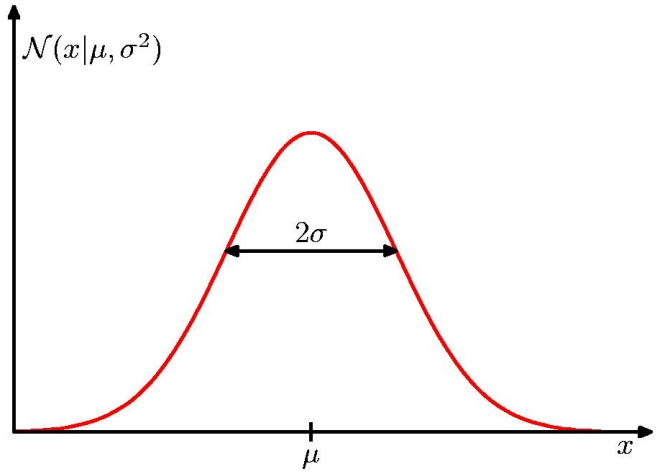
Introduction to Bayesian Methods - 3

Edoardo Milotti

Università di Trieste and INFN-Sezione di Trieste

The (multivariate) Gaussian distribution:

1-dimensional Gaussian distribution



Mean

Variance

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

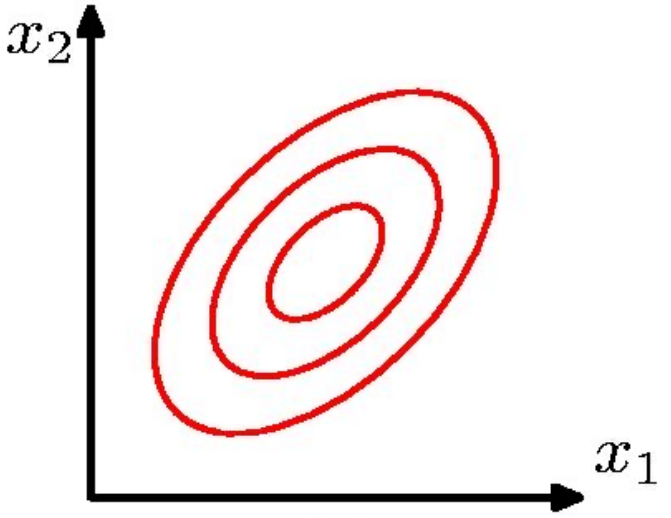
D-dimensional Gaussian distribution

Vector of means

Covariance matrix

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

Square root of the determinant of the covariance matrix



$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

$$\Delta^2 = (\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$$

Mahalanobis distance

$$\boldsymbol{\Sigma}\mathbf{u}_i = \lambda_i\mathbf{u}_i$$

Eingenvalue equation for the **covariance matrix**

$$\boldsymbol{\Sigma}^{-1} = \sum_{i=1}^{D}\frac{1}{\lambda_i}\mathbf{u}_i\mathbf{u}_i^{\mathrm{T}}$$

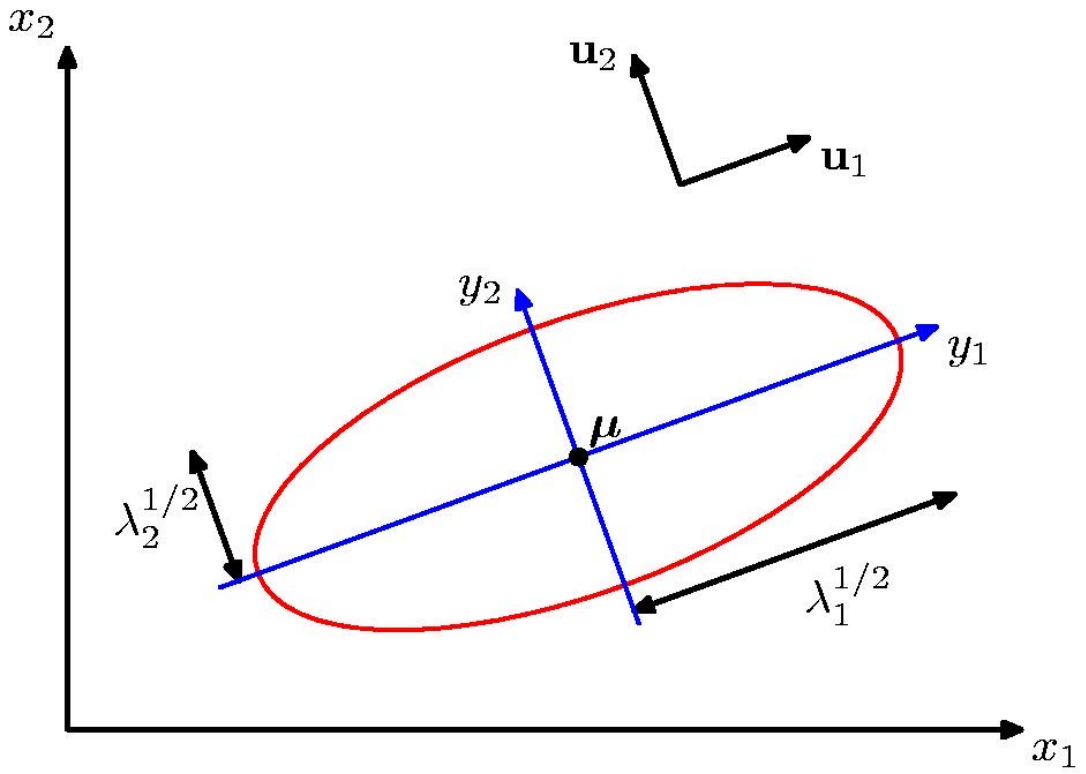
Diagonal form of the **precision matrix**

$$\Delta^2 = \sum_{i=1}^{D}\frac{y_i^2}{\lambda_i}$$

Mahalanobis distance in the rotated/translated reference system

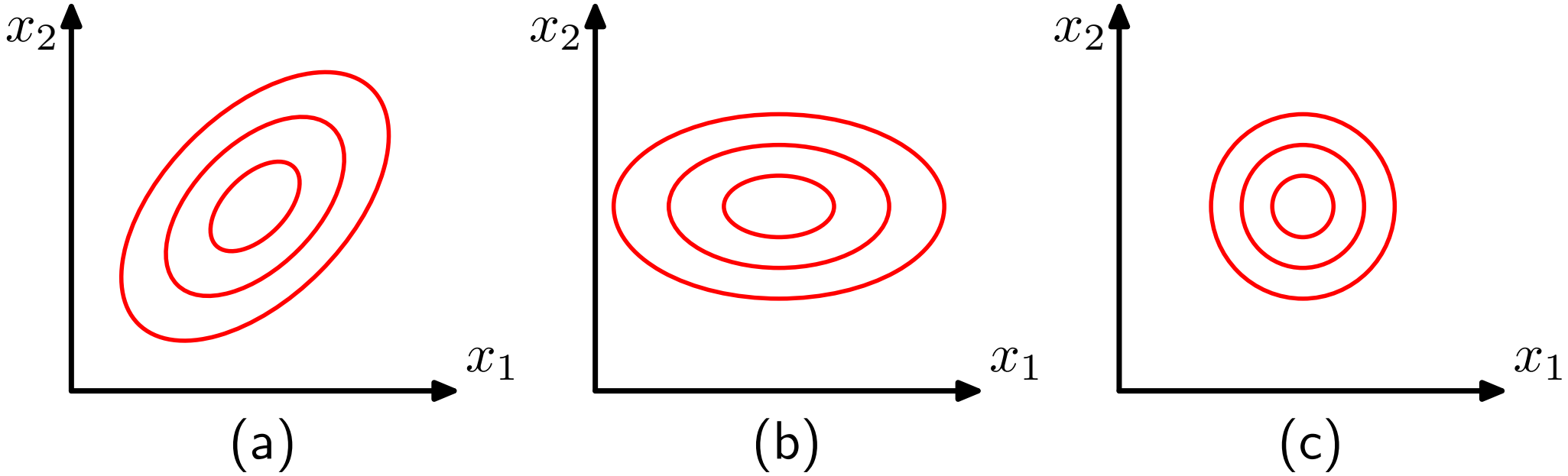
$$y_i = \mathbf{u}_i^{\mathrm{T}}(\mathbf{x}-\boldsymbol{\mu})$$

Rotated/translated coordinates



$$p(\mathbf{y}) = \prod_{j=1}^{D}\frac{1}{(2\pi\lambda)^{1/2}}\exp\left(-\frac{y_j^2}{2\lambda_j}\right)$$

pdf in the rotated/translated system



Contours of constant probability density for a Gaussian distribution in two dimensions in which the covariance matrix is (a) of general form, (b) diagonal, in which the elliptical contours are aligned with the coordinate axes, and (c) proportional to the identity matrix, in which the contours are concentric circles.

Conditional Gaussian distribution/Marginalization

An important property of the multivariate Gaussian distribution is that if two sets of variables are jointly Gaussian, then the conditional distribution of one set conditioned on the other is again Gaussian. Similarly, the marginal distribution of either set is also Gaussian.

Here, we assume that the a set contains p variables and the b set contains q variables, so that the partitioned matrix contains one p x p matrix and one p x q matrix in the first row, and one q x p matrix and a q x q matrix in the second row.

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix} \qquad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}$$

$$\boldsymbol{\Lambda} \equiv \boldsymbol{\Sigma}^{-1} \qquad \boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_{aa} & \boldsymbol{\Lambda}_{ab} \\ \boldsymbol{\Lambda}_{ba} & \boldsymbol{\Lambda}_{bb} \end{pmatrix}$$

Sometimes it is more convenient to work with the precision matrix. This is the partitioned form of the precision matrix.
Note that in general the off-diagonal terms are not square matrices.

Conditional Gaussian distribution/Marginalization – "Completing the square"

Now consider the following

$$p(\mathbf{x}_a|\mathbf{x}_b) \longrightarrow -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})$$

$$= -\frac{1}{2}(\mathbf{x}_a-\boldsymbol{\mu}_a)^T\Lambda_{aa}(\mathbf{x}_a-\boldsymbol{\mu}_a) - \frac{1}{2}(\mathbf{x}_a-\boldsymbol{\mu}_a)^T\Lambda_{ab}(\mathbf{x}_b-\boldsymbol{\mu}_b)$$

$$-\frac{1}{2}(\mathbf{x}_b-\boldsymbol{\mu}_b)^T\Lambda_{ba}(\mathbf{x}_a-\boldsymbol{\mu}_a) - \frac{1}{2}(\mathbf{x}_b-\boldsymbol{\mu}_b)^T\Lambda_{bb}(\mathbf{x}_b-\boldsymbol{\mu}_b)$$

Quadratic term in a after fixing the b part

Linear terms in a after fixing the b part

Constant term in a after fixing the b part

and

$$p(\mathbf{x}) \longrightarrow -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}) = \underbrace{-\frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x}}_{\text{Quadratic term}} + \underbrace{\mathbf{x}^T\Sigma^{-1}\boldsymbol{\mu}}_{\text{Linear term}} + \text{const}$$

Conditional Gaussian distribution/Marginalization – "Completing the square" – 2

$$
\begin{aligned}
p(\mathbf{x}_a|\mathbf{x}_b) \longrightarrow & -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}) \\
&= -\frac{1}{2}(\mathbf{x}_a-\boldsymbol{\mu}_a)^T\Lambda_{aa}(\mathbf{x}_a-\boldsymbol{\mu}_a) - \frac{1}{2}(\mathbf{x}_a-\boldsymbol{\mu}_a)^T\Lambda_{ab}(\mathbf{x}_b-\boldsymbol{\mu}_b) \\
&\quad -\frac{1}{2}(\mathbf{x}_b-\boldsymbol{\mu}_b)^T\Lambda_{ba}(\mathbf{x}_a-\boldsymbol{\mu}_a) - \frac{1}{2}(\mathbf{x}_b-\boldsymbol{\mu}_b)^T\Lambda_{bb}(\mathbf{x}_b-\boldsymbol{\mu}_b) \\
&= -\frac{1}{2}\mathbf{x}_a^T\Lambda_{aa}\mathbf{x}_a + \mathbf{x}_a^T\left[\Lambda_{aa}\boldsymbol{\mu}_a - \Lambda_{ab}(\mathbf{x}_b-\boldsymbol{\mu}_b)\right] + \text{ const} \qquad \textbf{(*)}
\end{aligned}
$$

$$
p(\mathbf{x}_a) \longrightarrow -\frac{1}{2}(\mathbf{x}_a-\boldsymbol{\mu}_{a|b})^T\Sigma_{a|b}^{-1}(\mathbf{x}_a-\boldsymbol{\mu}_{a|b}) = -\frac{1}{2}\mathbf{x}_a^T\Sigma_{a|b}^{-1}\mathbf{x}_a + \mathbf{x}_a^T\Sigma_{a|b}^{-1}\boldsymbol{\mu}_{a|b} + \text{const}
$$

Then, by comparing the expressions we find

$$
\Sigma_{a|b}^{-1} = \Lambda_{aa}
$$

$$
\Sigma_{a|b}^{-1}\boldsymbol{\mu}_{a|b} = \Lambda_{aa}\boldsymbol{\mu}_a - \Lambda_{ab}(\mathbf{x}_b-\boldsymbol{\mu}_b)
$$

$$
\Sigma_{a|b}^{-1} = \Lambda_{aa}
$$

$$
\boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a - \Lambda_{aa}^{-1}\Lambda_{ab}(\mathbf{x}_b-\boldsymbol{\mu}_b)
$$

Conditional Gaussian distribution/Marginalization – "Completing the square" – 3

General result on partitioned matrices (see also https://en.wikipedia.org/wiki/Schur_complement)

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{pmatrix}$$

$$\begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}^{-1} = \begin{pmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ba} & \Lambda_{bb} \end{pmatrix}$$

$$= \begin{pmatrix} (\Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba})^{-1} & -(\Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba})^{-1}\Sigma_{ab}\Sigma_{bb}^{-1} \\ -(\Sigma_{bb} - \Sigma_{ba}\Sigma_{aa}^{-1}\Sigma_{ab})^{-1}\Sigma_{ba}\Sigma_{aa}^{-1} & (\Sigma_{bb} - \Sigma_{ba}\Sigma_{aa}^{-1}\Sigma_{ab})^{-1} \end{pmatrix}$$

$$\Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}$$

$$\boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a + \Sigma_{ab}\Sigma_{bb}^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b)$$

Conditional Gaussian distribution/Marginalization – Marginalization with respect to x_b

$$p(\mathbf{x}_a) = \int p(\mathbf{x}_a, \mathbf{x}_b)\, \mathrm{d}\mathbf{x}_b$$

Marginalized distribution

We use the previous results and pick terms quadratic and linear in $\mathbf{x}_b$ (see *)

$$-\frac{1}{2}\mathbf{x}_b^{\mathrm{T}}\boldsymbol{\Lambda}_{bb}\mathbf{x}_b + \mathbf{x}_b^{T}\mathbf{m} = -\frac{1}{2}(\mathbf{x}_b - \boldsymbol{\Lambda}_{bb}^{-1}\mathbf{m})^{\mathrm{T}}\boldsymbol{\Lambda}_{bb}(\mathbf{x}_b - \boldsymbol{\Lambda}_{bb}^{-1}\mathbf{m}) + \frac{1}{2}\mathbf{m}^{\mathrm{T}}\boldsymbol{\Lambda}_{bb}^{-1}\mathbf{m}$$

where

$$\mathbf{m} = \boldsymbol{\Lambda}_{bb}\boldsymbol{\mu}_b - \boldsymbol{\Lambda}_{ba}(\mathbf{x}_a - \boldsymbol{\mu}_a)$$

Then, integrating we obtain a standard normalization factor (which does not depend on $\mathbf{x}_a$) times a normal distribution with exponent

$$\frac{1}{2}\left[\boldsymbol{\Lambda}_{bb}\boldsymbol{\mu}_b - \boldsymbol{\Lambda}_{ba}(\mathbf{x}_a - \boldsymbol{\mu}_a)\right]^{\mathrm{T}}\boldsymbol{\Lambda}_{bb}^{-1}\left[\boldsymbol{\Lambda}_{bb}\boldsymbol{\mu}_b - \boldsymbol{\Lambda}_{ba}(\mathbf{x}_a - \boldsymbol{\mu}_a)\right]$$

a-dependent exponent after integrating over the b part (the constant in the b part)

$$-\frac{1}{2}\mathbf{x}_a^{\mathrm{T}}\boldsymbol{\Lambda}_{aa}\mathbf{x}_a + \mathbf{x}_a^{\mathrm{T}}(\boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a + \boldsymbol{\Lambda}_{ab}\boldsymbol{\mu}_b) + \mathrm{const}$$

previous quadratic and linear terms

Conditional Gaussian distribution/Marginalization – Marginalization with respect to x_b – 2

Expanding and simplifying we find

$$\frac{1}{2}[\Lambda_{bb}\boldsymbol{\mu}_b - \Lambda_{ba}(\mathbf{x}_a - \boldsymbol{\mu}_a)]^T\Lambda_{bb}^{-1}[\Lambda_{bb}\boldsymbol{\mu}_b - \Lambda_{ba}(\mathbf{x}_a - \boldsymbol{\mu}_a)] - \frac{1}{2}\mathbf{x}_a^T(\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba})\mathbf{x}_a + \text{const}$$
$$= -\frac{1}{2}\mathbf{x}_a^T(\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba})\mathbf{x}_a + \mathbf{x}_a^T(\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba})\boldsymbol{\mu}_a + \text{const}$$

and by comparison with

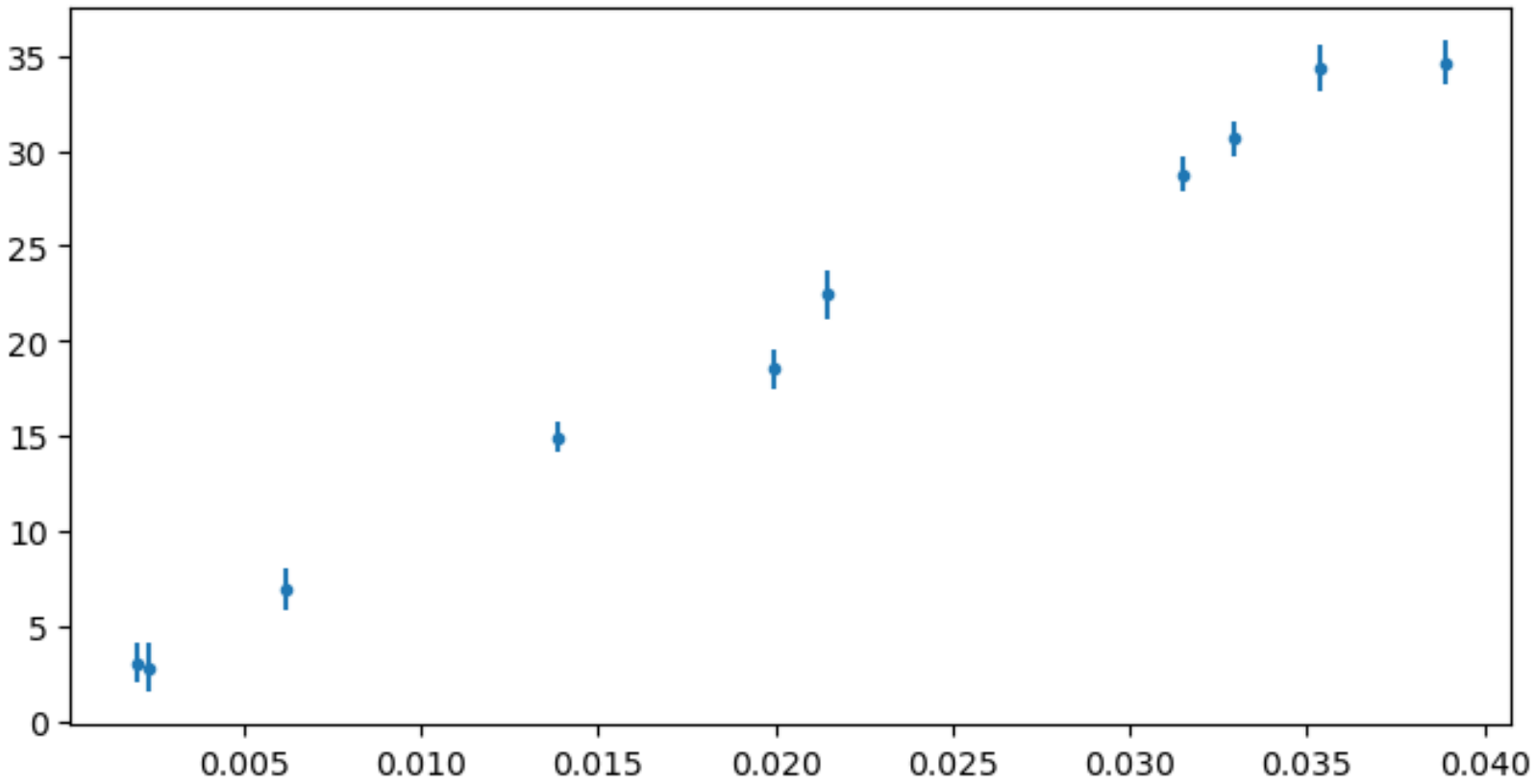
$$p(\mathbf{x}) \longrightarrow -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}) = -\frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x} + \mathbf{x}^T\Sigma^{-1}\boldsymbol{\mu} + \text{ const}$$

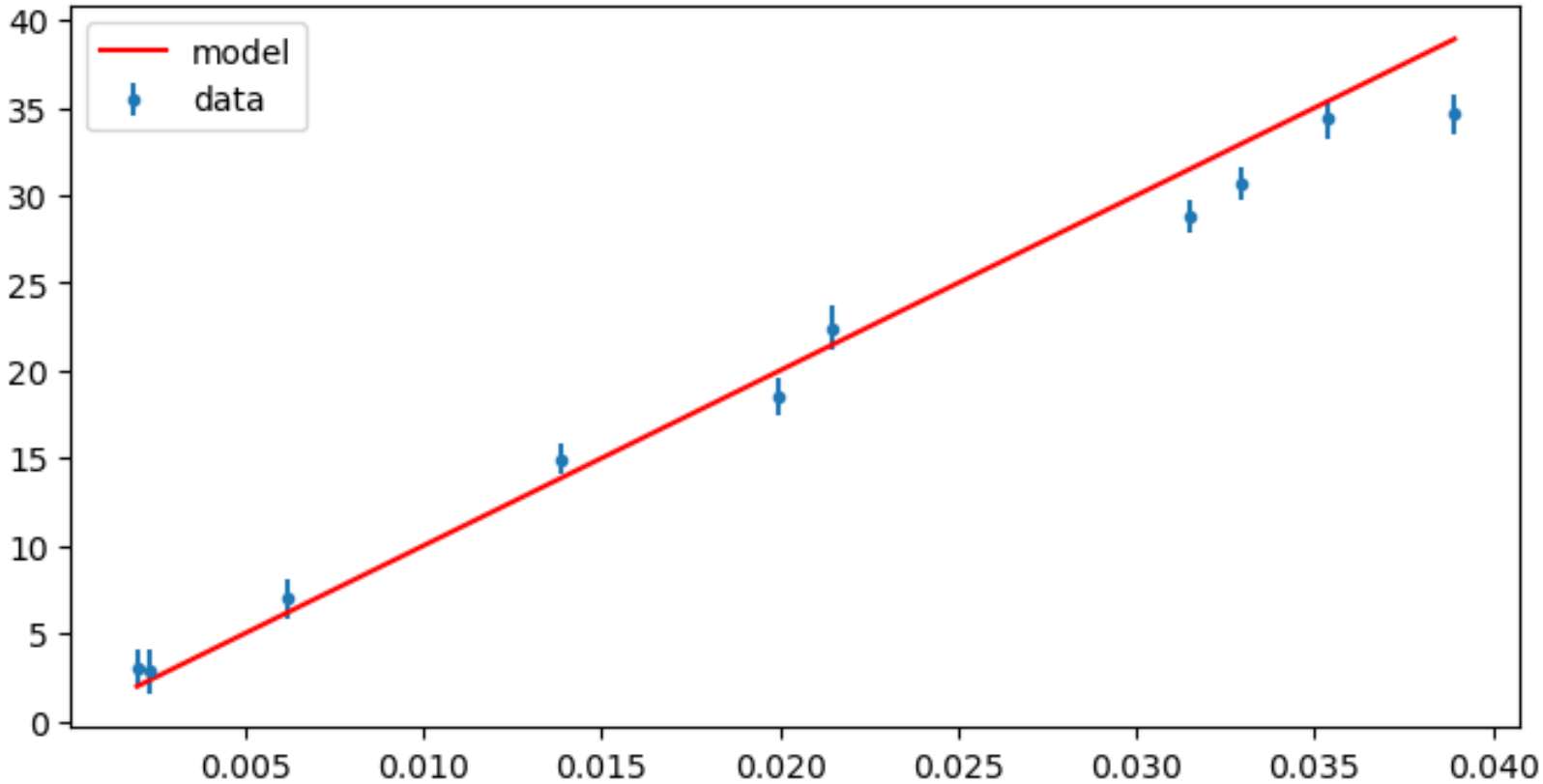
we find (the mean is unchanged by the marginalization integral)

$$\Sigma_a^{-1} = \Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba}$$
$$\boldsymbol{\mu}_a = \boldsymbol{\mu}_a$$

Table 1.1: A *mistery dataset*.

x	y	σ
0.002	3.009	1.040
0.002	2.811	1.310
0.006	6.958	1.107
0.014	14.938	0.842
0.020	18.505	1.079
0.021	22.443	1.317
0.031	28.772	0.928
0.033	30.641	0.923
0.035	34.363	1.172
0.039	34.632	1.172


35
30
25
20
15
10
5
0
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040


model
data
40
35
30
25
20
15
10
5
0
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040

Higher polynomial orders

$$y(x, \mathbf{c}) = \sum_{k=0,n} c_k x^k$$

order *n* polynomial function

Higher polynomial orders

$$y(x, \mathbf{c}) = \sum_{k=0,M} c_k x^k$$

order *M* polynomial function (M+1 coefficients)

$$S(\mathbf{x}, \mathbf{y}, \mathbf{c}) = \frac{1}{2} \sum_{i=1}^{N} \frac{[y_i - y(x_i, \mathbf{c})]^2}{\sigma_i^2}$$

error function (or ***cost function***, or ***loss function***)

Higher polynomial orders

$$y(x, \mathbf{c}) = \sum_{k=0,n} c_k x^k$$

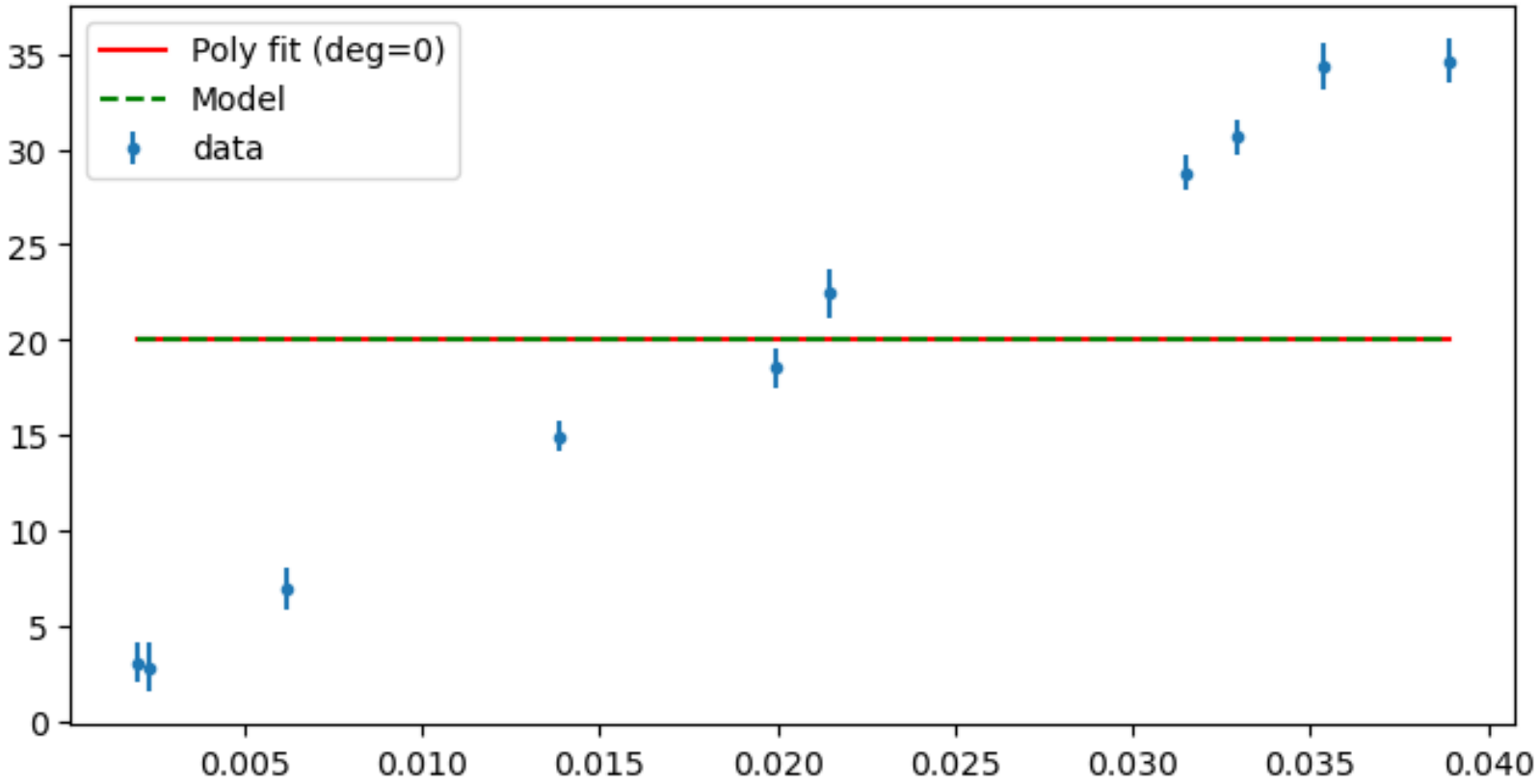
order *n* polynomial function

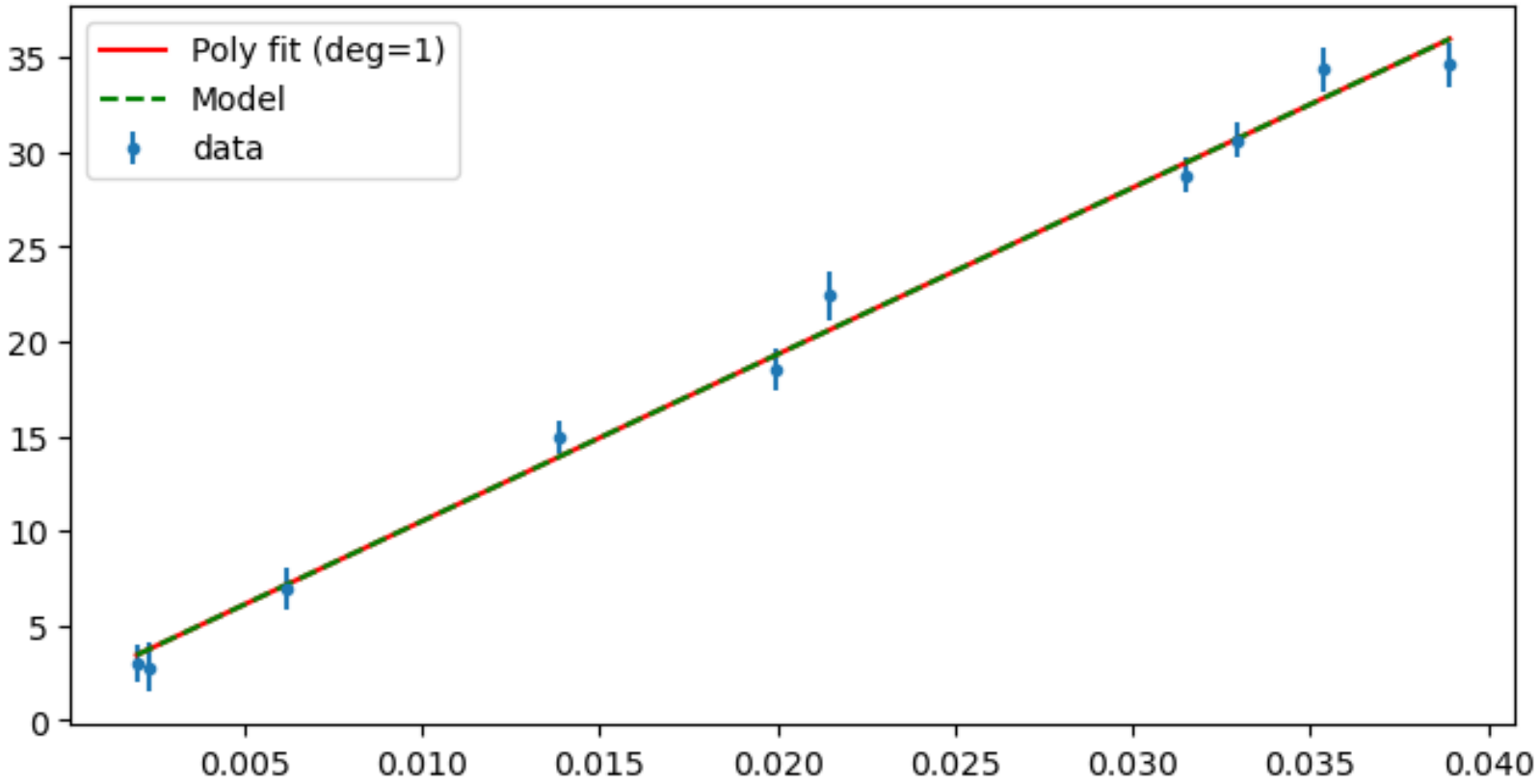
$$S(\mathbf{x}, \mathbf{y}, \mathbf{c}) = \frac{1}{2} \sum_{i=1}^{N} \frac{[y_i - y(x_i, \mathbf{c})]^2}{\sigma_i^2}$$

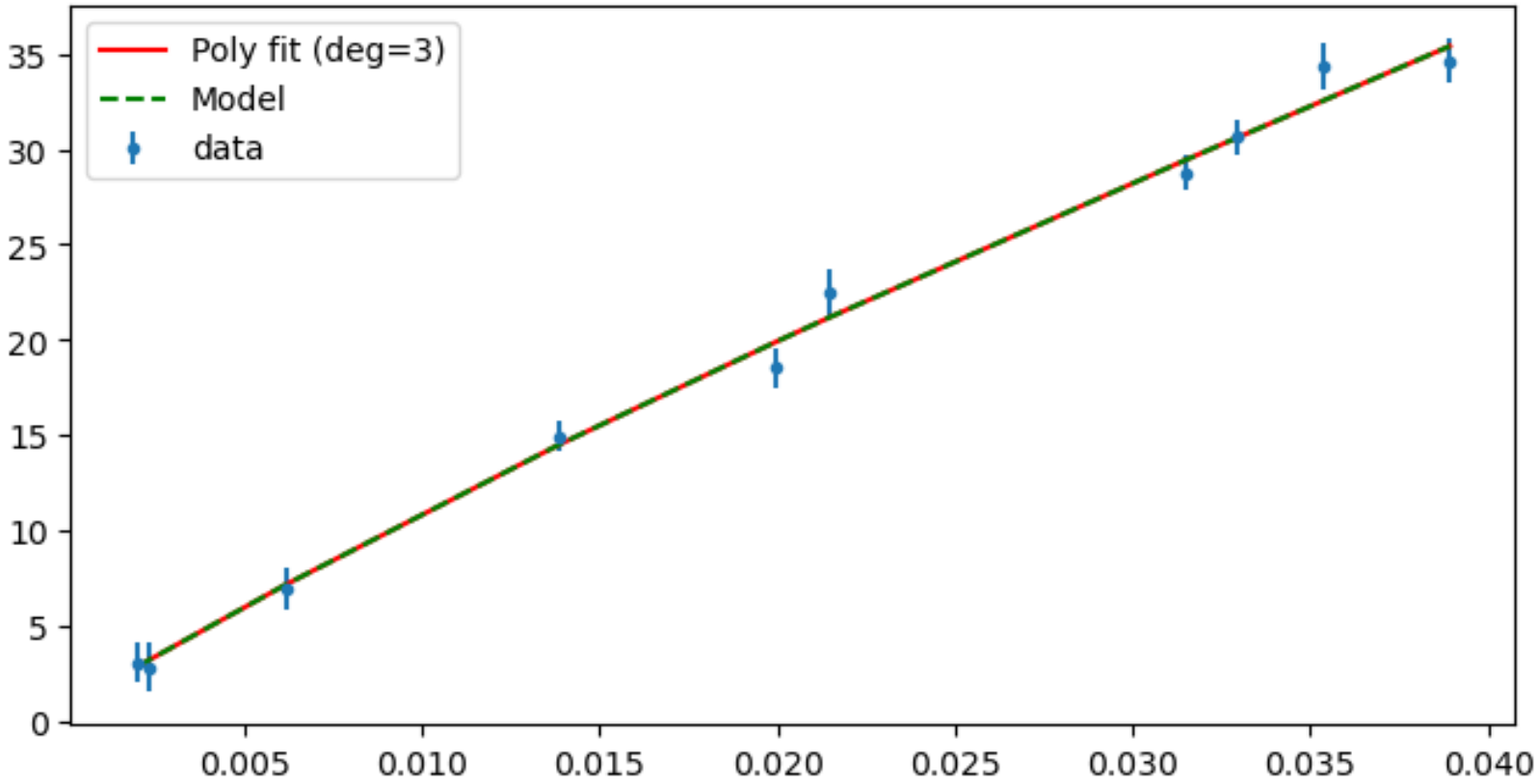
error function (or ***cost function***, or ***loss function***)

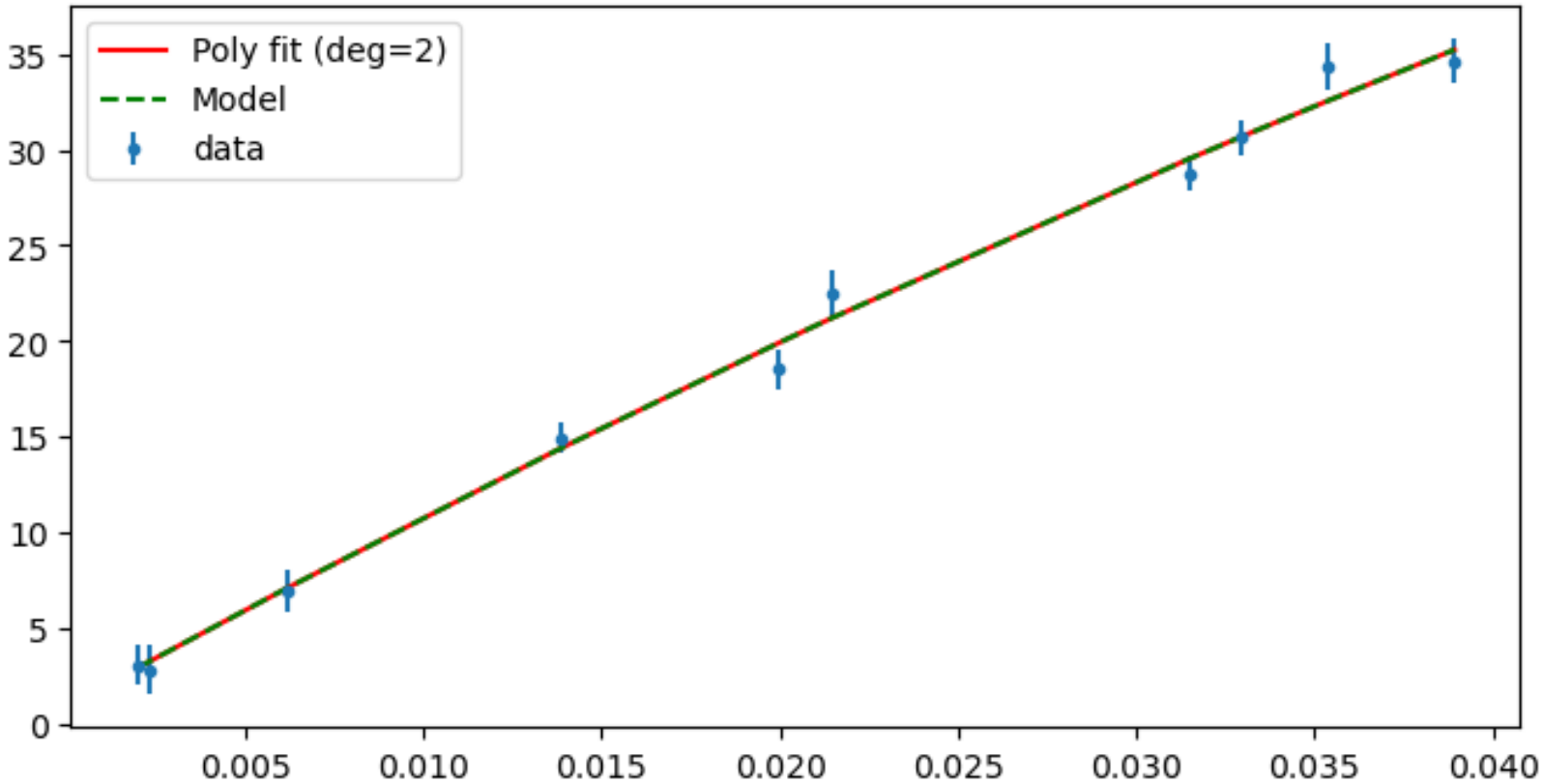
$$\frac{\partial S}{\partial c_\ell} = -\sum_{i=1}^{N} x^\ell \frac{[y_i - y(x_i, \mathbf{c})]}{\sigma_i^2} = 0$$

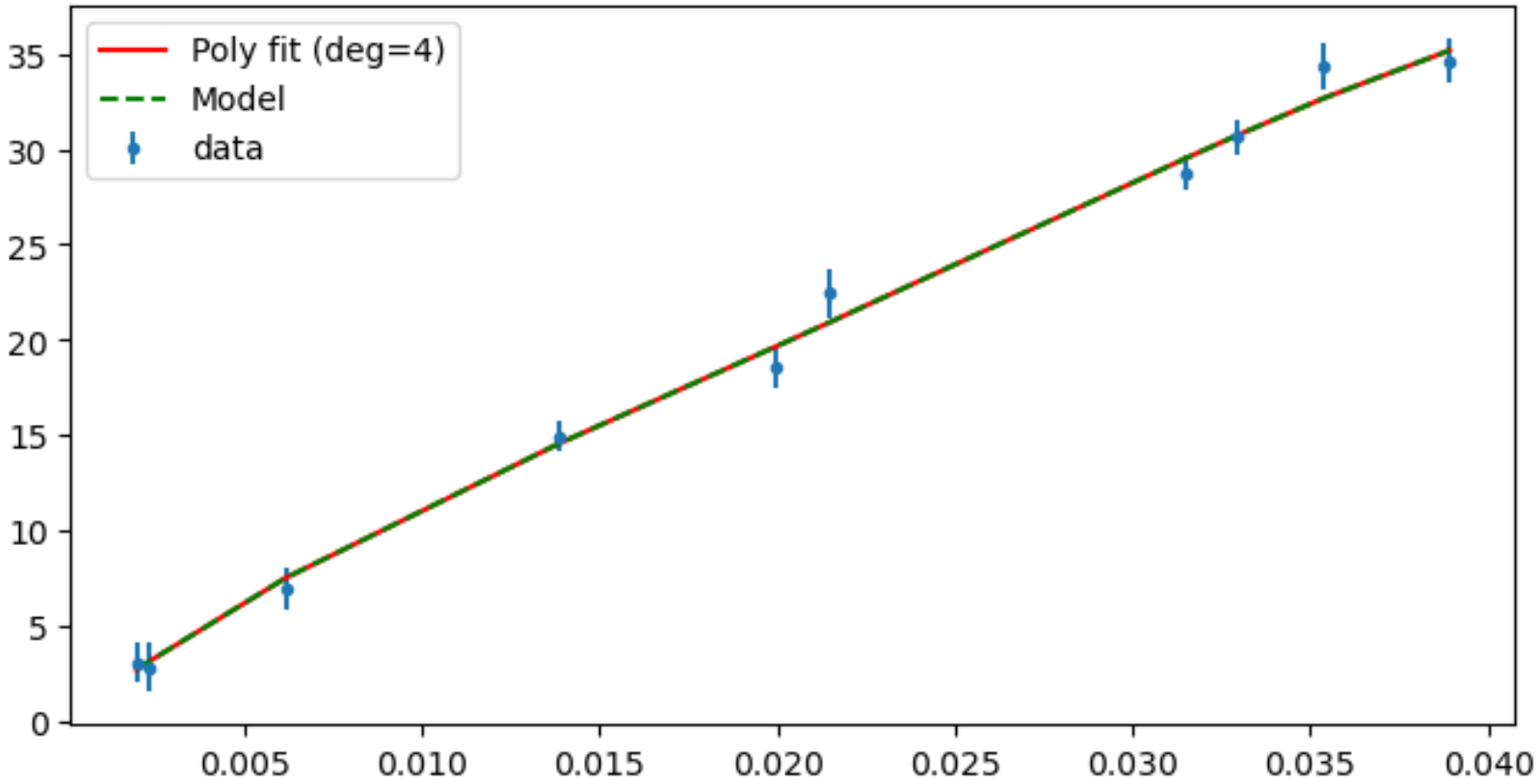
error function minimization;
n+1 linear equations solved with standard linear algebra methods

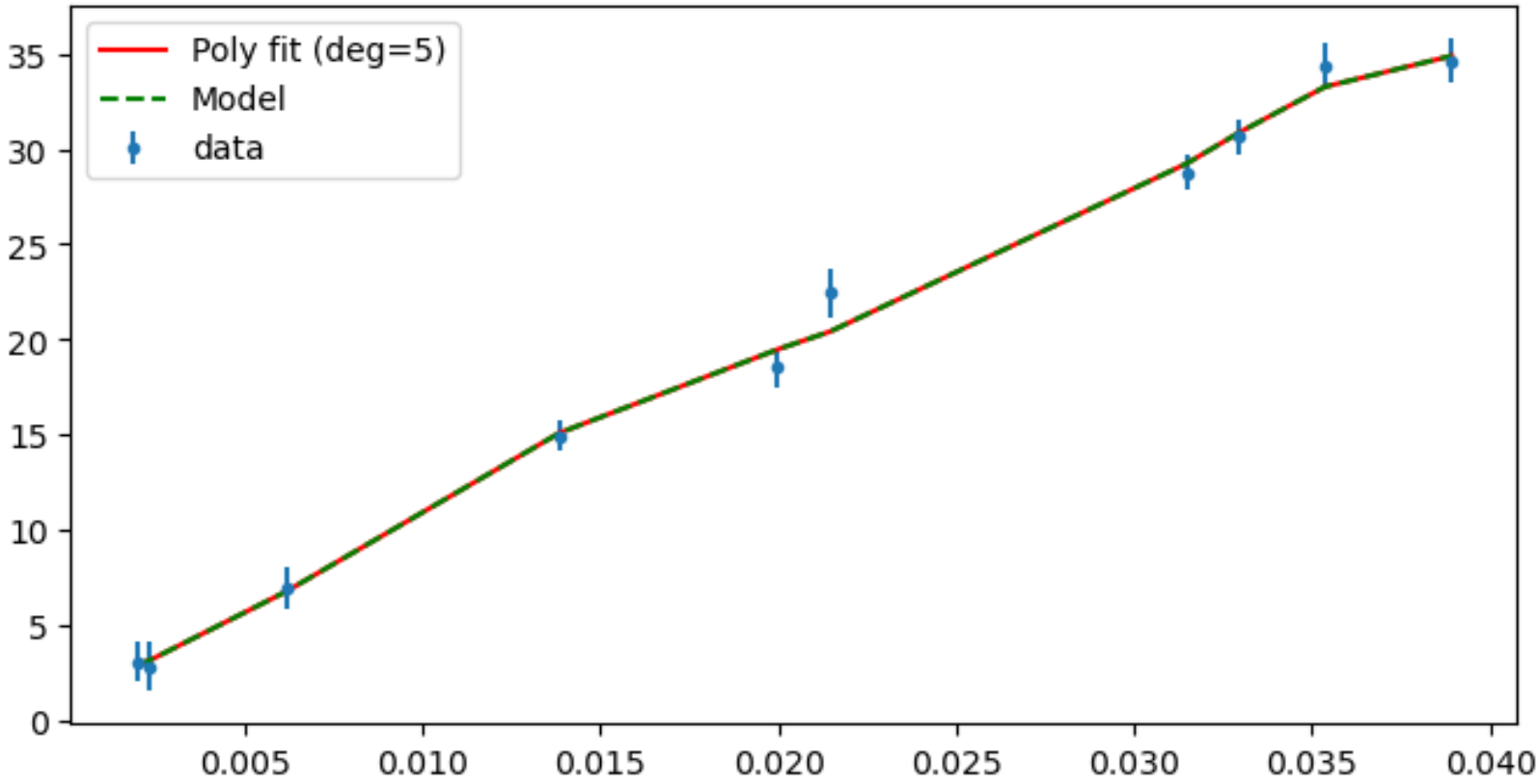

Poly fit (deg=0)
Model
data
35
30
25
20
15
10
5
0
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040

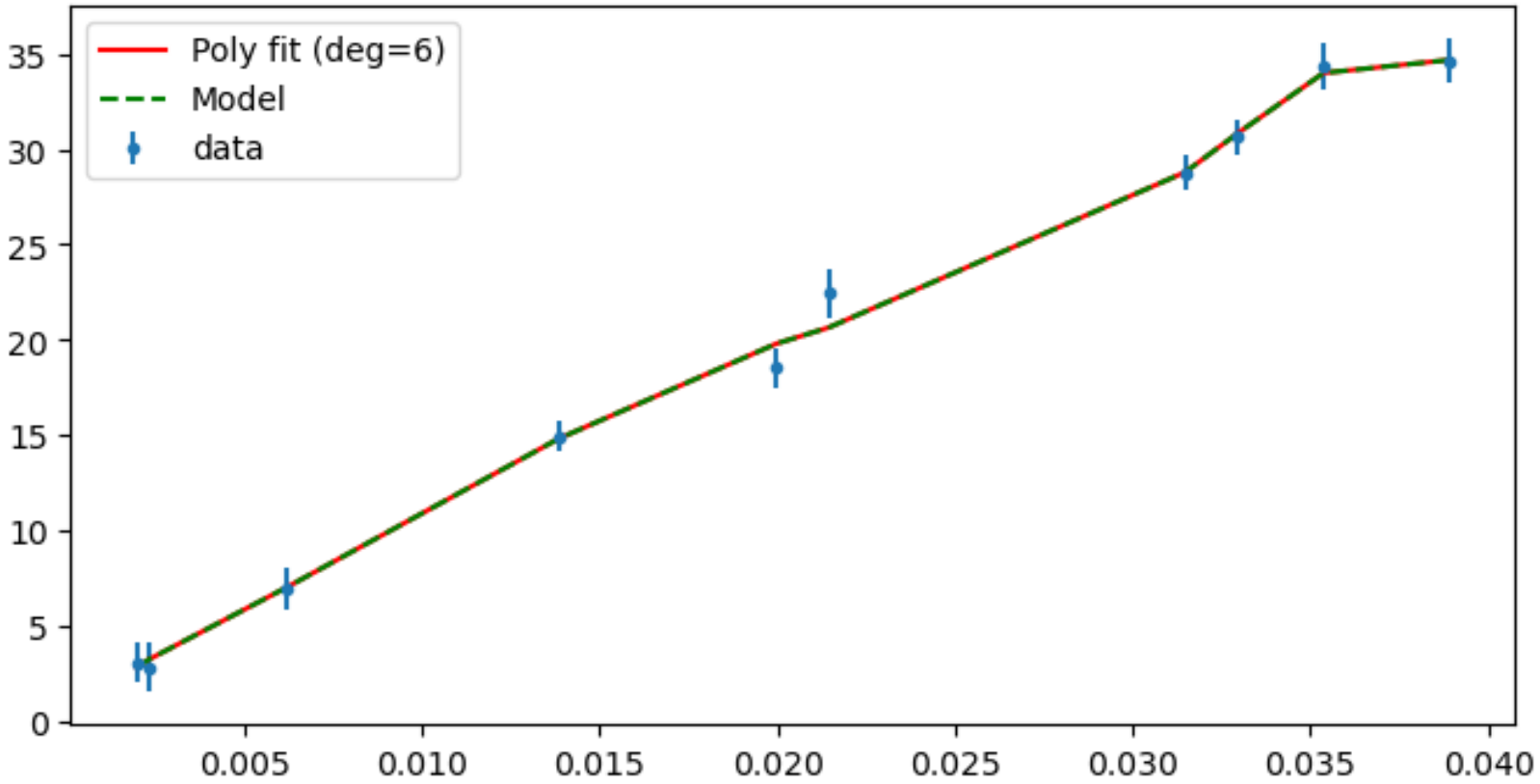

Poly fit (deg=1)
Model
data
35
30
25
20
15
10
5
0
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040

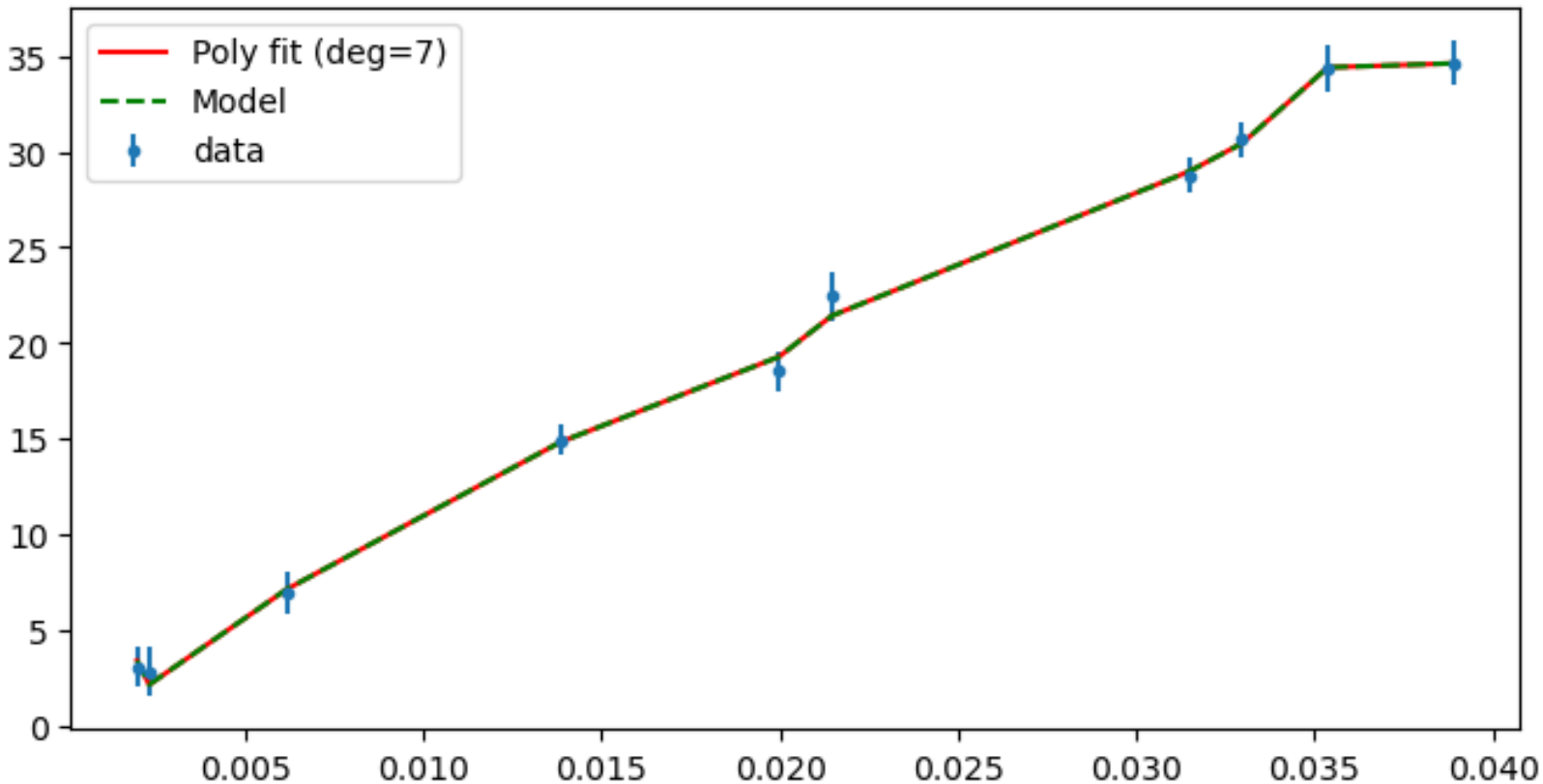

Poly fit (deg=3)
Model
data
35
30
25
20
15
10
5
0
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040

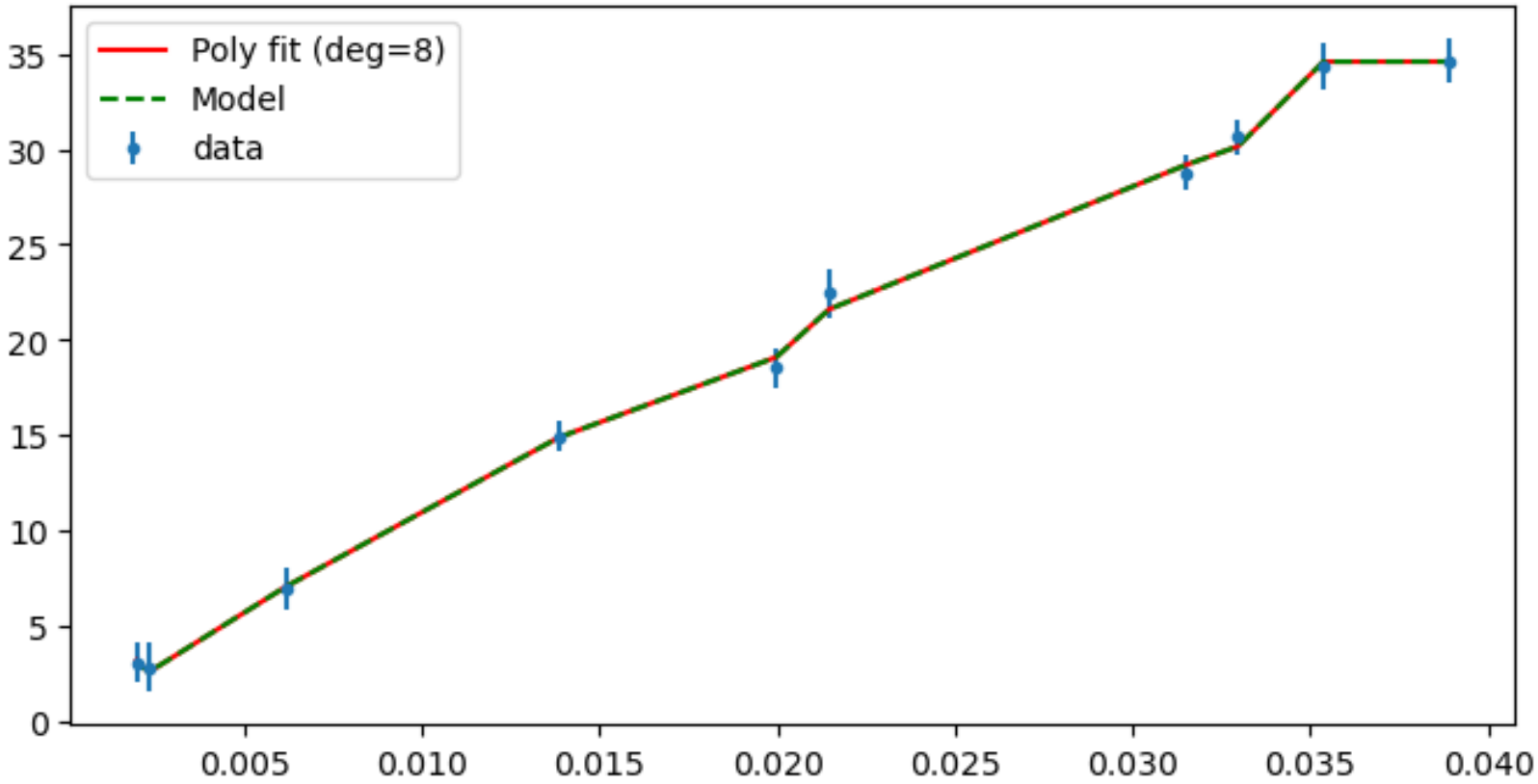

Poly fit (deg=2)
Model
data
35
30
25
20
15
10
5
0
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040

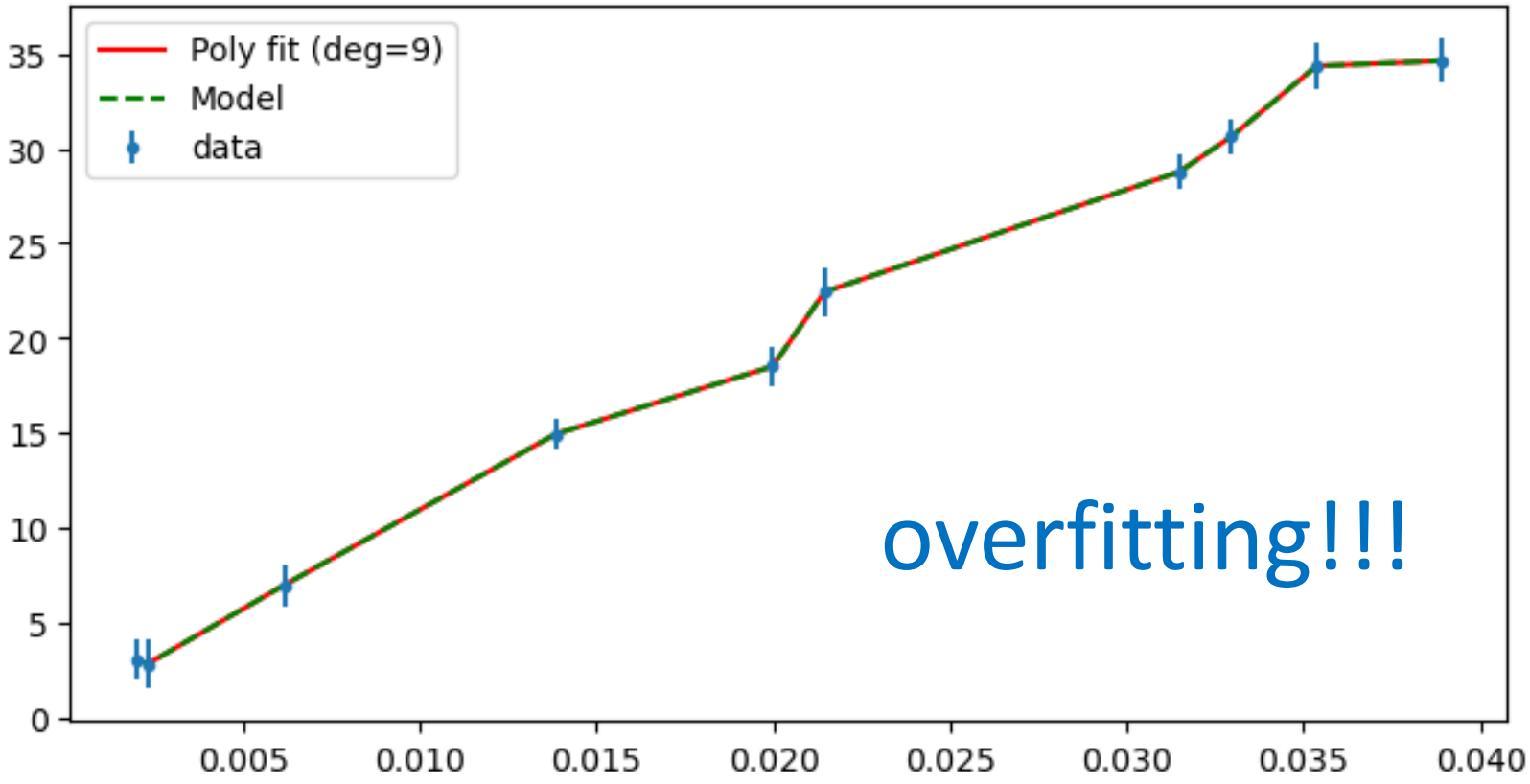

Poly fit (deg=4)
Model
data
35
30
25
20
15
10
5
0
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040


Poly fit (deg=5)
Model
data
35
30
25
20
15
10
5
0
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040


Poly fit (deg=6)
Model
data
35
30
25
20
15
10
5
0
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040


Poly fit (deg=7)
Model
data
35
30
25
20
15
10
5
0
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040


Poly fit (deg=8)
Model
data
35
30
25
20
15
10
5
0
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040


Poly fit (deg=9)
Model
data
35
30
25
20
15
10
5
0
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040
overfitting!!!

How do we select the "best" model? How do we avoid overfitting ???

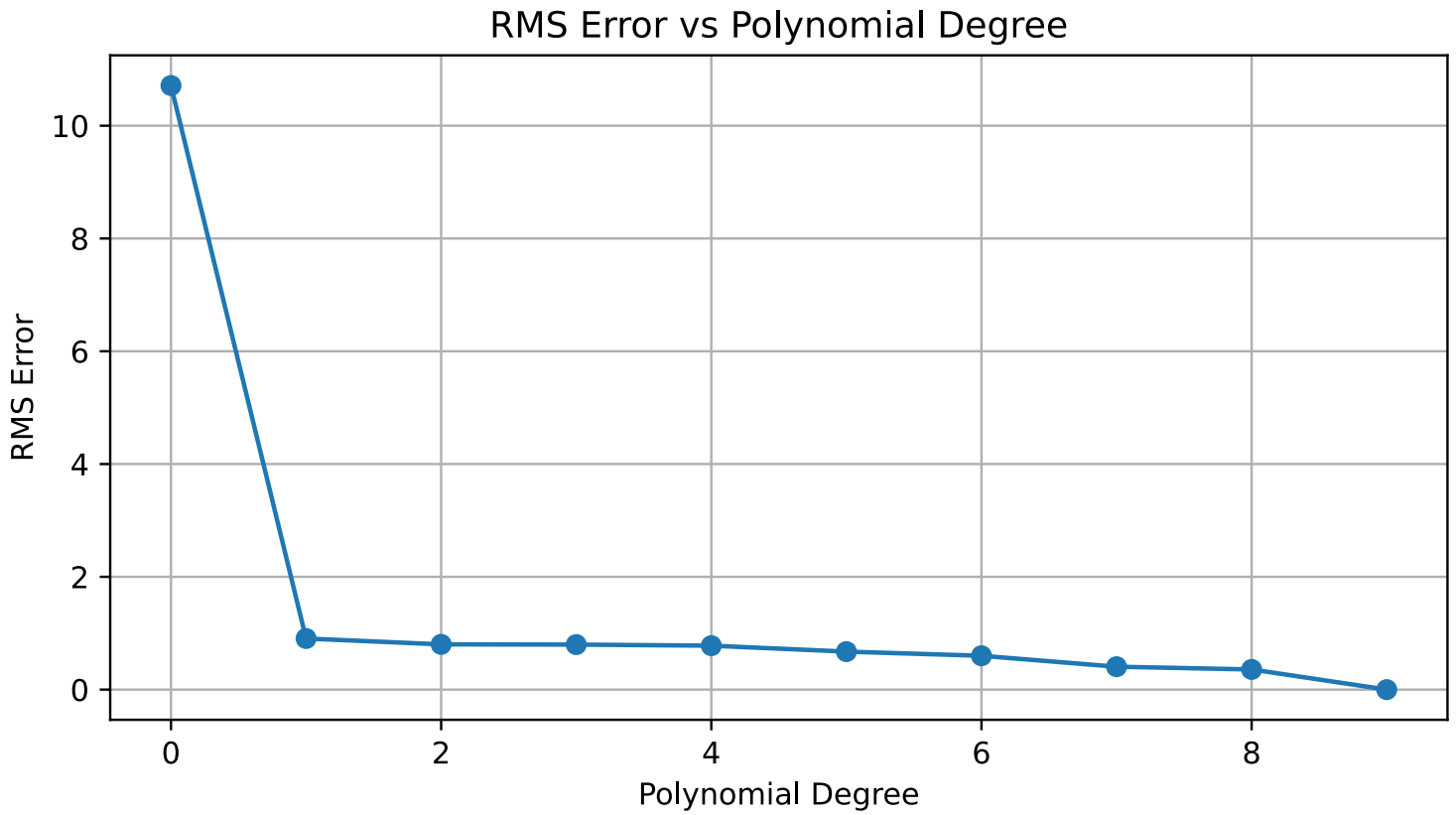
The RMS error function

$$E_{\mathrm{RMS}} = \sqrt{S(\mathbf{x}, \mathbf{y}, \mathbf{c}^*)}$$

How do we select the "best" model? How do we avoid overfitting ???

The RMS error function

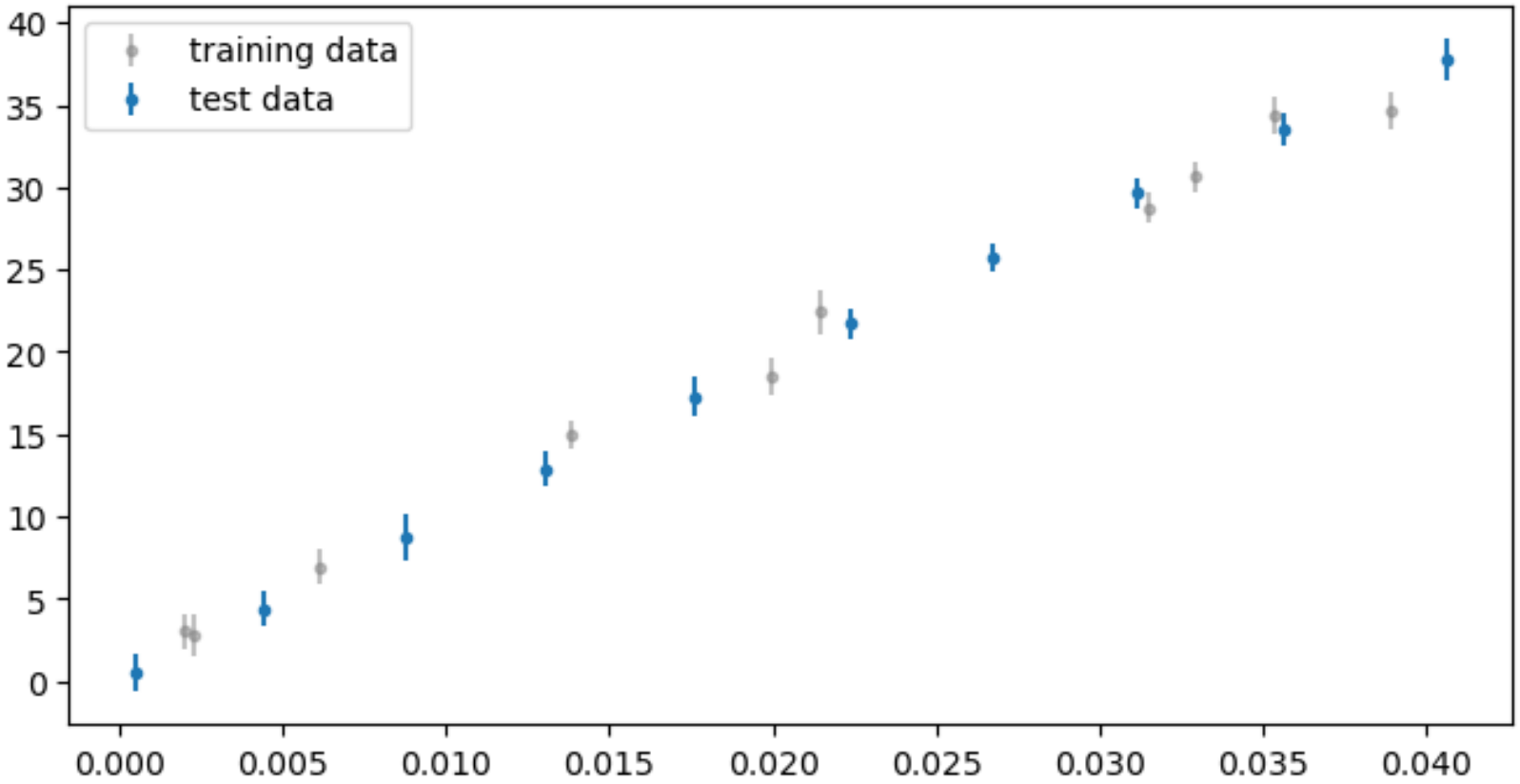
$$E_{\mathrm{RMS}} = \sqrt{S(\mathbf{x}, \mathbf{y}, \mathbf{c}^*)}$$



The "predictive" aspects of model fitting

We can cast the model fitting procedure in the context of modern Machine Learning

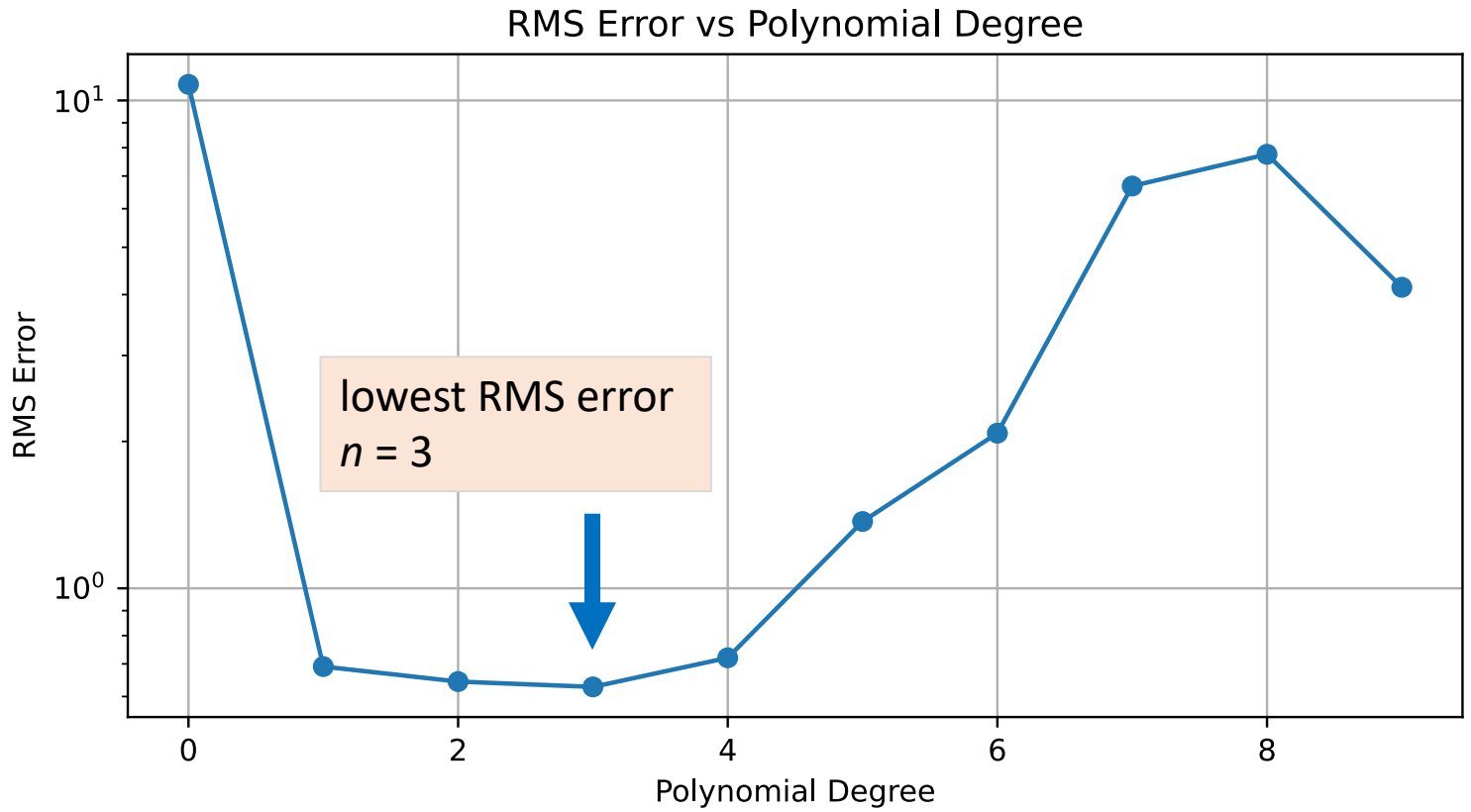
- the data we used to find the model coefficients are the "training data" (DONE)
- to validate the model, we use a "test set" of new data and the same RMS error



The "predictive" aspects of model fitting

We can cast the model fitting procedure in the context of modern Machine Learning

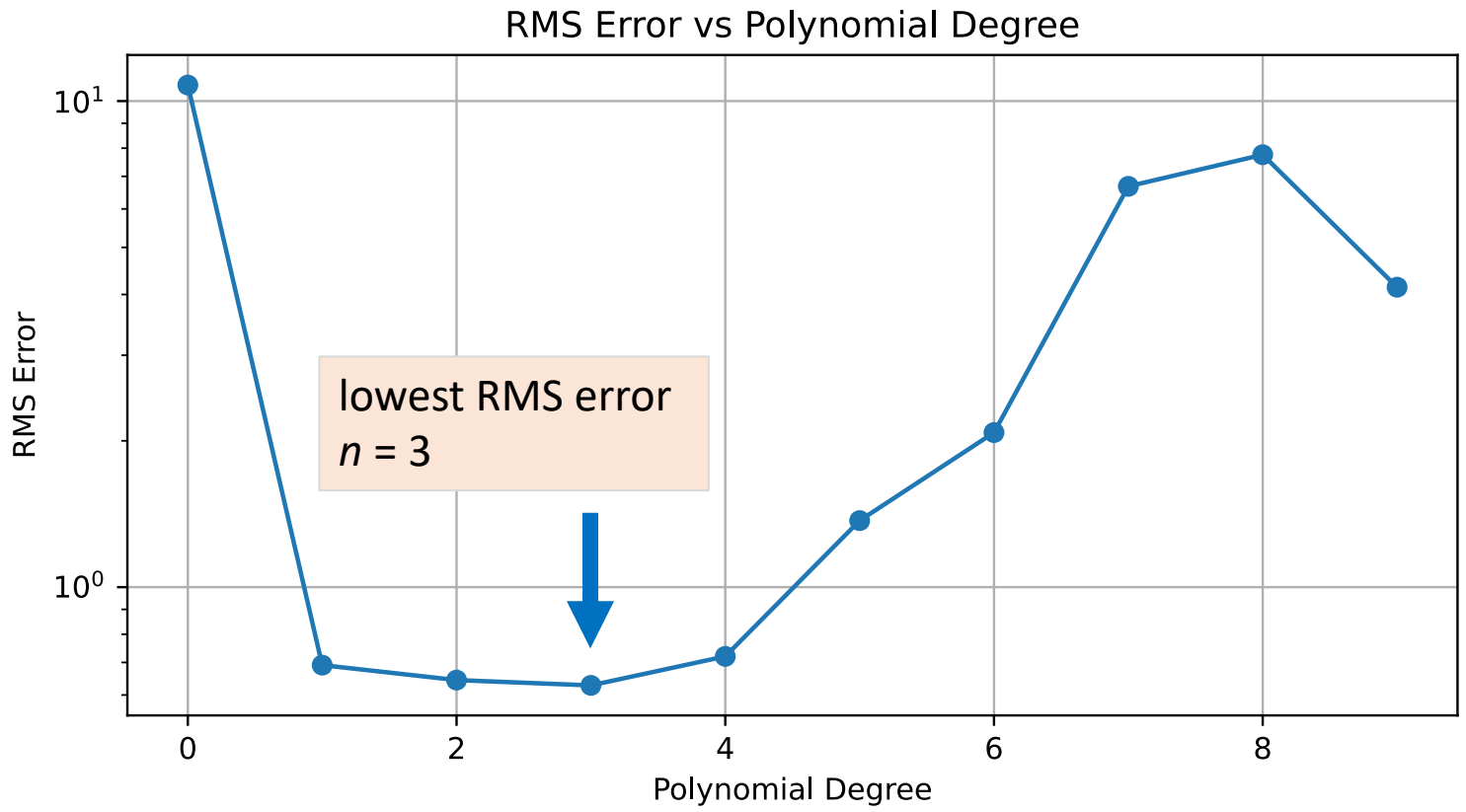
- the data we used to find the model coefficients are the "training data" (DONE)
- to validate the model, we use a "test set" of new data and the same RMS error



The "predictive" aspects of model fitting

We can cast the model fitting procedure in the context of modern Machine Learning

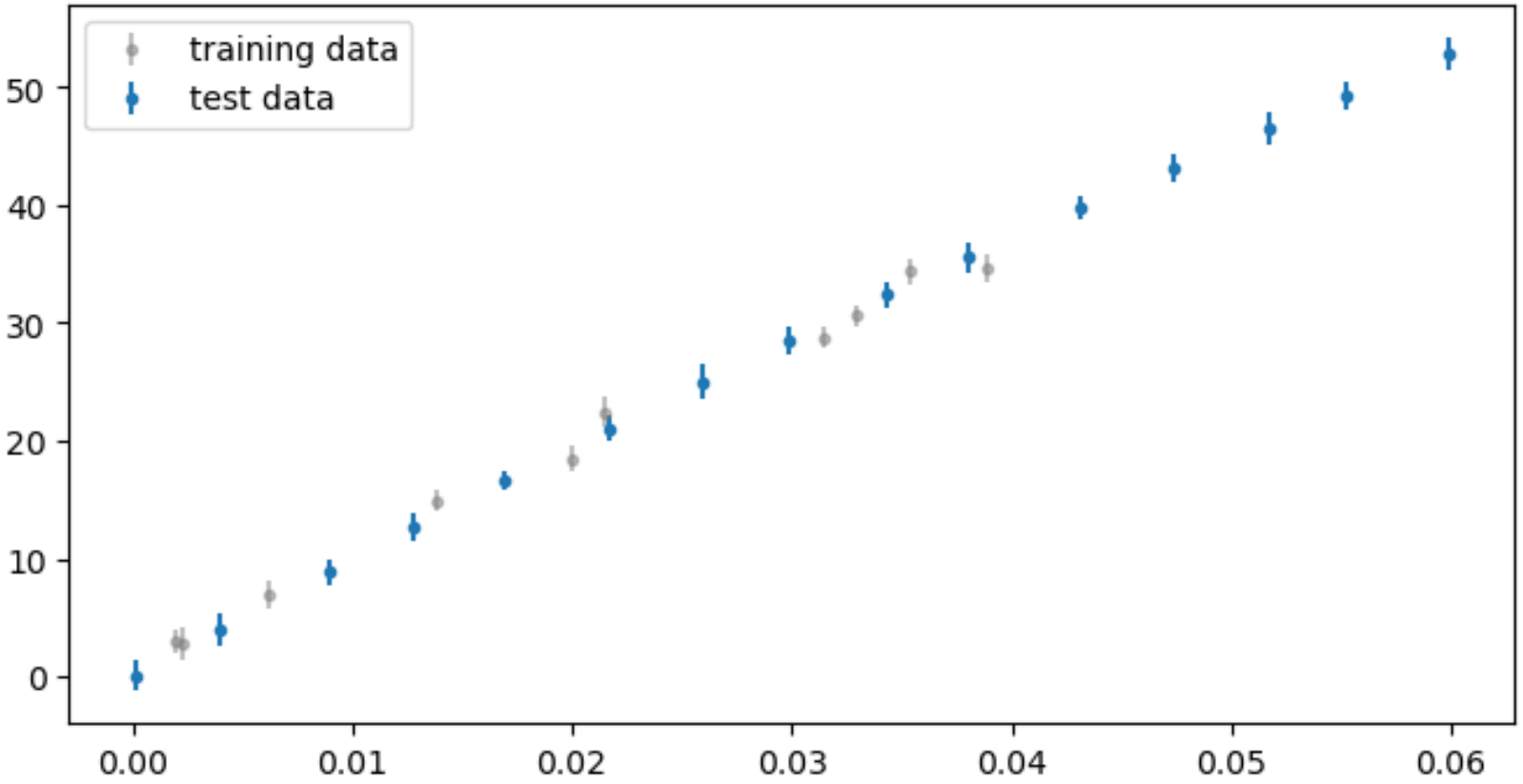
In this test, the polynomial *n* = 3 is the one that best generalizes (interpolates) the training data



The "predictive" aspects of model fitting (extrapolation)

We can cast the model fitting procedure in the context of modern Machine Learning

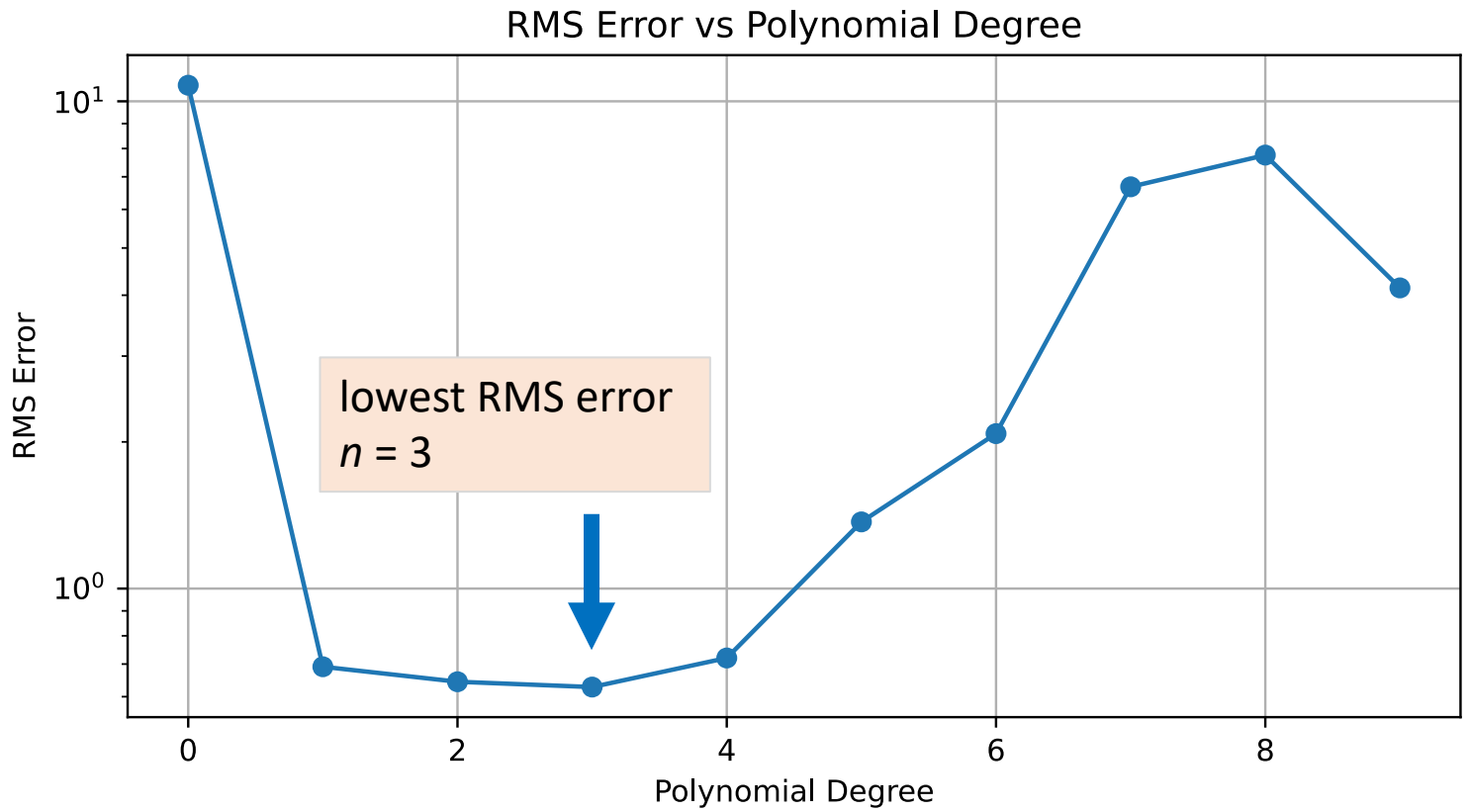
- the data we used to find the model coefficients are the "training data" (DONE)
- to validate the model, we use a "test set" of new data and the same RMS error



The "predictive" aspects of model fitting (extrapolation)

We can cast the model fitting procedure in the context of modern Machine Learning

In this test, the polynomial *n* = 3 is again the best one at generalizing (extrapolating) the training data, but ...



An interesting feature of polynomial fits

Table 1.2: Coefficients of polynomial fits.

degree	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
0	20									
1	1.72	880								
2	0.876	1.03e+03	-3.68e+03							
3	0.684	1.09e+03	-8.16e+03	7.61e+04						
4	-0.0337	1.45e+03	-4.58e+04	1.45e+06	-1.63e+07					
5	2.61	-247	2.34e+05	-1.67e+07	4.82e+08	-4.88e+09				
6	-0.362	2.04e+03	-2.83e+05	3.3e+07	-1.81e+09	4.53e+10	-4.18e+11			
7	25.3	-1.99e+04	5.6e+06	-6.75e+08	4.21e+10	-1.41e+12	2.42e+13	-1.66e+14		
8	13.2	-8.59e+03	2.01e+06	-1.22e+08	-4.9e+09	8.96e+11	-4.09e+13	8.13e+14	-6.06e+15	
9	-7.83	1.67e+04	-9.21e+06	2.3e+09	-2.91e+11	2.06e+13	-8.53e+14	2.06e+16	-2.68e+17	1.46e+18

Higher order fits have larger and larger coefficients. Can we select "simpler" models putting a penalty on larger coefficients?

An interesting feature of polynomial fits

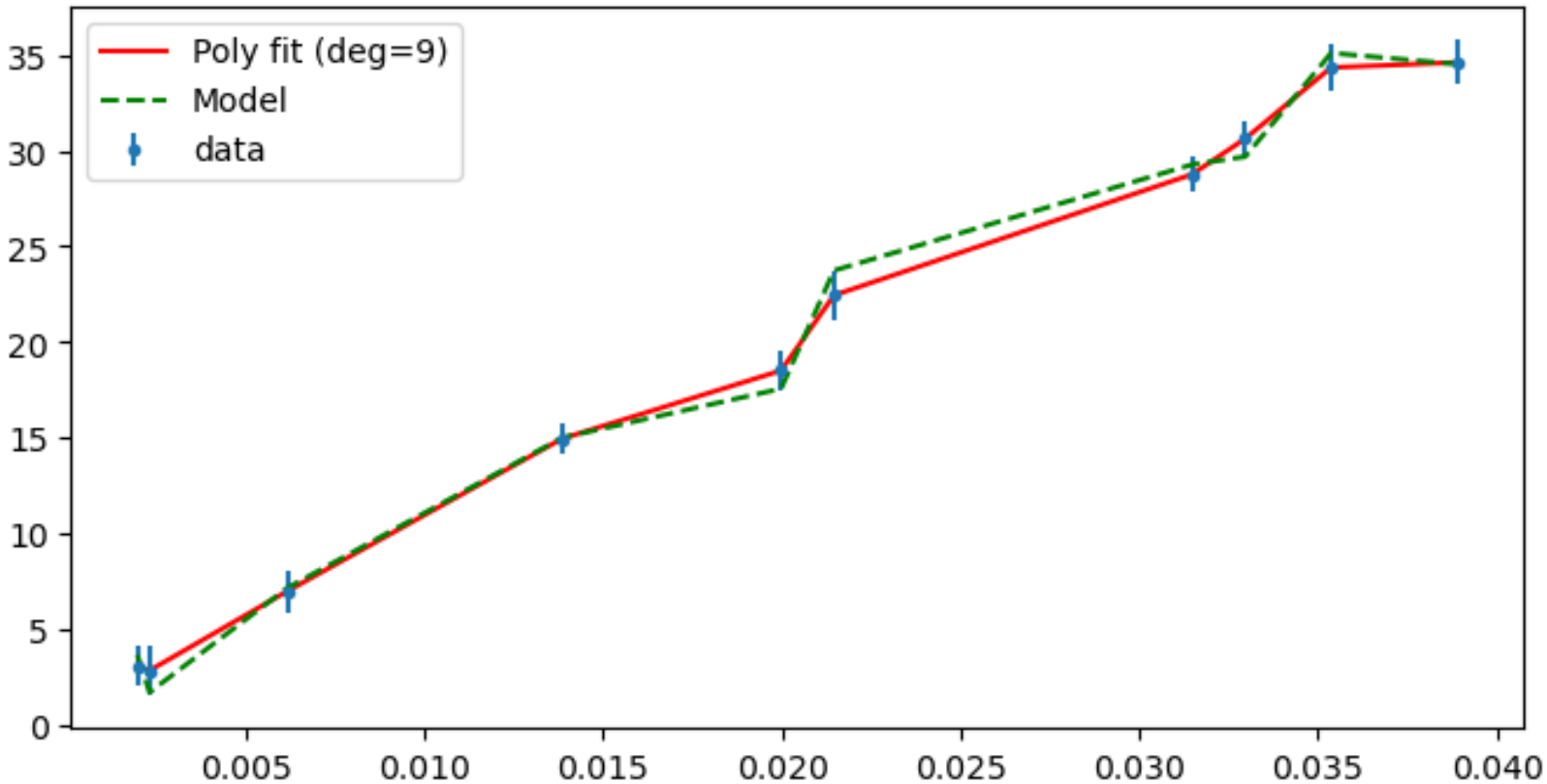
Table 1.2: Coefficients of polynomial fits.

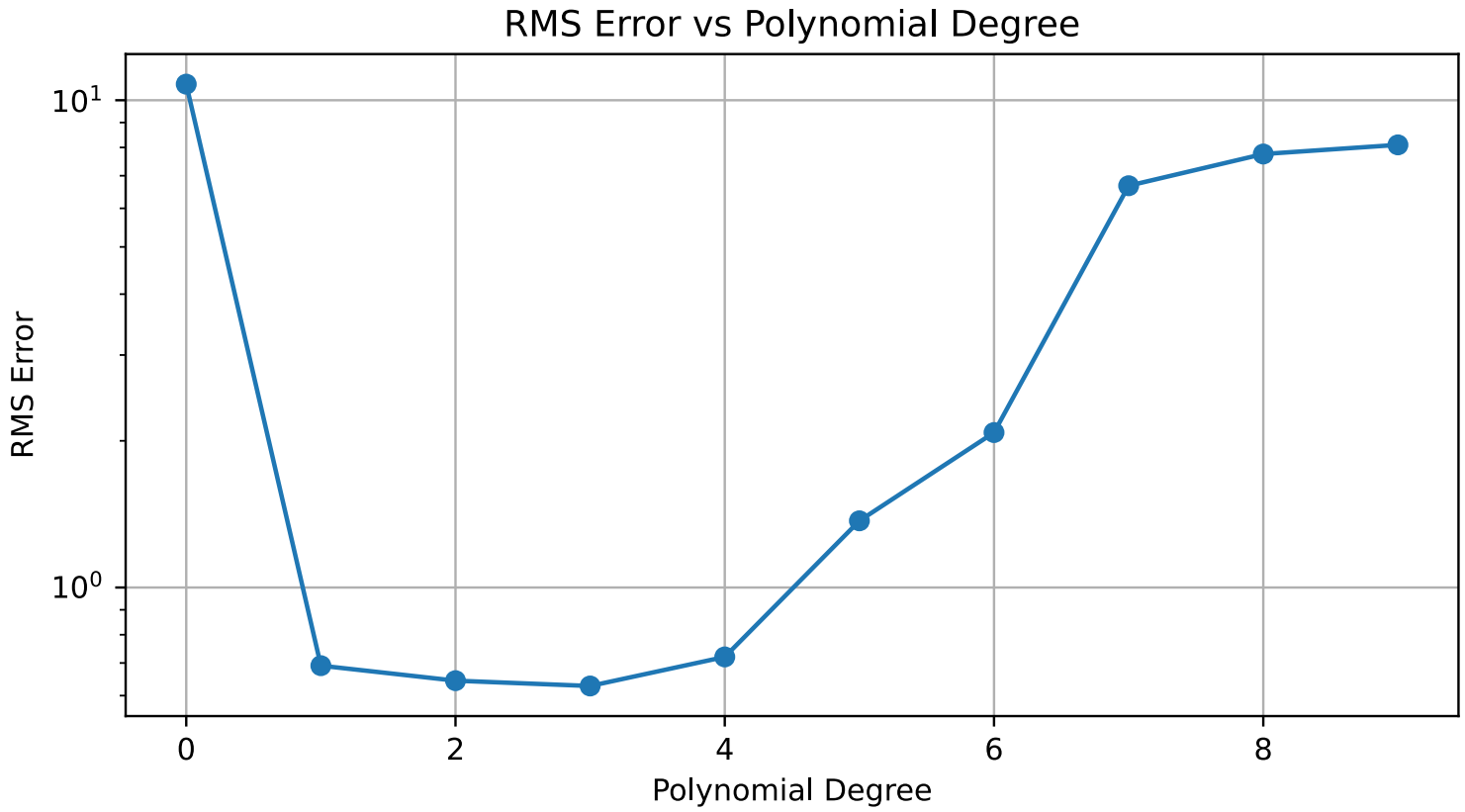
degree	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
0	20									
1	1.72	880								
2	0.876	1.03e+03	-3.68e+03							
3	0.684	1.09e+03	-8.16e+03	7.61e+04						
4	-0.0337	1.45e+03	-4.58e+04	1.45e+06	-1.63e+07					
5	2.61	-247	2.34e+05	-1.67e+07	4.82e+08	-4.88e+09				
6	-0.362	2.04e+03	-2.83e+05	3.3e+07	-1.81e+09	4.53e+10	-4.18e+11			
7	25.3	-1.99e+04	5.6e+06	-6.75e+08	4.21e+10	-1.41e+12	2.42e+13	-1.66e+14		
8	13.2	-8.59e+03	2.01e+06	-1.22e+08	-4.9e+09	8.96e+11	-4.09e+13	8.13e+14	-6.06e+15	
9	-7.83	1.67e+04	-9.21e+06	2.3e+09	-2.91e+11	2.06e+13	-8.53e+14	2.06e+16	-2.68e+17	1.46e+18

Higher order fits have larger and larger coefficients. Can we select "simpler" models putting a penalty on larger coefficients?

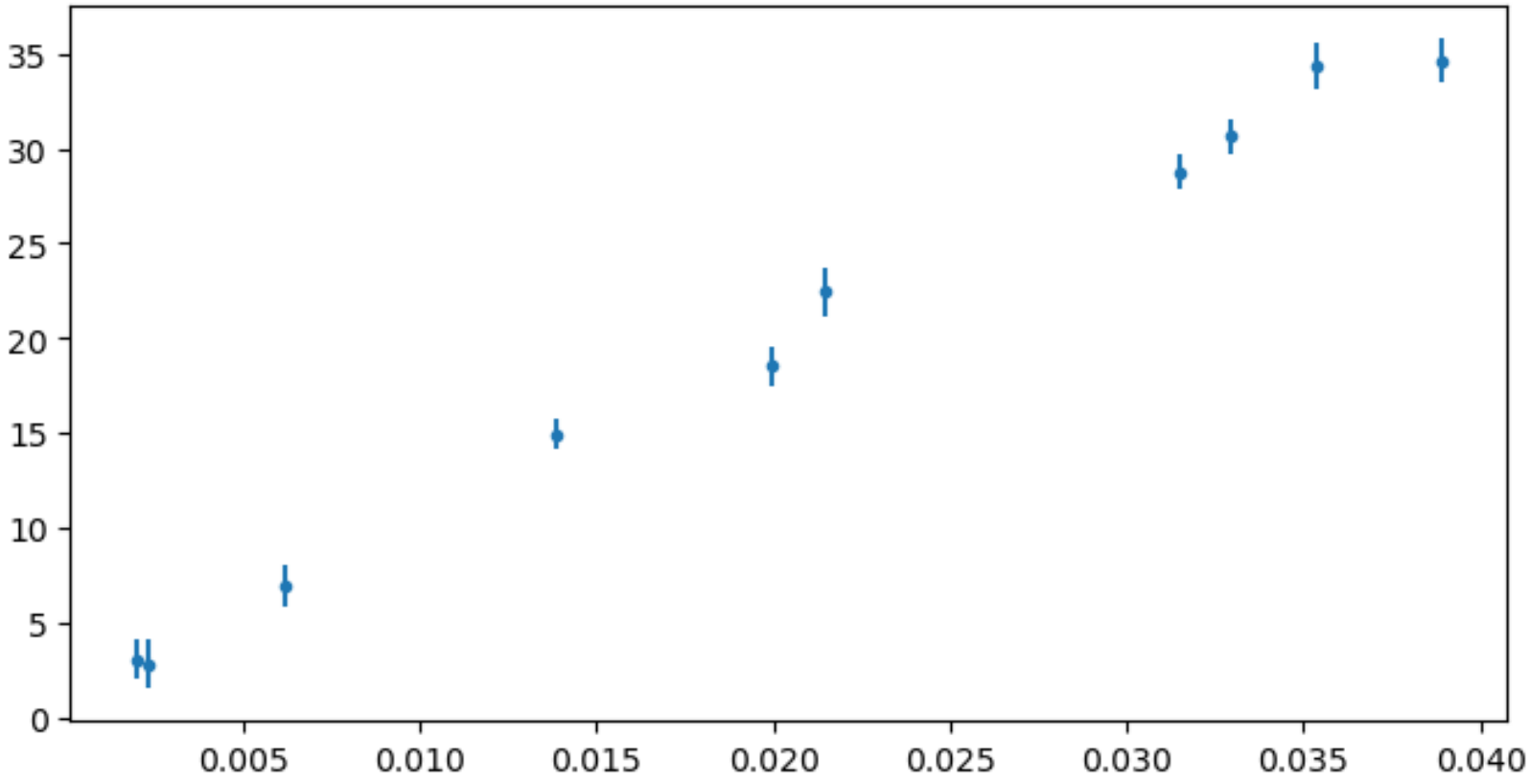
$$S_r(\mathbf{x},\mathbf{y},\mathbf{c}) = \frac{1}{2}\sum_{i=1}^{N}\frac{[y_i - y(x_i,\mathbf{c})]^2}{\sigma_i^2} + \frac{\alpha}{2}\sum_{k=0}^{n} c_k^2$$

regularization term

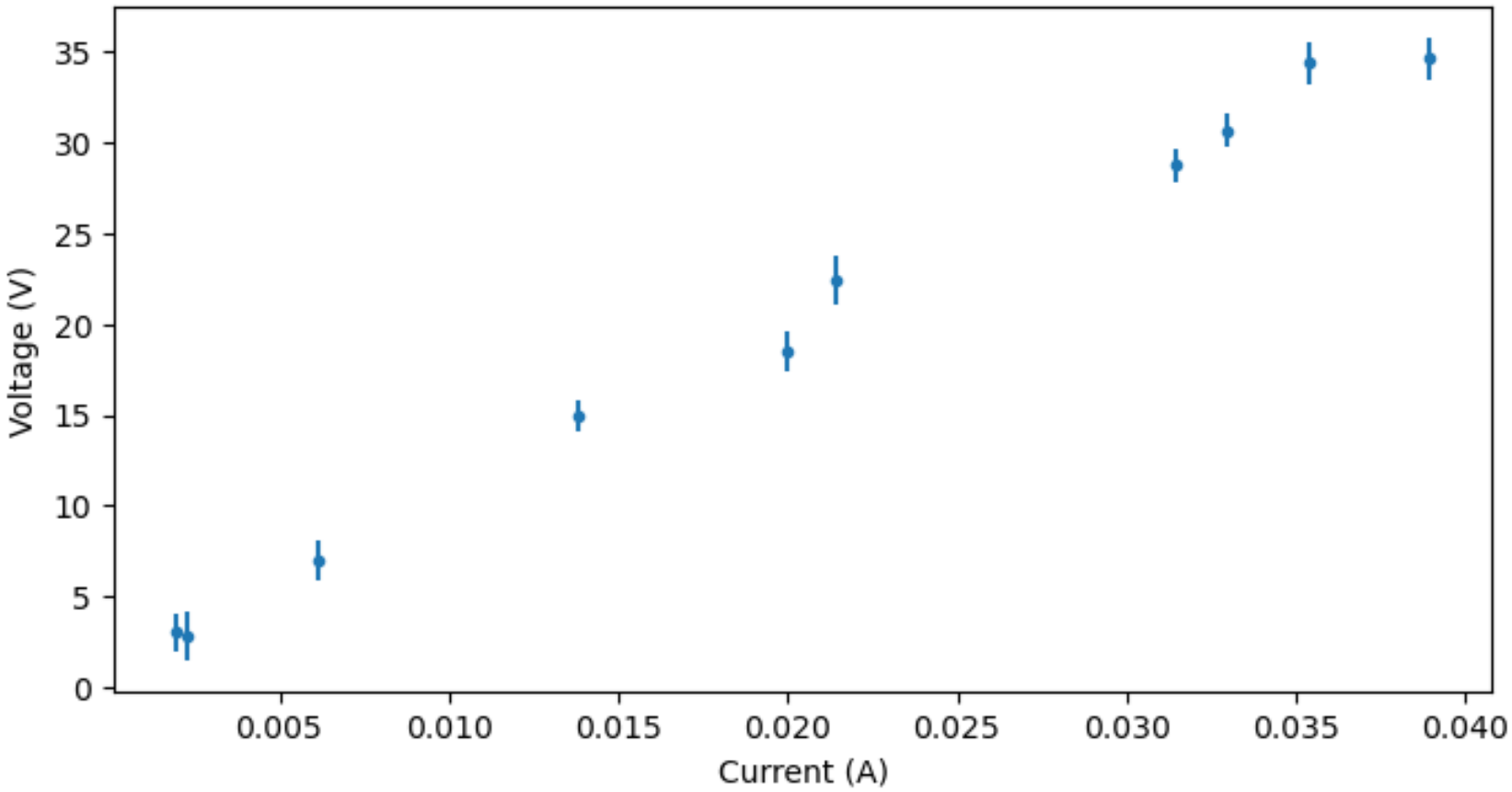

Poly fit (deg=9)
Model
data
35
30
25
20
15
10
5
0
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040


RMS Error vs Polynomial Degree
10^1
10^0
RMS Error
0
2
4
6
8
Polynomial Degree

Adding physical information



Adding physical information



The underlying physical model

We used simulated data for the characteristic curve of a 1 kΩ resistor, taking into account its temperature coefficient and its size.

Specifications:

Case Style	Ceramic
Diameter	3.2 mm
Dimensions	3.2 (dia.) x 9 mm
Lead Diameter	0.65 mm
Length	9 mm
Maximum Operating Temperature	+155°C
Maximum Temperature Coefficient	+350 ppm/°C
Minimum Operating Temperature	-55°C
Minimum Temperature Coefficient	-500 ppm/°C
Power Rating	1 W
Resistance	1 kΩ
Technology	Carbon Film
Temperature Coefficient	-500 to +350 ppm/°C
Termination Style	Axial
Tolerance	±5%
Lead Length	26 mm
Maximum Operating Voltage	400 V
Maximum Overload Voltage	800 V

The underlying physical model

Here, we used simulated data for the characteristic curve of a 1 kΩ resistor, taking into account its temperature coefficient and its size.

electric power = total power losses

$$R(T)\ I^2 = (1 - f)R(T)\ I^2 + \epsilon\sigma_{\mathrm{SB}}A(T^4 - T_0^4)$$

where

$$R(T) = R_0 + c_T(T - T_0)$$

The underlying physical model

Here, we used simulated data for the characteristic curve of a 1 kΩ resistor, taking into account its temperature coefficient and its size.

electric power = total power losses

electric power | power loss through leads | radiative power loss

$$R(T)\ I^2 = (1-f)R(T)\ I^2 + \epsilon\sigma_{\mathrm{SB}}A(T^4 - T_0^4)$$

where

$$R(T) = R_0 + c_T(T - T_0)$$

The numerical solution

We have to solve the following nonlinear equation

$$F(T) = f\left[R_0 + c_T(T - T_0)\right] I^2 - \epsilon\sigma_{\mathrm{SB}} A(T^4 - T_0^4) = 0$$

and to this end we use Newton's method which works as follows: we assume a reasonable starting value and we approximate the distance to the true solution by taking the linear approximation

$$0 = F(\theta_0 + \delta\theta) \approx F(\theta_0) + F'(\theta_0)\delta\theta$$

then

$$\delta\theta \approx -F(\theta_0)/F'(\theta_0)$$

and we find a new (better) approximation of the solution

$$\theta_1 = \theta_0 + \delta\theta = \theta_0 - F(\theta_0)/F'(\theta_0)$$

We iterate this scheme until we reach a predefined convergence threshold.

Fitting data – the Bayesian way

In the following we assume that the uncertainty over data points is described by a Gaussian distribution, i.e.,

$$p(t|x, \mathbf{c}, \beta) = \mathcal{N}(t|y(x, \mathbf{c}), \beta^{-1}), \qquad \beta = 1/\sigma^2$$

precision

and we obtain the likelihood

$$p(\mathbf{t}|\mathbf{x}, \mathbf{c}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \mathbf{c}), \beta^{-1}) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\beta^{-1}}} \exp\left\{-\frac{[t_n - y(x_n, \mathbf{c})]^2}{2\beta^{-1}}\right\}$$

and finally, the log-likelihood

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

Fitting data – the Bayesian way - 2

We can maximize this likelihood

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

with respect to the parameters, and we see that we recover the least squares method used earlier in a purely frequentist context.

To this we can add the maximization with respect to the beta parameter, in case this is unknown. This adds one more equation and we find

$$\frac{1}{\hat{\beta}} = \frac{1}{N}\sum_{n=1}^{N}\left[t_n - y(x_n, \mathbf{c})\right]^2$$

Fitting data – the Bayesian way - 3

The Bayesian solution adds more than just a point estimate plus uncertainty, it yields a distribution and we apply this to prediction

$$p(t|x, \hat{\mathbf{c}}, \hat{\beta}) = \mathcal{N}(t|y(x, \hat{\mathbf{c}}), \hat{\beta}^{-1})$$

However, we are still missing a prior for the cofficients that define the polynomial.

We select a Gaussian prior for the coefficients:

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

so that the posterior pdf is

$$p(\mathbf{c}|\mathbf{t}, \mathbf{x}, \beta) \propto p(\mathbf{t}|\mathbf{x}, \mathbf{c}, \beta)p(\mathbf{c}|\alpha)$$

Fitting data – the Bayesian way - 4

We maximize the posterior pdf to find the MAP estimate of the coefficients.

We also note that the maximization of the posterior amounts to the minimization of the expression

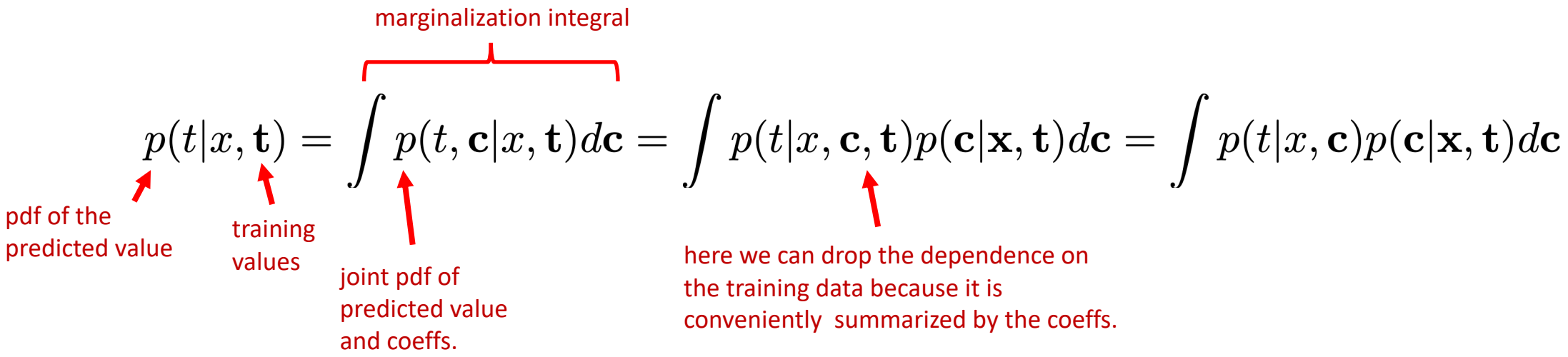
$$\frac{\beta}{2}\sum_{n=1}^{N}\left\{[t_n - y(x_n, \mathbf{c})]^2\right\} + \frac{\alpha}{2}\mathbf{c}^T\mathbf{c}$$

where we obtain in a "natural" way the previous regularizing term.

Fitting data – the Bayesian way - 5

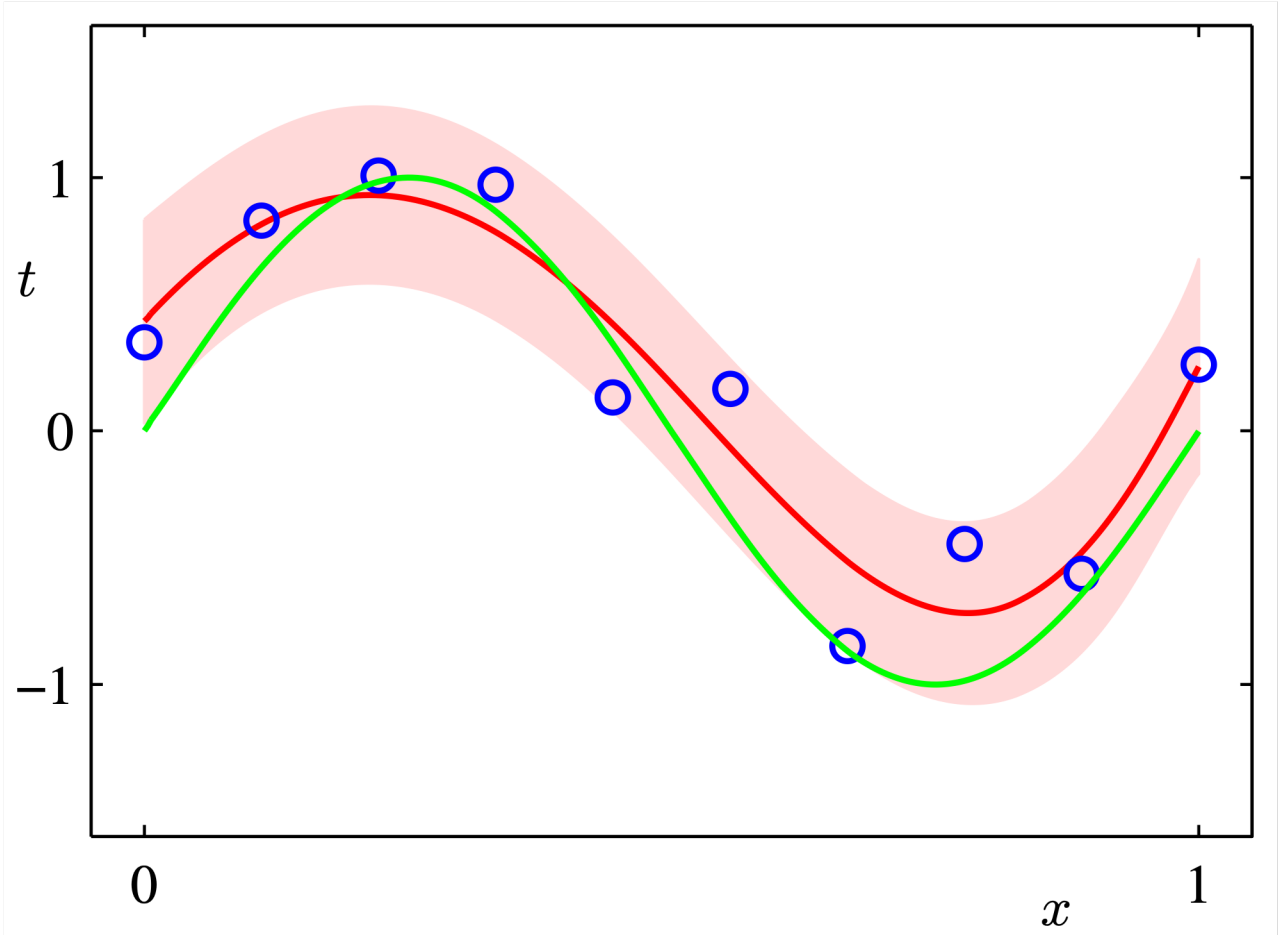
We are still missing one aspect of the Bayesian solution, the marginalized predictive posterior.

The marginalized predictive posterior is



With a Gaussian likelihood and a Gaussian prior such as in the previous examples, it is possible to carry out all calculations analytically. In other cases we must resort to numerical methods.

The predictive distribution resulting from a Bayesian treatment of polynomial curve fitting using an $M = 9$ polynomial, with the fixed parameters $\alpha = 5 \times 10^{-3}$ and $\beta = 11.1$ (corresponding to the known noise variance), in which the red curve denotes the mean of the predictive distribution and the red region corresponds to ± 1 standard deviation around the mean.



(figure from C.M. Bishop, PRML)

Linear prediction (frequentist approach)

This is an important application of the linear regression formalism and is generally based on the so-called ARMA model (or similar).

ARMA stands for AutoRegressive, Moving Average, and is defined by the following equation

$$s_n = -\sum_{k=1}^{p} a_k s_{n-k} + \sum_{\ell=0}^{q} b_\ell u_{n-\ell}$$

autoregressive term

moving average term

Linear prediction

This is an important application of the linear regression formalism and is generally based on the so-called ARMA model (or similar).

ARMA stands for AutoRegressive, Moving Average, and is defined by the following equation

$$s_n = -\sum_{k=1}^{p} a_k s_{n-k} + \sum_{\ell=0}^{q} b_\ell u_{n-\ell}$$

autoregressive term

moving average term

Linear predictive coding (LPC) has important applications in many different fields such as radar tracking, speech recognition and spectral analysis. An important application of LPC is found in the GSM standard (2G), where LPC is used to compress speech (the LPC coefficients are sent instead of the stream of audio samples, achieving a good sound compression).

Linear prediction: finding the model parameters of an AR(p) model

Consider the model

$$s_n = -\sum_{k=1}^{p} a_k s_{n-k}$$

and define the error function

$$S = \sum_n \left(s_n + \sum_{k=1}^{p} a_k s_{n-k} \right)^2$$

We find the normal equations as usual

$$\frac{\partial S}{\partial a_j} = 2 \sum_n s_{n-j} \left(s_n + \sum_{k=1}^{p} a_k s_{n-k} \right) = 2 \sum_n \left(s_{n-j} s_n + \sum_{k=1}^{p} a_k s_{n-j} s_{n-k} \right)$$

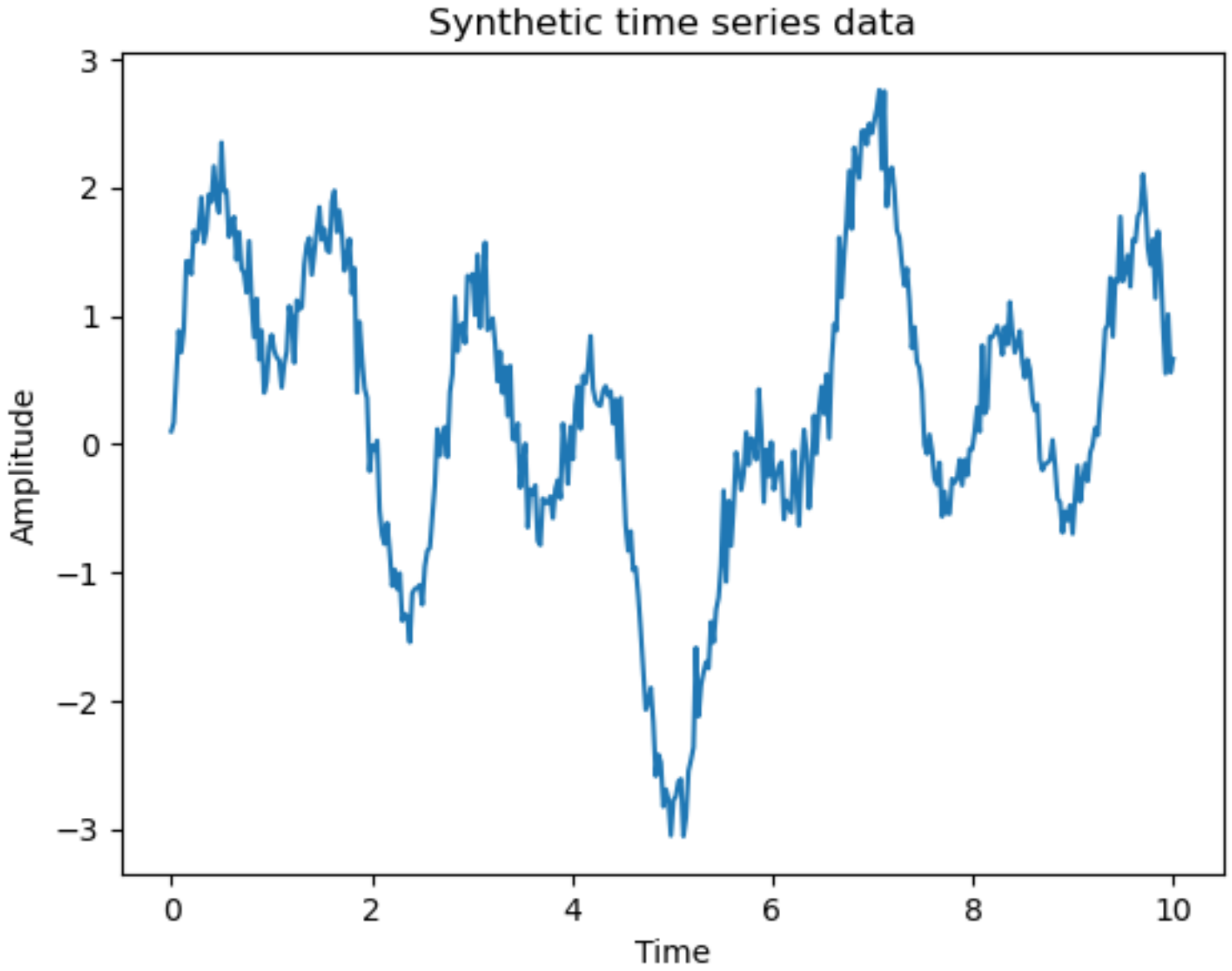
$$\propto 2 \left(\langle s_{n-j} s_n \rangle + \sum_{k=1}^{p} a_k \langle s_{n-j} s_{n-k} \rangle \right) = 0$$

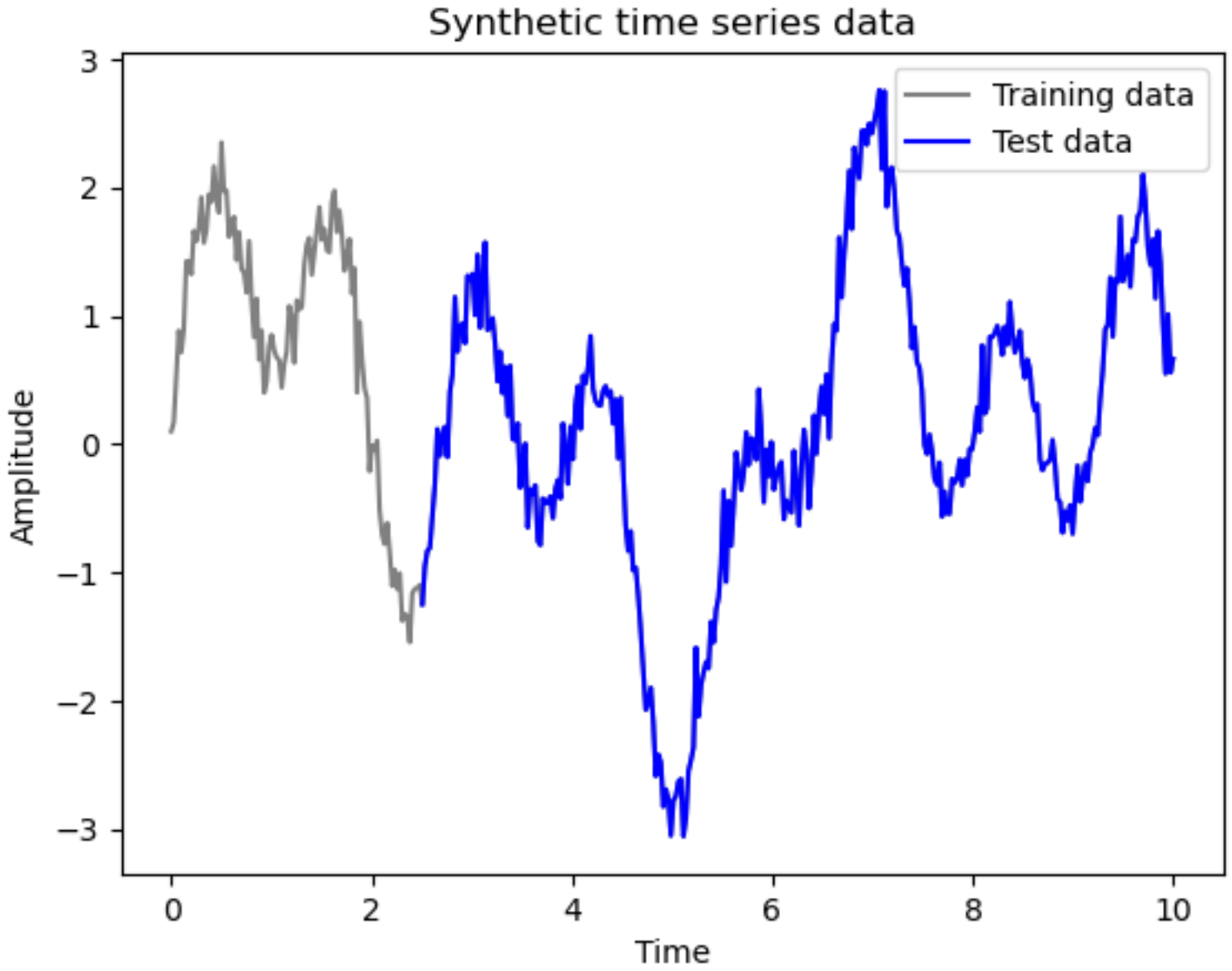
$$\left(\text{where: } \langle s_{n-j} s_{n-k} \rangle \propto \sum_n s_{n-j} s_{n-k} \right)$$

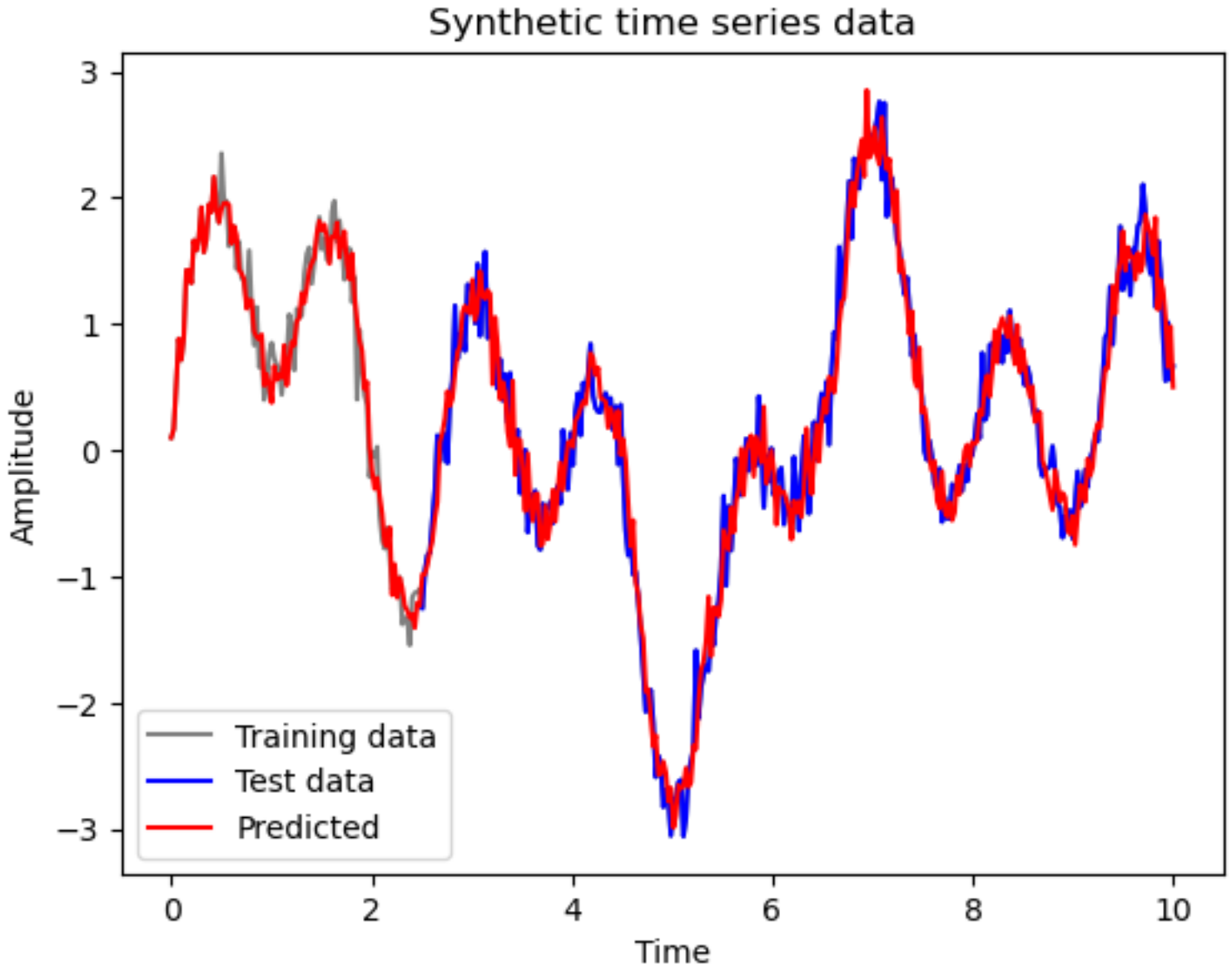
Linear prediction: finding the model parameters of an AR(p) model / 2

We write the normal equations in matrix form and find their solution with the usual methods of linear algebra

$$\begin{pmatrix} \langle s_{n-1}s_{n-1}\rangle & \langle s_{n-1}s_{n-2}\rangle & \cdots & \langle s_{n-1}s_{n-p}\rangle \\ \langle s_{n-2}s_{n-1}\rangle & \langle s_{n-2}s_{n-2}\rangle & \cdots & \langle s_{n-2}s_{n-p}\rangle \\ \vdots & \vdots & & \vdots \\ \langle s_{n-p}s_{n-1}\rangle & \langle s_{n-p}s_{n-2}\rangle & \cdots & \langle s_{n-p}s_{n-p}\rangle \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix} = \begin{pmatrix} \langle s_{n-1}s_n\rangle \\ \langle s_{n-2}s_n\rangle \\ \vdots \\ \langle s_{n-p}s_n\rangle \end{pmatrix}$$


Synthetic time series data
Amplitude
Time
3
2
1
0
−1
−2
−3
0
2
4
6
8
10


Synthetic time series data
Training data
Test data
Amplitude
Time
3
2
1
0
−1
−2
−3
0
2
4
6
8
10


Synthetic time series data
Amplitude
Time
Training data
Test data
Predicted
3
2
1
0
−1
−2
−3
0
2
4
6
8
10

Meaning of the AR model coefficients

Consider the sinusoidal function

$$y(t) = A\sin(\omega t + \varphi)$$

and take discrete steps forward in time

$$\begin{aligned} y(t+2\Delta t) &= A\sin(\omega t + \varphi + 2\omega\Delta t) \\ &= A\sin(\omega t + \varphi + \omega\Delta t)\cos\omega\Delta t + A\cos(\omega t + \varphi + \omega\Delta t)\sin\omega\Delta t \\ &= y(t+\Delta t)\cos\omega\Delta t + A\cos(\omega t + \varphi)\cos\omega\Delta t\sin\omega\Delta t - y(t)\sin^2\omega\Delta t \end{aligned}$$

$$\begin{aligned} y(t+\Delta t) &= A\sin(\omega t + \varphi + \omega\Delta t) \\ &= A\sin(\omega t + \varphi)\cos\omega\Delta t + A\cos(\omega t + \varphi)\sin\omega\Delta t \\ &= y(t)\cos\omega\Delta t + A\cos(\omega t + \varphi)\sin\omega\Delta t \end{aligned}$$

Meaning of the AR model coefficients

Consider the sinusoidal function

$$y(t) = A\sin(\omega t + \varphi)$$

and take discrete steps forward in time

$$\begin{aligned} y(t+2\Delta t) &= A\sin(\omega t + \varphi + 2\omega\Delta t) \\ &= A\sin(\omega t + \varphi + \omega\Delta t)\cos\omega\Delta t + A\cos(\omega t + \varphi + \omega\Delta t)\sin\omega\Delta t \\ &= y(t+\Delta t)\cos\omega\Delta t + A\cos(\omega t + \varphi)\cos\omega\Delta t\sin\omega\Delta t - y(t)\sin^2\omega\Delta t \end{aligned}$$

$$\begin{aligned} y(t+\Delta t) &= A\sin(\omega t + \varphi + \omega\Delta t) \\ &= A\sin(\omega t + \varphi)\cos\omega\Delta t + A\cos(\omega t + \varphi)\sin\omega\Delta t \\ &= y(t)\cos\omega\Delta t + A\cos(\omega t + \varphi)\sin\omega\Delta t \end{aligned}$$

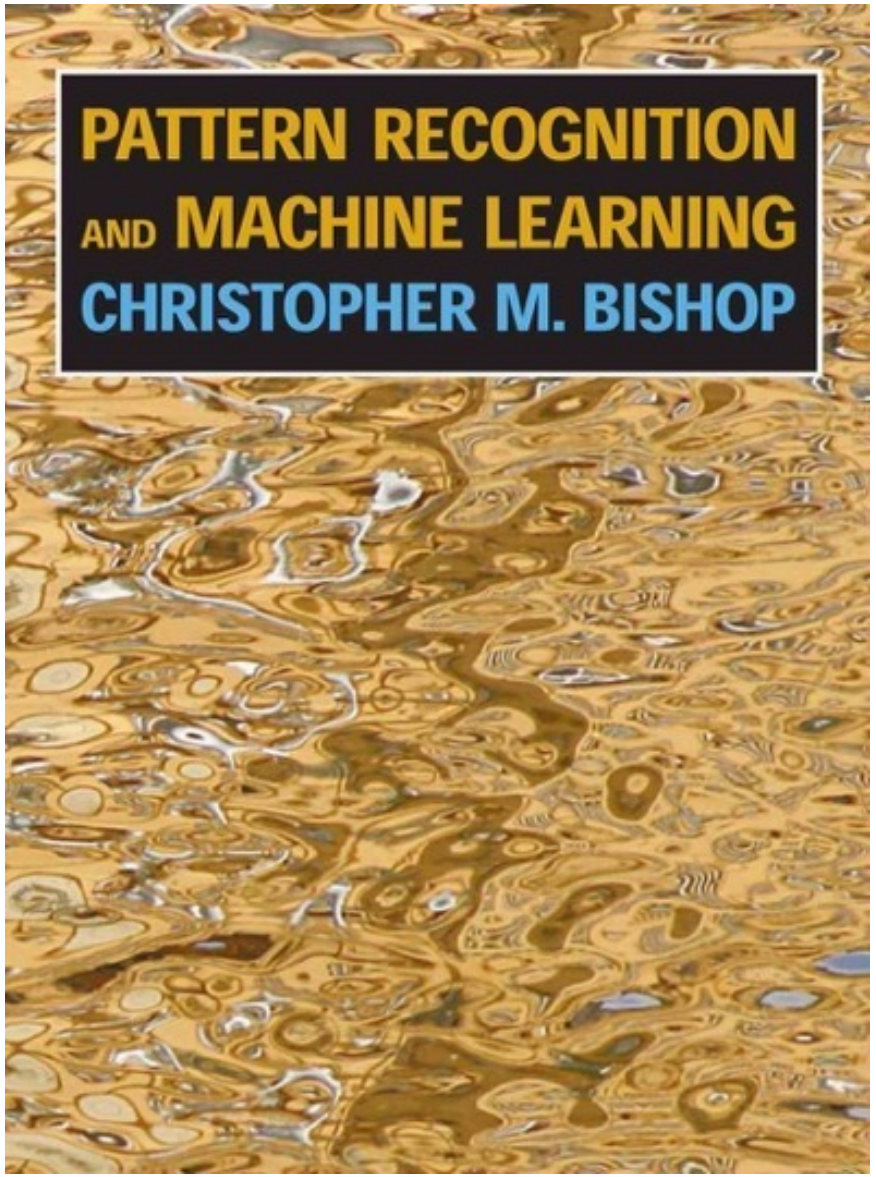
$$y(t+2\Delta t) = 2\cos\omega\Delta t\; y(t+\Delta t) - y(t)$$

This is an AR(2) model !!!

$$y_n = 2\cos\omega\Delta t\; y_{n-1} - y_{n-2}$$

Research problem:

1. how would you modify the frequentist scheme, casting linear prediction into a Bayesian framework?
2. which kind of data would you test your scheme on?
3. how would you validate your scheme?



Part of the material in these slides is taken from this book.

The book is freely downloadable, see the website

https://www.microsoft.com/en-us/research/people/cmbishop/prml-book/