Advanced Machine Learning
Summer 2019

Part 1 – Introduction
03.04.2019

Prof. Dr. Bastian Leibe

RWTH Aachen University, Computer Vision Group
http://www.vision.rwth-aachen.de

Organization

- Lecturer
  - Prof. Bastian Leibe (leibe@vision.rwth-aachen.de)
- Teaching Assistants
  - Jonathan Luiten (luiten@vision.rwth-aachen.de)
  - Ömer Sali (sali@vision.rwth-aachen.de)
- Course webpage
  - http://www.vision.rwth-aachen.de/courses/
  - Slides will be made available on the webpage
  - There is also an electronic repository (moodle)
- Please subscribe to the lecture on RWTH Online!
  - Important to get email announcements and moodle 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 winter semester)
  - Introduction to ML
  - Classification
  - Graphical models
- This course:
  - 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?*
- This year: changed lecture content (compared to WS'16)
  - Large lecture block on Probabilistic Graphical Models
  - Updated with some exciting new topics (GANs, VAEs, Deep RL)

Organization

- Structure: 3V (lecture) + 1Ü (exercises)
  - 6 EECS credits
  - Part of the area "Applied Computer Science"
- Place & Time
  - Lecture/Exercises: Wed 10:30 – 12:00 room H06
  - Lecture/Exercises: Thu 10:30 – 12:00 room H04
- Exam
  - Oral or written exam, depending on number of participants

Course Webpage

Course Schedule

Date	Title	Content	Material	
Wed, 2019-04-03	Introduction	Introduction, Polynomial Fitting, Least-Squares Regression, Overfitting, Regularization, Ridge Regression		
Thu, 2019-04-04	Linear Regression I	Probabilistic View of Regression, Maximum Likelihood, MAP, Bayesian Curve Fitting		
Wed, 2019-04-10	Linear Regression II	Basis Functions, Sequential Learning, Multiple Outputs, Regularization, Lasso, Bias-Variance Decomposition		
Thu, 2019-04-11	Linear Regression III	Kernels, Kernel Ridge Regression		
Wed, 2019-04-17	Deep Reinforcement Learning I	Reinforcement Learning, TD Learning, Q-Learning, SARSA, Deep RL		
Thu, 2019-04-18	Deep Reinforcement Learning II	Deep RL, Deep Q-Learning, Deep Policy Gradients, Case studies		
Wed, 2019-04-24	Exercise 1	Regression, Least-Squares, Ridge, Kernel		

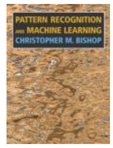
http://www.vision.rwth-aachen.de/courses/

Exercises and Supplementary Material

- Exercises
  - Typically 1 exercise sheet every 2 weeks.
  - Pen & paper and programming exercises
    - Matlab / numpy for early topics
    - Theano for Deep Learning topics
  - 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

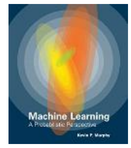
- Many lecture topics will be covered in Bishop's book.
- Some additional topics can be found in Murphy's book



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

(available in the library's "Handapparat")

Kevin P. Murphy
Machine Learning – A Probabilistic Perspective
MIT Press, 2012



- 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 see us.
  - My regular office hours will be announced.
  - 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

Slide credit: Bernt Schiele

What Is Machine Learning Useful For?

Automatic Speech Recognition

Slide adapted from Zoubin Gharamani

What Is Machine Learning Useful For?

Computer Vision
(Object Recognition, Segmentation, Scene Understanding)

Slide adapted from Zoubin Gharamani

What Is Machine Learning Useful For?

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

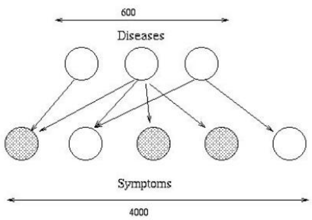
Slide adapted from Zoubin Gharamani

What Is Machine Learning Useful For?

Financial Prediction
(Time series analysis, ...)

Slide adapted from Zoubin Gharamani

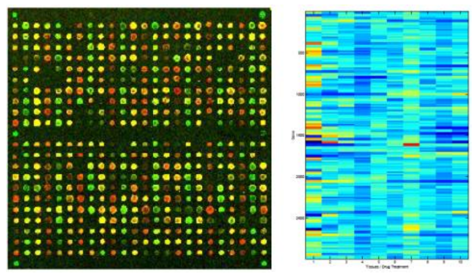
What Is Machine Learning Useful For?



Medical Diagnosis
(Inference from partial observations)

Slide adapted from Zoubin Gharamani
Image from Kevin Murphy

What Is Machine Learning Useful For?



Bioinformatics
(Modelling gene microarray data,...)

Slide adapted from Zoubin Gharamani

What Is Machine Learning Useful For?

Robotics & Autonomous Driving

Slide adapted from Zoubin Gharamani

Machine Learning: Core Questions

- *Learning to perform a task from experience*
- Task
  - Can often be expressed through a mathematical function

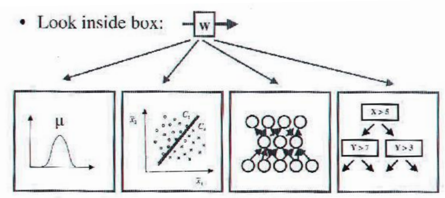
$$y = f(x; w)$$

  - 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

Slide credit: Bernt Schiele

Machine Learning: Core Questions

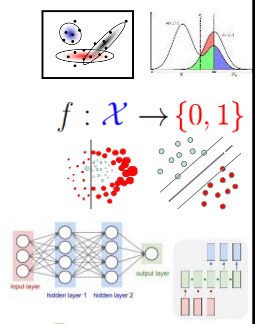
- $y = f(x; w)$
  - w: characterizes the family of functions
  - w: indexes the space of hypotheses
  - w: vector, connection matrix, graph, …



Slide credit: Bernt Schiele

A Look Back: Lecture *Machine Learning*

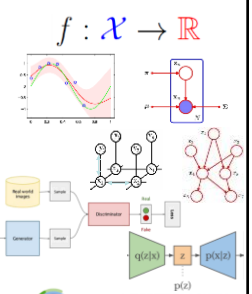
- Fundamentals
  - Bayes Decision Theory
  - Probability Density Estimation
  - Mixture Models and EM
- Classification Approaches
  - Linear Discriminant Functions
  - Support Vector Machines
  - Ensemble Methods & Boosting
- Deep Learning
  - Foundations
  - Convolutional Neural Networks
  - Recurrent Neural Networks



This Lecture: *Advanced Machine Learning*

Extending lecture *Machine Learning* from last semester…

- Regression Techniques
  - Regularization (Ridge, Lasso)
  - Bayesian Regression
- Probabilistic Graphical Models
  - Bayesian Networks
  - Markov Random Fields
  - Inference (exact & approximate)
- Deep Generative Models
  - Generative Adversarial Networks
  - Variational Autoencoders
- Deep Reinforcement Learning



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 accessible to everyone.
  - We'll review the main concepts before applying them.
  - I'll point out chapters to review from the ML lecture whenever knowledge from there is needed/helpful.
- But…
  - This *is* an advanced topics class.
  - There *will* be math involved.
  - We will take a deeper look into the theory than in the ML lecture.

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

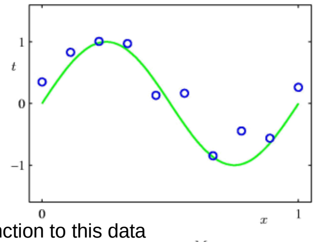
- Learning to predict a continuous function value
  - Given: training set $\mathbf{X} = \{x_1, \ldots, x_N\}$ with target values $\mathbf{T} = \{t_1, \ldots, t_N\}$.
  - $\Rightarrow$ Learn a continuous function $y(x)$ to predict the function value for a new input x.
- Steps towards a solution
  - Choose 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…

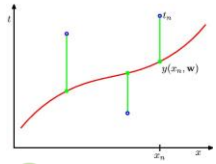


Image source: C.M. Bishop, 2006

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.

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

  - 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: $X = \{\mathbf{x}_1 \in \mathbb{R}^d, \ldots, \mathbf{x}_n\}$
  - Associated function values: $T = \{t_1 \in \mathbb{R}, \ldots, t_n\}$
- Start with linear regressor:
  - Try to enforce $\mathbf{x}_i^T \mathbf{w} + w_0 = t_i, \quad \forall i = 1, \ldots, n$
  - 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.

Slide credit: Bernt Schiele

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

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

  - 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).

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

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

basis functions

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

Slide credit: Bernt Schiele

Varying the Order of the Polynomial.

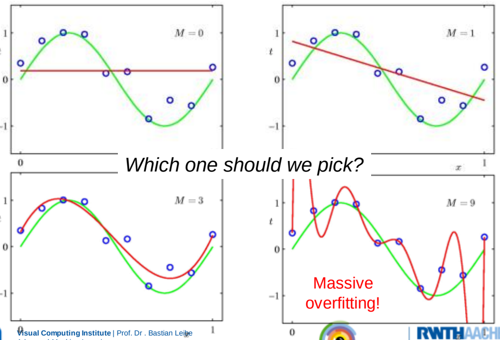
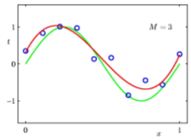


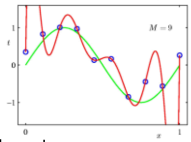
Image source: C.M. Bishop, 2006

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!

Image source: C.M. Bishop, 2006

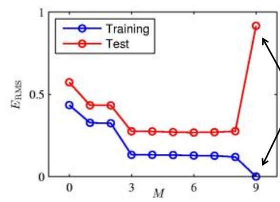
Overfitting

- 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

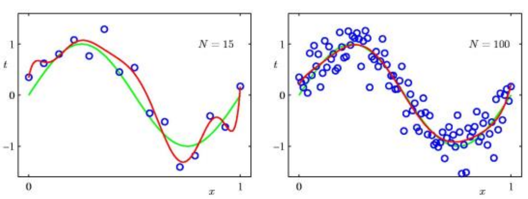


- E.g., Root Mean Square Error (RMS): $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$
- Motivation
  - Division by N lets us compare different data set sizes.
  - Square root ensures E_{RMS} is measured on the same scale (and in the same units) as the target variable t.

Image source: C.M. Bishop, 2006

Analyzing Overfitting

- Example: Polynomial of degree 9



Relatively little data
Overfitting typical

Enough data
Good estimate

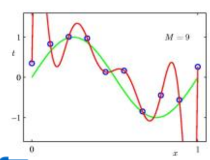
$\Rightarrow$ Overfitting becomes less of a problem with more data.

Slide adapted from Bernt Schiele
Image source: C.M. Bishop, 2006

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



Slide credit: Bernt Schiele
Image source: C.M. Bishop, 2006

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

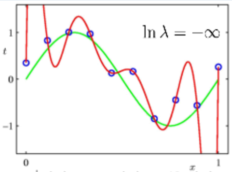
$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

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

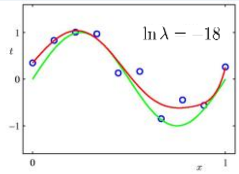
$$\|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w} = \cancel{w_0^2} + w_1^2 + \ldots + w_M^2$$

  - 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



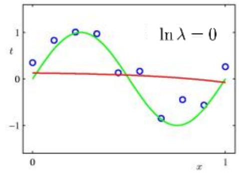
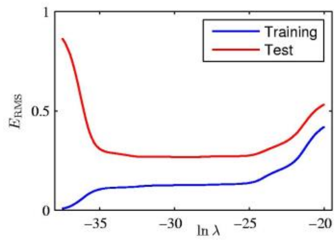


Image source: C.M. Bishop, 2006

RMS Error for Regularized Case



- Effect of regularization
  - The trade-off parameter λ now controls the effective model complexity and thus the degree of overfitting.

Image source: C.M. Bishop, 2006

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
  - New: Bayesian Estimation
- A Probabilistic View on Regression
  - Least-Squares Estimation as Maximum Likelihood

Recap: The Rules of Probability

- Basic rules

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

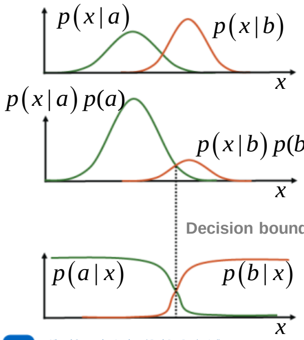
Product Rule $\quad p(X,Y) = p(Y|X)p(X)$

- From those, we can derive

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

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

Recap: Bayes Decision Theory



Likelihood

Likelihood $\times$ *Prior*

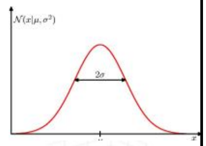
$$Posterior = \frac{Likelihood \times Prior}{NormalizationFactor}$$

Slide credit: Bernt Schiele

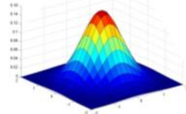
Recap: Gaussian (or Normal) Distribution

- One-dimensional case
  - Mean μ
  - Variance σ^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 μ
  - Covariance Σ



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

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 Σ :

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

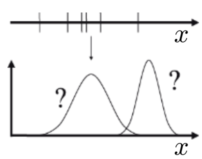
  - We call $\boldsymbol{\Lambda}$ the precision matrix.
  - We can therefore also write the Gaussian as

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

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

Recap: Parametric Methods

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



- Learning
  - Estimation of the parameters θ
- Likelihood of θ
  - Probability that the data X have indeed been generated from a probability density with parameters θ

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

Slide adapted from Bernt Schiele

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 θ (Learning)
  - Maximize the likelihood (=minimize the negative log-likelihood)
  - $\Rightarrow$ Take the derivative and set it to zero.

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

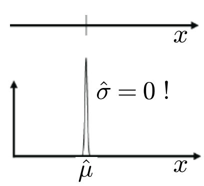
Slide credit: Bernt Schiele

Recap: Maximum Likelihood Approach

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

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

$\Rightarrow$ 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.

Slide adapted from Bernt Schiele

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…

Bayesian vs. Frequentist View

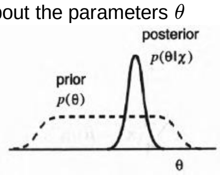
- To see the difference…
  - Suppose we want to estimate the uncertainty whether the Arctic ice cap will have disappeared by the end of the century.
  - This question makes no sense in a Frequentist view, since the event cannot be repeated numerous times.
  - In the Bayesian view, we generally have a prior, e.g. from calculations how fast the polar ice is melting.
  - If we now get fresh evidence, e.g. from a new satellite, we may revise our opinion and update the uncertainty from the prior.

$$\mathit{Posterior} \propto \mathit{Likelihood} \times \mathit{Prior}$$

  - This generally allows to get better uncertainty estimates for many situations.
- Main Frequentist criticism
  - The prior has to come from somewhere and if it is wrong, the result will be worse.

Bayesian Approach to Parameter Learning

- Conceptual shift
  - Maximum Likelihood views the true parameter vector θ to be unknown, but fixed.
  - In Bayesian learning, we consider θ to be a random variable.
- This allows us to use knowledge about the parameters θ
  - i.e. to use a prior for θ
  - Training data then converts this prior distribution on θ into a posterior probability density.



  - The prior thus encodes knowledge we have about the type of distribution we expect to see for θ.

Bayesian Learning Approach

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

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

Assumption: given θ, 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 θ (i.e. by the parametric form of the pdf).

Bayesian Learning Approach

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

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{p(X)} = \frac{p(\theta)}{p(X)} L(\theta)$$

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

- Inserting this above, we obtain

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

Bayesian Learning Approach

- Discussion

Likelihood of the parametric form θ given the data set X.

Estimate for x based on parametric form θ

Prior for the parameters θ

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

Normalization: integrate over all possible values of θ

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

Bayesian Density Estimation

- Discussion

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

  – The probability $p(\theta|X)$ makes the dependency of the estimate on the data explicit.

  – If $p(\theta|X)$ is very small everywhere, but is large for one $\hat{\theta}$, then

$$p(x|X) \approx p(x|\hat{\theta})$$

  $\Rightarrow$ The more uncertain we are about θ, the more we average over all parameter values.

- Problem
  – In the general case, the integration over θ is not possible (or only possible stochastically).

Slide credit: Bernt Schiele

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