Theoretical Concepts of Machine Learning
Part 1

Sepp Hochreiter

Institute of Bioinformatics
Johannes Kepler University, Linz, Austria

Course

3 ECTS 2 SWS VO (class)

1.5 ECTS 1 SWS UE (exercise)

Basic Course: Master Bioinformatics
Elective Course: Major Comp. Engineering Master Comp. Science

Class: Thu 15:30-17:00 (S3 055)

Exercise: Thu 14:30-15:15 (S3 55)

VO: **three part-exams**
UE: weekly homework (evaluated)

Other Courses

	Lecture		Lecturer	
365,077	**Machine Learning: Unsupervised Tech.**	VL	Hochreiter	Mon 15:30-17:00/HS 18
365,078	**Machine Learn.: Unsup. Tech. – G1**	UE	Hochreiter	Mon 13:45-14:30/S2 053
365,095	**Machine Learn.: Unsup. Tech. – G2+3**	UE	Hochreiter	Mon 14:30-15:15/S2 053
365,041	**Theoretical Concepts of Machine Learn.**	VL	Hochreiter	Thu 15:30-17:00/S3 055
365,042	**Theoretical Concepts of Machine Learn.**	UE	Hochreiter	Thu 14:30-15:15/S3 055
365,081	**Genome Analysis & Transcriptomics**	KV	Regl	Fri 8:30-11:00/S2 053
365,082	**Structural Bioinformatics**	KV	Regl	Tue 8:30-11:00/HS 11
365,093	**Deep Learning and Neural Networks**	KV	Unterthiner	Thu 10:15-11:45/MT 226
365,090	**Special Topics: Population genetics**	KV	Klambauer	block
365,096	**Special Topics: AI in Life Sciences**	KV	Klambauer	block
365,079	**Introduction to R**	KV	Bodenhofer	Wed 15:30-17:00/MT 127
365,067	Master's Seminar	SE	Hochreiter	Mon 10:15-11:45/S3 318
365,080	Master's Thesis Seminar SS	SE	Hochreiter	Mon 10:15-11:45/S3 318
365,091	Bachelor's Seminar	SE	Hochreiter	-
365,019	Dissertantenseminar Informatik 3	SE	Hochreiter	Mon 10:15-11:45/S3 318
347,337	Bachelor Sem. Bio. Chemistry (+Thesis)	SE	Hochreiter	-
347,327	Sem. in Structural and Comp. Biochemistry	SE	Hochreiter	-
365,083	Projektpraktikum	PR	Hochreiter	-

Outline

Outline

Outline

Outline

8 Bayes Techniques

8.1 Likelihood, Prior, Posterior, Evidence

8.2 Maximum A Posteriori Approach

8.3 Posterior Approximation

8.4 Error Bars and Confidence Intervals

8.5 Hyper-parameter Selection: Evidence Framework

8.6 Hyper-parameter Selection: Integrate Out

8.7 Model Comparison

8.8 Posterior Sampling

Literature

- **Stat. Lern. Th.:** V. N. Vapnik; Statistical Learning Theory, Wiley & Sons, 1998
- **Stat. Lern. Th.:** Schölkopf, Smola; Learning with kernels, MIT Press, 2002
- **Estimation Th.:** S. M. Kay; Fundamentals of Statistical Signal Processing, Prentice Hall, 1993
- **Estimation Th.:** M. I. Jordan (ed.); Learning in Graphical Models, MIT Press, 1998
- **ML:** Duda, Hart, Stork; Pattern Classification; Wiley & Sons, 2001
- **ML:** C. M. Bishop; Neural Networks for Pattern Recognition, Oxford Univ. Press, 1995
- **ML:** C. M. Bishop; Pattern Recognition and Machine Learning, Springer, 2006

Chapter 1

Introduction

Introduction

- part of curriculum “master of science in bioinformatics”
- part of curriculum “computer science” (major CE, major int. sys.)

- Machine learning major research topic: Google, Microsoft, Amazon, Facebook, AltaVista, Zalando, and many more

- Applications: computer vision (image recognition), speech recognition, recommender systems, analysis of Big Data, information retrieval

- Mining the web: search engines, social networks, videos, music

- Machine learning applications in biology and medicine:
  - microarrays, sequencing
  - alternative splicing, nucleosome positions, gene regulation
  - single nucleotide polymorphisms / variants (SNPs, SNVs)
  - copy number variations (CNVs)
  - diseases: Alzheimer, Parkinson, cancer, multiples sclerosis, schizophrenia or alcohol dependence

Introduction

This course introduces theoretical concepts machine learning methods.

maximum likelihood estimator is motivated by estimation theory:

- bias
- efficient
- Cramer-Rao lower bound
- Fisher information matrix
- consistent estimator

Consistent: more data lead to better results

Introduction

Statistical learning theory:

- empirical risk minimization
- complexity of model classes: VC-dim., growth, annealed entropy
- bounds on the generalization error: Vapnik, Chernoff, covering #
- structural risk minimization

Optimization

- gradient-based
- convex optimization

Bayes framework:

- estimate the posterior
- derive error bounds for model predictions
- optimize hyperparameters
    - integrating out the posterior
    - evidence framework

Introduction

We define quality criteria for selected models in order to pin down a goal for model selection, i.e. learning.

maximum likelihood:

- Unbiased
- Efficient

maximum likelihood in supervised learning: noise models

Introduction

A central question in machine learning is:
Does learning from examples help in the future?

Learning on the training data is called "empirical risk minimization" (ERM) in statistical learning theory.

Complexity is restricted: ERM helps
Under mild conditions the convergence is uniform and even fast, i.e. exponentially.

To measure the complexity of the model class we will introduce the VC-dimension (Vapnik-Chervonenkis).

theoretical upper bounds on the generalization error
→ "structural risk minimization"

We introduce basic techniques for minimizing the error that is techniques for model selection for a parameterized model class.

Introduction

- does learning from examples help in the future?
- "empirical risk minimization" (ERM)
- complexity is restricted and dynamics fixed
- "learning helps": more training examples improve the model
- converges to the best model for all future data
- convergence is fast
- complexity of a model class: VC-dimension (Vapnik-Chervonenkis)
- "structural risk minimization" (SRM): complexity and model quality
- bounds on the generalization error

Chapter 2

Generalization Error

Generalization Error

- quality criteria → goal for model selection / learning
- approximations
- unsupervised learning: maximum likelihood
- concepts: bias and variance, efficient estimator, Fisher information
- supervised learning considered in an unsupervised framework: error model

Model Quality

- learning equivalent to model selection
- quality criteria: future data is optimally processed
- other concepts: visualization, modeling, data compression
- Kohonen networks= self-organizing maps (SOMs):
  no scalar quality criterion (potential function)
- advantage quality criteria:
  - comparison of different models
  - quality during learning known
- supervised quality criteria: rate of misclassifications or squared error
- unsupervised criteria:
  - likelihood
  - ratio of between and within cluster distance
  - independence of the components
  - information content

Generalization Error: Definition

- performance of a model on future data: generalization error
- error on **one** example: loss or error
- expected loss: risk or generalization error

Generalization Error: Definition

Training set: $X = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^l\} \in \mathbb{R}^d$

Label or target value: $y^i \in \mathbb{R}$

Simple: $\boldsymbol{z} = (\boldsymbol{x}, y)$ and $\boldsymbol{z} \in Z = \mathbb{R}^{d+1}$

Training set: $\{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^l\}$

Matrix notation for training inputs: $\boldsymbol{X} = \left(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^l\right)^T$

Vector notation for labels: $\boldsymbol{y} = \left(y^1, \ldots, y^l\right)^T$

Matrix notation for training set: $\boldsymbol{Z} = \left(\boldsymbol{z}^1, \ldots, \boldsymbol{z}^l\right)$

Generalization Error: Definition

The loss function

quadratic loss: $$L(y, g(\boldsymbol{x}; w)) = (y - g(\boldsymbol{x}; w))^2$$

zero-one loss: $$L(y, g(\boldsymbol{x}; w)) = \begin{cases} 0 & \text{for} \quad y = g(\boldsymbol{x}; w) \\ 1 & \text{for} \quad y \neq g(\boldsymbol{x}; w) \end{cases}$$

Generalization error:

$$R(g(.; w)) = \mathrm{E}_{\boldsymbol{z}} \left(L(y, g(\boldsymbol{x}; w)) \right) = \int_Z L(y, g(\boldsymbol{x}; w)) \, p(\boldsymbol{z}) \, d\boldsymbol{z}$$

Generalization Error: Definition

y is a function of $\boldsymbol{x}$ (target function: $y = f(\boldsymbol{x})$) plus noise:

$$y = f(\boldsymbol{x}) + \epsilon$$

$$p(y \mid \boldsymbol{x}) = p_n(y - f(\boldsymbol{x}))$$

$$p(\boldsymbol{z}) = p(\boldsymbol{x})\ p(y \mid \boldsymbol{x}) = p(\boldsymbol{x})\ p_n(y - f(\boldsymbol{x}))$$

Now the risk can be computed as

$$R(g(.;w)) = \int_Z L(y, g(\boldsymbol{x}; w))\ p(\boldsymbol{x})\ p_n(y - f(\boldsymbol{x}))\ d\boldsymbol{z} =$$

$$\int_X p(\boldsymbol{x}) \int_{\mathbb{R}} L(y, g(\boldsymbol{x}; w))\ p_n(y - f(\boldsymbol{x}))\ dy\ d\boldsymbol{x}$$

Generalization Error: Definition

$$R(g(\boldsymbol{x}; w)) = \mathrm{E}_{y|\boldsymbol{x}} \left(L(y, g(\boldsymbol{x}; w)) \right) =$$

$$\int_{\mathbb{R}} L(y, g(\boldsymbol{x}; w)) \; p_n(y - f(\boldsymbol{x})) \; dy$$

Generalization Error: Definition

The noise-free case is $y = f(\boldsymbol{x})$

$$R(g(\boldsymbol{x}; w)) = L(f(\boldsymbol{x}), g(\boldsymbol{x}; w)) = L(y, g(\boldsymbol{x}; w))$$

simplifies to:

$$R(g(.; w)) = \int_X p(\boldsymbol{x}) L(f(\boldsymbol{x}), g(\boldsymbol{x}; w)) \, d\boldsymbol{x}$$

Generalization Error: Estimation

- $p(\boldsymbol{z})$ is unknown
- especially $p(y|\boldsymbol{x})$
- risk cannot be computed
- practical applications: approximation of the risk
- model performance estimation for the user

Generalization Error: Estimation

Test set approximation:

$$R(g(.;w)) \;=\; \mathrm{E}_{\boldsymbol{z}}\left(L(y, g(\boldsymbol{x};w))\right)$$

expectation can be approximated using

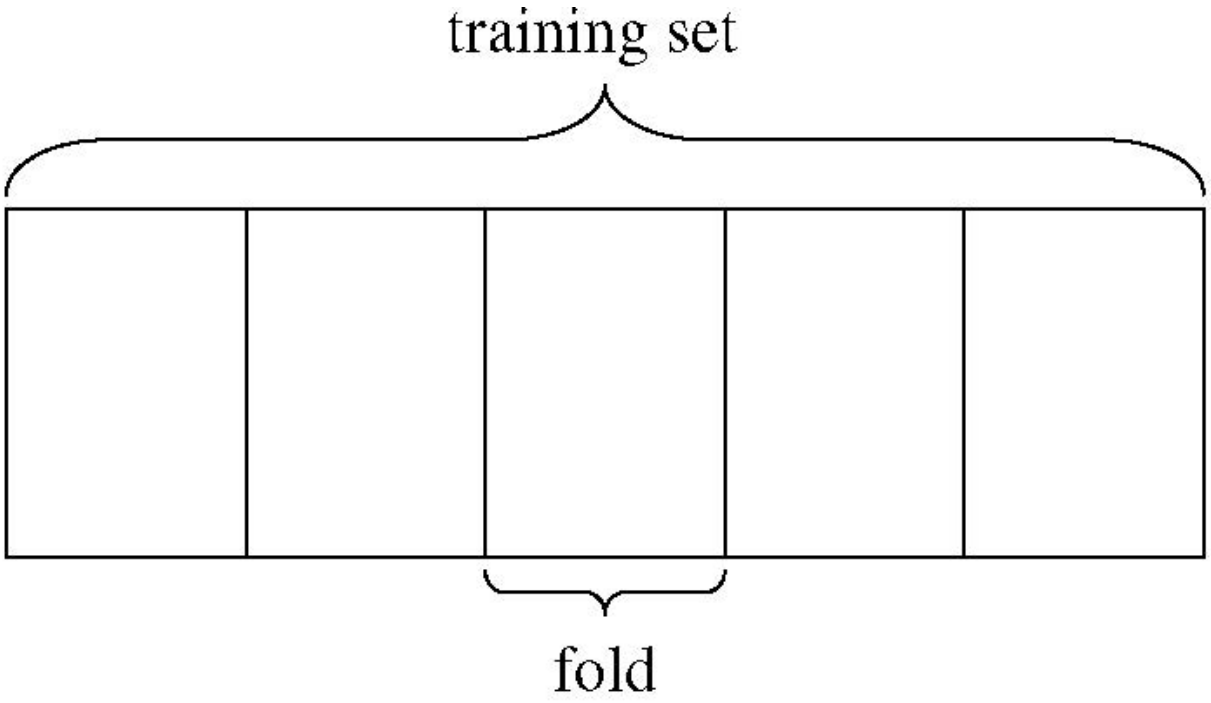
$$R(g(.;w)) \;\approx\; \frac{1}{m}\sum_{i=l+1}^{l+m} L\left(y^i, g(\boldsymbol{x}^i;w)\right)$$

with test set: $\left\{\boldsymbol{z}^{l+1}, \ldots, \boldsymbol{z}^{l+m}\right\}$

Generalization Error: Estimation

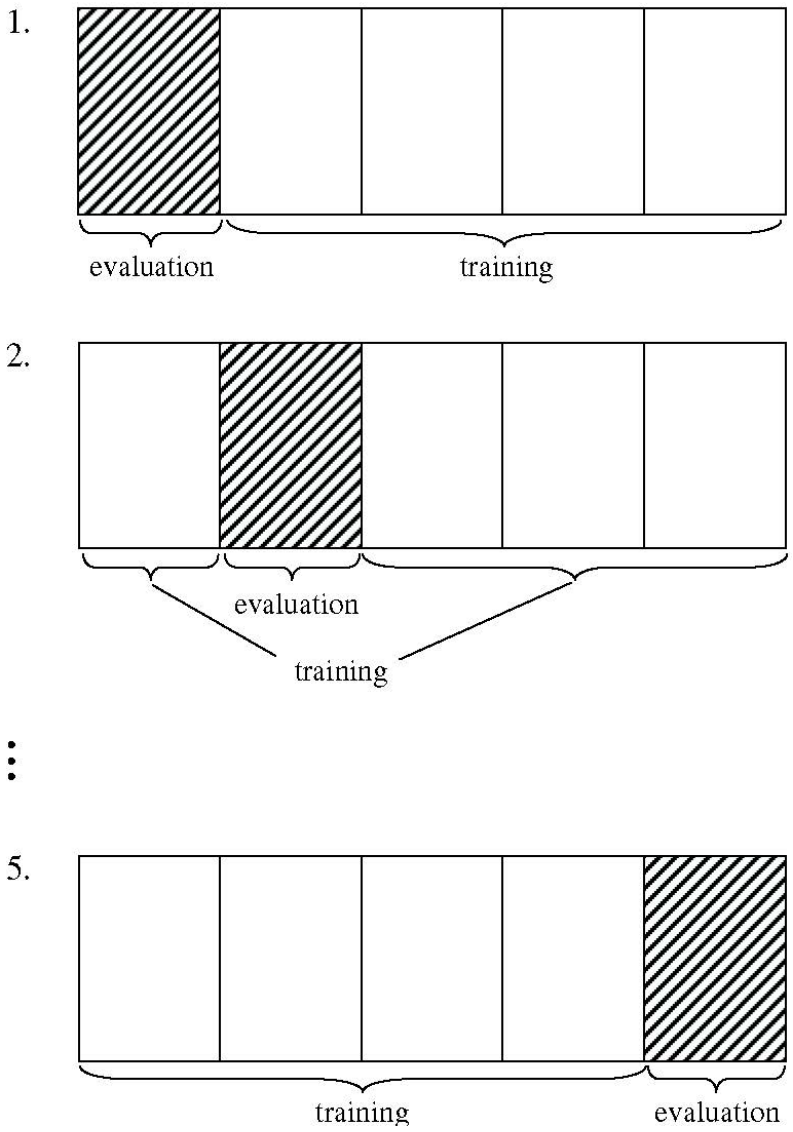
- not enough data for test set (needed for training)
- cross-validation

Cross-validation folds:



Generalization Error: Estimation

n-fold cross-validation (here 5-fold):



Generalization Error: Estimation

$$R_{n-\mathrm{cv}}(Z_l) = \frac{1}{n}\sum_{j=1}^{n} \underbrace{\frac{n}{l}\sum_{\boldsymbol{z}\in Z^j_{l/n}} \left(L\left(y, g\left(\boldsymbol{x}; w_j\left(Z_l \setminus Z^j_{l/n}\right)\right)\right)\right)}_{R_{n-\mathrm{cv},j}(Z_l)}$$

cross-validation is an almost unbiased estimator for the generalization error:

$$\mathrm{E}_{Z_{l(1-1/n)}}\left(L\left(g\left(.; w\left(Z_{l(1-1/n)}\right)\right)\right)\right) = \mathrm{E}_{Z_l}\left(R_{n-\mathrm{cv}}\left(Z_l\right)\right)$$

Generalization error on trainings size without one fold $l - l/n$ can be estimated by cross-validation on training data l by n-fold cross-validation

Generalization Error: Estimation

- advantage: test examples only once used
  (better than multiple dividing the data into training and test set)

- disadvantage:
  - training sets are overlapping
  - one fold on same model → test examples dependent
  - these dependencies → cv has high variance
    (one outlier influences all estimates)

- special case: leave-one-out cross-validation (LOO-CV)
  - l-fold cross-validation, where each fold is one example
  - test examples to not use the same model
  - training sets are maximal overlapping

Minimal Risk for a Gaussian Classification Task

Class y = 1 data points are drawn according to

$$p(\boldsymbol{x} \mid y = 1) \propto \mathcal{N}(\boldsymbol{\mu}_1, \Sigma_1)$$

and class y = -1 according to

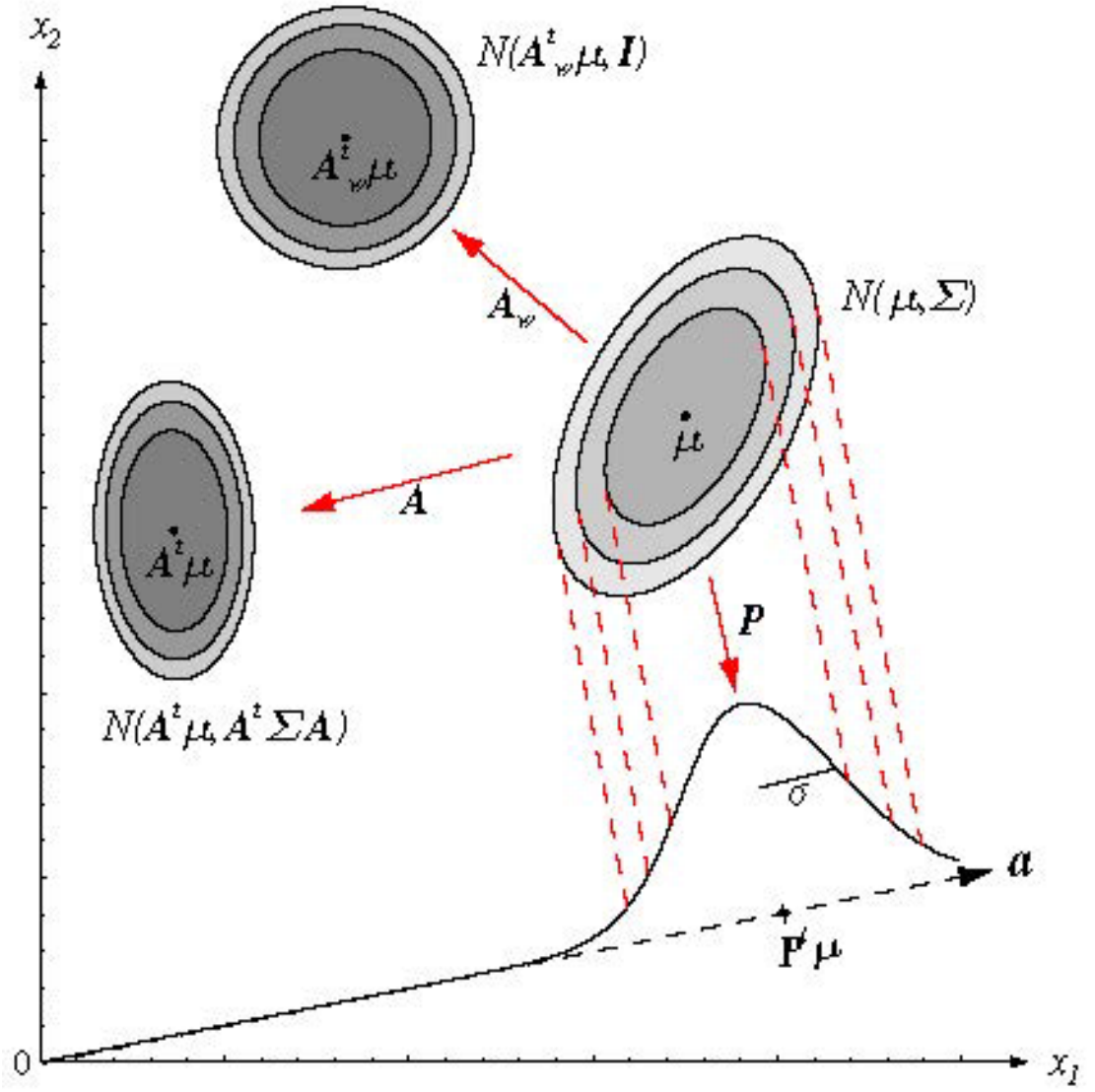
$$p(\boldsymbol{x} \mid y = -1) \propto \mathcal{N}(\boldsymbol{\mu}_{-1}, \Sigma_{-1})$$

where the Gaussian $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ has density

$$p(\boldsymbol{x}) = \frac{1}{(2\,\pi)^{d/2}\,|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}\,(\boldsymbol{x}-\boldsymbol{\mu})^T\,\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

Minimal Risk for a Gaussian Classification Task

Linear transformations of Gaussians lead to Gaussians



Minimal Risk for a Gaussian Classification Task

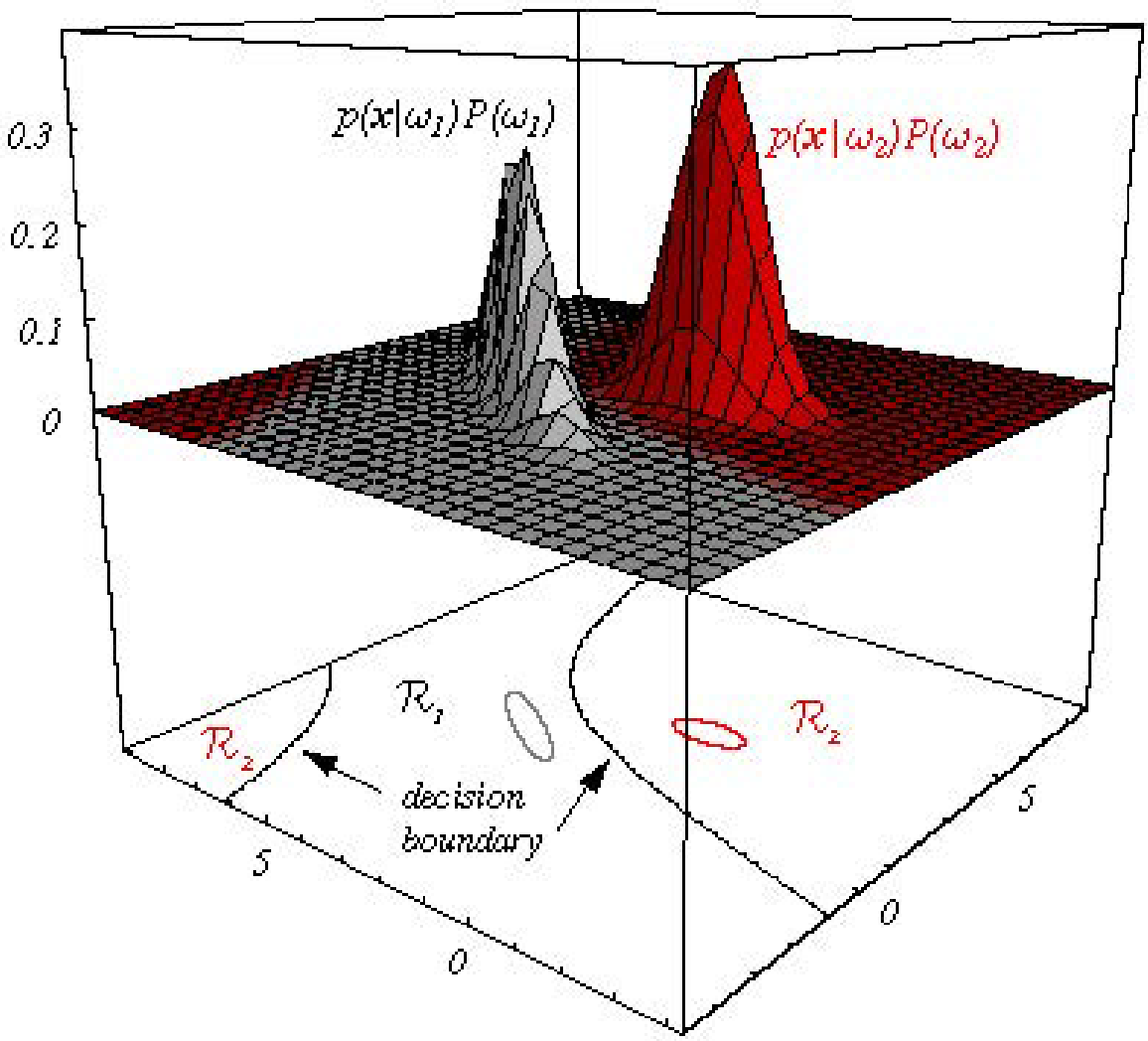
- probability of observing a point from class $y=1$ at $\boldsymbol{x}$: $p(\boldsymbol{x}, y = 1)$
- probability of observing a point from class $y=-1$ at $\boldsymbol{x}$: $p(\boldsymbol{x}, y = -1)$
- Conditional probability: $p(\boldsymbol{x}, y = 1) = p(\boldsymbol{x} \mid y = 1)\, p(y = 1)$
- probability of observing a point at $\boldsymbol{x}$:

$$p(\boldsymbol{x}) = p(\boldsymbol{x}, y = 1) + p(\boldsymbol{x}, y = -1)$$

y is “integrated out” - here “summed out”

Minimal Risk for a Gaussian Classification Task

- two-dimensional classification task
- data for each class from a Gaussian (black: class 1, red: class -1)
- optimal discrimina functions are two hyperbolas



Minimal Risk for a Gaussian Classification Task

- Bayes rule for probability of $\boldsymbol{x}$ belonging to class $y = 1$:

$$p(y = 1 \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid y = 1)\ p(y = 1)}{p(\boldsymbol{x})}$$

regions of predicted class $y = 1$:

$$X_1 = \{\boldsymbol{x} \mid g(\boldsymbol{x}) > 0\}$$

regions of predicted class $y = -1$:

$$X_{-1} = \{\boldsymbol{x} \mid g(\boldsymbol{x}) < 0\} \ .$$

loss function:

$$L(y, g(\boldsymbol{x}; \boldsymbol{w})) = \begin{cases} 0 & \text{for} \quad y\ g(\boldsymbol{x}; \boldsymbol{w}) > 0 \\ 1 & \text{for} \quad y\ g(\boldsymbol{x}; \boldsymbol{w}) < 0 \end{cases}$$

Minimal Risk for a Gaussian Classification Task

Risk: $R(g(.;w)) = \int_Z L(y, g(\boldsymbol{x}; w))\ p(\boldsymbol{z})\ d\boldsymbol{z}$

Loss function contributions:

$$X_1: \quad p(\boldsymbol{x}, y = -1)$$
$$X_{-1}: \quad p(\boldsymbol{x}, y = 1)$$

$$R(g(.;w)) = \int_{X_1} p(\boldsymbol{x}, y = -1)\ d\boldsymbol{x} + \int_{X_{-1}} p(\boldsymbol{x}, y = 1)\ d\boldsymbol{x} =$$

$$\int_{X_1} p(y = -1 \mid \boldsymbol{x})\ p(\boldsymbol{x})\ d\boldsymbol{x} + \int_{X_{-1}} p(y = 1 \mid \boldsymbol{x})\ p(\boldsymbol{x})\ d\boldsymbol{x} =$$

$$\int_X \left\{ \begin{array}{lll} p(y = -1 \mid \boldsymbol{x}) & \text{for} & g(\boldsymbol{x}) > 0 \\ p(y = 1 \mid \boldsymbol{x}) & \text{for} & g(\boldsymbol{x}) < 0 \end{array} \right\} p(\boldsymbol{x})\ d\boldsymbol{x}$$

Minimal Risk for a Gaussian Classification Task

Optimal discriminant (see later) function:

$$g(\boldsymbol{x}; w) \begin{cases} > 0 & \text{for} \quad p(y=1 \mid \boldsymbol{x}) > p(y=-1 \mid \boldsymbol{x}) \\ < 0 & \text{for} \quad p(y=-1 \mid \boldsymbol{x}) > p(y=1 \mid \boldsymbol{x}) \end{cases}$$

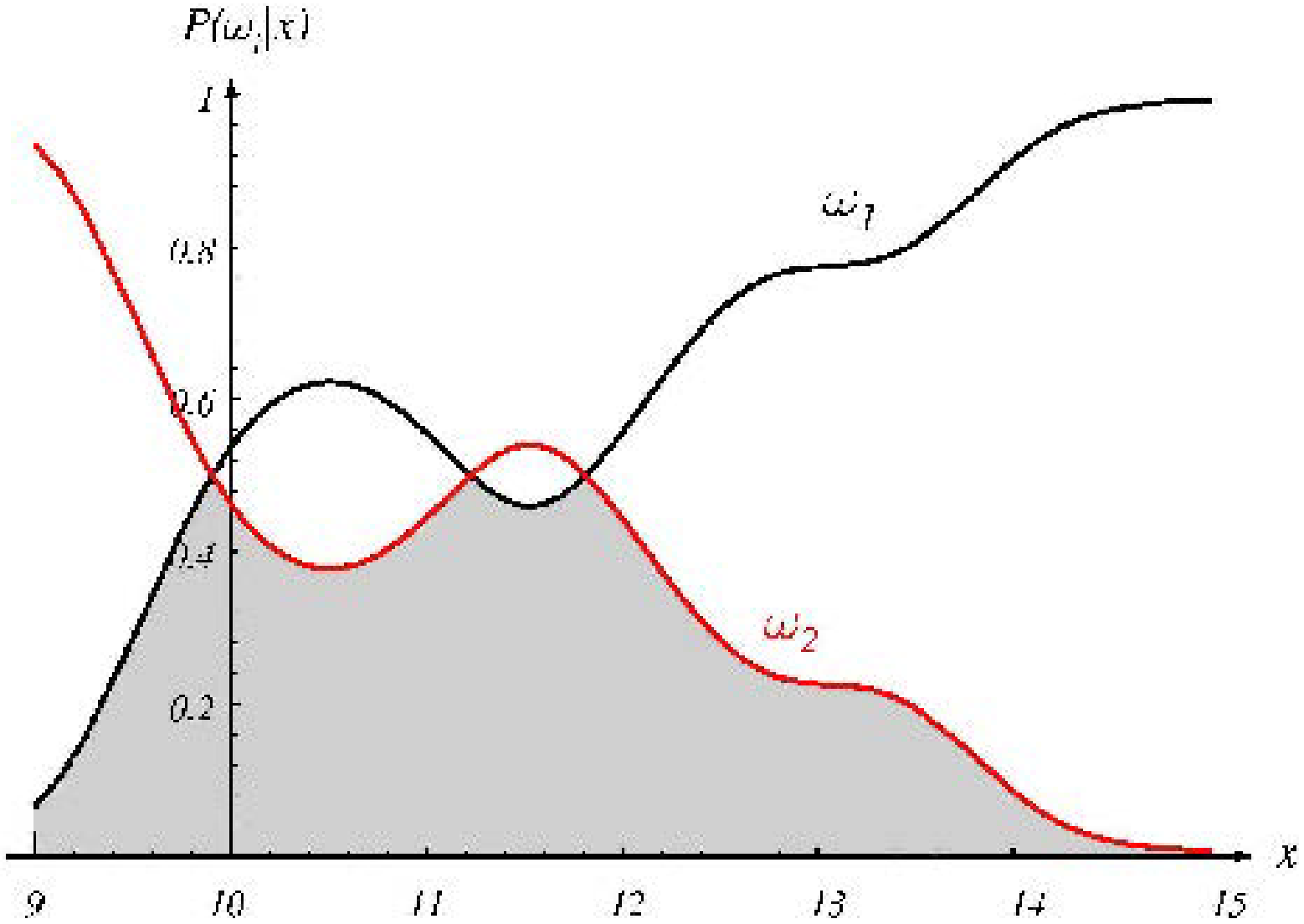
at each position $\boldsymbol{x}$ take smallest value

The minimal risk is

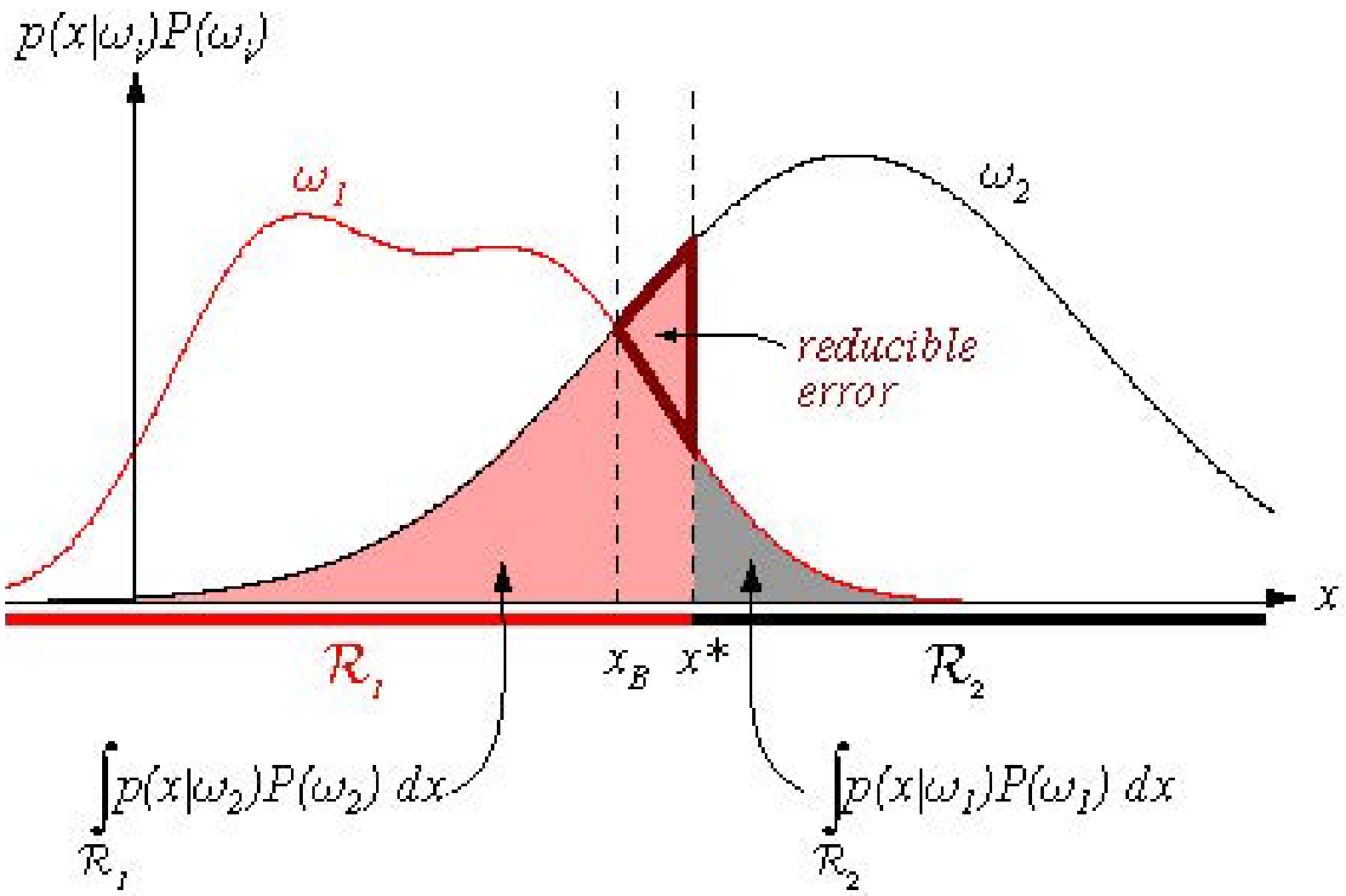
$$R_{\min} = \int_X \min\{p(\boldsymbol{x}, y=-1), p(\boldsymbol{x}, y=1)\} \, d\boldsymbol{x} =$$

$$\int_X \min\{p(y=-1 \mid \boldsymbol{x}), p(y=1 \mid \boldsymbol{x})\} \, p(\boldsymbol{x}) \, d\boldsymbol{x}$$

Minimal Risk for a Gaussian Classification Task



Minimal Risk for a Gaussian Classification Task



Minimal Risk for a Gaussian Classification Task

- discriminant function g: $g(\boldsymbol{x})>0$ then $\boldsymbol{x}$ is assigned to $y = 1$
  $g(\boldsymbol{x})<0$ then $\boldsymbol{x}$ is assigned to $y = -1$

- classification function $\hat{y}(\boldsymbol{x})$: $\hat{y}(\boldsymbol{x}) = \text{sign}(g(\boldsymbol{x}))$

- optimal discriminant functions (minimal risk):

$$g(\boldsymbol{x}) = p(y = 1 \mid \boldsymbol{x}) - p(y = -1 \mid \boldsymbol{x})$$

or

$$g(\boldsymbol{x}) = \ln p(y = 1 \mid \boldsymbol{x}) - \ln p(y = -1 \mid \boldsymbol{x}) =$$

$$\ln \frac{p(\boldsymbol{x} \mid y = 1)}{p(\boldsymbol{x} \mid y = -1)} + \ln \frac{p(y = 1)}{p(y = -1)}$$

Minimal Risk for a Gaussian Classification Task

For Gaussians:

$$g(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T \Sigma_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) - \frac{d}{2}\ln 2\pi -$$

$$\frac{1}{2}\ln|\Sigma_1| + \ln p(y=1) +$$

$$\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^T \Sigma_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2) + \frac{d}{2}\ln 2\pi + \frac{1}{2}\ln|\Sigma_2| - \ln p(y=-1) =$$

$$-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T \Sigma_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) - \frac{1}{2}\ln|\Sigma_1| + \ln p(y=1) +$$

$$\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^T \Sigma_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2) + \frac{1}{2}\ln|\Sigma_2| - \ln p(y=-1) =$$

$$-\frac{1}{2}\boldsymbol{x}^T \underbrace{\left(\Sigma_1^{-1} - \Sigma_2^{-1}\right)}_{A} \boldsymbol{x} + \boldsymbol{x}^T \underbrace{\left(\Sigma_1^{-1}\boldsymbol{\mu}_1 - \Sigma_2^{-1}\boldsymbol{\mu}_2\right)}_{w} -$$

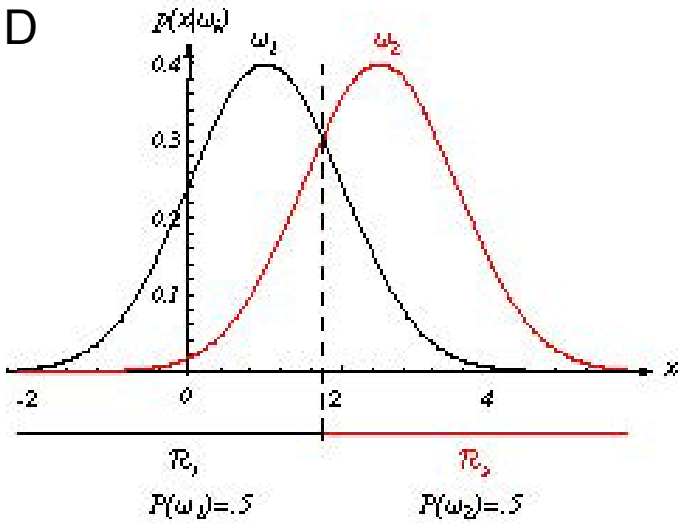
$$\frac{1}{2}\boldsymbol{\mu}_1^T\Sigma_1^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T\Sigma_2^{-1}\boldsymbol{\mu}_2 - \frac{1}{2}\ln|\Sigma_1| + \frac{1}{2}\ln|\Sigma_2| +$$

$$\ln p(y=1) - \ln p(y=-1) =$$

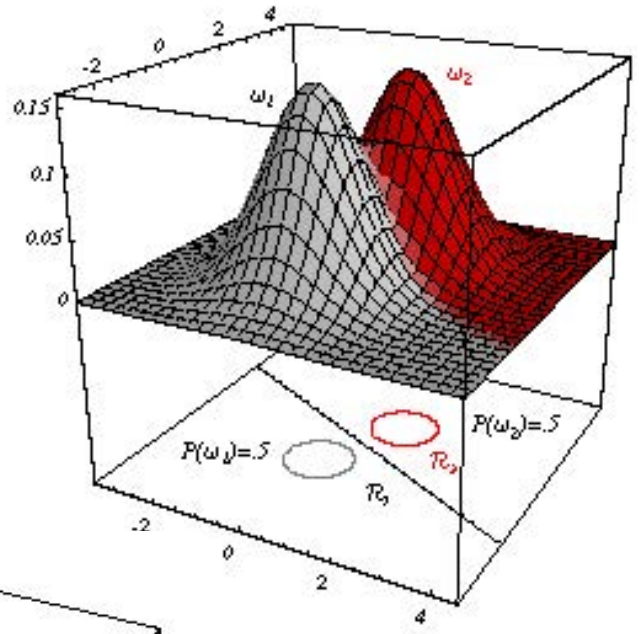
$$\boxed{-\frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} + w^T\boldsymbol{x} + b}$$

Minimal Risk for a Gaussian Classification Task

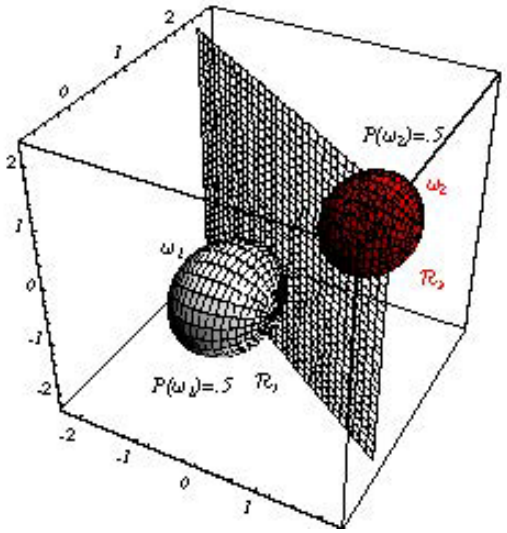
1D



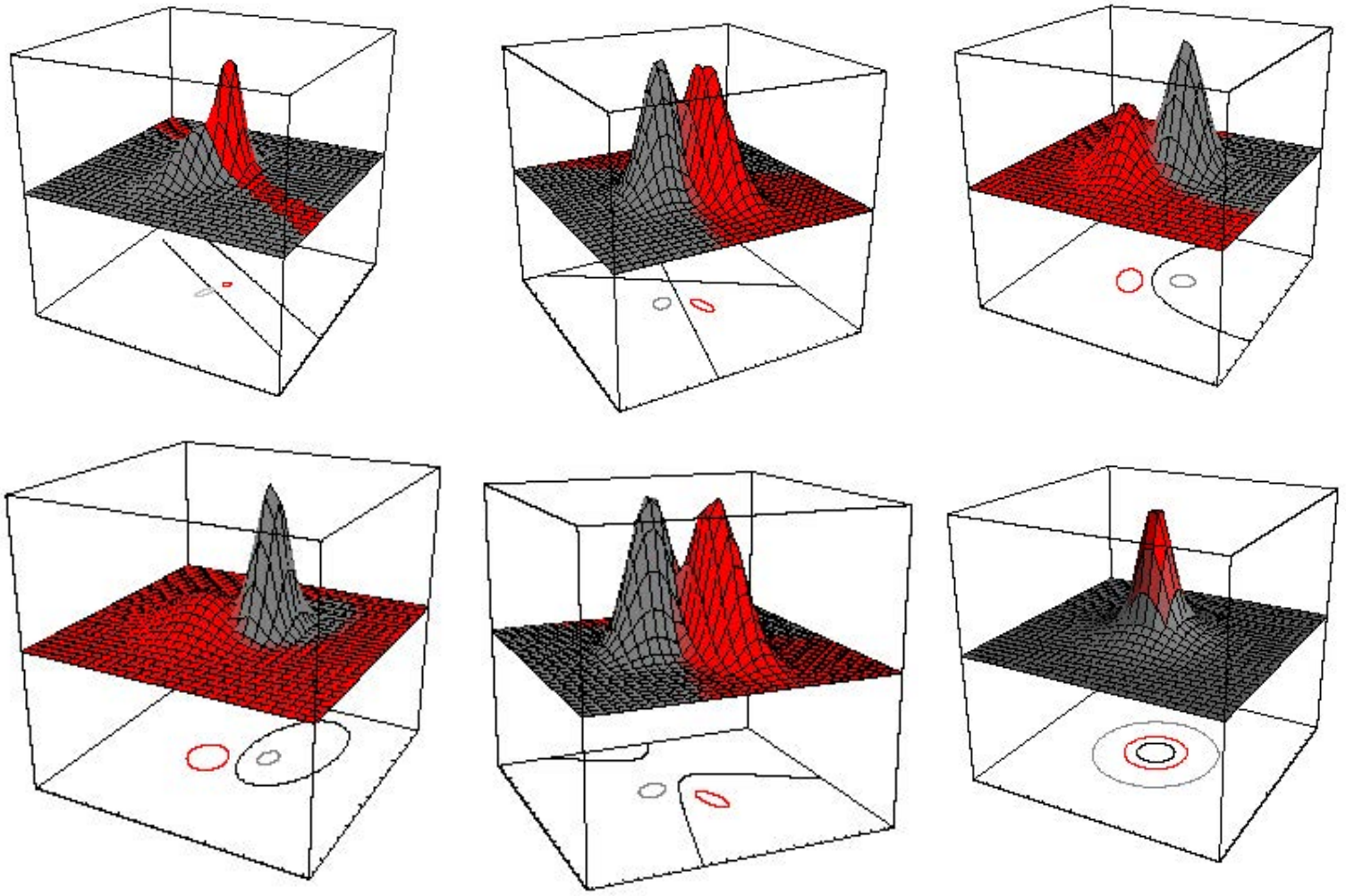
2D



3D



Minimal Risk for a Gaussian Classification Task



Chapter 3

Maximum Likelihood

Maximum Likelihood

- One of the major objectives if learning generative models
- It has certain theoretical properties
- Theoretical concepts like efficient estimator or biased estimator are introduced
- Even supervised methods can be viewed as special case of maximum likelihood

Loss for Unsupervised Learning

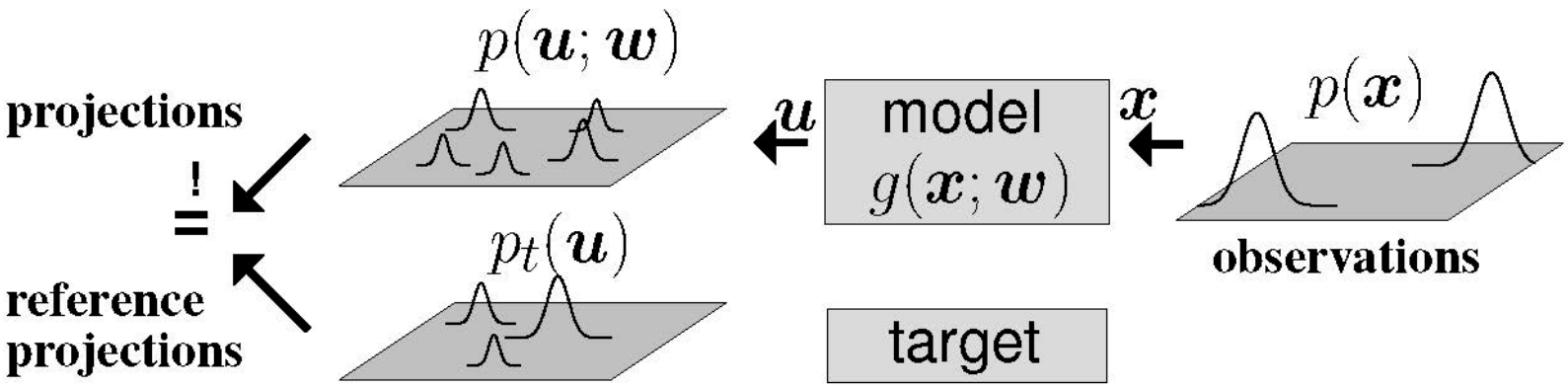
First we consider different loss functions which are used for unsupervised learning

Generative approaches → maximum likelihood

Projection methods → desired property

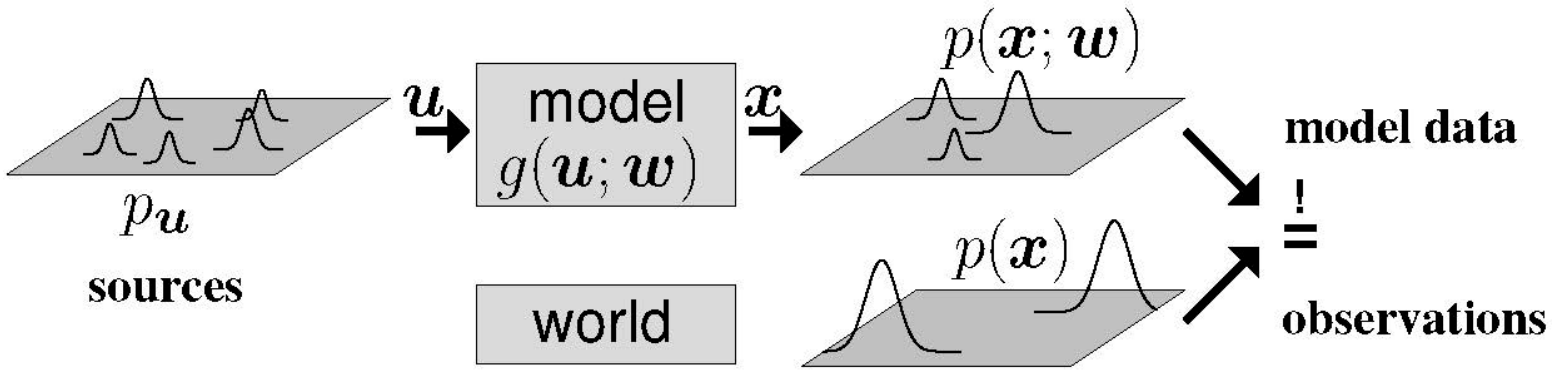
Loss for Unsupervised Learning

- data projection into another space with desired requirements



Loss for Unsupervised Learning

"generative model": model simulates the world and produces the same data as the world



Loss for Unsupervised Learning

- parameterized model known
- task: **estimate actual parameters**
- loss: difference between true and estimated parameter
- evaluate estimator: expected loss

Mean Squared Error, Bias, and Variance

Theoretical concepts of parameter estimation

- training data: $\{\boldsymbol{x}\} = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^l\}$

simply $\boldsymbol{X} = \left(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^l\right)^T$ (the matrix of training data)

- true parameter vector: $\boldsymbol{w}$
- estimate of $\boldsymbol{w}$: $\hat{\boldsymbol{w}}$

Mean Squared Error, Bias, and Variance

- **unbiased** estimator: $\mathrm{E}_{\boldsymbol{X}} \hat{\boldsymbol{w}} = \boldsymbol{w}$

  on average (over training set) the true parameter is obtained

- **bias**: $b(\hat{\boldsymbol{w}}) = \mathrm{E}_{\boldsymbol{X}} \hat{\boldsymbol{w}} - \boldsymbol{w}$

- **variance**: $\mathrm{var}(\hat{\boldsymbol{w}}) = \mathrm{E}_{\boldsymbol{X}} \left((\hat{\boldsymbol{w}} - \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}))^T (\hat{\boldsymbol{w}} - \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}})) \right)$

- **mean squared error** (MSE, different to supervised loss):

$$\mathrm{mse}(\hat{\boldsymbol{w}}) = \mathrm{E}_{\boldsymbol{X}} \left((\hat{\boldsymbol{w}} - \boldsymbol{w})^T (\hat{\boldsymbol{w}} - \boldsymbol{w}) \right)$$

expected squared error between the estimated and true parameter

Mean Squared Error, Bias, and Variance

$$\boxed{\text{mse}(\hat{\boldsymbol{w}})} = \mathrm{E}_{\boldsymbol{X}}\left((\hat{\boldsymbol{w}} - \boldsymbol{w})^T(\hat{\boldsymbol{w}} - \boldsymbol{w})\right) =$$

$$\mathrm{E}_{\boldsymbol{X}}\left(((\hat{\boldsymbol{w}} - \underbrace{\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}})}) + \underbrace{(\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}})}_{\text{zero}} - \boldsymbol{w}))^T\right.$$

$$\left.((\hat{\boldsymbol{w}} - \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}})) + (\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}) - \boldsymbol{w}))\right) =$$

$$\mathrm{E}_{\boldsymbol{X}}\left((\hat{\boldsymbol{w}} - \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}))^T(\hat{\boldsymbol{w}} - \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}})) -\right.$$

$$2\ (\hat{\boldsymbol{w}} - \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}))^T(\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}) - \boldsymbol{w}) +$$

$$\left.(\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}) - \boldsymbol{w})^T(\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}) - \boldsymbol{w})\right) =$$

Only $\hat{\boldsymbol{w}}$ depends on $\boldsymbol{X}$

$$\mathrm{E}_{\boldsymbol{X}}\left((\hat{\boldsymbol{w}} - \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}))^T(\hat{\boldsymbol{w}} - \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}))\right) +$$

$$(\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}) - \boldsymbol{w})^T(\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}) - \boldsymbol{w}) =$$

$$\boxed{\text{var}(\hat{\boldsymbol{w}}) + b^2(\hat{\boldsymbol{w}})}$$

$$\mathrm{E}_{\boldsymbol{X}}\left((\hat{\boldsymbol{w}} - \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}))^T(\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}) - \boldsymbol{w})\right) =$$

$$(\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}) - \mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}))^T(\mathrm{E}_{\boldsymbol{X}}(\hat{\boldsymbol{w}}) - \boldsymbol{w}) = 0$$

Mean Squared Error, Bias, and Variance

Averaging reduces variance – each of the N subsets has l/N examples which gives l examples in total

Average is $\hat{\boldsymbol{w}}_{a_N} = \frac{1}{N}\sum_{p=1}^{N}\hat{\boldsymbol{w}}_i$ where

$$\hat{\boldsymbol{w}}_i = \hat{\boldsymbol{w}}_i\left(\boldsymbol{X}_i\right) \quad \boldsymbol{X}_i = \left\{\boldsymbol{x}^{(i-1)l/N+1}, \ldots, \boldsymbol{x}^{il/N}\right\}$$

Unbiased: $\mathrm{E}_{\boldsymbol{X}}\left(\hat{\boldsymbol{w}}_{a_N}\right) = \frac{1}{N}\sum_{p=1}^{N}\mathrm{E}_{\boldsymbol{X}_i}\hat{\boldsymbol{w}}_i = \frac{1}{N}\sum_{p=1}^{N}\boldsymbol{w} = \boldsymbol{w}$

Variance: $\mathrm{covar}_{\boldsymbol{X}}\left(\hat{\boldsymbol{w}}_{a_N}\right) = \frac{1}{N^2}\sum_{p=1}^{N}\mathrm{covar}_{\boldsymbol{X}_i}\left(\hat{\boldsymbol{w}}_i\right) =$

$$\frac{1}{N^2}\sum_{p=1}^{N}\mathrm{covar}_{\boldsymbol{X},l/N}\left(\hat{\boldsymbol{w}}\right) = \frac{1}{N}\mathrm{covar}_{\boldsymbol{X},l/N}\left(\hat{\boldsymbol{w}}\right)$$

Mean Squared Error, Bias, and Variance

- averaging: training sets $\boldsymbol{X}_i$ are independent, therefore covariance between them vanishes
- **Minimal Variance Unbiased** (MVU) estimator: construct from all **unbiased** estimators the one with minimal variance
- MVU estimator does not always exist
- methods to check whether a given estimator is a MVU

Fisher Information Matrix, Cramer-Rao Lower Bound, and Efficiency

- We will find a lower bound for the variance of an unbiased estimator: Cramer-Rao Lower Bound (that is a lower bound for the MSE)

- We need the Fisher information matrix $\boldsymbol{I}_F$:

$$\boldsymbol{I}_F(\boldsymbol{w}): \quad [\boldsymbol{I}_F(\boldsymbol{w})]_{ij} = \mathrm{E}_{p(\boldsymbol{x};\boldsymbol{w})}\left(\frac{\partial \ln p(\boldsymbol{x};\boldsymbol{w})}{\partial w_i} \; \frac{\partial \ln p(\boldsymbol{x};\boldsymbol{w})}{\partial w_j}\right)$$

$$\mathrm{E}_{p(\boldsymbol{x};\boldsymbol{w})}\left(\frac{\partial \ln p(\boldsymbol{x};\boldsymbol{w})}{\partial w_i} \; \frac{\partial \ln p(\boldsymbol{x};\boldsymbol{w})}{\partial w_j}\right) =$$

$$\int \frac{\partial \ln p(\boldsymbol{x};\boldsymbol{w})}{\partial w_i} \; \frac{\partial \ln p(\boldsymbol{x};\boldsymbol{w})}{\partial w_j} \; p(\boldsymbol{x};\boldsymbol{w}) \; d\boldsymbol{x}$$

Fisher Information Matrix, Cramer-Rao Lower Bound, and Efficiency

If $p(\boldsymbol{x}; \boldsymbol{w})$ satisfies $\forall_{\boldsymbol{w}}: \quad \mathrm{E}_{p(\boldsymbol{x};\boldsymbol{w})} \left(\frac{\partial \ln p(\boldsymbol{x}; \boldsymbol{w})}{\partial \boldsymbol{w}} \right) = \mathbf{0}$

then the Fisher information matrix is

$$\boldsymbol{I}_F(\boldsymbol{w}): \quad \boldsymbol{I}_F(\boldsymbol{w}) = -\mathrm{E}_{p(\boldsymbol{x};\boldsymbol{w})} \left(\frac{\partial^2 \ln p(\boldsymbol{x}; \boldsymbol{w})}{\partial \boldsymbol{w} \, \partial \boldsymbol{w}} \right)$$

Fisher information: information of observation $\boldsymbol{x}$ about parameter $\boldsymbol{w}$ upon which the parameterized density function $p(\boldsymbol{x}; \boldsymbol{w})$ of $\boldsymbol{x}$ depends

Fisher Information Matrix, Cramer-Rao Lower Bound, and Efficiency

Theorem 1 (Cramer-Rao Lower Bound (CRLB))
Assume that

$$\forall_{\boldsymbol{w}}: \quad \mathrm{E}_{p(\boldsymbol{x};\boldsymbol{w})}\left(\frac{\partial \ln p(\boldsymbol{x};\boldsymbol{w})}{\partial \boldsymbol{w}}\right) = \mathbf{0}$$

and that the estimator $\hat{\boldsymbol{w}}$ *is unbiased.*
Then,

$$\mathrm{covar}(\hat{\boldsymbol{w}}) - \boldsymbol{I}_F^{-1}(\boldsymbol{w})$$

is positive definite:

$$\mathrm{covar}(\hat{\boldsymbol{w}}) - \boldsymbol{I}_F^{-1}(\boldsymbol{w}) \geq \mathbf{0}\ .$$

An unbiased estimator attains the bound in that $\mathrm{covar}(\hat{\boldsymbol{w}}) = \boldsymbol{I}_F^{-1}(\boldsymbol{w})$ *if and only if*

$$\frac{\partial \ln p(\boldsymbol{x};\boldsymbol{w})}{\partial \boldsymbol{w}} = \boldsymbol{A}(\boldsymbol{w})\ (\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{w})$$

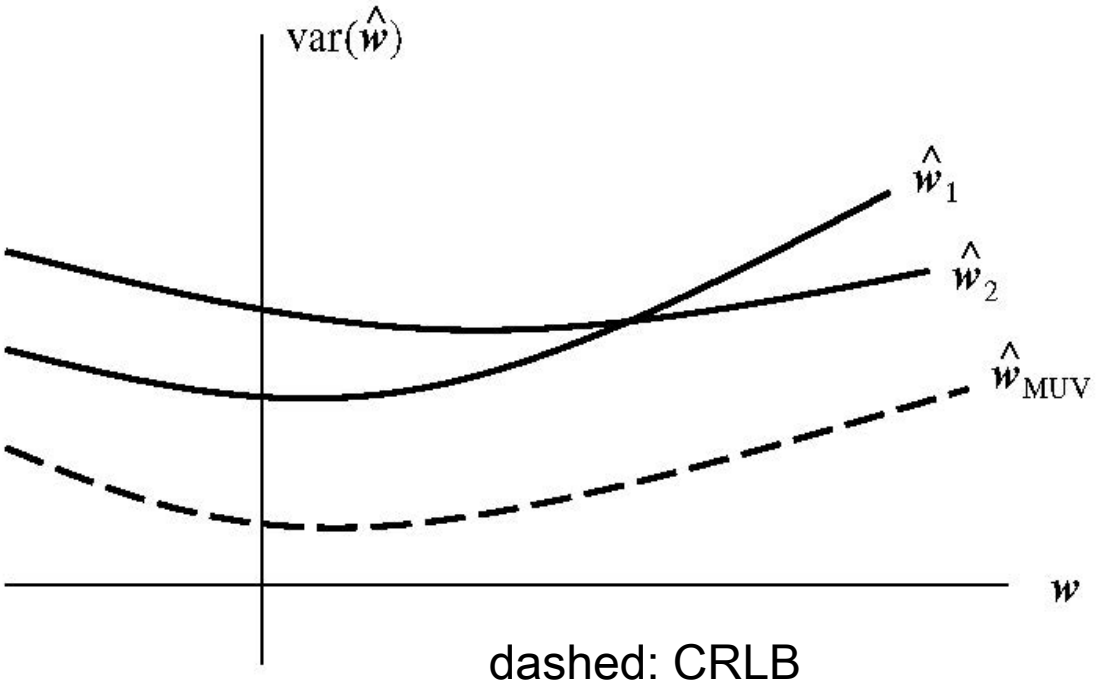
for some function $\boldsymbol{g}$ *and square matrix* $\boldsymbol{A}(\boldsymbol{w})$.
In this case the MVU estimator is

$$\hat{\boldsymbol{w}} = \boldsymbol{g}(\boldsymbol{x}) \quad \textit{with} \quad \mathrm{covar}(\hat{\boldsymbol{w}}) = \boldsymbol{A}^{-1}(\boldsymbol{w})\ .$$

Fisher Information Matrix, Cramer-Rao Lower Bound, and Efficiency

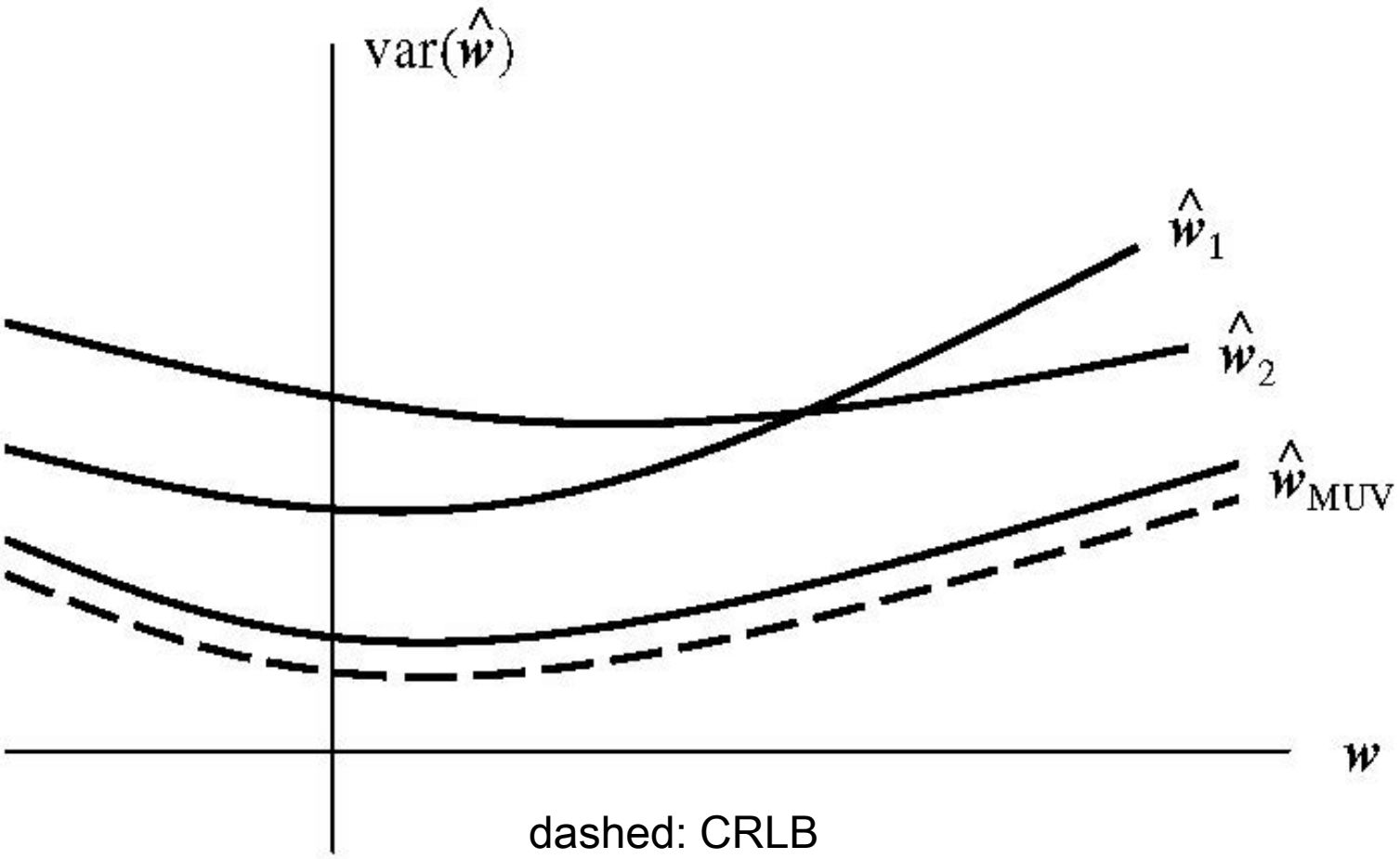
$$\mathrm{var}(\hat{w}_i) \;=\; [\mathrm{covar}(\hat{\boldsymbol{w}})]_{ii} \;\geq\; \left[\boldsymbol{I}_F^{-1}(\boldsymbol{w})\right]_{ii}$$

- **efficient** estimator: reaches the CRLB (efficiently uses the data)
- MVU estimator can be efficient but need not



dashed: CRLB

Fisher Information Matrix, Cramer-Rao Lower Bound, and Efficiency



Maximum Likelihood Estimator

- MVU estimator is unknown or does not exist
- Maximum Likelihood Estimator (MLE)
- MLE can be applied to a broad range of problems
- MLE approximates the MVU estimator for large data sets
- MLE is even asymptotically efficient and unbiased
- MLE does everything right and this efficiently (enough data)

Maximum Likelihood Estimator

The likelihood $\mathcal{L}$ of the data set $\{\boldsymbol{x}\} = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^l\}$:

$$\mathcal{L}(\{\boldsymbol{x}\}; \boldsymbol{w}) = p(\{\boldsymbol{x}\}; \boldsymbol{w})$$

probability of the model $p(\boldsymbol{x}; \boldsymbol{w})$ to produce the data

iid (independent identical distributed) data:

$$\mathcal{L}(\{\boldsymbol{x}\}; \boldsymbol{w}) = p(\{\boldsymbol{x}\}; \boldsymbol{w}) = \prod_{i=1}^{l} p(\boldsymbol{x}^i; \boldsymbol{w})$$

Negative log-likelihood:

$$- \ln \mathcal{L}(\{\boldsymbol{x}\}; \boldsymbol{w}) = - \sum_{i=1}^{l} \ln p(\boldsymbol{x}^i; \boldsymbol{w})$$

Maximum Likelihood Estimator

- likelihood is based on finite many densities values which have zero measure: problem?

- assume instead of $p(\boldsymbol{x}; \boldsymbol{w})$ the volume element $p(\boldsymbol{x}^i; \boldsymbol{w})\ d\boldsymbol{x}$ (region around $\boldsymbol{x}^i$)

- MLE popular:
  - simple use
  - properties

Properties of Maximum Likelihood Estimator

MLE:

- invariant under parameter change
- asymptotically unbiased and efficient → asymptotically optimal
- consistent for zero CRLB

Properties of Maximum Likelihood Estimator

Theorem 1 (Parameter Change Invariance)
Let g be a function changing the parameter $\boldsymbol{w}$ into parameter $\boldsymbol{u}$: $\boldsymbol{u} = g(\boldsymbol{w})$, then

$$\hat{\boldsymbol{u}} = g(\hat{\boldsymbol{w}}) \ ,$$

where the estimators are MLE.
If g changes $\boldsymbol{w}$ into different $\boldsymbol{u}$
then $\hat{\boldsymbol{u}} = g(\hat{\boldsymbol{w}})$ maximizes the likelihood function

$$\max_{\boldsymbol{w}:\boldsymbol{u}=g(\boldsymbol{w})} p(\{\boldsymbol{x}\}; \boldsymbol{w}) \ .$$

Properties of Maximum Likelihood Estimator

The maximum likelihood estimator is asymptotically unbiased:

$$\mathrm{E}_{p(\boldsymbol{x};\boldsymbol{w})}\left(\hat{\boldsymbol{w}}\right) \overset{l\to\infty}{\to} \boldsymbol{w}$$

The maximum likelihood estimator is asymptotically efficient:

$$\mathrm{covar}(\hat{\boldsymbol{w}}) \overset{l\to\infty}{\to} \mathrm{CRLB}$$

Properties of Maximum Likelihood Estimator

Theorem 1 (MLE Asymptotic Properties)
If $p(\boldsymbol{x}; \boldsymbol{w})$ *satisfies*

$$\forall_{\boldsymbol{w}}: \quad \mathrm{E}_{p(\boldsymbol{x};\boldsymbol{w})}\left(\frac{\partial \ln p(\boldsymbol{x};\boldsymbol{w})}{\partial \boldsymbol{w}}\right) = \mathbf{0}$$

then the MLE which maximizes $p(\{\boldsymbol{x}\}; \boldsymbol{w})$
is asymptotically distributed according to

$$\hat{\boldsymbol{w}} \overset{l\to\infty}{\propto} \mathcal{N}\left(\boldsymbol{w}, \boldsymbol{I}_F^{-1}(\boldsymbol{w})\right) ,$$

where $\boldsymbol{I}_F(\boldsymbol{w})$ *is the Fisher information matrix evaluated at the unknown parameter* $\boldsymbol{w}$.

Properties of Maximum Likelihood Estimator

- practical applications: finite examples →MLE performance unknown

- Example: general linear model $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{w} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \propto \mathcal{N}(\boldsymbol{0}, \boldsymbol{C})$

  MLE is $\hat{\boldsymbol{w}} = \left(\boldsymbol{A}^T \boldsymbol{C}^{-1} \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \boldsymbol{C}^{-1} \boldsymbol{x}$ which is efficient and MUV

$$\hat{\boldsymbol{w}} \propto \mathcal{N}\left(\boldsymbol{w}\ ,\ \left(\boldsymbol{A}^T \boldsymbol{C}^{-1} \boldsymbol{A}\right)^{-1}\right)\ .$$

Note the noise covariance must be known

Properties of Maximum Likelihood Estimator

- consistent: $\hat{\boldsymbol{w}} \overset{l \to \infty}{\to} \boldsymbol{w}$

  for large training sets the estimator approaches the true value (difference to unbiased → variance decreases)

- more formal definition for consistency as

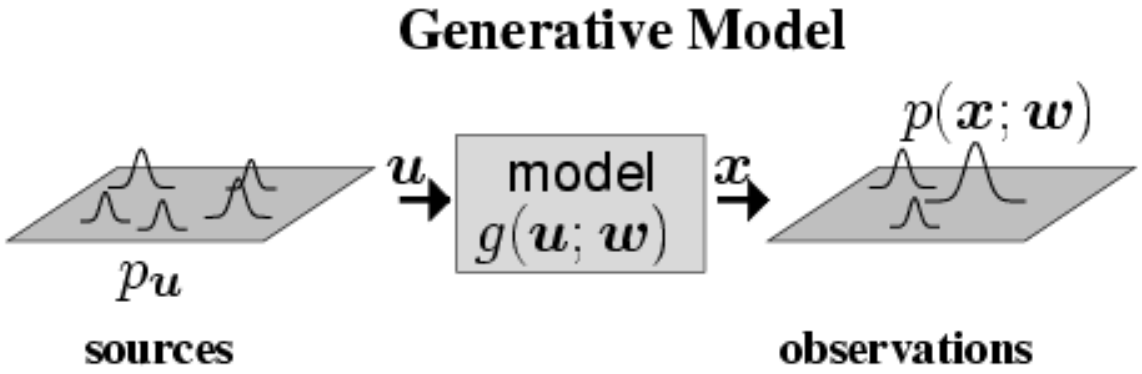
$$\lim_{l \to \infty} p\left(|\hat{\boldsymbol{w}} - \boldsymbol{w}| > \epsilon\right) = 0$$

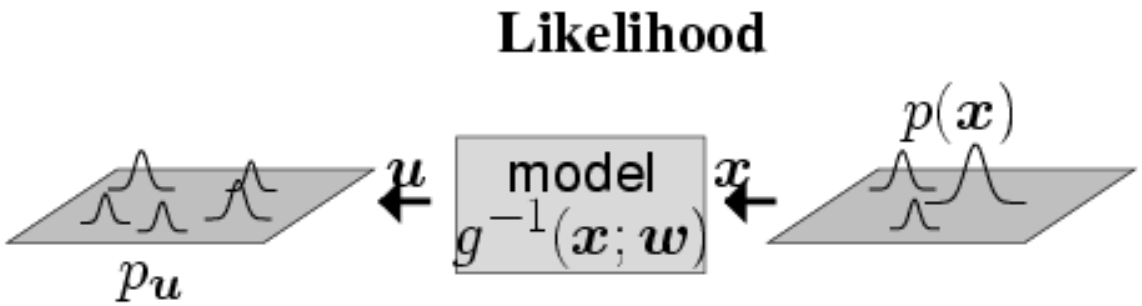
Thus, the MLE is consistent if the CRLB is zero

Expectation Maximization

likelihood can be optimized by gradient descent methods

likelihood cannot be computed analytically:
- hidden states
- many-to-one output mapping
- non-linearities





$$p(\boldsymbol{x};\boldsymbol{w}) = \int_U p_{\boldsymbol{u}}(\boldsymbol{u})\, \delta(\boldsymbol{x} = g(\boldsymbol{u};\boldsymbol{w}))\, d\boldsymbol{u}$$

Expectation Maximization

- hidden variables, latent variables, unobserved variables $\boldsymbol{u}$
- likelihood is determined by all $\boldsymbol{u}$ mapped to $\boldsymbol{x}$

$$p(\boldsymbol{x}; \boldsymbol{w}) = \int_U p_{\boldsymbol{u}}(\boldsymbol{u}) \; \delta(\boldsymbol{x} = g(\boldsymbol{u}; \boldsymbol{w})) \; d\boldsymbol{u}$$

Expectation Maximization

Expectation Maximization (EM) algorithm:

- joint probability $p(\boldsymbol{x}, \boldsymbol{u}; \boldsymbol{w})$ is easier to compute than likelihood

- estimate $p(\boldsymbol{u} \mid \boldsymbol{x}; \boldsymbol{w})$ by $Q(\boldsymbol{u} \mid \boldsymbol{x})$

$$\boxed{\ln \mathcal{L}(\{\boldsymbol{x}\}; \boldsymbol{w})} = \ln p(\{\boldsymbol{x}\}; \boldsymbol{w}) = \ln \int_U p(\{\boldsymbol{x}\}, \boldsymbol{u}; \boldsymbol{w}) \, d\boldsymbol{u} =$$

$$\ln \int_U \frac{Q(\boldsymbol{u} \mid \{\boldsymbol{x}\})}{Q(\boldsymbol{u} \mid \{\boldsymbol{x}\})} p(\{\boldsymbol{x}\}, \boldsymbol{u}; \boldsymbol{w}) \, d\boldsymbol{u} \ \boxed{\geq}$$ Jensen's inequality

$$\int_U Q(\boldsymbol{u} \mid \{\boldsymbol{x}\}) \ln \frac{p(\{\boldsymbol{x}\}, \boldsymbol{u}; \boldsymbol{w})}{Q(\boldsymbol{u} \mid \{\boldsymbol{x}\})} \, d\boldsymbol{u} =$$

$$\int_U Q(\boldsymbol{u} \mid \{\boldsymbol{x}\}) \ln p(\{\boldsymbol{x}\}, \boldsymbol{u}; \boldsymbol{w}) \, d\boldsymbol{u} -$$

$$\int_U Q(\boldsymbol{u} \mid \{\boldsymbol{x}\}) \ln Q(\boldsymbol{u} \mid \{\boldsymbol{x}\}) \, d\boldsymbol{u} =$$

$$\boxed{\mathcal{F}(Q, \boldsymbol{w})}$$

Expectation Maximization

EM algorithm is an iteration between ”E”-step and “M”-step:

E-step:

$$Q_{k+1} = \arg\max_{Q} \mathcal{F}(Q, \boldsymbol{w}_k)$$

M-step:

$$\boldsymbol{w}_{k+1} = \arg\max_{\boldsymbol{w}} \mathcal{F}(Q_{k+1}, \boldsymbol{w})$$

Expectation Maximization

After E-step:

$$\begin{aligned} Q_{k+1}(\boldsymbol{u} \mid \{\boldsymbol{x}\}) &= p(\boldsymbol{u} \mid \{\boldsymbol{x}\}; \boldsymbol{w}_k) \\ \mathcal{F}(Q_{k+1}, \boldsymbol{w}_k) &= \ln \mathcal{L}(\{\boldsymbol{x}\}; \boldsymbol{w}_k) \end{aligned}$$

Proof: $p(\boldsymbol{u}, \{\boldsymbol{x}\}; \boldsymbol{w}_k) = p(\boldsymbol{u} \mid \{\boldsymbol{x}\}; \boldsymbol{w}_k)\, p(\{\boldsymbol{x}\}; \boldsymbol{w}_k)$

$$\mathcal{F}(Q, \boldsymbol{w}) = \int_U Q(\boldsymbol{u} \mid \{\boldsymbol{x}\}) \ln \frac{p(\{\boldsymbol{x}\}, \boldsymbol{u}; \boldsymbol{w})}{Q(\boldsymbol{u} \mid \{\boldsymbol{x}\})}\, d\boldsymbol{u} =$$

$$\int_U Q(\boldsymbol{u} \mid \{\boldsymbol{x}\}) \ln \frac{p(\boldsymbol{u} \mid \{\boldsymbol{x}\}; \boldsymbol{w})}{Q(\boldsymbol{u} \mid \{\boldsymbol{x}\})}\, d\boldsymbol{u} + \ln p(\{\boldsymbol{x}\}; \boldsymbol{w}) =$$

$$\int_U Q(\boldsymbol{u} \mid \{\boldsymbol{x}\}) \ln \frac{p(\boldsymbol{u} \mid \{\boldsymbol{x}\}; \boldsymbol{w})}{Q(\boldsymbol{u} \mid \{\boldsymbol{x}\})}\, d\boldsymbol{u} + \ln \mathcal{L}(\{\boldsymbol{x}\}; \boldsymbol{w})$$

Kullback-Leibler divergence:

$$D_{\mathrm{KL}}(Q \parallel p) = \int_U Q(\boldsymbol{u} \mid \{\boldsymbol{x}\}) \ln \frac{Q(\boldsymbol{u} \mid \{\boldsymbol{x}\})}{p(\boldsymbol{u} \mid \{\boldsymbol{x}\}; \boldsymbol{w})}\, d\boldsymbol{u} \geq 0$$

Zero for: $Q(\boldsymbol{u} \mid \{\boldsymbol{x}\}) = p(\boldsymbol{u} \mid \{\boldsymbol{x}\}; \boldsymbol{w})$

Expectation Maximization

EM increases the lower bound in both steps

Beginning of the M-step: $\mathcal{F}(Q_{k+1}, \boldsymbol{w}_k) = \ln \mathcal{L}(\{\boldsymbol{x}\}; \boldsymbol{w}_k)$

E-step does not change the parameters

$$\ln \mathcal{L}(\{\boldsymbol{x}\}; \boldsymbol{w}_k) = \mathcal{F}(Q_{k+1}, \boldsymbol{w}_k) \leq$$
$$\mathcal{F}(Q_{k+1}, \boldsymbol{w}_{k+1}) \leq \mathcal{F}(Q_{k+2}, \boldsymbol{w}_{k+1}) = \ln \mathcal{L}(\{\boldsymbol{x}\}; \boldsymbol{w}_{k+1})$$

EM algorithm:

- hidden Markov models
- mixture of Gaussians
- factor analysis
- independent component analysis

Maximum Entropy

maximum entropy probability distribution:

- maximal entropy given a class of distributions
- minimal prior assumptions
- physical systems converge to maximal entropy configurations
- most likely observed solution
- connection: statistical mechanics and information theory

principle of maximum entropy first expounded by E.T. Jaynes in 1957

Maximum Entropy

Entropy

$$H = -\sum_{k \geq 1} p_k \log p_k \qquad p_k \log p_k = 0 \text{ for } p_k = 0$$

$$H = -\int_{-\infty}^{\infty} p(x) \log p(x) \; dx \qquad p(x) \log p(x) = 0 \text{ for } p(x) = 0$$

Examples:

- normal distribution: given mean and standard deviation
- uniform distribution: supported in the interval $[a, b]$
- exponential distribution: given mean in $[0, \infty]$

Maximum Entropy

Theorem 1 (Boltzmann's Theorem: Discrete)
Suppose $S = \{x_1, x_2, \ldots\}$ is a (finite or infinite) discrete subset of the reals and n functions $f_1, \ldots, f_n$ and n numbers $a_1, \ldots, a_n$ are given. Let be C the class of all discrete random variables X which are supported on S and which satisfy the n conditions:

$$\mathrm{E}(f_j(X)) \;=\; a_j \qquad \text{for } j = 1, \ldots, n$$

If there exists a member of C which assigns positive probability to all members of S and if there exists a maximum entropy distribution for C, then this distribution has the following shape:

$$\Pr(X = x_k) \;=\; c \;\exp\left(\sum_{j=1}^{n} \lambda_j \; f_j(x_k)\right) \qquad \text{for } k = 1, 2, \ldots \;,$$

where the constants c and λ_j have to be determined so that the sum of the probabilities is 1 and the above conditions for the expected values are satisfied.

Conversely, if constants c and λ_j as above can be found, then the above distribution is indeed the maximum entropy distribution for class C.

Maximum Entropy

Theorem 1 (Boltzmann's Theorem: Continuous)
Suppose S is a closed subset of the real numbers $\mathbb{R}$ and n measurable functions $f_1, \ldots, f_n$ and n numbers $a_1, \ldots, a_n$ are given. Let be C the class of all continuous random variables which are supported on S and which satisfy the n expected value conditions:

$$\mathrm{E}(f_j(X)) \;=\; a_j \quad \text{for } j = 1, \ldots, n$$

If there is a member in C whose density function is positive everywhere in S, and if there exists a maximal entropy distribution for C, then its probability density $p(x)$ has the following shape:

$$p(x) \;=\; c \;\exp\left(\sum_{j=1}^{n} \lambda_j \; f_j(x)\right) \quad \text{for all } x \in S \;,$$

where the constants c and λ_j have to be determined so that the integral of $p(x)$ over S is 1 and the above conditions for the expected values are satisfied.

Conversely, if constants c and λ_j like this can be found, then $p(x)$ is indeed the density of the (unique) maximum entropy distribution for class C.

Maximum Entropy

Not all classes of distributions contain a maximum entropy distribution:

- arbitrarily large entropy: distributions with mean
- entropies of a class are bounded from above but not attained: distributions with mean zero, second moment one, and third moment one

Maximum Entropy: Discrete Solution

Constraints: $\sum_{i=1}^{n} p(x_i)\ f_k(x_i) \ = \ F_k \qquad k = 1, \ldots, m$

$$\sum_{i=1}^{n} p(x_i) \ = \ 1$$

Solution, the **Gibbs distribution**

$$p(x_i) \ = \ \frac{1}{Z(\lambda_1, \ldots, \lambda_m)} \ \exp\left(\lambda_1 f_1(x_i) \ + \ \cdots \ + \ \lambda_m f_m(x_i)\right)$$

with **partition function**

$$Z(\lambda_1, \ldots, \lambda_m) \ = \ \sum_{i=1}^{n} \exp\left(\lambda_1 f_1(x_i) \ + \ \cdots \ + \ \lambda_m f_m(x_i)\right)$$

The **Lagrange multipliers** are determined by the equation system

$$F_k \ = \ \frac{\partial}{\partial \lambda_k} \log Z(\lambda_1, \ldots, \lambda_m)$$

Maximum Entropy: Continuous Solution

Instead of the entropy, we use the Kullback-Leibler divergence:

$$D_{\mathrm{KL}}(p \parallel m) = -\int p(x) \log \frac{p(x)}{m(x)}\, dx$$

$m(x)$ is proportional to the limiting density of discrete points and is assumed to be known

Constraints: $\int p(x) f_k(x)\, dx = F_k \qquad k = 1, \ldots, m \qquad \int p(x)\, dx = 1$

Solution, is **Gibbs distribution**:

$$p(x) = \frac{1}{Z(\lambda_1, \ldots, \lambda_m)}\, m(x)\, \exp\left(\lambda_1 f_1(x) + \cdots + \lambda_m f_m(x)\right)$$

with **partition function**

$$Z(\lambda_1, \ldots, \lambda_m) = \int m(x)\, \exp\left(\lambda_1 f_1(x) + \cdots + \lambda_m f_m(x)\right)\, dx$$

and the equation system for the **Lagrange multipliers**:

$$F_k = \frac{\partial}{\partial \lambda_k} \log Z(\lambda_1, \ldots, \lambda_m)$$

Maximum Entropy

The invariant measure function $m(x)$ is actually the prior density function representing the lack of relevant information.
It can be determined by the principle of transformation groups or marginalization theory.

If x takes values only in (a, b), then the maximum entropy probability density function is $p(x) = Z \cdot m(x)\ , \qquad a < x < b$
where Z is a normalization constant.

Chapter 4

Noise Models

Noise Models

- connecting unsupervised and supervised learning
- quality measure
- noise on the targets
- apply maximum likelihood

Gaussian Noise

- Gaussian target noise
- linear model

$$\boldsymbol{s} = \boldsymbol{X}\ \boldsymbol{w}$$

$$\boldsymbol{y} = \boldsymbol{s} + \boldsymbol{\epsilon} = \boldsymbol{X}\ \boldsymbol{w} + \boldsymbol{\epsilon}$$

$$\boldsymbol{\epsilon} \propto \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$$

$$\mathcal{L}((\boldsymbol{y}, \boldsymbol{X}); \boldsymbol{w}) = \frac{1}{(2\ \pi)^{d/2}\ |\boldsymbol{\Sigma}|^{1/2}}\ \exp\left(-\frac{1}{2}\ (\boldsymbol{y} - \boldsymbol{X}\ \boldsymbol{w})^T\ \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{X}\ \boldsymbol{w})\right)$$

log-likelihood:

$$\ln \mathcal{L}((\boldsymbol{y}, \boldsymbol{X}); \boldsymbol{w}) = -\frac{d}{2} \ln(2\ \pi) - \frac{1}{2} \ln|\boldsymbol{\Sigma}| - \frac{1}{2}\ (\boldsymbol{y} - \boldsymbol{X}\ \boldsymbol{w})^T\ \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{X}\ \boldsymbol{w})$$

Gaussian Noise

- minimize $(\boldsymbol{y} - \boldsymbol{X}\ \boldsymbol{w})^T\ \boldsymbol{\Sigma}^{-1}(\boldsymbol{y} - \boldsymbol{X}\ \boldsymbol{w})$

least square criterion

linear least square estimator

derivative with respect to $\boldsymbol{w}$:

$$-\ 2\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{y}\ +\ 2\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\ \boldsymbol{X}\ \boldsymbol{w}$$

Setting the derivative to zero (Wiener-Hopf equations):

$$\hat{\boldsymbol{w}}\ =\ \left(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\ \boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{y}$$

Gaussian Noise

Noise covariance matrix gives the noise for each measure
In most cases we have the same noise for each observation:

$$\boldsymbol{\Sigma}^{-1} = \frac{1}{\sigma} \boldsymbol{I}$$

We obtain

$$\hat{\boldsymbol{w}} = \left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \boldsymbol{y}$$

$\left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T$: pseudo inverse or Moore-Penrose inverse

minimal value: $\frac{1}{\sigma} \boldsymbol{y}^T \left(\boldsymbol{I} - \boldsymbol{X} \left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T\right) \boldsymbol{y}$

Laplace Noise and Minkowski Error

Laplace noise assumption:

$$\|\boldsymbol{y} - \boldsymbol{g}(\boldsymbol{x}; \boldsymbol{w})\|_1$$

$$p(y - g(\boldsymbol{x}; \boldsymbol{w})) = \frac{\beta}{2} \exp\left(-\beta \, |y - g(\boldsymbol{x}; \boldsymbol{w})|\right)$$

More general **Minkowski error**:

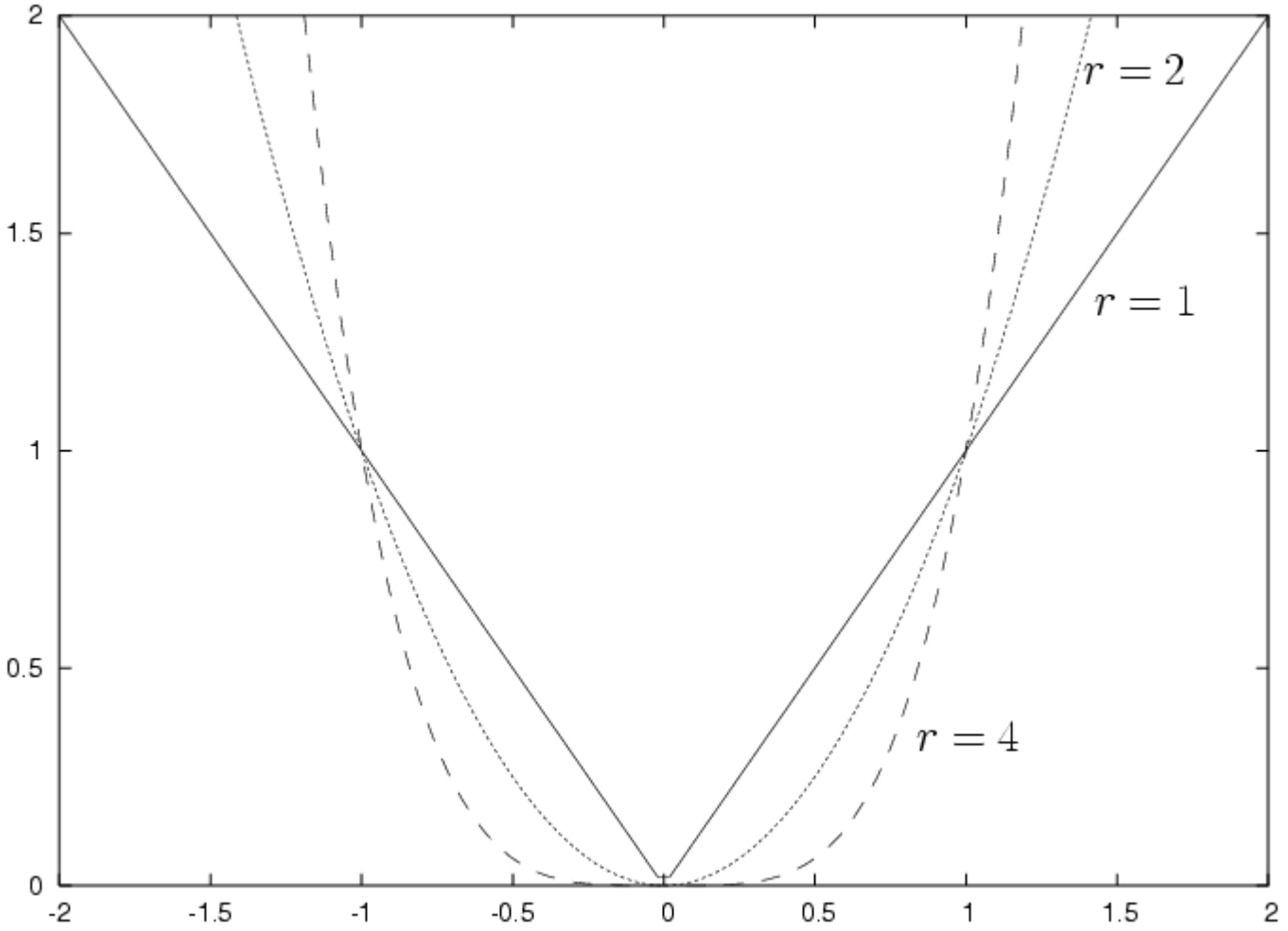
$$|y - g(\boldsymbol{x}; \boldsymbol{w})|^r$$

$$p(y - g(\boldsymbol{x}; \boldsymbol{w})) = \frac{r \, \beta^{1/r}}{2 \, \Gamma(1/r)} \exp\left(-\beta \, |y - g(\boldsymbol{x}; \boldsymbol{w})|^r\right) \, ,$$

gamma function

$$\Gamma(a) = \int_0^\infty u^{a-1} \, e^{-u} \, du \quad , \quad \Gamma(n) = (n-1)!$$

Laplace Noise and Minkowski Error



Binary Models

- noise considerations do not hold for binary target
- classification not treated

Cross-Entropy

classification problem with K classes:

$$g_k(\boldsymbol{x}; \boldsymbol{w}) = p(\boldsymbol{y} = \boldsymbol{e}_k \mid \boldsymbol{x})$$

$$\boldsymbol{y} \in \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_K\}$$

position k

$$\text{If } \boldsymbol{x} \text{ is in the } k\text{th class then } \boldsymbol{y} = (0, \ldots, 0, 1, 0, \ldots, 0)$$

Likelihood:

$$\mathcal{L}(\{\boldsymbol{z}\}; \boldsymbol{w}) = p(\{\boldsymbol{z}\}; \boldsymbol{w}) = \prod_{i=1}^{l} \prod_{k=1}^{K} p(\boldsymbol{y}^i = \boldsymbol{e}_k \mid \boldsymbol{x}^i; \boldsymbol{w})^{[\boldsymbol{y}^i]_k} p(\boldsymbol{x}^i)$$

$$\prod_{k=1}^{K} p(\boldsymbol{y}^i = \boldsymbol{e}_k \mid \boldsymbol{x}^i; \boldsymbol{w})^{[\boldsymbol{y}^i]_k} = p(\boldsymbol{y}^i = \boldsymbol{e}_r \mid \boldsymbol{x}^i; \boldsymbol{w}) \text{ for } \boldsymbol{y}^i = \boldsymbol{e}_r$$

Cross-Entropy

The log-likelihood:

$$\ln \mathcal{L}(\{\boldsymbol{z}\}; \boldsymbol{w}) = \sum_{k=1}^{K} \sum_{i=1}^{l} \left[\boldsymbol{y}^i\right]_k \ln p(\boldsymbol{y}^i = \boldsymbol{e}_k \mid \boldsymbol{x}^i; \boldsymbol{w}) + \sum_{i=1}^{l} \ln p(\boldsymbol{x}^i)$$

loss function:

$$\sum_{k=1}^{K} \sum_{i=1}^{l} \left[\boldsymbol{y}^i\right]_k \ln p(\boldsymbol{y}^i = \boldsymbol{e}_k \mid \boldsymbol{x}^i; \boldsymbol{w})$$

cross entropy (Kullback-Leibler)

$$\left[\boldsymbol{y}^i\right]_k \sim p(\boldsymbol{y}^i = \boldsymbol{e}_k)$$

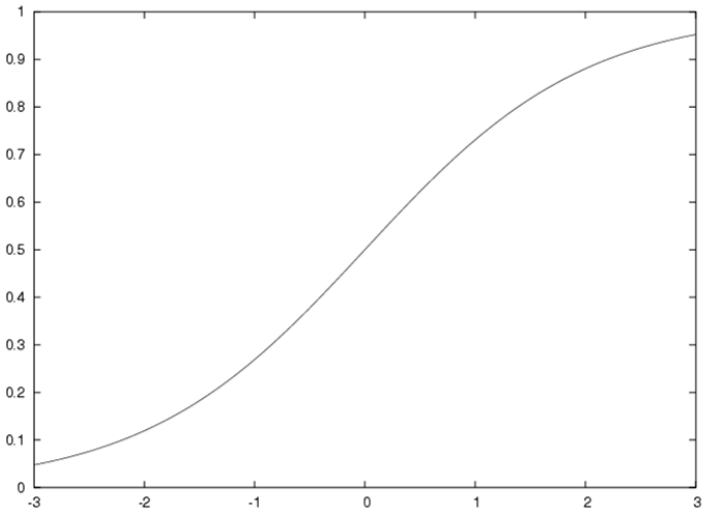
$$\left[\boldsymbol{y}^i\right]_k = \begin{cases} 1 & \text{for} \quad \boldsymbol{y}^i = \boldsymbol{e}_k \\ 0 & \text{for} \quad \boldsymbol{y}^i \neq \boldsymbol{e}_k \end{cases}$$

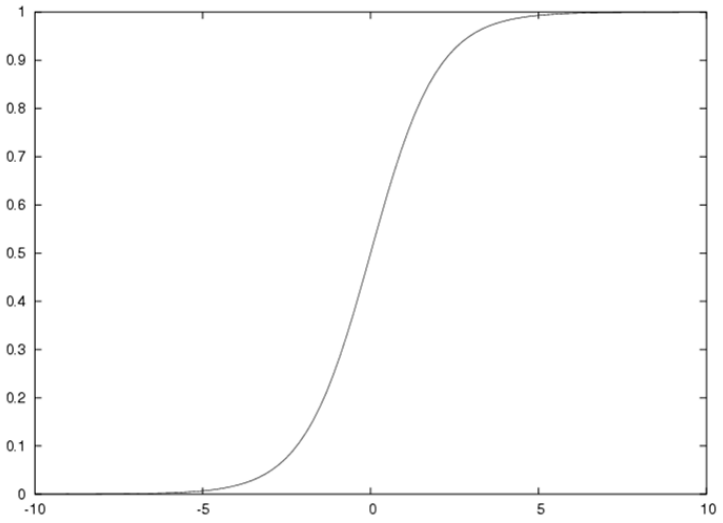
$$\sum_{k=1}^{K} \sum_{i=1}^{l} \left[\boldsymbol{y}^i\right]_k \ln \frac{p(\boldsymbol{y}^i = \boldsymbol{e}_k \mid \boldsymbol{x}^i; \boldsymbol{w})}{\left[\boldsymbol{y}^i\right]_k} - \sum_{k=1}^{K} \sum_{i=1}^{l} \left[\boldsymbol{y}^i\right]_k \ln \left[\boldsymbol{y}^i\right]_k$$

Logistic Regression

a function g mapping $\boldsymbol{x}$ onto R can be transformed into a probability:

$$p(y = 1 \mid \boldsymbol{x}; \boldsymbol{w}) = \frac{1}{1 + e^{-g(\boldsymbol{x};\boldsymbol{w})}}$$





Logistic Regression

If follows:

$$g(\boldsymbol{x}; \boldsymbol{w}) = \ln \left(\frac{p(y = 1 \mid \boldsymbol{x})}{1 - p(y = 1 \mid \boldsymbol{x})} \right) .$$

log-likelihood:

$$\ln \mathcal{L}(\{\boldsymbol{z}\}; \boldsymbol{w}) = \sum_{i=1}^{l} \ln p(\boldsymbol{z}_i; \boldsymbol{w}) = \sum_{i=1}^{l} \ln p(y^i, \boldsymbol{x}^i; \boldsymbol{w}) =$$

$$\sum_{i=1}^{l} \ln p(y^i \mid \boldsymbol{x}^i; \boldsymbol{w}) + \sum_{i=1}^{l} \ln p(\boldsymbol{x}^i)$$

maximum likelihood maximizes

$$\sum_{i=1}^{l} \ln p(y^i \mid \boldsymbol{x}^i; \boldsymbol{w})$$

Logistic Regression

derivative of the log-likelihood:

$$\frac{\partial}{\partial w_j} \sum_{i=1}^{l} \ln p(y^i \mid \boldsymbol{x}^i; \boldsymbol{w}) = \sum_{i=1}^{l} \left(p(y = 1 \mid \boldsymbol{x}^i; \boldsymbol{w}) - y^i\right) \frac{\partial\, g(\boldsymbol{x}^i; \boldsymbol{w})}{\partial w_j}$$

similar to the derivative of the quadratic loss function in the regression:

$$\frac{1}{2} \frac{\partial}{\partial w_j} \sum_{i=1}^{l} \left(g(\boldsymbol{x}^i; \boldsymbol{w}) - y^i\right)^2 = \sum_{i=1}^{l} \left(g(\boldsymbol{x}^i; \boldsymbol{w}) - y^i\right) \frac{\partial\, g(\boldsymbol{x}^i; \boldsymbol{w})}{\partial w_j}$$

$\left(g(\boldsymbol{x}^i; \boldsymbol{w}) - y^i\right)$ instead of $\left(p(y = 1 \mid \boldsymbol{x}^i; \boldsymbol{w}) - y^i\right)$

$$p(y = 1 \mid \boldsymbol{x}; \boldsymbol{w}) = \frac{1}{1 + e^{-g(\boldsymbol{x}; \boldsymbol{w})}}$$

Logistic Regression

linear case: $g(\boldsymbol{x}^i; \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{x}^i$

labels: $y \in \{+1, -1\}$

$$\frac{\partial L}{\partial w_j} = -\sum_{i=1}^{l} y^i \, x_{ij} \, \left(1 - p(y^i \mid \boldsymbol{x}^i; \boldsymbol{w})\right)$$

(We drop i from $\boldsymbol{x}$!)

second order derivatives with H as the Hessian

$$H_{jk} = \frac{\partial L}{\partial w_j \, \partial w_k} = \sum_{i=1}^{l} \left(y^i\right)^2 \, x_{ij} \, x_{ik} \, p(y^i \mid \boldsymbol{x}; \boldsymbol{w}) \, \left(1 - p(y^i \mid \boldsymbol{x}; \boldsymbol{w})\right)$$

$$\rho_{ij} = x_{ij} \, \sqrt{p(y^i \mid \boldsymbol{x}; \boldsymbol{w}) \, \left(1 - p(y^i \mid \boldsymbol{x}; \boldsymbol{w})\right)}$$

bilinear form of the Hessian with a vector:

$$\boldsymbol{a}^T \, \boldsymbol{H} \, \boldsymbol{a} = \sum_{i=1}^{l} \sum_{j=1}^{d} \sum_{k=1}^{d} x_{ij} \, x_{ik} \, a_j \, a_k \, p(y^i \mid \boldsymbol{x}; \boldsymbol{w}) \, \left(1 - p(y^i \mid \boldsymbol{x}; \boldsymbol{w})\right) =$$

$$\sum_{i=1}^{l} \sum_{j=1}^{d} a_j \, x_{ij} \, \sqrt{p(y^i \mid \boldsymbol{x}; \boldsymbol{w}) \, \left(1 - p(y^i \mid \boldsymbol{x}; \boldsymbol{w})\right)} \sum_{k=1}^{d} a_k \, x_{ik} \, \sqrt{p(y^i \mid \boldsymbol{x}; \boldsymbol{w}) \, \left(1 - p(y^i \mid \boldsymbol{x}; \boldsymbol{w})\right)} =$$

$$\sum_{i=1}^{l} \left(\boldsymbol{a}^T \, \boldsymbol{\rho}_i\right) \left(\boldsymbol{a}^T \, \boldsymbol{\rho}_i\right) = \sum_{i=1}^{l} \left(\boldsymbol{a}^T \, \boldsymbol{\rho}_i\right)^2 \geq 0$$

→ Hessian is positive definite

Softmax

Multi-class: logistic regression can be generalized by Softmax

We assume K classes with $y \in \{1, \ldots, K\}$

$$p(y = k \mid \boldsymbol{x}; g_1, \ldots, g_K, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_K) = \frac{e^{\, g_k(\boldsymbol{x}; \boldsymbol{w}_k)}}{\sum_{j=1}^{K} e^{\, g_j(\boldsymbol{x}; \boldsymbol{w}_j)}}$$

→ multinomial distribution across the classes

Objective: $L = -\sum_{i=1}^{l} \ln p(y = y^i \mid \boldsymbol{x}^i; \boldsymbol{w}) = \sum_{i=1}^{l} \ln \left(\sum_{j=1}^{K} e^{g_j(\boldsymbol{x}; \boldsymbol{w}_j)} \right) - g_{y^i}(\boldsymbol{x}; \boldsymbol{w}_{y^i})$

In the following we set $p(y = k \mid \boldsymbol{x}; g_1, \ldots, g_K, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_K) = p(k \mid \boldsymbol{x}; \boldsymbol{W})$

Derivatives:

$$\frac{\partial L}{\partial w_{kn}} = \sum_{i=1}^{l} \frac{\partial \, g_k(\boldsymbol{x}^i; \boldsymbol{w}_k)}{\partial w_{kn}} p(k \mid \boldsymbol{x}^i; \boldsymbol{W}) - \delta_{y^i = k} \sum_{i=1}^{l} \frac{\partial \, g_k(\boldsymbol{x}^i; \boldsymbol{w}_k)}{\partial w_{kn}}$$

Softmax

We show that linear Softmax is strictly convex

linear case: $g_k(\boldsymbol{x}^i; \boldsymbol{w}_k) = \boldsymbol{w}_k^T \boldsymbol{x}^i$ $\boldsymbol{g}(\boldsymbol{x}^i; \boldsymbol{W}) = \boldsymbol{W}^T \boldsymbol{x}^i$

derivatives: $\frac{\partial L}{\partial w_{kn}} = \sum_{i=1}^{l} x_{in} p(k \mid \boldsymbol{x}^i; \boldsymbol{W}) - \delta_{y^i=k} \sum_{i=1}^{l} x_{in}$

second order derivatives:

$$\frac{\partial p(v \mid \boldsymbol{x}^i; \boldsymbol{W})}{\partial w_{vm}} = x_{im}\, p(k \mid \boldsymbol{x}^i; \boldsymbol{W}) \left(1 - p(k \mid \boldsymbol{x}^i; \boldsymbol{W})\right)$$

$$\frac{\partial p(k \mid \boldsymbol{x}^i; \boldsymbol{W})}{\partial w_{vm}} = x_{im}\, p(k \mid \boldsymbol{x}^i; \boldsymbol{W})\, p(v \mid \boldsymbol{x}^i; \boldsymbol{W})$$

$$H_{kn,vm} = \frac{\partial L}{\partial w_{kn} \partial w_{vm}} =$$

$$\sum_{i=1}^{l} x_{in}\, x_{im}\, p(k \mid \boldsymbol{x}^i; \boldsymbol{W}) \left(\delta_{k=v} \left(1 - p(k \mid \boldsymbol{x}^i; \boldsymbol{W})\right) - (1 - \delta_{k=v}) p(v \mid \boldsymbol{x}^i; \boldsymbol{W})\right)$$

Softmax

We consider the bilinear from

$$\boldsymbol{a}^T \boldsymbol{H}\ \boldsymbol{a}\ =$$

$$\sum_{k,n} \sum_{v,m} \sum_{i} a_{kn}\ a_{vm}\ x_{in}\ x_{im}\ p(k \mid \boldsymbol{x}^i; \boldsymbol{W})\ \left(\delta_{k=v}\ (1\ -\ p(k \mid \boldsymbol{x}^i; \boldsymbol{W}))\ -\ (1\ -\ \delta_{k=v})\ p(v \mid \boldsymbol{x}^i; \boldsymbol{W})\right)\ =$$

$$\sum_{k,n} \sum_{i} a_{kn}\ x_{in}\ p(k \mid \boldsymbol{x}^i; \boldsymbol{W})\ \sum_{m} x_{im}\ \left(a_{km}\ -\ \sum_{v} a_{vm} p(v \mid \boldsymbol{x}^i; \boldsymbol{W})\right)\ =$$

$$\sum_{i} \sum_{n} x_{in}\ \sum_{k} a_{kn}\ p(k \mid \boldsymbol{x}^i; \boldsymbol{W})\ \sum_{m} x_{im}\ \left(a_{km}\ -\ \sum_{v} a_{vm} p(v \mid \boldsymbol{x}^i; \boldsymbol{W})\right)\ =$$

$$\sum_{i} -\ \left\{ \left(\sum_{n} x_{in}\ \sum_{k} a_{kn}\ p(k \mid \boldsymbol{x}^i; \boldsymbol{W})\right)\ \left(\sum_{m} x_{im}\ \sum_{v} a_{vm} p(v \mid \boldsymbol{x}^i; \boldsymbol{W})\right) \right\}\ +$$

$$\left\{ \sum_{n} x_{in}\ \sum_{k} a_{kn}\ p(k \mid \boldsymbol{x}^i; \boldsymbol{W})\ \sum_{m} x_{im}\ a_{km} \right\}\ =$$

$$\sum_{i} -\ \left\{ \left(\sum_{n} x_{in}\ \sum_{k} a_{kn}\ p(k \mid \boldsymbol{x}^i; \boldsymbol{W})\right)^2 \right\}\ +\ \left\{ \sum_{k} p(k \mid \boldsymbol{x}^i; \boldsymbol{W}) \left(\sum_{n} x_{in}\ a_{kn}\right)\ \left(\sum_{m} x_{im}\ a_{km}\right) \right\}\ =$$

$$\sum_{i} -\ \left\{ \left(\sum_{n} x_{in}\ \sum_{k} a_{kn}\ p(k \mid \boldsymbol{x}^i; \boldsymbol{W})\right)^2 \right\}\ +\ \left\{ \sum_{k} p(k \mid \boldsymbol{x}^i; \boldsymbol{W}) \left(\sum_{n} x_{in}\ a_{kn}\right)^2 \right\}$$

Softmax

If for each summand of the sum over i

$$\sum_k p(k \mid \boldsymbol{x}^i; \boldsymbol{W}) \left(\sum_n x_{in} \; a_{kn} \right)^2 \; - \; \left(\sum_k \; p(k \mid \boldsymbol{x}^i; \boldsymbol{W}) \sum_n x_{in} \; a_{kn} \right)^2 \; \geq \; 0$$

holds, then the Hessian is positive semidefinite.

Both sums are an expectation with density $p(k \mid \boldsymbol{x}^i; \boldsymbol{W})$ of the value $\sum_n x_{in} \; a_{kn}$ or its square.

The second moment minus the expection squared is the variance. Therefore the left hand side of inequality is a second central moment, which is larger than zero.

→ the Hessian is positive semidefinite

Chapter 5

Statistical Learning Theory

Statistical Learning Theory

- Does learning help for future tasks?
- Explains a model which explains the training data also new data?
- Yes, if complexity is bounded
- VC-dimension as complexity measure

- *statistical learning theory*: bounds for the generalization error (future)
- bounds comprise training error and complexity
- structural risk minimization minimizes both terms simultaneously

Statistical Learning Theory

- statistical learning theory:

  - the uniform law of large numbers (empirical risk minimization)

  - complexity constrained models (structural risk minimization)

- error bound on the mean squared error: bias-variance formulation

  - bias is training error = empirical risk

  - variance is model complexity
    high complexity → more models → more solutions → large variance

Error Bounds for a Gaussian Classification Task

We revisit the Gaussian classification task

$$R_{\min} = \int_X \min\{p(\boldsymbol{x}, y=-1), p(\boldsymbol{x}, y=1)\}\, d\boldsymbol{x}$$

$$= \int_X \min\{p(\boldsymbol{x} \mid y=-1)\ p(y=-1)\ ,\ p(\boldsymbol{x} \mid y=1)\ p(y=1)\}\, d\boldsymbol{x}$$

$$\forall_{a,b>0}: \ \forall_{0\leq\beta\leq 1}: \quad \min\{a,b\} \ \leq \ a^{\beta}\ b^{1-\beta}$$

$$\forall_{0\leq\beta\leq 1}: \quad R_{\min} \ \leq \ (p(y=1))^{\beta}\ \ (p(y=-1))^{1-\beta}$$

$$\int_X (p(\boldsymbol{x} \mid y=1))^{\beta}\ \ (p(\boldsymbol{x} \mid y=-1))^{1-\beta}\ \ d\boldsymbol{x}$$

Error Bounds for a Gaussian Classification Task

Gaussian assumption:

$$\int_X \left(p\left(\boldsymbol{x} \mid y=1\right)\right)^{\beta} \; \left(p\left(\boldsymbol{x} \mid y=-1\right)\right)^{1-\beta} \; d\boldsymbol{x} \;=\; \exp(-\,v(\beta))$$

where

$$\begin{aligned} v(\beta) \;&=\; \frac{\beta(1-\beta)}{2} \\ &\left(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1\right)^T \; \left(\beta\; \boldsymbol{\Sigma}_1 \;+\; (1-\beta)\; \boldsymbol{\Sigma}_2\right)^{-1} \left(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1\right) \\ &+\; \frac{1}{2} \ln \frac{\left|\beta\; \boldsymbol{\Sigma}_1 \;+\; (1-\beta)\; \boldsymbol{\Sigma}_2\right|}{\left|\boldsymbol{\Sigma}_1\right|^{\beta} \; \left|\boldsymbol{\Sigma}_2\right|^{1-\beta}} \end{aligned}$$

Chernoff bound: maximizing $v(\beta)$ with respect to β

Bhattacharyya bound: $\beta = \frac{1}{2}$

$$\begin{aligned} v(1/2) \;&=\; \frac{1}{4} \left(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1\right)^T \; \left(\boldsymbol{\Sigma}_1 \;+\; \boldsymbol{\Sigma}_2\right)^{-1} \left(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1\right) \\ &+\; \frac{1}{2} \ln \frac{\left|\frac{\boldsymbol{\Sigma}_1 \;+\; \boldsymbol{\Sigma}_2}{2}\right|}{\sqrt{\left|\boldsymbol{\Sigma}_1\right| \; \left|\boldsymbol{\Sigma}_2\right|}} \end{aligned}$$

Empirical Risk Minimization

empirical risk minimization (ERM) principle states:

if the training set is explained by the model then the model generalizes to future examples

restrict the complexity of the model class

empirical risk minimization (ERM): minimize error on training set

Complexity: Finite Number of Functions

- intuition why complexity matters
- complexity is just the number M of functions in model class
- difference training error (empirical risk) and test error (risk)

empirical risk: $R_{\text{emp}}(g, \boldsymbol{Z}) = \frac{1}{l} \sum_{i=1}^{l} L\left(y^i, g\left(\boldsymbol{x}^i\right)\right)$

finite set of functions $\{g_1, \ldots, g_M\}$

worst case (learning chooses unknown function):

$$\max_{j=1,\ldots,M} \| R_{\text{emp}}(g_j, l) - R(g_j) \|$$

Complexity: Finite Number of Functions

$$p\left(\max_{j=1,\ldots,M} \|R_{\text{emp}}(g_j,l) - R(g_j)\| > \epsilon\right) \leq$$

$$\sum_{j=1}^{M} p\left(\|R_{\text{emp}}(g_j,l) - R(g_j)\| > \epsilon\right) \leq$$

$$M\ 2\ \exp\left(-2\,\epsilon^2\,l\right) = 2\ \exp\left(\left(\frac{\ln M}{l} - 2\,\epsilon^2\right) l\right) = \delta$$

$\max_{j=1,\ldots,M} f_j > \epsilon$

$f_1 > \epsilon \text{ OR } f_2 > \epsilon \ \ldots$

$p(a \text{ OR } b) \leq p(a) + p(b)$: union bound

distance of average and expectation: Chernoff inequality

(for each j) $p(\mu_l - s > \epsilon) < \exp\left(-2\,\epsilon^2\,l\right)$

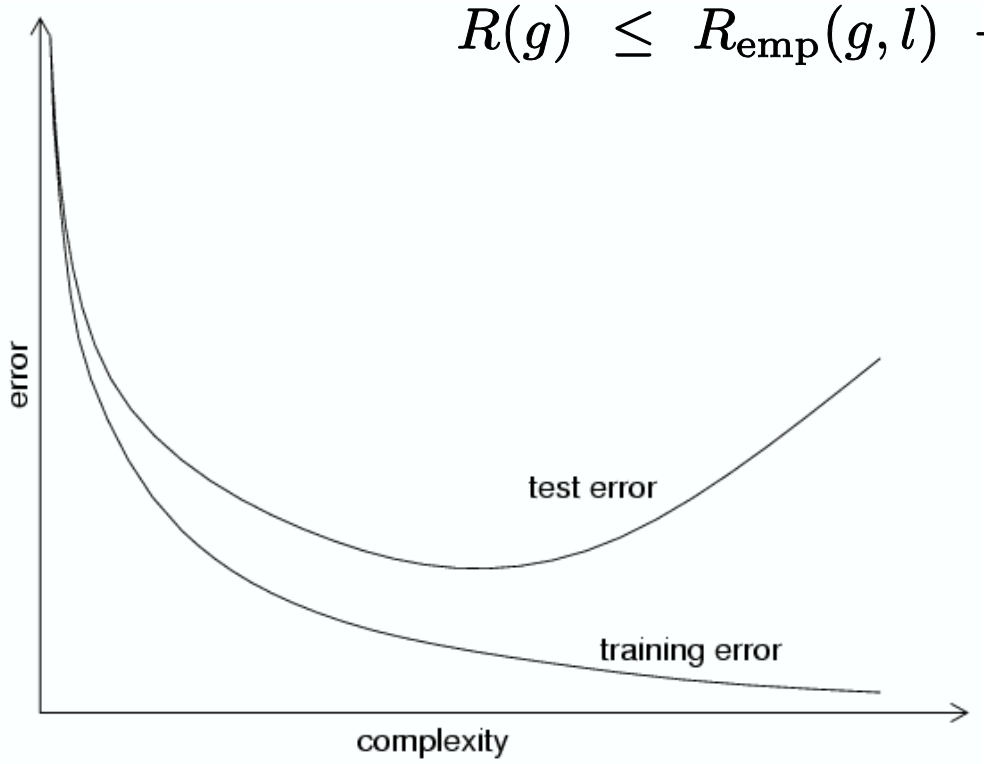
where μ_l is the empirical mean of the true value s for l trials

we obtain complexity term $\epsilon(l, M, \delta) = \sqrt{\frac{\ln M + \ln(2/\delta)}{2\,l}}$

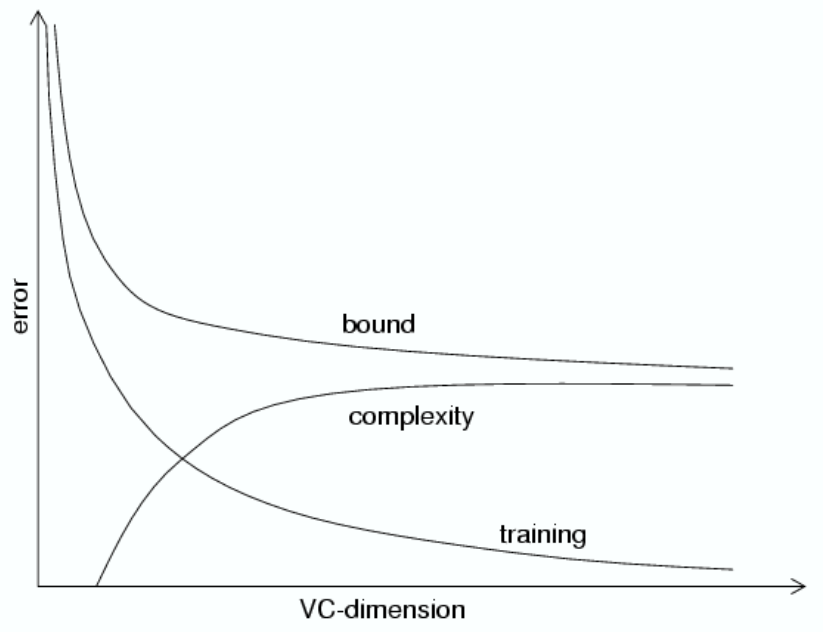
Complexity: Finite Number of Functions

Theorem 1 (Finite Set Error Bound)
With probability of at least $(1 - \delta)$
over possible training sets with l *elements*
and for M *possible functions we have*

$$R(g) \leq R_{\text{emp}}(g, l) + \epsilon(l, M, \delta) .$$



Complexity: Finite Number of Functions



$\epsilon(l, M, \delta)$ should converge to zero as l increases, therefore

$$\frac{\ln M}{l} \overset{l \to \infty}{\to} 0$$

$$\epsilon(l, M, \delta) = \sqrt{\frac{\ln M + \ln(2/\delta)}{2\, l}}$$

Complexity: VC-Dimension

- we want apply the previous bound for infinite function classes
- idea: on training set only finite number of functions is different
- example: all discriminant functions g giving the same classification function sign $g(.)$
- parametric models $g(.;\boldsymbol{w})$ with parameter vector $\boldsymbol{w}$
- does minimizing the parameter on the training set convergence to the best solution with increasing training set?
- empirical risk minimization (ERM): consistent or not?
- do we select better models with larger training sets?

Complexity: VC-Dimension

- parameter which minimizes the empirical risk for l training examples:

$$\hat{\boldsymbol{w}}_l \;=\; \arg\min_{\boldsymbol{w}} R_{\text{emp}}(g(.;\boldsymbol{w}), l)$$

- ERM is *consistent* if

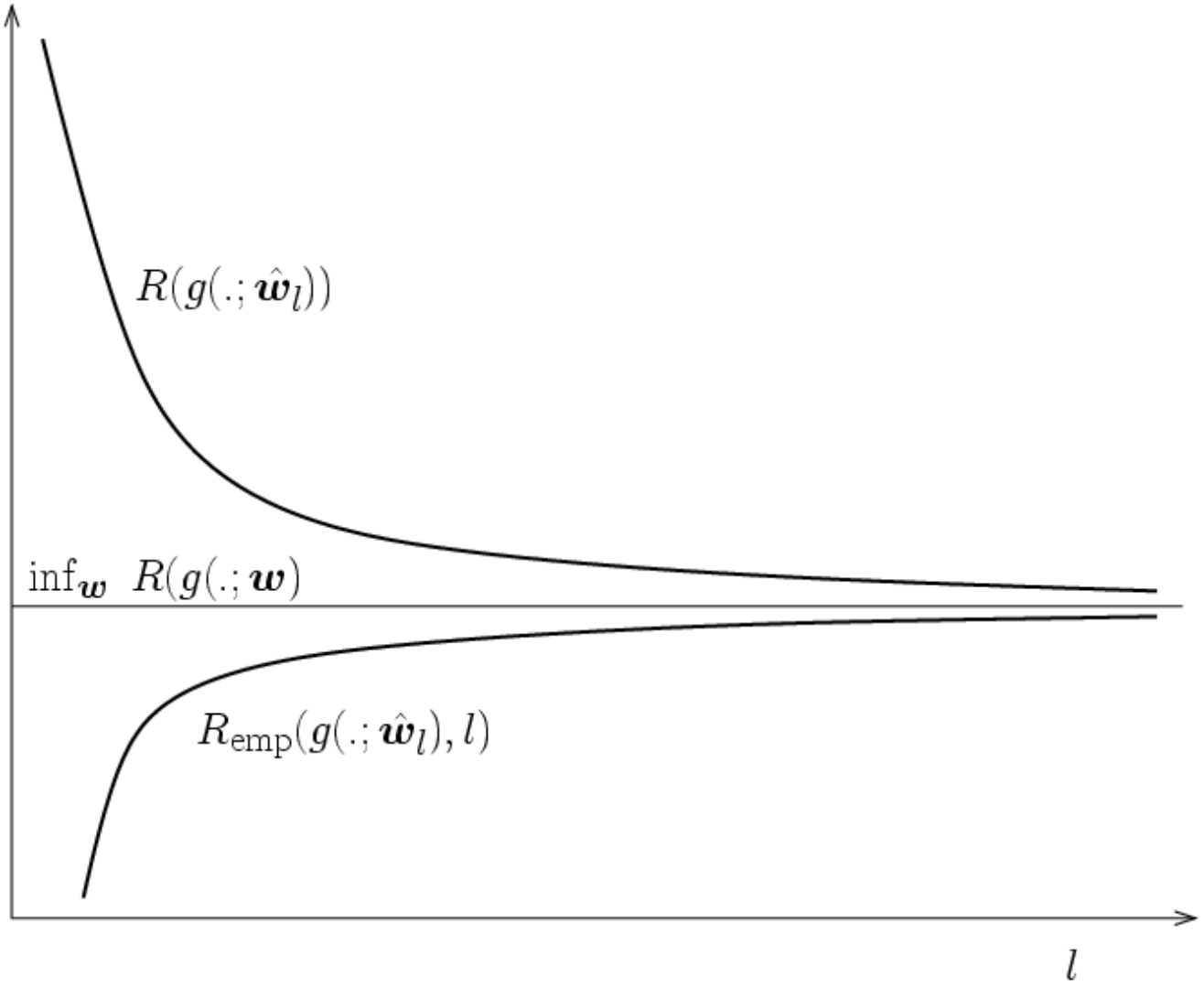
$$R(g(.;\hat{\boldsymbol{w}}_l)) \overset{l\to\infty}{\to} \inf_{\boldsymbol{w}} R(g(.;\boldsymbol{w}))$$ (overfitting)

convergence in probability

$$R_{\text{emp}}(g(.;\hat{\boldsymbol{w}}_l), l) \overset{l\to\infty}{\to} \inf_{\boldsymbol{w}} R(g(.;\boldsymbol{w}))$$ (underfitting)

Empirical risk and expected risk converge to minimal risk

Complexity: VC-Dimension



Complexity: VC-Dimension

ERM is strictly consistent if for all

$$\Lambda(c) = \left\{ \boldsymbol{w} \mid \boldsymbol{z} = (\boldsymbol{x}, y), \int L(y, g(\boldsymbol{x}; \boldsymbol{w})) p(\boldsymbol{z}) d\boldsymbol{z} \geq c \right\}$$

$$\inf_{\boldsymbol{w} \in \Lambda(c)} R_{\text{emp}}(g(.; \boldsymbol{w}), l) \overset{l \to \infty}{\to} \inf_{\boldsymbol{w} \in \Lambda(c)} R(g(.; \boldsymbol{w}))$$

holds (convergence in probability)

Instead of “strictly consistent” we write “consistent”

maximum likelihood is consistent for a set of densities if

$$0 < a \leq p(\boldsymbol{x}; \boldsymbol{w}) \leq A < \infty$$

$\exists_{\boldsymbol{w}_1}$:

$$\inf_{\boldsymbol{w}} \frac{1}{l} \sum_{i=1}^{l} (-\ln p(\boldsymbol{x}; \boldsymbol{w})) \overset{l \to \infty}{\to} \inf_{\boldsymbol{w}} \int_X (-\ln p(\boldsymbol{x}; \boldsymbol{w})) \, p(\boldsymbol{x}; \boldsymbol{w}_1) \, d\boldsymbol{x}$$

Complexity: VC-Dimension

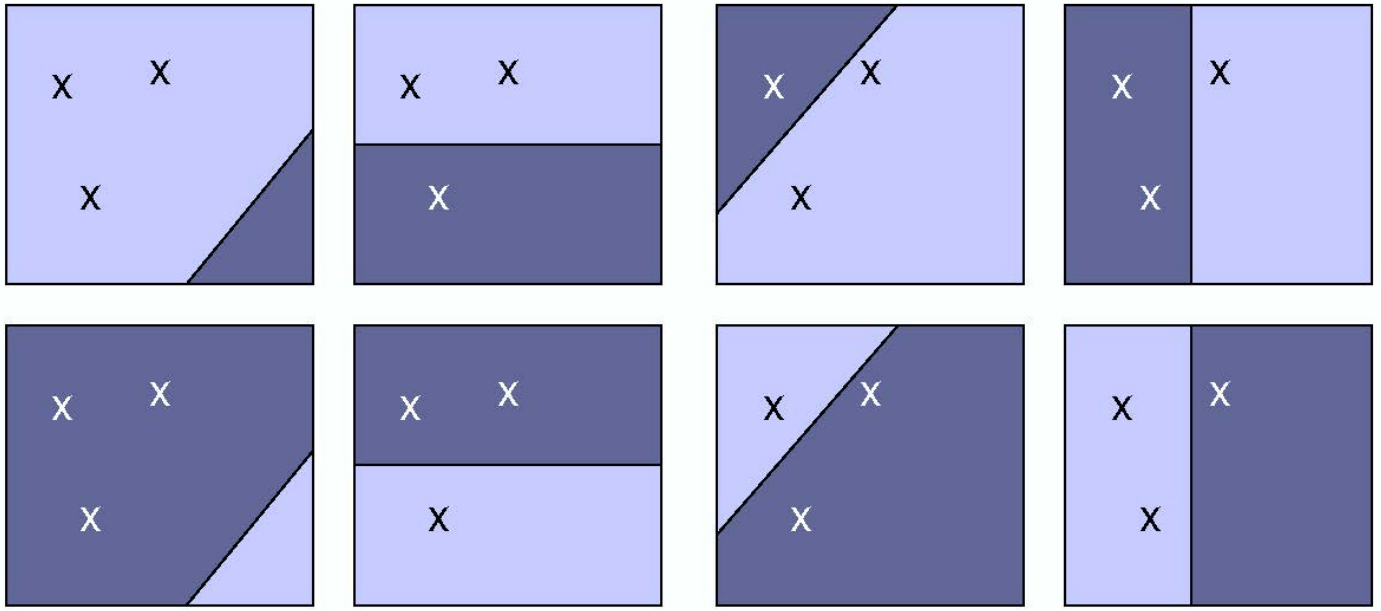
- Under what conditions is the ERM consistent?

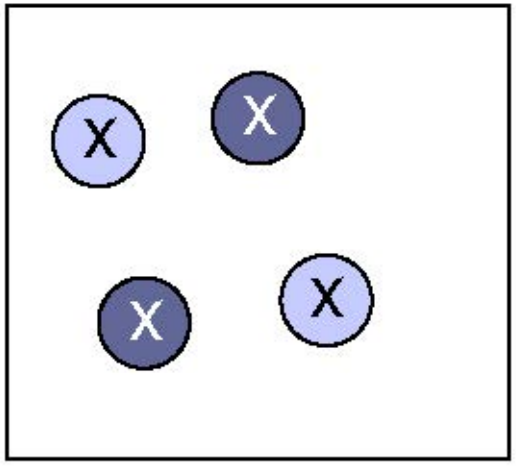
- New concepts and new capacity measures:
  - points to be shattered
  - annealed entropy
  - entropy (new definition)
  - growth function
  - VC-dimension

Possibilities to label the input data $\boldsymbol{x}^i$ by binary labels $y^i \in \{-1, 1\}$
2^l → shattering the input data

complexity of a model class: number different labelings
how many points can be shattered

Complexity: VC-Dimension





Note, that each “x” is placed in a circle around its position independent of the other “x”. Therefore each constellation represents a set with non-zero probability mass.

Complexity: VC-Dimension

- number of points a function class can shatter: *VC-dimension* (later)
- function class $\mathcal{F}$
- shattering coefficient: $N_{\mathcal{F}}(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^l)$ (# labeling class can shatter)
- entropy of a function class: $H_{\mathcal{F}}(l) = \mathrm{E}_{(\boldsymbol{x}^1,\ldots,\boldsymbol{x}^l)} \ln N_{\mathcal{F}}(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^l)$
- annealed entropy of a function class:

$$H^{\mathrm{ann}}_{\mathcal{F}}(l) = \ln \mathrm{E}_{(\boldsymbol{x}^1,\ldots,\boldsymbol{x}^l)} N_{\mathcal{F}}(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^l)$$

- growth function of a function class:

$$G_{\mathcal{F}}(l) = \ln \sup_{(\boldsymbol{x}^1,\ldots,\boldsymbol{x}^l)} N_{\mathcal{F}}(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^l)$$

$$H_{\mathcal{F}}(l) \underset{\text{Jensen}}{\leq} H^{\mathrm{ann}}_{\mathcal{F}}(l) \underset{\text{supremum}}{\leq} G_{\mathcal{F}}(l)$$

Complexity: VC-Dimension

ERM is consistent:

Theorem 1 (Sufficient Condition for Consistency of ERM)
If

$$\lim_{l \to \infty} \frac{H_{\mathcal{F}}(l)}{l} = 0$$

then ERM is consistent.

ERM has fast rate of convergence (exponential convergence):

$$p\left(\sup_{\boldsymbol{w}} |R(g(.;\boldsymbol{w})) - R_{\text{emp}}(g(.;\hat{\boldsymbol{w}}_l), l)| > \epsilon\right) < b \exp\left(-c\, \epsilon^2\, l\right)$$

Theorem 1 (Sufficient Condition for Fast Rate)
If

$$\lim_{l \to \infty} \frac{H^{\text{ann}}_{\mathcal{F}}(l)}{l} = 0$$

then ERM has a fast rate of convergence.

Complexity: VC-Dimension

theorems valid for a given probability measure on the observations probability measure enters the formulas via the expectation

Theorem 1 (Consistency of ERM for Any Probability)
The condition

$$\lim_{l \to \infty} \frac{G_{\mathcal{F}}(l)}{l} = 0$$

is necessary and sufficient for the ERM to be consistent and also ensures a fast rate of convergence.

Complexity: VC-Dimension

- VC (Vapnik-Chervonenkis) dimension d_{VC} is the largest integer for which $G_{\mathcal{F}}(l) = l \ln 2$ holds

$$d_{\mathrm{VC}} = \max_l \{ l \mid G_{\mathcal{F}}(l) = l \ln 2 \} .$$

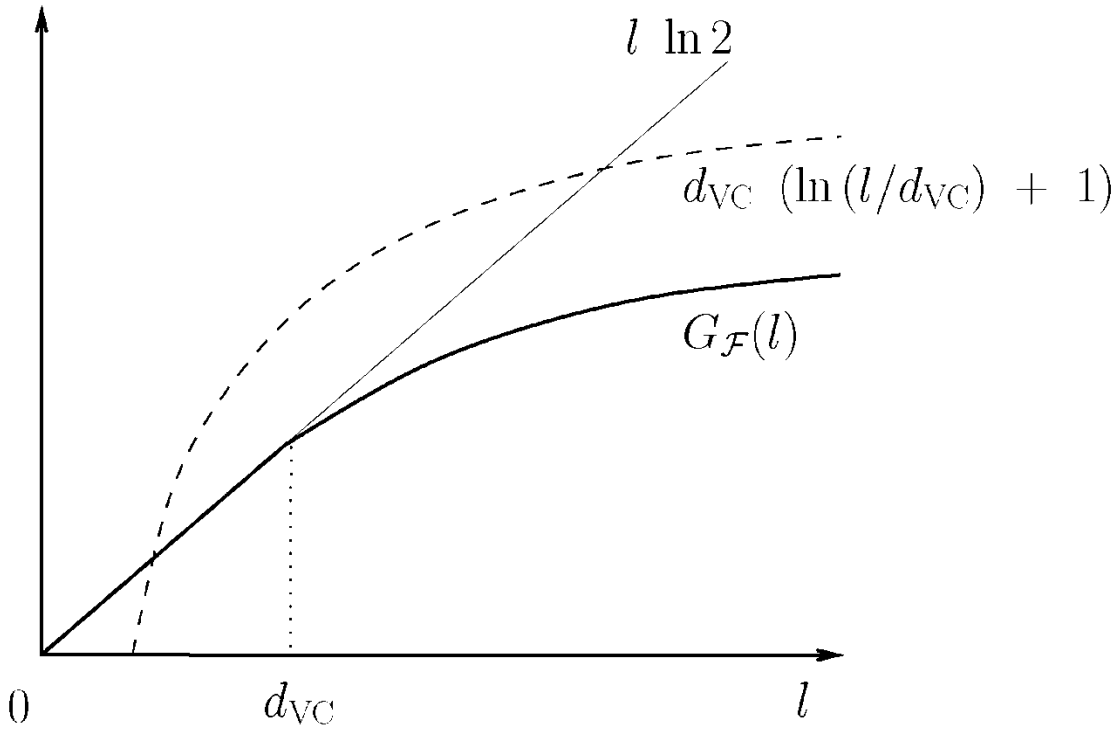
If the maximum does not exists: $d_{\mathrm{VC}} = \infty$

- VC-dimension is the maximum number of vectors that can be shattered by the function class

Complexity: VC-Dimension

Theorem 1 (VC-Dimension Bounds the Growth Function)
The growth function is bounded by

$$G_{\mathcal{F}}(l) \begin{cases} = & l \ln 2 & \text{if } l \leq d_{\text{VC}} \\ \leq & d_{\text{VC}} \left(1 + \ln \frac{l}{d_{\text{VC}}}\right) & \text{if } l > d_{\text{VC}} \end{cases} .$$



Complexity: VC-Dimension

- function class with finite VC-dim.: consistent and converges fast

- Linear functions in d-dimensional of the input space:

$$g(\boldsymbol{x}; \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{x} \qquad d_{\text{VC}} = d$$

$$g(\boldsymbol{x}; \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{x} + b \qquad d_{\text{VC}} = d + 1$$

- Nondecreasing nonlinear one-dimensional functions

$$\sum_{i=1}^{k} \left| a_i \; x^i \right| \; \text{sign} x + a_0 \qquad d_{\text{VC}} = 1$$

- Nonlinear one-dimensional functions:

$$\sin(w \; z) \qquad d_{\text{VC}} = \infty$$

Complexity: VC-Dimension

- Neural Networks:

$$d_{\text{VC}} \leq 2\, W\, \log_2(e\, M)$$

M are the number of units, W is the number of weights, e is the base of the natural logarithm (Baum & Haussler 89, Shawe-Taylor & Anthony 91)

$$d_{\text{VC}} \leq 2\, W\, \log_2(24\, e\, W\, D)$$

inputs restricted to $[-D; D]$

Bartlett & Williamson (1996)

Error Bounds

- idea of deriving the error bounds: set of distinguishable functions cardinality given by $N_{\mathcal{F}}$

- trick of two half-samples and their difference ("symmetrization"):

$$p\left(\sup_{\boldsymbol{w}}\left|\frac{1}{l}\sum_{i=1}^{l}L\left(y^i,g\left(\boldsymbol{x}^i;\boldsymbol{w}\right)\right)-\frac{1}{l}\sum_{i=l+1}^{2l}L\left(y^i,g\left(\boldsymbol{x}^i;\boldsymbol{w}\right)\right)\right|>\epsilon-\frac{1}{l}\right)\geq$$

$$\frac{1}{2}\,p\left(\sup_{\boldsymbol{w}}\left|\frac{1}{l}\sum_{i=1}^{l}L\left(y^i,g\left(\boldsymbol{x}^i;\boldsymbol{w}\right)\right)-R(g(.;\boldsymbol{w})\right|>\epsilon\right)$$

therefore in the following we use $2\ l$
→ l example used for complexity definition and l for empirical error

minimal possible risk:

$$\boldsymbol{w}_0 = \arg\min_{\boldsymbol{w}} R(g(.;\boldsymbol{w}))$$

$$R_{\min} = \min_{\boldsymbol{w}} R(g(.;\boldsymbol{w})) = R(g(.;\boldsymbol{w}_0))$$

Error Bounds

Theorem 1 (Error Bound)
With probability of at least $(1 - \delta)$ *over possible training sets with* l *elements, the parameter* $\boldsymbol{w}_l$ *which minimizes the empirical risk we have*

$$R(g(.; \boldsymbol{w}_l)) \leq R_{\text{emp}}(g(.; \boldsymbol{w}_l), l) + \sqrt{\epsilon(l, \delta)} .$$

With probability of at least $(1 - 2\delta)$ *the difference between the optimal risk and the risk of* $\boldsymbol{w}_l$ *is bounded by*

$$R(g(.; \boldsymbol{w}_l)) - R_{\min} < \sqrt{\epsilon(l, \delta)} + \sqrt{\frac{-\ln \delta}{l}}$$

Here $\epsilon(l, \delta)$ *can be defined for a specific probability as*

$$\epsilon(l, \delta) = \frac{8}{l} \left(H_{\mathcal{F}}^{\text{ann}}(2l) + \ln(4/\delta) \right)$$

or for any probability as

$$\epsilon(l, \delta) = \frac{8}{l} \left(G_{\mathcal{F}}(2l) + \ln(4/\delta) \right)$$

where the later can be expressed though the VC-dimension d_{VC}

$$\epsilon(l, \delta) = \frac{8}{l} \left(d_{\text{VC}} \left(\ln(2l/d_{\text{VC}}) + 1 \right) + \ln(4/\delta) \right)$$

Error Bounds

- complexity measure depend on the ratio $\frac{d_{\text{VC}}}{l}$
- The bound above is from Anthony and Bartlett whereas an older bound from Vapnik is

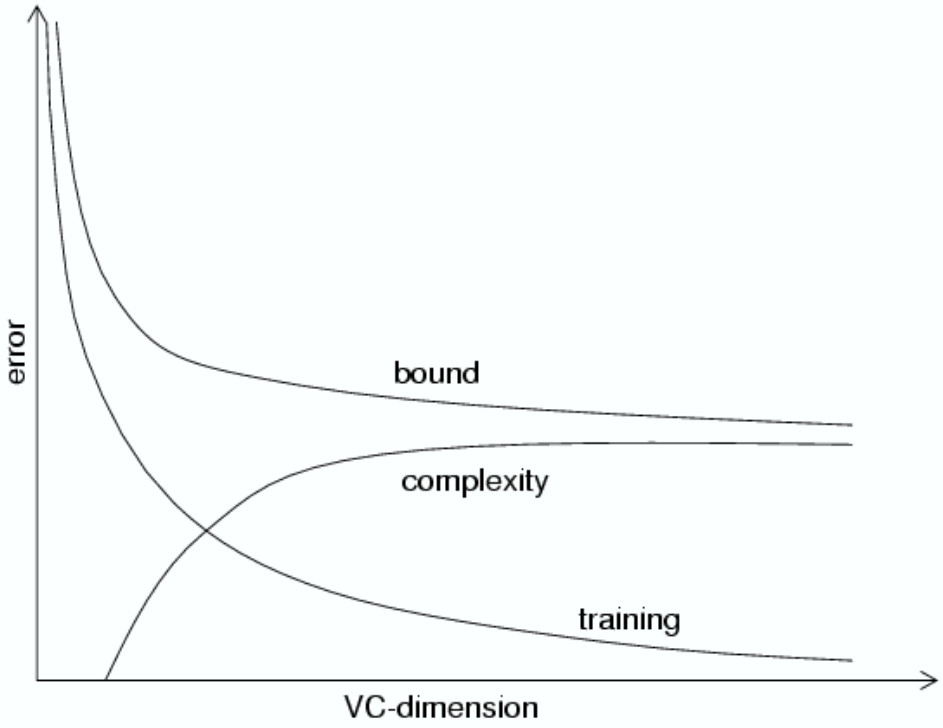
$$R(g(.;\boldsymbol{w}_l)) \leq R_{\text{emp}}(g(.;\boldsymbol{w}_l), l) + \frac{\epsilon(l,\delta)}{2}\left(1 + \sqrt{1 + \frac{R_{\text{emp}}(g(.;\boldsymbol{w}_l), l)}{\epsilon(l,\delta)}}\right)$$

- complexity term decreases with $\frac{1}{\sqrt{l}}$
- zero empirical risk then the bound on the risk decreases with $\frac{1}{\sqrt{l}}$
- later: *expected* risk decreases with $\frac{1}{l}$

Error Bounds

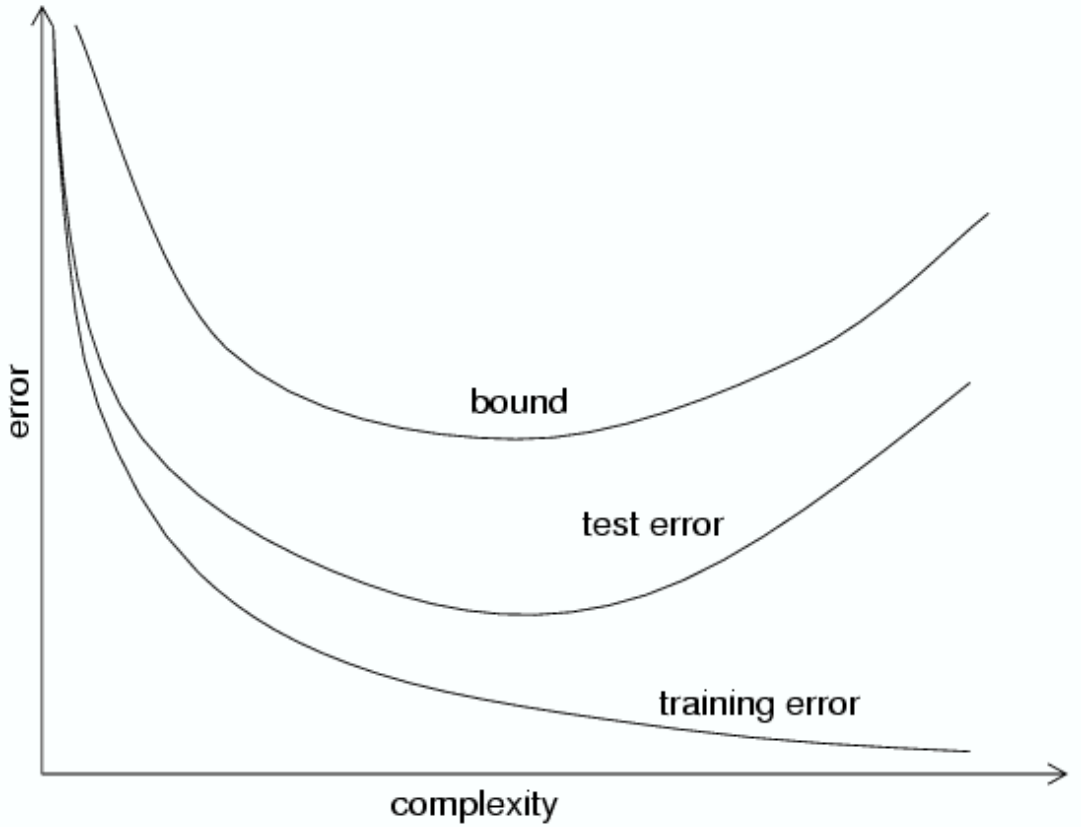
- bound on the risk $R \leq R_{\text{emp}} + \text{complexity}$

- bound is similar to the bias-variance formulation
  - bias corresponds to empirical risk
  - variance corresponds to complexity



Error Bounds

- In many practical cases the bound is not useful: not tight
- However in many practical cases the minimum of the bound is close to the minimum of the test error



Error Bounds

- regression: instead of the shattering coefficient **covering number** (covering of the functions with distance epsilon)

- growth function is then: $G(\varepsilon, \mathcal{F}, l) = \ln \sup_{\boldsymbol{X}} \mathcal{N}(\varepsilon, \mathcal{F}, \boldsymbol{X}_\infty)$

bounds on the generalization error:

$$R(g(.; \boldsymbol{w}_l)) \leq R_{\text{emp}}(g(.; \boldsymbol{w}_l), l) + \sqrt{\epsilon(\varepsilon, l, \delta)}$$

where

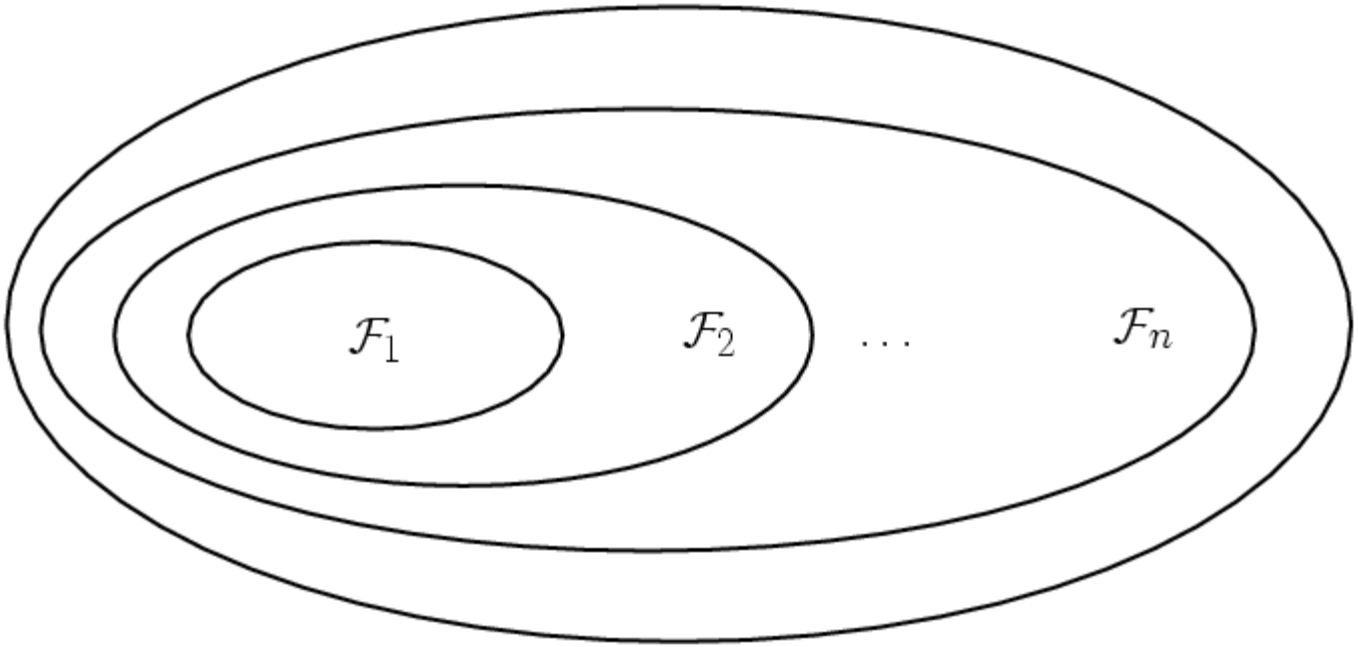
$$\epsilon(\varepsilon, l, \delta) = \frac{36}{l} \left(\ln(12\ l) + G(\varepsilon/6, \mathcal{F}, l) - \ln \delta\right)$$

Structural Risk Minimization

The Structural Risk Minimization (SRM) principle minimizes the guaranteed risk that is a bound on the risk instead of the empirical risk alone

Structural Risk Minimization

- nested set of function classes: $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \ldots \subset \mathcal{F}_n \subset \ldots$ where class $\mathcal{F}_n$ possesses VC-dimension d_{VC}^n and $d_{\text{VC}}^1 \leq d_{\text{VC}}^2 \leq \ldots d_{\text{VC}}^n \leq \ldots$



Structural Risk Minimization

- Example for SRM: **minimum description length**
  - sender transmits a model (once) and the inputs and errors
  - receiver has to recover the labels

  goal: minimize transmission costs (description length)

$$\text{transmissioncosts} \;=\; R_{\text{emp}} \;+\; \text{complexity}$$

- Is the SRM principle consistent? How fast does it converge?

  SRM is consistent !!

asymptotic rate of convergence:

$$r(l) \;=\; |R^n_{\min} \;-\; R_{\min}| \;+\; \sqrt{\frac{d^n_{\text{VC}} \;\ln l}{l}} \;,$$

where $R^n_{\min}$ is the minimal risk of the function class $\mathcal{F}_n$

$$p\left(\lim_{l\to\infty} \;\sup\; r^{-1}(l) \;\left|R\left(g\left(.\,;\,\boldsymbol{w}_l^{\mathcal{F}_n}\right)\right) \;-\; R_{\min}\right| \;<\; \infty\right) \;=\; 1$$

Structural Risk Minimization

- $|R^n_{\min} - R_{\min}| \overset{l \to \infty}{\to} 0$

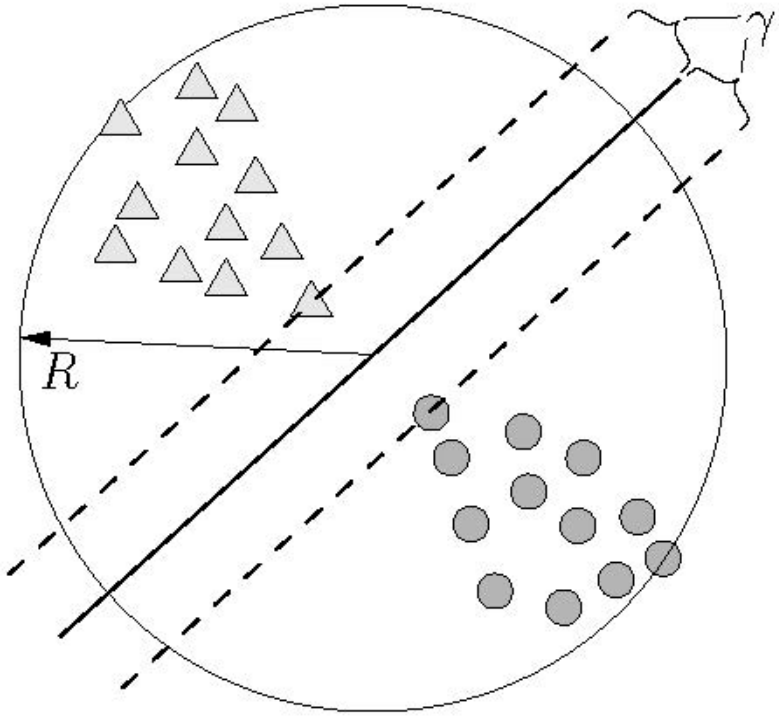
If the optimal solution belongs to some class $\mathcal{F}_n$ then the convergence rate is

$$r(l) = O\left(\sqrt{\frac{\ln l}{l}}\right)$$

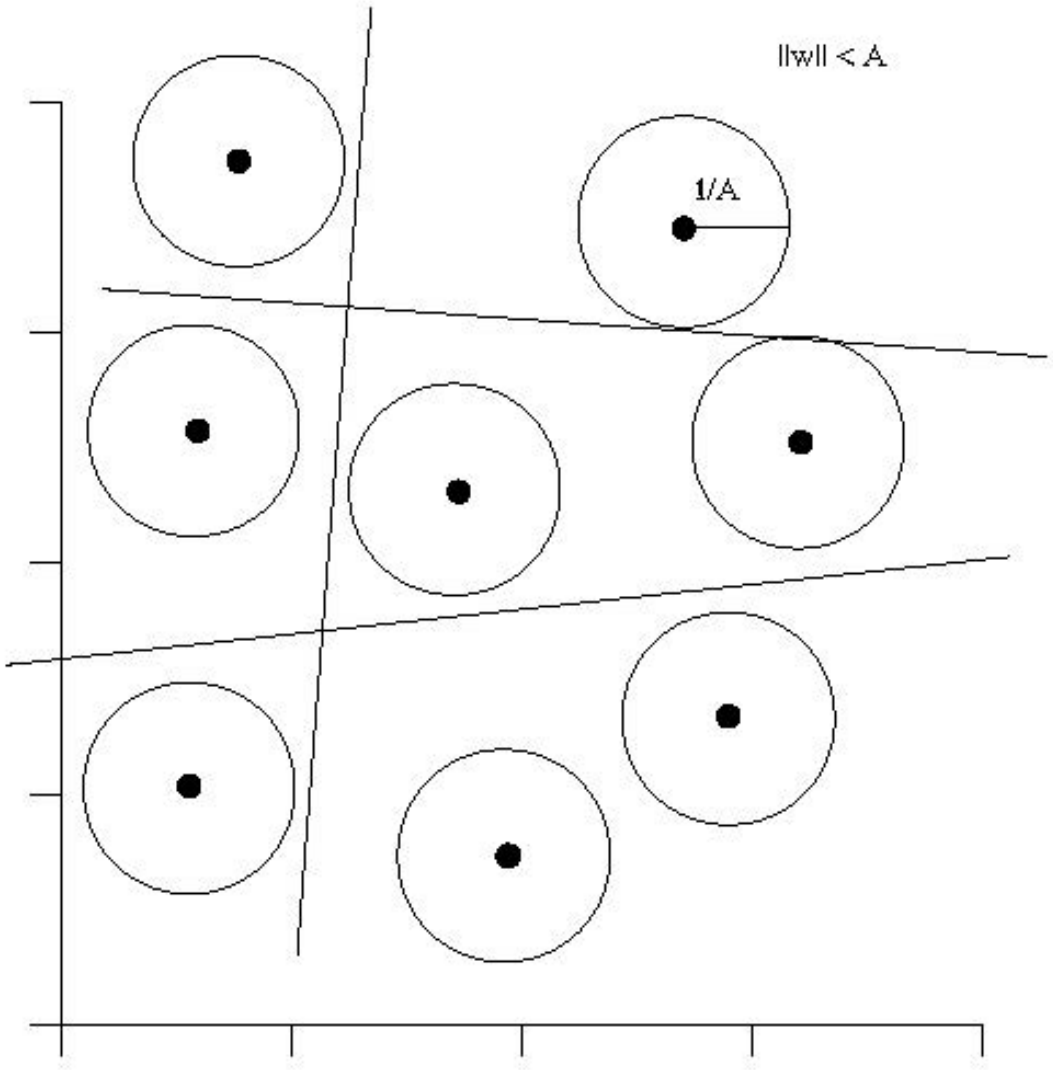
Margin as Complexity Measure

- VC-dimension: restrictions on the class of function

- most famous:
  zero isoline of the discriminant function has minimal distance γ (margin) to all training data points which are contained in a sphere with radius $\boldsymbol{R}$



Margin as Complexity Measure



Margin as Complexity Measure

- linear discriminant functions $\boldsymbol{w}^T \boldsymbol{x} + b$
- classification function $\text{sign}\left(\boldsymbol{w}^T \boldsymbol{x} + b\right)$
- scaling $\boldsymbol{w}$ and b does not change classification function
- classification function: one representative discriminant function
- canonical form w.r.t. the training data $\boldsymbol{X}$:

$$\min_{i=1,\ldots,l} \left| \boldsymbol{w}^T \boldsymbol{x}^i + b \right| = 1$$

Margin as Complexity Measure

Theorem 1 (Margin bounds VC-dimension)
The class of classification functions $sign\left(\boldsymbol{w}^T\boldsymbol{x} + b\right)$, *where the discriminant function* $\boldsymbol{w}^T\boldsymbol{x} + b$ *is in its canonical form versus* $\boldsymbol{X}$ *which is contained in a sphere of radius* R, *and where* $\|\boldsymbol{w}\| \leq \frac{1}{\gamma}$ *satisfy*

$$d_{\mathrm{VC}} \leq \frac{R^2}{\gamma^2} .$$

Margin as Complexity Measure

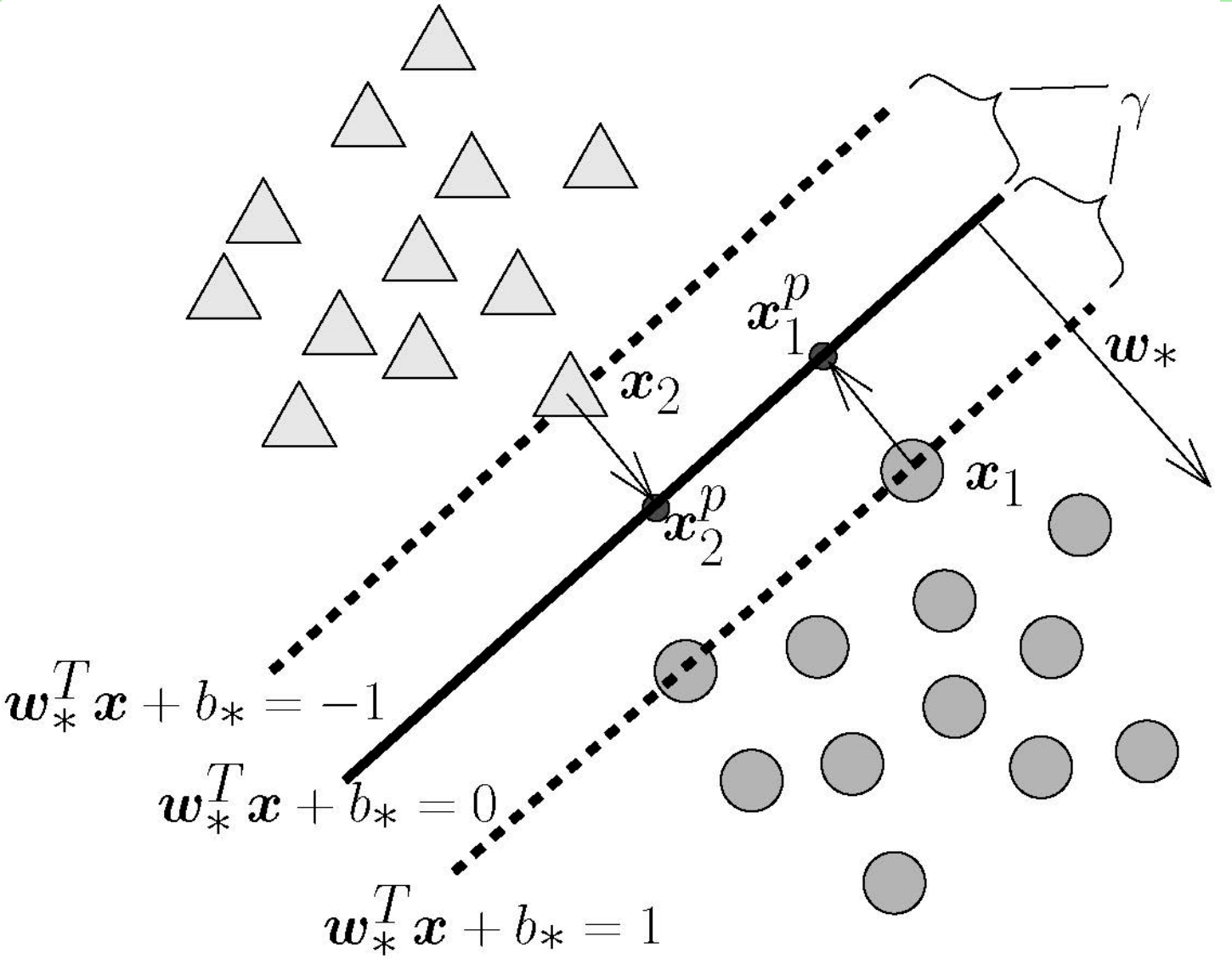
If at least one data point exists for which the discriminant function is positive and at least one data point exists for which it is negative, then we can optimize b and rescale $\|\boldsymbol{w}\|$ in order to obtain the smallest $\|\boldsymbol{w}\|$

This gives the tightest bound and smallest VC-dimension

After optimizing b and rescaling $\boldsymbol{w}$ we have points for which

$$\boldsymbol{w}^T \boldsymbol{x}^1 + b = 1 \qquad \boldsymbol{w}^T \boldsymbol{x}^2 + b = -1$$

Margin as Complexity Measure



Margin as Complexity Measure

After this optimization: the distance of $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ to the boundary function is $\frac{1}{\|\boldsymbol{w}_*\|} = \gamma$

Theorem 1 (Margin Error Bound)
The classification functions sign $\left(\boldsymbol{w}^T\boldsymbol{x} + b\right)$ are restricted to $\|\boldsymbol{w}\| \leq \frac{1}{\gamma}$ and $\|\boldsymbol{x}\| < R$. Let ν be the fraction of training examples which have a margin (distance to $\boldsymbol{w}^T\boldsymbol{x} + b = 0$) smaller than $\frac{\rho}{\|\boldsymbol{w}\|}$.

With probability at least of $(1 - \delta)$ of drawing l examples, the probability to misclassify a new example is bounded from above by

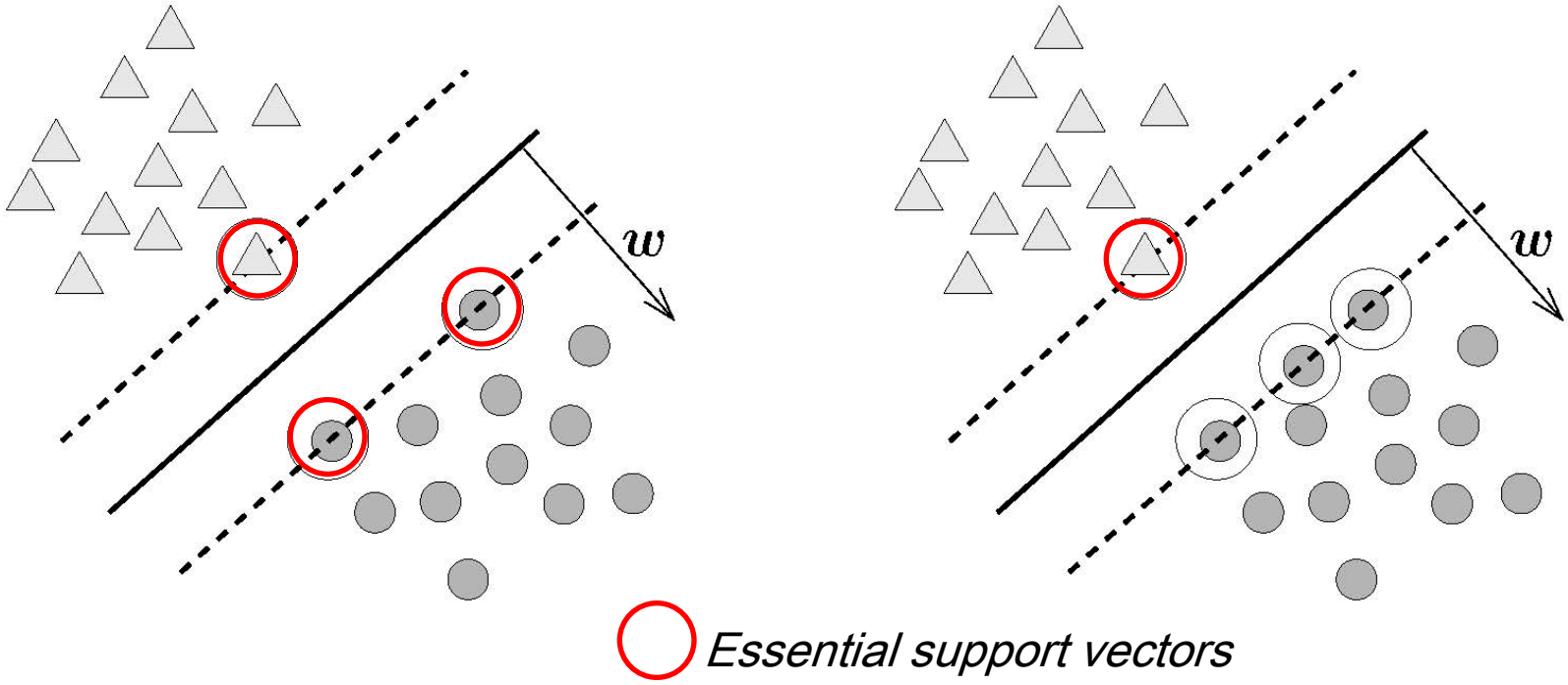
$$\nu + \sqrt{\frac{c}{l}\left(\frac{R^2}{\rho^2\,\gamma^2}\ln^2 l + \ln(1/\delta)\right)},$$

where c is a constant.

Average Error Bounds for SVMs

Leave-One-Out Cross Validation (LOO CV) estimator is almost unbiased (see Section 2.2.2).

Essential support vectors: support vectors for which the solution changes if they are removed from the training set



Average Error Bounds for SVMs

number of essential support vectors: $k_l \leq d+1$

d is the dimension of the space

radius of the sphere which contains all essential support vectors: r_l

expected risk $ER(g(.;\boldsymbol{w}_l))$, where the expectation is taken over the training set of size l and a test data point

Average Error Bounds for SVMs

Theorem 1 (Average Bounds for SVMs)
For the expected risk with above definitions

$$ER(g(.;\boldsymbol{w}_l)) \leq \frac{Ek_{l+1}}{l+1} \qquad ER(g(.;\boldsymbol{w}_l)) \leq \frac{d+1}{l+1}$$

$$ER(g(.;\boldsymbol{w}_l)) \leq \frac{E\left(\frac{r_{l+1}}{\gamma_{l+1}}\right)^2}{l+1}$$

$$ER(g(.;\boldsymbol{w}_l)) \leq \frac{E\min\left\{k_{l+1}\ ,\ \left(\frac{r_{l+1}}{\gamma_{l+1}}\right)^2\right\}}{l+1}$$

$$ER(g(.;\boldsymbol{w}_l)) \leq \frac{E\left(\left(k^*_{l+1}\right)^2 \sum_{i^*} \alpha_{i^*} + m\right)}{l+1}$$

$$C \leq r_l^{-2}: \quad ER(g(.;\boldsymbol{w}_l)) \leq \frac{E\left(\left(k_{l+1}\right)^2 \sum_i \alpha_i\right)}{l+1}$$

i^* are SVs with $0 < \alpha_{i^*} < C$ and m is number SVs with $\alpha_i = C$

Average Error Bounds for SVMs

bounds are based on the fact that the leave-one-out cross validation is almost unbiased → $\frac{1}{l+1}$

these average bounds are of the order $\frac{1}{l}$

worst case bounds (Vapnik bounds) are of the order $\frac{1}{\sqrt{l}}$

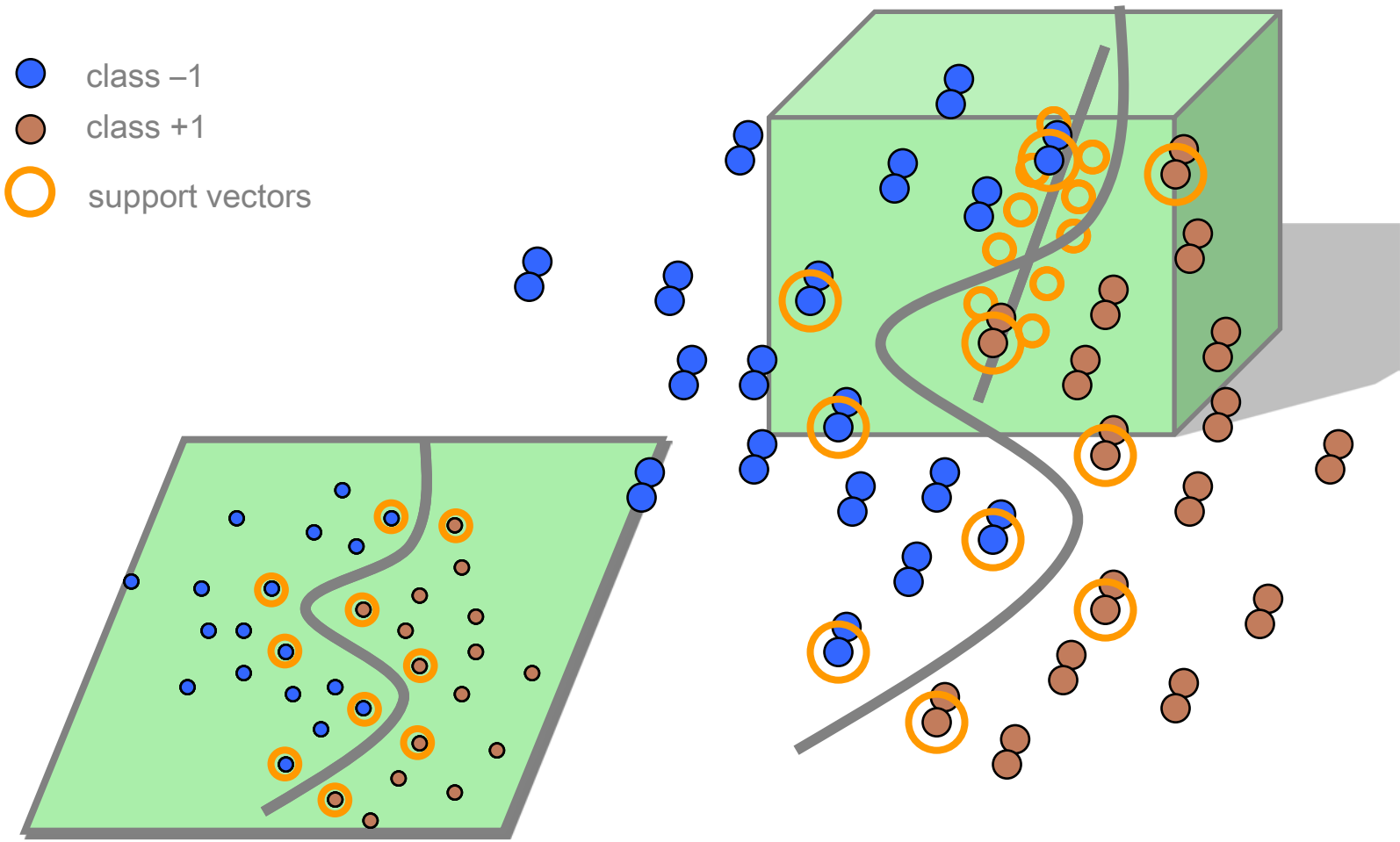
variance of the expected risk?

Chapter 6

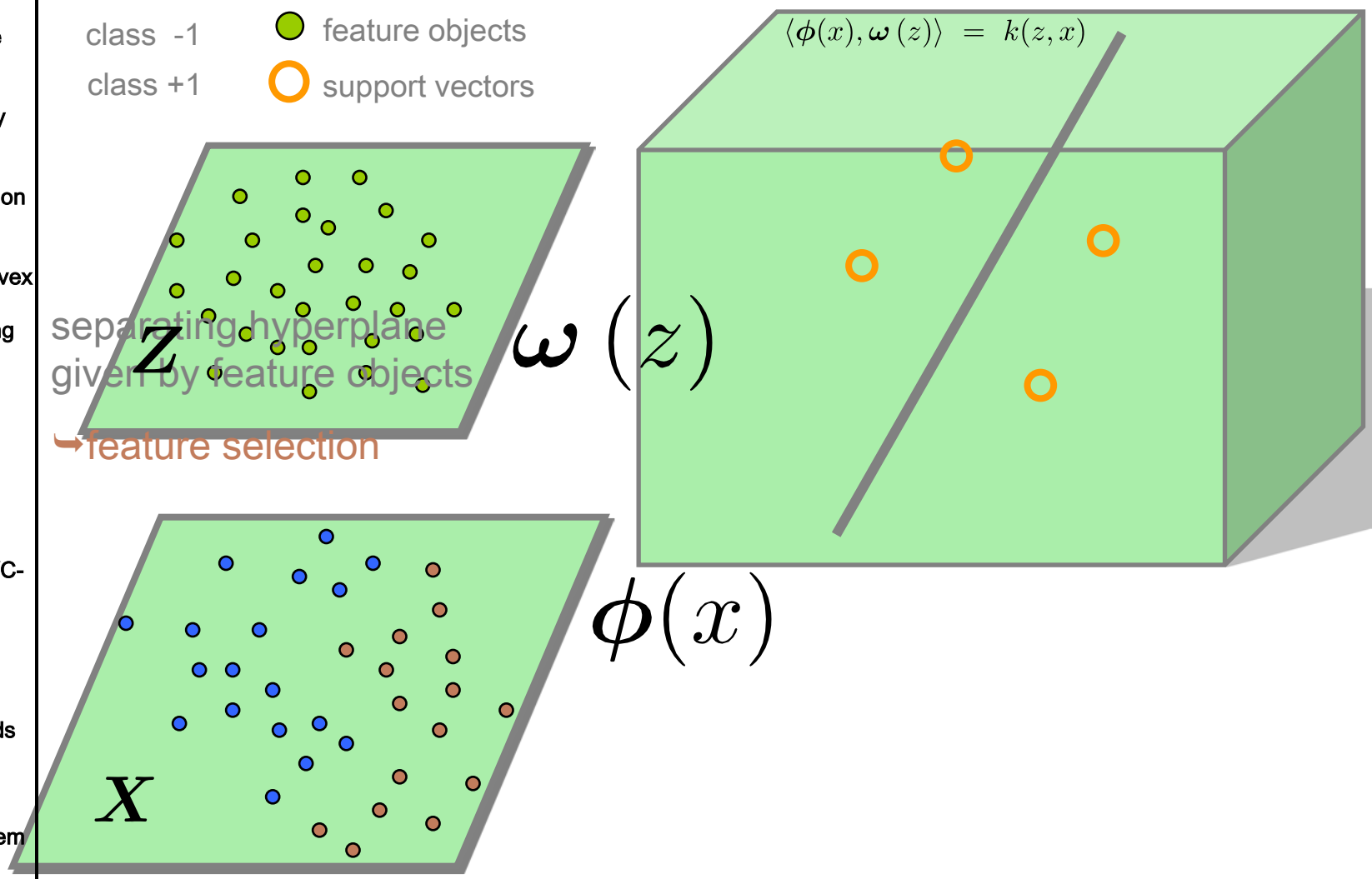
Theory of Kernels and Dot Products

Support Vector Machine

non-linear support vector machine



Support Vector Machine



➥feature selection

Kernels, Dot Products, and Mercer's Theorem

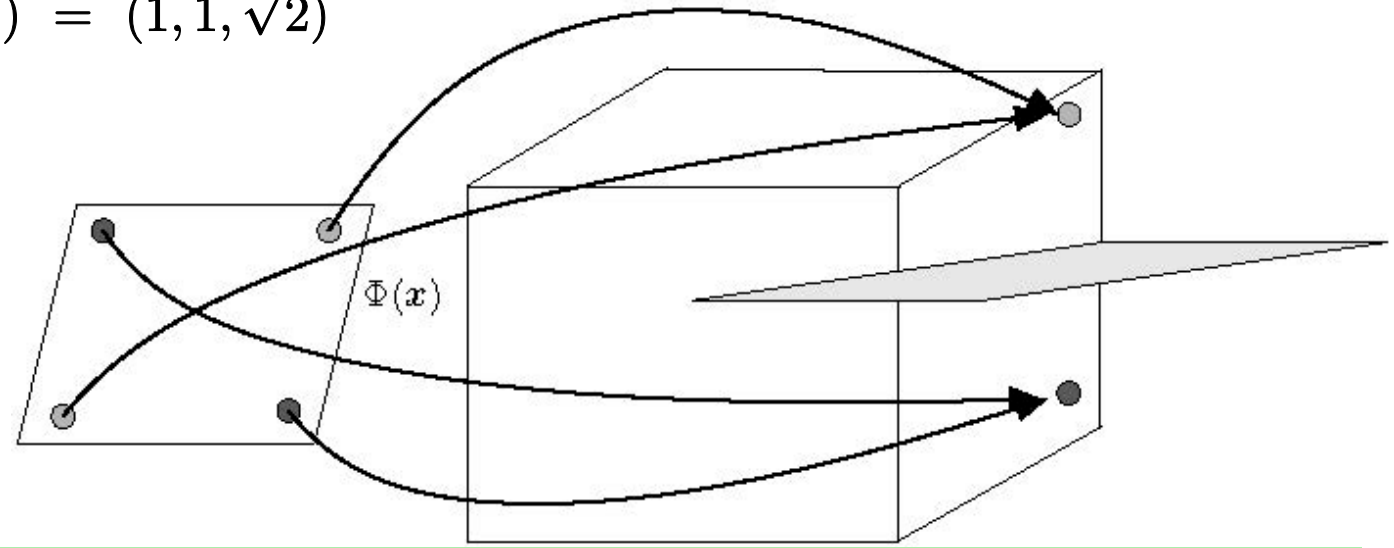
Example for a kernel: $\boldsymbol{\Phi}(\boldsymbol{x}) = \left(x_1^2\ ,\ x_2^2\ ,\ \sqrt{2}\ x_1\ x_2\right)$

$$\boldsymbol{x}^1 = (1,1), \boldsymbol{x}^2 = (1,-1), \boldsymbol{x}^3 = (-1,1), \boldsymbol{x}^4 = (-1,-1)$$

with labels $y^1 = -1, y^2 = 1, y^3 = 1, y^4 = -1$

not separable in 2D but in feature space

$$\boldsymbol{\Phi}(\boldsymbol{x}^1) = (1,1,\sqrt{2})$$
$$\boldsymbol{\Phi}(\boldsymbol{x}^2) = (1,1,-\sqrt{2})$$
$$\boldsymbol{\Phi}(\boldsymbol{x}^3) = (1,1,-\sqrt{2})$$
$$\boldsymbol{\Phi}(\boldsymbol{x}^4) = (1,1,\sqrt{2})$$



Kernels, Dot Products, and Mercer's Theorem

$$x_{mn} \equiv \left(\boldsymbol{x}^m\right)_n$$

$$\boldsymbol{\Phi}^T(\boldsymbol{x}^i)\boldsymbol{\Phi}(\boldsymbol{x}^j) = x_{i1}^2\, x_{j1}^2 + x_{i2}^2\, x_{j2}^2 + 2\, x_{i1}\, x_{i2}\, x_{j1}\, x_{j2} =$$

$$(x_{i1}\, x_{j1} + x_{i2}\, x_{j2})^2 = \left(\left(\boldsymbol{x}^i\right)^T \boldsymbol{x}^j\right)^2$$

Another example:

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \left(x_1^3\,,\ x_2^3\,,\ \sqrt{3}\, x_1^2\, x_2\,,\ \sqrt{3}\, x_2^2\, x_1\,,\right.$$

$$\left.\sqrt{3}\, x_1^2\,,\ \sqrt{3}\, x_2^2\,,\ \sqrt{6}\, x_1\, x_2\,,\ \sqrt{3}\, x_1\,,\ \sqrt{3}\, x_2\right)$$

$$\boldsymbol{\Phi}^T(\boldsymbol{x}^i)\boldsymbol{\Phi}(\boldsymbol{x}^j) =$$

$$x_{i1}^3\, x_{j1}^3 + x_{i2}^3\, x_{j2}^3 +$$

$$3\, x_{i1}^2\, x_{i2}\, x_{j1}^2\, x_{j2} + 3\, x_{i2}^2\, x_{i1}\, x_{j2}^2\, x_{j1} +$$

$$3\, x_{i1}^2\, x_{j1}^2 + 3\, x_{i2}^2\, x_{j2}^2 +$$

$$6\, x_{i1}\, x_{i2}\, x_{j1}\, x_{j2} + 3\, x_{i1}\, x_{j1} + 3\, x_{i2} x_{j2} =$$

$$= \left(\left(\boldsymbol{x}^i\right)^T \boldsymbol{x}^j + 1\right)^3 - 1$$

Kernels, Dot Products, and Mercer's Theorem

kernel k: a function which produces a scalar out of two vectors

$$k\left(\boldsymbol{x}^i, \boldsymbol{x}^j\right) = \left(\left(\boldsymbol{x}^i\right)^T \boldsymbol{x}^j + 1\right)^3$$

Certain kernels represent the mapping of vectors into a feature space and a dot product in this space.

The following theorem characterizes functions which build a dot product in some space.

Kernels, Dot Products, and Mercer's Theorem

Theorem 1 (Mercer)
Let the kernel k be symmetric and from $L_2(X \times X)$ defining a Hilbert-Schmidt operator

$$T_k(f)(\boldsymbol{x}) \;=\; \int_X k(\boldsymbol{x}, \boldsymbol{x}')\; f(\boldsymbol{x}')\; d\boldsymbol{x}' \;.$$

If T_k is positive definite, i.e. for all $f \in L_2(X)$

$$\int_{X \times X} k(\boldsymbol{x}, \boldsymbol{x}')\; f(\boldsymbol{x})\; f(\boldsymbol{x}')\; d\boldsymbol{x}\; d\boldsymbol{x}' \;\geq\; 0 \;,$$

then T_k has eigenvalues $\lambda_j \;\geq\; 0$ with associated eigenfunctions $\psi_j \in L_2(X)$. Further

$$(\lambda_1, \lambda_2, \ldots) \in \ell_1$$
$$k(\boldsymbol{x}, \boldsymbol{x}') \;=\; \sum_j \lambda_j\; \psi_j(\boldsymbol{x})\; \psi_j(\boldsymbol{x}') \;,$$

where ℓ_1 is the space of vectors with finite one-norm and the last sum converges absolutely and uniformly for almost all $\boldsymbol{x}$ and $\boldsymbol{x}'$.

Kernels, Dot Products, and Mercer's Theorem

The sum may be an infinite sum for which the eigenvalues converge to zero. In this case the feature space is an infinite dimensional space.

„for almost all“ means „except for a set with zero measure“: single points may lead to an absolute and uniform convergence

That the convergence is „absolutely and uniformly“ is important because the sum can be resorted and derivative and sum can be exchanged.

If X is a compact interval $[a, b]$ and k is continuous then positive definiteness of T_k is equivalent to positive definiteness of k. A kernel k is *positive semi-definite* if for all l, all $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^l$, and all $\alpha_i, 1 \leq i \leq l$

$$\sum_{i,j=1,1}^{l,l} \alpha_i \; \alpha_j \; k(\boldsymbol{x}^i, \boldsymbol{x}^j) \; \geq \; 0$$

Using the gram matrix $\boldsymbol{K}$ with $K_{ij} = k(\boldsymbol{x}^i, \boldsymbol{x}^j)$ and the vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_l)$ this is

$$\boldsymbol{\alpha}^T \; \boldsymbol{K} \; \boldsymbol{\alpha} \; \geq \; 0$$

Reproducing Kernel Hilbert Space

Reproducing kernel Hilbert spaces (RKHSs) are an important theoretical tool for proofing properties of kernel methods.

X is a set and H a Hilbert space of complex-valued functions on X.

H is a reproducing kernel Hilbert space if every linear evaluation map

$$L_x : f \mapsto f(x)$$

from H to the complex numbers is continuous for any x in X.

Reproducing Kernel Hilbert Space

The next theorem is essential to show that every Hilbert space uniquely definites a kernel (from Hilbert space to kernel):

Theorem 1 (Riesz representation theorem)
Let H^* denote H's dual space, consisting of all continuous linear functionals from H into complex numbers $\mathbb{C}$. If x is an element of H, then the function ϕ_x defined by

$$\phi_x(y) = \langle y, x \rangle \quad \forall y \in H \ ,$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product of the Hilbert space, is an element of H^*. The Riesz representation theorem states that every element of H^* can be written uniquely in this form, that is the mapping

$$\Phi : \ H \to H^*, \quad \Phi(x) = \phi_x$$

is an isometric (anti-) isomorphism.

Reproducing Kernel Hilbert Space

Riesz theorem for kernel k: for every x in X there exists an element k_x of H, the „point-evaluation functional“ at the point x, with the property:

$$f(x) = \langle f, k_x \rangle \quad \forall f \in H$$

We define the kernel as a function $k : X \times X \to \mathbb{C}$

$$k(x, y) \stackrel{\text{def}}{=} k_x(y)$$

k is called the reproducing kernel for the Hilbert space H.

k is completely determined by H because of Riesz theorem.

H is called „**reproducing kernel Hilbert space**“ (RKHS).

Reproducing Kernel Hilbert Space

The next theorem states that every symmetric, positive definite kernel uniquely defines an RKHS (from kernel to Hilbert space):

Theorem 1 (Moore-Aronszajn)
Suppose k is a symmetric, positive definite kernel on a set X. Then there is a unique Hilbert space of functions on X for which k is a reproducing kernel.

Proof.
Define, for all x in X, $k_x = k(x, \cdot)$. Let H_0 be the linear span of $\{k_x : x \in X\}$. Define an inner product on H_0 by

$$\left\langle \sum_{j=1}^{n} b_j k_{y_j}, \sum_{i=1}^{m} a_i k_{x_i} \right\rangle = \sum_{i=1}^{m} \sum_{j=1}^{n} \overline{a_i} b_j k(y_j, x_i) .$$

The symmetry of this inner product follows from the symmetry of k and the non-degeneracy follows from the fact that k is positive definite.
Let H be the completion of H_0 with respect to this inner product. Then H consists of functions of the form

$$f(x) = \sum_{i=1}^{\infty} a_i k_{x_i}(x) ,$$

where $\sum_{i=1}^{\infty} a_i^2 k(x_i, x_i) < \infty$. The fact that the above sum converges for every x follows from the Cauchy-Schwartz inequality.

We confirm the RKHS property:

$$\langle f, k_x \rangle = \left\langle \sum_{i=1}^{\infty} a_i k_{x_i}, k_x \right\rangle = \sum_{i=1}^{\infty} a_i k(x_i, x) = f(x) .$$

Therefore we have

$$f(x) = \sum_{i=1}^{\infty} a_i k(x_i, x) .$$

which is for example the model class of support vector machines.
To prove uniqueness, let G be another Hilbert space of functions for which k is a reproducing kernel. For any x and y in X, The RKHS property implies that

$$\langle k_x, k_y \rangle_H = k(x, y) = \langle k_x, k_y \rangle_G .$$

By linearity,

$$\langle \cdot, \cdot \rangle_H = \langle \cdot, \cdot \rangle_G$$

on the span of $\{k_x : x \in X\}$. Then $G = H$ by the uniqueness of the completion.
End Proof.

Reproducing Kernel Hilbert Space

Properties of the reproducing kernel and the RKHS:

- reproducing property: $k(x,y) = \langle k(x,\cdot), k(y,\cdot) \rangle$
- orthonormal sequences, kernel expansion: If $\{\phi_k\}_{k=1}^{\infty}$ is an orthonormal sequence and the closure of its span is H, then

$$k(x,y) = \sum_{k=1}^{\infty} \phi_k(x)\,\phi_k(y)$$

- Moore-Aronszajn Theorem: every symmetric, positive definite kernel defines a unique reproducing kernel Hilbert space.

For machine learning, the model class

$$f(x) = \sum_{i=1}^{\infty} a_i k_{x_i}(x) = \sum_{i=1}^{\infty} a_i k(x_i, x)$$

is of importance and can be represented by the RKHS.