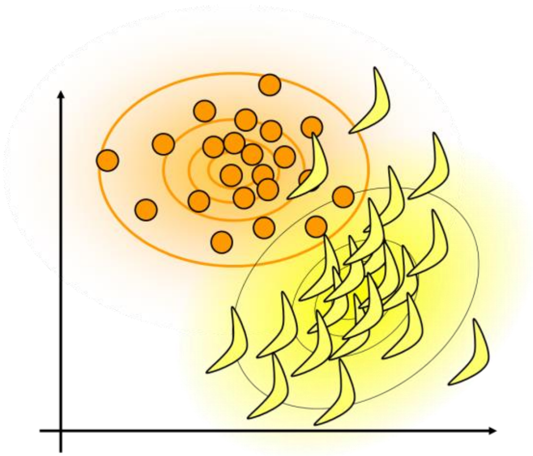
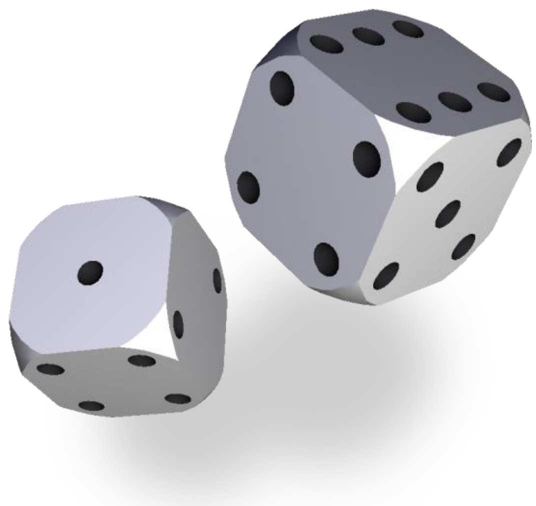
Modelling 2

STATISTICAL DATA MODELLING





Chapter 5

Bayesian Data Analysis & Classical ML

Michael Wand · Institut für Informatik, JGU Mainz · michael.wand@uni-mainz.de

Video #05

Statistics & Machine Learning

Classical Machine Learning

- **Modeling 1 Recap:** Least-Squares, PCA
- **Old-School:** Classical Classifiers

Bayesian Data Analysis

- **Example 1:** MAP Image Reconstruction
- **Example 2:** Bayesian Regression

Some Classical ML Methods

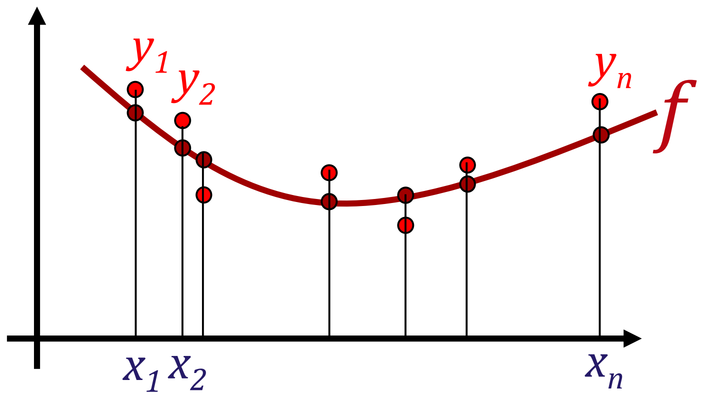
Recap from Modeling 1

- **Regression:** Least-squares fitting
- **A generative model:** Gaussian fitting
- **Dimensionality reduction:** PCA
  (Principle Component Analysis)

Regression with Linear Models

via Least-Squares Function Fitting

Situation



Situation

- Sample points taken at x_i from original f.
- Unknown Gaussian i.i.d. noise added to each y_i.
- Reconstruct $\tilde{f}$.

Summary (→ Mod-1)

Statistical model: least-squares criterion

$$\arg\min_{\tilde{f}} \sum_{i=1}^{n} \left(\tilde{f}(x_i) - y_i\right)^2$$

Linear ansatz: quadratic objective

$$\tilde{f}_{\lambda_1,\ldots,\lambda_k}(x) = \sum_{j=1}^{k} \lambda_j b_j(x) \quad \rightarrow \quad \arg\min_{\lambda_1,\ldots,\lambda_k} \sum_{i=1}^{n} \left(\left(\sum_{j=1}^{k} \lambda_j b_j(x_i) \right) - y_i \right)^2$$

Critical point: linear system

$$\begin{pmatrix} \langle \mathbf{b}_1, \mathbf{b}_1 \rangle & \cdots & \langle \mathbf{b}_1, \mathbf{b}_k \rangle \\ \vdots & \ddots & \vdots \\ \langle \mathbf{b}_k, \mathbf{b}_1 \rangle & \cdots & \langle \mathbf{b}_k, \mathbf{b}_k \rangle \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix} = \begin{pmatrix} \langle \mathbf{y}, \mathbf{b}_1 \rangle \\ \vdots \\ \langle \mathbf{y}, \mathbf{b}_k \rangle \end{pmatrix} \quad \text{with} \quad \begin{cases} \langle \mathbf{b}_i, \mathbf{b}_j \rangle := \sum_{t=1}^{n} b_i(x_t) \cdot b_j(x_t) \\ \langle \mathbf{y}, \mathbf{b}_i \rangle := \sum_{t=1}^{n} y_t \cdot b_i(x_t) \end{cases} \tag{8}$$

Maximum Likelihood Estimation

$$
\begin{aligned}
\arg\max_{\tilde{f}} \prod_{i=1}^{n} N_{0,\sigma}\left(\tilde{f}(x_i) - y_i\right) &= \arg\max_{\tilde{f}} \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(\tilde{f}(x_i) - y_i\right)^2}{2\sigma^2}\right) \\
&= \arg\max_{\tilde{f}} \ln\left[\prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(\tilde{f}(x_i) - y_i\right)^2}{2\sigma^2}\right)\right] \\
&= \arg\max_{\tilde{f}} \sum_{i=1}^{n} \left(\left(\ln\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{\left(\tilde{f}(x_i) - y_i\right)^2}{2\sigma^2}\right) \\
&= \arg\min_{\tilde{f}} \sum_{i=1}^{n} \left(+\frac{\left(\tilde{f}(x_i) - y_i\right)^2}{2\sigma^2}\right) \\
&= \arg\min_{\tilde{f}} \sum_{i=1}^{n} \left(\tilde{f}(x_i) - y_i\right)^2
\end{aligned} \tag{9}
$$

Estimating Gaussian

(Maximum Likelihood)

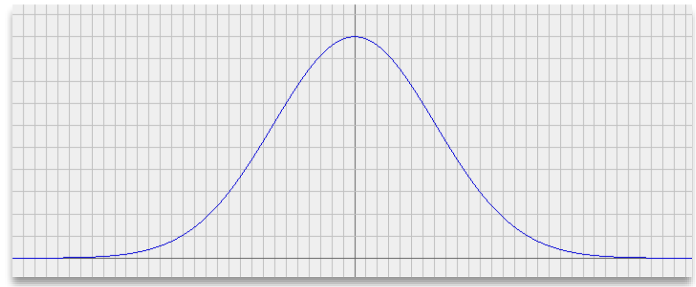
Gaussians

Gaussian Normal Distribution

- Two parameters: μ, σ
- Density:

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

- Mean: μ
- Variance: σ^2



Gaussian normal distribution

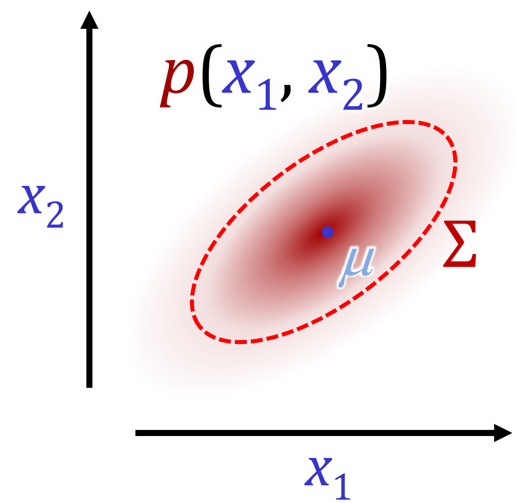
Multi-Variate Gaussians

Gaussian Normal Distribution in d Dimensions

- Two parameters: $\boldsymbol{\mu}$ (d-dim-vector), $\boldsymbol{\Sigma}$ ($d \times d$ matrix)
- Density:

$$\mathcal{N}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{x}) := \left(\frac{1}{(2\pi)^{-\frac{d}{2}} \det(\boldsymbol{\Sigma})^{-\frac{1}{2}}} \right) e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}$$

- Mean: $\boldsymbol{\mu}$
- Covariance Matrix: $\boldsymbol{\Sigma}$



ML-Estimation from Data

Task

- Data (i.i.d.) $\mathbf{d}_1, \dots, \mathbf{d}_n$ from Gaussian distribution
- Estimate parameters

Maximum Likelihood Estimation

$$\boldsymbol{\mu}_{ml} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{d}_i$$

mean

$$\boldsymbol{\Sigma}_{ml} = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{d}_i - \boldsymbol{\mu})(\mathbf{d}_i - \boldsymbol{\mu})^{\mathrm{T}}$$

covariance

PCA

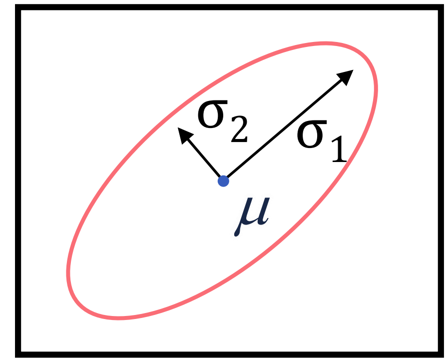
Least-Squares-Optimal Dimensionality Reduction

The Shape of Gaussians

Probability Density

$$\mathcal{N}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{x}) := \left(\frac{1}{(2\pi)^{-\frac{d}{2}}\det(\boldsymbol{\Sigma})^{-\frac{1}{2}}}\right) e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

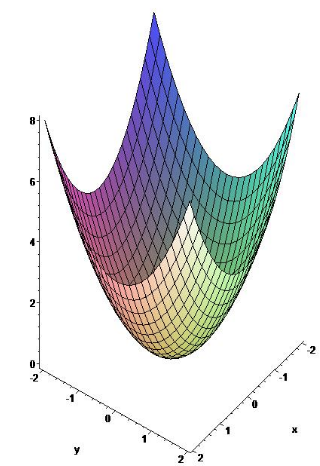
normalization



Neg-Log Density:

- $\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) + const$

normalization



Geometry

- Iso-probability profiles are ellipsoids
- Eigenvectors of Σ are main axes

General Case

Principal Component Analysis (PCA)

- $(\lambda_1, \mathbf{v}_1), \dots, (\lambda_n, \mathbf{v}_n)$: sorted eigenvalue/vector pairs of $\mathbf{\Sigma}$
  - λ_1 is the largest
  - $\|\mathbf{v}_i\| = 1$
- Select subspace spanned by

$$\mathbf{x}_0 + \operatorname{span}\{\mathbf{v}_1, \dots, \mathbf{v}_d\}, \qquad 0 \leq d \leq n$$

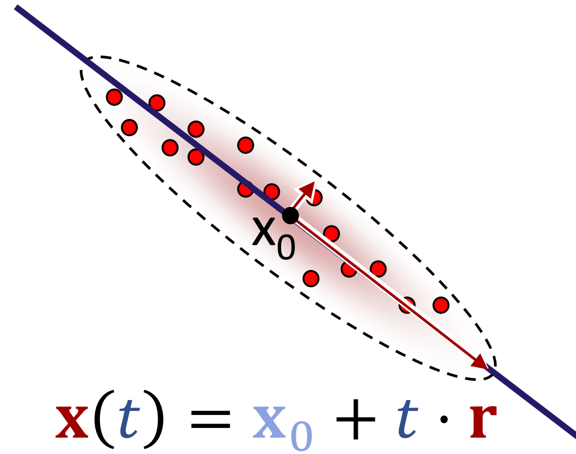
- Subspace-projection is optimal:
  - Yields optimal d-dim approximation among all possible affine subspaces (Wrt. squared distances)

} Linear Dimensionality Reduction

Example Application

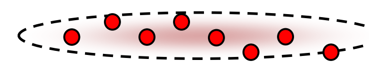
Fitting a line to a point cloud

- Sample mean and direction of maximum eigenvalue

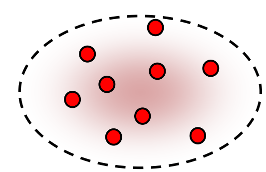


Plane Fitting in $\mathbb{R}^3$:

- Smallest eigenvalue: *normal* direction
- Aspect ratio λ_3/λ_2 is a measure of "flatness" (quality of fit)

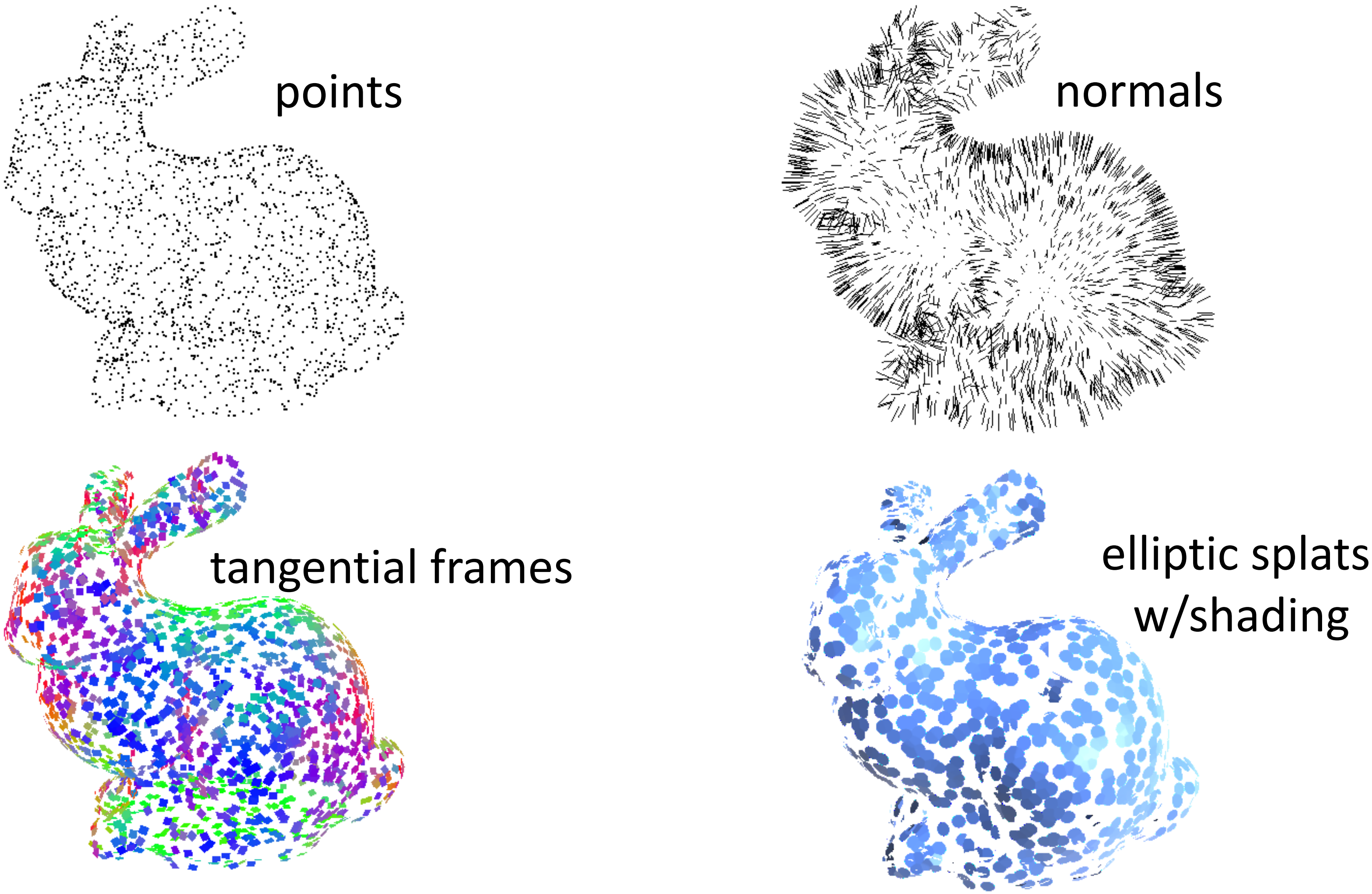


$\frac{\lambda_d}{\lambda_{d-1}}$ small

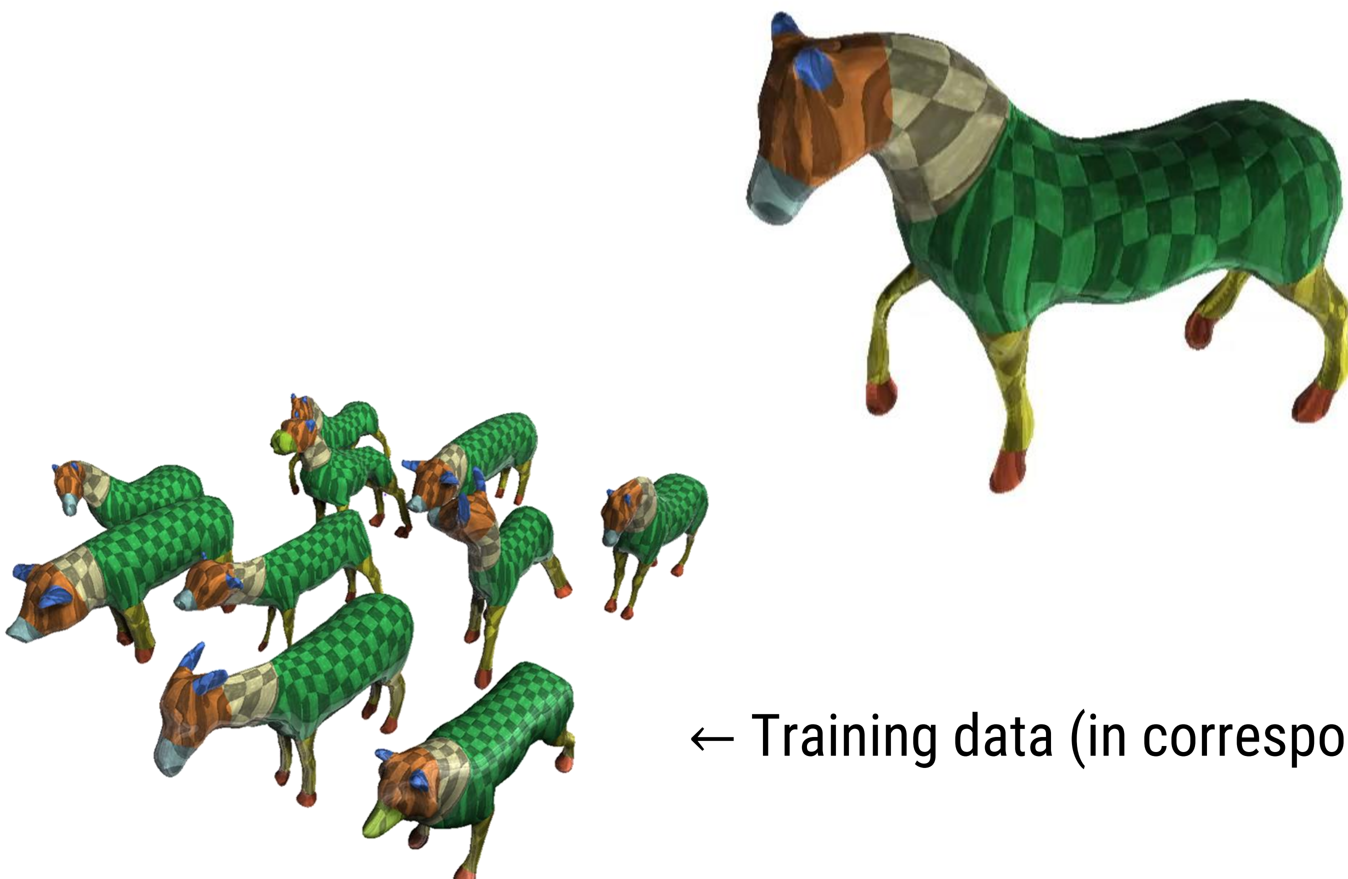


$\frac{\lambda_d}{\lambda_{d-1}}$ larger

Example Application



PCA-Model: 4-Legged Animals



← Training data (in correspondence)

Video #05a

Summary

Summary

Fun with Gaussians!

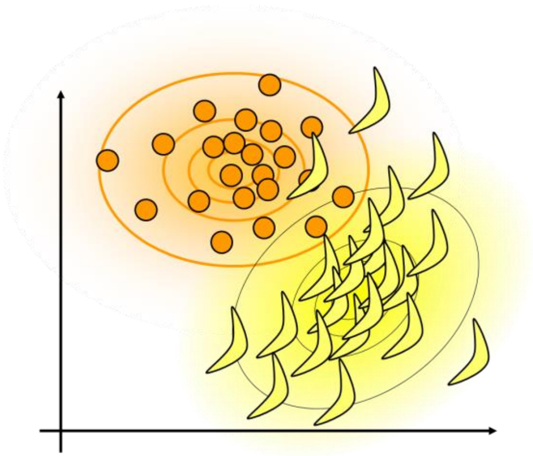
- Least-squares fitting: Gaussian
- PCA: Gaussian
- Effort level: Solving linear systems

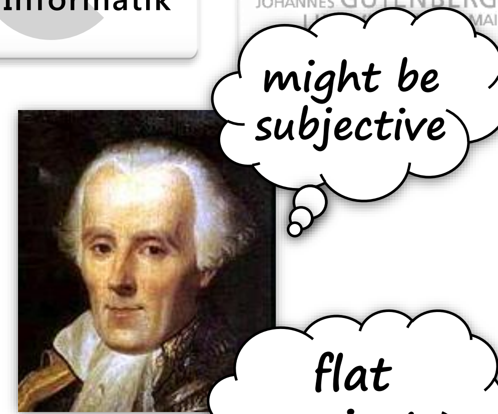
Ask-Me-Anything

- We can answer complicated / fancy questions without computational pain
- Unfortunately, not everything on earth is Gaussian

Modelling 2

STATISTICAL DATA MODELLING





Chapter 5

Bayesian Data Analysis & Classical ML

Michael Wand · Institut für Informatik, JGU Mainz · michael.wand@uni-mainz.de

Video #05

Statistics & Machine Learning

Classical Machine Learning

- **Modeling 1 Recap:** LS, PCA
- **Old-School:** Classical Classifiers

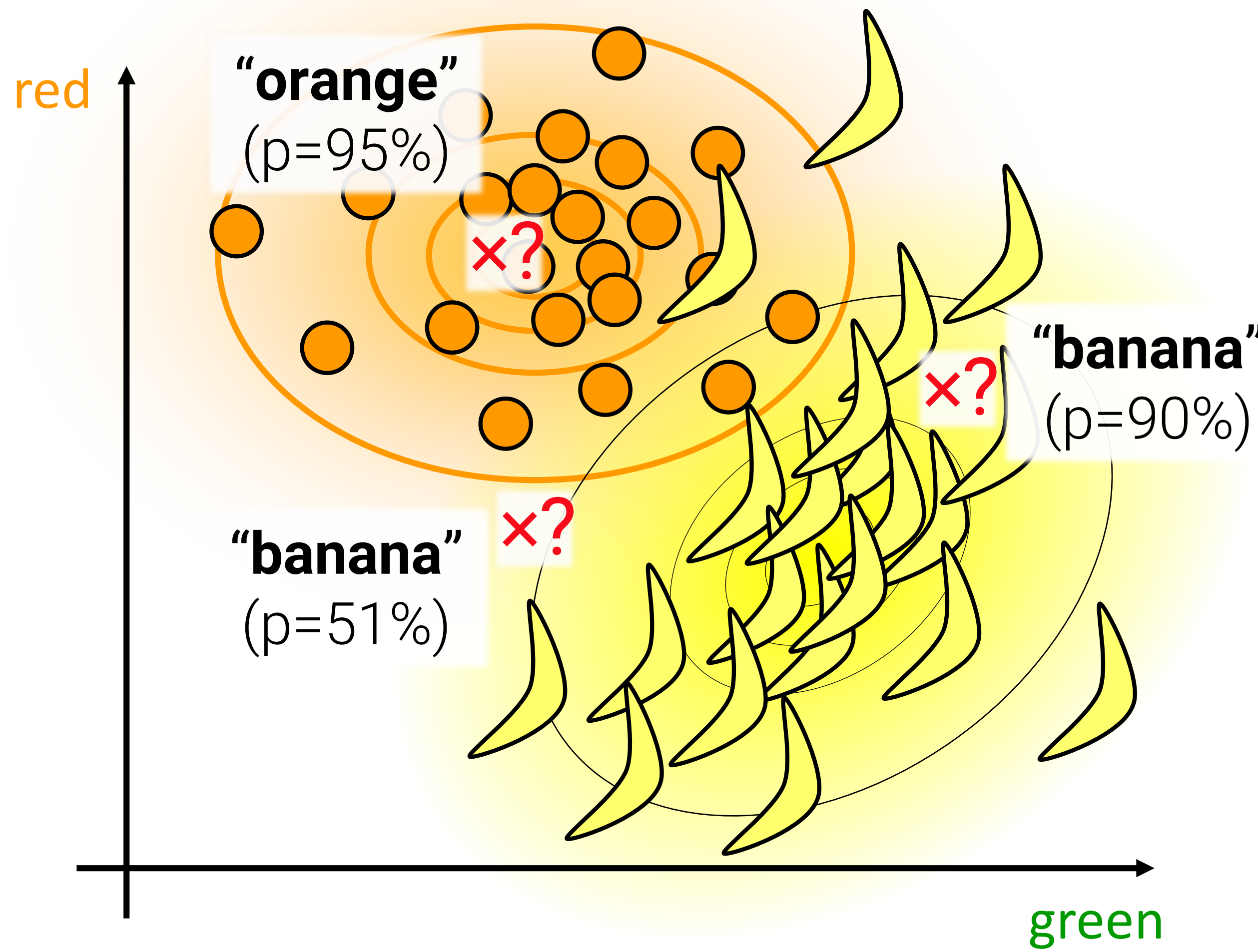
Bayesian Data Analysis

- **Example 1:** MAP Image Reconstruction
- **Example 2:** Bayesian Regression

Logistic Regression

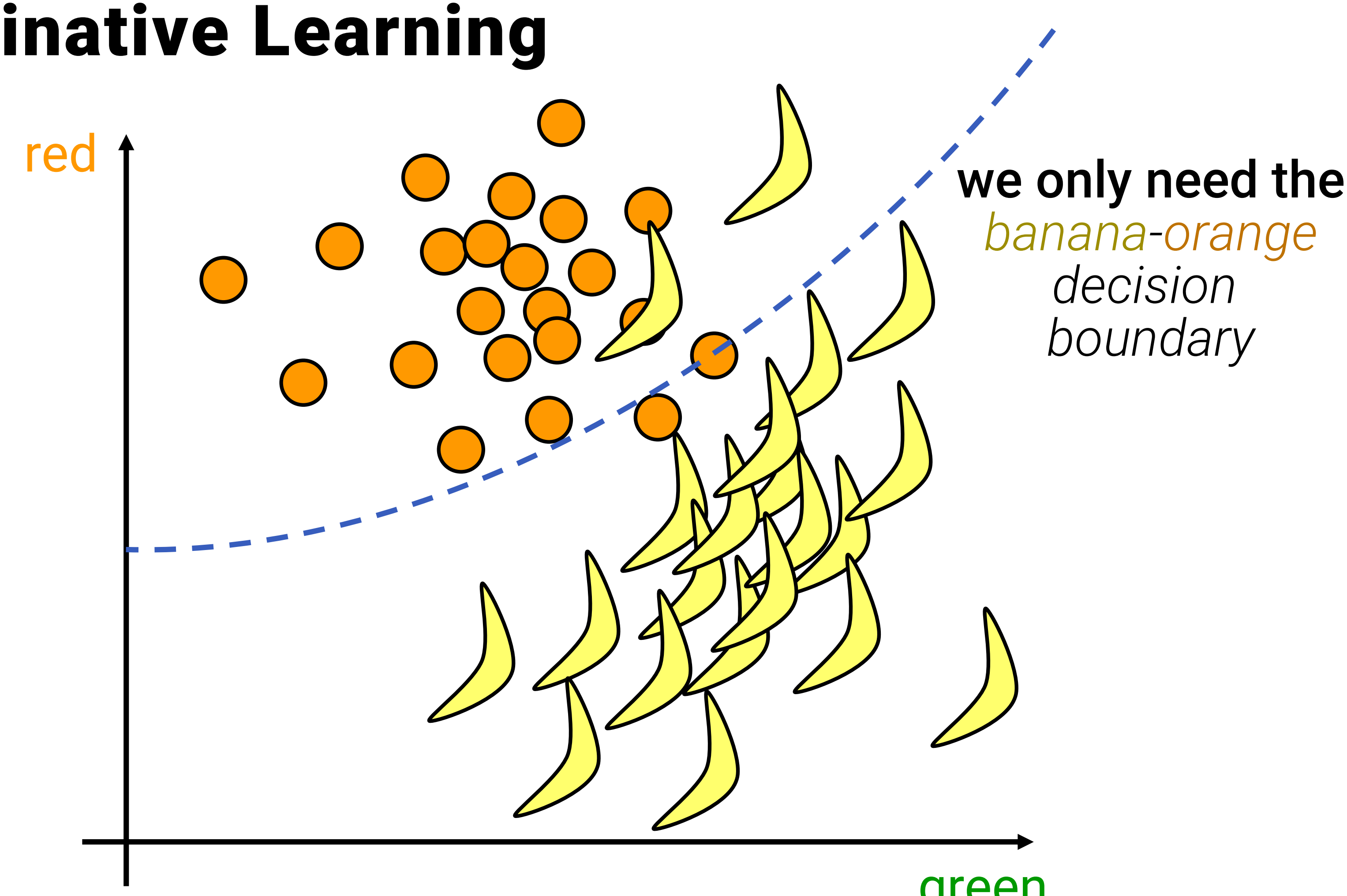
Example from Video #04

Gaussian Fitting: *Yes, we can* now do this.



Example from Video #04

Discriminative Learning



Let's Build a Classifier

Simple discriminative model

- Two classes, probabilities p, $(1 - p)$
- Need (only) odds-ratio

$$\frac{p}{1-p}$$

Logistic Regression

- **Model:** linear log-likelihood for p

$$p(\mathbf{x}) = e^{\boldsymbol{\theta}^T \mathbf{x}} = e^{\theta_1 x_1 + \cdots + \theta_d x_d}$$

  for input / feature vector $\mathbf{x} \in \mathbb{R}^d$
- Always positive

Let's Build a Classifier

We get

- Odds-ratio

$$\left.\begin{aligned}\frac{p}{1-p} &= \frac{e^{\theta^T \mathbf{x}}}{1-e^{\theta^T \mathbf{x}}}\\ &= \frac{1}{1-e^{-\theta^T \mathbf{x}}}\\ &= \sigma(\theta^T \mathbf{x})\end{aligned}\right\} =: h_{\theta}(\mathbf{x})$$

with

$$\sigma(z) := \frac{1}{1-e^{-(z)}}$$

(„Sigmoid function")

Training

Given

- Training examples $\{(\mathbf{x}_i, y_i)\}_{i=1\ldots n}$
  - "Feature Vectors" $\mathbf{x}_i \in \mathbb{R}^d$
  - "Labels" $y_i \in \{0,1\}$
    - Banana or not banana
    - (not banana = orange)
- Task
  - Find "good" $\boldsymbol{\theta} \in \mathbb{R}^d$
- Approach: Maximum Likelihood

MLE Logistic Regression

Maximum Likelihood Estimation

- We want

$$h_{\theta}(\mathbf{x}) = 1 \text{ for } y_i = 1 \quad \text{and} \quad h_{\theta}(\mathbf{x}) = 0 \text{ for } y_i = 0$$

- Likelihood for class $y = 1$

$$p(y|x, \theta) = h_{\theta}(\mathbf{x})$$

- Maximum likelihood

$$\hat{\theta} = \arg\max_{\theta \in \mathbb{R}^{d+1}} \prod_{i=1}^{n} \underbrace{h_{\theta}(\mathbf{x})}_{\substack{\text{maximize} \\ \text{for } y_i = 1}}{}^{y_i} \underbrace{\left(1 - h_{\theta}(\mathbf{x})\right)}_{\substack{\text{maximize} \\ \text{for } y_i = 0}}{}^{(1-y_i)}$$

MLE Logistic Regression

Maximum Likelihood Estimation

- MLE objective

$$\prod_{i=1}^{n} \underbrace{\left(\frac{p(x_i)}{1-p(x_i)}\right)^{y_i}}_{\text{maximize for } y_i=1} \underbrace{\left(1-\frac{p(x_i)}{1-p(x_i)}\right)^{(1-y_i)}}_{\text{maximize for } y_i=0} \rightarrow \max$$

- Using $\frac{1}{1-e^{-t}} + \frac{1}{1-e^{t}} = 1$, we get

$$\prod_{i=1}^{n} \left(\frac{1}{1-e^{-\theta^T \mathbf{x}}}\right)^{y_i} \left(\frac{1}{1-e^{\theta^T \mathbf{x}}}\right)^{(1-y_i)}$$

MLE Logistic Regression

Log-Likelihood

$$\sum_{i=1}^{n}[y_i \log(h_{\boldsymbol{\theta}}(\mathbf{x}_i)\ ) + (1 - y_i) \log(1 - h_{\boldsymbol{\theta}}(\mathbf{x}_i)\ )]$$

$$= \cdots$$

$$= -\sum_{i=1}^{n} \log\left(1 + e^{-y_i \boldsymbol{\theta}^T \mathbf{x}_i}\right)$$

Derivation / further readings:
http://cs229.stanford.edu/extra-notes/loss-functions.pdf
http://cs229.stanford.edu/notes2020spring/cs229-notes1.pdf
https://en.wikipedia.org/wiki/Logistic_regression

Optimization

$$\hat{\theta} = \arg\min_{\theta \in \mathbb{R}^d} \sum_{i=1}^{n} \log\left(1 + e^{-y_i \theta^T \mathbf{x}_i}\right)$$

How do we get $\hat{\theta}$?

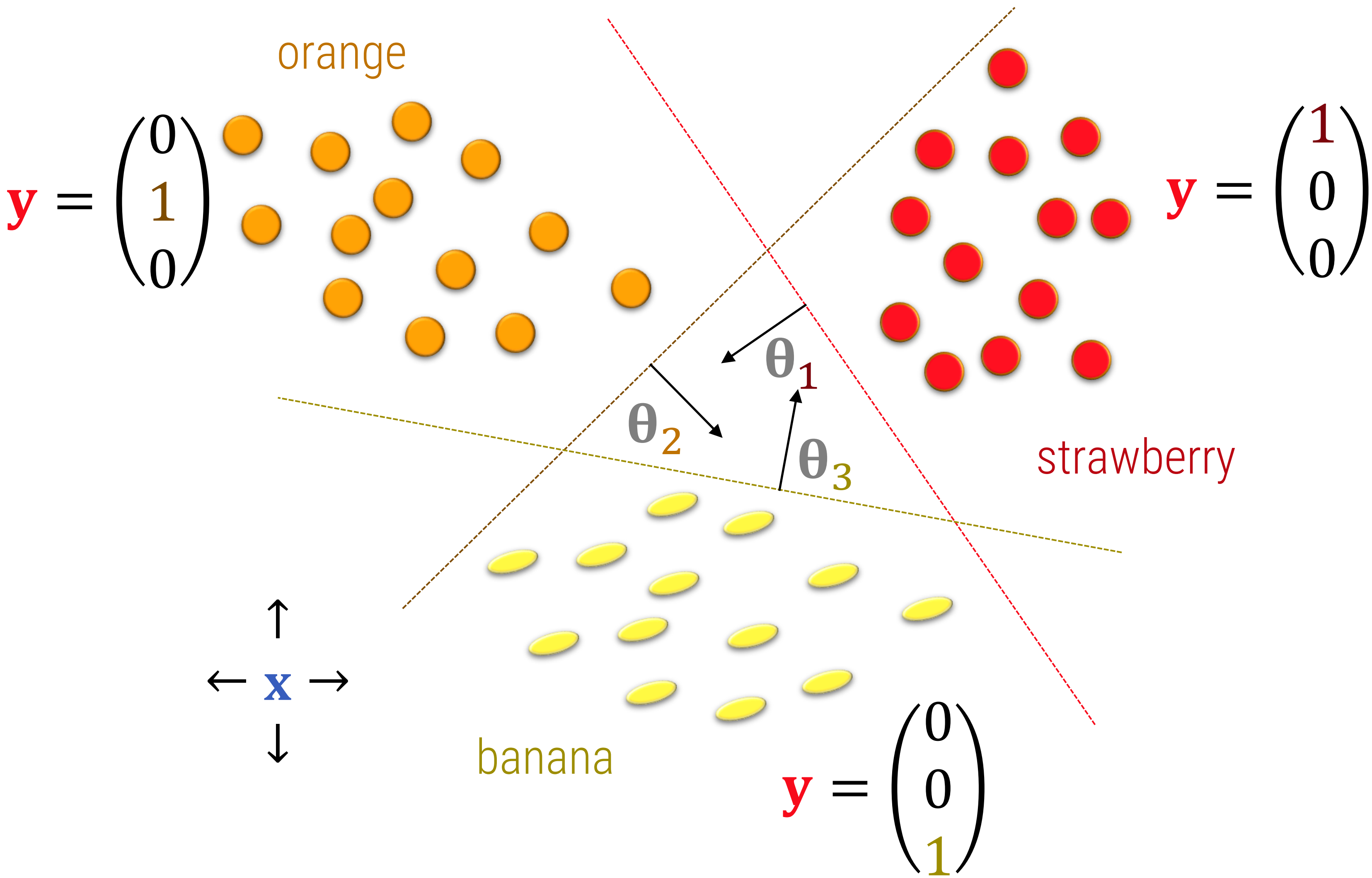
- Non-quadratic objective
  - Non-linear optimization problem
- Fortunately, the function is convex
  - Sigmoid is convex
  - Sum of convex functions
- Numerical optimization
  - Gradient descent -or- (quasi-) Newton methods
  - Stochastic (batch-) gradient descent for "big data"

Multi-Label Case

Task

- Again, n Data points, indexed by $i = 1 \ldots n$
  - Data $\mathbf{x}_i \in \mathbb{R}^d$ with…
  - …label vectors $\mathbf{y}_i \in \{0,1\}^K$
    - "One hot vectors"
    - Only one entry is 1 (correct class), the rest is zero
- Learn class-specific parameters $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K \in \mathbb{R}^d$

Geometry



orange
$\mathbf{y} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$
$\mathbf{y} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$
θ_1
θ_2
θ_3
strawberry
$\mathbf{x}$
banana
$\mathbf{y} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

Multi-Label Case

Replace sigmoid function $\sigma\colon \mathbb{R} \to \mathbb{R}$

$$\sigma(z) := \frac{1}{1 - e^{-(z)}} = \frac{e^{z}}{e^{z} + 1}$$

by "softmax" function $\boldsymbol{\sigma}\colon \mathbb{R}^K \to \mathbb{R}^K$

$$\boldsymbol{\sigma}(\mathbf{z}) := \begin{pmatrix} \frac{e^{z_1}}{\sum_{j=1}^{K} e^{z_j}} \\ \vdots \\ \frac{e^{z_K}}{\sum_{j=1}^{K} e^{z_j}} \end{pmatrix}, \qquad \sigma_m(\mathbf{z}) := \frac{e^{z_m}}{\sum_{j=1}^{K} e^{z_j}}$$

(38)

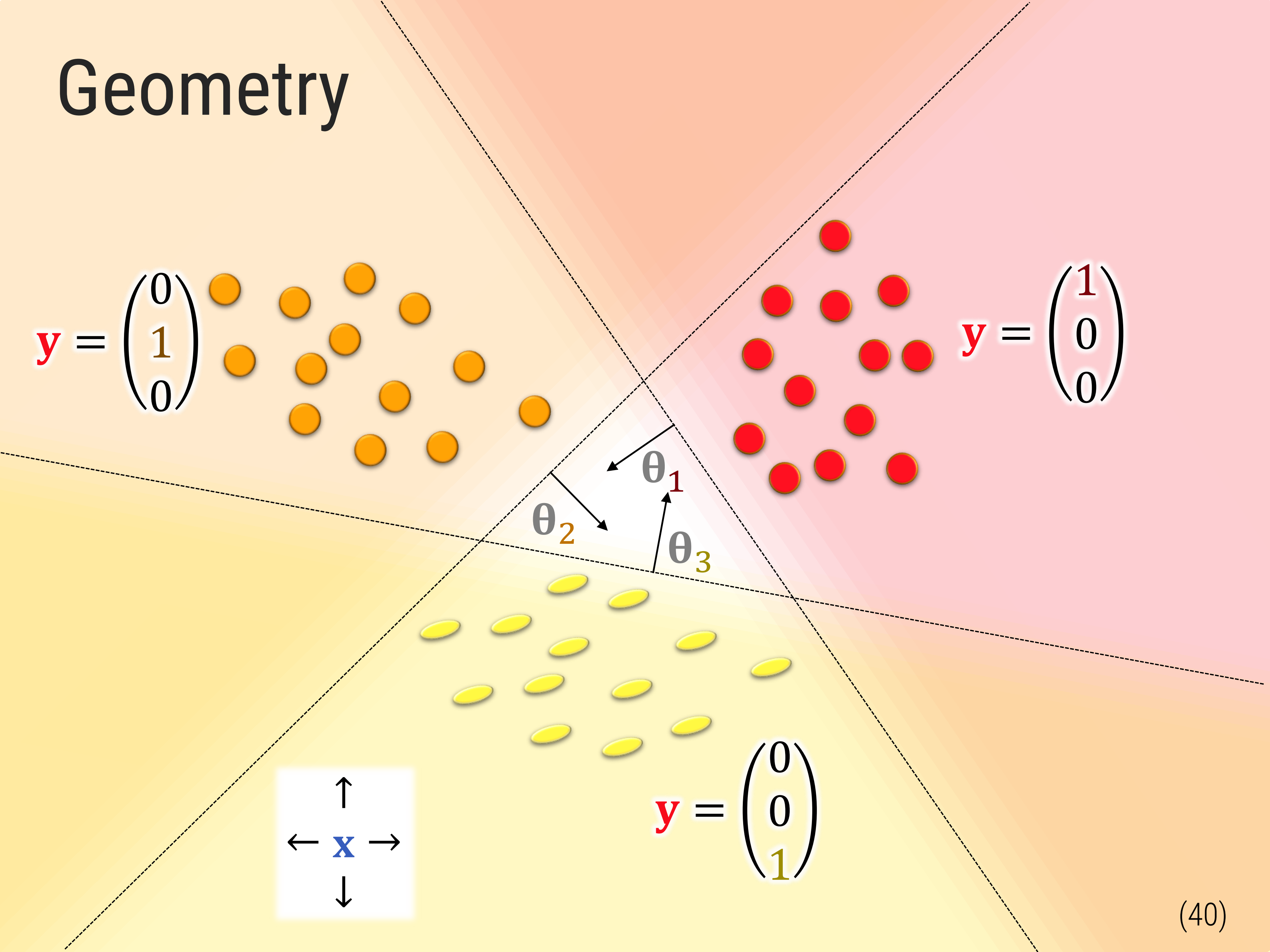
Classifier

Classifier

$$h_{\theta}(\mathbf{x}) \coloneqq \boldsymbol{\sigma}\left(\underbrace{\begin{bmatrix} \boldsymbol{\theta}_1^T \cdot \mathbf{x} \\ \vdots \\ \boldsymbol{\theta}_K^T \cdot \mathbf{x} \end{bmatrix}}_{\mathbf{u}(\boldsymbol{\theta},\mathbf{x})}\right) = \boldsymbol{\sigma}(\mathbf{u}(\boldsymbol{\theta}, \mathbf{x}))$$

- Outputs class-probabilities
  - All output vector entries in [0,1]
  - Entries sum up to one

Geometry



$\mathbf{y} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$
$\mathbf{y} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$
θ_1
θ_2
θ_3
$\mathbf{y} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$
$\mathbf{x}$

Classifier

- MLE-Training via

$$\arg\min_{\theta \in \mathbb{R}^{K \times d}} \sum_{i=1}^{n} \left[\underbrace{\log\left(\sum_{j=1}^{K} e^{\theta_j^T \cdot \mathbf{x}}\right)}_{\text{normalization factor } Z} - \sum_{m=1}^{K} \underbrace{\mathbf{y}_{i,m}}_{\text{1 only for correct class}} \cdot \underbrace{\log \sigma_m(\mathbf{u}(\theta, \mathbf{x}))}_{\text{(neg)-log-likelihood of correct class}} \right]$$

$$= \arg\min_{\theta \in \mathbb{R}^{K \times d}} \sum_{i=1}^{n} \left[\underbrace{\log(Z)}_{\text{normalization}} - \underbrace{\log \sigma_{class_i}(\mathbf{u}(\theta, \mathbf{x}))}_{\text{(neg)-log-likelihood of correct class}} \right]$$

Support Vector Machines

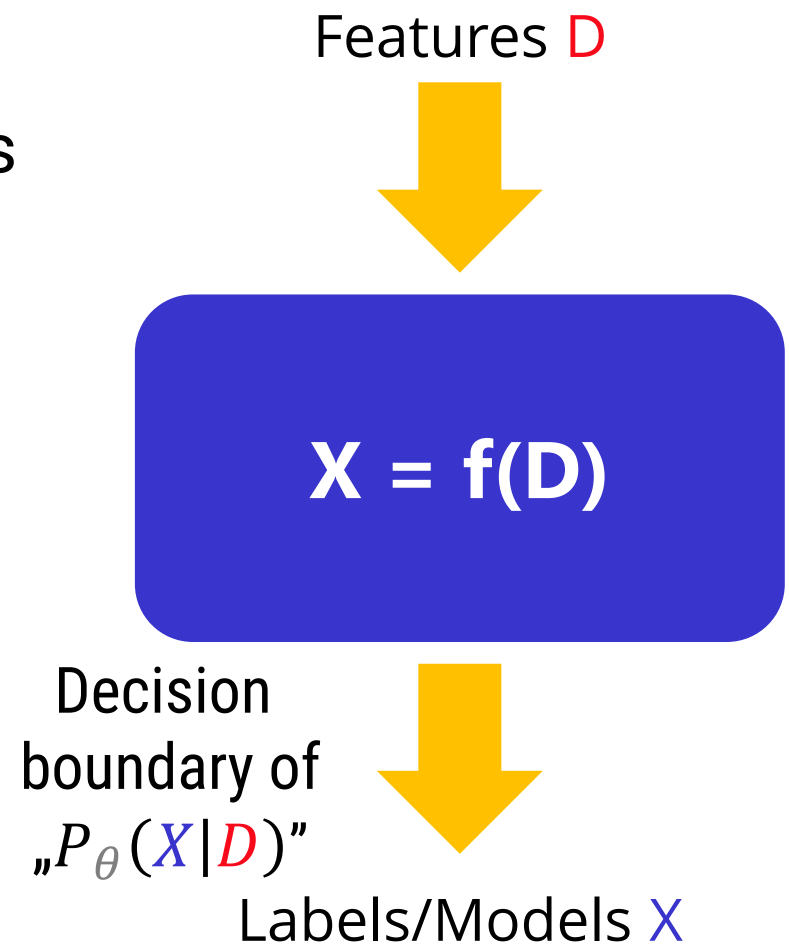
Discriminative Learning

Not strictly statistical

- Optimize for good decisions

Black box classifer

- Input: features
- Output: jugdement
  - Make it work!
- Discriminative model
  - No distributions, no posterior
  - No sampling from posterior
  - No generative model



Linear SVMs

Support Vector Machine

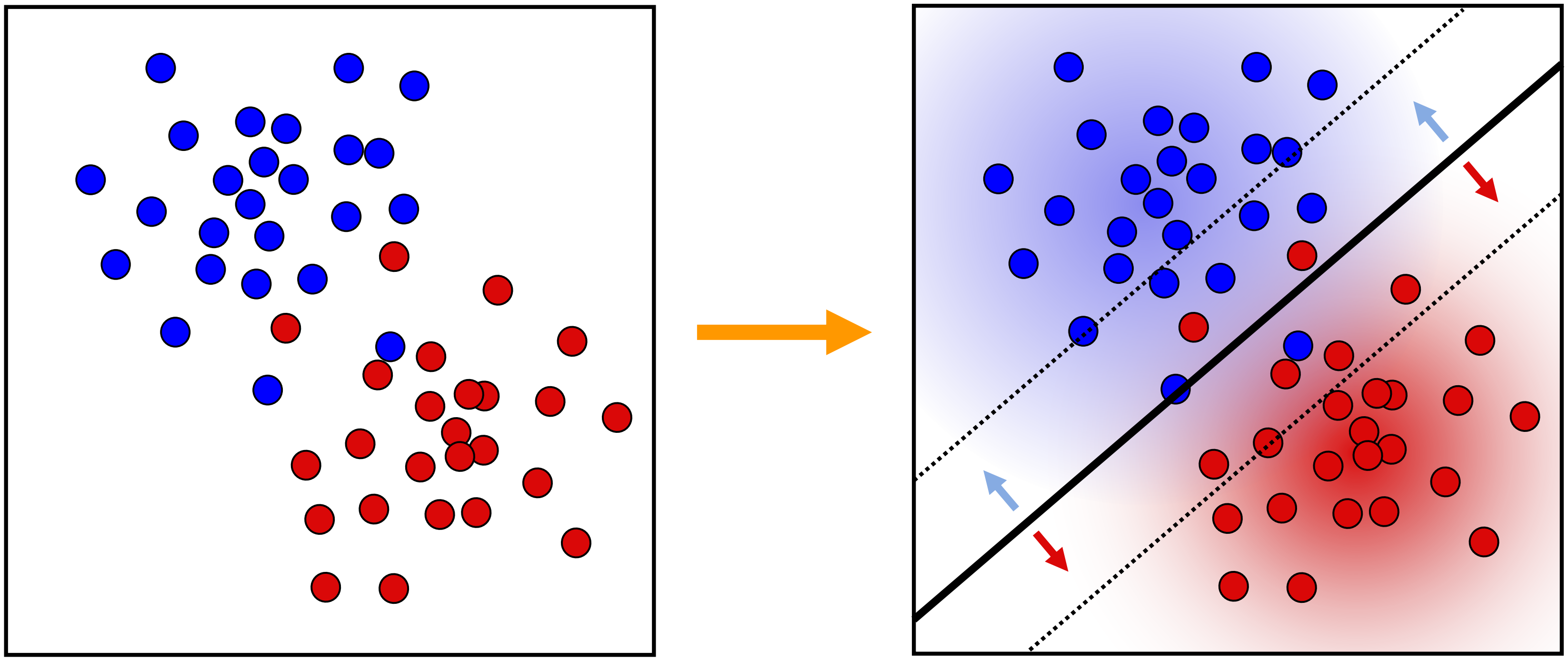
- Consider two labels $x \in \{-1,1\}$
- Data $\mathbf{d} \in \mathbb{R}^n$
- Classifier

$$x = f(\mathbf{d}) = \langle \mathbf{d}, \boldsymbol{\theta} \rangle + \theta_0$$

Training

- Maximize margin between classes ($x = -1,\ x = 1$)

Support Vector Machines



training set

separating hyperplane, minimal penetration of margin (L_1)

Linear SVMs

Support Vector Machine

- Consider two labels $y \in \{-1, 1\}$
- Data $\mathbf{x} \in \mathbb{R}^d$
- Classifier

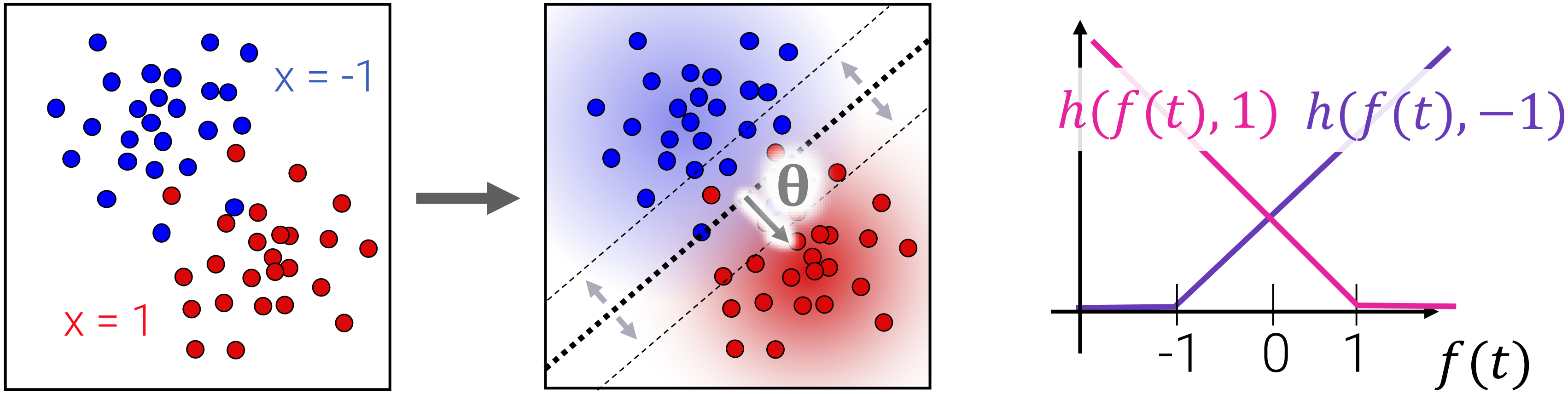
$$y = f(\mathbf{x}) = \langle \mathbf{x}, \boldsymbol{\theta} \rangle + \theta_0$$

to optimize:
parameters $\boldsymbol{\theta}, \theta_0$

Training

- Maximize margin between classes $(y = -1,\ y = 1)$

Linear SVMs



Classifier

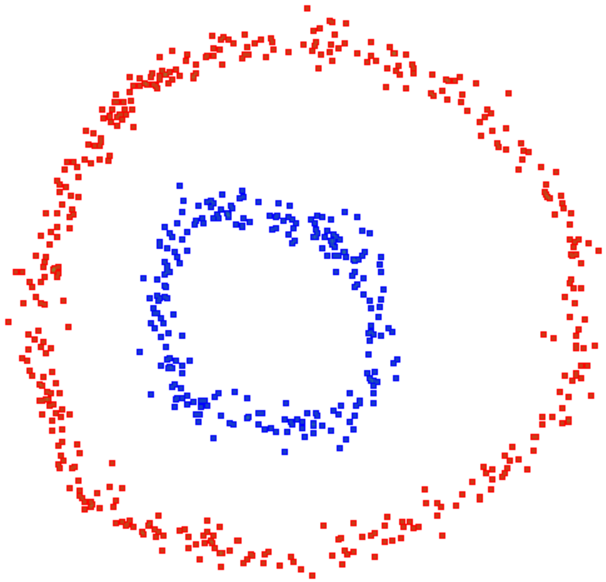
$$y = f(\mathbf{x}) = \langle \mathbf{x}, \boldsymbol{\theta} \rangle + \theta_0$$

Hinge loss

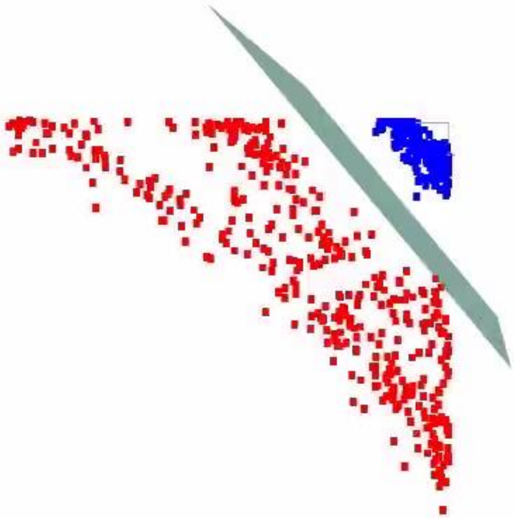
$$(\boldsymbol{\theta}, \theta_0) = \underset{(\boldsymbol{\theta}, \theta_0) \in \mathbb{R}^{d+1}}{\arg\min} \; C \cdot \left[\sum_{i=1}^{n} \underbrace{\max\big(0, 1 - y_i f(\mathbf{x}_i)\big)}_{h(t, x_i)} \right] + \|\boldsymbol{\theta}\|^2 \tag{47}$$

Kernel Support Vector Machine

Example Mapping:



original space



“feature space”

$$\phi\colon \mathbb{R}^2 \to \mathbb{R}^3$$

$$(x, y) \mapsto \left(x^2, xy, y^2\right)$$

Algorithm

Consider Gram Matrix

$$\mathbf{G} = \begin{pmatrix} \langle \phi(x_1), \phi(x_1) \rangle & \cdots & \langle \phi(x_n), \phi(x_1) \rangle \\ \vdots & \ddots & \vdots \\ \langle \phi(x_1), \phi(x_n) \rangle & \cdots & \langle \phi(x_n), \phi(x_n) \rangle \end{pmatrix}$$

$$= \begin{pmatrix} k(x_1, x_1) & \cdots & k(x_n, x_1) \\ \vdots & \ddots & \vdots \\ k(x_1, x_n) & \cdots & k(x_n, x_n) \end{pmatrix}$$

← often easier to compute than via $\phi(x), \phi(y) \rightarrow \langle \phi(x), \phi(y) \rangle$

Factorize Gram Matrix

- $G = X^T X$ (spectral embedding / MDS)
- Apply linear SVM, as we know it

Standard Kernels

Polynomial Kernel

- $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^d$
- Implicitly creates all multivariate polynomials up to degree d

Exponential Kernel

- $k(\mathbf{x}, \mathbf{y}) = \exp(- (\mathbf{x} - \mathbf{y})^2 / \sigma^2)$
- Infinite dimensional feature space (clustering by density)

Video #05b

Summary

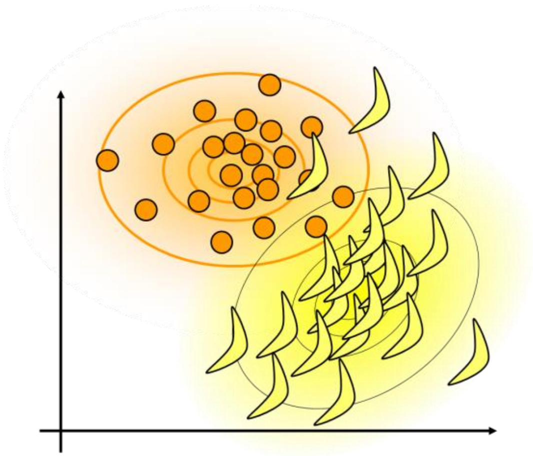
Summary

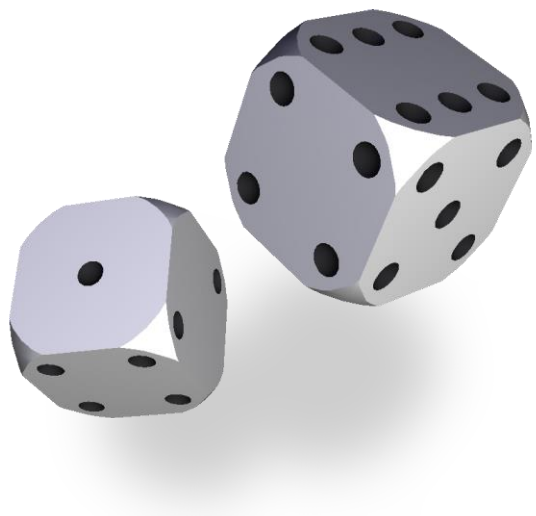
Basic classification methods

- Linear model (log-likelihood)
- Two objectives
  - Probabilistic: odds-ratios / logistic regression
  - Geometric: Max-margin (SVM)
- Convex numerical optimization
- Both work in practice
  - SVM works well with kernelization
  - Similar performance in “deep” architectures

Modelling 2

STATISTICAL DATA MODELLING





Chapter 5

Bayesian Data Analysis & Classical ML

Michael Wand · Institut für Informatik, JGU Mainz · michael.wand@uni-mainz.de

Video #05

Statistics & Machine Learning

Classical Machine Learning

- **Modeling 1 Recap:** LS, PCA
- **Old-School:** Classical Classifiers

Bayesian Data Analysis

- **Example 1:** MAP Image Reconstruction
- **Example 2:** Bayesian Regression

Bayesian
Data Analysis

Example 1: **Variational Reconstruction** (Mod-1)
(MAP-solution w/priors)

Image Reconstruction Model

Problem statement

- Measured 2D pixel image
- Distorted by noise
- Want to remove noise

Bayesian problem modeling

- Model of measurement process
  - Hand-crafted, not learned from data
- Prior distribution on images (thus "Bayesian")

Inference: Maximum-a-posteriori

Model

Image

- $x_{i,j}$ with $i = 1 \dots w, j = 1, \dots, h$
- continuous model: $f: [1, w] \times [1, h] \to \mathbb{R}$

Probability space

- $\Omega = \mathbb{R}^{w \times h}$
- Probability density on $\mathbb{R}^{w \times h}$
- Continuous model "$f \in \mathbb{R}^{[0,w]\times[0,h]}$" would be "*mathematically involved*"
  - We consider only finite-dimensional densities

Model

Bayes rule

$$P(X|D) \sim P(D|X) \cdot P(X)$$

Likelihood

- $P(D|X) = \prod_{i=1}^{w} \prod_{j=1}^{h} \mathcal{N}_{d_i,\sigma_D}(x_i)$ (i.i.d. Gaussian noise)

$$= \prod_{i=1}^{w} \prod_{j=1}^{h} \left[\frac{1}{\sigma_D \sqrt{2\pi}} e^{-\frac{(x_i - d_i)^2}{2\sigma_D^2}} \right]$$

(Gaussian distribution)

- Not so unrealistic
  - Real cameras: Poisson distribution + Gaussian circuit noise
  - "Realistic" model: $\sigma_i \sim x_i + \sigma_0$

Model

Likelihood

- $P(D|X) = \prod_{i=1}^{w} \prod_{j=1}^{h} \left[\frac{1}{\sigma_D \sqrt{2\pi}} e^{-\frac{(x_i - d_i)^2}{2\sigma_D^2}} \right]$

Neg-Log-Likelihood

$$E(D|X) := -\ln P(D|X) = \sum_{i=1}^{w} \sum_{j=1}^{h} \frac{(x_i - d_i)^2}{2\sigma_D^2} + \frac{wh}{\sigma_D \sqrt{2\pi}}$$

independent of x_i

Least Squares!

Model

Prior

- Assumption: Large image gradients are unlikely
  - Gaussian distribution on Gradients
  - Neg-log-likelihood: $\frac{1}{2\sigma^2}\|\nabla f\|^2$
- Discretization

$$E(X) := -\ln P(X) = \sum_{i=1}^{w-1}\sum_{j=1}^{h-1} \frac{\left(x_{i+1,j} - x_{i,j}\right)^2 + \left(x_{i,j+1} - x_{i,j}\right)^2}{2\sigma_X^2} + \frac{wh}{\sigma_X\sqrt{2\pi}}$$

independent of x_i

- (This is not very realistic)
  - (But what can you do?)
  - (This is *very Bayesian*)

Minimization Problem

Minimize

$$E(D|X) + \mathrm{E}(X)$$

$$= \sum_{i=1}^{w}\sum_{j=1}^{h}\frac{(x_i - d_i)^2}{2\sigma_D^2} + \sum_{i=1}^{w-1}\sum_{j=1}^{h-1}\frac{\left(x_{i+1,j} - x_{i,j}\right)^2 + \left(x_{i,j+1} - x_{i,j}\right)^2}{2\sigma_X^2}$$

Equivalent minimization objective

$$\sum_{i=1}^{w}\sum_{j=1}^{h}(x_i - d_i)^2 + \frac{\sigma_X^2}{\sigma_D^2}\sum_{i=1}^{w-1}\sum_{j=1}^{h-1}\left(x_{i+1,j} - x_{i,j}\right)^2 + \left(x_{i,j+1} - x_{i,j}\right)^2$$

Continuous

$$\int_\Omega \left(f(\mathbf{x}) - d(\mathbf{x})\right)^2 d\mathbf{x} + \frac{\sigma_X^2}{\sigma_D^2}\int_\Omega \|\nabla f(\mathbf{x})\|^2 d\mathbf{x}$$

Technical Remark

Image Prior

$$-\ln P(X) = \sum_{i=1}^{w-1} \sum_{j=1}^{h-1} \frac{\left(x_{i+1,j} - x_{i,j}\right)^2 + \left(x_{i,j+1} - x_{i,j}\right)^2}{2\sigma_X^2} + \frac{wh}{\sigma_X \sqrt{2\pi}}$$

- This is an "improper prior"
  - Does not integrate to one!
  - Infinite subspaces without penalty
- Formal fix
  - Assume broader prior on function value itself: $f \sim N_{0,\sigma_{very\ large}}$
- For MAP estimation, this does not matter
  - We just find a point of maximum density
  - Integration not required

Solution

Derivative of objective function

- Regularizer is a Laplace matrix (Euler-Lagrange-Eq.)
- Data term is an identity matrix + rhs = target values

$$\text{solve}\ \left(\mathbf{I} + \frac{\sigma_X^2}{\sigma_D^2}\mathbf{L}\right)\mathbf{x} = \mathbf{d}$$

Linear system of equations

- Setup sparse linear system
- Solve using iterative solver (e.g. conjugate gradients)
- **Remark:** shift-invariant system can be solved directly using Fourier transform (no LSE)

Modeling I

Looks familiar?

- Seen in Modeling 1

Variant

- Penalize l_1 norm instead of l_2 norm of gradients

$$\int_\Omega \left(f(\mathbf{x}) - d(\mathbf{x})\right)^2 d\mathbf{x} + \frac{\sigma_X^2}{\sigma_D^2} \int_\Omega \|\nabla f(\mathbf{x})\|^1 d\mathbf{x}$$

- Laplace distribution (double exponential)
  - Yields sharper images
  - Justification: natural image statistics*)
  - Simplest solution via IRLS (iteratively reweighted quadr. solver)

*) Daniel L. Ruderman: The statistics of natural images. In: Network: Computation in Neural Systems (5) 1994 517-548.


ground truth
reconstruction (Gaussian prior)
input (noisy)
reconstruction (Laplace prior)

Video #05c

Summary

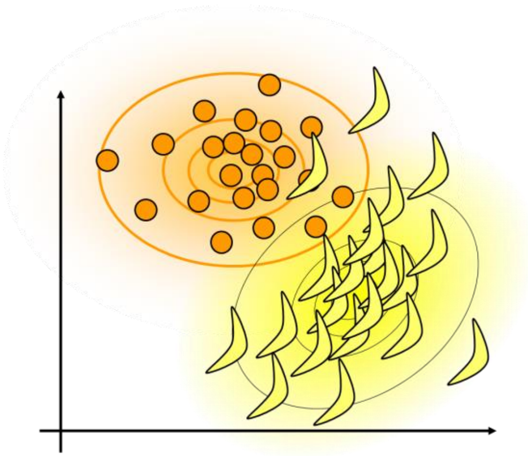
Summary

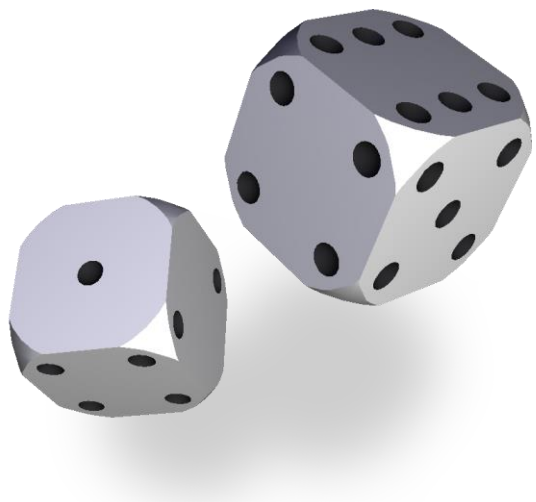
Image Reconstruction Example

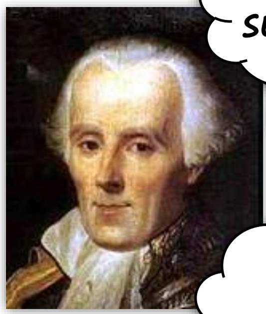
- Modelling 1: Interpretation as “regularization” of inverse problem
- Modelling 2: Maximum-A-Priori Estimation with a natural image prior
  - Gaussian Noise
  - Statistics of image gradients
- Unrealistic Prior
  - Only gradient statistics
  - Very Low-dimensional projection/approximation
- Nonetheless: Correct statistics matters

Modelling 2

STATISTICAL DATA MODELLING







Chapter 4

Statistics and Machine Learning

Michael Wand · Institut für Informatik, JGU Mainz · michael.wand@uni-mainz.de

Video #05

Statistics & Machine Learning

Classical Machine Learning

- **Modeling 1 Recap:** LS, PCA
- **Old-School:** Classical Classifiers

Bayesian Data Analysis

- **Example 1:** MAP Image Reconstruction
- **Example 2:** Bayesian Regression

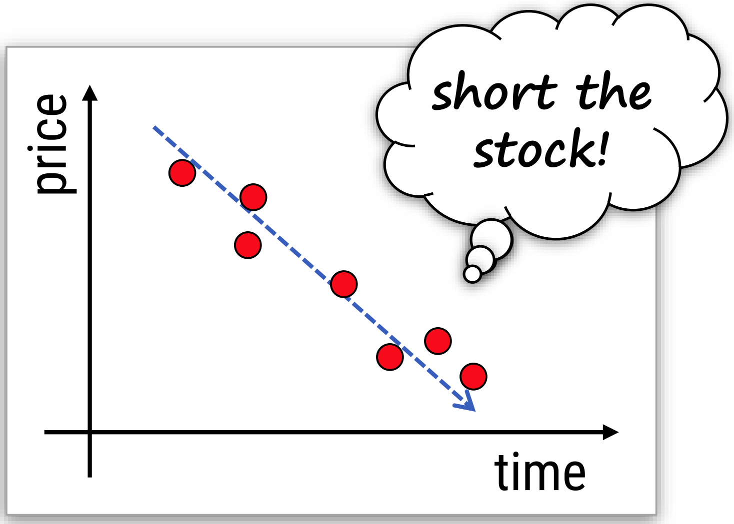
Bayesian Data Analysis

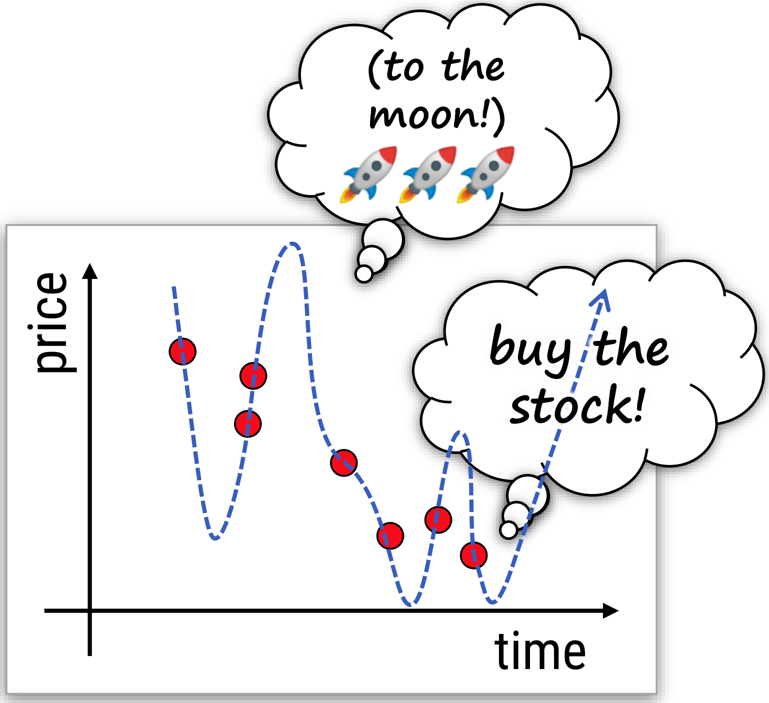
Example 2: Bayesian Regression
(full inference w/model averaging)

Example: Regression

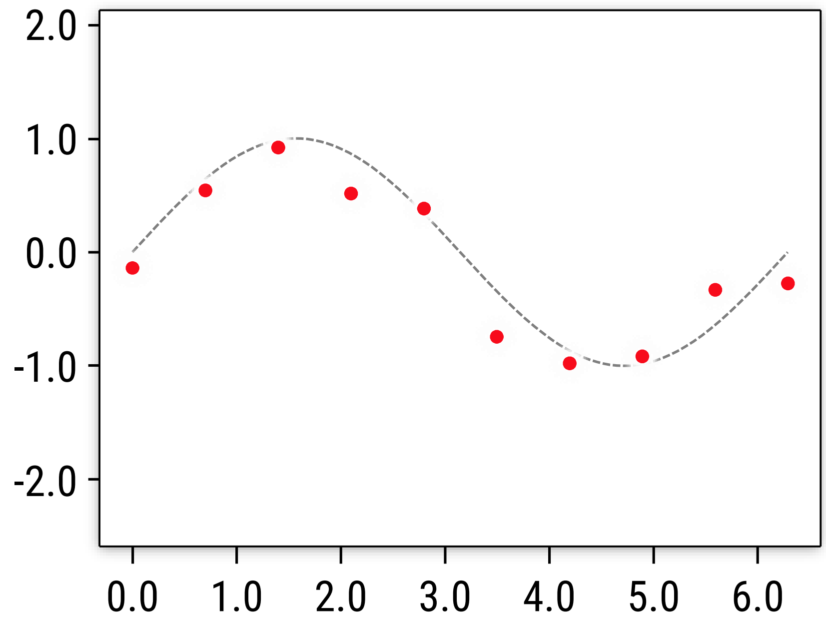
Regression example

- We do not know how smooth the curve should be
- Using marginalization for "model selection"





Data & Model



Data

- 10 samples from sine curve

$$\mathbf{x} = \left(0,\ \frac{1}{9}\pi,\ \frac{2}{9}\pi,\ \ldots,\ \pi\right), \qquad y_i = \sin(x_i) + \eta_i, \qquad i = 1 \ldots 10$$

- Distorted by random noise η_i
  - Additive, Gaussian, i.i.d., $\sigma = 0.2$, unbiased ($\mu = 0$)

Data & Model

Model

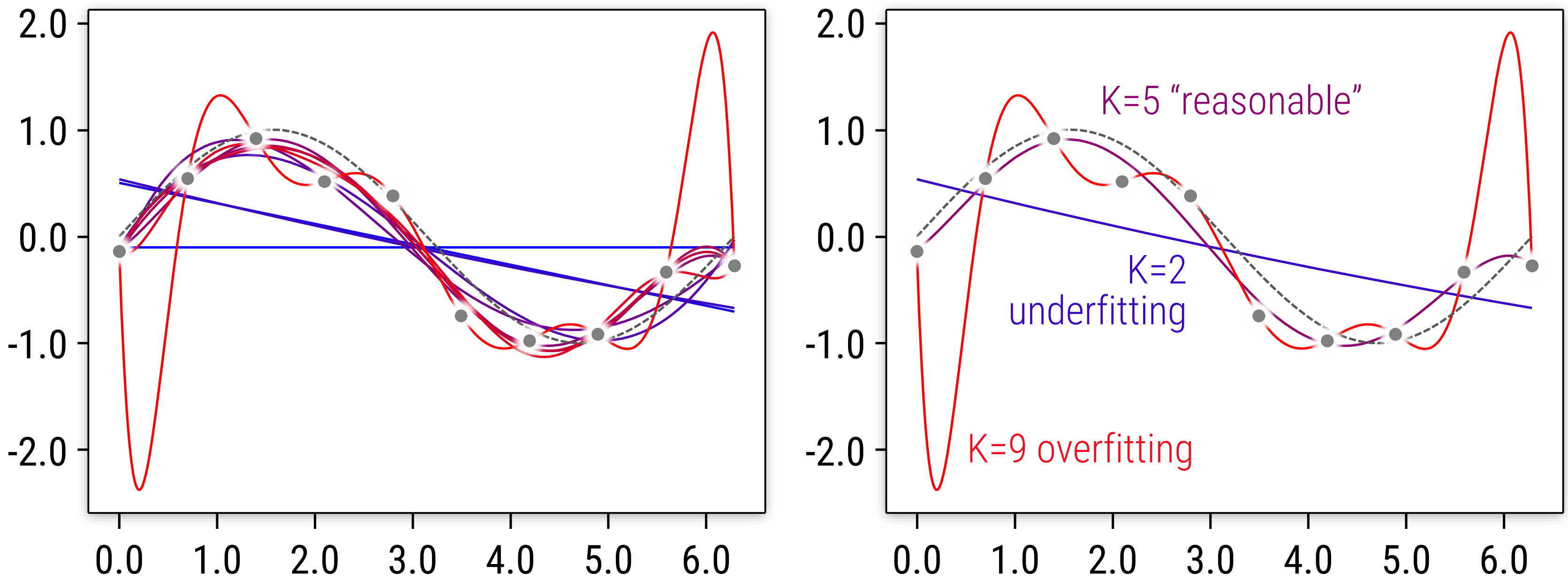
- Polynomial of degree K (with $0 \leq K \leq 9$)

$$f_{\mathbf{c}}^{(K)}(x) = \sum_{k=0}^{K} c_k \, x^k$$

for data $\mathbf{D} = (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^n$, $\mathbf{c} \in \mathbb{R}^D$

- We do not fix degree K, but use marginalization (Bayesian model averaging) over K

Polynomial Least-Squares Fit (MLE)



Degree 0 to 9

Degrees 2, 5, 9

Bayesian Inference

Abstract inference rule

$$\bar{X} = \mathbb{E}_{X \sim P(X|D)}[X] = \int_{\Omega(\theta)} \bar{X}_\theta \cdot P(\theta|D)\, d\theta$$

$$= \frac{1}{P(D)} \int_{\Omega(\theta)} \bar{X}_\theta \cdot P(D|\theta) \cdot P(\theta)\, d\theta$$

Bayesian Inference

Abstract inference rule

$$\bar{X} = \mathbb{E}_{X \sim P(X|D)}[X] = \int_{\Omega(\theta)} \underbrace{\hat{X}_\theta}_{\substack{\text{mean} \\ \text{for fixed } \theta}} \cdot \underbrace{P(\theta|D)}_{\substack{\text{likelihood of } \theta \\ \text{given the data}}} \, d\theta$$

$$= \overbrace{\frac{1}{P(D)}}^{\text{normalization}} \int_{\Omega(\theta)} \underbrace{\hat{X}_\theta}_{\substack{\text{mean} \\ \text{for fixed } \theta}} \cdot \overbrace{P(D|\theta)}^{\substack{\text{likelihood of the data} \\ \text{given } \theta}} \cdot \underbrace{P(\theta)}_{\substack{\text{prior} \\ \text{for } \theta}} \, d\theta$$

Bayesian Inference

Abstract inference rule

$$\bar{c} = \mathbb{E}_{c \sim P(c|D)}[c] = \underbrace{\frac{1}{P(D)}}_{\text{normalization}} \sum_{K=0}^{9} \underbrace{\hat{c}_K}_{\text{MAP estimate for fixed } K} \cdot \underbrace{P(D|K)}_{\text{likelihood of the data given } K} \cdot \underbrace{P(K)}_{\text{prior for } K}$$

Bayesian Inference

Abstract inference rule

normalize ad hoc

likelihood of the data given K

$$\bar{c} = \mathbb{E}_{c \sim P(c|D)}[c] = \frac{1}{P(D)} \sum_{K=0}^{9} \widehat{c}_K \cdot P(D|K) \cdot P(K)$$

MAP estimate for fixed K

uniform prior for K

Computing the Likelihood

Data likelihood

- $P(D|K)$ – marginal likelihood for fixed K
  - Fixed K
- We obtain

$$P(D|K) = \int_{\mathbf{c}_K \in \mathbb{R}^K} P(D|\mathbf{c}_K, K)\, P(\mathbf{c}_K|K) d\mathbf{c}_K$$

Computing the Likelihood

$$P(D|K) = \int_{\mathbf{c}_K \in \mathbb{R}^K} P(D|\mathbf{c}_K, K)\, P(\mathbf{c}_K|K)$$

Plugging in model assumptions

- Data has normal-distributed noise
- Simple Gaussian prior

$$P(D|\mathbf{c}_K, K) = \prod_{i=1}^{n} \mathcal{N}_{0,\sigma_D}\left(f_{\mathbf{c}_K}(x_i) - y_i\right)$$

$$P(\mathbf{c}_K|K) = \mathcal{N}_{0,\sigma_c \cdot \mathbf{I}_K}(\mathbf{c}_K)$$

$\sigma_D = 0.2$, σ_c large (prior), $\mathbf{I}_K$ = identity in $\mathbb{R}^{K \times K}$

Computing the Likelihood

Data likelihood

- For fixed K and in our case $X = \mathbf{c}$

$$P(D|\mathbf{c}_K, K) = \left(\prod_{i=1}^{n} \mathcal{N}_{0,\sigma_D}\left(f_{\mathbf{c}_K}(x_i) - y_i\right)\right) \mathcal{N}_{0,\sigma_c \cdot \mathbf{I}_K}(\mathbf{c}_K)$$

with $f_{\mathbf{c}_K}(x_i) = \underbrace{\left(x_i^0, \ldots, x_i^d, \ldots, x_i^K\right)}_{\boldsymbol{\xi}_i^T} \cdot \mathbf{c}_K = \boldsymbol{\xi}_i^T \cdot \mathbf{c}_K$:

$$\prod_{i=1}^{n} \mathcal{N}_{0,\sigma_D}\left(f_{\mathbf{c}_K}(x_i) - y_i\right) = \frac{1}{\sigma_D^n (2\pi)^{\frac{n+K}{2}}} \cdot \prod_{i=1}^{n} e^{-\frac{\left(\boldsymbol{\xi}_i^T \mathbf{c}_K - y_i\right)^2}{2\sigma_D{}^2}}$$

Computing the Likelihood

Data likelihood

$$\prod_{i=1}^{n} \mathcal{N}_{0,\sigma_D}\left(f_{\mathbf{c}_K}(x_i) - y_i\right) = \prod_{i=1}^{n} \frac{1}{\sigma_D\sqrt{2\pi}} e^{-\frac{1}{2\sigma_D^2}\left(\boldsymbol{\xi}_i^T \mathbf{c}_K - y_i\right)^2}$$

$$= \left(\sigma_D^2 2\pi\right)^{-\frac{n}{2}} \cdot \prod_{i=1}^{n} e^{-\frac{1}{2\sigma_D^2}\left(\mathbf{c}_K^T \boldsymbol{\xi}_i \boldsymbol{\xi}_i^T \mathbf{c}_K - 2\mathbf{c}_K \boldsymbol{\xi}_i^T y_i + y_i^2\right)}$$

$$= \left(\sigma_D^2 2\pi\right)^{-\frac{n}{2}} \cdot e^{-\frac{1}{2\sigma_D^2}\left(\mathbf{c}_K^T \underbrace{\sum_{i=1}^{n} \boldsymbol{\xi}_i \boldsymbol{\xi}_i^T}_{\mathbf{A}} \mathbf{c}_K - 2\mathbf{c}_K \underbrace{\sum_{i=1}^{n} \boldsymbol{\xi}_i^T y_i}_{\mathbf{b}} + \sum_{i=1}^{n} y_i^2\right)}$$

$$= \left(\sigma_D^2 2\pi\right)^{-\frac{n}{2}} \cdot e^{-\frac{1}{2\sigma_D^2}\left(\mathbf{c}_K^T \mathbf{A} \mathbf{c}_K - 2\mathbf{c}_K \mathbf{b} + \sum_{i=1}^{n} y_i^2\right)}$$

Mod 1: $\hat{\mathbf{c}}_K = \mathbf{A}^{-1}\mathbf{b}$

$$= \left(\sigma_D^2 2\pi\right)^{-\frac{n}{2}} \cdot e^{-\frac{1}{2\sigma_D^2}\left((\mathbf{c}_K - \hat{\mathbf{c}}_K)^T \mathbf{A} (\mathbf{c}_K - \hat{\mathbf{c}}_K) - \hat{\mathbf{c}}_K^2 + \sum_{i=1}^{n} y_i^2\right)}$$

Computing the Likelihood

Data likelihood

$$P(D|K) = \int_{\mathbf{c}_K \in \mathbb{R}^K} \boldsymbol{P}(D|\mathbf{c}_K, K)\, P(\mathbf{c}_K|K) d\mathbf{c}_K$$

$$= (\sigma_D^2 2\pi)^{-\frac{n}{2}} (\sigma_c^2 2\pi)^{-\frac{K}{2}} \cdot \int_{\mathbf{c}_K \in \mathbb{R}^K} e^{-\frac{1}{2\sigma_D^2}\left((\mathbf{c}_K - \hat{\mathbf{c}}_K)^T \mathbf{A} (\mathbf{c}_K - \hat{\mathbf{c}}_K) - \hat{\mathbf{c}}_K^2 + \sum_{i=1}^n y_i^2\right)} e^{-\frac{(\mathbf{c}_K)^2}{2\sigma_c^2}} d\mathbf{c}_K$$

$$= (\sigma_D^2 2\pi)^{-\frac{n}{2}} (\sigma_c^2 2\pi)^{-\frac{K}{2}} \cdot \int_{\mathbf{c}_K \in \mathbb{R}^K} e^{-\frac{1}{2\sigma_D^2}\left((\mathbf{c}_K - \hat{\mathbf{c}}_K)^T \left[\mathbf{A} + \frac{\sigma_D^2}{\sigma_c^2}\mathbf{I}\right] (\mathbf{c}_K - \hat{\mathbf{c}}_K) - \hat{\mathbf{c}}_K^2 + \sum_{i=1}^n y_i^2\right)} d\mathbf{c}_K$$

$$= (\sigma_D^2 2\pi)^{-\frac{n}{2}} (\sigma_c^2 2\pi)^{-\frac{K}{2}} \cdot e^{-\frac{1}{2\sigma_D^2}(\sum_{i=1}^n y_i^2 - \hat{\mathbf{c}}_K^2)} \cdot \int_{\mathbf{c}_K \in \mathbb{R}^K} e^{-\frac{1}{2\sigma_D^2}\left((\mathbf{c}_K - \hat{\mathbf{c}}_K)^T \left[\mathbf{A} + \frac{\sigma_D^2}{\sigma_c^2}\mathbf{I}\right] (\mathbf{c}_K - \hat{\mathbf{c}}_K)\right)} d\mathbf{c}_K$$

Computing the Likelihood

Data likelihood

$$P(D|K) = \int_{\mathbf{c}_K \in \mathbb{R}^K} \mathrm{P}(D|\mathbf{c}_K, K)\,\mathrm{P}(\mathbf{c}_K|K) d\mathbf{c}_K$$

$$= (\sigma_D^2 2\pi)^{-\frac{n}{2}} (\sigma_c^2 2\pi)^{-\frac{K}{2}} \cdot e^{-\frac{1}{2\sigma_D^2}(\sum_{i=1}^n y_i^2 - \hat{\mathbf{c}}_K^2)} \cdot \int_{\mathbf{c}_K \in \mathbb{R}^K} e^{-\frac{1}{2\sigma_D^2}\left((\mathbf{c}_K - \hat{\mathbf{c}}_K)^T \left[\mathbf{A} + \frac{\sigma_D^2}{\sigma_c^2}\mathbf{I}\right](\mathbf{c}_K - \hat{\mathbf{c}}_K)\right)} d\mathbf{c}_K$$

$$= (\sigma_D^2 2\pi)^{-\frac{n}{2}} (\sigma_c^2 2\pi)^{-\frac{K}{2}} \cdot e^{-\frac{1}{2\sigma_D^2}(\sum_{i=1}^n y_i^2 - \hat{\mathbf{c}}_K^2)} \cdot \underbrace{(2\pi)^{\frac{K}{2}} \det\left(\mathbf{A} + \frac{\sigma_D^2}{\sigma_c^2}\mathbf{I}\right)^{-\frac{1}{2}}}_{\substack{\text{normalization factor} \\ \text{for normal distribution}}}$$

$$= (\sigma_D^2 2\pi)^{-\frac{n}{2}} \sigma_c^{-K} \cdot e^{-\frac{1}{2\sigma_D^2}(\sum_{i=1}^n y_i^2 - \hat{\mathbf{c}}_K^2)} \cdot \det\left(\mathbf{A} + \frac{\sigma_D^2}{\sigma_c^2}\mathbf{I}\right)^{-\frac{1}{2}}$$

$$\sim \sigma_c^{-K} \cdot e^{-\frac{1}{2\sigma_D^2}(\sum_{i=1}^n y_i^2 - \hat{\mathbf{c}}_K^2)} \cdot \det\left(\mathbf{A} + \frac{\sigma_D^2}{\sigma_c^2}\mathbf{I}\right)^{-\frac{1}{2}}$$

Computing the Likelihood

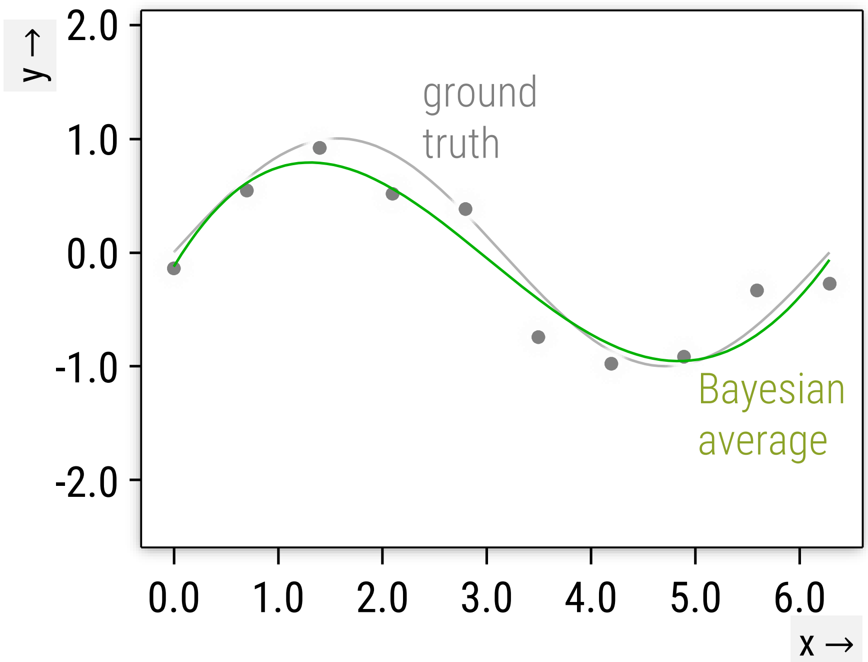
Data likelihood

$$P(D|K) \sim \sigma_c^{-K} \cdot e^{-\frac{1}{2\sigma_D^2}\left(\sum_{i=1}^{n} y_i^2 - \hat{\mathbf{c}}_K^2\right)} \cdot \det\left(\mathbf{A} + \frac{\sigma_D^2}{\sigma_c^2}\mathbf{I}\right)^{-\frac{1}{2}}$$

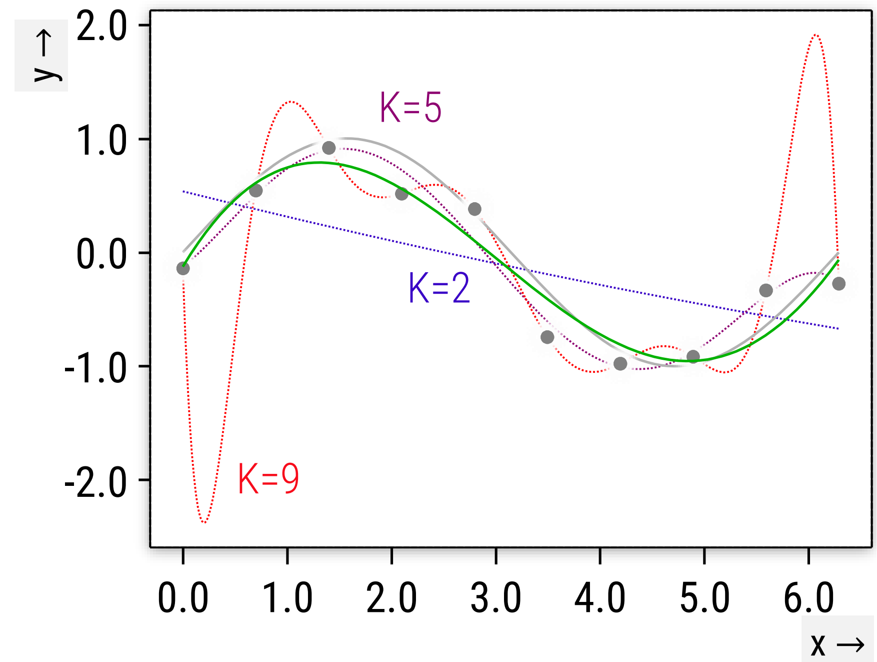
Flat (improper) Prior

$$P(D|K) \sim \underbrace{e^{-\frac{1}{2\sigma_D^2}\left(\sum_{i=1}^{n} y_i^2 - \hat{\mathbf{c}}_K^2\right)}}_{\text{data fit}} \cdot \underbrace{\det(\mathbf{A})^{-\frac{1}{2}}}_{\text{complexity penalty}}$$

Result

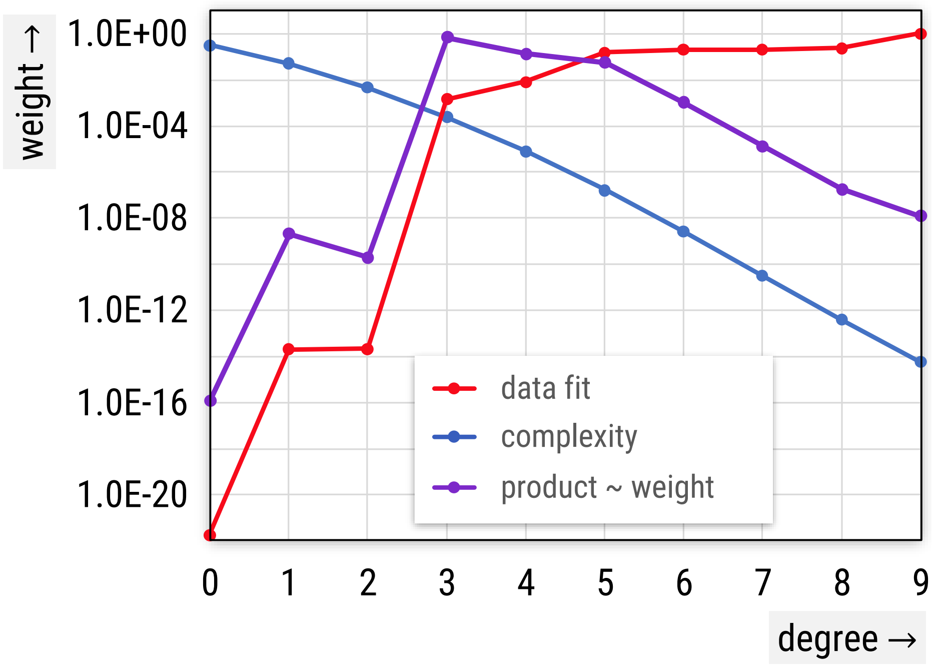


Model averaging

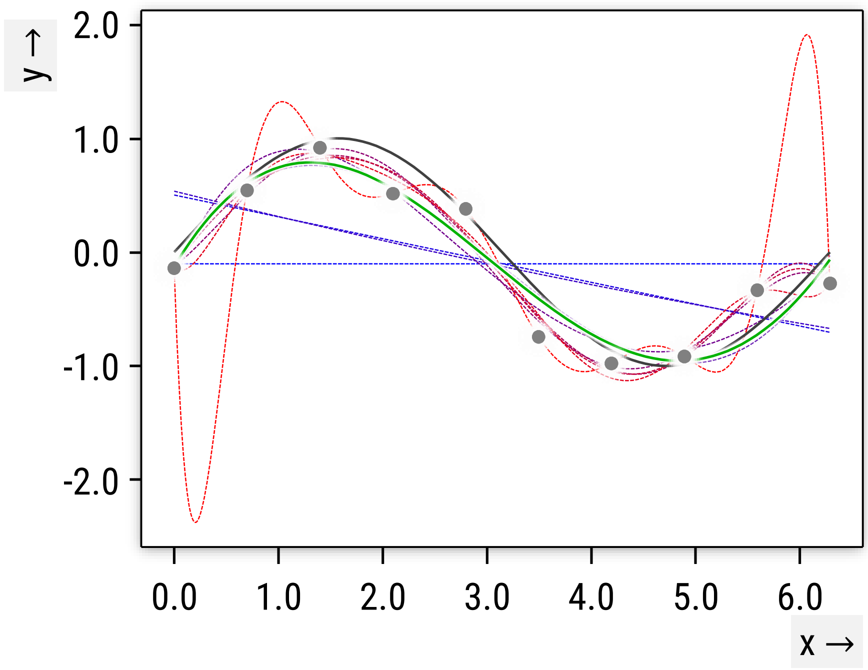


Degrees 2, 5, 9

Result



weighting



degree 0 to 9

Some Observations / Remarks

Bayesian inference

- Unknown parameter K, many possible models $\hat{X}_K$
- Weight model $\hat{X}_K$ by "evidence" $P(D|K)$ (marginal likelihood)
- Sum up (normalize weights, if not done yet)

Compute

$$\hat{X} = \int_{\Omega(K)} \hat{X}_K \cdot P(D|K)\, dK$$

Some Observations / Remarks

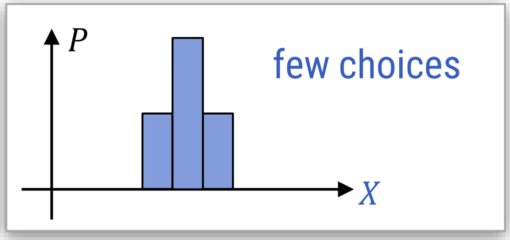
Structure of Marginal likelihood

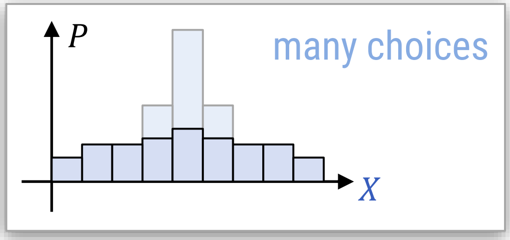
$$P(D) = \int_X \underbrace{\underbrace{P(D|X)}_{\text{quality of fit}} \quad \underbrace{P(X)}_{\text{model prior}}}_{P(D,X)} \; dX$$

(leaving out K for clarity)

What does it do?

- $P(D, X)$ contains two parts
  - Likelihood of the data (quality of fit)
  - Complexity penalty
    - Density $P(D, X)$ more spread out if X has much choice





Occam's Razor

"Full" Bayesian inference

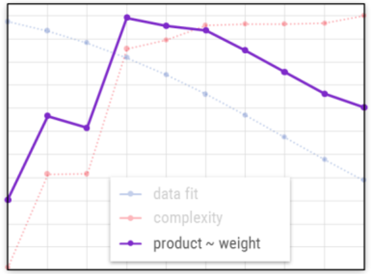
- Less weight on complex models
  - Equivalent to description length priors (more later)

Utility for Estimation

- Can also be use for model comparison: Compare marginal likelihoods ("evidences")

$$P(D|K_1) \text{ vs. } P(D|K_2)$$

- …and select more likely model
  - Evaluates trade-off between data-fit and complexity



Gaussian Models

Special structure for Gaussians

- Marginal Likelihood

$$P(D|K) \sim \underbrace{P(D|X = \bar{X}_K, K)}_{\substack{\text{data fit at} \\ \text{mean / peak}}} \cdot \underbrace{\det(\boldsymbol{\Sigma})^{-\frac{1}{2}}}_{\substack{\text{complexity} \\ \text{penalty}}}$$

- $\boldsymbol{\Sigma}$ = covariance matrix of the posterior
  - Will see later: $\det(\boldsymbol{\Sigma})^{-\frac{1}{2}}$ shrinks with growing information content

Video #05d

Summary

Summary

"Full Bayesian" Inference

- Reduced overfitting
  - Estimation methods are much more "risky"
- Amounts to weighting solutions by likelihood
  - "Bayesian model averaging"

Why does it help?

- Prefer simple models
- Model with many parameters θ (Here: high degree)
  - Model has more spread out density
  - Lower weight in likelihood weighting