2. Introduction to Supervised Learning

Ludwig Krippahl

Supervised Learning

Summary

- Supervised learning, basic concepts
- Regression and classification
- Fitting curves with Least Mean Squares

Basic concepts

Supervised Learning

Basic idea

- We have a set of labelled data

$$\{(x^1, y^1), \ldots, (x^n, y^n)\}$$

- We assume there is some function

$$F(X) : X \rightarrow Y$$

- The goal of Supervised Learning is to find (from the examples)

$$g(\theta, X) : X \rightarrow Y$$

- such that $g(\theta, X)$ approximates $F(X)$
- Supervised because we can compare $g(\theta, X)$ to Y

Training (Supervised learning)

- Ideally, we want to approximate $F(X) : X \rightarrow Y$ for all X
- But, for now, we'll consider only our Training Set

$$\{(x^1, y^1), \ldots, (x^n, y^n)\}$$

- Training Set

- The data we use to adjust the parameters θ in our model.
- More generally: data used to choose a hypothesis

- Training Error or Empirical Error

- The error on the training set for each instance of θ.
- (Sample Error in Mitchell 1997)

Supervised Learning

Our ML problem for today:

- Goal: Predict the Y values in our training set
- Performance: minimise training error
- Data: $\{(x_1, y_1), \ldots, (x_n, y_n)\}$

Classification and Regression

- In Classification Y is discrete.

- Examples: SPAM detection, predict if mushrooms are poisonous
- Find function to split data in differente sets

- In Regression Y is continuous.

- Examples: predicting trends, prices, purchase probabilities
- Find function that approximates Y

Regression

Regression example

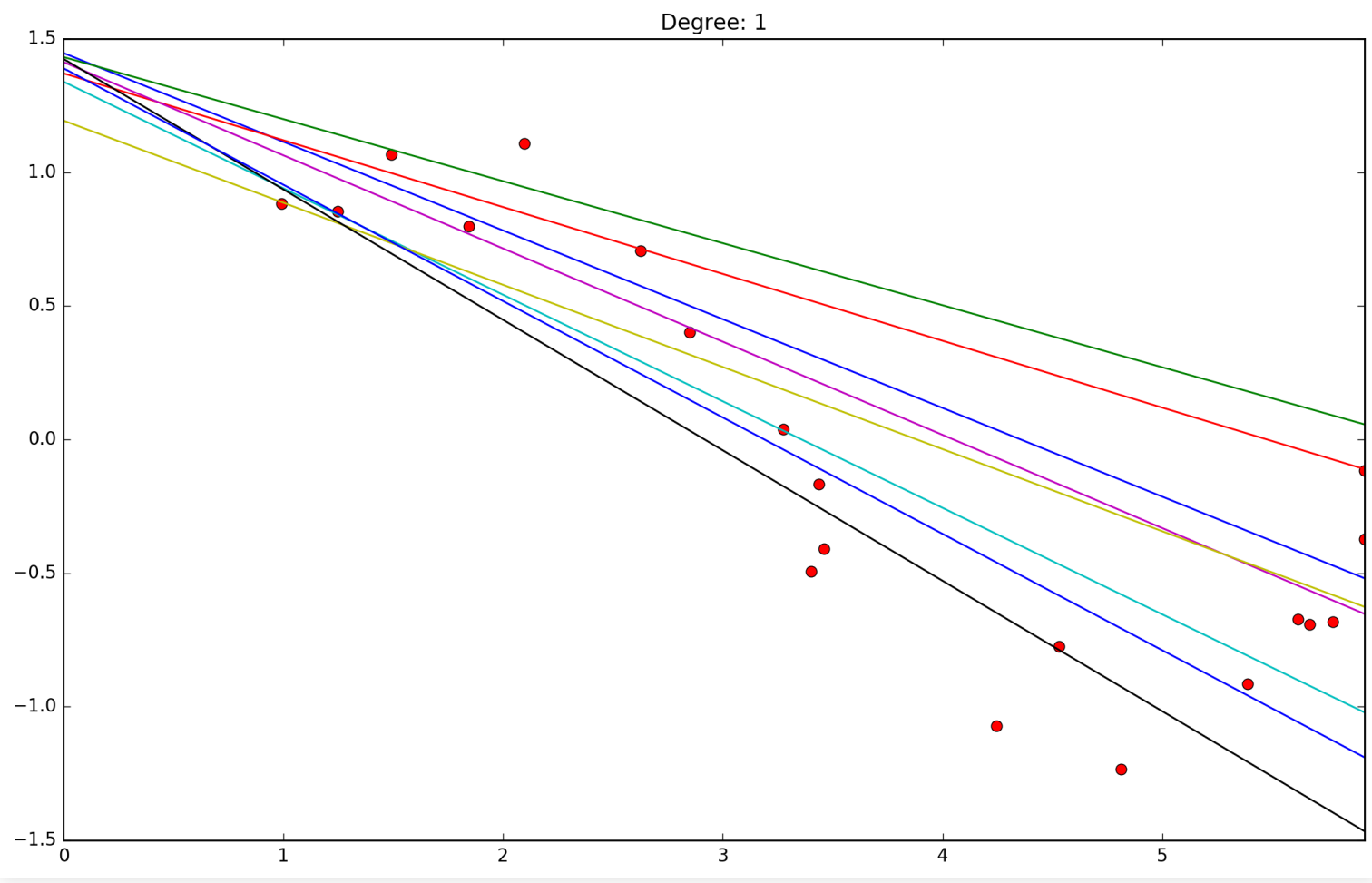
- Polynomial fitting: a simple example of linear regression.

$$y = \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_{n+1}$$

- Example: we have a set of (x, y) points and want to fit the best line: $y = \theta_1 x + \theta_2$
- How to find the best line?

Regression

- How to find the best line?



Finding the best line

- Assume y is a function of x plus some error:

$$y = F(x) + \epsilon$$

- We want to approximate $F(x)$ with some $g(x, \theta)$.
- Assuming $\epsilon \sim N(0, \sigma^2)$ and $g(x, \theta) \sim F(x)$, then:

$$p(y|x) \sim \mathcal{N}(g(x, \theta), \sigma^2)$$

- Given $\mathcal{X} = \{x^t, y^t\}_{t=1}^N$ and
- knowing that $p(x, y) = p(y|x)p(x)$

$$p(X, Y) = \prod_{t=1}^{n} p(x^t, y^t) = \prod_{t=1}^{n} p(y^t|x^t) \times \prod_{t=1}^{n} p(x^t)$$

Finding the best line

- The probability of (X, Y) given some $g(x, \theta)$ is the
- likelihood of parameters θ:

$$l(\theta|\mathcal{X}) = \prod_{t=1}^{n} p(x^t, y^t) = \prod_{t=1}^{n} p(y^t|x^t) \times \prod_{t=1}^{n} p(x^t)$$

Likelihood

- Data points (x, y) are randomly sampled from all possible values.
- But θ is not a random variable.
- Find the θ that, if true, would make the data is most probable
- In other words, find the θ of maximum likelihood

Maximum likelihood

$$l(\theta|\mathcal{X}) = \prod_{t=1}^{n} p(x^t, y^t) = \prod_{t=1}^{n} p(y^t|x^t) \times \prod_{t=1}^{n} p(x^t)$$

- First, take the logarithm (same maximum)

$$L(\theta|\mathcal{X}) = log \left(\prod_{t=1}^{n} p(y^t|x^t) \times \prod_{t=1}^{n} p(x^t) \right)$$

- We ignore $p(X)$, since it's independent of θ

$$L(\theta|\mathcal{X}) \propto log \left(\prod_{t=1}^{n} p(y^t|x^t) \right)$$

- Replace the expression for the normal:

$$\mathcal{L}(\theta|\mathcal{X}) \propto log \prod_{t=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{-[y^t - g(x^t|\theta)]^2/2\sigma^2}$$

Maximum likelihood

- Replace the expression for the normal:

$$\mathcal{L}(\theta|\mathcal{X}) \propto log \prod_{t=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{-[y^t - g(x^t|\theta)]^2/2\sigma^2}$$

- Simplify:

$$\mathcal{L}(\theta|\mathcal{X}) \propto log \prod_{t=1}^{n} e^{-[y^t - g(x^t|\theta)]^2}$$

$$\mathcal{L}(\theta|\mathcal{X}) \propto -\sum_{t=1}^{n} [y^t - g(x^t|\theta)]^2$$

Maximum likelihood

$$\mathcal{L}(\theta|\mathcal{X}) \propto -\sum_{t=1}^{n}[y^t - g(x^t|\theta)]^2$$

Under our assumptions:

- Max(likelihood) = Min(squared error):

$$E(\theta|\mathcal{X}) = \sum_{t=1}^{n}[y^t - g(x^t|\theta)]^2$$

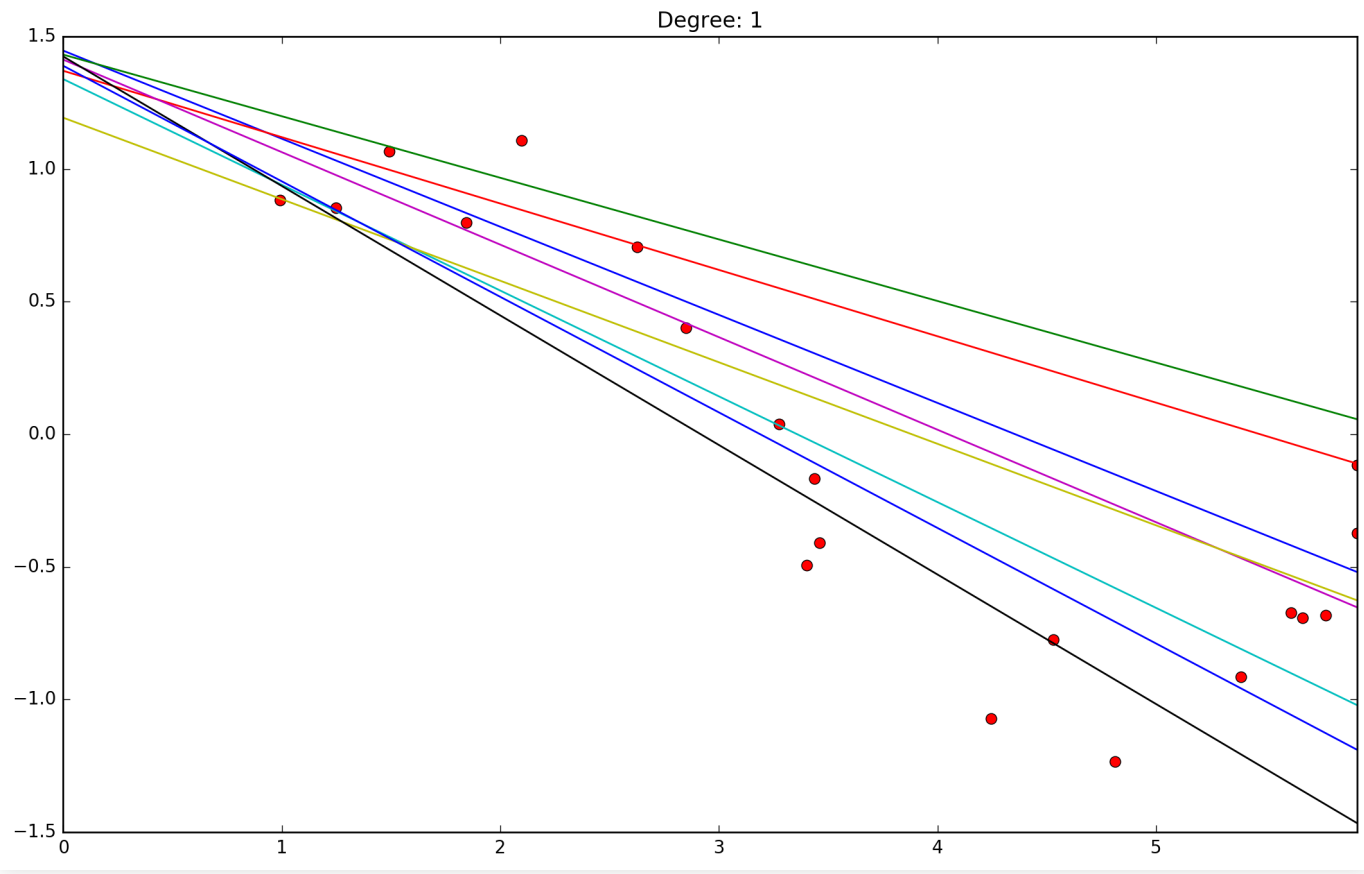
- Note: the squared error is often written

$$E(\theta|\mathcal{X}) = \frac{1}{2}\sum_{t=1}^{n}[y^t - g(x^t|\theta)]^2$$

- (but this is just for convenience in computing the derivative)

Least Mean Squares Minimization

How to find the best line?



How to find the best line?

- We find the parameters for

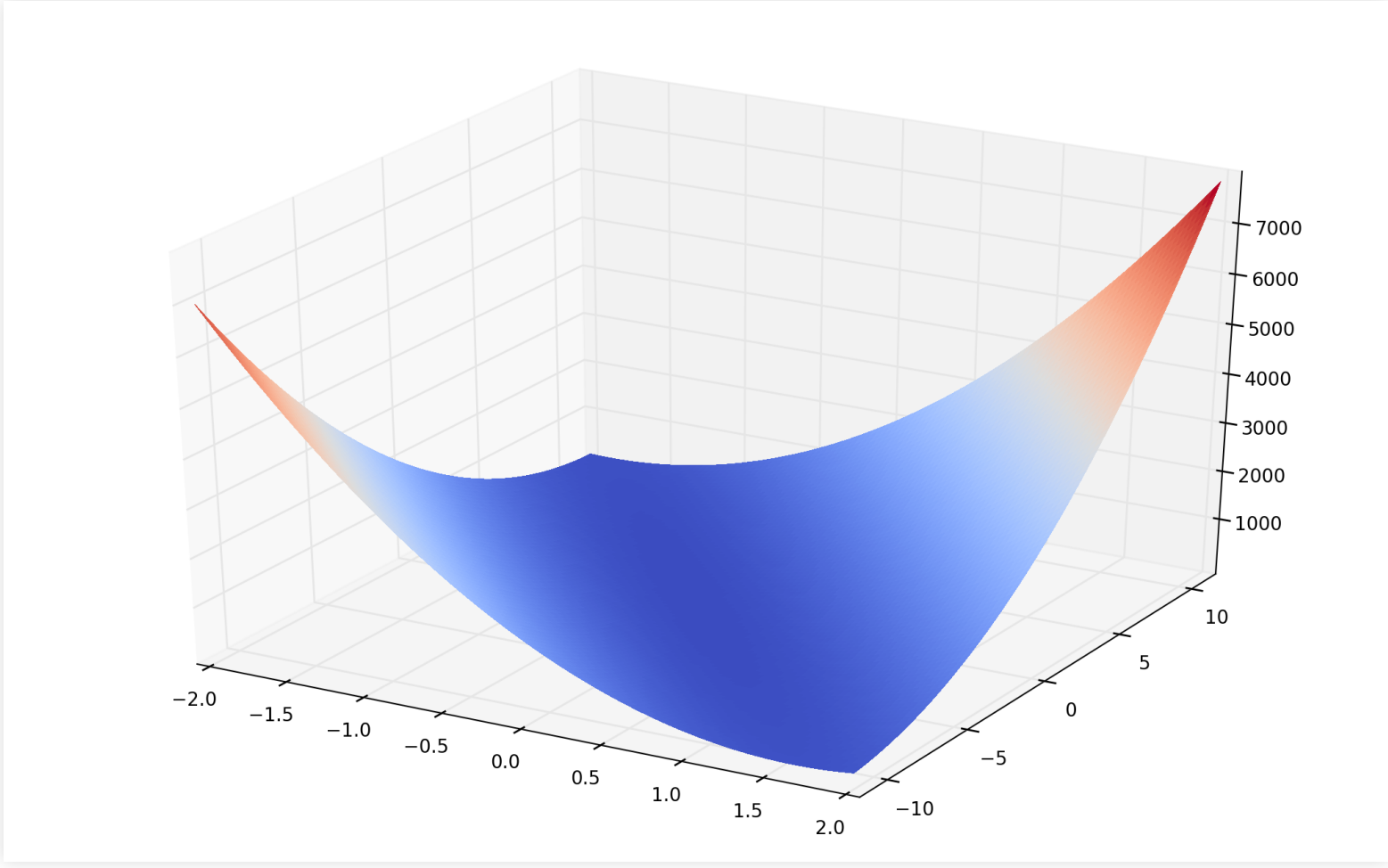
$$g(x) = x\theta_1 + \theta_2$$

- that minimize the squared error

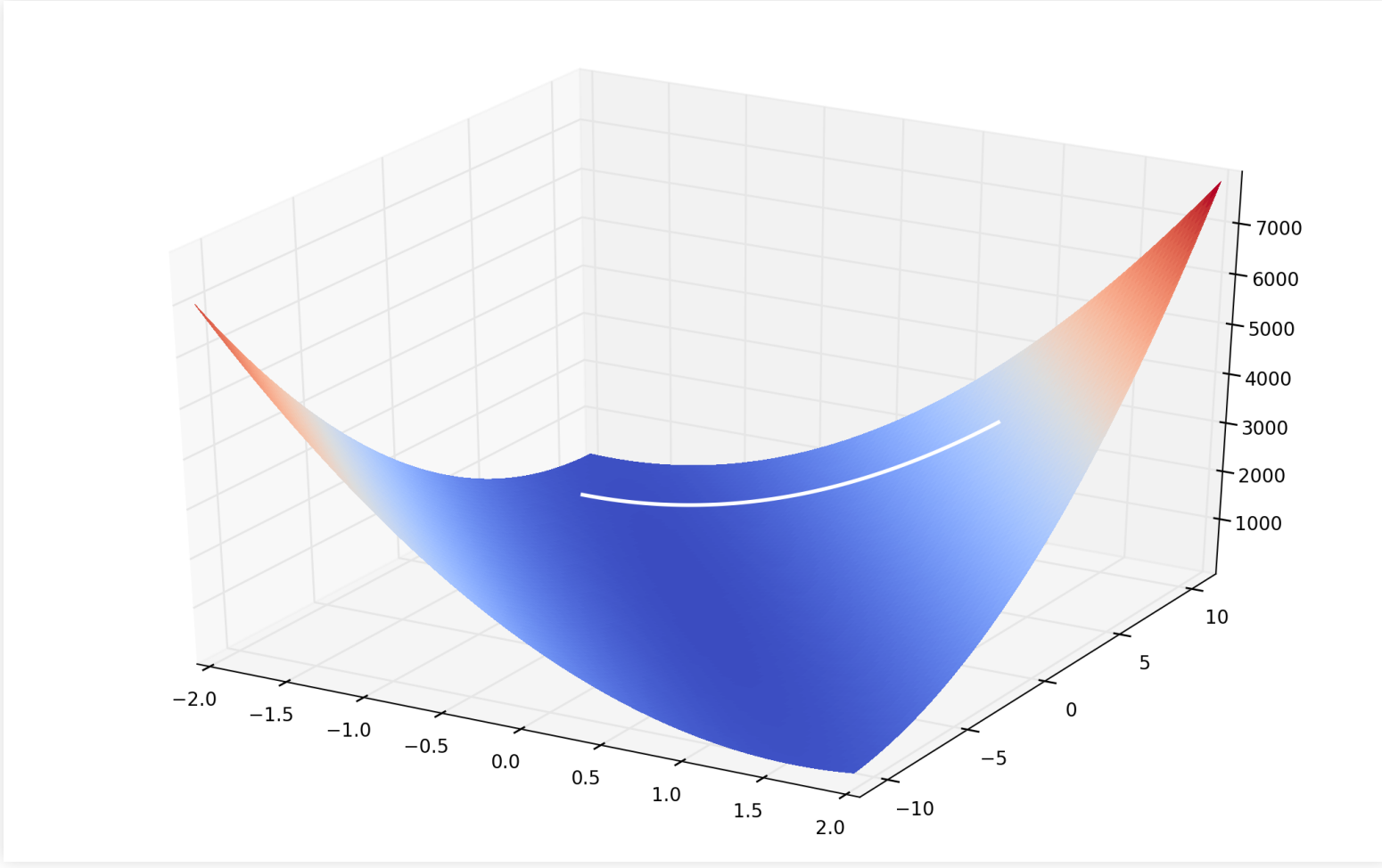
$$E(\theta|\mathcal{X}) = \sum_{t=1}^{n} [y^t - g(x^t)]^2$$

- Let's visualise this surface wrt θ

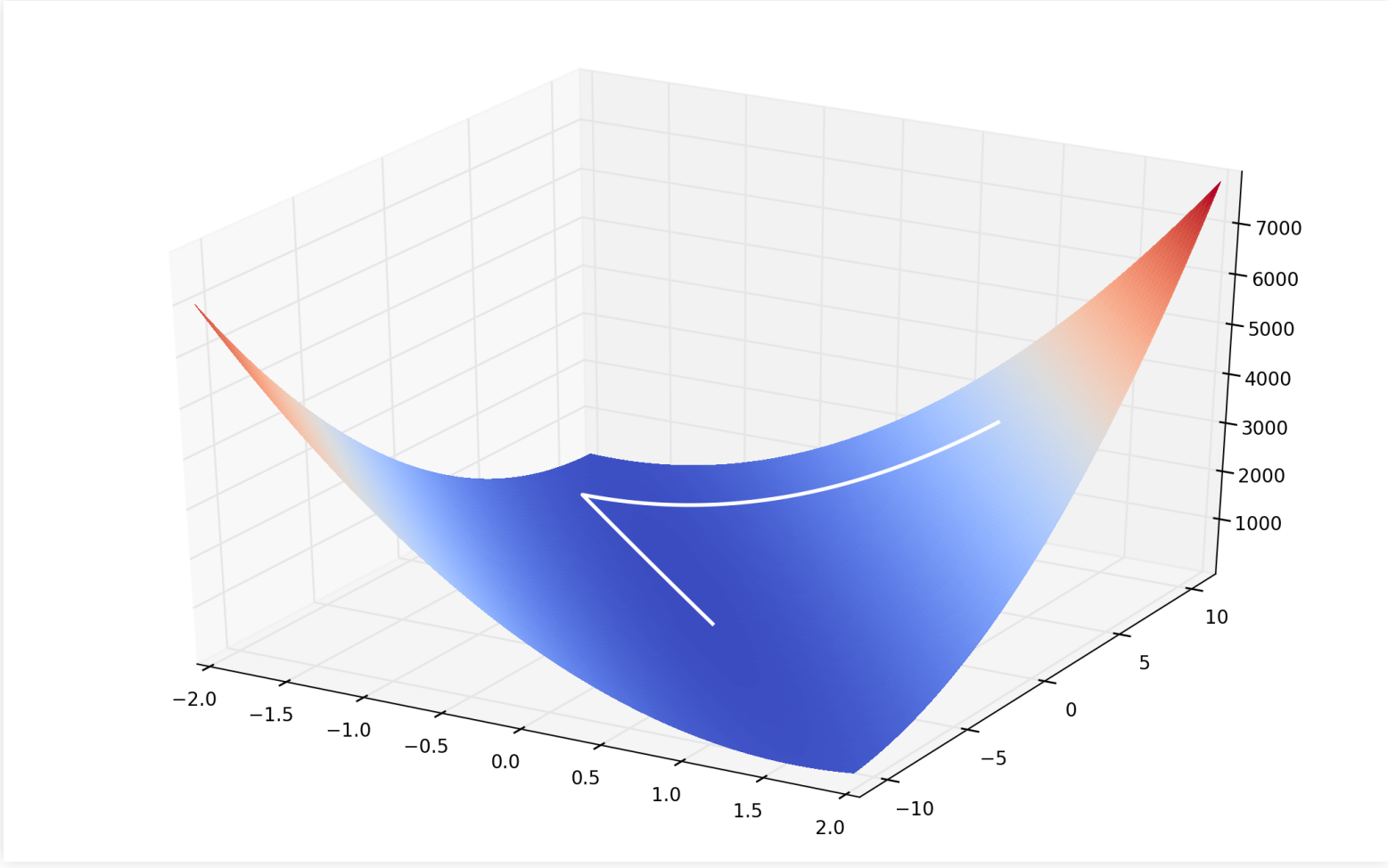
LMS


7000
6000
5000
4000
3000
2000
1000
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
−10
−5
0
5
10

LMS

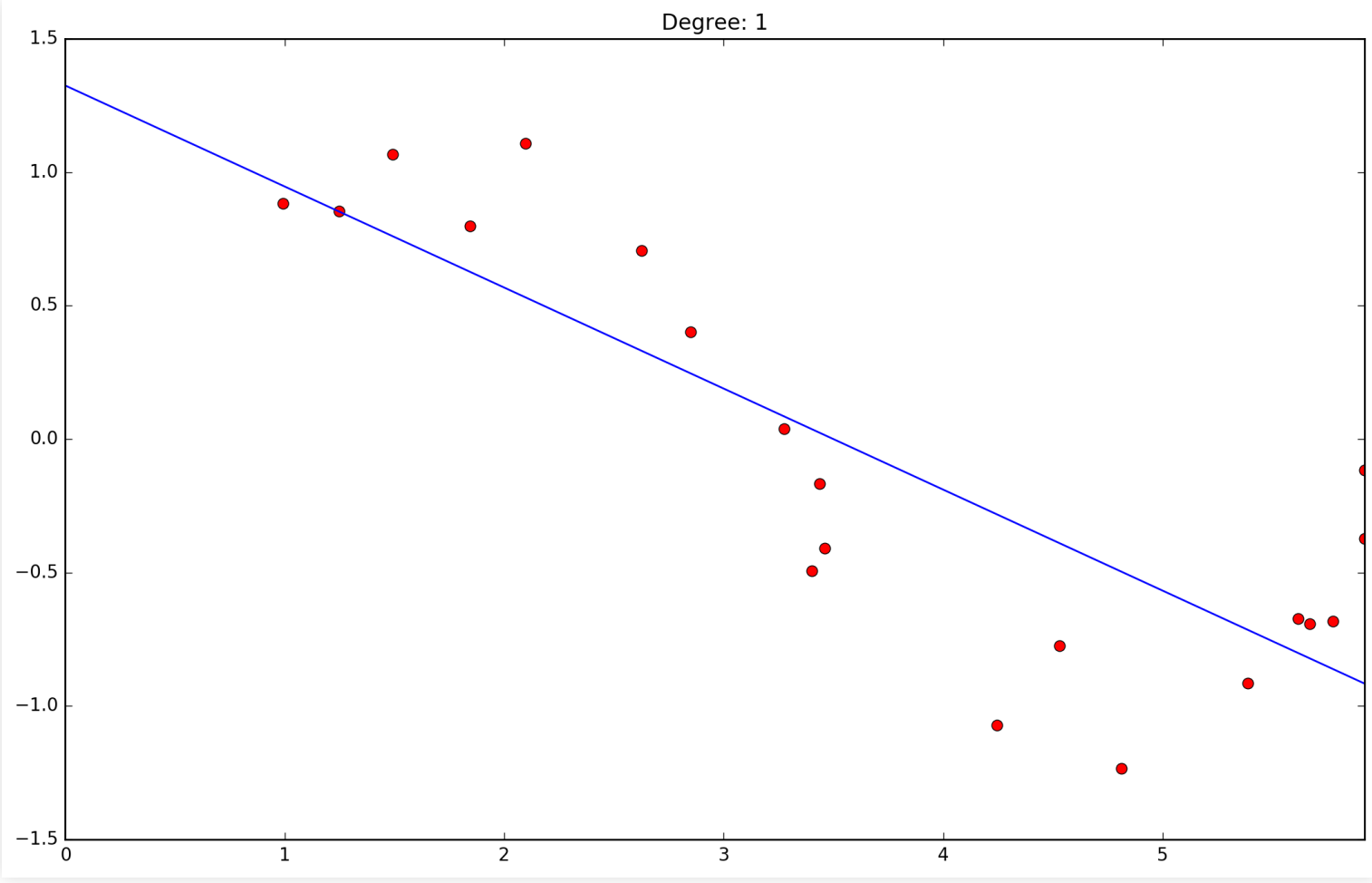

7000
6000
5000
4000
3000
2000
1000
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
−10
−5
0
5
10

LMS


7000
6000
5000
4000
3000
2000
1000
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
−10
−5
0
5
10

LMS

- This allows us to find the best θ_1, θ_2 (not a very good model...)



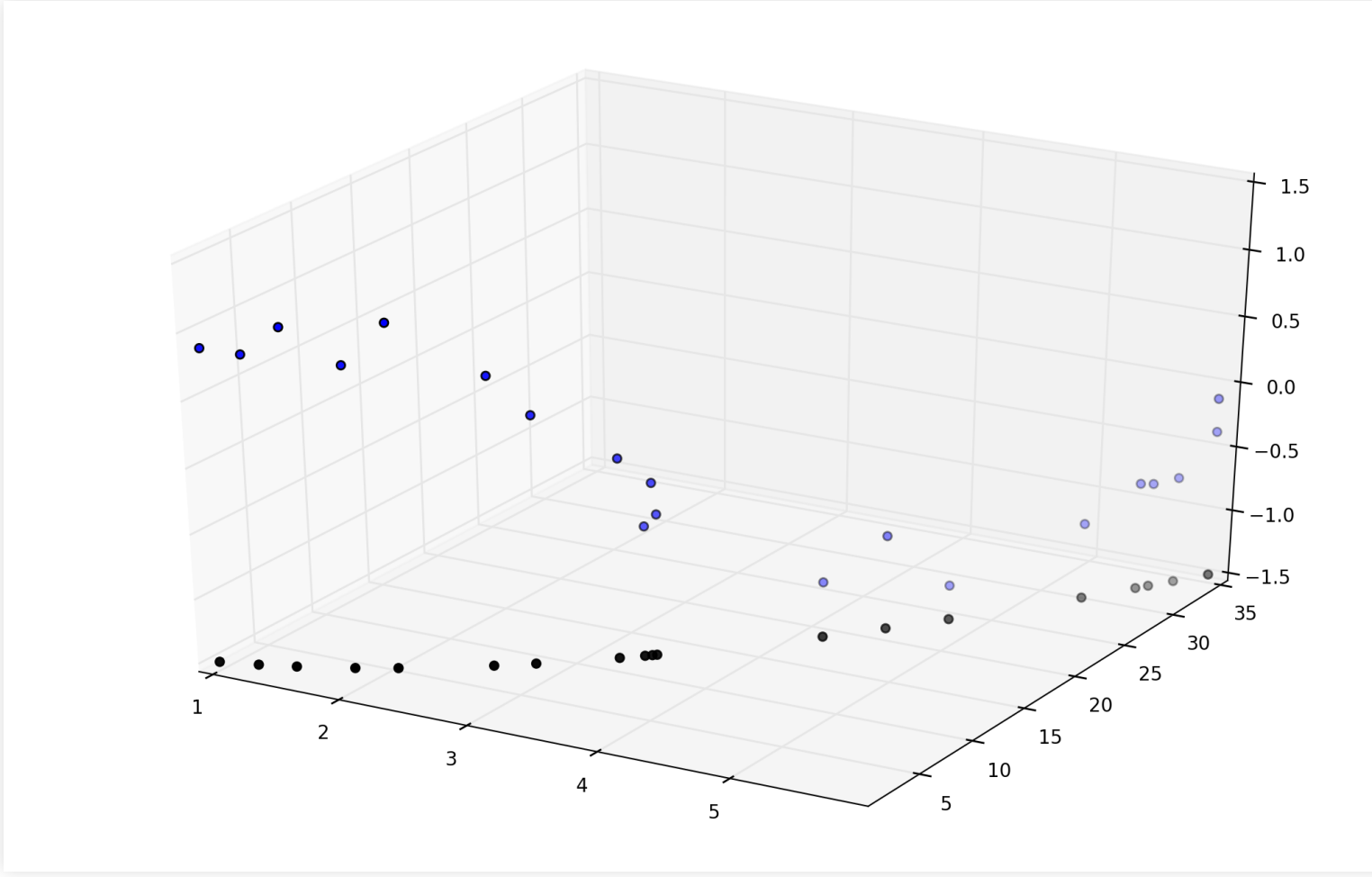
Curves

Curves

Linear Regression

- How to fit curves with something straight?
- We can change the data:
  - $\mathcal{X}_2 = \{x_1^t, x_2^t, y^t\}$, where $x_1 = x^2$ and $x_2 = x$
- Using a nonlinear transformation we project the data into a curved surface

Curves



Linear Regression

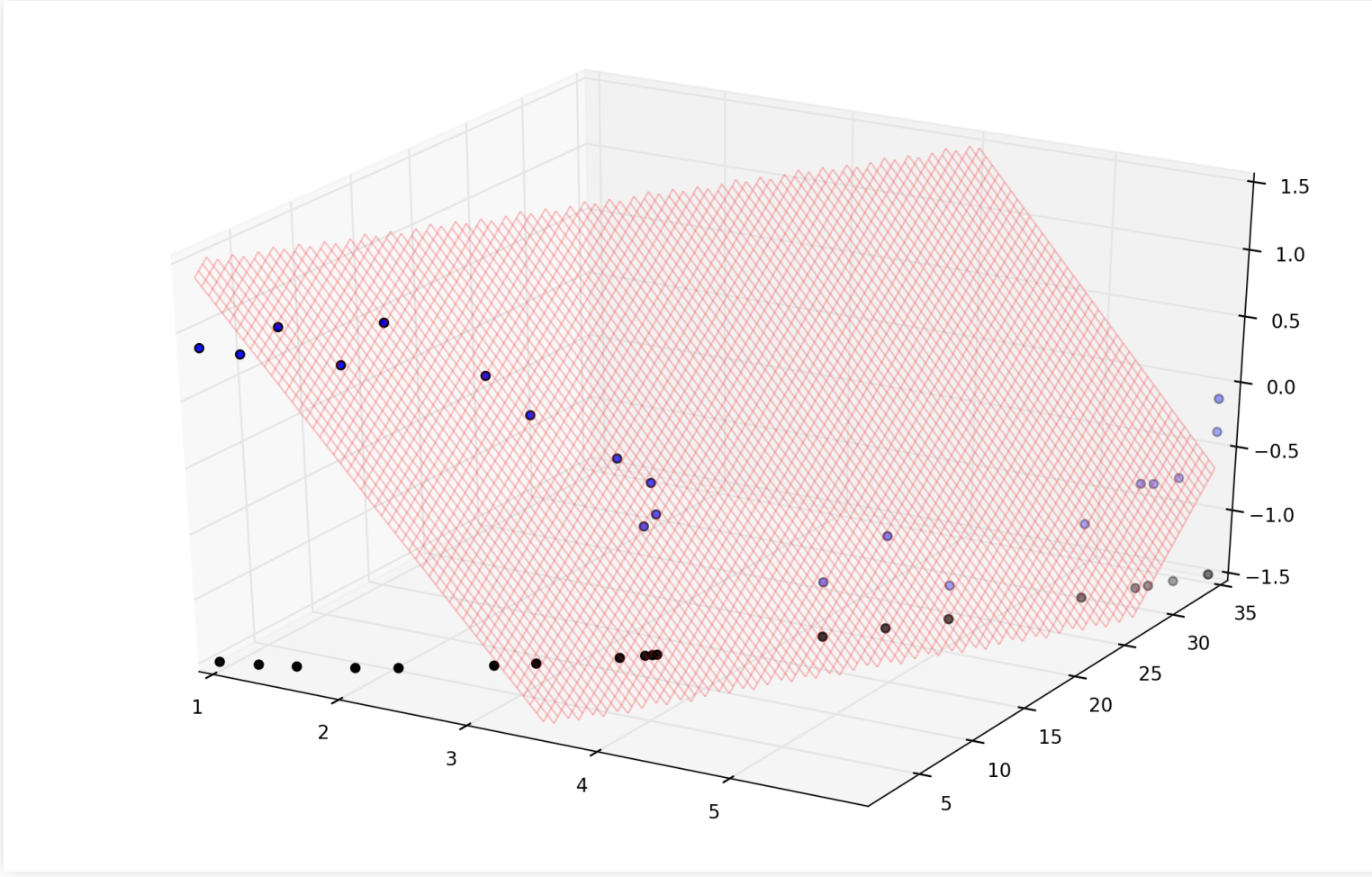
- Now we fit our new data set

$$\mathcal{X}_2 = \{x_1^t, x_2^t, y^t\}$$

- With the (linear) model in three dimensions

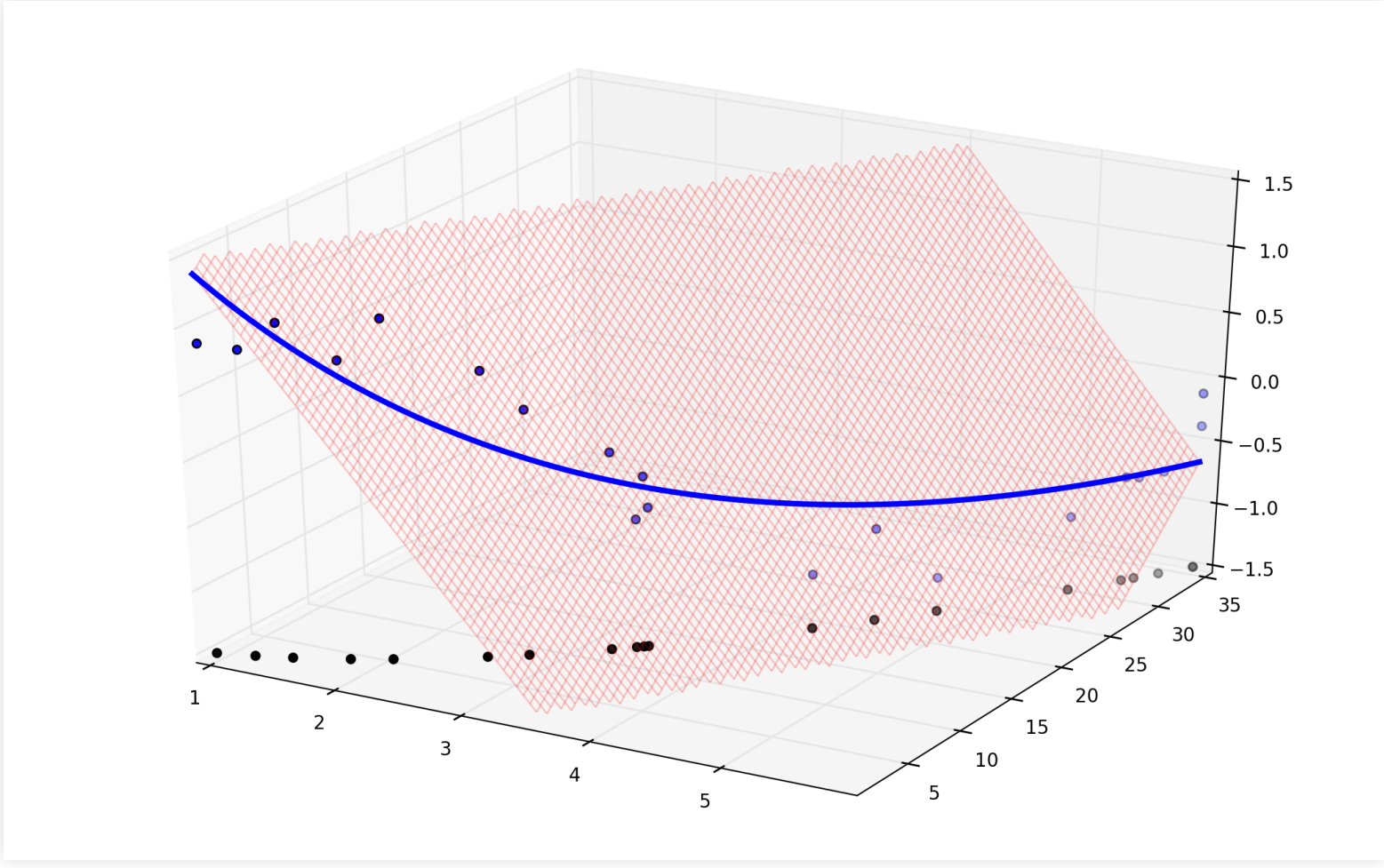
$$y = \theta_1 x_1 + \theta_2 x_2 + \theta_3$$

Curves

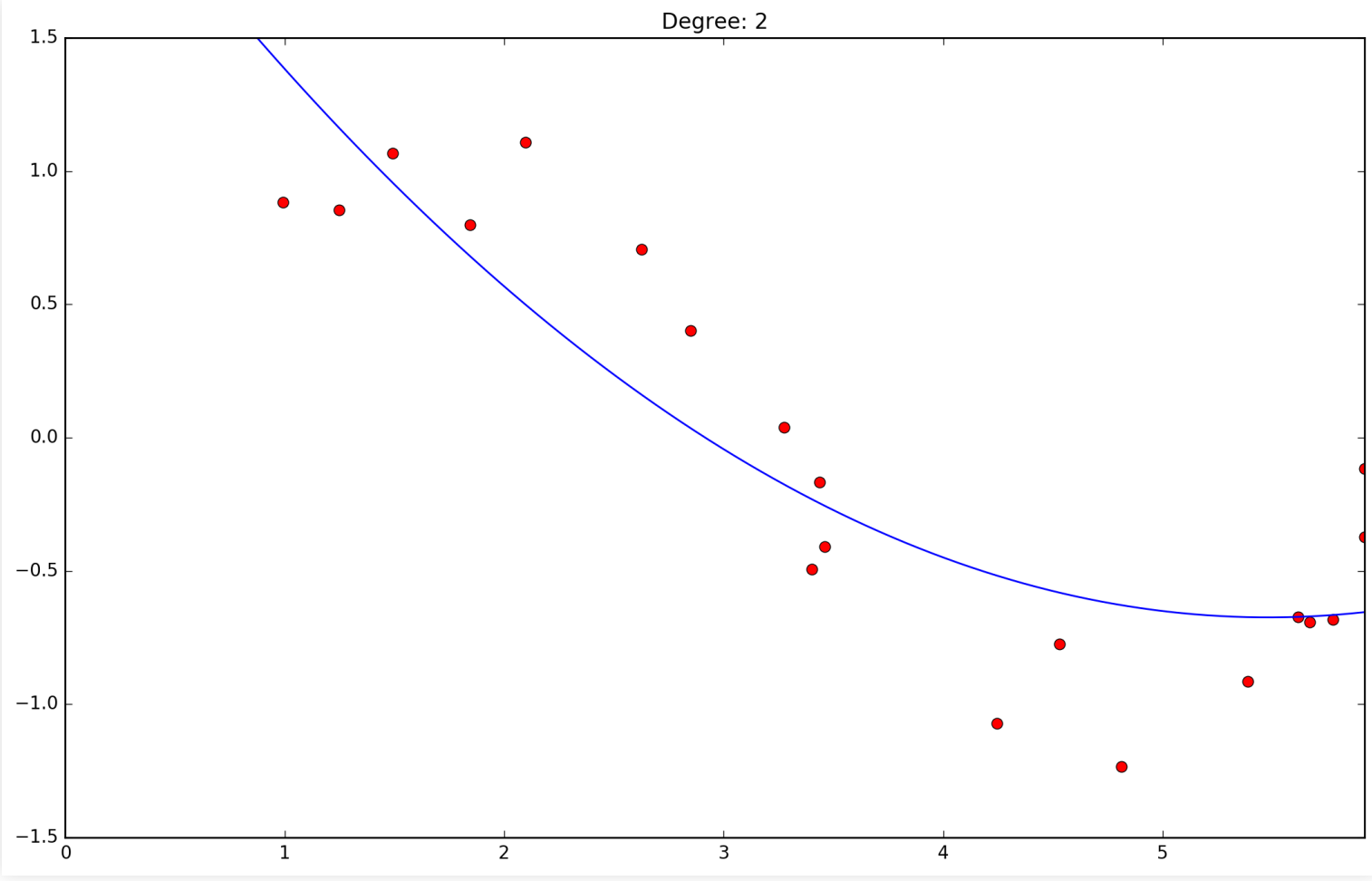


Curves

- Then we project it back using $x_1 = x^2$ and $x_2 = x$



Curves



Linear Regression

- This is the equivalent of fitting a second degree polynomial

$$y = \theta_1 x^2 + \theta_2 x + \theta_3$$

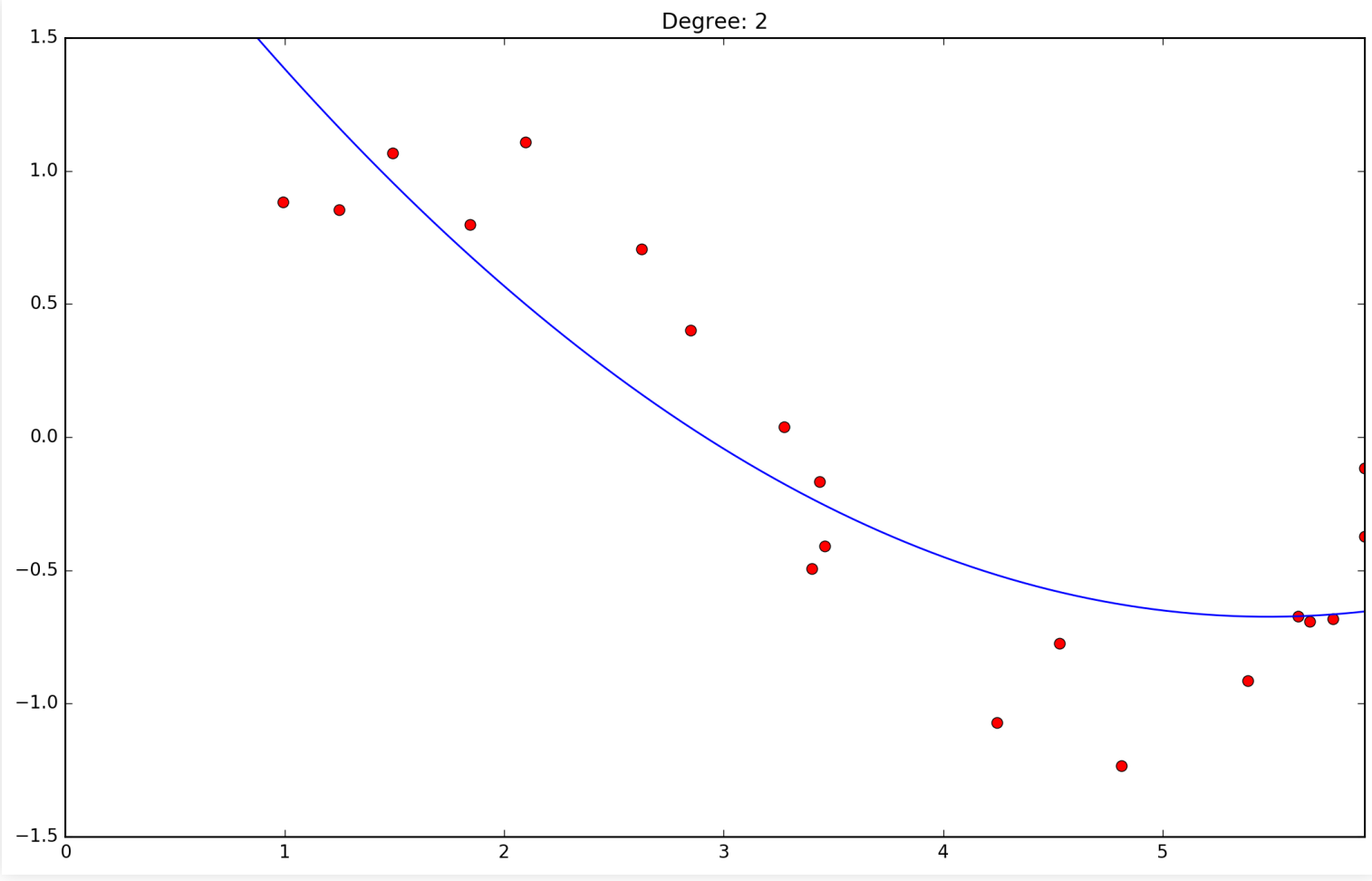
```
import numpy as np
import matplotlib.pyplot as plt

mat = np.loadtxt('polydata.csv',delimiter=';')
x = mat[:,0]
y = mat[:,1]
coefs = np.polyfit(x,y,2)

pxs = np.linspace(0,max(x),100)
poly = np.polyval(coefs,pxs)

plt.figure(figsize=(12, 8))
plt.plot(x,y,'or')
plt.plot(pxs,poly,'-')
plt.axis([0,max(x),-1.5,1.5])
plt.title('Degree: 2')
plt.savefig('testplot.png')
plt.close()
```

Curves



Curves

Linear Regression

- How to fit curves with something straight?
- Important idea:

- Add dimensions with nonlinear transformations
- Use something straight in this higher dimension space

Assumption (Inductive Bias)

- We can adjust the data with polynomials

- (or hyperplanes in higher dimensions after expansion)

Hypothesis Classes

- Polynomials of some degree

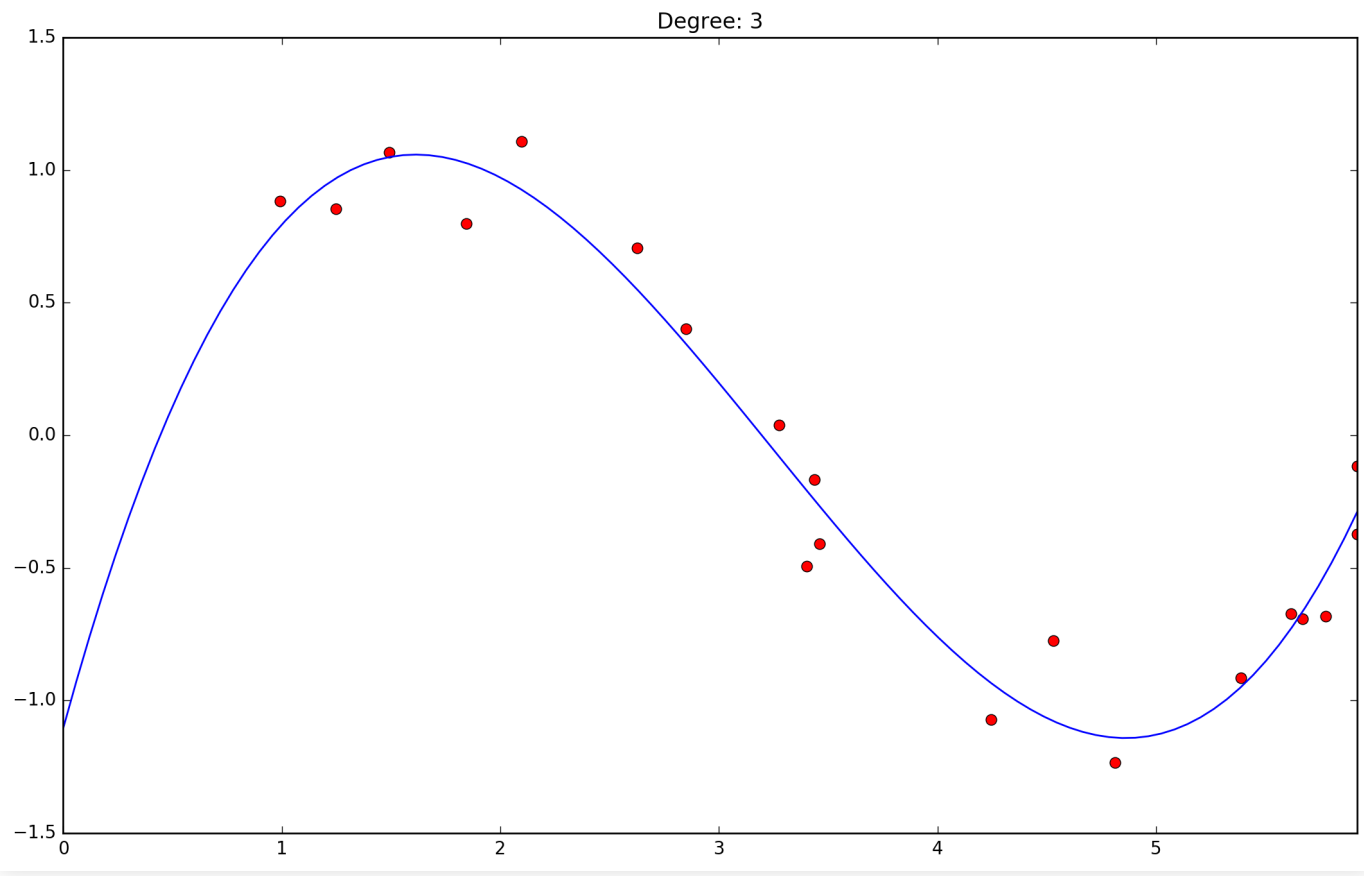
- (or straight surfaces in higher dimensions)

Curve More!

Curve more

- Improving the fit with higher polynomials, degree 3

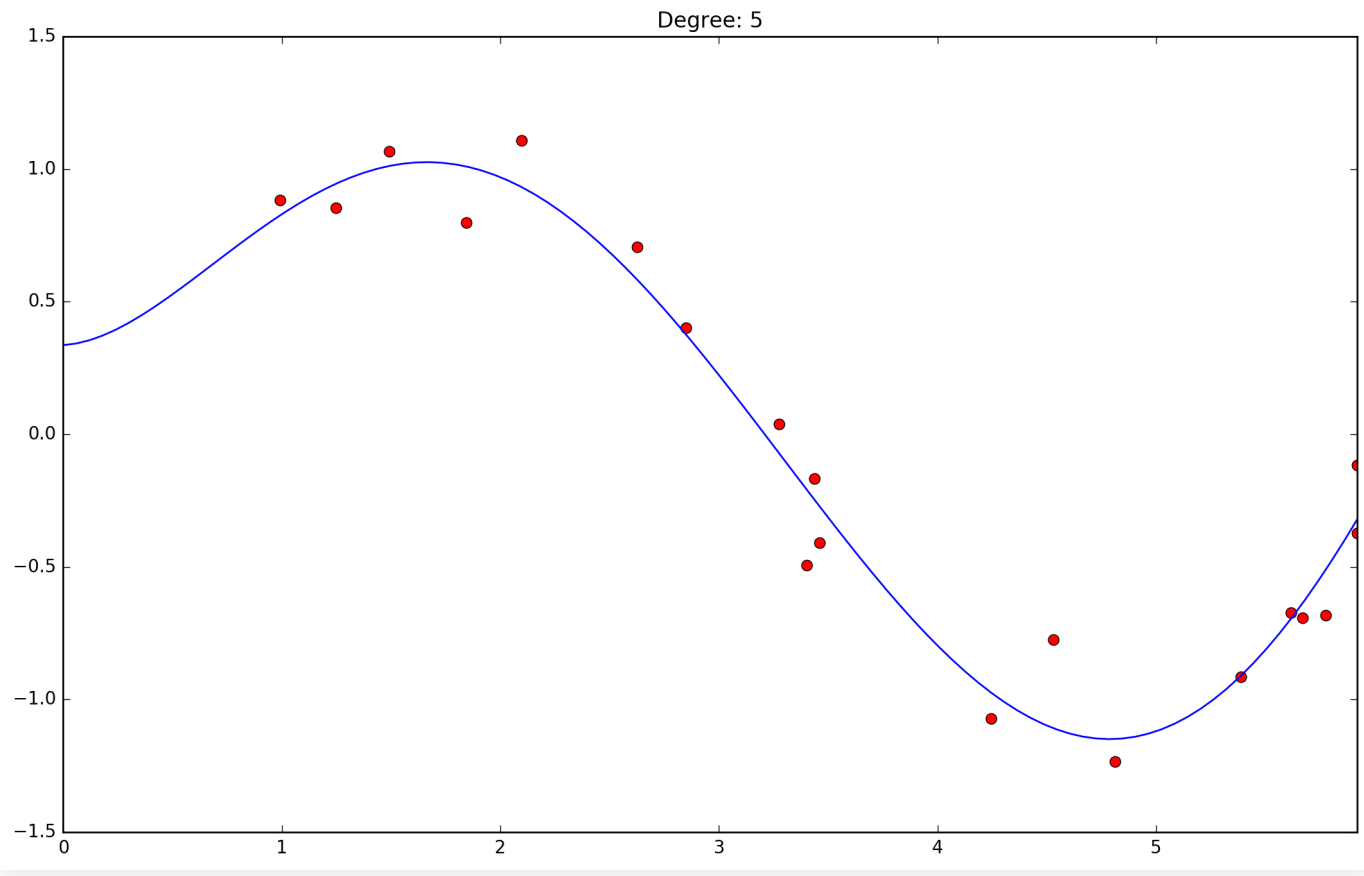
$$y = \theta_1 x^3 + \theta_2 x^2 + \theta_3 x + \theta_4$$



Curve more

- Improving the fit with higher polynomials, degree 5

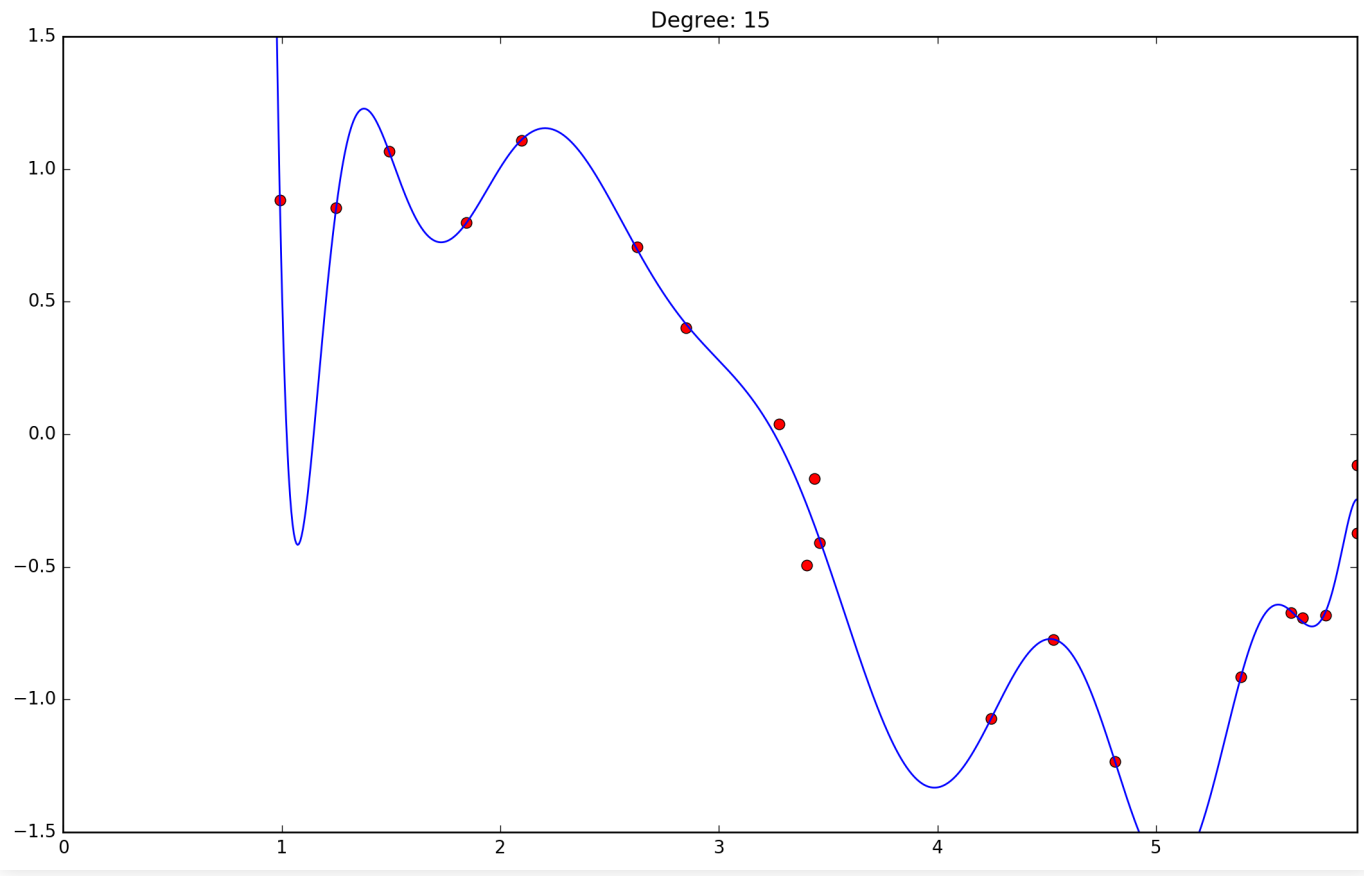
$$y = \theta_1 x^5 + \theta_2 x^4 + \ldots + \theta_5 x + \theta_6$$



Curve more

- Improving the fit with higher polynomials, degree 15

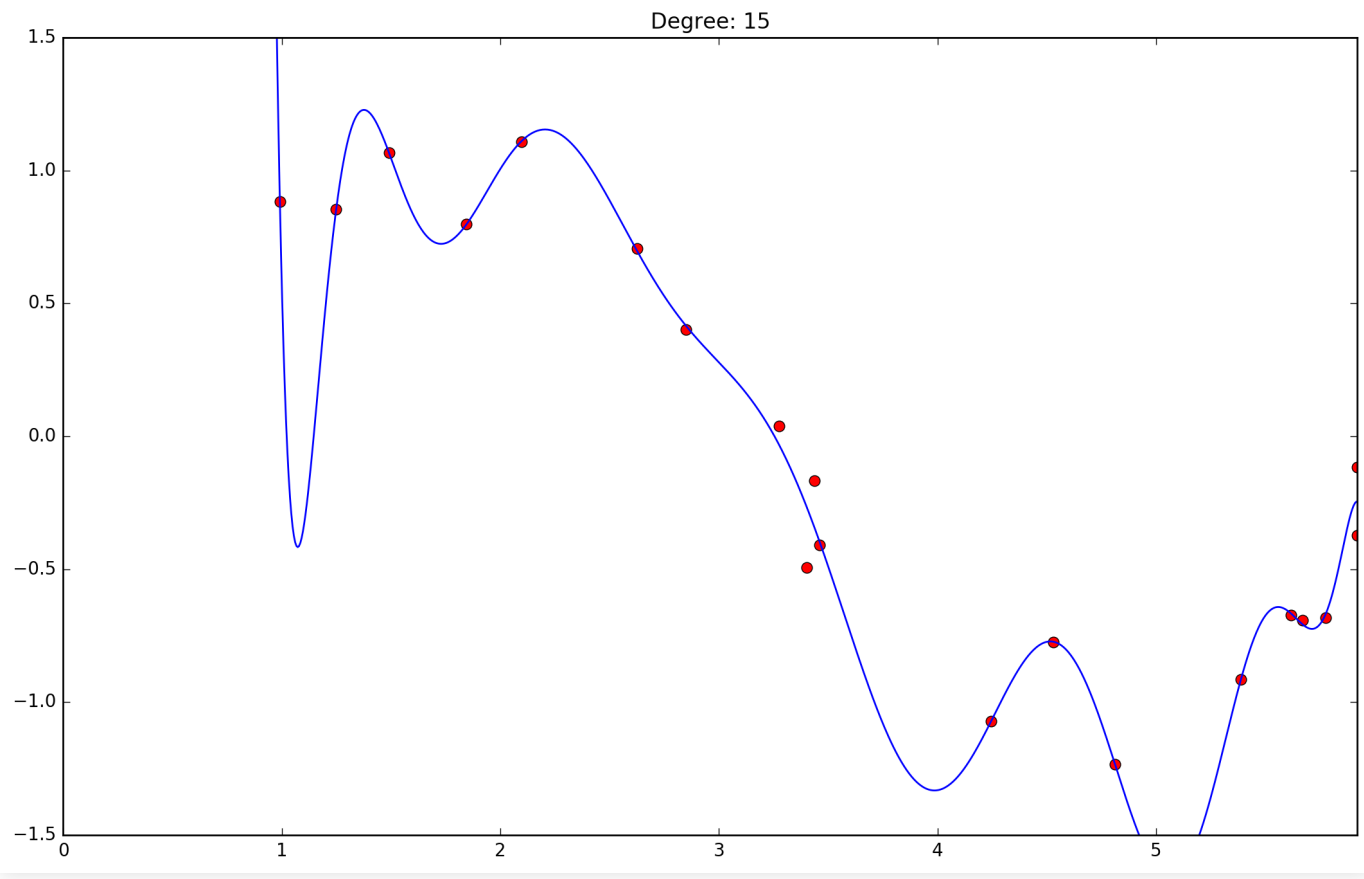
$$y = \theta_1 x^{15} + \theta_2 x^{14} + \ldots + \theta_{15} x + \theta_{16}$$



Curve more

Improving the fit?

- Degree 15 is probably not a good idea...



Curve more

Improving the fit?

- Degree 15 is probably not a good idea...

Overfitting

- The hypothesis adjusts too much to the data
- Training error is small, but increases error outside
- How can we prevent getting carried away?
- Next lecture: overfitting

Summary

Summary

- Supervised learning: Classification and Regression
- Linear regression: maximum likelihood and least mean squares
- Polynomial regression is linear regression

• (nonlinear transformation to higher dimensions)

- Overfitting:

• Nonlinear expansion can go too far

Further reading

- Bishop, Chapter 1
- Alpaydin, Section 2.6
- Marsland, Sections 1.4 and 2.4