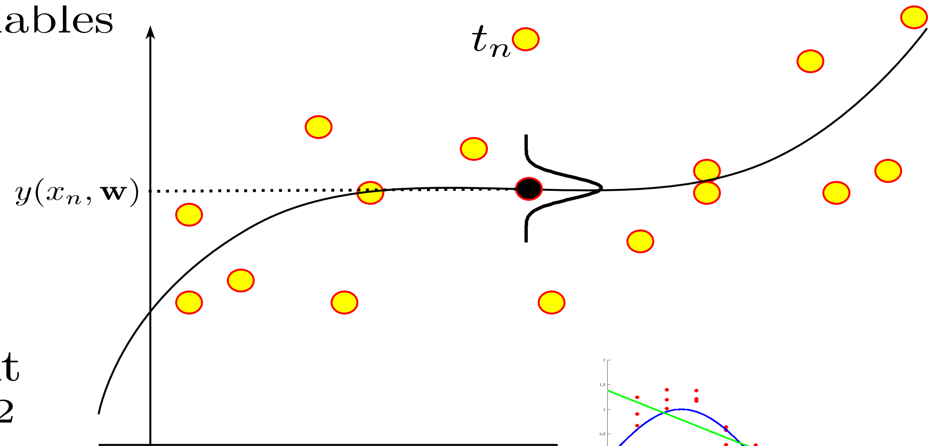
Consider the more general model

$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$ where $\{\phi_j(\mathbf{x})\}$ are **basis functions**

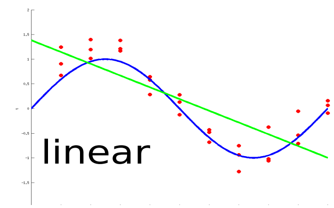
The Maximum Likelihood solution to the Regression Problem

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



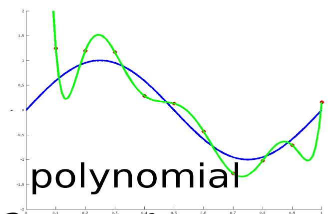
Maximizing the likelihood is equivalent to minimizing $\sum_{n=1}^{N}(y(x_n, \mathbf{w}) - t_n)^2$



So far we have considered regression models

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_D x_D$$

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



Consider the more general model

$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$ where $\{\phi_j(\mathbf{x})\}$ are **basis functions**

For the sake of pretty formulas: define $\phi_o(\mathbf{x}) := 1$

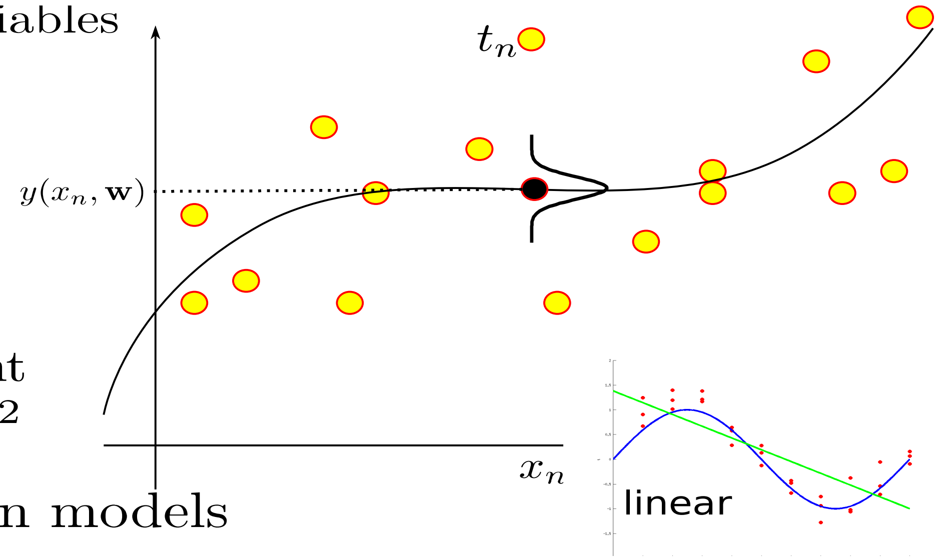
Then $y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$

$$\text{where } \mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{pmatrix} \text{and } \boldsymbol{\phi} = \begin{pmatrix} \phi_0 \\ \phi_1 \\ \vdots \\ \phi_{M-1} \end{pmatrix}$$

The Maximum Likelihood solution to the Regression Problem

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$

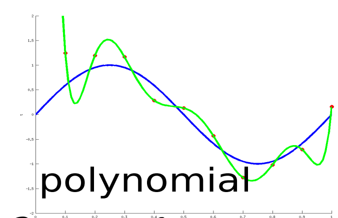


Maximizing the likelihood is equivalent to minimizing $\sum_{n=1}^{N}(y(x_n, \mathbf{w}) - t_n)^2$

So far we have considered regression models

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_D x_D$$

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



Consider the more general model

$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$ where $\{\phi_j(\mathbf{x})\}$ are **basis functions**

For the sake of pretty formulas: define $\phi_o(\mathbf{x}) := 1$

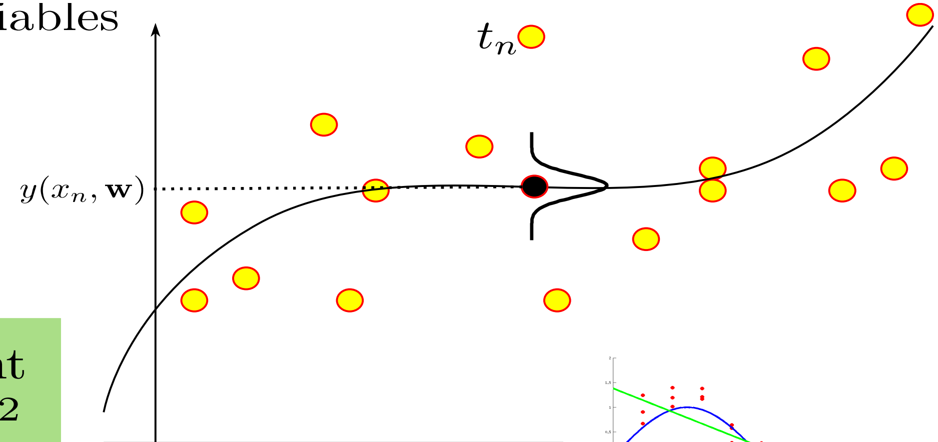
Then $y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$

where $\mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{pmatrix}$ and $\boldsymbol{\phi} = \begin{pmatrix} \phi_0 \\ \phi_1 \\ \vdots \\ \phi_{M-1} \end{pmatrix}$

* Linear model (linear in $\mathbf{w}$)
* Nonlinear $y(\mathbf{x}, \mathbf{w})$ if the ϕ_i are nonlinear

The Maximum Likelihood solution to the Regression Problem

N input variables $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$ with target variables $\mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$

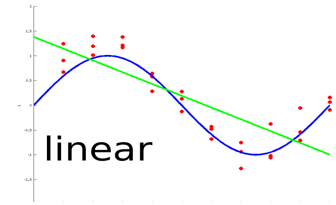


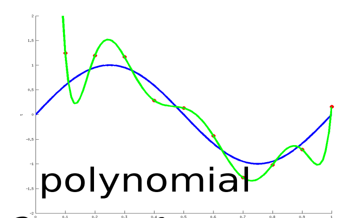
Maximizing the likelihood is equivalent to minimizing $\sum_{n=1}^{N}(y(x_n, \mathbf{w}) - t_n)^2$

So far we have considered regression models

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_D x_D$$

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





Consider the more general model

$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$ where $\{\phi_j(\mathbf{x})\}$ are **basis functions**

For the sake of pretty formulas: define $\phi_o(\mathbf{x}) := 1$

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

where $\mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{pmatrix}$ and $\boldsymbol{\phi} = \begin{pmatrix} \phi_0 \\ \phi_1 \\ \vdots \\ \phi_{M-1} \end{pmatrix}$

- $*$ Linear model (linear in $\mathbf{w}$)
- $*$ Nonlinear $y(\mathbf{x}, \mathbf{w})$ if the ϕ_i are nonlinear

Compute ML solution

Minimizing $\sum_{n=1}^{N}(y(\mathbf{x}_n,\mathbf{w})-t_n)^2$ **when** $y(\mathbf{x},\mathbf{w})=\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x})$.

$$\begin{aligned}
&\frac{\partial}{\partial w_i}[\textstyle\sum_{n=1}^{N}(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)-t_n)^2]\\
&=\textstyle\sum_{n=1}^{N}\frac{\partial}{\partial w_i}[(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)-t_n)^2]\\
&=\textstyle\sum_{n=1}^{N}2(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)-t_n)\cdot\frac{\partial}{\partial w_i}(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)-t_n)\\
&=2\textstyle\sum_{n=1}^{N}(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)-t_n)\cdot\phi_i(\mathbf{x}_n)-t_n)=0 \qquad \text{for all } i
\end{aligned}$$

Since $\boldsymbol{\phi}(\bar{x})^T=(\phi_0(\mathbf{x}),\phi_1(\mathbf{x}),\ldots,\phi_{M-1}(\mathbf{x})$, we get

$$\sum_{n=1}^{N}\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)\boldsymbol{\phi}(\mathbf{x}_n)^T-\sum_{n=1}^{N}t_n\boldsymbol{\phi}(\mathbf{x}_n)^T=0,$$

or

$$0=\mathbf{w}^T\sum_{n=1}^{N}\boldsymbol{\phi}(\mathbf{x}_n)\boldsymbol{\phi}(\mathbf{x}_n)^T-\sum_{n=1}^{N}t_n\boldsymbol{\phi}(\mathbf{x}_n)^T \quad (*)$$

OBS!
Highly relevant
for assignment 2...

Setting $\Phi=\begin{pmatrix}\phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \ldots & \phi_{M-1}(\mathbf{x}_1)\\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \ldots & \phi_{M-1}(\mathbf{x}_2)\\ \vdots & & & \vdots\\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \ldots & \phi_{M-1}(\mathbf{x}_N)\end{pmatrix}$

we rewrite $(*)$ as $0=\mathbf{w}^T(\Phi^T\Phi)-\mathbf{t}^T\Phi$

$$\begin{aligned}
&\Rightarrow \mathbf{w}^T(\Phi^T\Phi)=\mathbf{t}^T\Phi\\
&\Rightarrow (\Phi^T\Phi)^T\mathbf{w}=(\Phi^T\Phi)\mathbf{w}=\Phi^T\mathbf{t} \quad \text{(transpose)}\\
&\Rightarrow \mathbf{w}=(\Phi^T\Phi)^{-1}\Phi^T\mathbf{t}
\end{aligned}$$

What can go wrong?

What can go wrong?

- Overfitting
- Lack of generalization
- Training samples have to represent “typical” samples
- Curse of dimensionality

Curse of Dimensionality

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

$$\mathrm{y}(\mathbf{x}, \mathbf{w}) = \mathrm{w}_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D} \sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D} \sum_{j=1}^{D} \sum_{k=1}^{D} w_{ijk} x_i x_j x_k$$

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

Case: Automated mammographic analysis

- Image texture measurements are predictive of breast cancer
- Can you pose “predict cancer” as a regression problem?
- What are the x and t?
- Given 1000 images with 1000 cancer scores, how would you build and evaluate a regression model?

Summary

- Linear models for regression with arbitrary basis functions
- Over-fitting: Model complexity vs. amount of training data.
- Generalization: Training and test data sets
- Regularization
- Probabilistic interpretation of regression
- Least squares and maximum likelihood solutions are equivalent under the Gaussian noise model.
- Maximum likelihood solutions for linear regression models with arbitrary basis functions

You should now...

- Be able to define different linear models for regression
- Be able to deduct and implement maximum likelihood solutions to regression problems phrased trough linear models
- Be able to recognize a regression problem in practical situations
- Know common pitfalls of regression and common techniques to avoid them (regularization)
- Understand the relationship between geometric (least squares) regression and maximum likelihood solutions to regression under a Gaussian noise model.
- This covered CB p 4-12, 28-30, 32-38, 137-142.

Next time

- Bayesian models for regression.
- You should read: CB 30-32, 142-147, 152-158, 172-173.