Machine Learning in just a few minutes

Jan Peters
Gerhard Neumann

Purpose of this Lecture

- Foundations of machine learning tools for robotics
- We focus on regression methods and general principles
  - Often needed in robotics

Content of this Lecture

- Math and Statistics Refresher
- What is Machine Learning?
- Model-Selection
- Linear Regression
  - Frequentist Approach
  - Bayesian Approach

Statistics Refresher:
Sweet memories from High School...

What is a random variable X **?**

X is a variable whose value x is subject to variations due to chance

What is a distribution $p(X = x)$ **?**

Describes the probability that the random variable will be equal to a certain value

What is an expectation?

$$\mathbb{E}_{p(X)}[f(x)] = \int p(x)f(x)dx$$

Statistics Refresher:
Sweet memories from High School...

- **What is a joint, a conditional and a marginal distribution?**

$$p(x,y) = p(y|x)p(x)$$

$$\text{joint} = \text{conditional} \times \text{marginal}$$

- **What is independence of random variables?**

$$p(x,y) = p(x)p(y)$$

- **What does marginalization mean?**

$$p(x) = \int p(x,y)dy$$

- **And finally... what is Bayes Theorem?**

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

Math Refresher:
Some more fancy math…

- From now on, matrices are your friends… derivatives too

$$\frac{dy}{d\boldsymbol{x}} = \left[\frac{dy}{dx_1}, \ldots, \frac{dy}{dx_n}\right] \qquad \frac{d\boldsymbol{y}}{d\boldsymbol{x}} = \begin{bmatrix} \frac{dy_1}{dx_1} & \cdots & \frac{dy_1}{dx_n} \\ \vdots & \vdots & \vdots \\ \frac{dy_m}{dx_1} & \cdots & \frac{dy_m}{dx_n} \end{bmatrix}$$

- Some more matrix calculus

$$\frac{d\boldsymbol{a}^T\boldsymbol{x}}{d\boldsymbol{x}} = \frac{d\boldsymbol{x}^T\boldsymbol{a}}{d\boldsymbol{x}} = \boldsymbol{a}^T \qquad \frac{d\boldsymbol{A}\boldsymbol{x}}{d\boldsymbol{x}} = \boldsymbol{A} \qquad \frac{d\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}}{d\boldsymbol{x}} = \boldsymbol{x}^T(\boldsymbol{A} + \boldsymbol{A}^T)$$

- Need more ? [Wikipedia on Matrix Calculus]

Math Refresher: Inverse of matrices

How can we invert a matrix that is not a square matrix?

$$\boldsymbol{J} \in \mathbb{R}^{n \times m}$$

Left-Pseudo Inverse:

$$\boldsymbol{J}^{\dagger}\boldsymbol{J} = \underbrace{(\boldsymbol{J}^T\boldsymbol{J})^{-1}\boldsymbol{J}^T}_{\text{left multiplied}}\boldsymbol{J} = \boldsymbol{I}_m$$

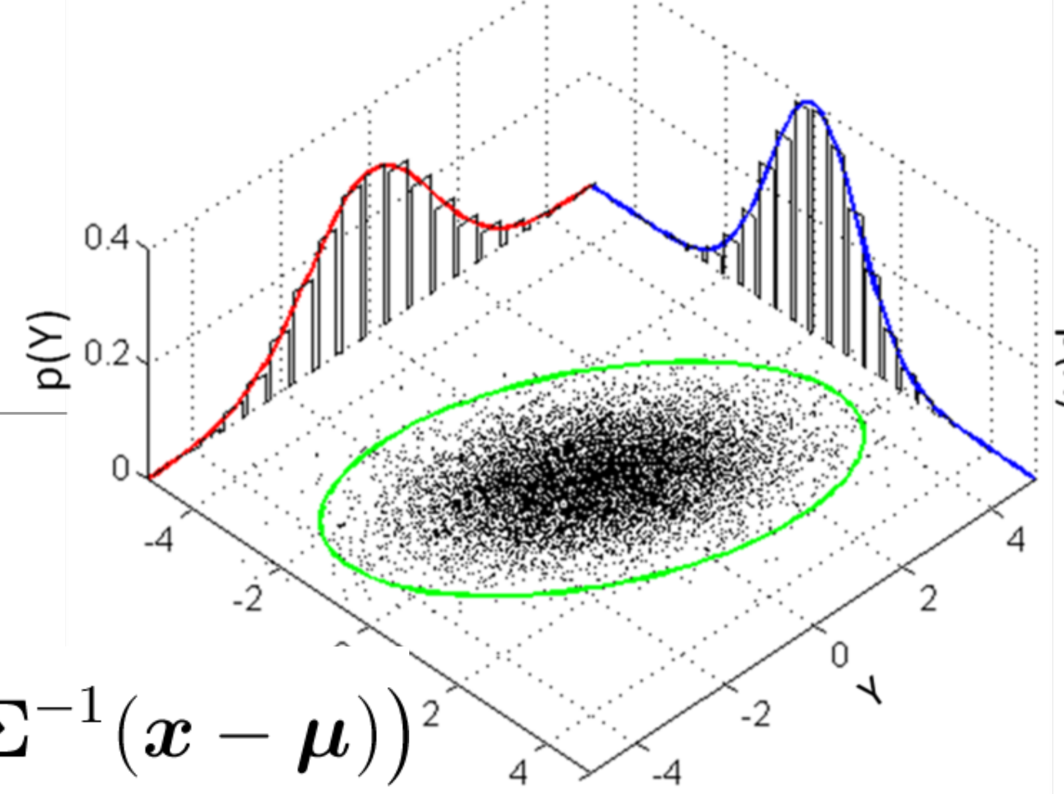
works, if J has full column rank

Right Pseudo Inverse:

$$\boldsymbol{J}\boldsymbol{J}^{\dagger} = \boldsymbol{J}\underbrace{\boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{J}^T)^{-1}}_{\text{right multiplied}} = \boldsymbol{I}_n$$

works, if J has full row rank

Statistics Refresher:
Meet some old friends…



- **Gaussian Distribution**

$$\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}\right)=\frac{1}{|2\pi\boldsymbol{\Sigma}|^{\frac{1}{2}}}\exp\left(-0.5(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)^2$$

- Covariance matrix $\boldsymbol{\Sigma}$ captures linear correlation

- Product: Gaussian stays Gaussian

$$\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{a},\boldsymbol{A}\right)\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{b},\boldsymbol{B}\right)=\mathcal{N}(\boldsymbol{x}|\ldots)$$

- Mean is also the mode

$$\boldsymbol{\mu}=\operatorname{argmax}_{\boldsymbol{x}}\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})$$

Statistics Refresher:
Meet some old friends…

- Joint from Marginal and Conditional

$$\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{a},\boldsymbol{A}\right)\mathcal{N}\left(\boldsymbol{y}|\boldsymbol{b}+\boldsymbol{F}\boldsymbol{x},\boldsymbol{B}\right)=\mathcal{N}\left(\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix}\middle|\begin{bmatrix}\boldsymbol{a}\\ \boldsymbol{b}+\boldsymbol{F}\boldsymbol{a}\end{bmatrix},\begin{bmatrix}\boldsymbol{A} & \boldsymbol{F}^T\boldsymbol{A}^T\\ \boldsymbol{F}^T\boldsymbol{A}^T & \boldsymbol{B}+\boldsymbol{F}\boldsymbol{A}^T\boldsymbol{F}^T\end{bmatrix}\right)$$

$$p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x}) \Longrightarrow p(\boldsymbol{x},\boldsymbol{y})$$

- Marginal and Conditional Gaussian from Joint

$$\mathcal{N}\left(\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix}\middle|\begin{bmatrix}\boldsymbol{a}\\ \boldsymbol{b}\end{bmatrix},\begin{bmatrix}\boldsymbol{A} & \boldsymbol{C}\\ \boldsymbol{C}^T & \boldsymbol{B}\end{bmatrix}\right)=\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{a},\boldsymbol{A}\right)\mathcal{N}\left(\boldsymbol{y}|\boldsymbol{b}+\boldsymbol{C}^T\boldsymbol{A}^{-1}(\boldsymbol{x}-\boldsymbol{a}),\boldsymbol{B}-\boldsymbol{C}^T\boldsymbol{A}^{-1}\boldsymbol{C}\right)$$

$$p(\boldsymbol{x},\boldsymbol{y}) \Longrightarrow p(\boldsymbol{x}) \qquad p(\boldsymbol{y}|\boldsymbol{x})$$

Statistics Refresher:
Meet some old friends…

- Bayes Theorem for Gaussians

$$p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x}) \Rightarrow p(\boldsymbol{x}|\boldsymbol{y})$$

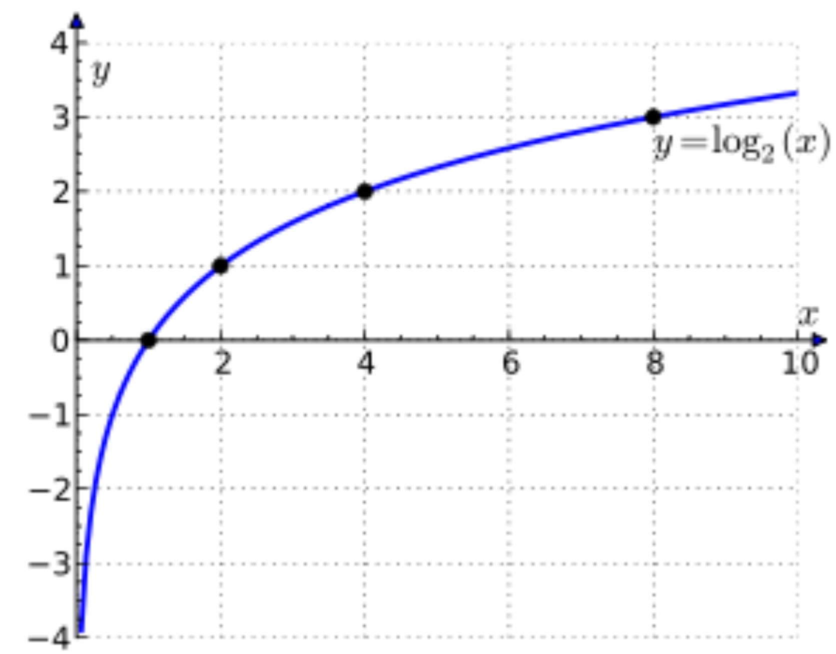
$$\begin{aligned} p(\boldsymbol{x}) &= \mathcal{N}\left(\boldsymbol{x}|\mathbf{0}, \boldsymbol{A}\right) \\ p(\boldsymbol{y}|\boldsymbol{x}) &= \mathcal{N}\left(\boldsymbol{y}|\boldsymbol{F}\boldsymbol{x}, \sigma^2\boldsymbol{I}\right) \end{aligned} \Rightarrow \begin{aligned} p(\boldsymbol{y}) &= \mathcal{N}\left(\boldsymbol{y}|\boldsymbol{F}\boldsymbol{a}, \sigma^2\boldsymbol{I} + \boldsymbol{F}\boldsymbol{A}\boldsymbol{F}^T\right) \\ p(\boldsymbol{x}|\boldsymbol{y}) &= \mathcal{N}\left(\boldsymbol{x}|\boldsymbol{\Sigma}\boldsymbol{F}^T\boldsymbol{y}, \sigma^2\boldsymbol{\Sigma}\right) \end{aligned}$$

$$\boldsymbol{\Sigma} = \left(\boldsymbol{F}^T\boldsymbol{F} + \sigma^2\boldsymbol{A}^{-1}\right)^{-1}$$

Damped Pseudo Inverse

May I introduce you? The good old logarithm

It's monoton $a > b \Rightarrow \log(a) > \log(b)$



… but not boring, as:

- Product is easy… $\log \prod_{i=1}^{N} a_i = \sum_{i=1}^{N} \log a_i$
- Division a piece of cake… $\log \frac{1}{a} = -\log a$
- Exponents also… $\log\left(a^b\right) = b \log a$

Content of this Lecture

- What is Machine Learning?
- Model-Selection
- Linear Regression
  - Frequentist Approach
  - Bayesian Approach

Why Machine Learning

- *We are drowning in information and starving for knowledge*. — John Naisbitt.
- **Era of big data:**
  - In 2008 there are about 1 trillion web pages
  - 20 hours of video are uploaded to YouTube every minute
  - Walmart handles more than 1M transactions per hour and has databases containing more than 2.5 petabytes (2.5×10^{15}) of information.

No human being can deal with the data avalanche!

Why Machine Learning?

I keep saying the sexy job in the next ten years will be statisticians and machine learners. People think I’m joking, but who would’ve guessed that computer engineers would’ve been the sexy job of the 1990s? The ability to take data — to be able to understand it, to process it, to extract value from it, to visualize it, to communicate it — that’s going to be a hugely important skill in the next decades.

Hal Varian, 2009

Chief Engineer of Google

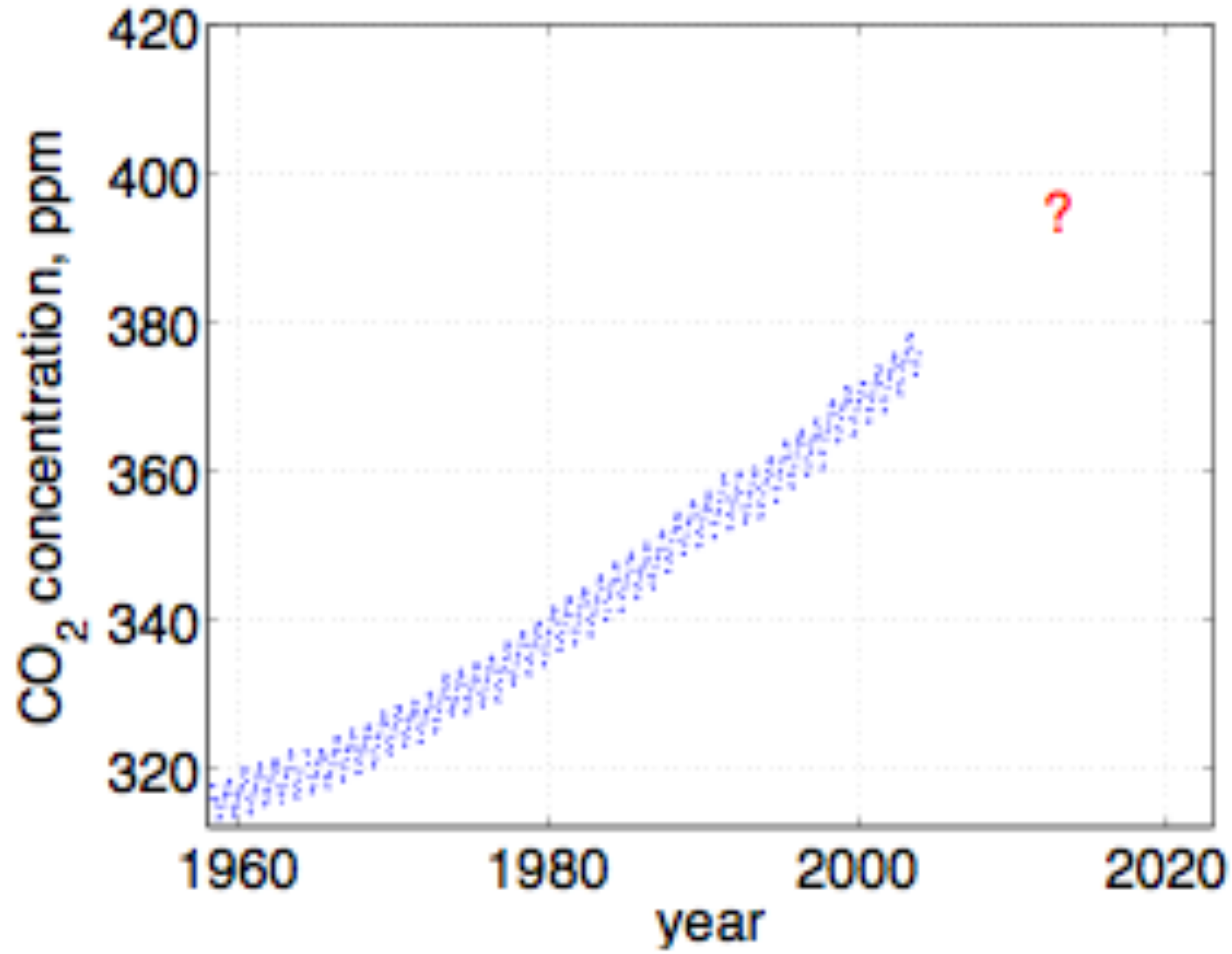
Types of Machine Learning

Machine Learning

- predictive (supervised)
- descriptive (unsupervised)
- Active (reinforcement learning)

Prediction Problem (=Supervised Learning)

What will be the CO^2 concentration in the future?

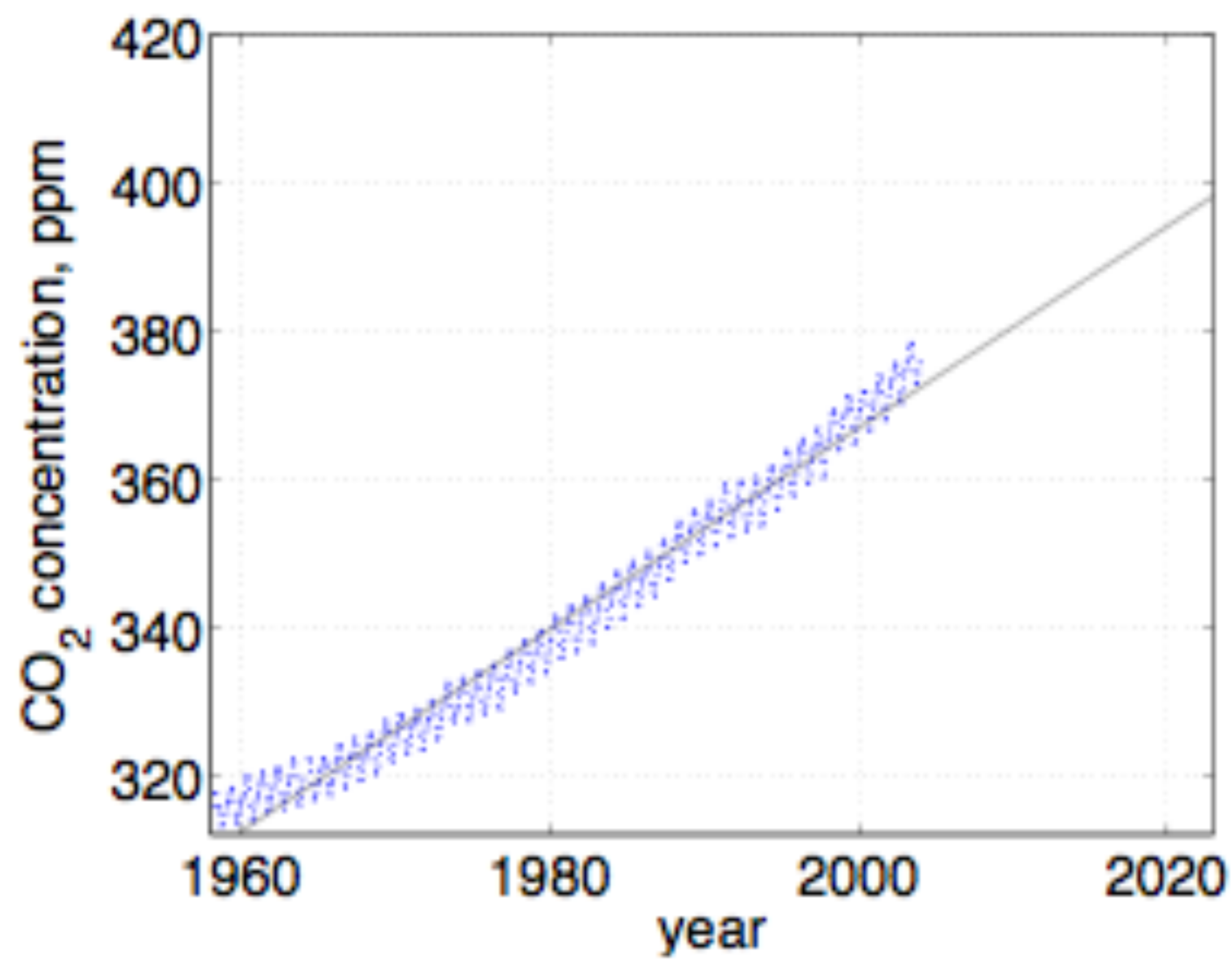


Prediction Problem (=Supervised Learning)

What will be the CO² concentration in the future?

Different prediction models possible

- Linear

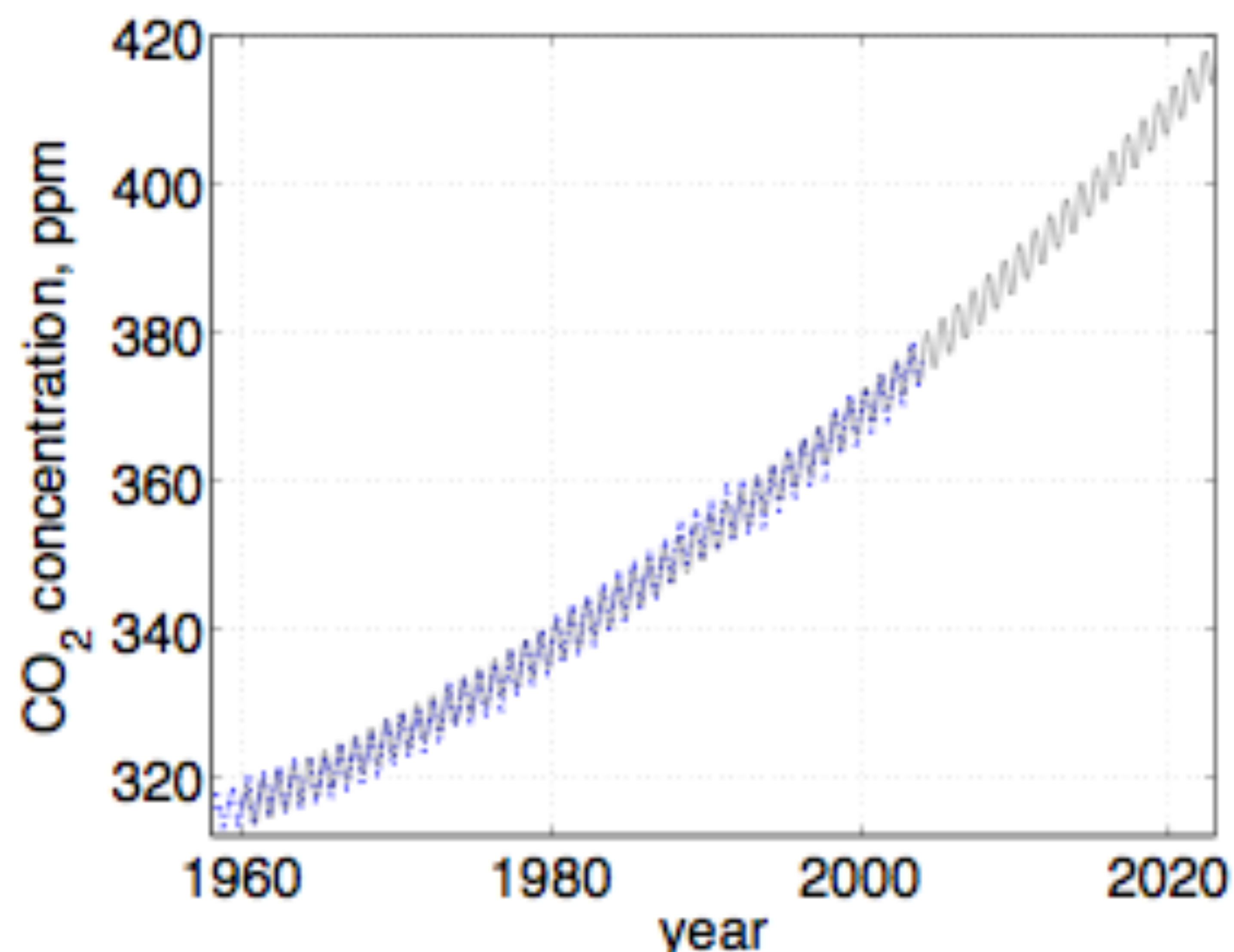


Prediction Problem (=Supervised Learning)

What will be the CO² concentration in the future?

Different prediction models possible

- Linear
- Exponential with seasonal trends



Formalization of Predictive Problems

In predictive problems, we have the following data-set

$$\mathscr{D} = \left\{ (\mathbf{x}_i, y_i) \middle| i = 1, 2, 3, \ldots, n \right\}$$

$\boldsymbol{x} \ldots$ inputs, $y \ldots$ output / target

Two most prominent examples are:

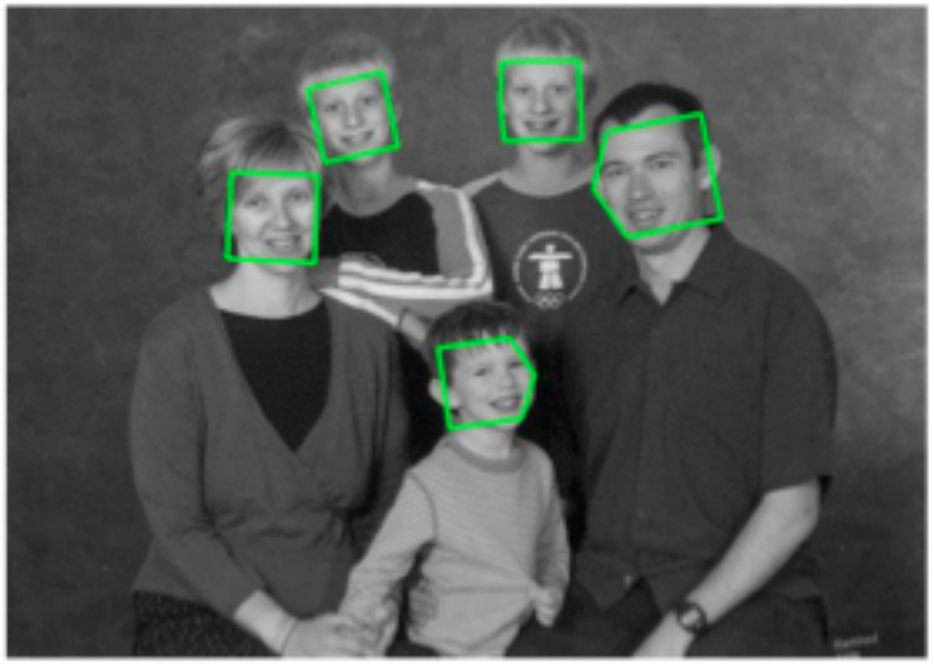
1. *Classification*: Discrete outputs or labels $y_q \in \{0, 1\}$

   Most likely class: $y_q = \operatorname{argmax}_y p(y|\mathbf{x} = \mathbf{x}_q)$.

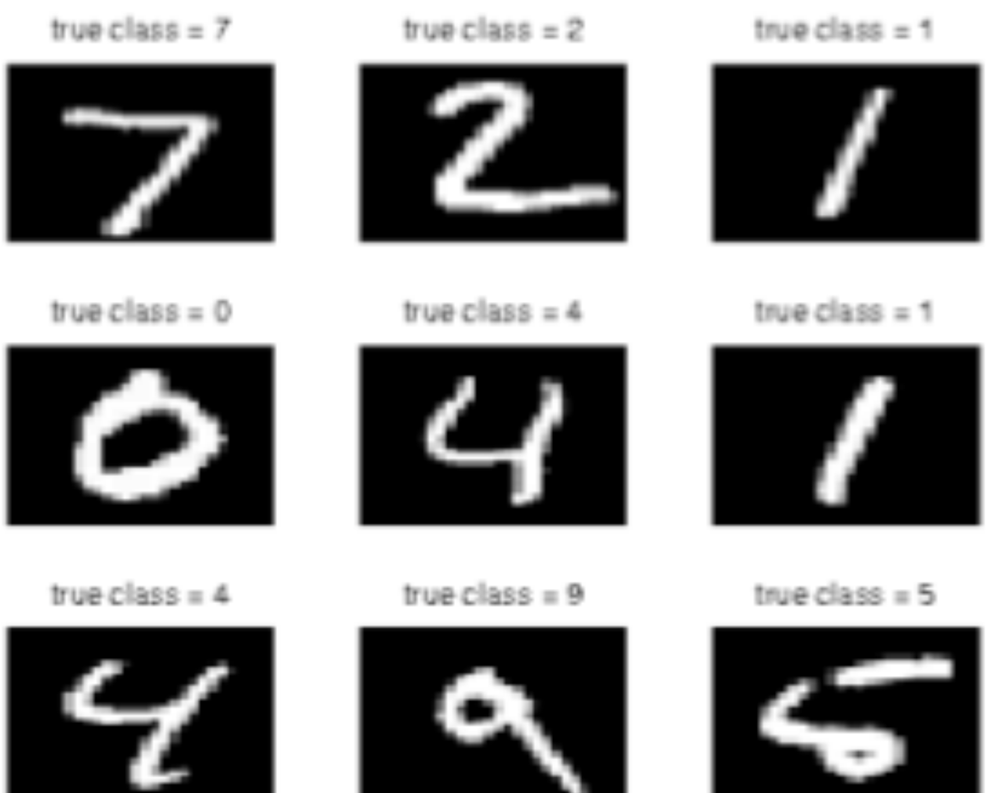
2. *Regression*: Continuous outputs or labels $y_q \in \mathbb{R}$.

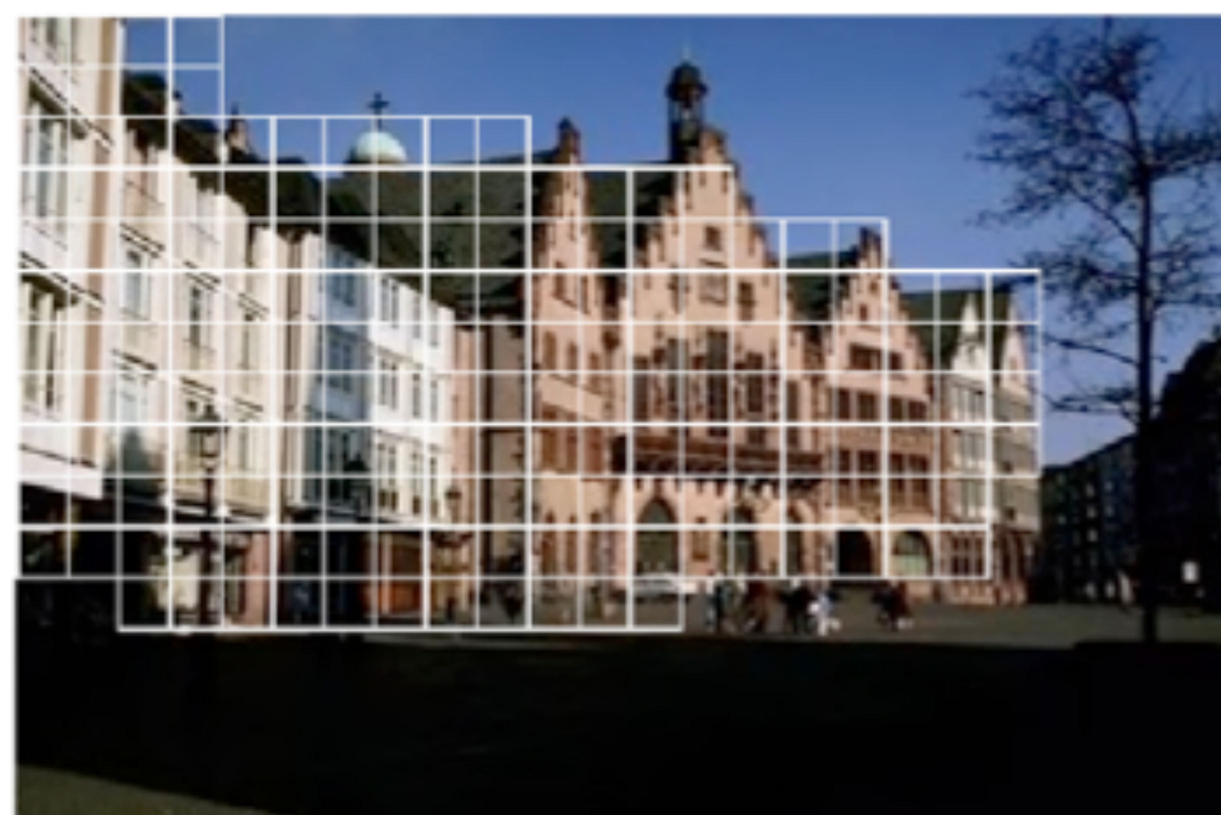
   Expected output: $y_q = E\{y|\mathbf{x} = \mathbf{x}_q\} = \int y p(y|\mathbf{x} = \mathbf{x}_q) dy$.

Examples of Classification



- Document classification, e.g., Spam Filtering
- Image classification: Classifying flowers, face detection, face recognition, handwriting recognition, ...





Examples of Regression

- Predict tomorrow's stock market price given current market conditions and other possible side information.
- Predict the amount of prostate specific antigen (PSA) in the body as a function of a number of different clinical measurements.
- Predict the temperature at any location inside a building using weather data, time, door sensors, etc.
- Predict the age of a viewer watching a given video on YouTube.

➡ Many problems in robotics can be addressed by regression!

Types of Machine Learning

Machine Learning

predictive
(supervised)

descriptive
(unsupervised)

Active
(reinforcement learning)

Formalization of Descriptive Problems

In descriptive problems, we have

$$\mathscr{D} = \left\{ \mathbf{x}_i \middle| i = 1, 2, 3, \ldots, n \right\}$$

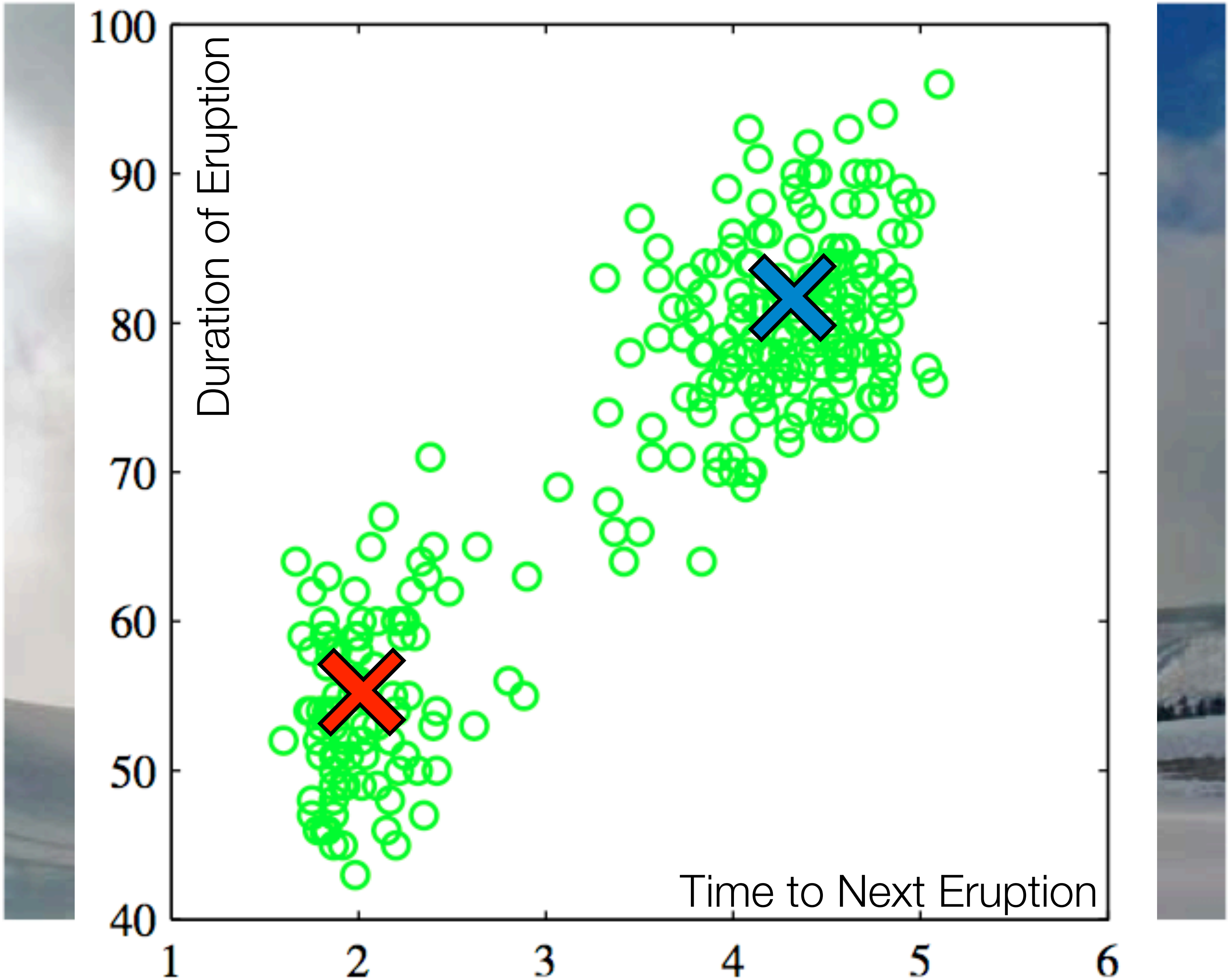
Three prominent examples are:

1. *Clustering*: Find groups of data which belong together.

2. *Dimensionality Reduction*: Find the latent dimension of your data.

3. *Density Estimation*: Find the probability of your data...

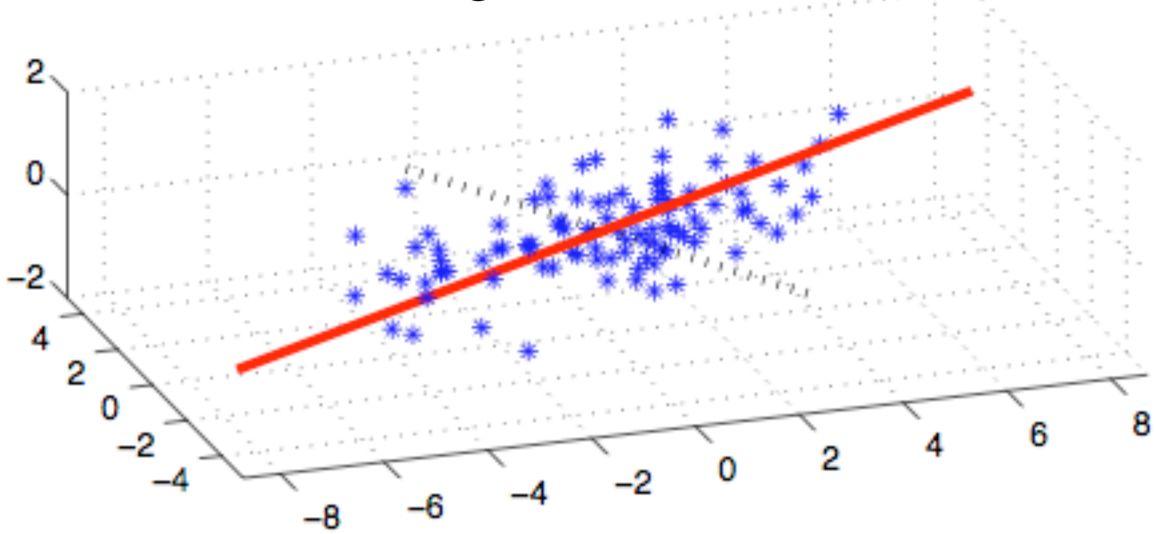
Old Faithful



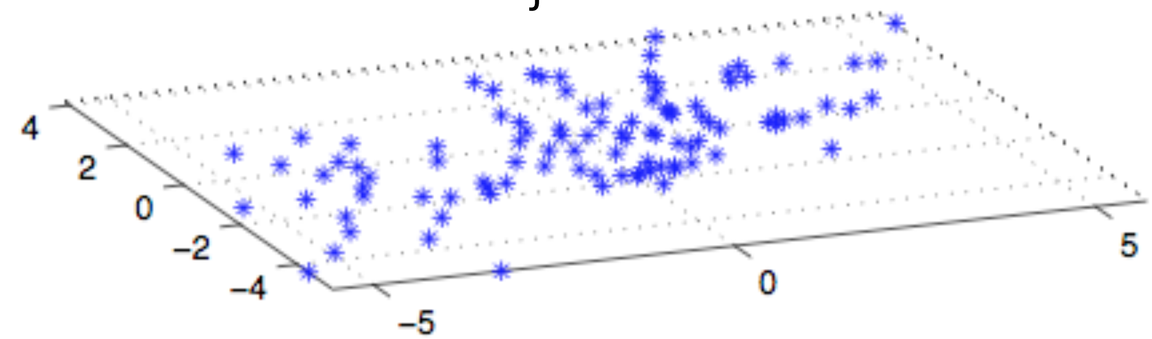
This is called Clustering!

Dimensionality Reduction

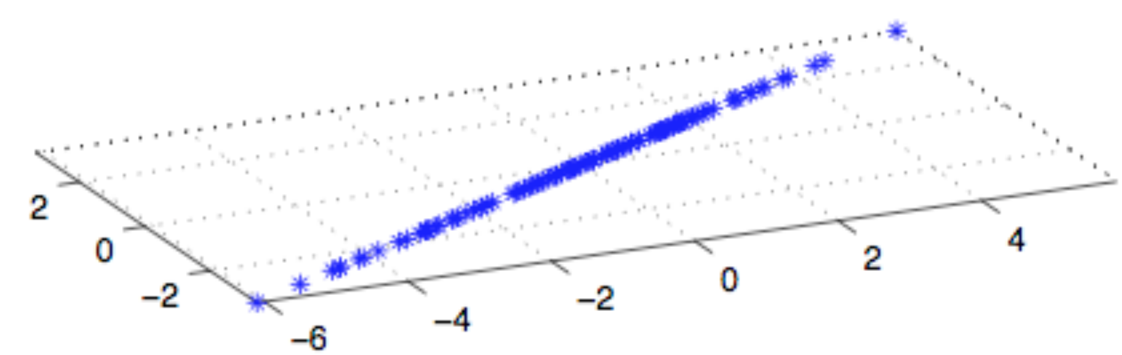
Original Data



2D Projection



1D Projection

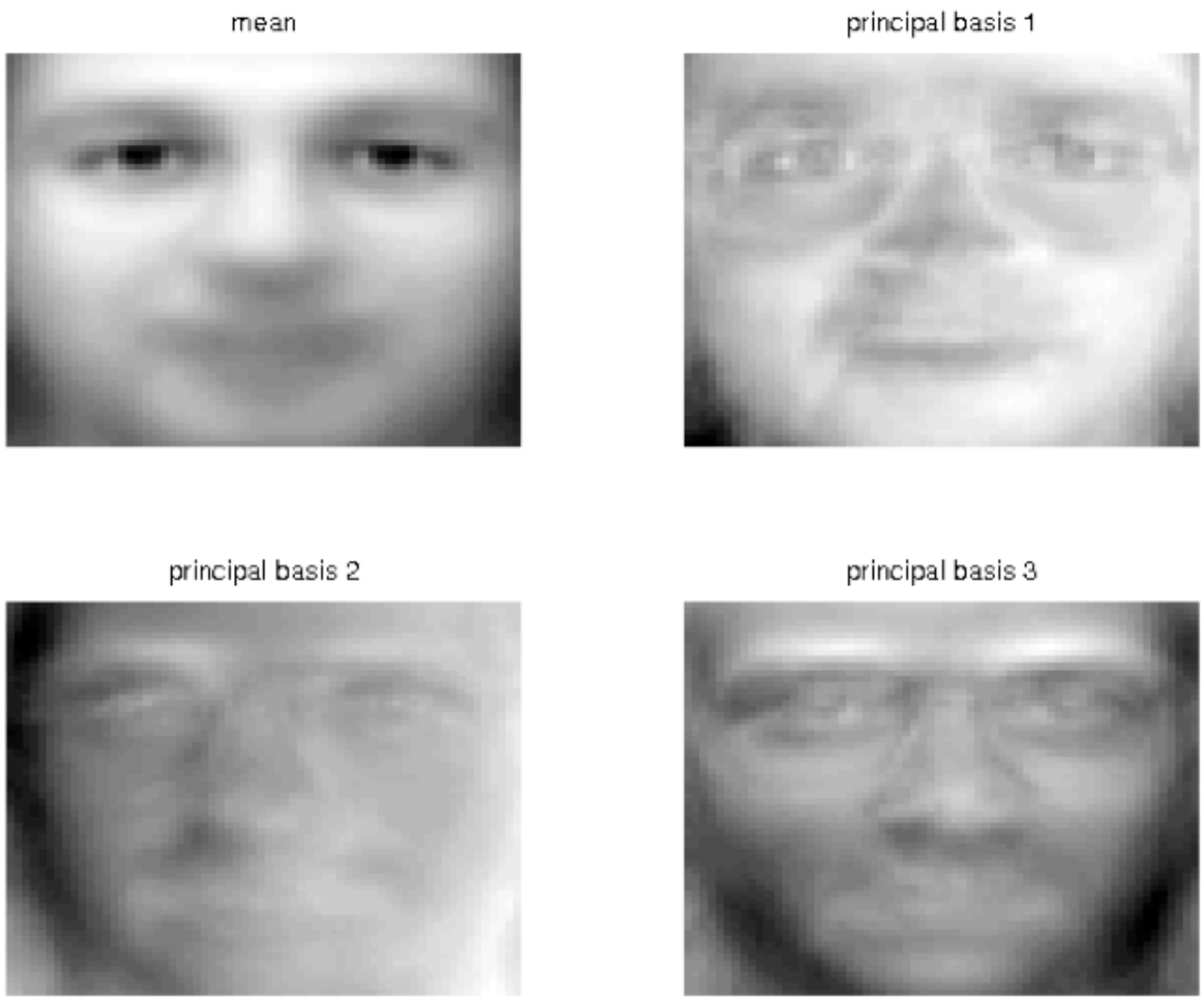


This is called Dimensionality Reduction!

Dimensionality Reduction Example: Eigenfaces

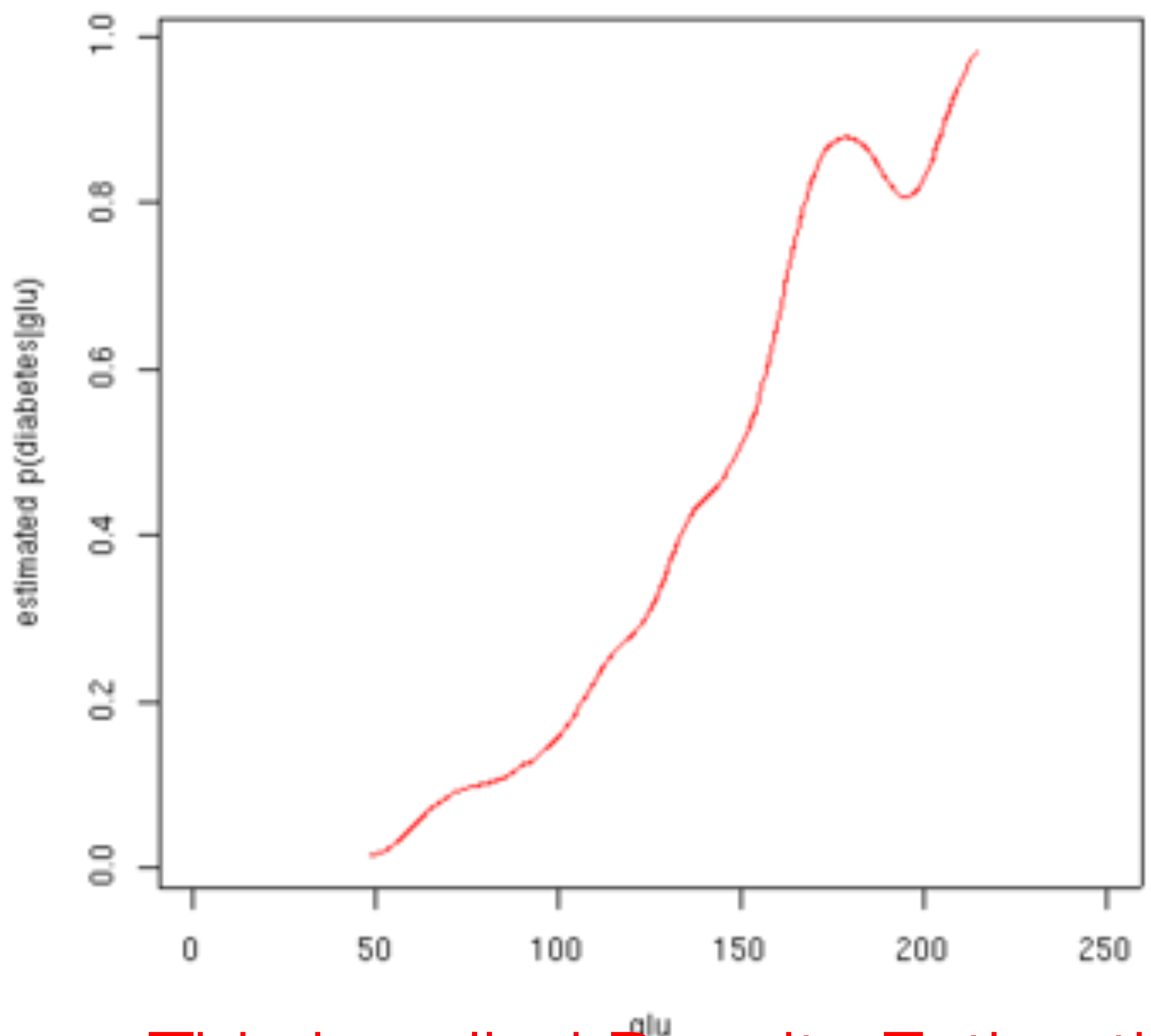
How many faces do you need to characterize these?

Example: Eigenfaces



Example: density of glu (plasma glucose concentration) for diabetes patients

Estimate relative occurance of a data point



This is called Density Estimation!

The bigger picture...

When we’re learning to see, nobody’s telling us what the right answers are – we just look. Every so often, your mother says ’that’s a dog,’ but that’s very little information. You’d be lucky if you got a few bits of information — even one bit per second — that way. The brain’s visual system has 10^{14} neural connections. And you only live for 10^{9} seconds. So it’s no use learning one bit per second. You need more like 10^{5} bits per second. And there’s only one place you can get that much information: from the input itself.

— Geoffrey Hinton, 1996

Types of Machine Learning

Machine Learning

predictive
(supervised)

descriptive
(unsupervised)

Active
(reinforcement learning)

That will be the main topic of the lecture!

How to attack a machine learning problem?

Machine learning problems essentially always are about two entities:

(i) *data model assumptions*:

- Understand your problem
- generate good features which make the problem easier
- determine the model class
- Pre-processing your data

(ii) *algorithms that can deal with* (i):

- Estimating the parameters of your model.

We are gonna do this for regression...

Content of this Lecture

- What is Machine Learning?
- Model-Selection
- Linear Regression
  - Frequentist Approach
  - Bayesian Approach

Important Questions

How does the data look like?

- Are you really learning a function?
- What data types do our outputs have?
- Outliers: Are there “data points in China”?

What is our model (relationship between inputs and outputs)?

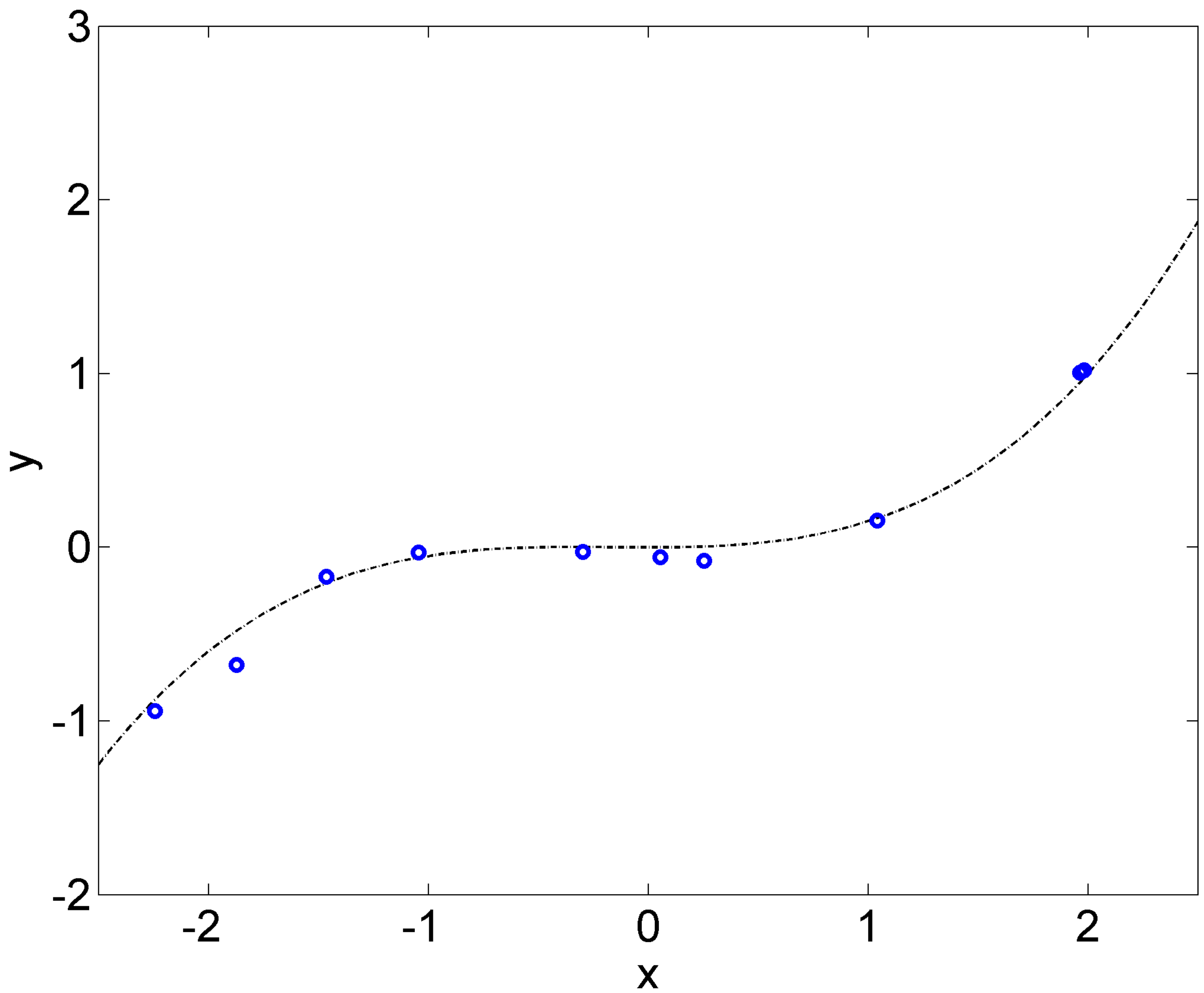
- Do you have features?
- What type of noise / What distribution models our outputs?
- Number of parameters?
- Is your model sufficiently rich?
- Is it robust to overfitting?

Important Questions

Requirements for the solution

- accurate
- efficient to obtain (computation/memory)
- interpretable

Example Problem: a data set



Task: Describe the outputs as a function of the inputs (**regression**)

Model Assumptions: Noise + Features

Additive Gaussian Noise:

$$y = \mathbf{f}_\theta(\mathbf{x}) + \epsilon \quad \text{with} \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

Equivalent Probabilistic Model

$$p(y|\boldsymbol{x}) = \mathcal{N}(y|f_\theta(\boldsymbol{x}), \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\frac{(y - f_\theta(\boldsymbol{x}))^2}{\sigma^2}\right)$$

Lets keep in simple: linear in Features

$$\mathbf{f}_\theta(\mathbf{x}) = \phi(\mathbf{x})^T \theta$$

Important Questions

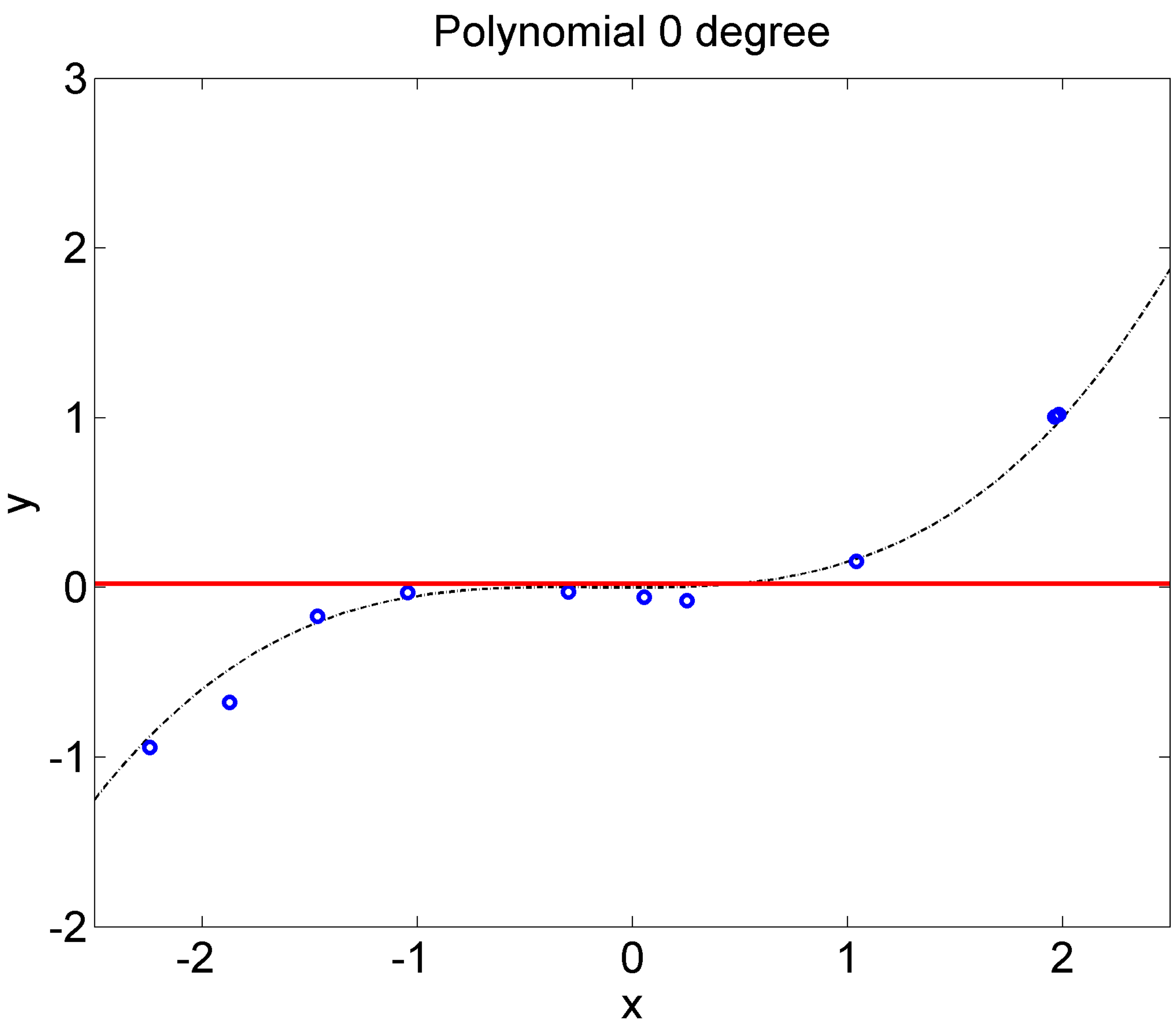
How does the data look like?

- What data types do our outputs have? $y_q \in \mathbb{R}$
- Outliers: Are there “data points in China”? NO
- Are you really learning a function? YES

What is our model? $y = \phi(\boldsymbol{x})^T \boldsymbol{\theta} + \epsilon$

- Do you have features? $\phi(\boldsymbol{x})$
- What type of noise / What distribution models our outputs? $\epsilon \sim \mathcal{N}(0, \sigma^2)$
- Number of parameters?
- Is your model sufficiently rich?
- Is it robust to overfitting?

Let us fit our model ...



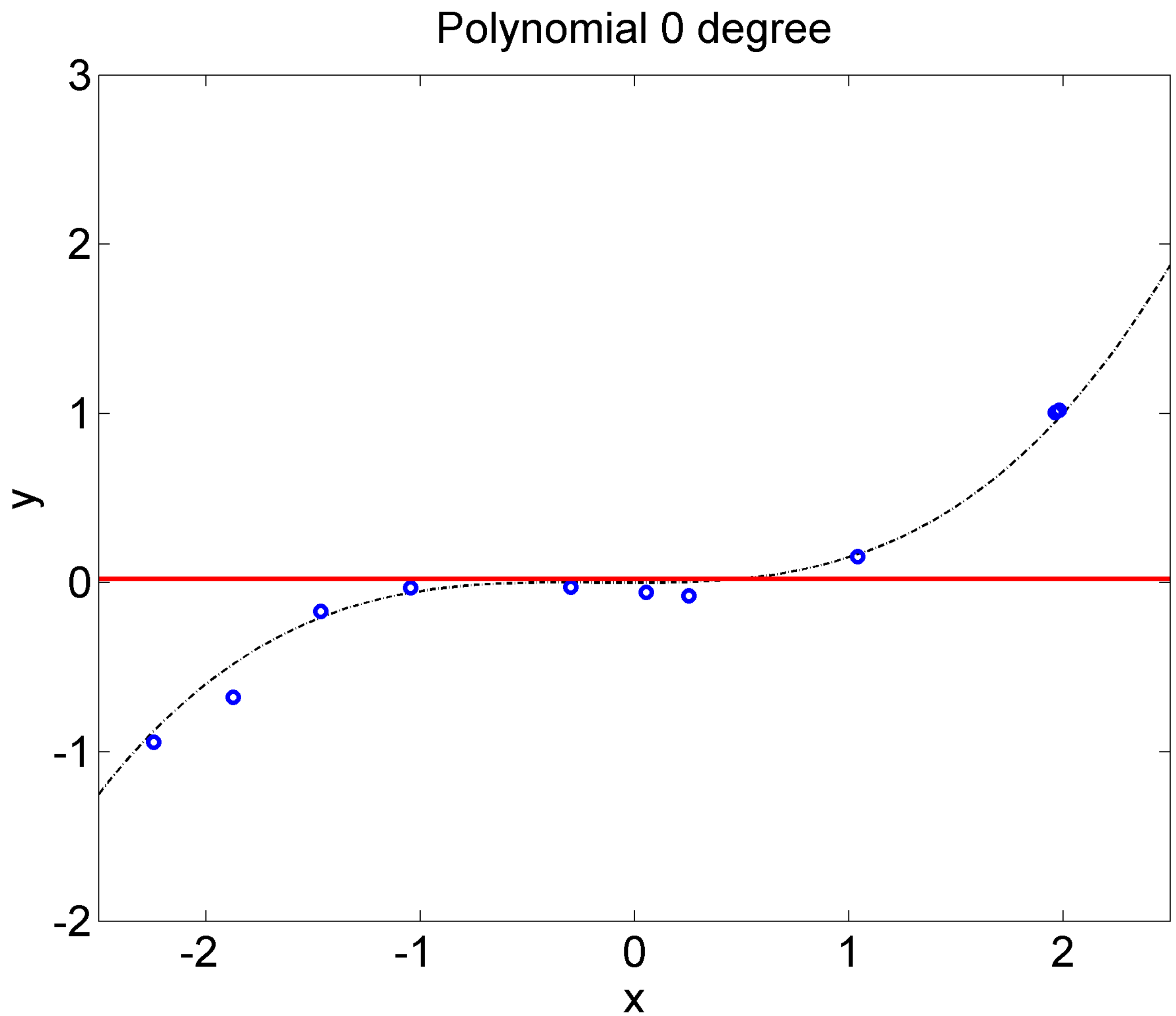
We need to answer:

- How many parameters?
- Is your model sufficiently rich?
- Is it robust to overfitting?

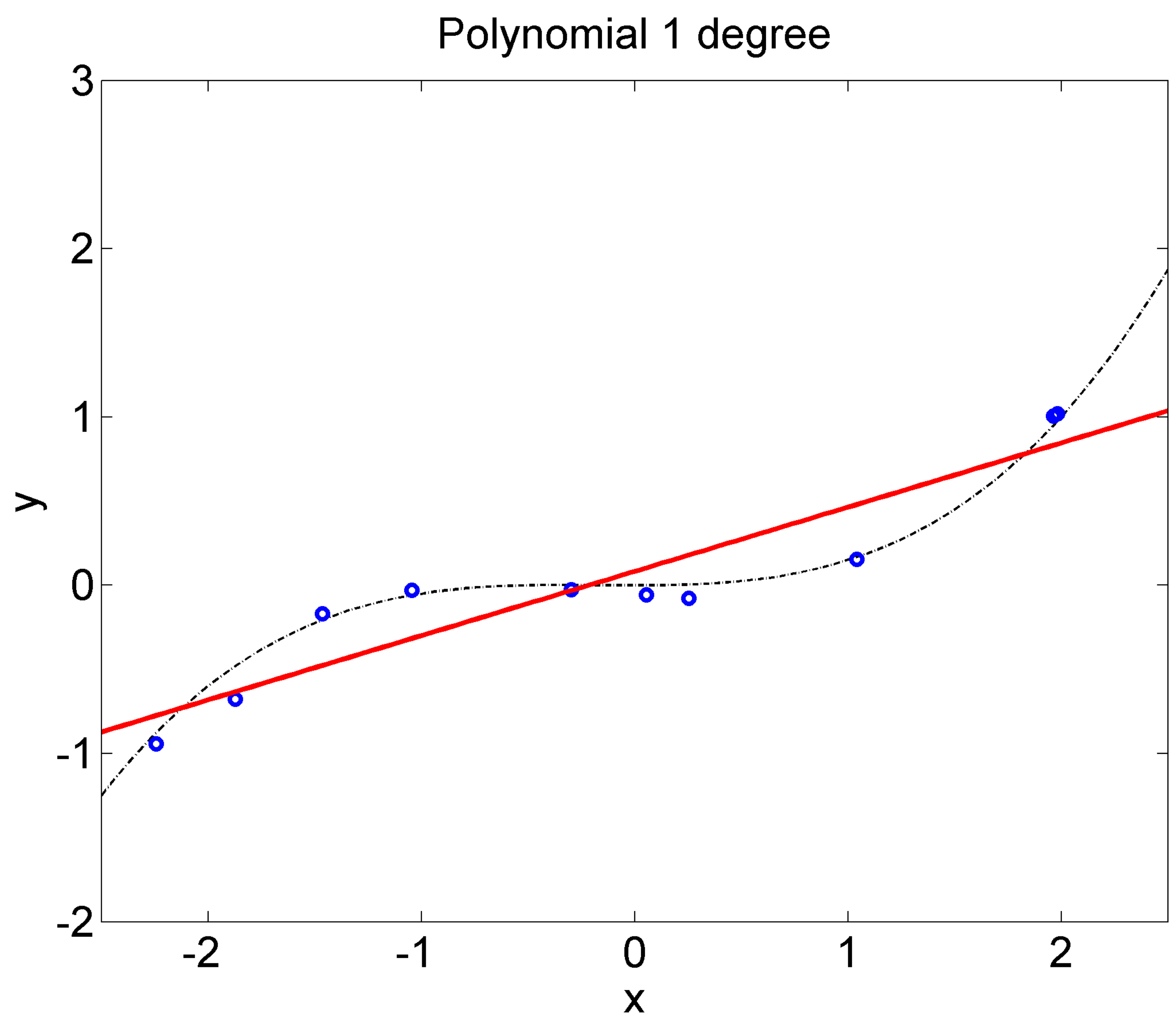
We assume a model class: polynomials of degree n

$$y = \phi(x)^T \theta + \epsilon = [1, x, x^2, x^3, \ldots, x^n]^T \theta + \epsilon$$

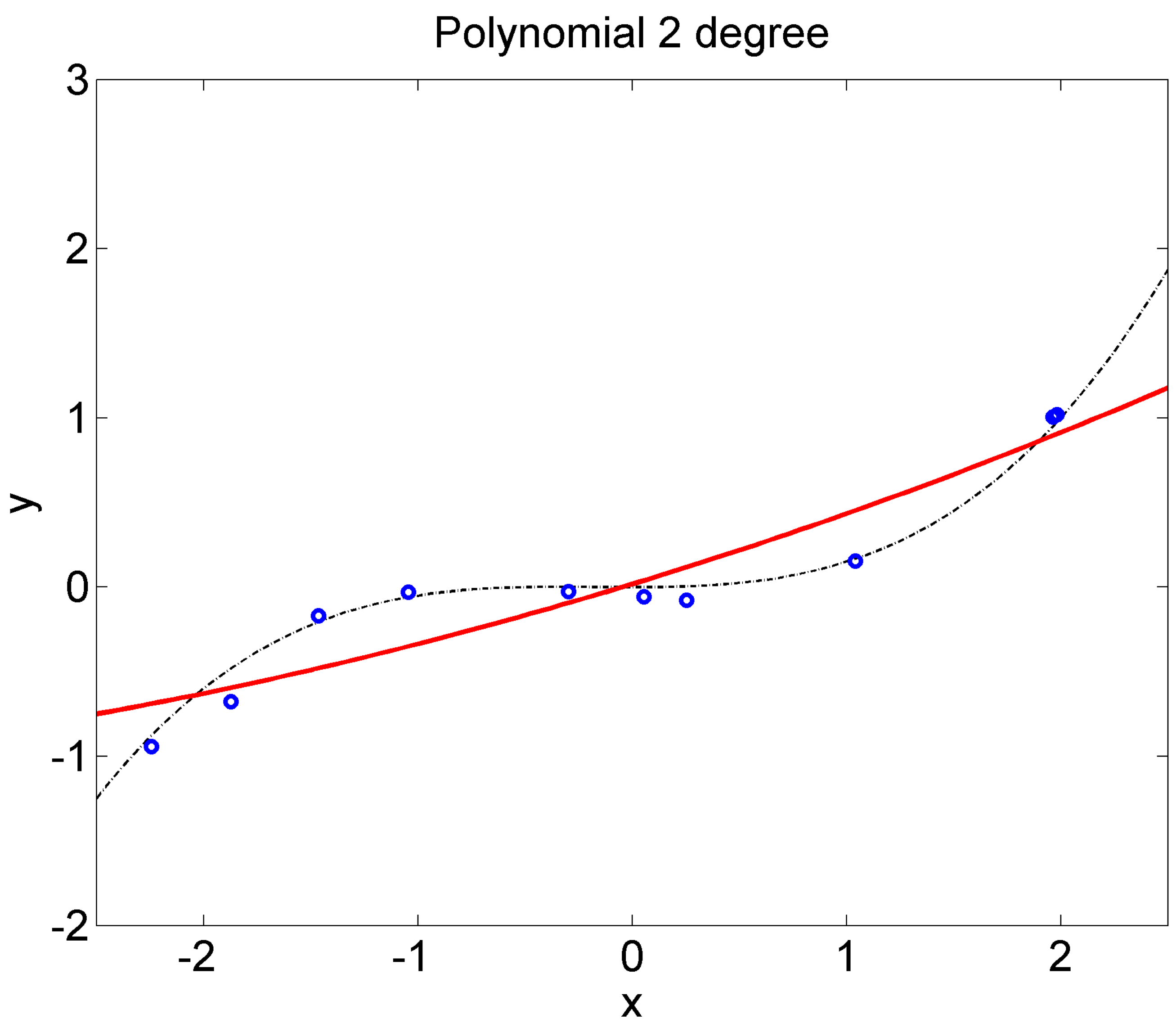
Fitting an Easy Model: n=0

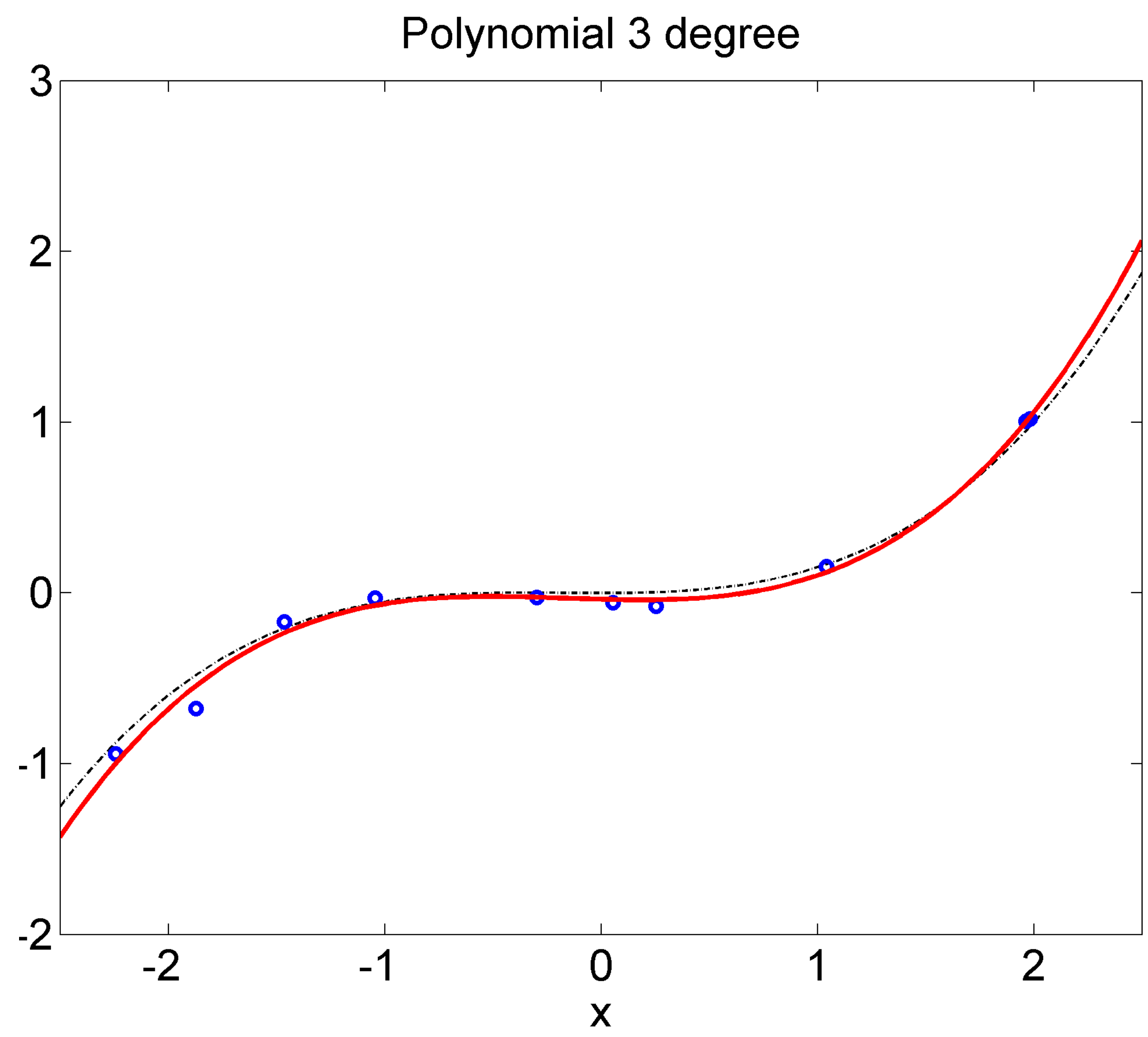


Add a Feature: n=1

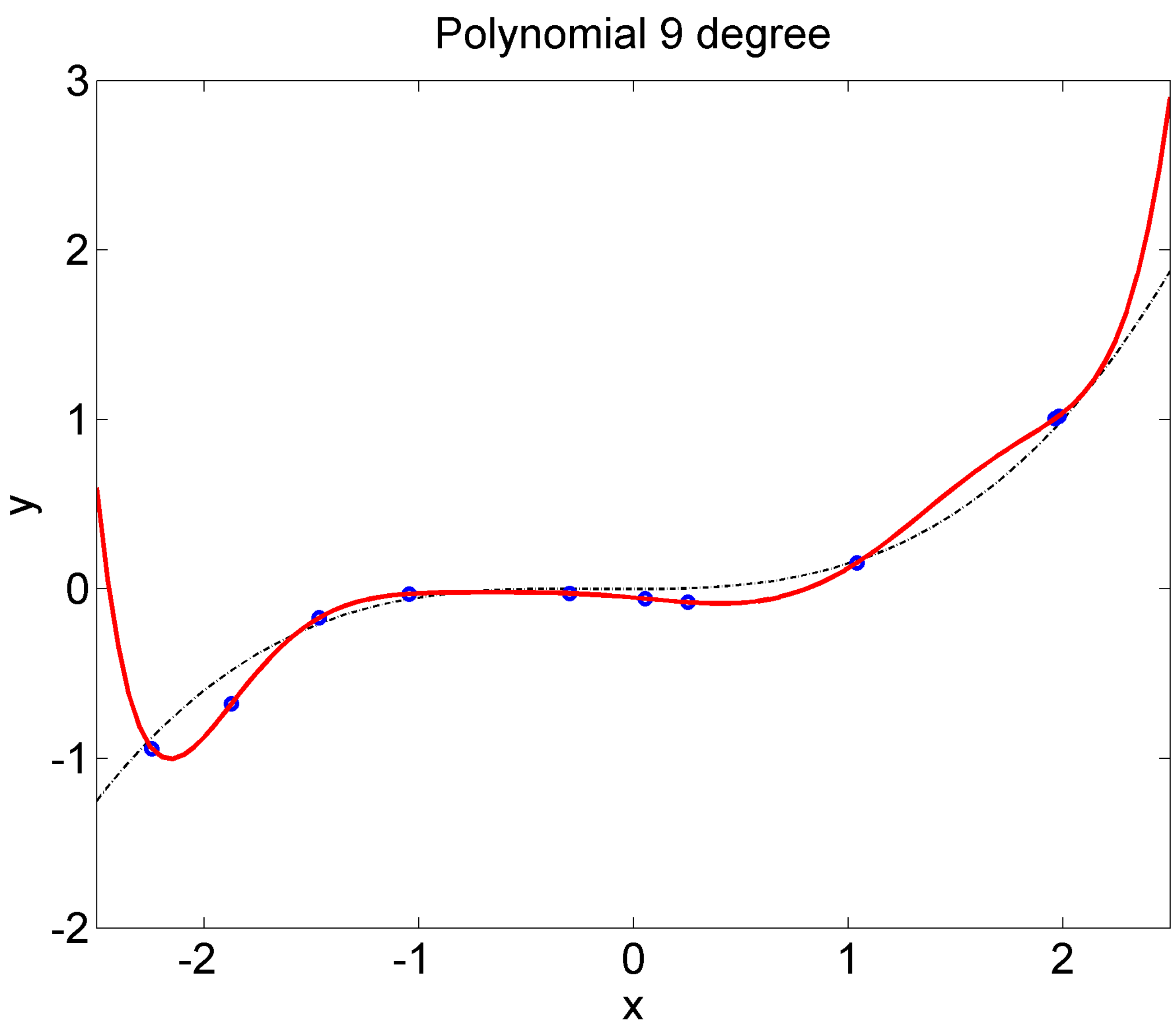


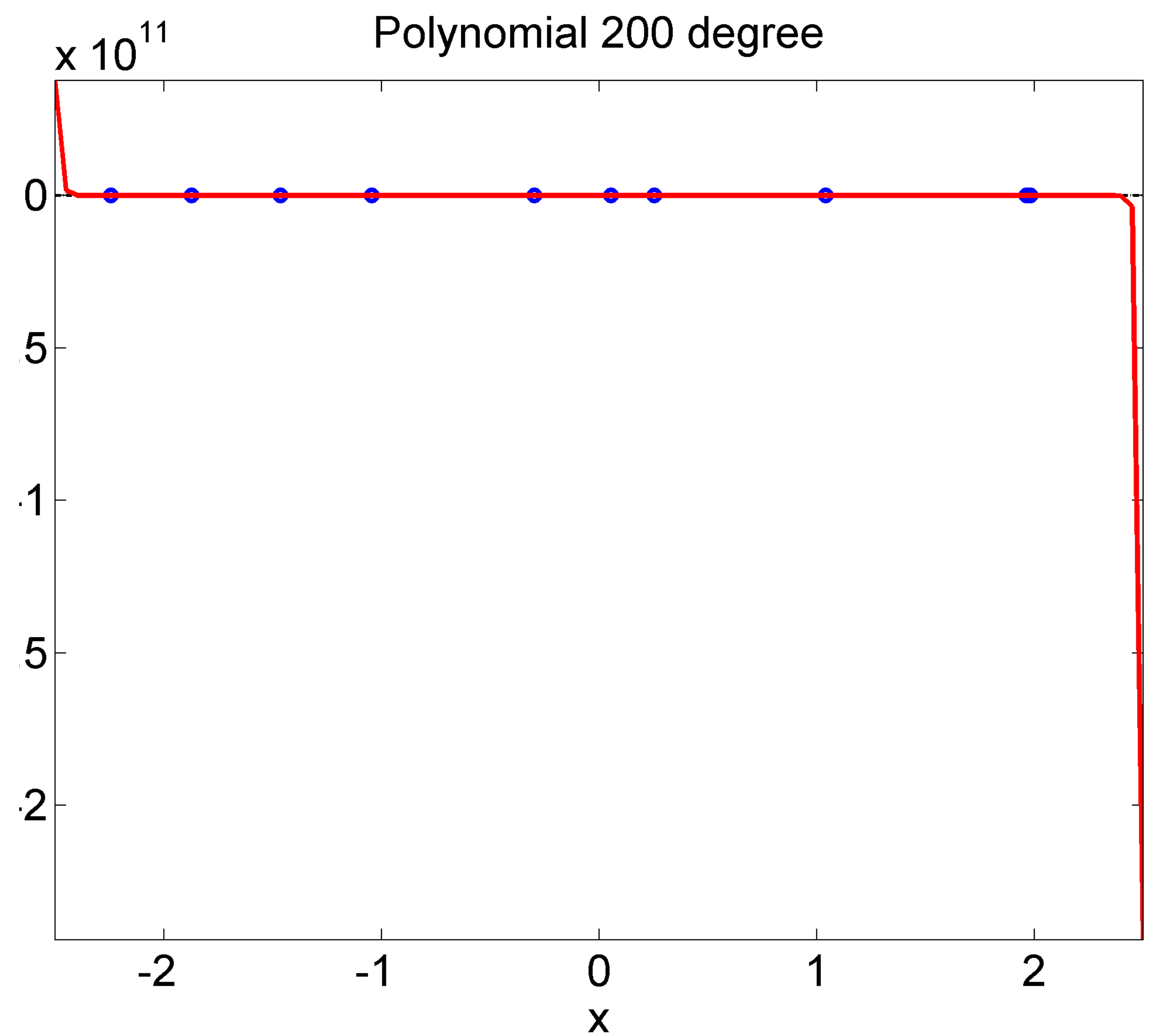
More features...



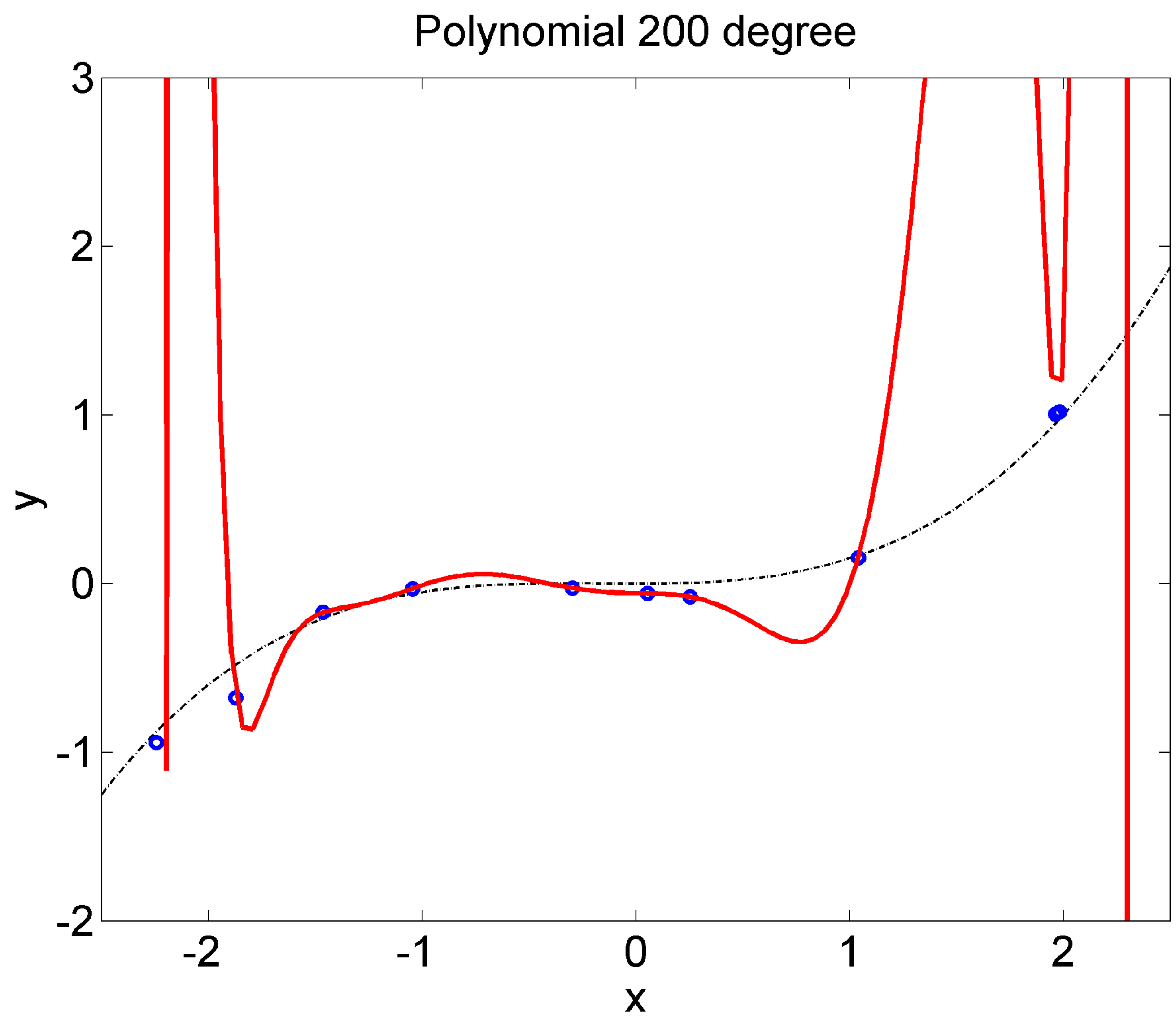


More features...





More features: n=200 (zoomed in)



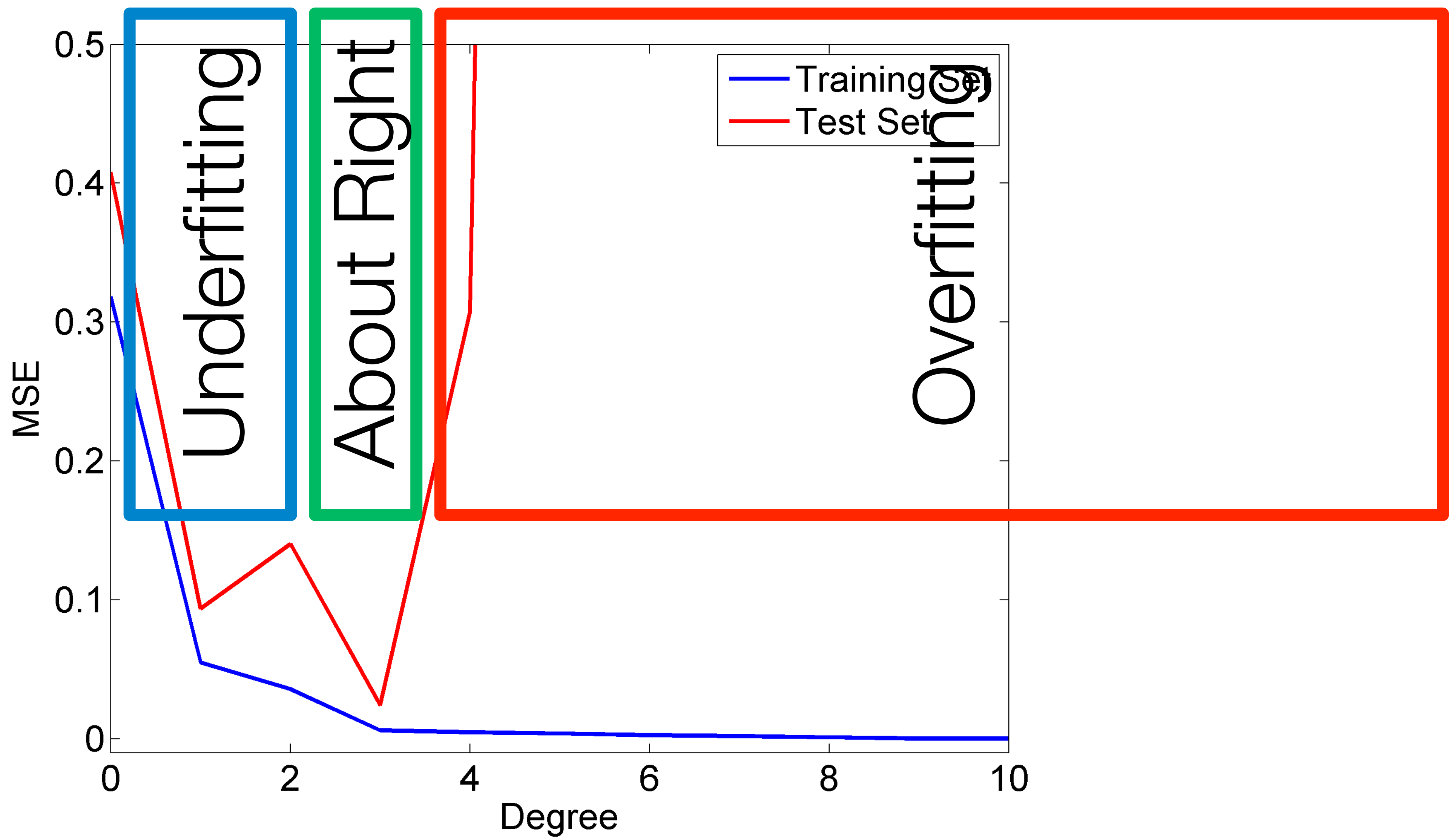
overfitting and numerical problems

Prominent example of overfitting...

Distinguish between Soviet (left) and US (right) tanks

DARPA Neural Network Study (1988-89), AFCEA International Press

Test Error vs Training Error



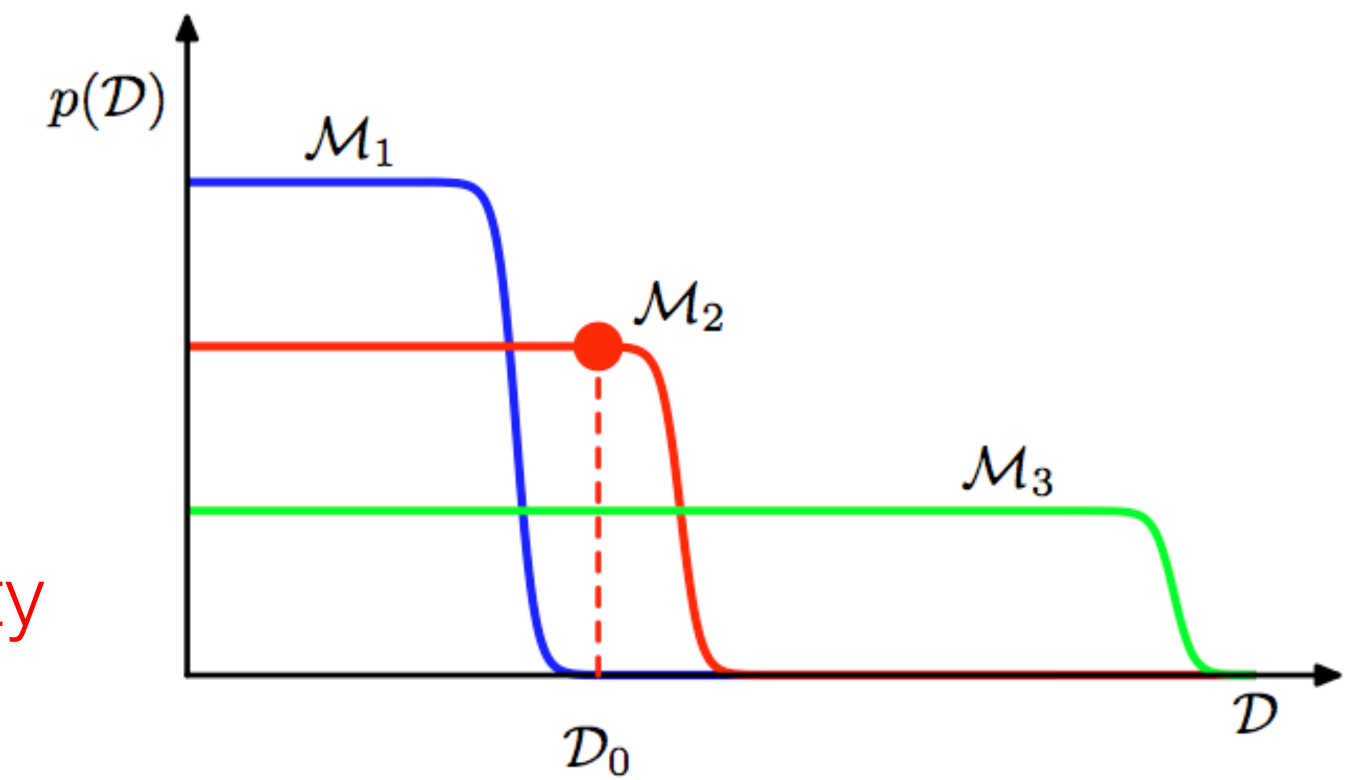
Does a small training error lead to a good model ??

NO ! We need to do model selection

Occam's Razor and Model Selection

Model Selection: How can we…

- choose number of features/parameters?
- choose type of features?
- prevent overfitting?



Some insights:

Always choose the model that fits the data and has the smallest model complexity

called occam's razor

Bias-Variance Tradeoff

$$\text{Expected Total Error} = \text{Bias}^2 + \text{Variance}$$

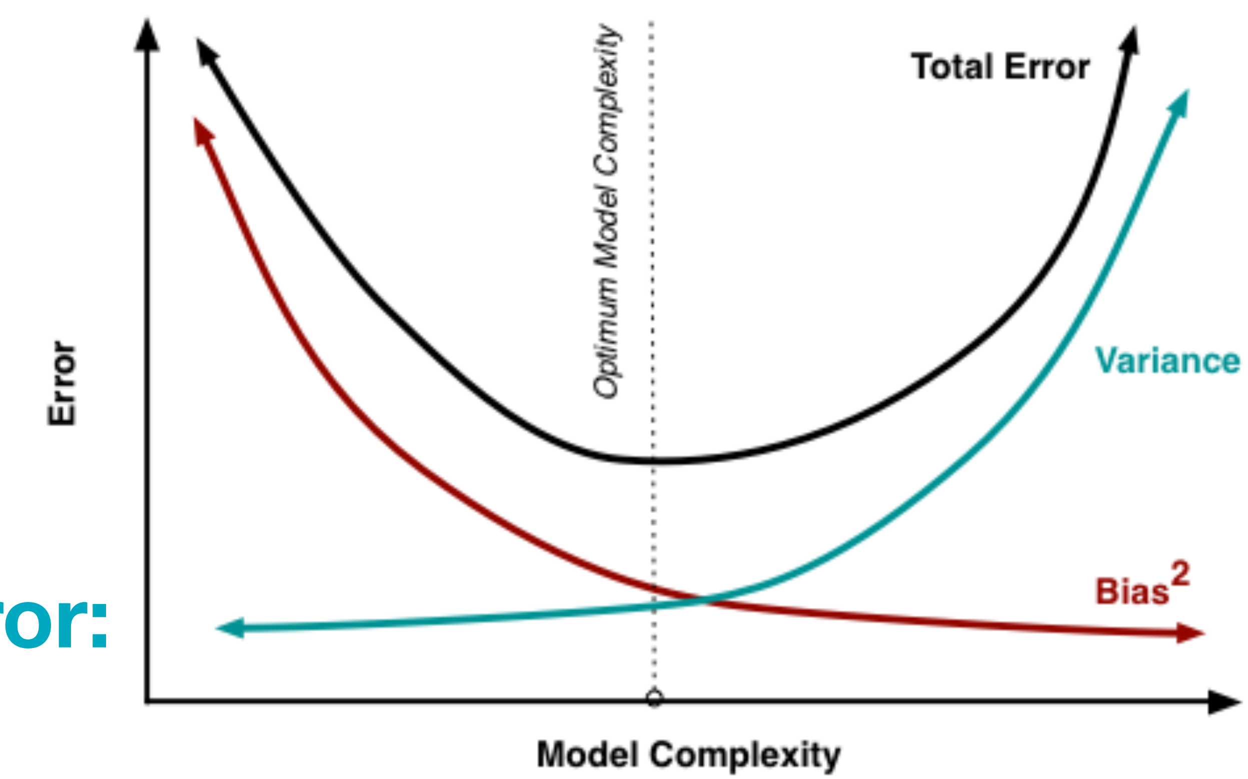
Typically, you can not minimize both!

- **Bias / Structure Error:**

  Error because our model can not do better

- **Variance / Approximation Error:**

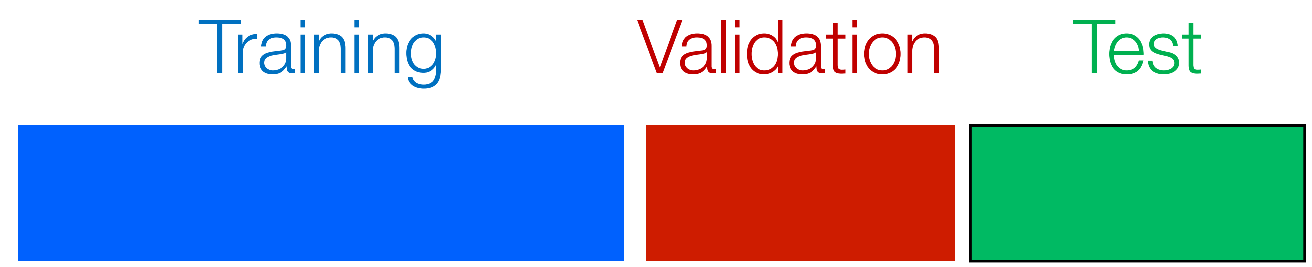
  Error because we estimate parameters on a limited data set



How do choose the model?

Goal: Find a good model $\mathcal{M}$ (e.g., good set of features)

Split the dataset into:



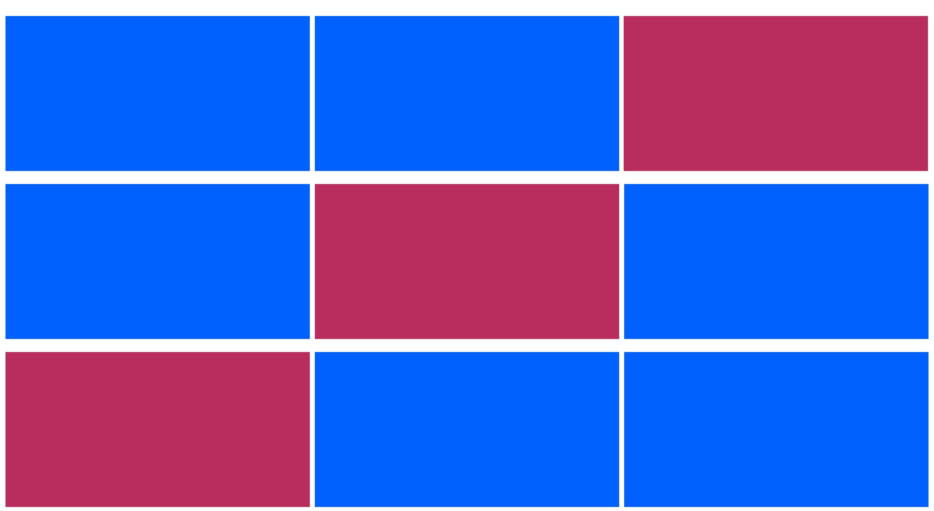
- Training Set: Fit Parameters
- Validation Set: Choose model class or single parameters
- Test Set: Estimate prediction error of trained model

Error needs to be estimated on independent set!

Model Selection:
K-fold Cross Validation

- Partition data into K sets, use K-1 as **training set** and 1 set as **validation set**

- For all possible ways of partitioning compute the **validation error** ***J*** ➡ computationally expensive!!



$$J(p_1, \mathcal{M}_i)$$

$$J(p_2, \mathcal{M}_i)$$

$$J(p_3, \mathcal{M}_i)$$

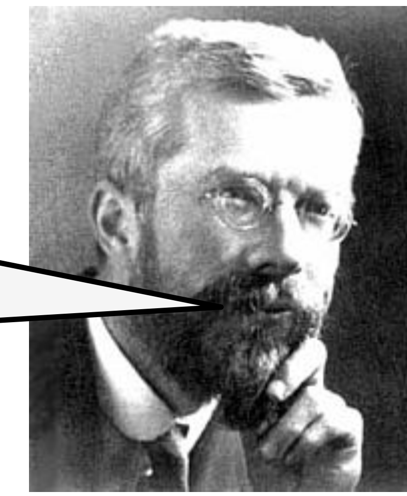
- Choose model $\mathcal{M}_i$ with **smallest average validation error**

Content of this Lecture

- What is Machine Learning?
- Model-Selection
- Linear Regression
  - Frequentist Approach
  - Bayesian Approach

How to find the parameters $\boldsymbol{\theta}$?

Fisher



Frequentist:

Lets minimize the error of our prediction

$$\boldsymbol{\theta}^* = \text{argmin}_{\boldsymbol{\theta}} \sum_{i=1}^{N} (f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)^2$$

Objective is defined by minimizing a certain cost function

Frequentist view: Least Squares

The classical cost function is the one of **least-squares**

$$J = \frac{1}{2}\sum_{i=1}^{N}(y_i - \mathbf{f}_\theta(\mathbf{x}_i))^2. \qquad y_i = \phi(\boldsymbol{x}_i)^T\boldsymbol{\theta} + \epsilon$$

Using

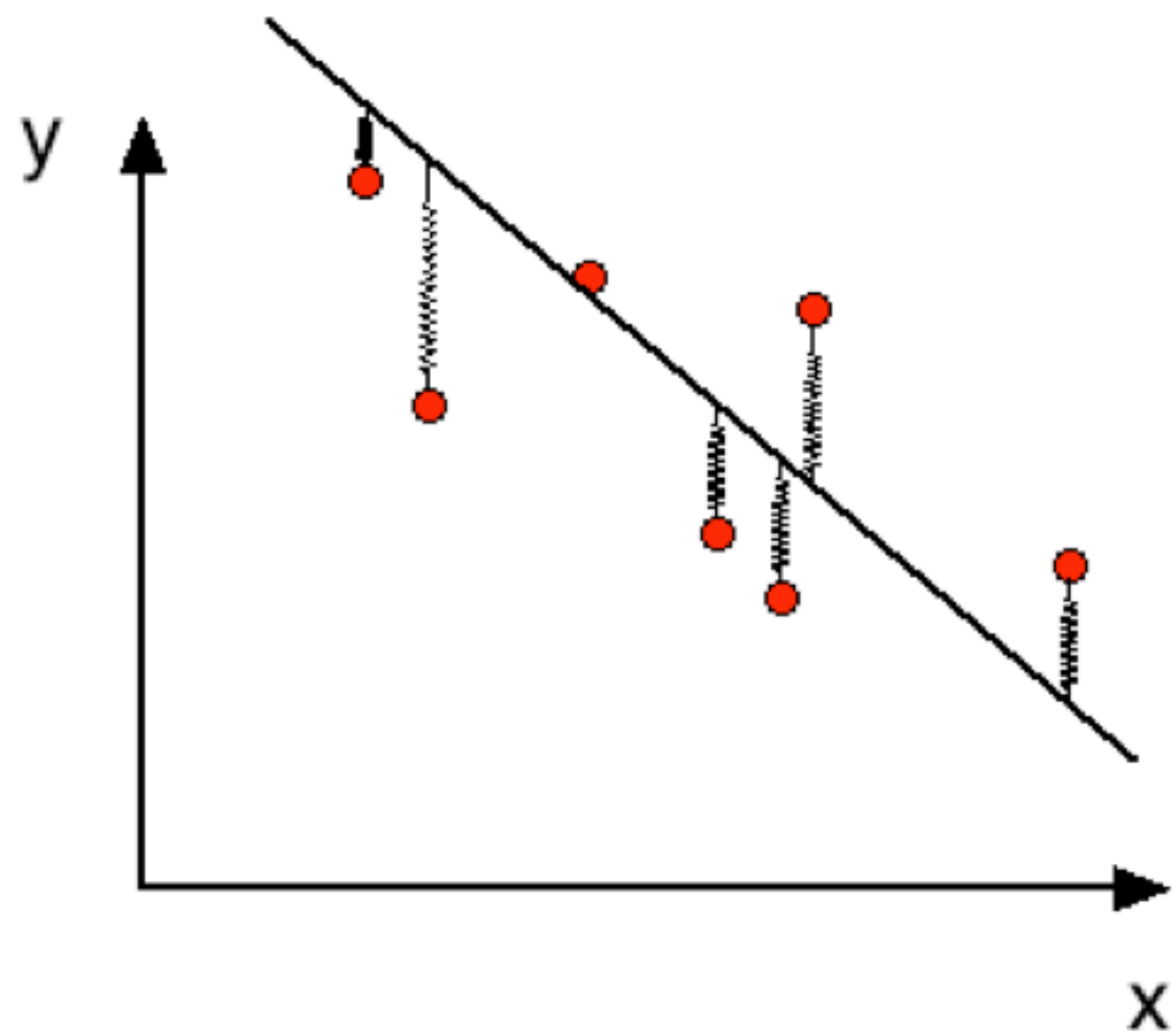
$$\begin{aligned} \boldsymbol{\Phi} &= \left[\ \phi(\mathbf{x}_1),\ \ \phi(\mathbf{x}_2),\ \ \phi(\mathbf{x}_3),\ \ \ldots,\ \ \phi(\mathbf{x}_n)\ \right]^T, \\ \mathbf{Y} &= \left[\ y_1,\ \ y_2,\ \ y_3,\ \ \ldots,\ \ y_n\ \right]^T. \end{aligned}$$

we can rewrite it as $$J = \frac{1}{2}(Y - \Phi\theta)^T(Y - \Phi\theta).$$ Scalar Product

and solve it $$\theta = (\Phi^T\Phi)^{-1}\Phi^T Y$$

Least Squares solution contains left pseudo-inverse

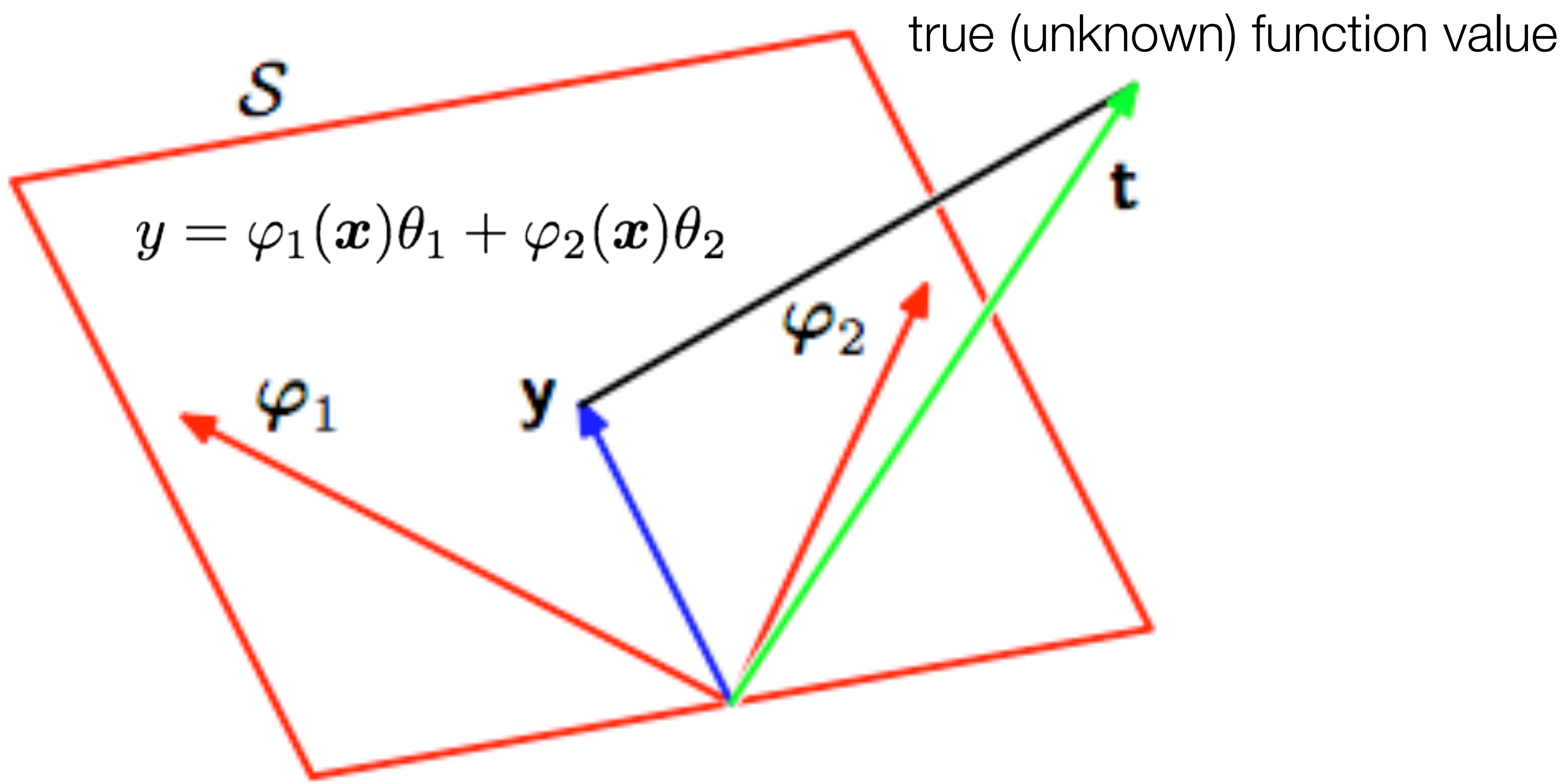
Physical Interpretation



Energy of springs ~ squared lengths

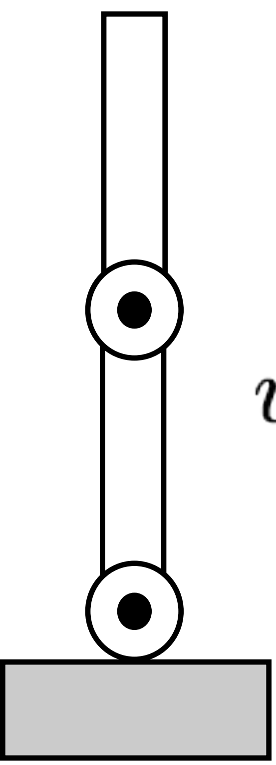
➡ minimize energy of system

Geometric Interpretation



Minimize projection error ➡ orthogonal projection

Robotics Example: Rigid-Body Dynamics
Known Features



$$
\begin{aligned}
u_1 &= [m_1 l_{g1}^2 + J_1 + m_2(l_1^2 + l_{g2}^2 + 2 l_1 l_{g2}\cos\theta_2) + J_2]\ddot{\theta}_1 \\
&+ [m_2(l_{g2}^2 + l_1 l_2 \cos\theta_2) + J_2]\ddot{\theta}_2
\end{aligned}
$$

Inertial Forces

$$- 2 m_2 l_1 l_{g2} \dot{\theta}_1 \dot{\theta}_2 \sin\theta_2$$

Coriolis Forces

$$- 2 m_2 l_1 l_{g2} \dot{\theta}_1^2 \sin\theta_2$$

Centripetal Forces

$$+ m_1 g l_{g1}\cos\theta_1 + m_2 g (l_1 \cos\theta_1 + l_{g2}\cos(\theta_1 + \theta_2))$$

Gravity

$$
\begin{aligned}
u_2 &= [m_2(l_{g2}^2 + l_1 l_{g2}\cos\theta_2) + J_2]\ddot{\theta}_1 \\
&+ (m_2 l_{g2}^2 + J_2)\ddot{\theta}_2
\end{aligned}
$$

Inertial Forces

$$- m_2 l_1 l_{g2} \dot{\theta}_1^2 \sin\theta_2$$

Centripetal Forces

$$+ m_2 g l_{g2}\cos(\theta_1 + \theta_2)$$

Gravity

Robotics Example: Rigid-Body Dynamics

We realize that rigid body dynamics is **linear in the parameters**

We can rewrite it as

$$\boldsymbol{u} = \boldsymbol{\phi}(\theta, \dot{\theta}, \ddot{\theta})^T \boldsymbol{\psi}$$

accelerations, velocities, sin and cos terms

masses, lengths, inertia, ...

For finding the parameters we can apply even the first machine learning method that comes to mind: Least-Squares Regression

Cost Function II: Ridge Regression

We punish the magnitude of the parameters

➡ Controls model complexity

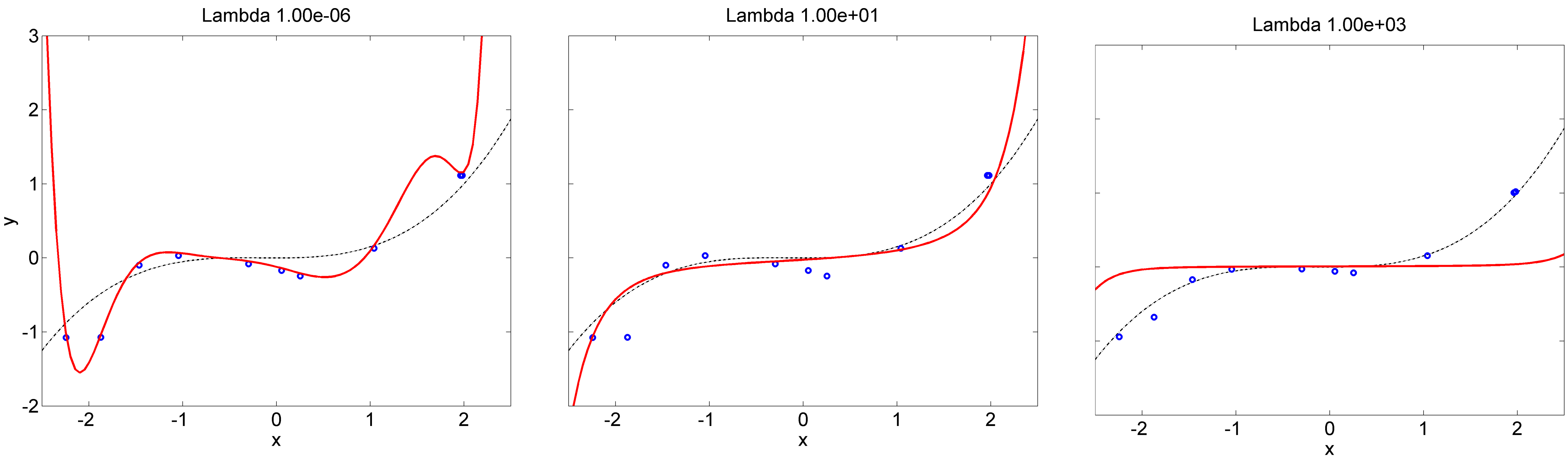
$$J_{\mathrm{RR}} = (\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta})^T(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta}) + \boldsymbol{\theta}^T\boldsymbol{W}\boldsymbol{\theta}$$

This yields **ridge regression** $\theta = (\Phi^T\Phi + \mathbf{W})^{-1}\Phi^T Y$

with $\mathbf{W} = \lambda\mathbf{I}$, where λ is called regularizer

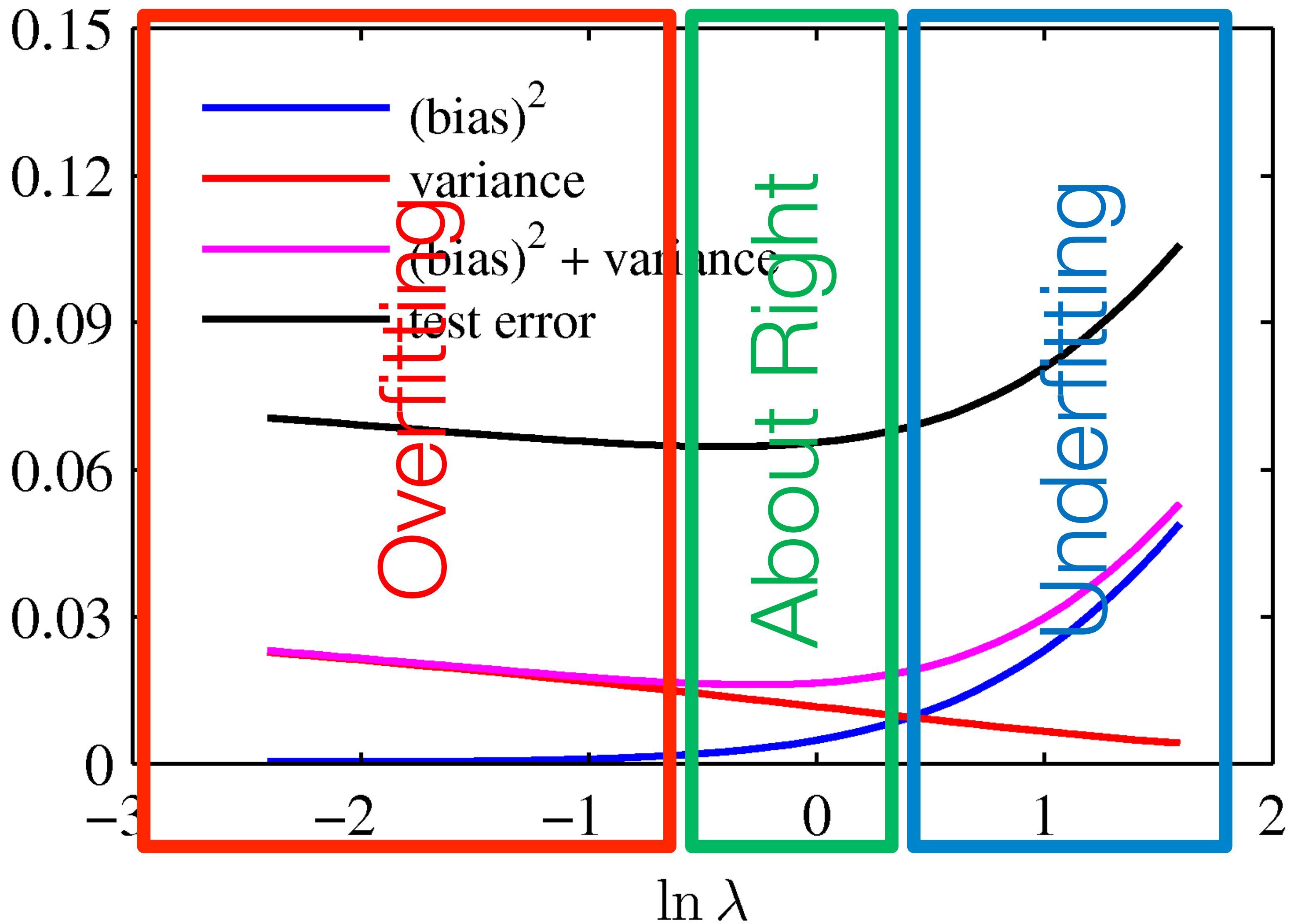
Numerically, this is much more stable

Ridge regression: n=15



Influence of the regularization constant

MAP: Back to the Overfitting Problem



We can also scale the model complexity with the regularization parameter!

Smaller lambda ⇒ higher model complexity

Content of this Lecture

- What is Machine Learning?
- Model-Selection
- Linear Regression
  - Frequentist Approach
  - Bayesian Approach

Maximum-Likelihood (ML) estimate

Lets look at the problem from a probabilistic perspective…

Alternatively, we can maximize the **likelihood of the parameters**:

$$\arg\max_{\theta} p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) = \arg\max_{\theta} \prod_{i=1}^{N} p(y_i|\boldsymbol{x}_i,\boldsymbol{\theta})$$

That's hard!

Do the 'log-trick':

$$J_{\mathrm{ML}} = \arg\max_{\theta} \log p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) = \arg\max_{\theta} \sum_{i=1}^{N} \log p(y_i|\boldsymbol{x}_i,\boldsymbol{\theta})$$

That's easy!

$$= \arg\min_{\theta} \cancel{\frac{1}{2\sigma^2}} \sum_{i=1}^{N} (y_i - f_{\theta}(\boldsymbol{x}_i))^2$$

$$\Rightarrow \boldsymbol{\theta}^*_{\mathrm{LS}} = \boldsymbol{\theta}^*_{\mathrm{ML}}$$

Least Squares Solution is equivalent to ML solution with Gaussian noise!!

Maximum a posteriori (MAP) estimate

Put a prior on our parameters

- E.g., $\boldsymbol{\theta}$ should be small: $p(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, \boldsymbol{W}^{-1})$

Find parameters that maximize the posterior

$$p(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{X}) = \frac{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{y}|\boldsymbol{X})} \propto \boxed{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})}\boxed{p(\boldsymbol{\theta})}$$

Do the 'log trick' again:

$$\begin{aligned}\arg\max_{\theta} p(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{X}) &= \arg\max_{\theta} p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})p(\boldsymbol{\theta}) = \arg\max_{\theta} \log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) \\ &= \arg\min_{\theta} \boxed{-\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})}\boxed{-\log p(\boldsymbol{\theta})} =: J_{\text{MAP}}\end{aligned}$$

Maximum a posteriori (MAP) estimate

The prior is just additive costs…

$$J_{\mathrm{MAP}} = -\log p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) - \log p(\boldsymbol{\theta})$$

Lets put in **our Model:**

$$p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) = \prod_i p(y_i|\boldsymbol{x}_i,\boldsymbol{\theta}) = \prod_i \mathcal{N}(y_i|\boldsymbol{\theta}^T\phi(\boldsymbol{x}_i),\sigma^2) \qquad p(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0},\boldsymbol{W}^{-1})$$

$$J_{\mathrm{MAP}} = \boxed{\frac{1}{2}\sum_i \frac{(y_i - \phi^T(\boldsymbol{x}_i)\boldsymbol{\theta})^2}{\sigma^2}} + \boxed{\frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{W}\boldsymbol{\theta}}$$

$$= \boxed{\frac{1}{2\sigma^2}(\boldsymbol{y}-\Phi(\boldsymbol{X})\boldsymbol{\theta})^T(\boldsymbol{y}-\Phi(\boldsymbol{X})\boldsymbol{\theta})} + \boxed{\frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{W}\boldsymbol{\theta}} = \frac{1}{\sigma^2}J_{\mathrm{RR}}$$

$$\Rightarrow \quad \boldsymbol{\theta}^*_{\mathrm{RR}} = \boldsymbol{\theta}^*_{\mathrm{MAP}}$$

Ridge Regression is equivalent to MAP estimate with Gaussian prior

Predictions with the Model

We found an amazing parameter set $\boldsymbol{\theta}^*$ (e.g., ML, MAP)

Let's do **predictions**!

parameter estimate

$$p(y_*|\boldsymbol{x}_*, \boldsymbol{\theta}^*) = \mathcal{N}\big(y_*|\mu(\boldsymbol{x}_*), \sigma^2(\boldsymbol{x}_*)\big)$$

pred. function value

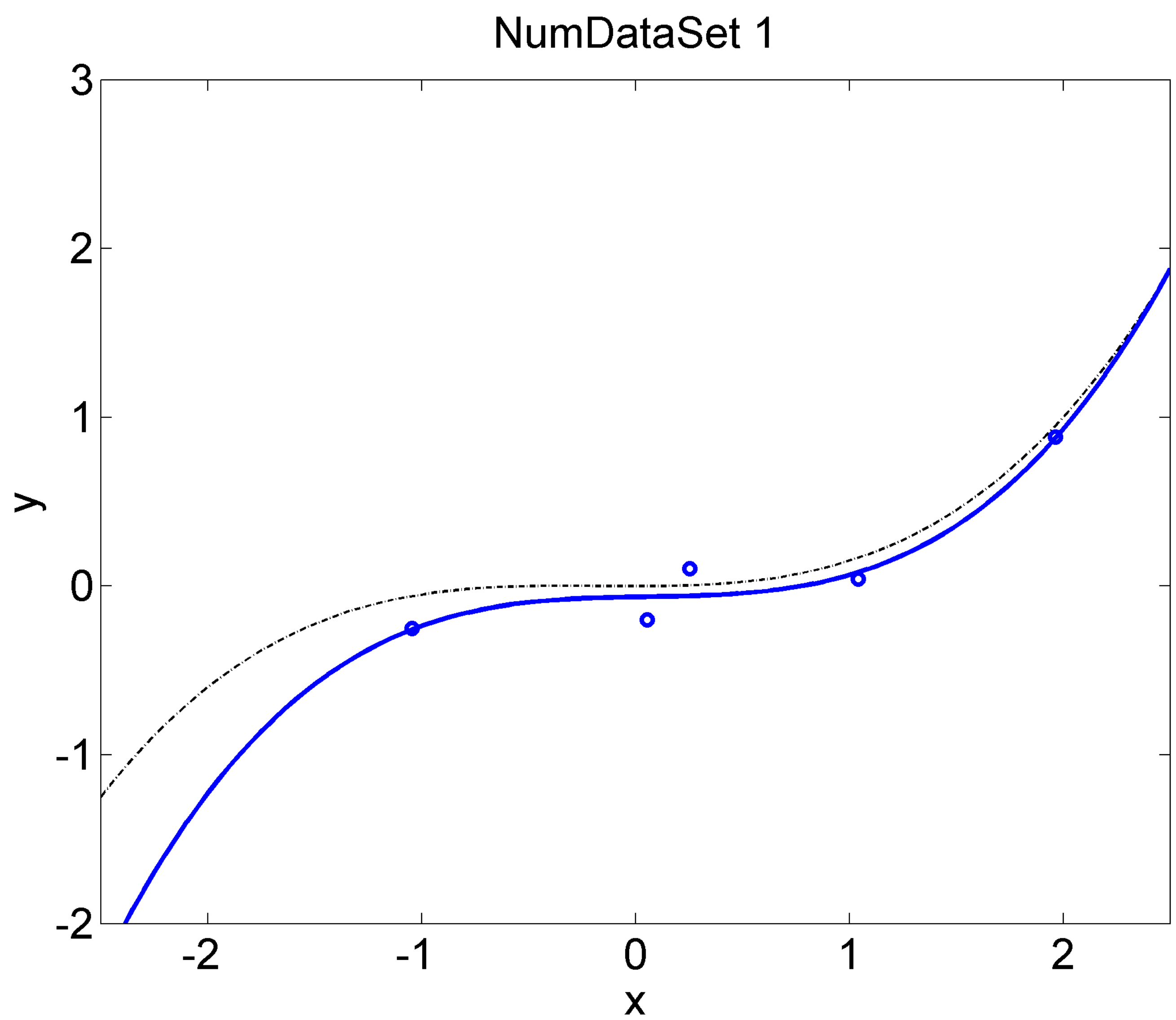
test input

Predictive mean: $\mu(\boldsymbol{x}_*) = \boldsymbol{\phi}^T(\boldsymbol{x}_*)\boldsymbol{\theta}^*$

Predictive variance: $\sigma^2(\boldsymbol{x}_*) = \sigma^2$

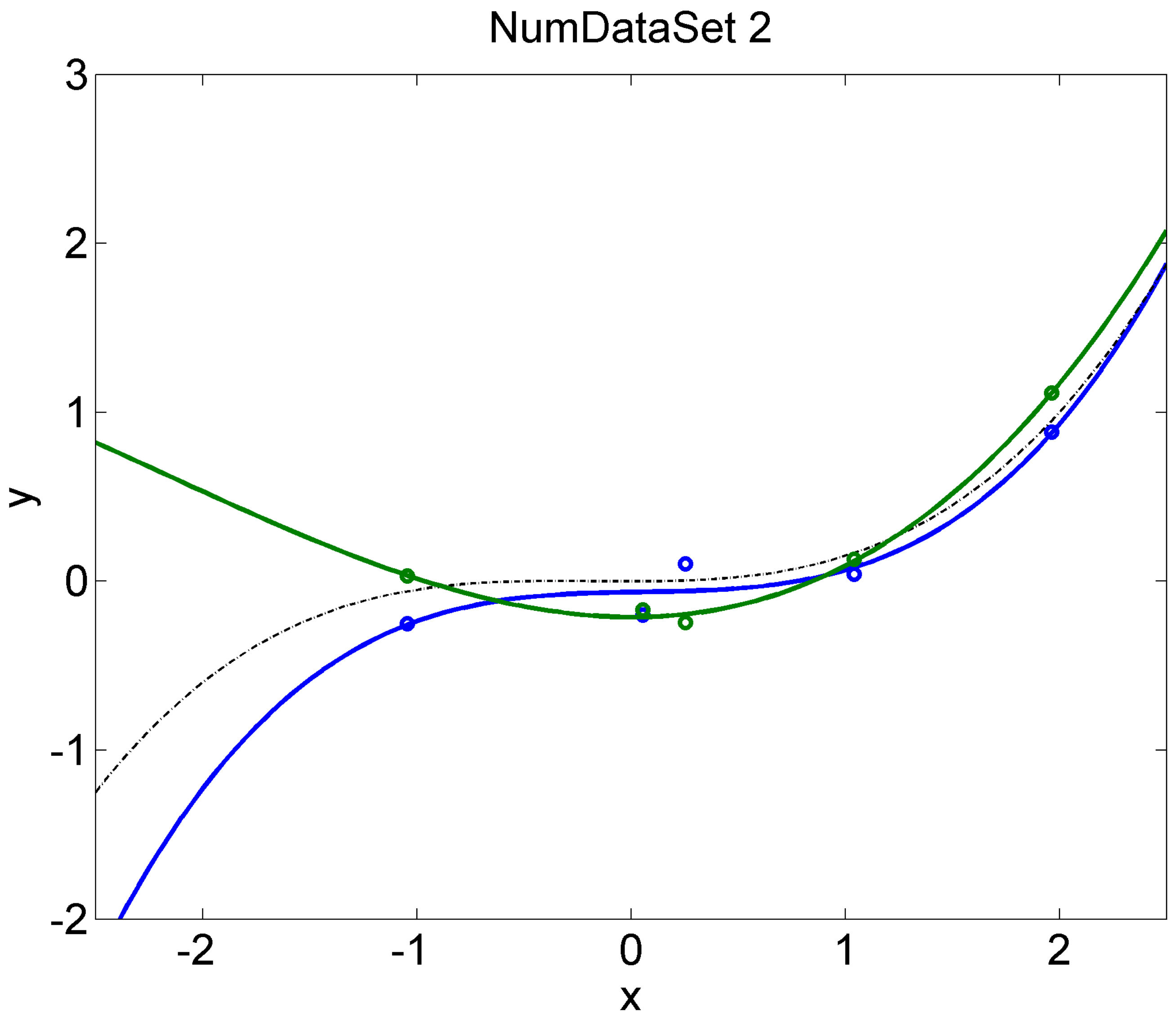
Comparing different data sets…

… with same input data, but different output values (due to noise):



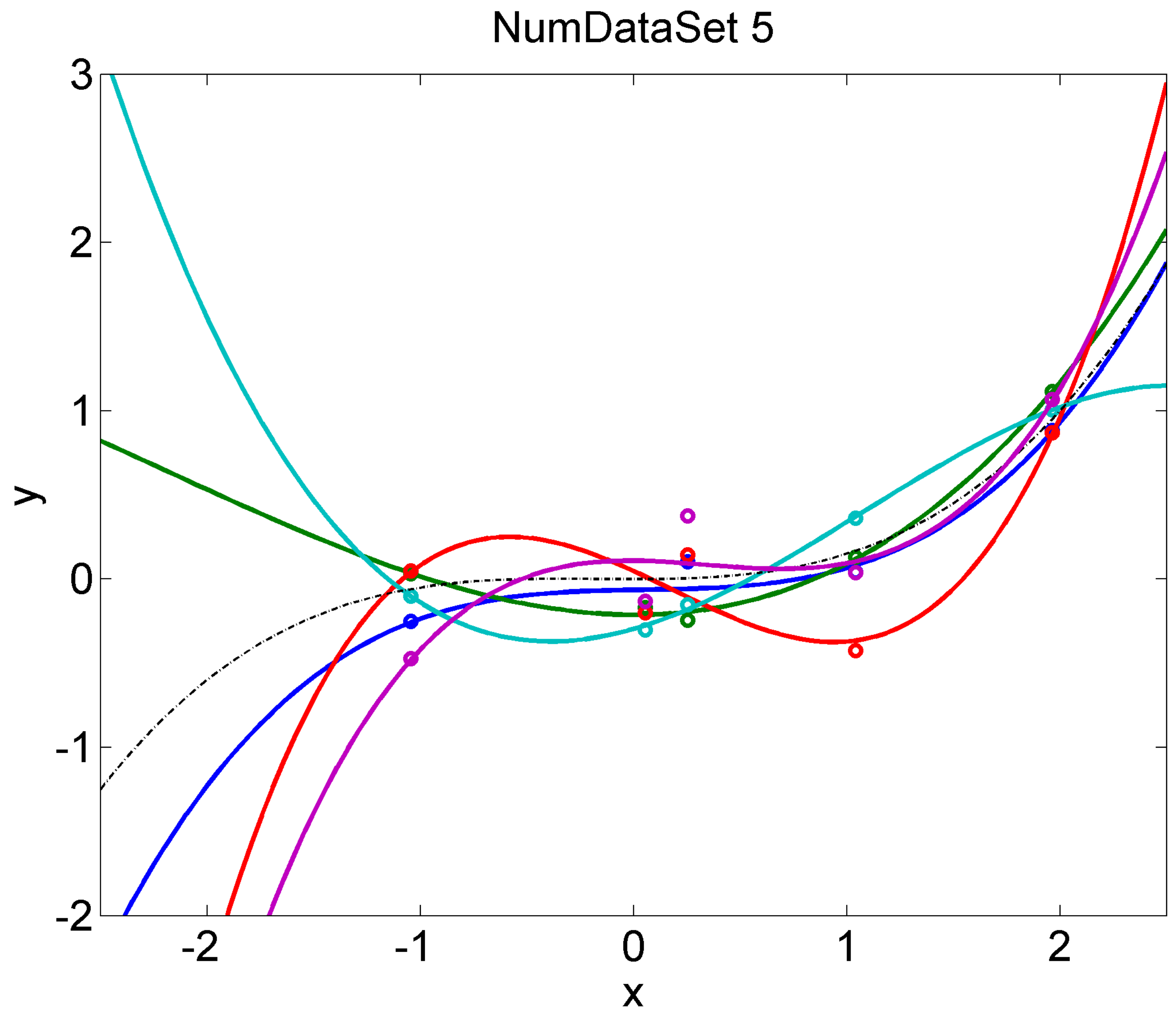
Comparing different data sets…

… with same input data, but different output values (due to noise):



Comparing different data sets…

… with same input data, but different output values (due to noise):



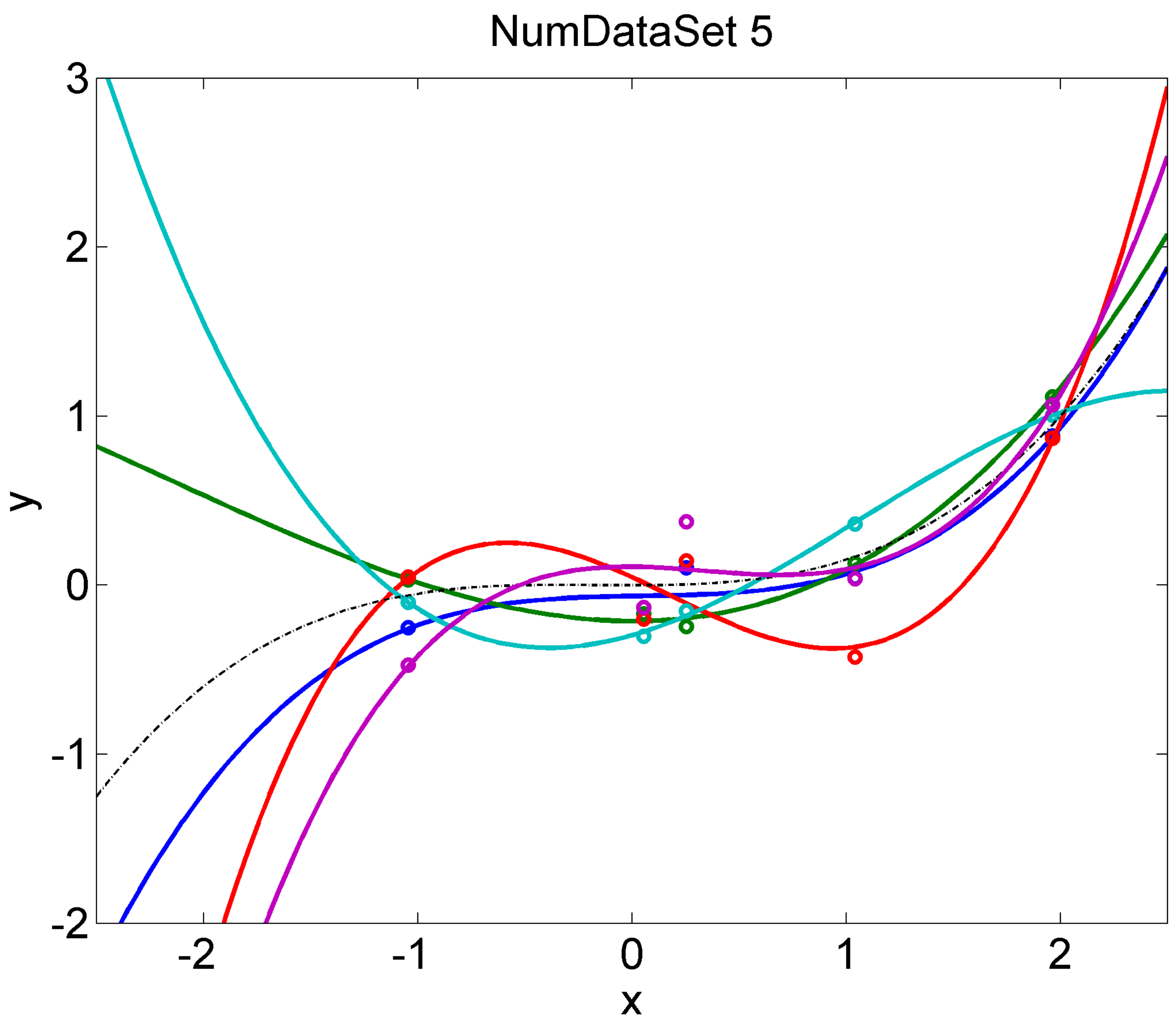
Our parameter estimate is also noisy!
It depends on the noise in the data

Comparing different data sets…

Can we also estimate our uncertainty in $\boldsymbol{\theta}$?

Compute probability of $\boldsymbol{\theta}$ given data

$$\Rightarrow \quad p(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y})$$



How to find the parameters $\boldsymbol{\theta}$?

Bayesian:

Bayes

Lets compute the probability of the parameters

likelihood prior

$$p(\boldsymbol{\theta}|\boldsymbol{X},\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{X}|\boldsymbol{y})}$$

posterior evidence

Intuition: If you assign each parameter estimator a “probability of being right”, the average of these parameter estimators will be better than the single one

How to get the posterior?

Bayes Theorem for Gaussians

$$p(\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x} | \mathbf{0}, \boldsymbol{A}\right)$$
$$p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{y} | \boldsymbol{F}\boldsymbol{x}, \sigma^2 \boldsymbol{I}\right)$$

$$p(\boldsymbol{x}|\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{x} | \boldsymbol{\Sigma}\boldsymbol{F}^T\boldsymbol{y}, \sigma^2 \boldsymbol{\Sigma}\right)$$
$$\boldsymbol{\Sigma} = \left(\boldsymbol{F}^T\boldsymbol{F} + \sigma^2 \boldsymbol{A}^{-1}\right)^{-1}$$

For our model:

Prior over parameters:

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, \lambda^{-1}\boldsymbol{I})$$

Data Likelihood

$$p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{\Phi}\boldsymbol{\theta}, \sigma^2\boldsymbol{I})$$

Posterior over parameters:

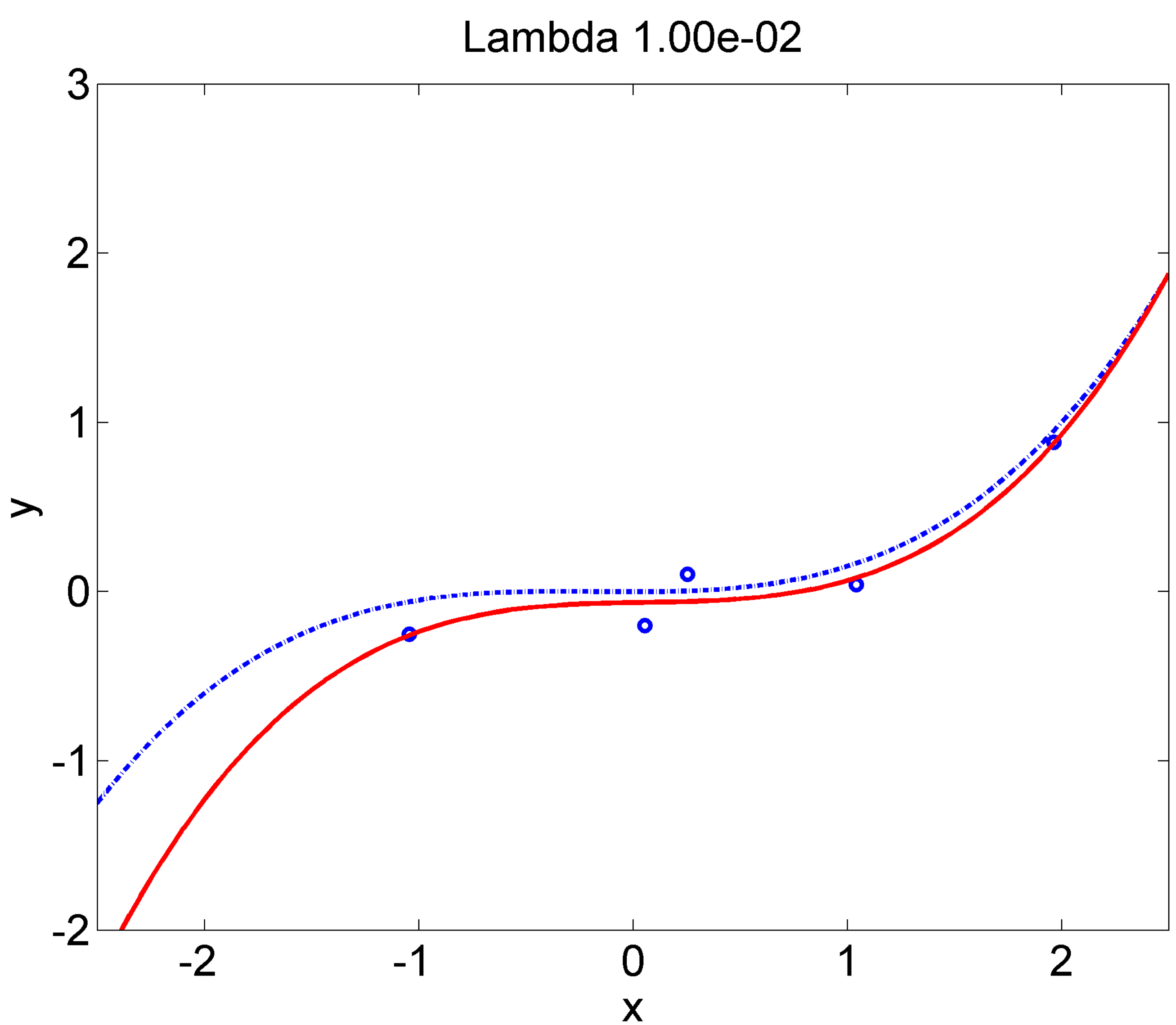
$$p(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{X}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$$

$$\boldsymbol{\Sigma}_N = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \boldsymbol{\sigma}^2\boldsymbol{\lambda}\boldsymbol{I})^{-1}$$
$$\boldsymbol{\mu}_N = \boldsymbol{\Sigma}_N\boldsymbol{\Phi}^T\boldsymbol{y}$$

What to do with the posterior?

We could sample from it to **estimate uncertainty**

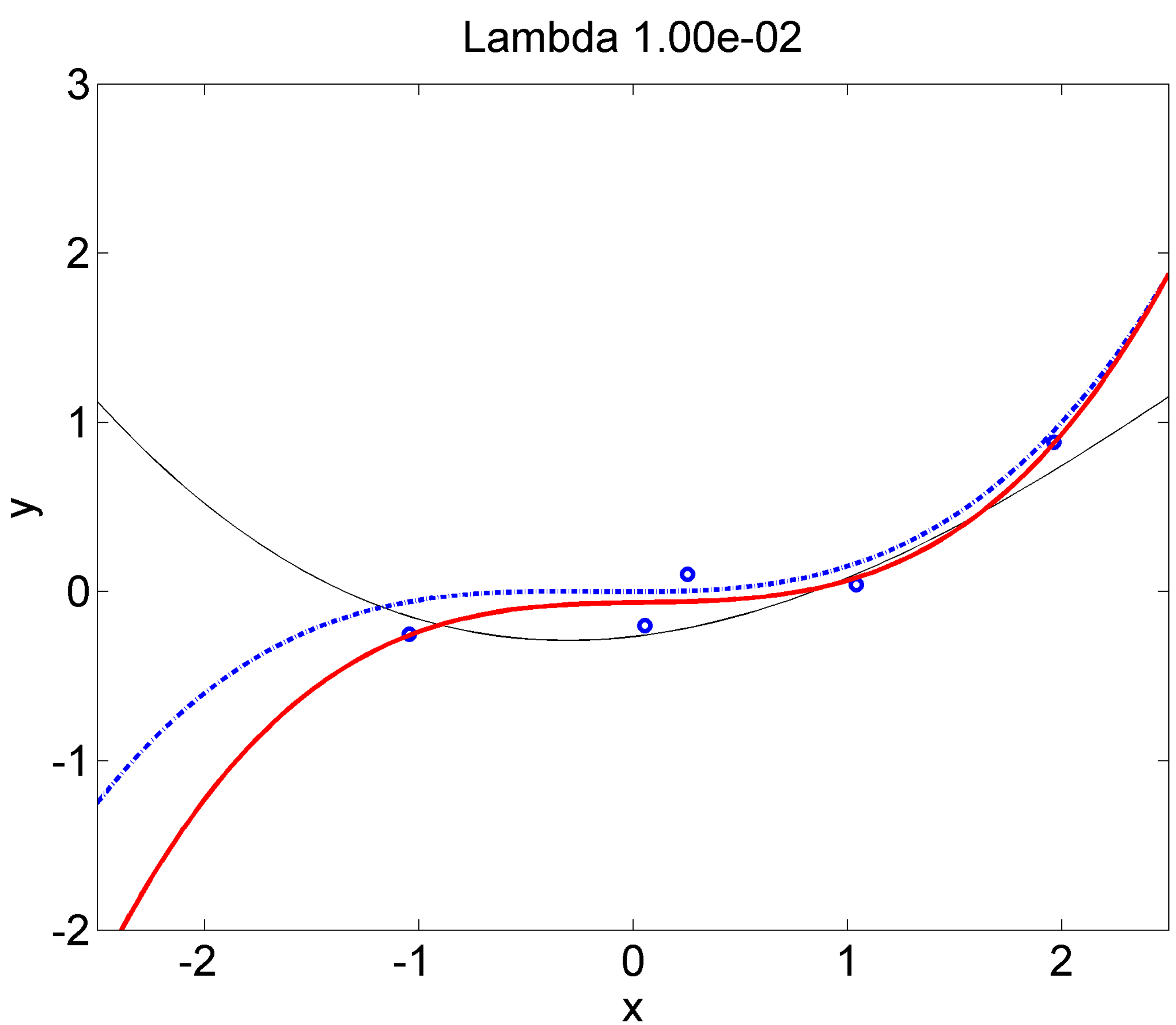
$$\boldsymbol{\theta}_i \sim p(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{X}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$$



What to do with the posterior?

We could sample from it to **estimate uncertainty**

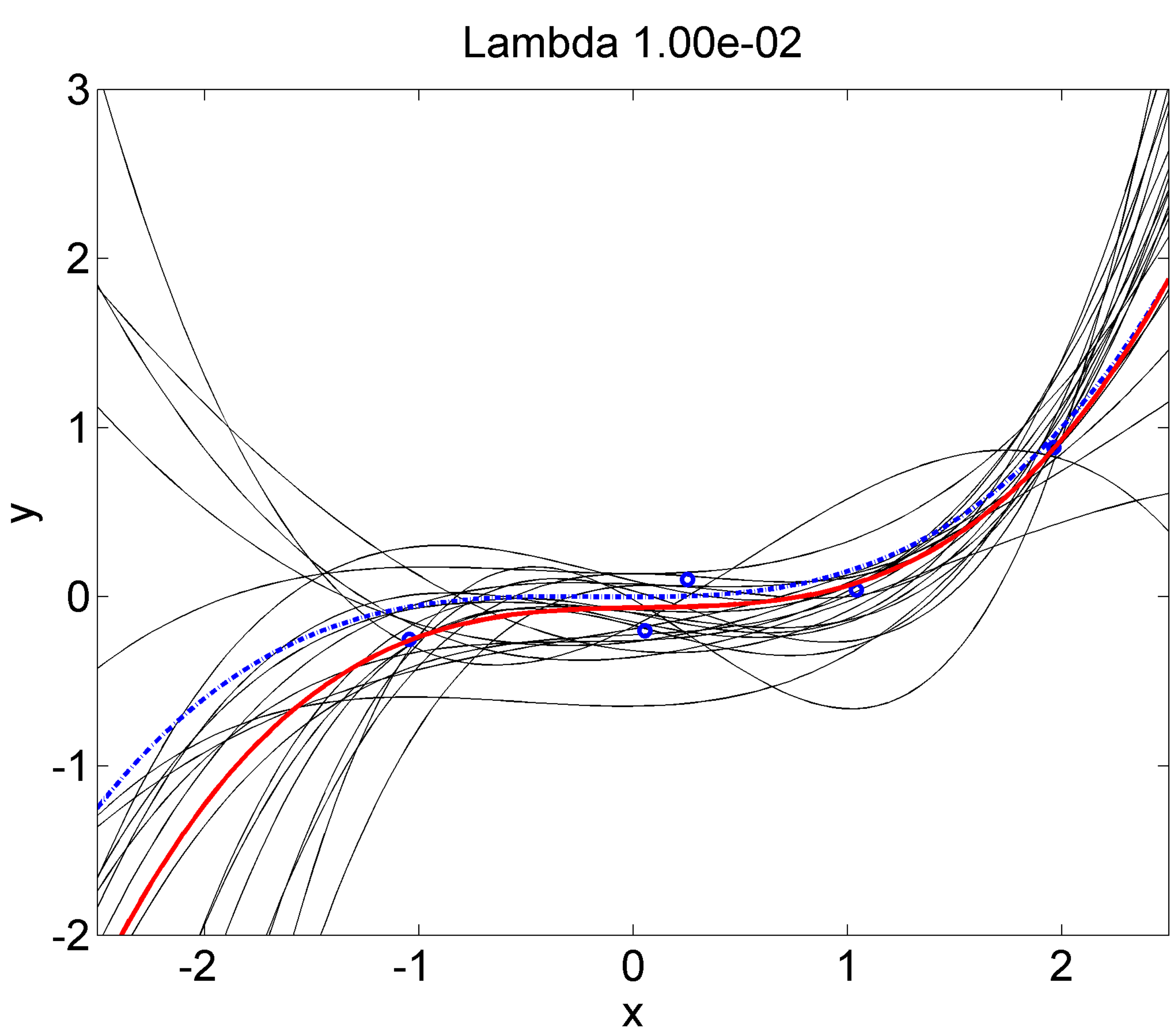
$$\boldsymbol{\theta}_i \sim p(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{X}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$$



What to do with the posterior?

We could sample from it to **estimate uncertainty**

$$\boldsymbol{\theta}_i \sim p(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{X}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$$



Full Bayesian Regression

We can also do that in closed form: integrate out all possible parameters

$$p(y_*|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) = \int p(y_*|\boldsymbol{x}_*, \boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y})d\boldsymbol{\theta}$$

likelihood parameter posterior

pred. function value test input training data

Predictive Distribution is again a Gaussian for Gaussion likelihood and parameter posterior

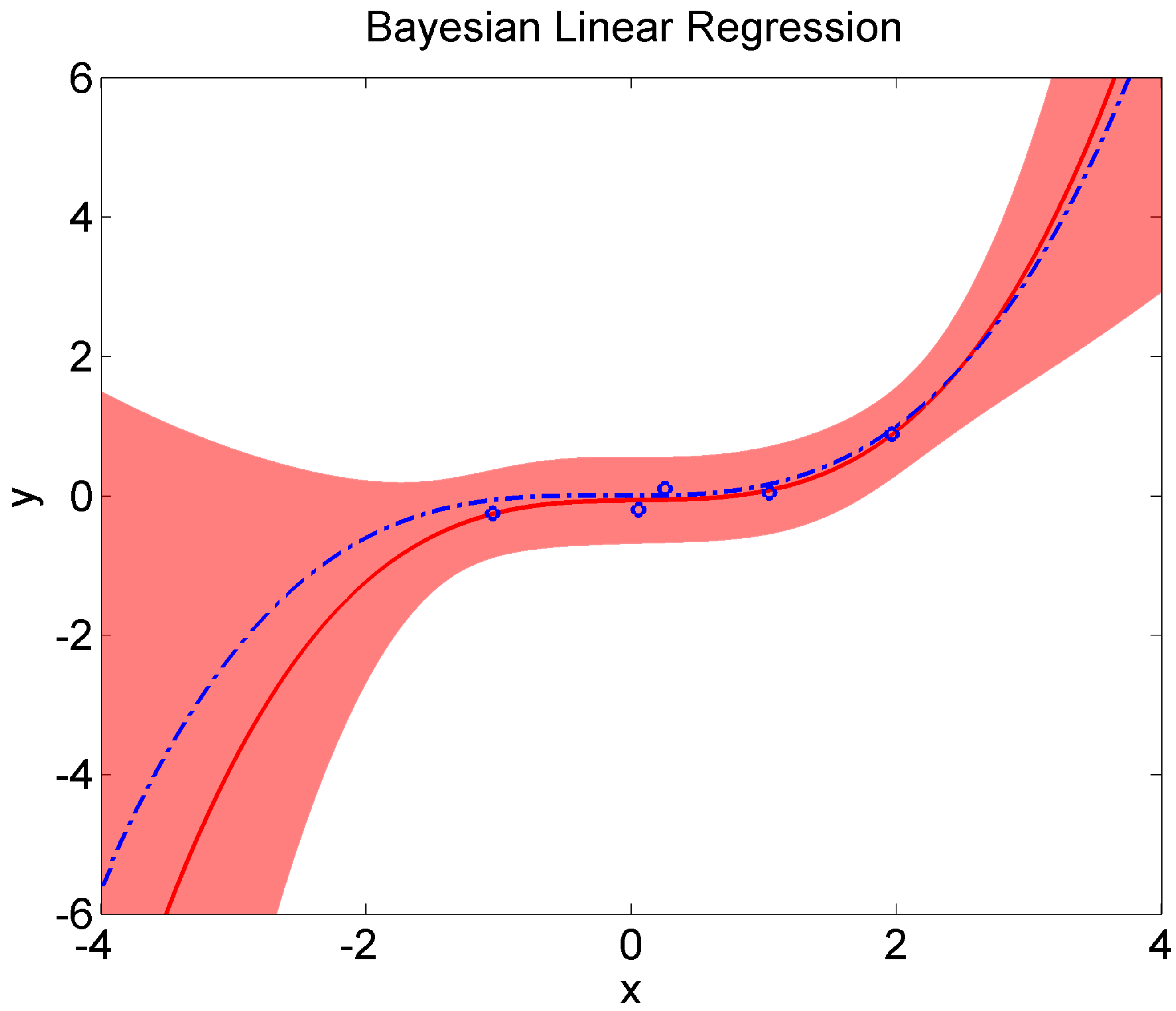
$$p(y_*|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{x}_*) = \mathcal{N}(\boldsymbol{y}_*|\mu(\boldsymbol{x}_*), \sigma^2(\boldsymbol{x}_*))$$

$$\mu(\boldsymbol{x}_*) = \boldsymbol{\phi}^T(\boldsymbol{x}_*)(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \boldsymbol{\sigma}^2\lambda\boldsymbol{I})^{-1}\boldsymbol{\Phi}^T\boldsymbol{y}$$

$$\sigma^2(\boldsymbol{x}_*) = \sigma^2(1 + \boldsymbol{\phi}^T(\boldsymbol{x}_*)\boldsymbol{\Sigma}_N\boldsymbol{\phi}(\boldsymbol{x}_*))$$

State Dependent Variance!

Integrating out the parameters



Variance depends on the information in the data!

Quick Summary

- Models that are linear in the parameters: $y = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\theta}$
- **Overfitting** is bad
- **Model selection** (leave-one-out cross validation)
- **Parameter Estimation:** Frequentist vs. Bayesian
  - **Least Squares** ~ Maximum Likelihood estimation (ML)
  - **Ridge Regression** ~ Maximum a Posteriori estimation (MAP)
- **Bayesian Regression** integrates out the parameters when predicting
  - State dependent uncertainty