# Advanced Machine Learning
# Lecture 1

## Introduction

**15.10.2012**

**Bastian Leibe**
**RWTH Aachen**
http://www.vision.rwth-aachen.de/

**leibe@vision.rwth-aachen.de**

# Organization

- Lecturer
  - Prof. Bastian Leibe (leibe@vision.rwth-aachen.de)
- Teaching Assistant
  - Patrick Sudowe (sudowe@vision.rwth-aachen.de)
- Course webpage
  - http://www.vision.rwth-aachen.de/teaching/
  - Slides will be made available on the webpage
  - There is also an L2P electronic repository
- Please subscribe to the lecture on the Campus system!
  - Important to get email announcements and L2P access!

# Language

- Official course language will be English
  - If at least one English-speaking student is present.
  - If not… you can choose.
- However…
  - Please tell me when I'm talking too fast or when I should repeat something in German for better understanding!
  - You may at any time ask questions in German!
  - You may turn in your exercises in German.
  - You may take the oral exam in German.

# Relationship to Previous Courses

- Lecture *Machine Learning* (past summer semester)
  - Introduction to ML
  - Classification
  - Graphical models
- This course: *Advanced Machine Learning*
  - Natural continuation of ML course
  - Deeper look at the underlying concepts
  - But: will try to make it accessible also to newcomers
  - *Quick poll: Who hasn't heard the ML lecture?*
- One-time only course
  - Lecture will be held in this format only once
  - After this semester, will reorganize material into ML1 & ML2

# Organization

- Structure: 3V (lecture) + 1Ü (exercises)
  - 6 EECS credits
  - Part of the area "Applied Computer Science"

- Place & Time

| | | | |
|---|---|---|---|
| Lecture: | Mon | 17:30 - 19:00 | room UMIC 025 |
| Lecture/Exercises: | Wed | 10:00 - 11:30 | room UMIC 025 |

- Exam
  - Oral or written exam, depending on number of participants
  - Towards the end of the semester, there will be a proposed date

# Course Webpage



## Tentative Schedule

| Date | Topic | Content | Slides | Related Material |
|---|---|---|---|---|
| 03.04.11 | *no class* | - | - | - |
| 05.04.12 | **Introduction** | Introduction, Probability Theory, Bayes Decision Theory, Minimizing Expected Loss | pdf, fullpage | Bishop Ch. 1.1, 1.2.1-1.2.3, 1.5.1-1.5.4 |
| 10.04.12 | *Exercise 0* | *Intro Matlab* | | - |
| 12.04.12 | **Prob. Density Estimation I** | Nonparametric Methods, Histograms, Kernel Density Estimation, Parametric Methods, Gaussian Distribution, Maximum Likelihood, Bayesian Learning, Bias-Variance Problem | | Bishop Ch. 2.5, 1.2.4, 2.3.1-2.3.4 |
| 17.04.12 | **Prob. Density Estimation II** | Mixture of Gaussians, k-Means Clustering, EM-Clustering, EM Algorithm | | Bishop chapter 9, original Dempster&Laird EM paper, Bilmes' EM tutorial |
| 19.04.12 | **Linear Discriminant Functions** | Linear Discriminant Functions, Least-squares Classification, Generalized Linear Models | | Bishop chapter 4.1 |

**http://www.vision.rwth-aachen.de/teaching/**

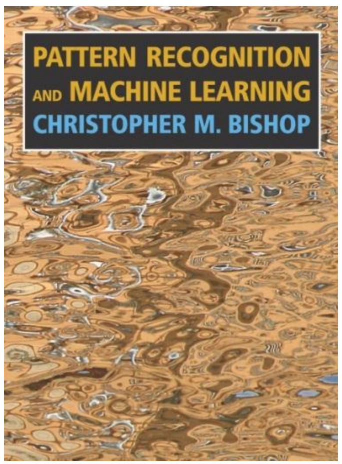
# Exercises and Supplementary Material

- Exercises
  - Typically 1 exercise sheet every 2 weeks.
  - Pen & paper and Matlab based exercises
  - Hands-on experience with the algorithms from the lecture.
  - Send your solutions the night before the exercise class.

- Supplementary material
  - Research papers and book chapters
  - Will be provided on the webpage.

# Textbooks

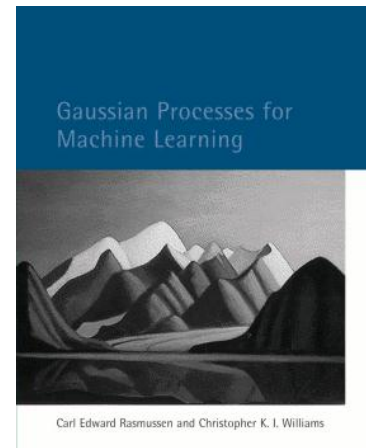
- Most lecture topics will be covered in Bishop's book.
- Some additional topics can be found in Rasmussen & Williams.




Christopher M. Bishop
Pattern Recognition and Machine Learning
Springer, 2006

(available in the library's "Handapparat")

Carl E. Rasmussen, Christopher K.I. Williams
Gaussian Processes for Machine Learning
MIT Press, 2006
(also available online: http://www.gaussianprocess.org/gpml/)




- Research papers will be given out for some topics.
  - Tutorials and deeper introductions.
  - Application papers

# How to Find Us

- Office:
  - UMIC Research Centre
  - Mies-van-der-Rohe-Strasse 15, room 124

- Office hours
  - If you have questions to the lecture, come to Patrick or me.
  - My regular office hours are Tue 15:30-16:30 (additional slots are available upon request)
  - Send us an email before to confirm a time slot.

***Questions are welcome!***

# Machine Learning

- Statistical Machine Learning
  - Principles, methods, and algorithms for learning and prediction on the basis of past evidence

- Already everywhere
  - Speech recognition (e.g. speed-dialing)
  - Computer vision (e.g. face detection)
  - Hand-written character recognition (e.g. letter delivery)
  - Information retrieval (e.g. image & video indexing)
  - Operation systems (e.g. caching)
  - Fraud detection (e.g. credit cards)
  - Text filtering (e.g. email spam filters)
  - Game playing (e.g. strategy prediction)
  - Robotics (e.g. prediction of battery lifetime)

Slide credit: Bernt Schiele

# What Is Machine Learning Useful For?



**Automatic Speech Recognition**

RWTH AACHEN UNIVERSITY

# What Is Machine Learning Useful For?


**Computer Vision**
**(Object Recognition, Segmentation, Scene Understanding)**

# What Is Machine Learning Useful For?

Google Search: Unsupervised Learning http://www.google.com/search?q=Unsupervised+Learning&sourceid=fir...

Web Images Groups News Froogle more »

Unsupervised Learning Search Advanced Search Preferences

**Web** Results **1** - **10** of about **150,000** for **Unsupervised Learning**. (**0.27** seconds)

Mixture modelling, Clustering, Intrinsic classification ...
Mixture Modelling page. Welcome to David Dowe's clustering, mixture modelling and **unsupervised learning** page. Mixture modelling (or ...
www.csse.monash.edu.au/~dld/mixture.modelling.page.html - 26k - 4 Oct 2004 - Cached - Similar pages

ACL'99 Workshop -- **Unsupervised Learning** in Natural Language ...
PROGRAM. ACL'99 Workshop **Unsupervised Learning** in Natural Language Processing. University of Maryland June 21, 1999. Endorsed by SIGNLL ...
www.ai.sri.com/~kehler/unsup-acl-99.html - 5k - Cached - Similar pages

**Unsupervised learning** and Clustering
cgm.cs.mcgill.ca/~soss/cs644/projects/wijhe/ - 1k - Cached - Similar pages

NIPS*98 Workshop - Integrating Supervised and **Unsupervised** ...
NIPS*98 Workshop "Integrating Supervised and **Unsupervised Learning**" Friday, December 4, 1998. ... 4:45-5:30, Theories of **Unsupervised Learning** and Missing Values. ...
www-2.cs.cmu.edu/~mccallum/supunsup/ - 7k - Cached - Similar pages

NIPS Tutorial 1999
Probabilistic Models for **Unsupervised Learning** Tutorial presented at the 1999 NIPS Conference by Zoubin Ghahramani and Sam Roweis. ...
www.gatsby.ucl.ac.uk/~zoubin/NIPStutorial.html - 4k - Cached - Similar pages

Gatsby Course: **Unsupervised Learning** : Homepage
**Unsupervised Learning** (Fall 2000). ... Syllabus (resources page): 10/10 1 - Introduction to **Unsupervised Learning** Geoff project: (ps, pdf). ...
www.gatsby.ucl.ac.uk/~quaid/course/ - 15k - Cached - Similar pages
[ More results from www.gatsby.ucl.ac.uk ]

[PDF] **Unsupervised Learning** of the Morphology of a Natural Language
File Format: PDF/Adobe Acrobat - View as HTML
Page 1. Page 2. Page 3. Page 4. Page 5. Page 6. Page 7. Page 8. Page 9. Page 10. Page 11. Page 12. Page 13. Page 14. Page 15. Page 16. Page 17. Page 18. Page 19 ...
acl.ldc.upenn.edu/J/J01/J01-2001.pdf - Similar pages

**Unsupervised Learning** - The MIT Press
... From Bradford Books: **Unsupervised Learning** Foundations of Neural Computation Edited by Geoffrey Hinton and Terrence J. Sejnowski Since its founding in 1989 by ...
mitpress.mit.edu/book-home.tcl?isbn=026258168X - 13k - Cached - Similar pages

[PS] **Unsupervised Learning** of Disambiguation Rules for Part of
File Format: Adobe PostScript - View as Text
**Unsupervised Learning** of Disambiguation Rules for Part of. Speech Tagging. Eric Brill. 1. ... It is possible to use **unsupervised learning** to train stochastic. ...
www.cs.jhu.edu/~brill/acl-wkshp.ps - Similar pages

The **Unsupervised Learning** Group (ULG) at UT Austin
The **Unsupervised Learning** Group (ULG). What ? The **Unsupervised Learning** Group (ULG) is a group of graduate students from the Computer ...
www.lans.ece.utexas.edu/ulg/ - 14k - Cached - Similar pages

Gooooooooooogle
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

1 of 2 06/10/04 15:44

**Information Retrieval**
**(Retrieval, Categorization, Clustering, ...)**

# What Is Machine Learning Useful For?

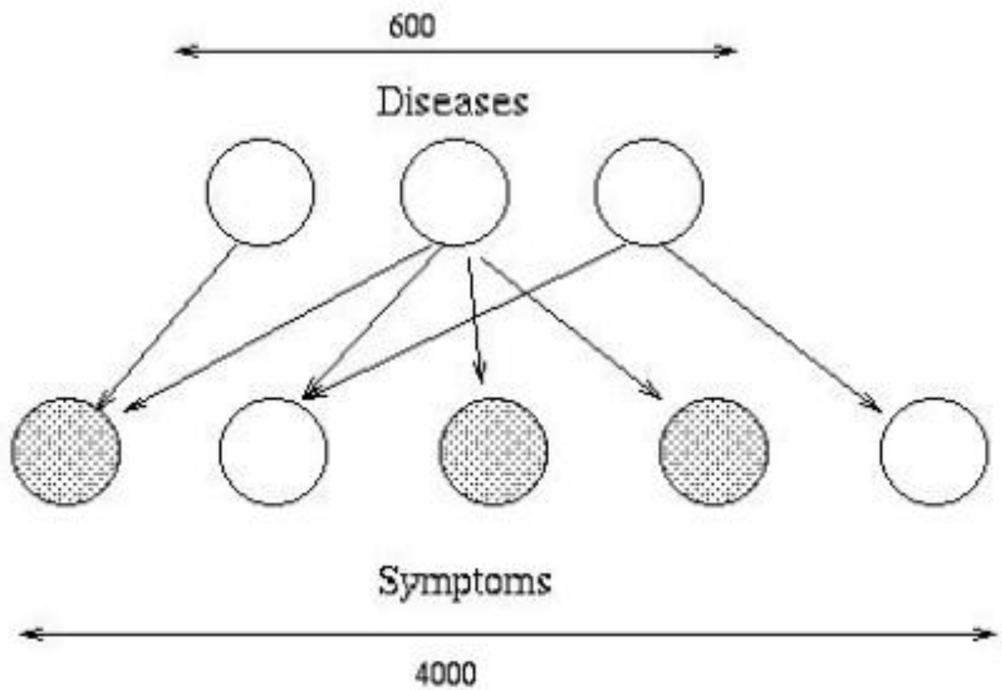

**Financial Prediction**
**(Time series analysis, ...)**



Slide adapted from Zoubin Gharamani

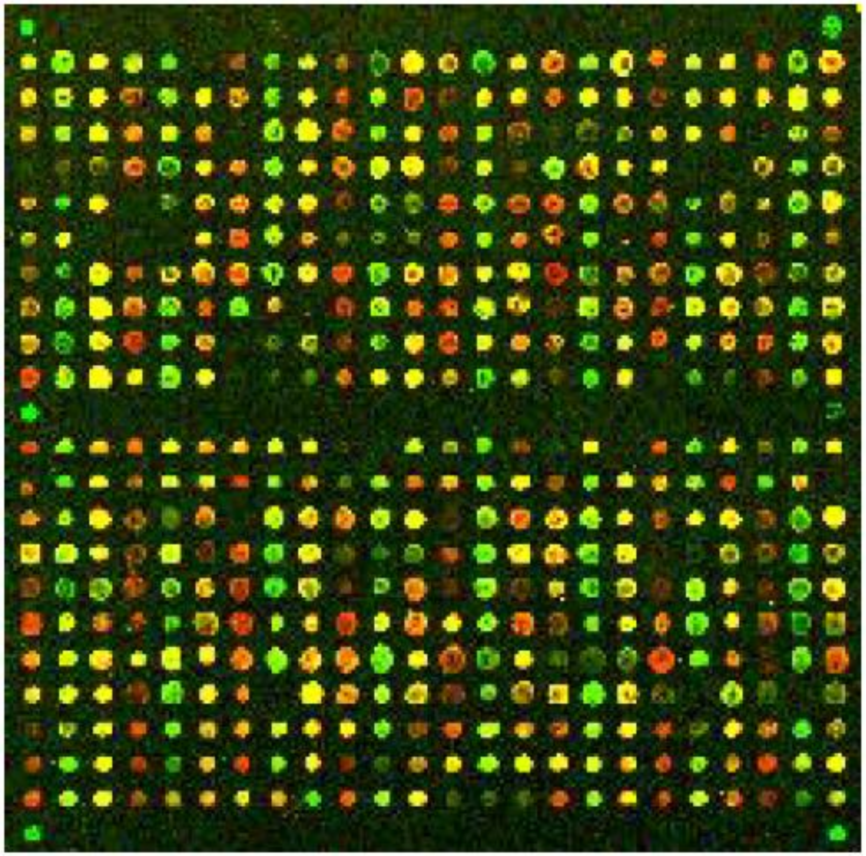
B. Leibe

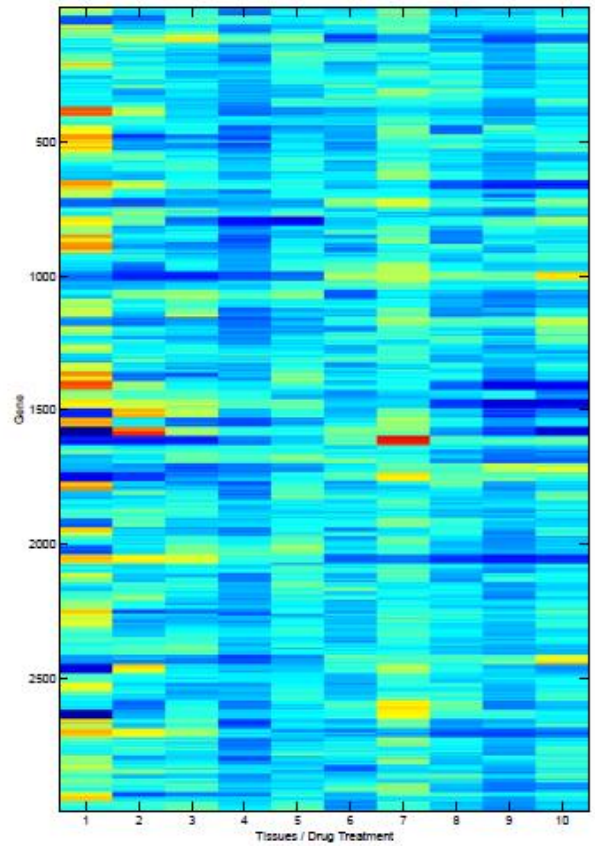
# What Is Machine Learning Useful For?




**Medical Diagnosis**
**(Inference from partial observations)**

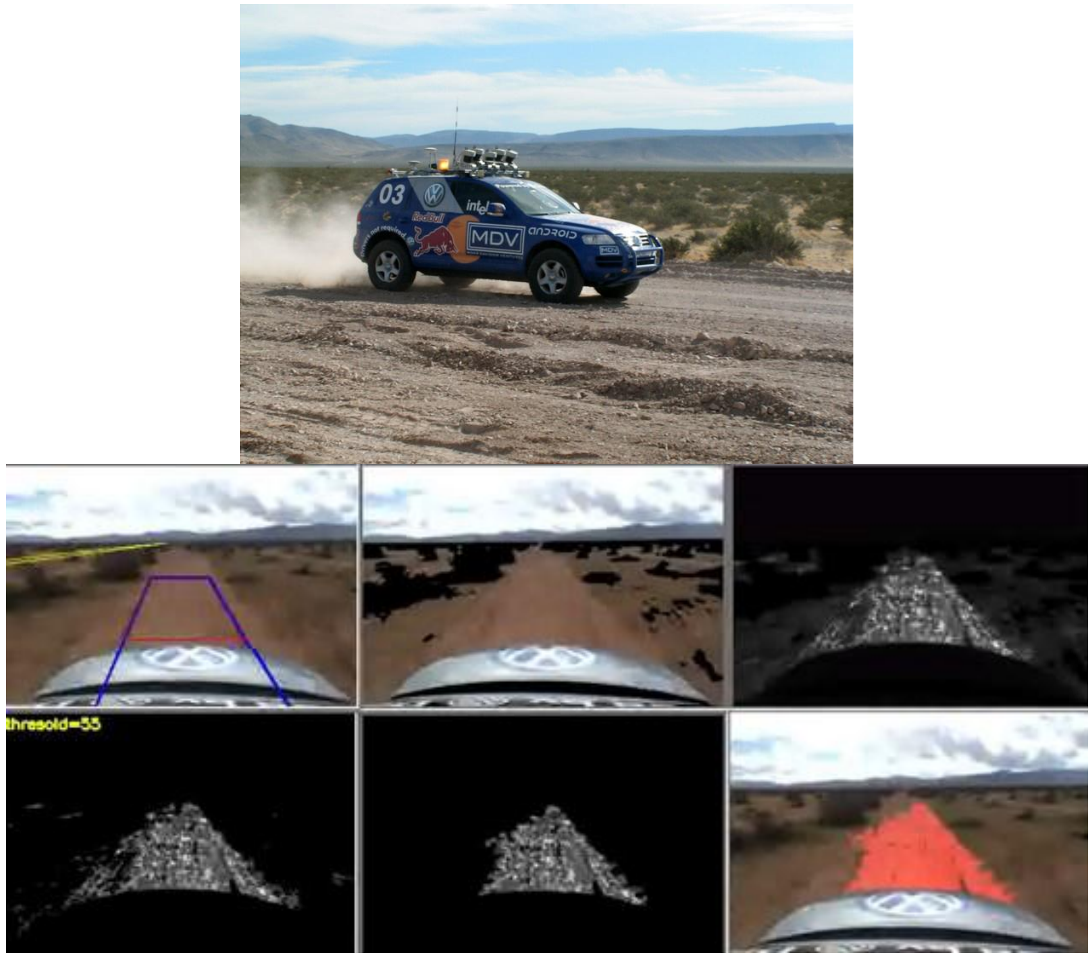
# What Is Machine Learning Useful For?






**Bioinformatics**
**(Modelling gene microarray data,...)**

# What Is Machine Learning Useful For?



**Robotics**
**(DARPA Grand Challenge,...)**

# Machine Learning: Core Questions

- *Learning to perform a task from experience*

- Task
  - Can often be expressed through a mathematical function

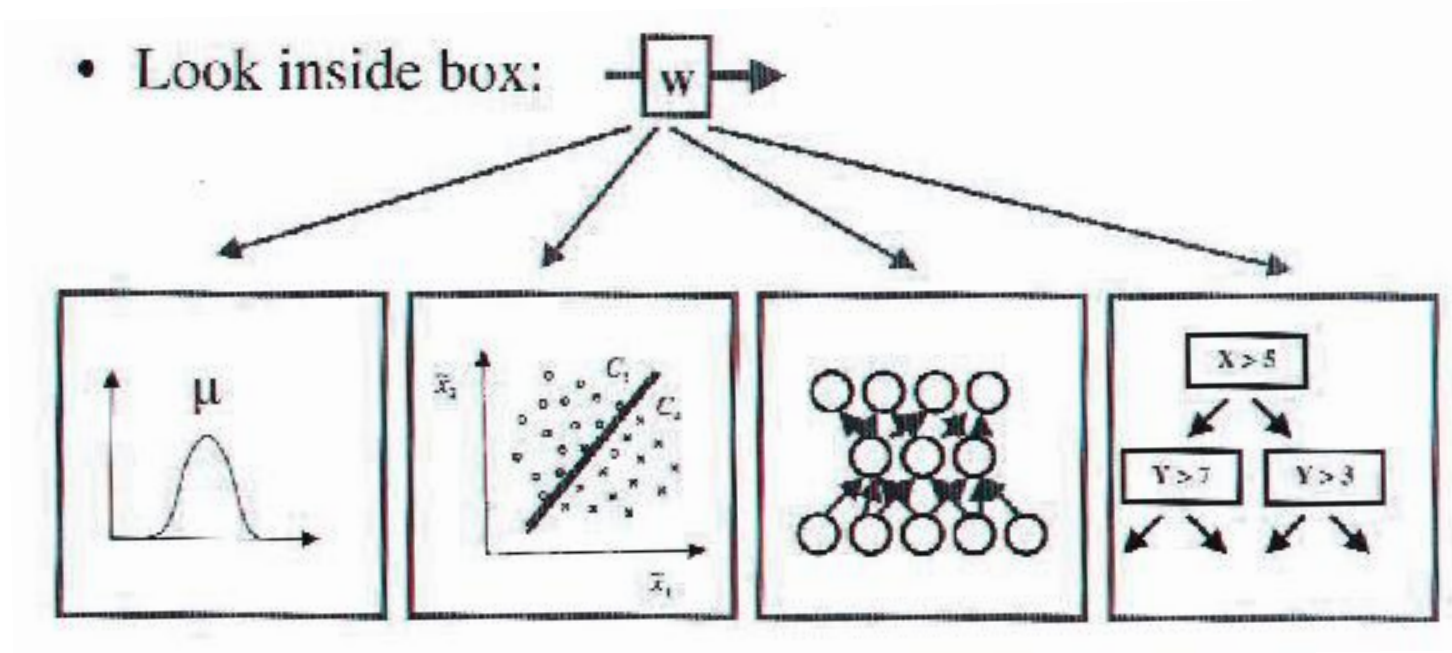
$$y = f(x; w)$$

  - $x$: Input
  - $y$: Output
  - $w$: Parameters (this is what is "learned")

- Classification vs. Regression
  - Regression: continuous $y$
  - Classification: discrete $y$
    - E.g. class membership, sometimes also posterior probability

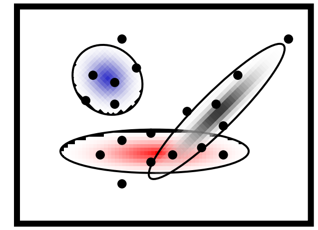
# Machine Learning: Core Questions

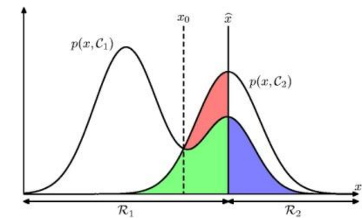
- $y = f(x; w)$
  - $w$: characterizes the family of functions
  - $w$: indexes the space of hypotheses
  - $w$: vector, connection matrix, graph, …



# A Look Back: Lecture *Machine Learning*

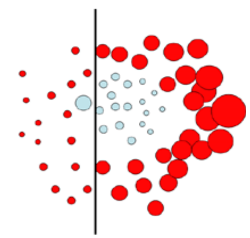
- Fundamentals
  - Bayes Decision Theory
  - Probability Density Estimation

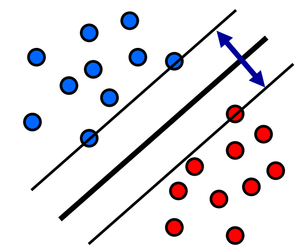




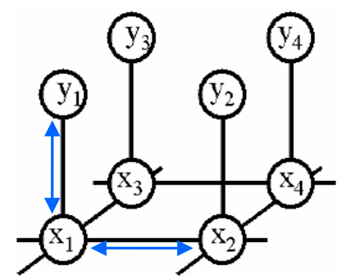

- Classification Approaches
  - Linear Discriminant Functions
  - Support Vector Machines
  - Ensemble Methods & Boosting
  - Randomized Trees, Forests & Ferns

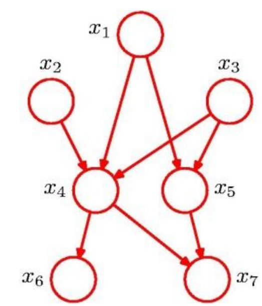
$$f : \mathcal{X} \rightarrow \{0, 1\}$$





- Generative Models
  - Bayesian Networks
  - Markov Random Fields



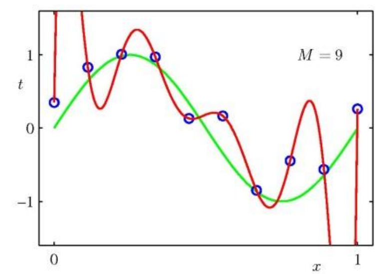



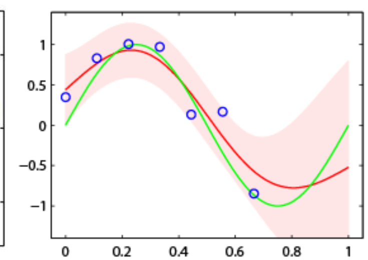
# This Lecture: *Advanced Machine Learning*

Extending lecture *Machine Learning* from last semester…

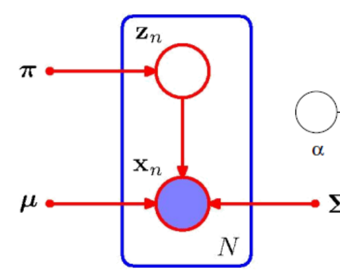
- Regression Approaches
  - Linear Regression
  - Regularization (Ridge, Lasso)
  - Support Vector Regression
  - Gaussian Processes

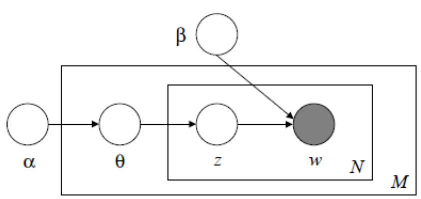
$$f : \mathcal{X} \to \mathbb{R}$$




- Learning with Latent Variables
  - EM and Generalizations
  - Dirichlet Processes




- Structured Output Learning
  - Large-margin Learning

$$f : \mathcal{X} \to \mathcal{Y}$$

# Let’s Get Started…

- Some of you already have basic ML background
  - *Who hasn’t?*

- We’ll start with a gentle introduction
  - I’ll try to make the lecture also accessible to newcomers
  - We’ll review the main concepts before applying them
  - I’ll point out chapters to review from ML lecture whenever knowledge from there is needed/helpful
  - But please tell me when I’m moving too fast (or too slow)

# Topics of This Lecture

- Regression: Motivation
  - Polynomial fitting
  - General Least-Squares Regression
  - Overfitting problem
  - Regularization
  - Ridge Regression
- Recap: Important Concepts from ML Lecture
  - Probability Theory
  - Bayes Decision Theory
  - Maximum Likelihood Estimation
  - Bayesian Estimation
- A Probabilistic View on Regression
  - Least-Squares Estimation as Maximum Likelihood

# Regression

- **Learning to predict a continuous function value**
  - Given: training set $\mathbf{X} = \{x_1, \ldots, x_N\}$ with target values $\mathbf{T} = \{t_1, \ldots, t_N\}$.

  ⇒ Learn a continuous function $y(x)$ to predict the function value for a new input $x$.

- **Steps towards a solution**
  - Choose a form of the function $y(x,\mathbf{w})$ with parameters $\mathbf{w}$.
  - Define an error function $E(\mathbf{w})$ to optimize.
  - Optimize $E(\mathbf{w})$ for $\mathbf{w}$ to find a good solution. (This may involve math).
  - Derive the properties of this solution and think about its limitations.

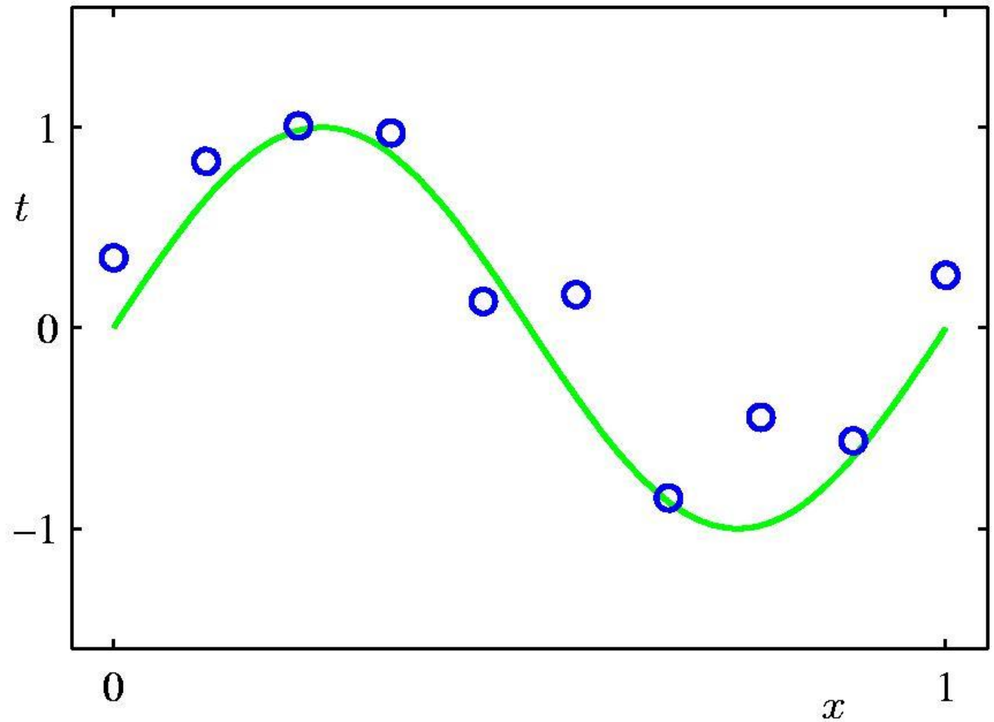
# Example: Polynomial Curve Fitting



- **Toy dataset**
  - Generated by function

    $$f(x) = sin(2\pi x) + \epsilon$$

  - Small level of random noise with Gaussian distribution added (blue dots)




- **Goal: fit a polynomial function to this data**

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

  - Note: Nonlinear function of $x$, but linear function of the $w_j$.



Image source: C.M. Bishop, 2006

# Error Function

- **How to determine the values of the coefficients $\mathbf{w}$?**
  - We need to define an **error function** to be minimized.
  - This function specifies how a deviation from the target value should be weighted.

- **Popular choice: sum-of-squares error**
  - Definition

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

  - We'll discuss the motivation for this particular function later…

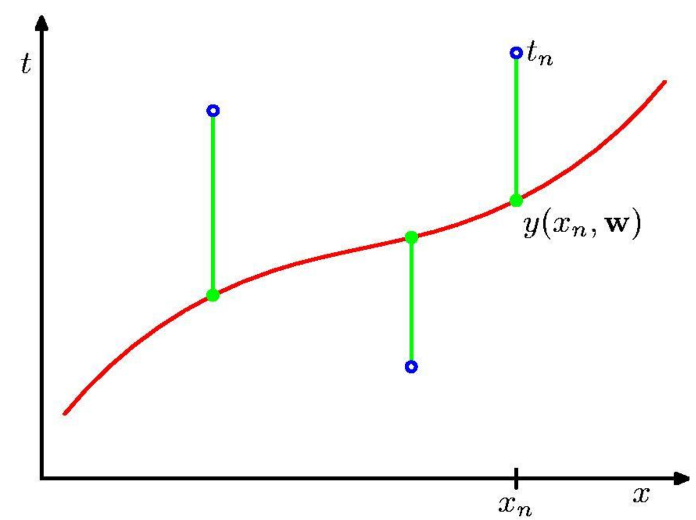



Image source: C.M. Bishop, 2006

# Minimizing the Error

- How do we minimize the error?

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

- Solution (Always!)
  - Compute the derivative and set it to zero.

$$\frac{\partial E(\mathbf{w})}{\partial w_j} = \sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}\frac{\partial y(x_n, \mathbf{w})}{\partial w_j} \stackrel{!}{=} 0$$

  - Since the error is a quadratic function of $\mathbf{w}$, its derivative will be linear in $\mathbf{w}$.

$\Rightarrow$ Minimization has a unique solution.

# Least-Squares Regression

- We have given
  - Training data points: $X = \{\mathbf{x}_1 \in \mathbb{R}^d, \dots, \mathbf{x}_n\}$
  - Associated function values: $T = \{t_1 \in \mathbb{R}, \dots, t_n\}$

- Start with **linear regressor**:
  - Try to enforce $\mathbf{x}_i^T \mathbf{w} + w_0 = t_i, \quad \forall i = 1, \dots, n$
  - One linear equation for each training data point / label pair.

  - **This is the same basic setup used for least-squares classification!**
    - Only the values are now continuous.

# Least-Squares Regression

$$\mathbf{x}_i^T \mathbf{w} + w_0 = t_i, \quad \forall i = 1, \ldots, n$$

- Setup
  - Step 1: Define $\tilde{\mathbf{x}}_i = \begin{pmatrix} \mathbf{x}_i \\ 1 \end{pmatrix}, \quad \tilde{\mathbf{w}} = \begin{pmatrix} \mathbf{w} \\ w_0 \end{pmatrix}$
  - Step 2: Rewrite $\tilde{\mathbf{x}}_i^T \tilde{\mathbf{w}} = t_i, \quad \forall i = 1, \ldots, n$
  - Step 3: Matrix-vector notation

$$\widetilde{\mathbf{X}}^T \tilde{\mathbf{w}} = \mathbf{t} \quad \text{with} \quad \begin{aligned} \widetilde{\mathbf{X}} &= [\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n] \\ \mathbf{t} &= [t_1, \ldots, t_n]^T \end{aligned}$$

  - Step 4: Find least-squares solution

$$\|\widetilde{\mathbf{X}}^T \tilde{\mathbf{w}} - \mathbf{t}\|^2 \to \min$$

  - Solution: $\tilde{\mathbf{w}} = (\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T)^{-1}\widetilde{\mathbf{X}}\mathbf{t}$

Slide credit: Bernt Schiele

# Regression with Polynomials

- How can we fit arbitrary polynomials using least-squares regression?
  - We introduce a feature transformation (as before in ML).

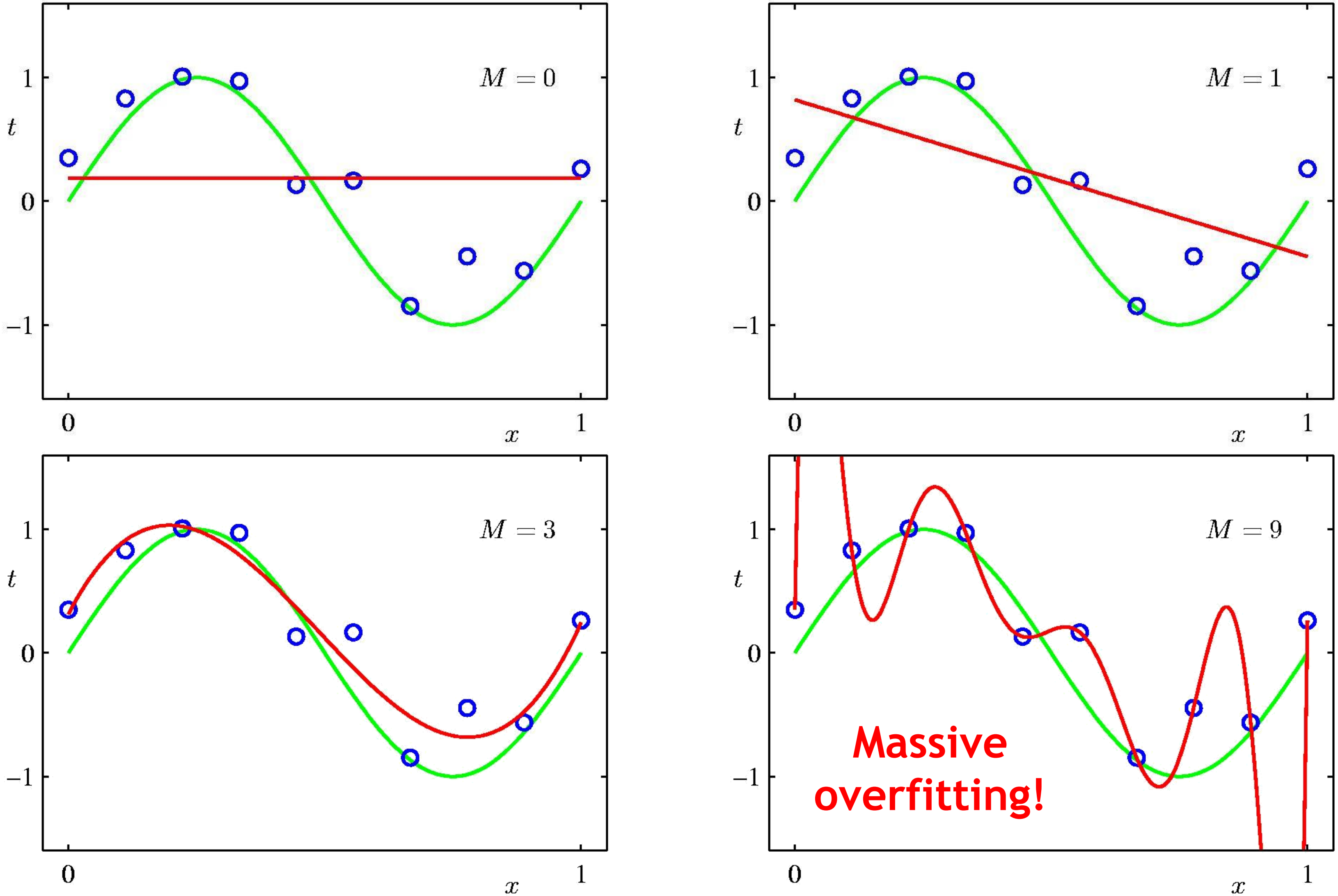
$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) = \sum_{i=0}^{M} w_i \phi_i(\mathbf{x})$$

assume $\phi_0(\mathbf{x}) = 1$

basis functions

  - E.g.: $\phi(\mathbf{x}) = (1, x, x^2, x^3)^T$
  - Fitting a cubic polynomial.

# Varying the Order of the Polynomial.




***Which one should we pick?***



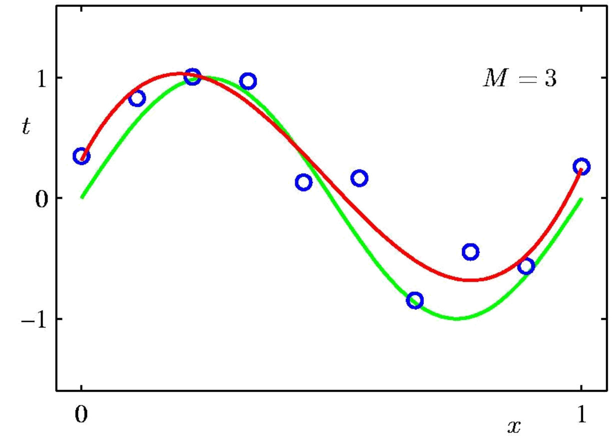
Image source: C.M. Bishop, 2006



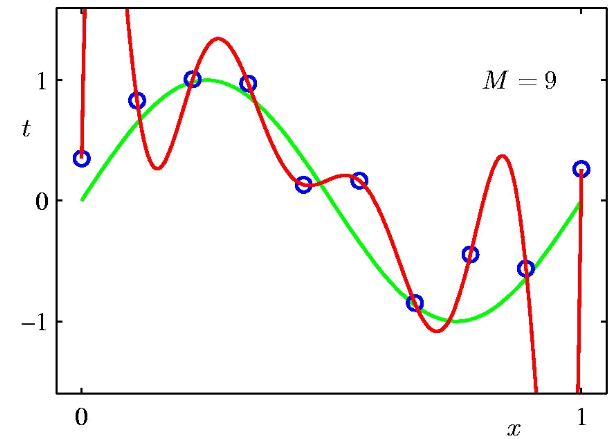
RWTH AACHEN UNIVERSITY

# Analysis of the Results

- Results for different values of $M$
  - Best representation of the original function $sin(2\pi x)$ with $M = 3$.




  - Perfect fit to the training data with $M = 9$, but poor representation of the original function.




- Why is that???
  - After all, $M = 9$ contains $M = 3$ as a special case!

# Overfitting

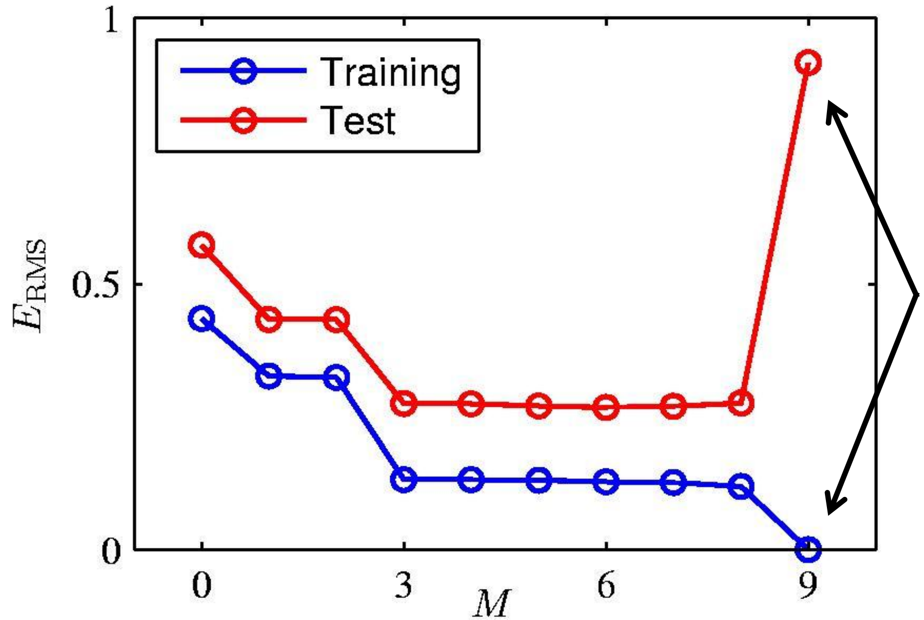
- Problem
  - Training data contains some noise

$$f(x) = sin(2\pi x) + \epsilon$$

  - Higher-order polynomial fitted perfectly to the noise.
  - We say it was overfitting to the training data.

- Goal is a good prediction of future data
  - Our target function should fit well to the training data, but also generalize.
  - Measure generalization performance on independent test set.

# Measuring Generalization

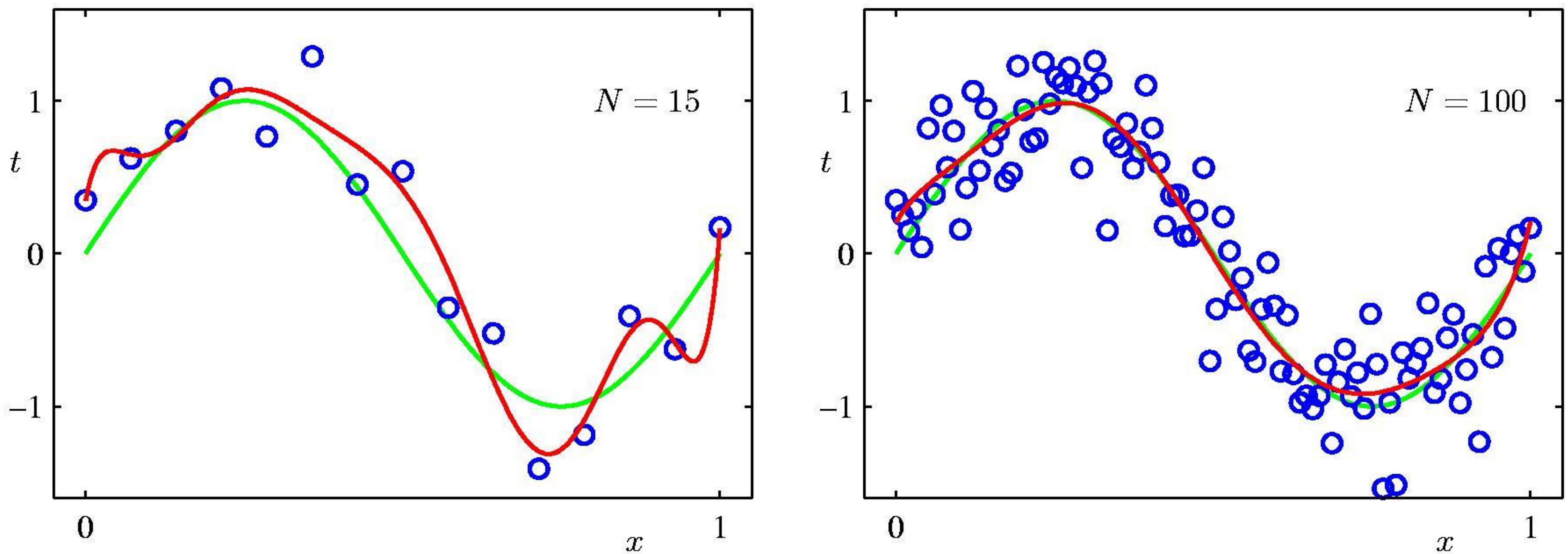



**Overfitting!**

- **E.g., Root Mean Square Error (RMS):** $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$
- **Motivation**
  - **Division by $N$ lets us compare different data set sizes.**
  - **Square root ensures $E_{RMS}$ is measured on the same scale (and in the same units) as the target variable $t$.**

# Analyzing Overfitting

- Example: Polynomial of degree 9



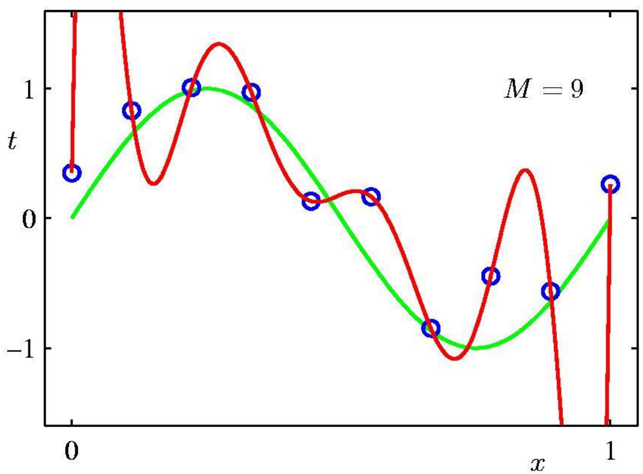

**Relatively little data**
**Overfitting typical**

**Enough data**
**Good estimate**

**⇒ Overfitting becomes less of a problem with more data.**

# What Is Happening Here?

- **The coefficients get very large:**
  - Fitting the data from before with various polynomials.
  - 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 |

# Regularization

- What can we do then?
  - How can we apply the approach to data sets of limited size?
  - We still want to use relatively complex and flexible models.
- Workaround: Regularization
  - Penalize large coefficient values

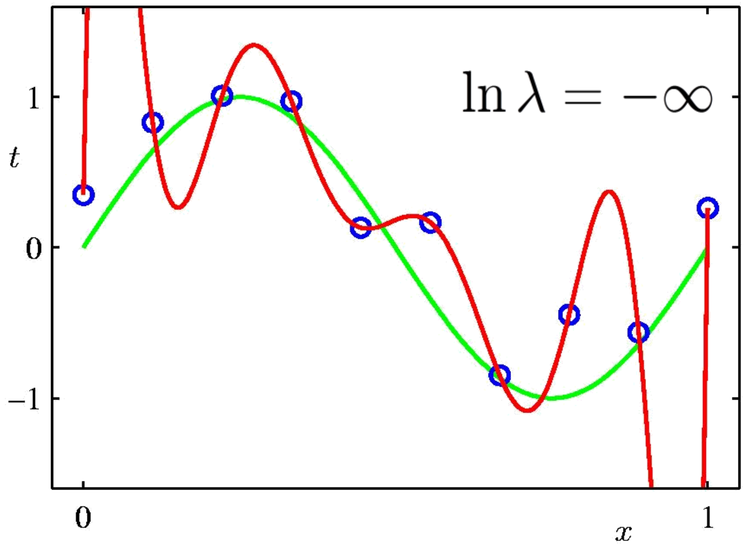
$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

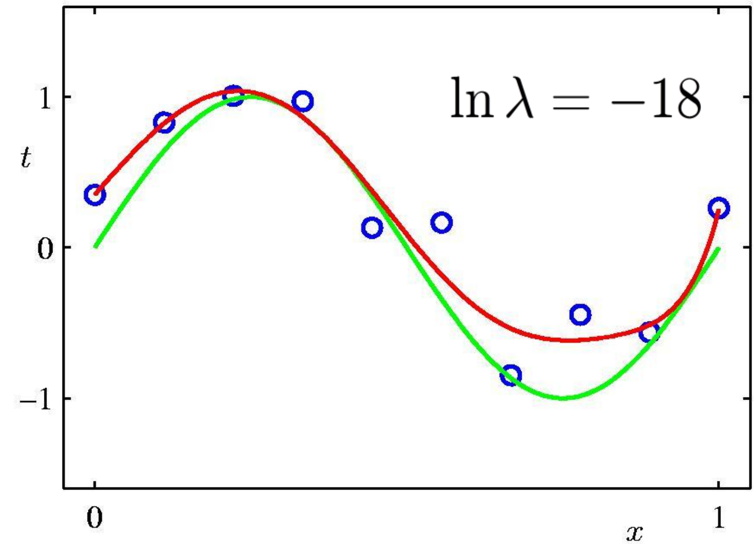
  - Here we've simply added a quadratic regularizer, which is simple to optimize

$$\|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w} = w_0^2 + w_1^2 + \ldots + w_M^2$$

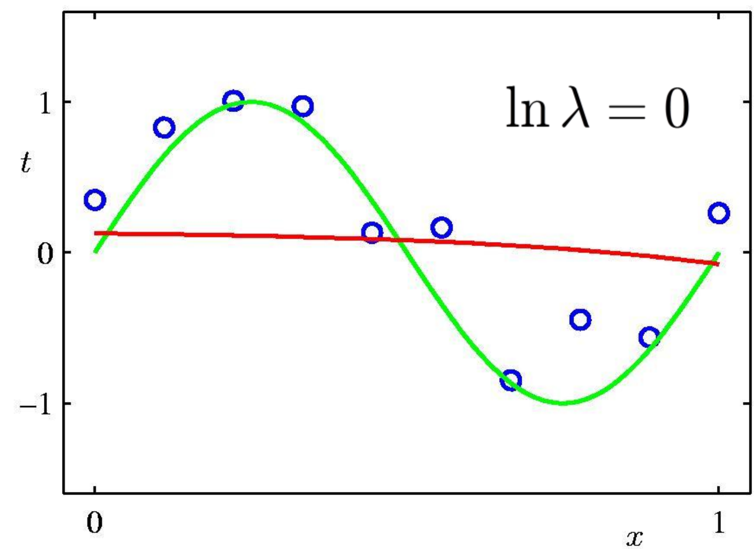
  - The resulting form of the problem is called Ridge Regression.
  - (*Note*: $w_0$ is often omitted from the regularizer.)

# Results with Regularization (M=9)



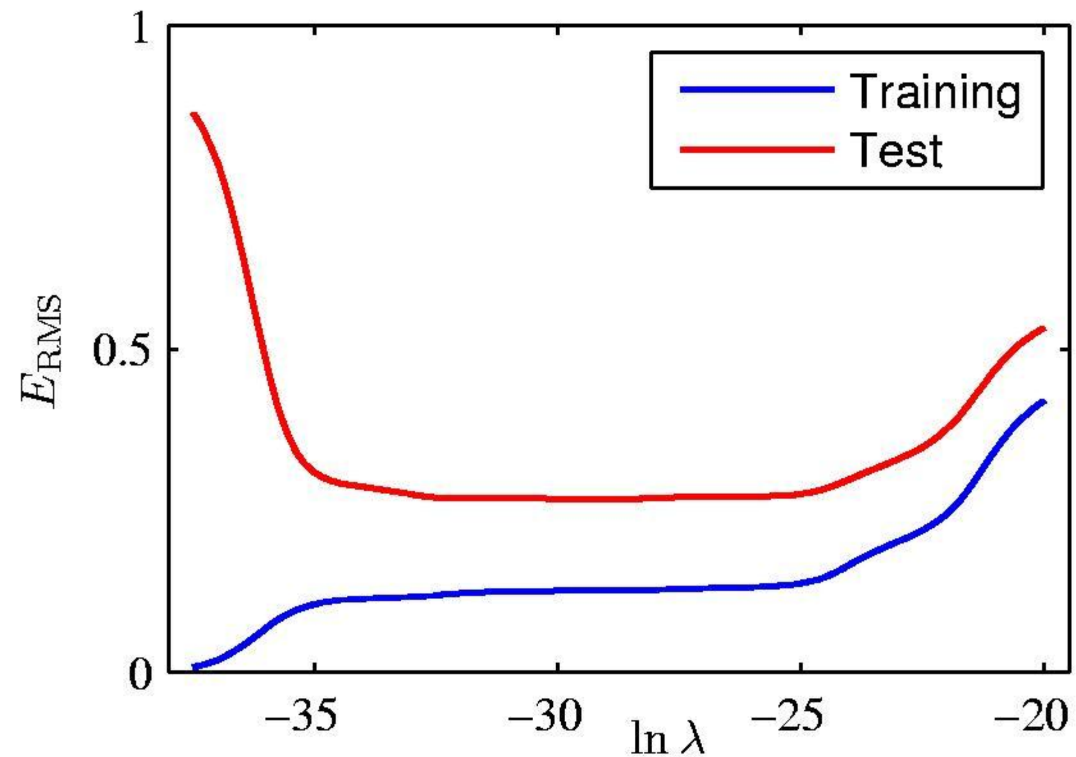




| | $\ln \lambda = -\infty$ | $\ln \lambda = -18$ | $\ln \lambda = 0$ |
|---|---|---|---|
| $w_0^\star$ | 0.35 | 0.35 | 0.13 |
| $w_1^\star$ | 232.37 | 4.74 | -0.05 |
| $w_2^\star$ | -5321.83 | -0.77 | -0.06 |
| $w_3^\star$ | 48568.31 | -31.97 | -0.05 |
| $w_4^\star$ | -231639.30 | -3.89 | -0.03 |
| $w_5^\star$ | 640042.26 | 55.28 | -0.02 |
| $w_6^\star$ | -1061800.52 | 41.32 | -0.01 |
| $w_7^\star$ | 1042400.18 | -45.95 | -0.00 |
| $w_8^\star$ | -557682.99 | -91.53 | 0.00 |
| $w_9^\star$ | 125201.43 | 72.68 | 0.01 |



RWTH AACHEN UNIVERSITY

# RMS Error for Regularized Case




- **Effect of regularization**
  - **The trade-off parameter $\lambda$ now controls the effective model complexity and thus the degree of overfitting.**

# Summary

- We've seen several important concepts
  - Linear regression
  - Overfitting
  - Role of the amount of data
  - Role of model complexity
  - Regularization

- How can we approach this more systematically?
  - Would like to work with complex models.
  - How can we prevent overfitting systematically?
  - How can we avoid the need for validation on separate test data?
  - What does it *mean* to do linear regression?
  - What does it *mean* to do regularization?

# Topics of This Lecture

- Regression: Motivation
  - Polynomial fitting
  - General Least-Squares Regression
  - Overfitting problem
  - Regularization
  - Ridge Regression
- Recap: Important Concepts from ML Lecture
  - Probability Theory
  - Bayes Decision Theory
  - Maximum Likelihood Estimation
  - Bayesian Estimation
- A Probabilistic View on Regression
  - Least-Squares Estimation as Maximum Likelihood

# Recap: The Rules of Probability

- Basic rules

**Sum Rule** $$p(X) = \sum_Y p(X, Y)$$

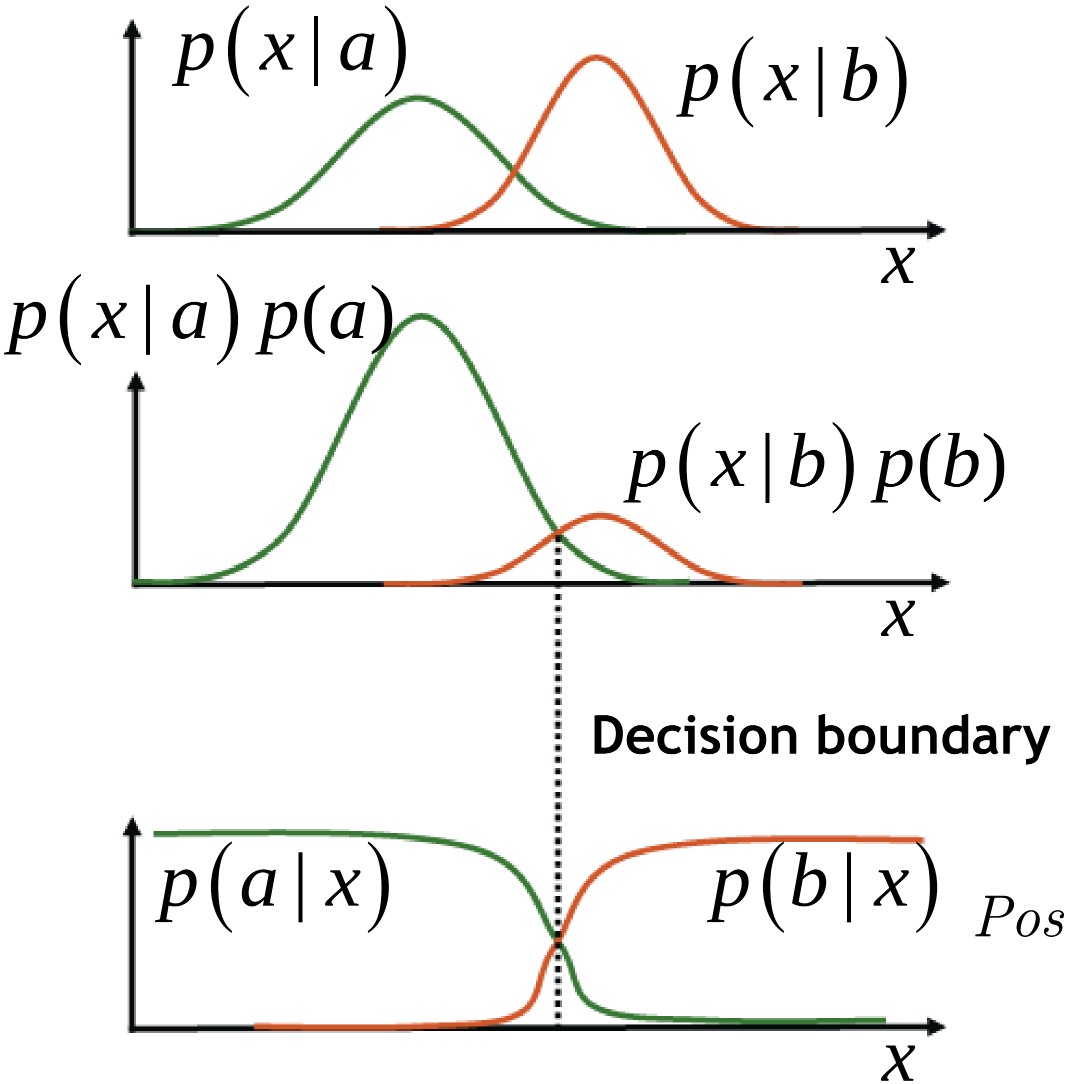
**Product Rule** $$p(X, Y) = p(Y|X)p(X)$$

- From those, we can derive

**Bayes' Theorem** $$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

**where** $$p(X) = \sum_Y p(X|Y)p(Y)$$

# Recap: Bayes Decision Theory




$Likelihood$

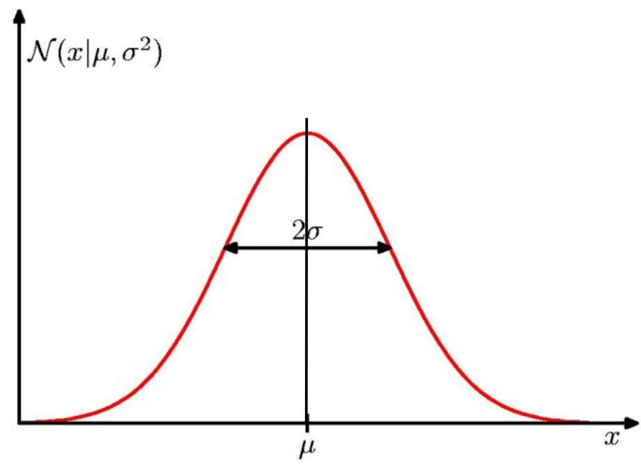
$Likelihood \times Prior$

$$Posterior = \frac{Likelihood \times Prior}{NormalizationFactor}$$

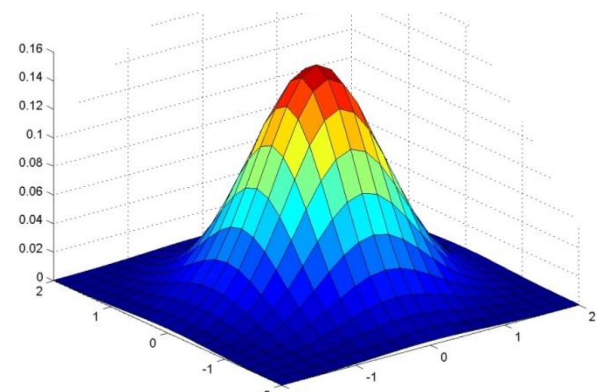
# Recap: Gaussian (or Normal) Distribution

- **One-dimensional case**
  - Mean $\mu$
  - Variance $\sigma^2$

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




- **Multi-dimensional case**
  - Mean $\boldsymbol{\mu}$
  - Covariance $\Sigma$




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


Image source: C.M. Bishop, 2006

# Side Note

- **Notation**
  - In many situations, it will be necessary to work with the inverse of the covariance matrix $\mathbf{\Sigma}$:

$$\mathbf{\Lambda} = \mathbf{\Sigma}^{-1}$$

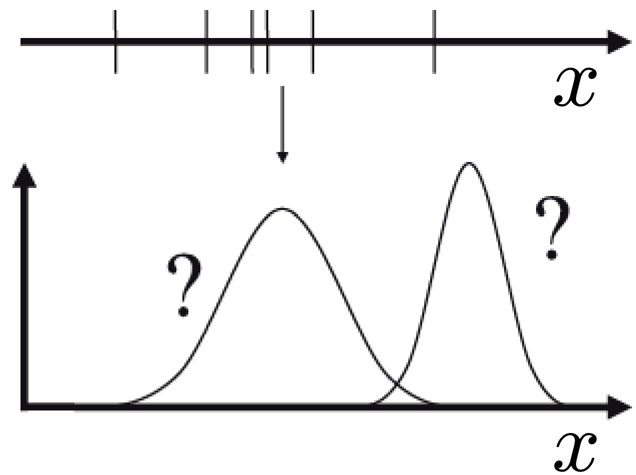
  - We call $\mathbf{\Lambda}$ the precision matrix.
  - We can therefore also write the Gaussian as

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

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

# Recap: Parametric Methods

- **Given**
  - Data $X = \{x_1, x_2, \ldots, x_N\}$
  - Parametric form of the distribution with parameters $\theta$
  - E.g. for Gaussian distrib.: $\theta = (\mu, \sigma)$




- **Learning**
  - Estimation of the parameters $\theta$

- **Likelihood of $\theta$**
  - Probability that the data $X$ have indeed been generated from a probability density with parameters $\theta$

$$L(\theta) = p(X|\theta)$$

# Recap: Maximum Likelihood Approach

- Computation of the likelihood
  - Single data point: $p(x_n|\theta)$
  - Assumption: all data points $X = \{x_1, \ldots, x_n\}$ are independent

$$L(\theta) = p(X|\theta) = \prod_{n=1}^{N} p(x_n|\theta)$$

  - Log-likelihood

$$E(\theta) = -\ln L(\theta) = -\sum_{n=1}^{N} \ln p(x_n|\theta)$$

- Estimation of the parameters $\theta$ (Learning)
  - Maximize the likelihood (=minimize the negative log-likelihood)

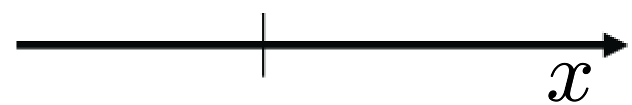
  $\Rightarrow$ Take the derivative and set it to zero.

$$\frac{\partial}{\partial\theta} E(\theta) = -\sum_{n=1}^{N} \frac{\frac{\partial}{\partial\theta} p(x_n|\theta)}{p(x_n|\theta)} \stackrel{!}{=} 0$$

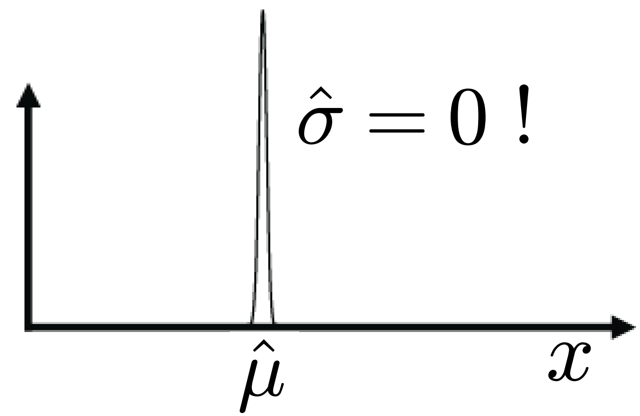
# Recap: Maximum Likelihood – Limitations

- **Maximum Likelihood has several significant limitations**
  - **It systematically underestimates the variance of the distribution!**
  - **E.g. consider the case**

$$N = 1, X = \{x_1\}$$



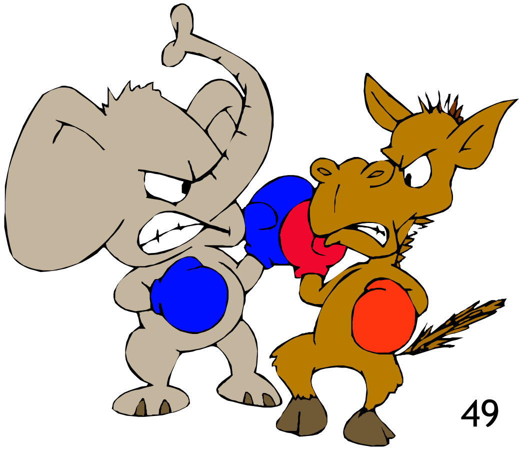

**⇒ Maximum-likelihood estimate:**




  - **We say ML *overfits to the observed data*.**
  - **We will still often use ML, but it is important to know about this effect.**

# Recap: Deeper Reason

- Maximum Likelihood is a Frequentist concept
  - In the Frequentist view, probabilities are the frequencies of random, repeatable events.
  - These frequencies are fixed, but can be estimated more precisely when more data is available.
- This is in contrast to the Bayesian interpretation
  - In the Bayesian view, probabilities quantify the uncertainty about certain states or events.
  - This uncertainty can be revised in the light of new evidence.
- Bayesians and Frequentists do not like each other too well…



# Recap: Bayesian Learning Approach

- **Bayesian view:**
  - Consider the parameter vector $\theta$ as a random variable.
  - When estimating the parameters, what we compute is

$$p(x|X) = \int p(x, \theta|X) d\theta$$

**Assumption: given $\theta$, this doesn't depend on X anymore**

$$p(x, \theta|X) = p(x|\theta, \cancel{X}) p(\theta|X)$$

$$p(x|X) = \int \underbrace{p(x|\theta)}p(\theta|X) d\theta$$

This is entirely determined by the parameter $\theta$ (i.e. by the parametric form of the pdf).

# Recap: Bayesian Learning Approach

- Discussion

**Likelihood** of the parametric form $\theta$ given the data set $X$.

**Estimate** for $x$ based on parametric form $\theta$

**Prior** for the parameters $\theta$

$$p(x|X) = \int \frac{p(x|\theta)L(\theta)p(\theta)}{\underbrace{\int L(\theta)p(\theta)d\theta}}d\theta$$

**Normalization**: integrate over all possible values of $\theta$

  - The more uncertain we are about $\theta$, the more we average over all possible parameter values.

# Topics of This Lecture

- Regression: Motivation
  - Polynomial fitting
  - General Least-Squares Regression
  - Overfitting problem
  - Regularization
  - Ridge Regression
- Recap: Important Concepts from ML Lecture
  - Probability Theory
  - Bayes Decision Theory
  - Maximum Likelihood Estimation
  - Bayesian Estimation
- A Probabilistic View on Regression
  - **Least-Squares Estimation as Maximum Likelihood**

# Next lecture…

# References and Further Reading

- More information, including a short review of Probability theory and a good introduction in Bayes Decision Theory can be found in Chapters 1.1, 1.2 and 1.5 of

Christopher M. Bishop
Pattern Recognition and Machine Learning
Springer, 2006