Lecture 2

Introduction

Physical Address for Prof. Ilias Tagkopoulos
Computer Science:
Office: 3063 Kemper Hall
Phone: (530) 752-4821
Fax: (530) 752-4767

Genome and Biomedical Sciences Facility:
Office: 5313 GBSF
Phone: (530) 752-7707
Fax: (530) 754-9658

Instructor: Ilias Tagkopoulos
iliast@ucdavis.edu

Syllabus

Lecture	Date	Topic	Comments
1	9/24/2015	**Introduction**	HW1 posted
2	9/29/2015	Linear Regression	
3	10/1/2015	Other Regression methods	
4	10/6/2015	**Classification**	HW1 due - HW2 posted
5	10/8/2015	Artificial Neural Networks	**Project Topics**
6	10/13/2015	Artificial Neural Networks	
7	10/15/2015	Support Vector Machines	**Projects Assigned**
8	10/20/2015	Support Vector Machines	
9	10/22/2015	Support Vector Machines	HW2 due - HW3 posted
10	10/27/2015	**Midterm**	
11	10/29/2015	Classification issues: Kernels, Overfitting, Regularization	
12	11/3/2015	Dimensionality Reduction	
13	11/5/2015	Reinforcement Learning	
14	11/10/2015	Decision support: Markov Decision Processes	
15	11/12/2015	Graphical Models - Naïve Bayes	
16	11/17/2015	**Clustering**: K-means - Hierarchical	HW3 due - HW4 posted
17	11/19/2015	Special topics: Deep Learning	
18	11/24/2015	Project Presentation I	**Project Reports Due**
19	11/26/2015	**NO CLASS** (Thanksgiving)	HW4 due
20	12/1/2015	Project Presentation II	
21	12/3/2015	Project Presentation III - Overview	

Homework Posted – Due 10/06/15

UNIVERSITY OF CALIFORNIA, DAVIS
DEPARTMENT OF COMPUTER SCIENCE

ECS 171: Homework Set 1

Instructor: Ilias Tagkopoulos
TAs: Minseung Kim and Ameen Eetemadi
{msgkim, eetemadi}@ucdavis.edu

September 25, 2015

General Instructions: The homework should be submitted electronically through Smartsite. Each submission should be a zip file that includes the following: (a) a report in pdf format ("report_HW1.pdf") that includes your answers to all questions, plots, figures and any instructions to run your code, (b) the matlab/octave code files. Please note: (a) do not include any other files, for instance files that we have provided such as datasets, (b) each function should be written in a seperate file, with the appropriate remarks in the code so it is generally understandable (what it does, how it does it), (c) do not use any toolbox unless is it explicitly allowed in the homework description. Shared/copied code from any source is not allowed, as it is considered plagiarism.

1 OF CARS AND MEN [100PT]

In this exercise, you will investigate the type of relationship that exists between the "miles per gallon" (mpg) rating of a car and several of its attributes. For this task, you will use the "Auto MPG" dataset ("auto-mpg.data" file; 398 cars, 9 features; remove the 6 records with missing values to end up with 392 samples) that is available in the UCI Machine Learning Repository:

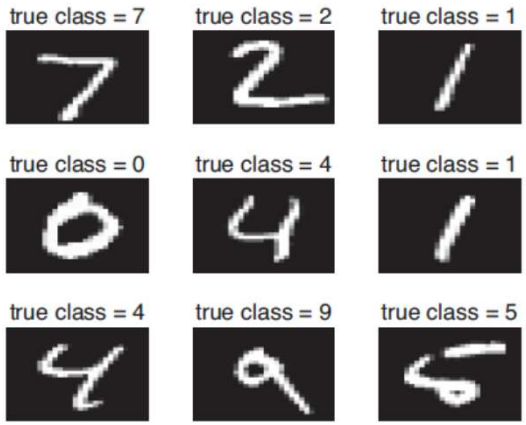
https://archive.ics.uci.edu/ml/datasets/Auto+MPG

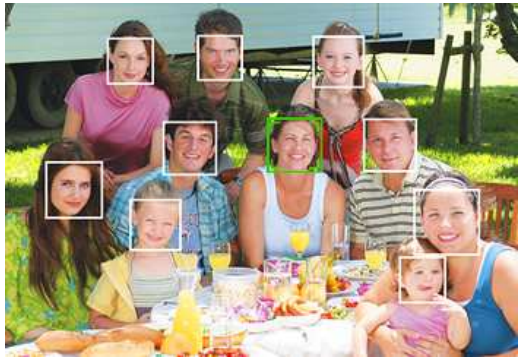
Perform and report (code and results) the following:

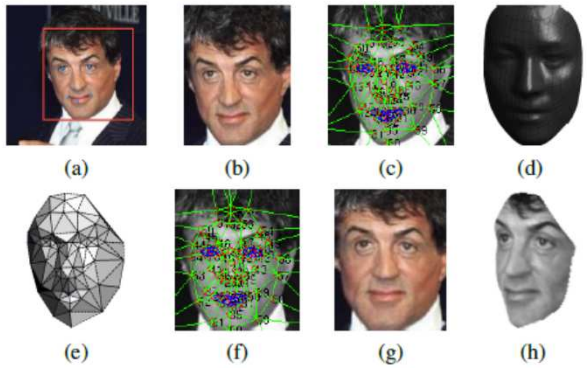
1. Assume that we want to classify the cars into 3 categories: low, medium and high mpg. Find what the threshold for each category should be, so that all samples are divided into three equally-sized bins. [10pt]
2. Create a 2D scatterplot matrix, similar to that of Figure 1.4 in the ML book (K. Murphy, page 6; also available on the lecture 1 slides - the figure with the flowers). You may use any published code to perform this. Which pair from all pair-wise feature combinations is the most informative regarding the three mpg categories? [10pt]
3. Write a linear regression solver that can accomodate polynomial basis functions on a single variable. Your code should use the Ordinary Least Squares (OLS) estimator which is also the Maximum-likelihood estimator for this problem (you will have to code it from scratch). [20p]
4. Split the dataset in the first 280 samples for training and the rest 112 samples for testing. Use your solver to regress for 0th to 4th order polynomial on a single independent variable (feature) each time by using mpg as the dependent variable. Report (a) the training and (b) the testing mean squared errors for each variable individually (except the "car name" string variable, so a total of 7 features that are independent variables). Plot the lines and data for the testing set, one plot per variable (so 5 lines in each plot, 7 plots total). Which polynomial order performs the best in the test set? Which feature is the most informative regarding mpg consumption in that case? [20pt]
5. Modify your solver to be able to handle second order polynomials of all 8 independent variables simultaneously (i.e. 15 terms). Regress with 0th, 1st and 2nd order and report (a) the training and (b) the testing mean squared error. Use the same 280/112 split as before. [20pt]
6. Modify your solver to allow for logistic regression (1st order) and report the training/testing mean squared error, as before. [10pt]
7. If a USA manufacturer (origin 1) had considered to introduce a model in 1980 with the following characteristics: 6 cylinders, 300 cc displacement, 170 horsepower, 3600 lb weight, 9 m/sec^2 acceleration, what is the MPG rating that we should have expected? In which mpg category (low,medium,high mpg) would it belong? Use second-order, multi-variate polynomial and logistic regression. [10pt]
8. Predict the mpg of the vehicle on the photo. Clearly state your assumptions and how your reached to that result. [3pt bonus]

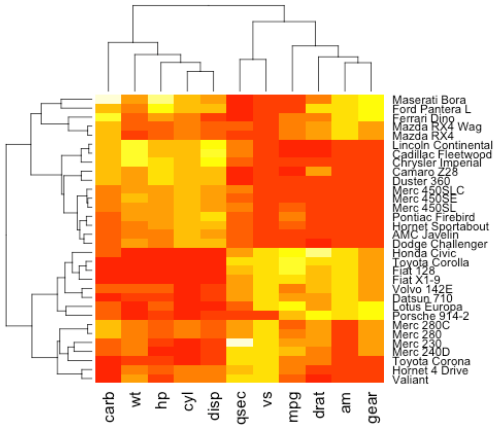
GOOD LUCK!

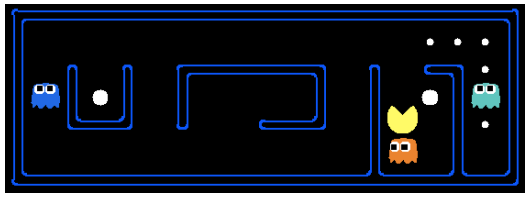
From last lecture: What are the different Learning problems?

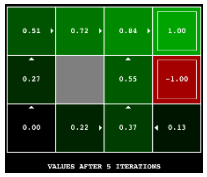


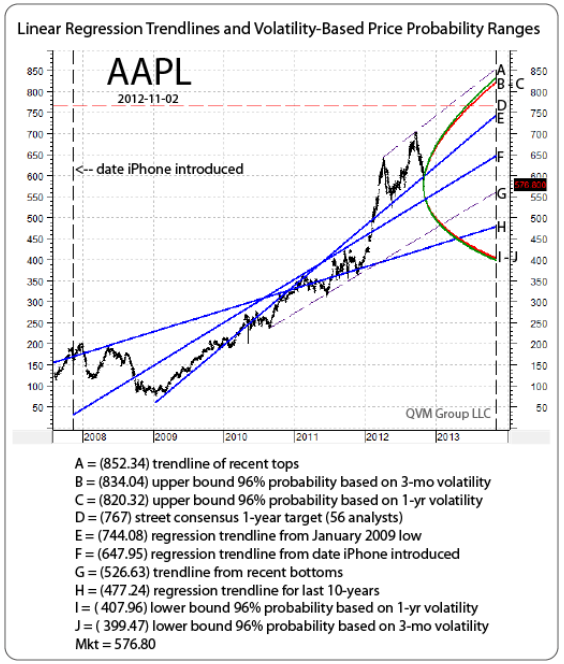












General overview of a machine learning workflow

Step 1. Get enough data!

Dataset

Step 2. Do all of the data samples have labels?

$$\begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix} \begin{matrix} y_1 \\ \dots \\ y_n \end{matrix}$$

Yes

No

$$\begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}$$

SUPERVISED LEARNING

Step 3: The task is to predict a continuous variable, assign a new sample to a class, or perform an optimal action?

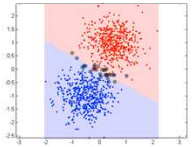
Assign to a class

Predict continuous variable

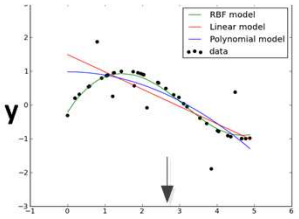
Perform optimal actions

- Bayesian Classification (Naïve Bayes)
- Linear Discriminant Analysis
- Artificial Neural Networks
- Decision Trees
- Support Vector Machines

CLASSIFICATION

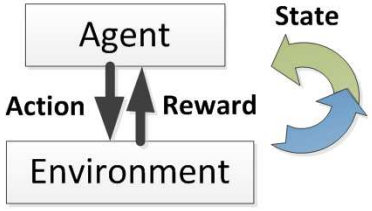


REGRESSION



Linear, polynomial, logistic, ...

REINFORCEMENT LEARNING (*)

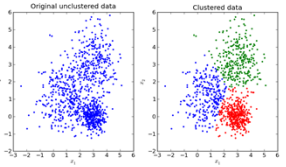


Markov Decision Process (MDP), POMDP, Q-learning, ...

UNSUPERVISED LEARNING

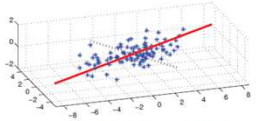
Step 3: The task is to cluster data together, find latent factors or complete missing data?

Clustering



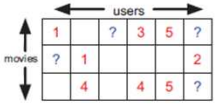
- K-means
- Hierarchical clustering
- SOM

Dimensionality Reduction



- PCA
- ICA

Missing Data

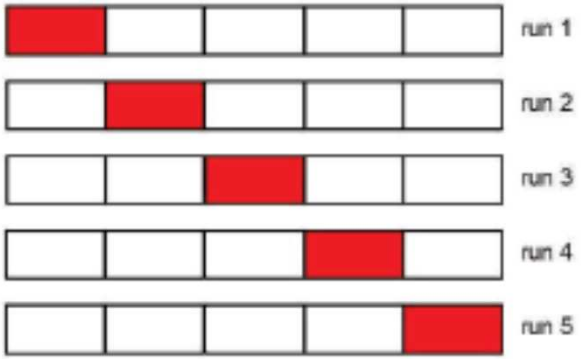


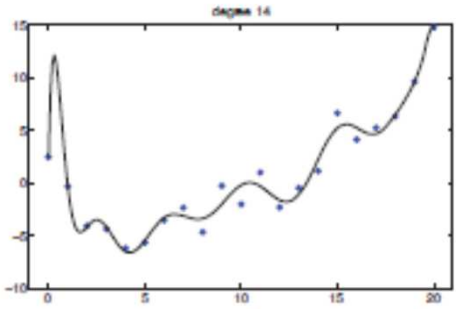
- Collaborative filtering
- Market Basket analysis
- ...

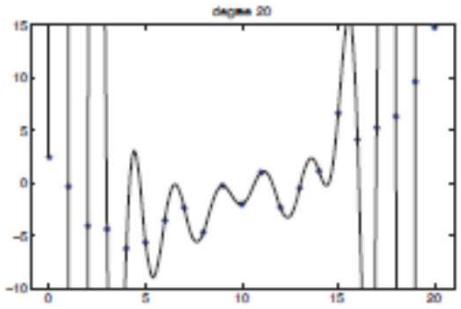
Some challenges in Machine Learning

Other issues:

- Linearity
- Dataset issues: size and structure
- Independence
- Curse of Dimensionality
- Feature Selection
- Overfitting and Noise
- Bias – variance trade-off
- Evaluation







PART I: Regression

"Those who do not remember the past are condemned to repeat it."
George Santayana, 1905

"Prediction is very difficult, especially if it's about the future."
Niels Bohr, 1970; Markus Ronner, 1918

Recommended sections in the ML book : 1.4.5-1.4.7, 7.1-7.3, 8.1-8.3.3

Regression

- **Curve fitting**: Find the curve/function that has **the best fit** (lowest error) to a series of data points.
  - Interpolation: curve fits exactly the data
  - Smoothing: curve approximately fits the data

- When the goal is to predict a continuous dependent variable (output) from one or more independent variables (input), we use **regression.**

- **Linear regression** is a special case of regression where the model depends linearly on the unknown parameters to be estimated from the data.

- **Ordinary Least squares (OLS)** is a well-known solver of linear regression.

Linear Regression

- **Linear regression:** The response $y(x)$ is a linear function of the input:

$$y(x) = w^T x + \epsilon = \sum_{i=0}^{n} w_i x_i + \epsilon$$

With **weight vector** $w = \begin{bmatrix} w_0 \\ \cdots \\ w_n \end{bmatrix}$ and **input vector** $x = \begin{bmatrix} 1 \\ x_1 \\ \cdots \\ x_n \end{bmatrix}$

The term $w^T x$ represents their inner or scalar product and ϵ is their **residual error** (i.e. difference between predicted and true response). The weight w_0 is often called **intercept** or **bias.**

Linear Regression and Conditional Probabilities

- **When does linear regression provides perfect answers?**
  - When the residual error ϵ is zero.
  - In most cases this is not the case. As such, we model the error following a distribution.
  - If we have no other information, a Gaussian (normal) distribution with zero mean and variance σ^2 is the most obvious choice:

$$\epsilon \sim N(0, \sigma^2)$$

- Copying from the previous slide, we have:

Observed output = function + noise

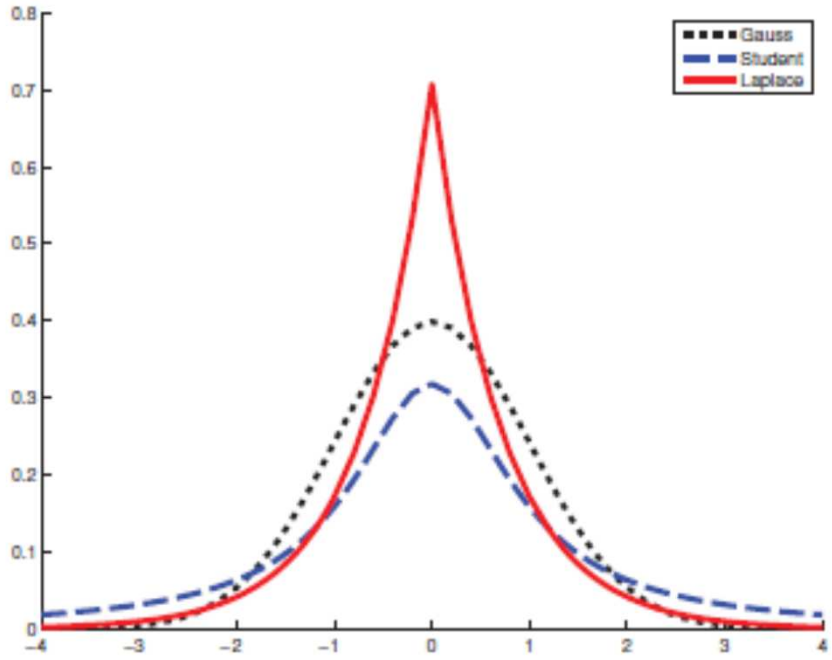
$$y(x) = w^T x + \epsilon$$

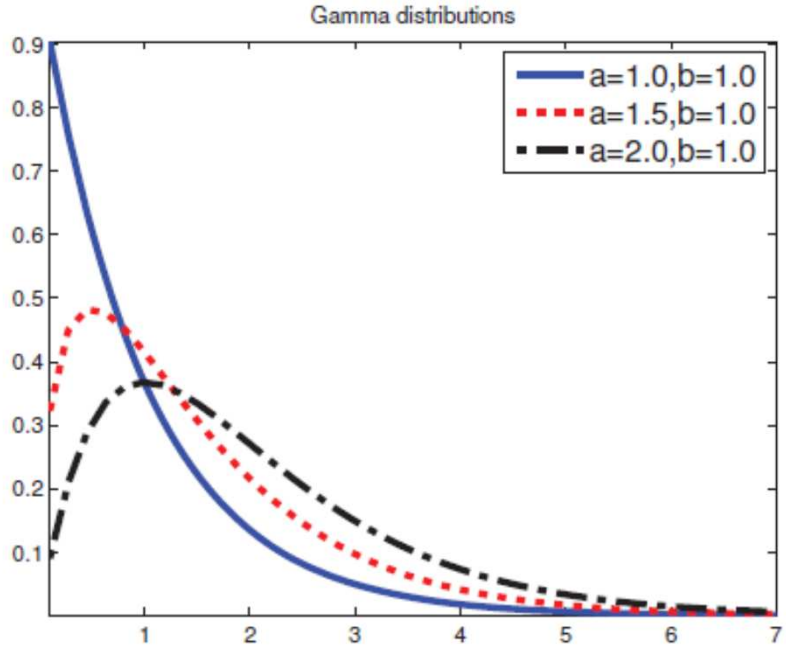
- In this case, $y(x)$ will also be normally distributed, with mean $w^T x$ (why?) and variance σ^2 (as an approximation – to be exact, it is a function of x). We can express this as a conditional probability density, with model parameters $\theta = (w, \sigma^2)$:

$$p(y|x, \theta) = N(y|w^T x, \sigma^2)$$

A note on distributions

- **While we often model assuming a Gaussian distribution, this is not our only option.**
  - For example:





$$\mathcal{N}(x|\mu,\sigma^2) \triangleq \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

$$\mathrm{Ga}(T|\text{shape} = a, \text{rate} = b) \triangleq \frac{b^a}{\Gamma(a)} T^{a-1} e^{-Tb}$$

$$\Gamma(x) \triangleq \int_0^\infty u^{x-1} e^{-u} du$$

Maximum Likelihood Estimation for Linear Regression

- Great, we have a probabilistic formulation for out response variable. If we can estimate the model parameters θ, we are done!
- To do so, we need to build a **MAXIMUM LIKELIHOOD ESTIMATOR (MLE)** to maximize the likelihood, i.e. the probability that we observe the data, given the model:

$$\theta \triangleq \underset{\theta}{\operatorname{argmax}}\, p(D|\theta)$$

- Or equivalently, we can maximize the log of the likelihood $l(\theta)$, as log is a strictly monotonically increasing function and more convenient to use:

$$\theta \triangleq \underset{\theta}{\operatorname{argmax}}\, logp(D|\theta)$$

- If we can assume that we have M training samples that are independent and identically distributed (i.i.d.), then we can treat each probability independently, i.e.

$$l(\theta) \triangleq logp(D|\theta) = \sum_{i=1}^{M} \log p(y^{(i)}|x^{(i)}, \theta)$$

Maximum Likelihood Estimation for Linear Regression

If we plug in the conditional probabilities for the response variable, we have:

$$l(\theta) = \sum_{i=1}^{M} \log\left[\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{1}{2}} e^{-\frac{(y^{(i)} - w^T x^{(i)})^2}{2\sigma^2}}\right]$$

Which can be decomposed to

$$l(\theta) = -\frac{M}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{M}(y^{(i)} - w^T x^{(i)})^2$$

Actually the term:

$$RSS(w) \triangleq \sum_{i=1}^{M}\left(y^{(i)} - w^T x^{(i)}\right)^2 = \|\epsilon\|_2^2$$

Is called the residual sum of squares (RSS), the sum of squared errors (SSE), the l_2 norm of the residual errors and its mean is the mean squared error (MSE = SSE/N). Since the MLE minimizes the RSS, this method is called **Ordinary Least Squares (OLS).**

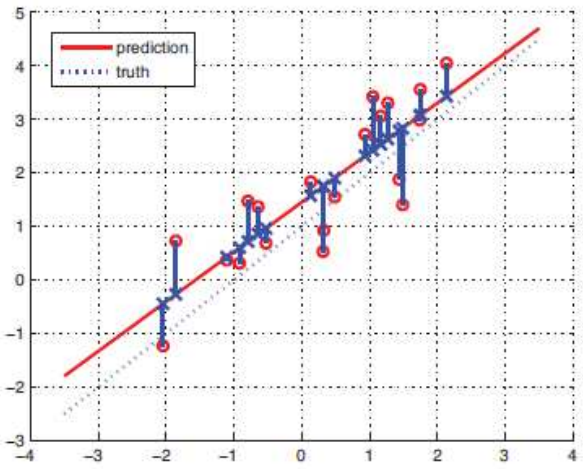
Maximum Likelihood Estimation for Linear Regression

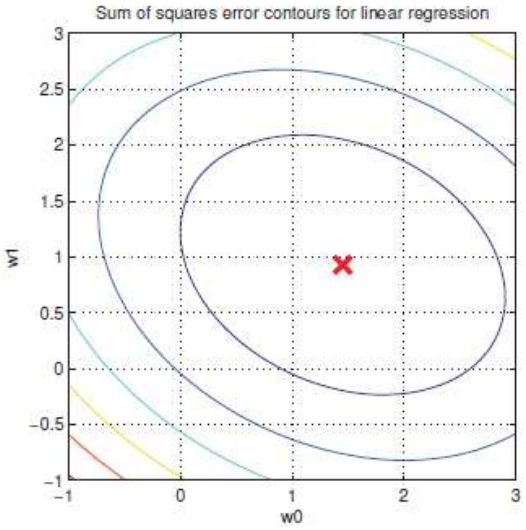
In next lecture we will see that the solution of the OLS problem is given by:

$$w_{OLS} = (X^T X)^{-1} X^T y$$

Bear in mind that instead of maximizing $l(\theta)$, we can minimize the negative log likelihood NLL:

- $NLL(\theta) \triangleq -logp(D|\theta) = -\sum_{i=1}^{M} \log p(y^{(i)}|x^{(i)}, \theta)$



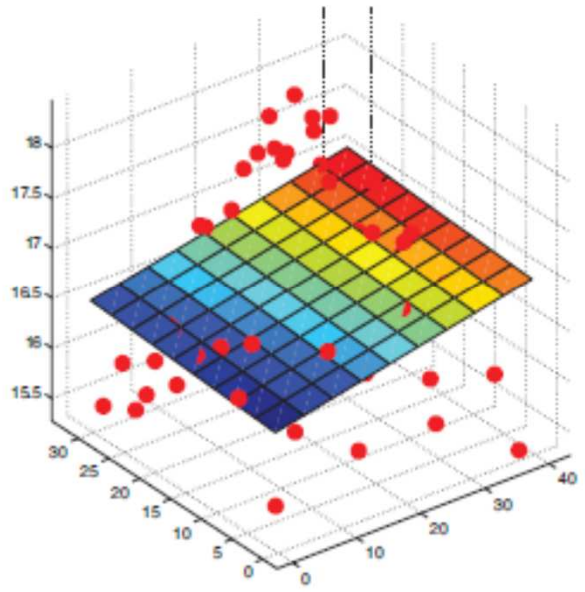


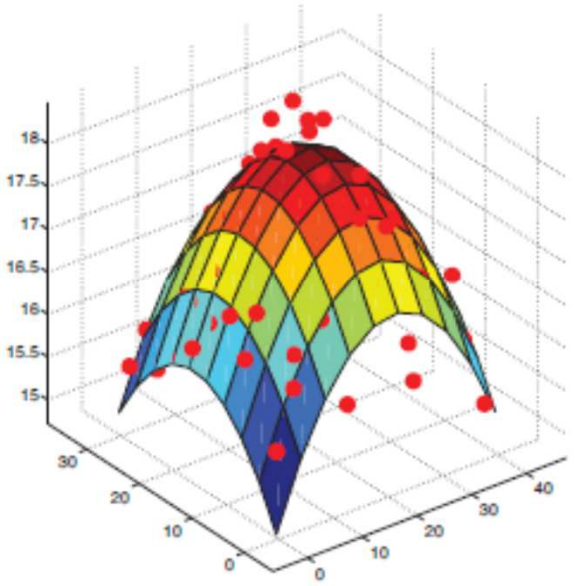
Maximum Likelihood Estimation for Linear Regression

We can also use linear regression to model non-linear relationships of the inputs, by using the basis function expansion $\varphi(x)$:

$$p(y|x,\theta) = N(y|w^T\varphi(x),\sigma^2)$$

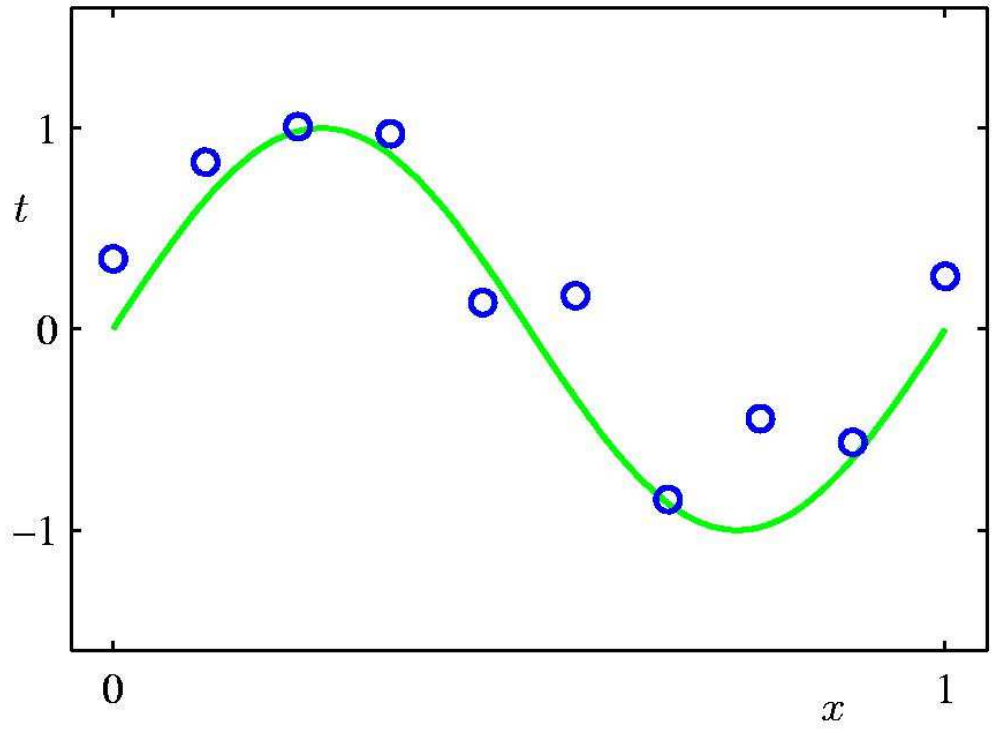
For example $\varphi(x)$ can be the vector $[1, x, \ldots, x^d]$.





Overfitting, polynomial order and sample size

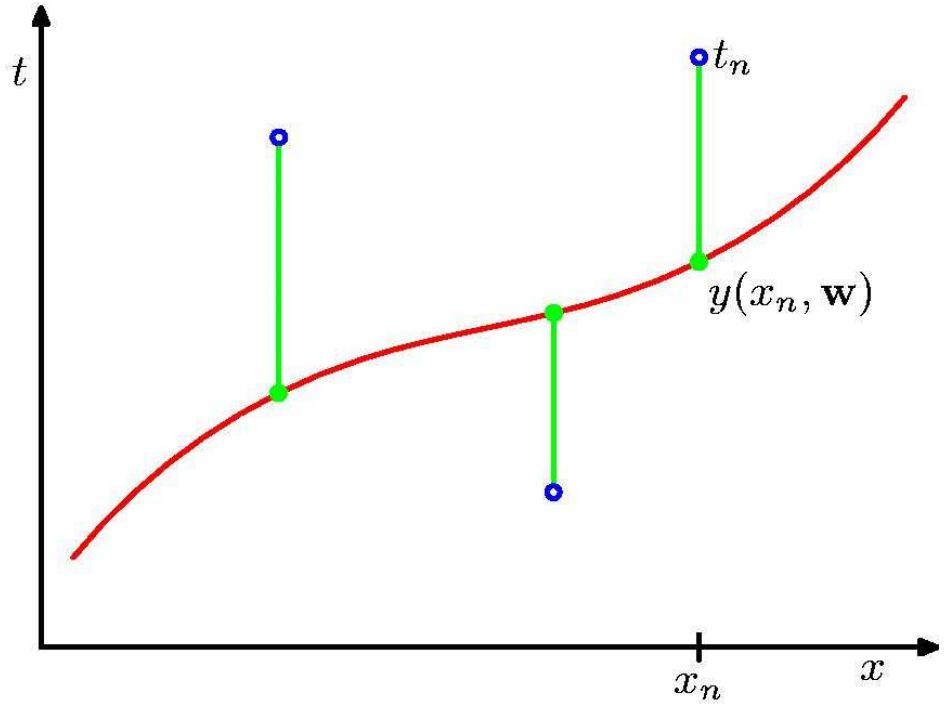
Regression and Polynomial Curve Fitting



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

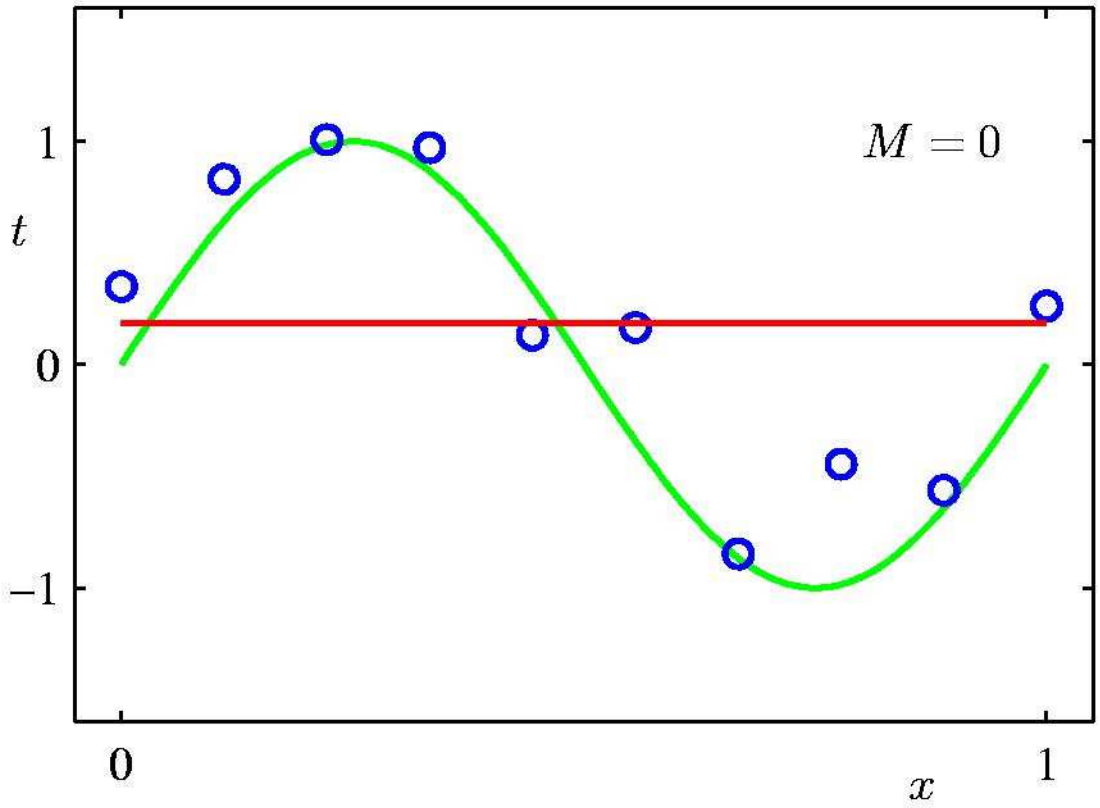
Note the difference from previous slides: *N* were the number of features/regression terms and *M* were the number of training samples. Here it is the inverse.

Regression and Sum-of-Squares Error Function

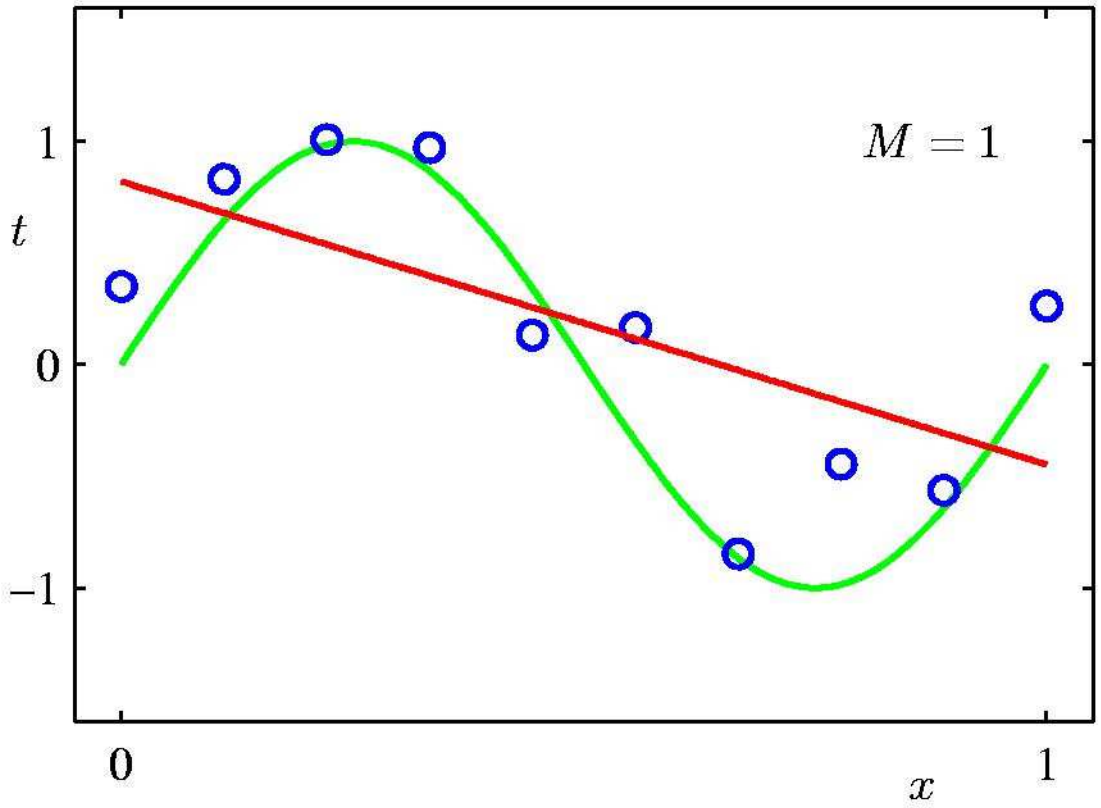


$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

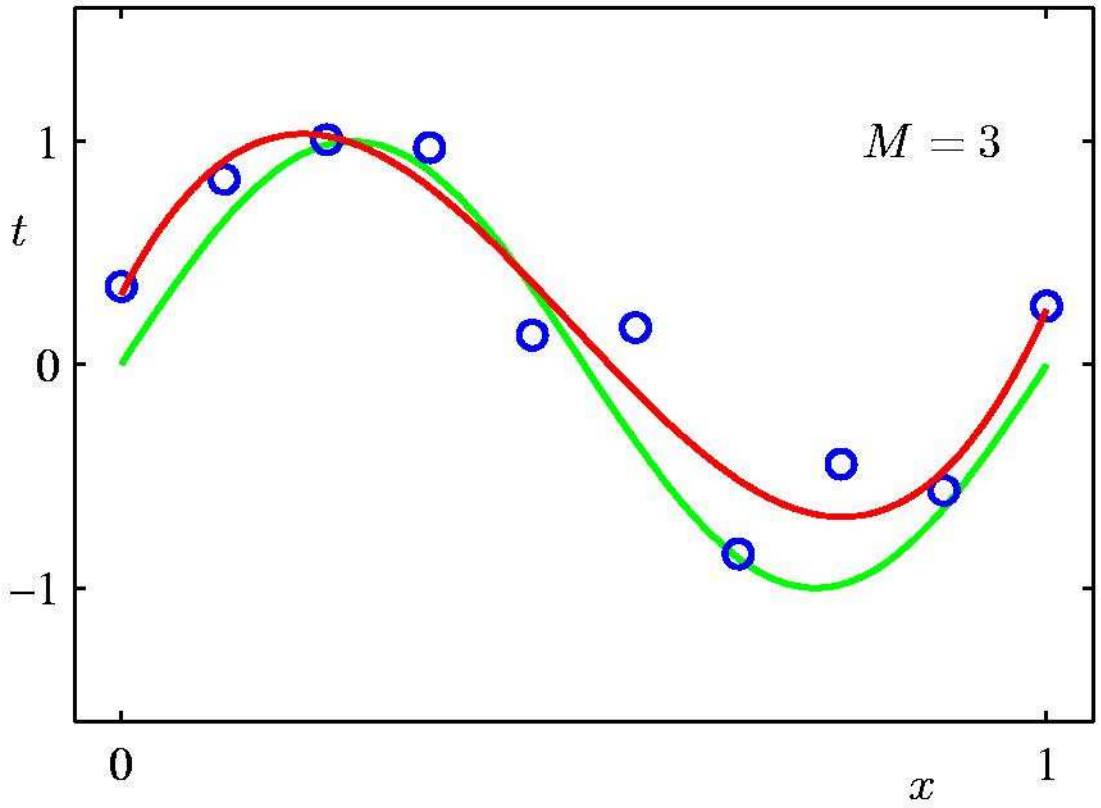
0^{th} Order Polynomial



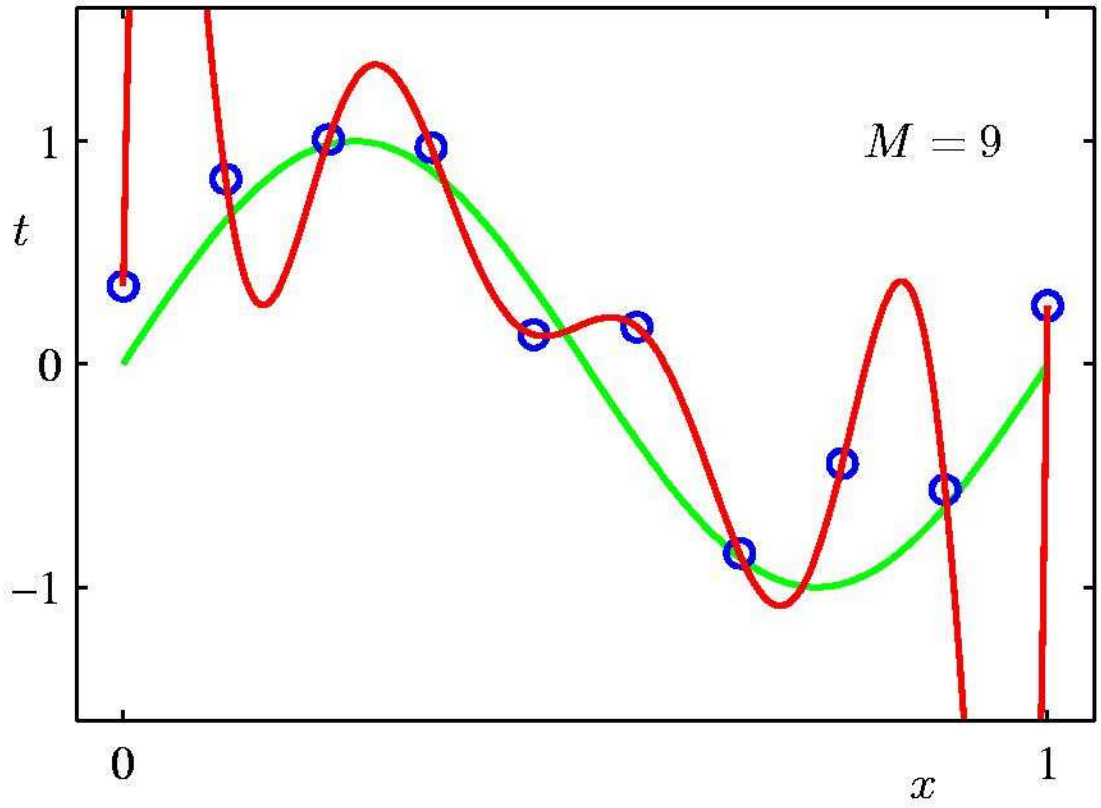
Effect of the polynomial order



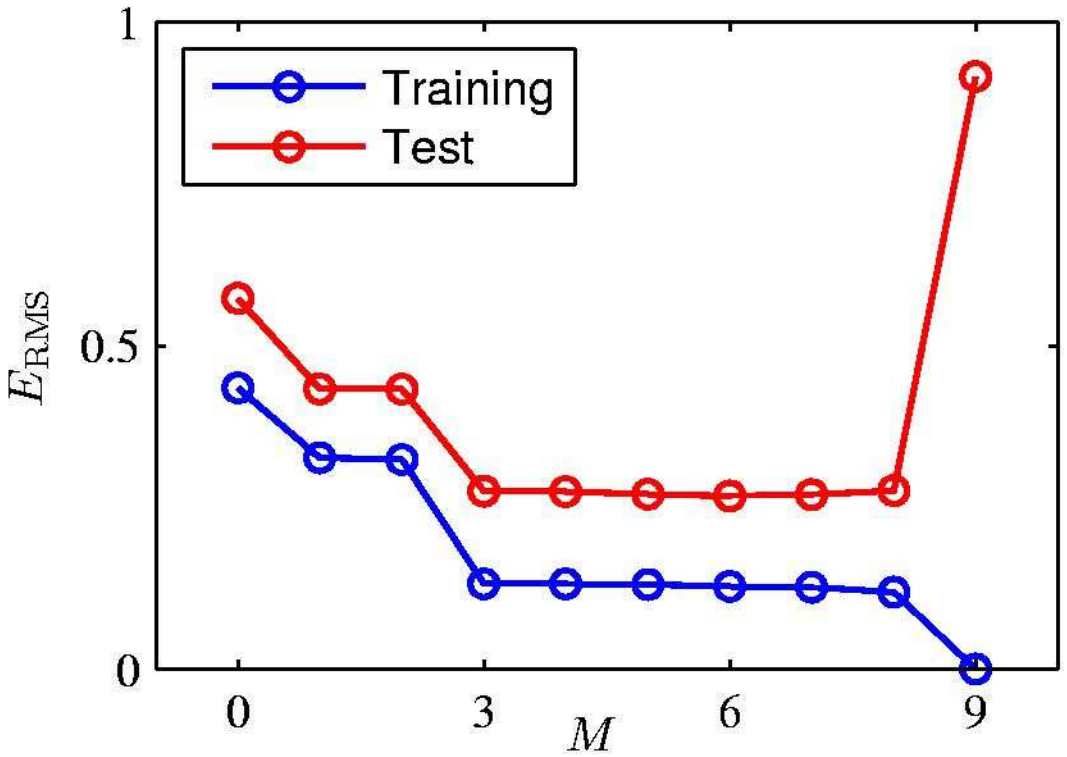
Effect of the polynomial order



Effect of the polynomial order



Zero Training Error



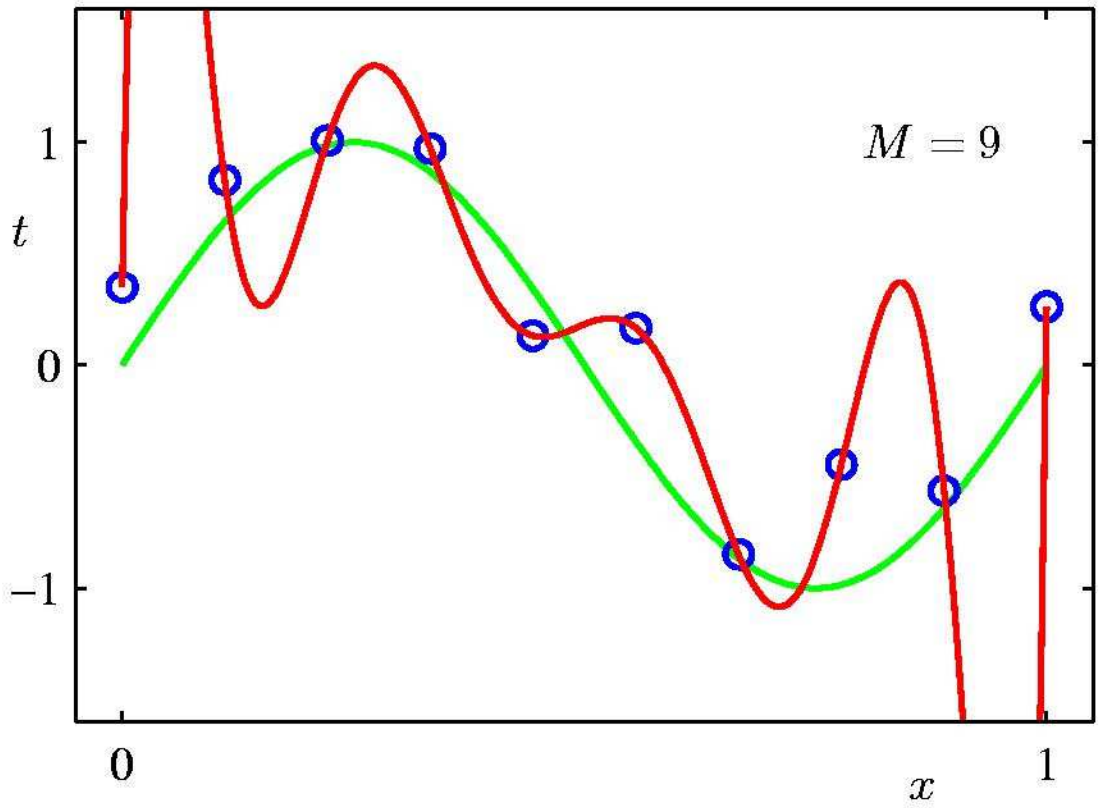
Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$

Overfitting (coefficients)

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

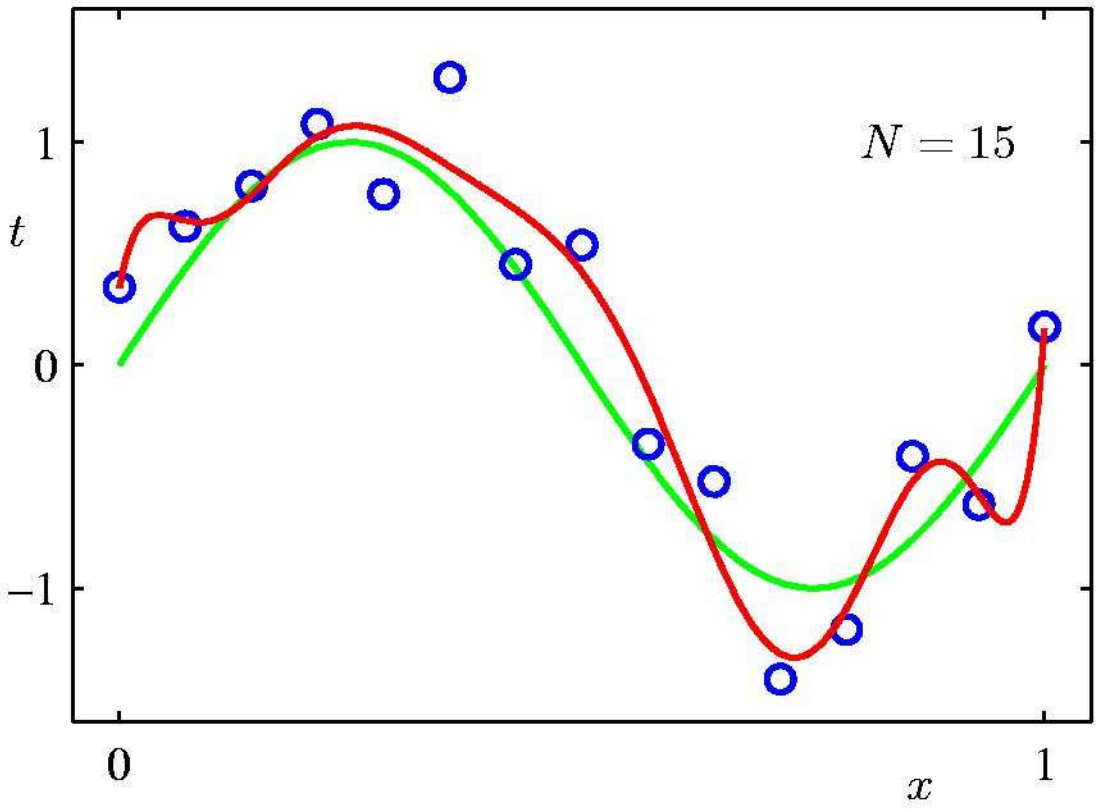
Pattern recognition and matching: Effect of data set size

9th Order Polynomial



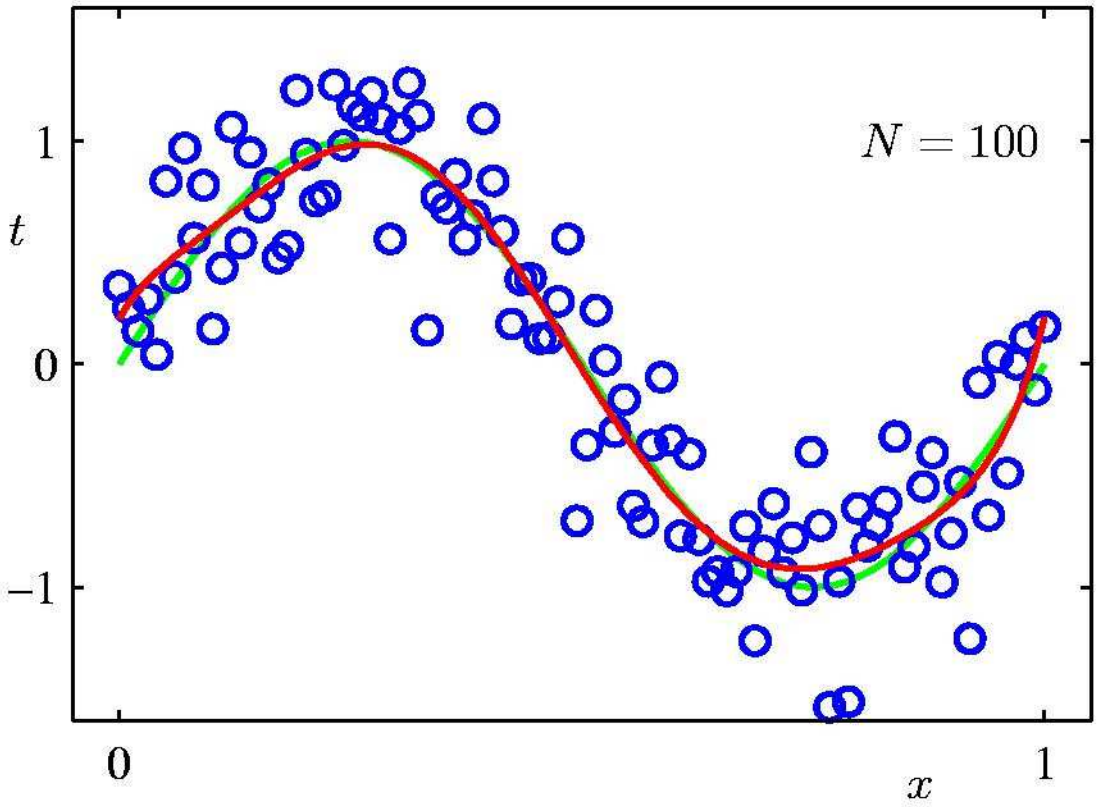
Pattern recognition and matching: Effect of dataset size

9th Order Polynomial



Pattern recognition and matching: Effect of dataset size

9th Order Polynomial



End of Lecture 2

Next time: OLS solution for Linear Regression and Gradient Descent