Abschätzung des Generalisierungsfehlers und Modellauswahl

Volker Tresp

Empirischer Modellvergleich

Modellvergleich

- Sei

$$f(\mathbf{x}, \mathcal{M}_i)$$

  die Vorhersage eines Modells $\mathcal{M}_i$ für Eingang $\mathbf{x}$

- Beispiel: $\mathcal{M}_1$ bezeichnet ein Neuronales Netz mit Parametervektor $\mathbf{w}$ und $\mathcal{M}_2$ ist lineare Regression mit Parametervektor $\mathbf{v}$

- Gegeben zwei Modellansätze $\mathcal{M}_i$ und $\mathcal{M}_j$, so wollen wir nachweisen, dass $\mathcal{M}_i$ bessere Performanz besitzt als $\mathcal{M}_j$, und zwar in Bezug auf eine Verlustfunktion

$$L[y, f(\mathbf{x}, \mathcal{M}_i)]$$

- Beispiel:

$$L[y, f(\mathbf{x}, \mathcal{M}_i)] = (y - f(\mathbf{x}, \mathcal{M}_i))^2$$

Generalisierungsfehler

- Von Interesse ist der Generalisierungsfehler (erwartete Verlust, Risikofunktional)

$$R(\mathcal{M}_i) = E_{P(\mathbf{x},y)} L[y, f(\mathbf{x}, \mathcal{M}_i)] = \int L[y, f(\mathbf{x}, \mathcal{M}_i)] P(\mathbf{x}, y) \; d\mathbf{x}dy$$

- $P(\mathbf{x}, y)$ ist fest aber unbekannt

Empirisches Risiko über Testdaten

- Ein Schätzer des Generalisierungsfehlers ist

$$R(\mathcal{M}_i) \approx J^{\mathsf{Test}}(M_i) = \frac{1}{T}\sum_{i=1}^{T} L[y_i, f(\mathbf{x}_i, \mathcal{M}_i)]$$

  also einfach der mittlere Fehler (Verlust) auf Testdaten $\mathsf{Test} = \{\mathbf{x}_i, y_i\}_{i=1}^{T}$

- Dieser Schätzer ist unverzerrt (*unbiased*, erwartungstreu)

- Ein Schätzer für die Varianz des Schätzers ist

$$\widehat{Var}(J^{\mathsf{Test}}(M_i)) = \frac{1}{T(T-1)}\sum_{i=1}^{T}(y_i - f(\mathbf{x}, \mathcal{M}_i))^2$$

  mit $\mathbf{x}_i, y_i \in \mathsf{Test}$

Empirisches Risiko über Trainingsdaten

- Man könnte das Risiko durch das empirische Risiko auf den Trainingsdaten abschätzen,

$$R(\mathcal{M}_i) \approx J^{\text{Train}}(\mathcal{M}_i) = \frac{1}{N} \sum_{i=1}^{N} L[y_i, f(\mathbf{x}_i, \mathcal{M}_i)]$$

  mit $\mathbf{x}_i, y_i \in$ Test

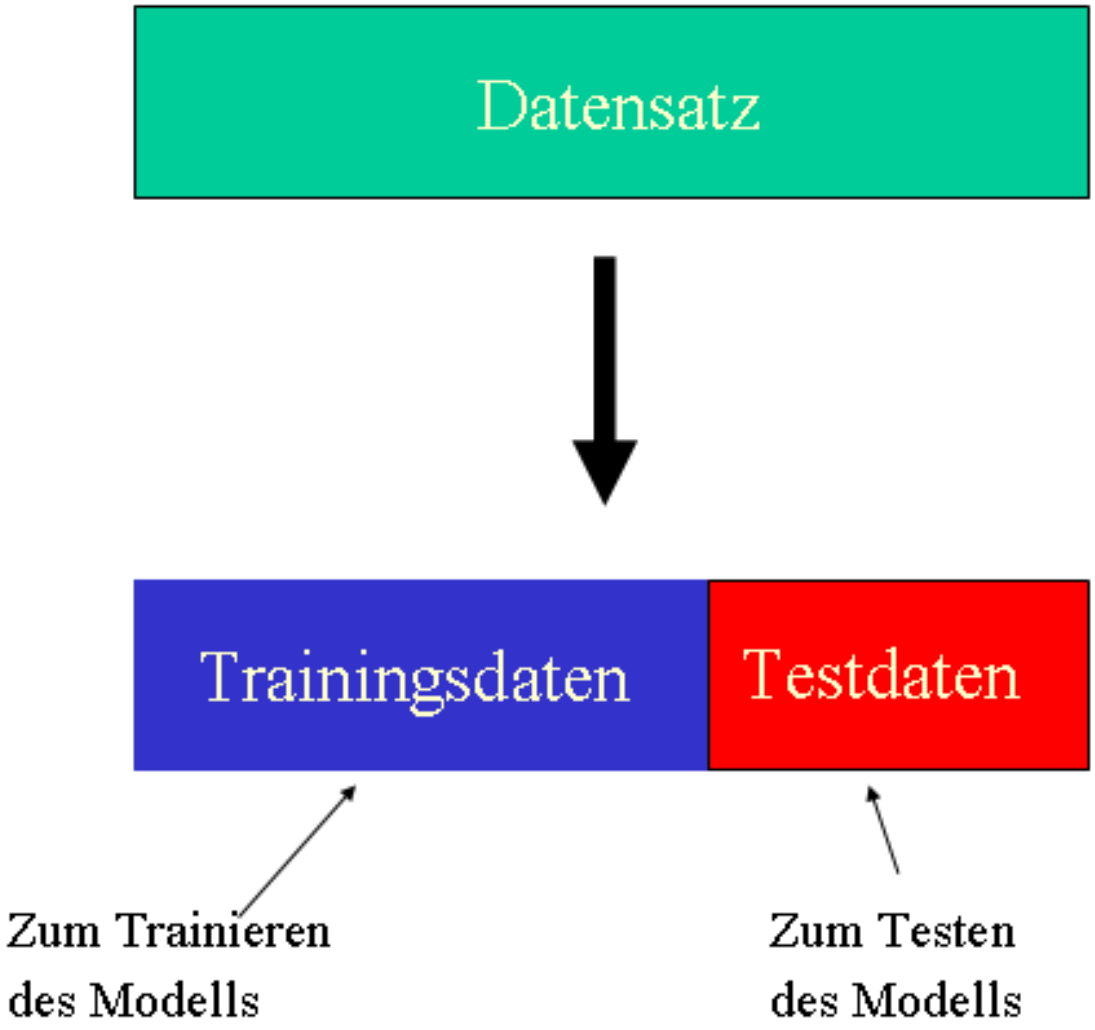
- Jedoch ist dieser Schätzer verzerrt (*biased*), da die Parameter basierend auf den Trainingsdaten optimiert wurden

- Ziel der theoretischen Ansätze ist genau die Abschätzung und Analyse von

$$R(\mathcal{M}_i) - J^{\text{Train}}(\mathcal{M}_i)$$

Modellselektion über Testdaten

- Man teilt die Daten in Trainingsdaten und Testdaten auf und trainiert das Modell nur auf den Trainingsdaten
- Man wählt das Modell aus, welches auf den Testdaten die beste Performanz besitzt


Datensatz
Trainingsdaten
Testdaten
Zum Trainieren des Modells
Zum Testen des Modells

Kreuzvalidierung

- Bei der Kreuzvalidierung können alle vorhandenen Daten verwandt werden, um das Risiko abzuschätzen
- Betrachten wir die K-fache Kreuzvalidierung; typische Zahlen sind $K = 5$ oder $K = 10$
- Die Daten werden in K gleichgroße Gruppen partitioniert
- Für $k = 1, \ldots, K$: Die $k-$te Menge ist der Testdatensatz und die übrigen Datensätze agieren als Trainingsdaten

Empirisches Risiko über Kreuzvalidierung

- So erhält man für jeden Modellansatz i nicht nur einen sondern K Testfehler

$$J_k^{\text{Test}}(\mathcal{M}_i), \quad k = 1, \ldots, K$$

- Wir können nun schätzen mit dem mittleren Testfehler

$$R(\mathcal{M}_i) \approx \mathbf{mean}(\mathcal{M}_i) = \frac{1}{K} \sum_{k=1}^{K} J_k^{\text{Test}}(M_i)$$

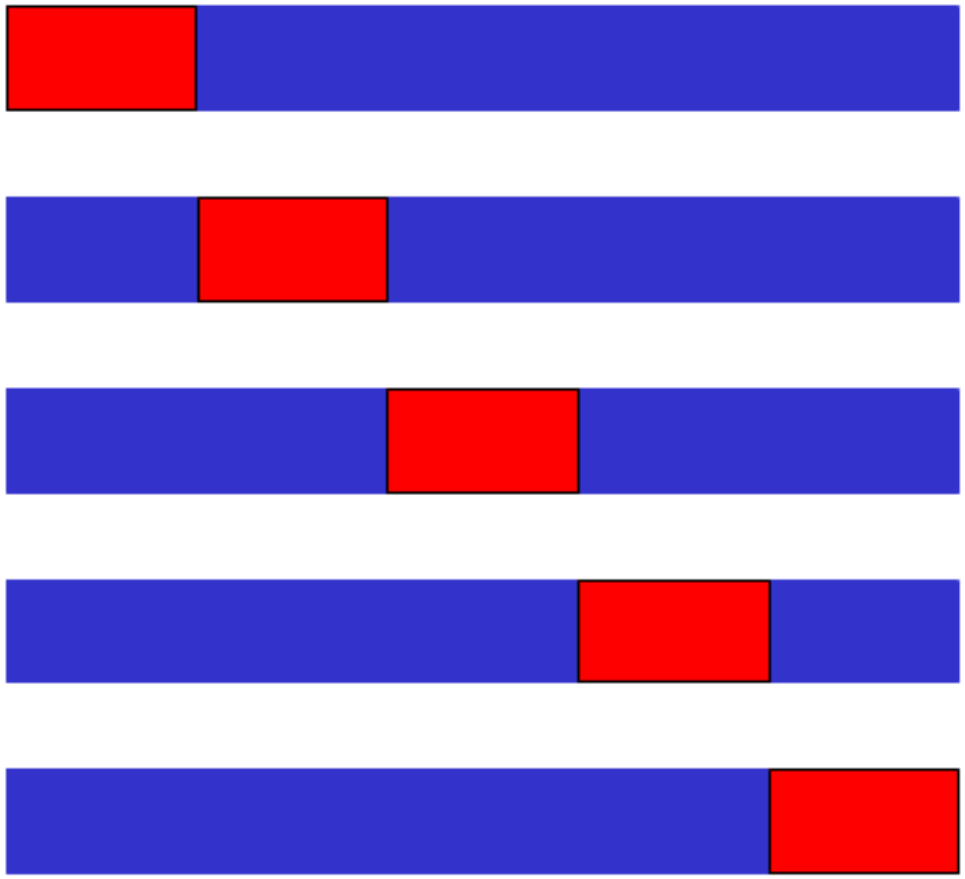
- Die Varianz des Schätzers kann man schätzen mit

$$\widehat{Var}(\mathbf{mean}(\mathcal{M}_i)) = \frac{1}{K(K-1)} \sum_{k=1}^{K} (J_k^{\text{Test}}(\mathcal{M}_i) - \mathbf{mean}(\mathcal{M}_i))^2$$

Modellselektion mit Kreuzvalidierung

- Man würde Modell $\mathcal{M}_i$ als besser als $\mathcal{M}_j$ einstufen, wenn sich die Standardabweichungen nicht überlappen, das heißt, falls

$$\mathbf{mean}(\mathcal{M}_i) + \widehat{Var}(\mathbf{mean}(\mathcal{M}_i)) < \mathbf{mean}(\mathcal{M}_j) - \widehat{Var}(\mathbf{mean}(\mathcal{M}_j))$$



5-Fache
Kreuzvalidierung:
Blau: Trainingsdaten
Rot: Testdaten

Gepaarte Tests

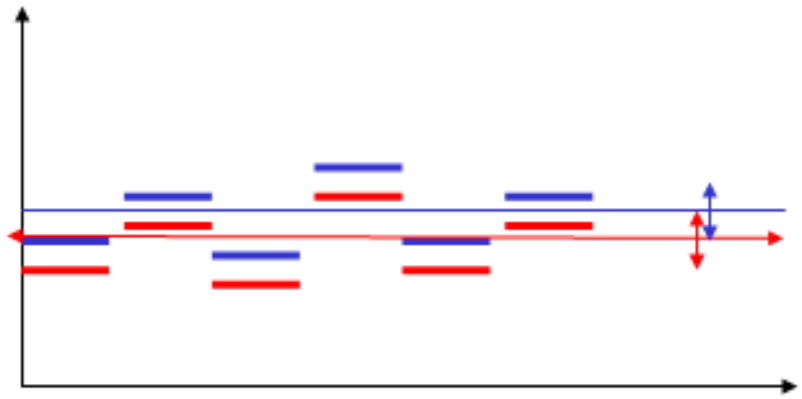
- Wenn man sehr wenige Daten hat, ist die Kreuzvalidierung manchmal nicht scharf genug
- Die Grundidee: nehmen wir an $K = 10$; wenn nun $\mathcal{M}_i$ in neun der zehn Tests besser abschneidet als $\mathcal{M}_j$, dann spricht dies stark für $\mathcal{M}_i$
- Man berechnet die mittlere Differenz der Verfahren

$$\mathsf{MeanDiff}_{i,j} = \frac{1}{K} \sum_{k=1}^{K} J_k^{\mathsf{Test}}(\mathcal{M}_i) - J_k^{\mathsf{Test}}(\mathcal{M}_j)$$

  und analysiert, ob diese Differenz signifikant positiv (oder negativ ist); eine sorgfältigere Analyse führt zum gepaarten T-Test (paired t-test)

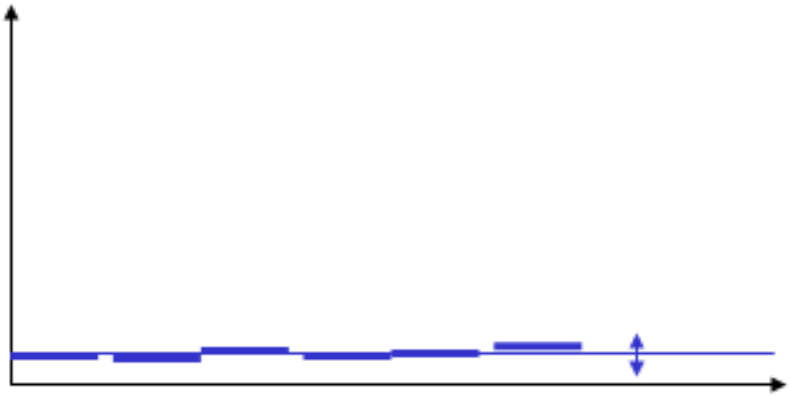
Testfehler für Schätzung eines Mittelwertes

Testfehler



- Basierend auf Mittelwert und Varianz kann nicht entschieden werde, dass Modellansatz 1(blau) signifikant besser ist als Modellansatz 2

Differenz der Testfehler



- Untersucht man jedoch die Differenz der Performanz ist die bessere Performanz von Modellansatz 1 (blau) signifikant

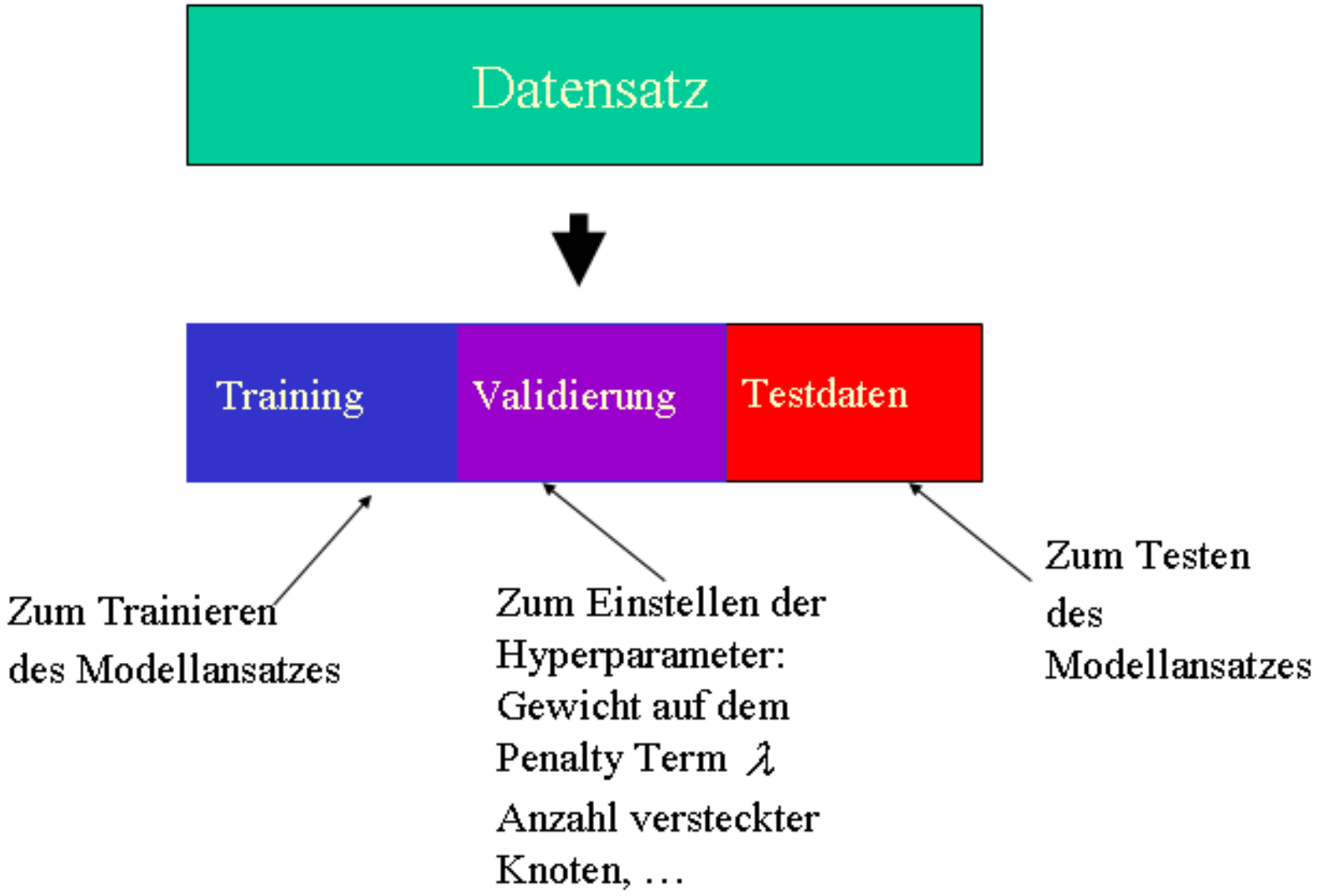
Empirische Einstellung der Hyperparameter

Hyperparameter

- Neben den eigentlichen Parametern, die durch den Lernprozess bestimmt werden, gibt es auch sogenannte Hyperparameter: typischerweise sind dies die Gewichtungen auf den Straftermen λ, die Anzahl der versteckten Knoten eines Neuronalen Netzes, ...
- Bayessche Verfahren haben hier einen Vorteil, da Hyperparameter einfach nur weitere Parameter im Modell darstellen, über die integriert werden muss
- Die meisten anderen Verfahren tun sich schwerer mit einer prinzipiellen Bestimmung der Hyperparameter; eine universelle Lösung stellt die empirische Bestimmung dar

Hyperparameter(2)

- Die Idee ist eine drei-Einteilung der Daten in Trainings-, Validierungs-, und Testdaten
  - Das Modell wird auf den Trainingsdaten mit verschiedenen Werten der Hyperparameter trainiert
  - Es wird das Modell mit den entsprechenden Hyperparametern ausgewählt, welches auf den Validierungsdaten die beste Performanz gezeigt hat
  - Es wird der Testfehlers dieses optimierten Modells berechnet
- Ähnlich wie bei der Modellauswahl, kann natürlich auch die Bestimmung der Hyperparameter über Kreuzvalidierung erfolgen


Datensatz
Training
Validierung
Testdaten
Zum Trainieren des Modellansatzes
Zum Einstellen der Hyperparameter: Gewicht auf dem Penalty Term λ Anzahl versteckter Knoten, …
Zum Testen des Modellansatzes

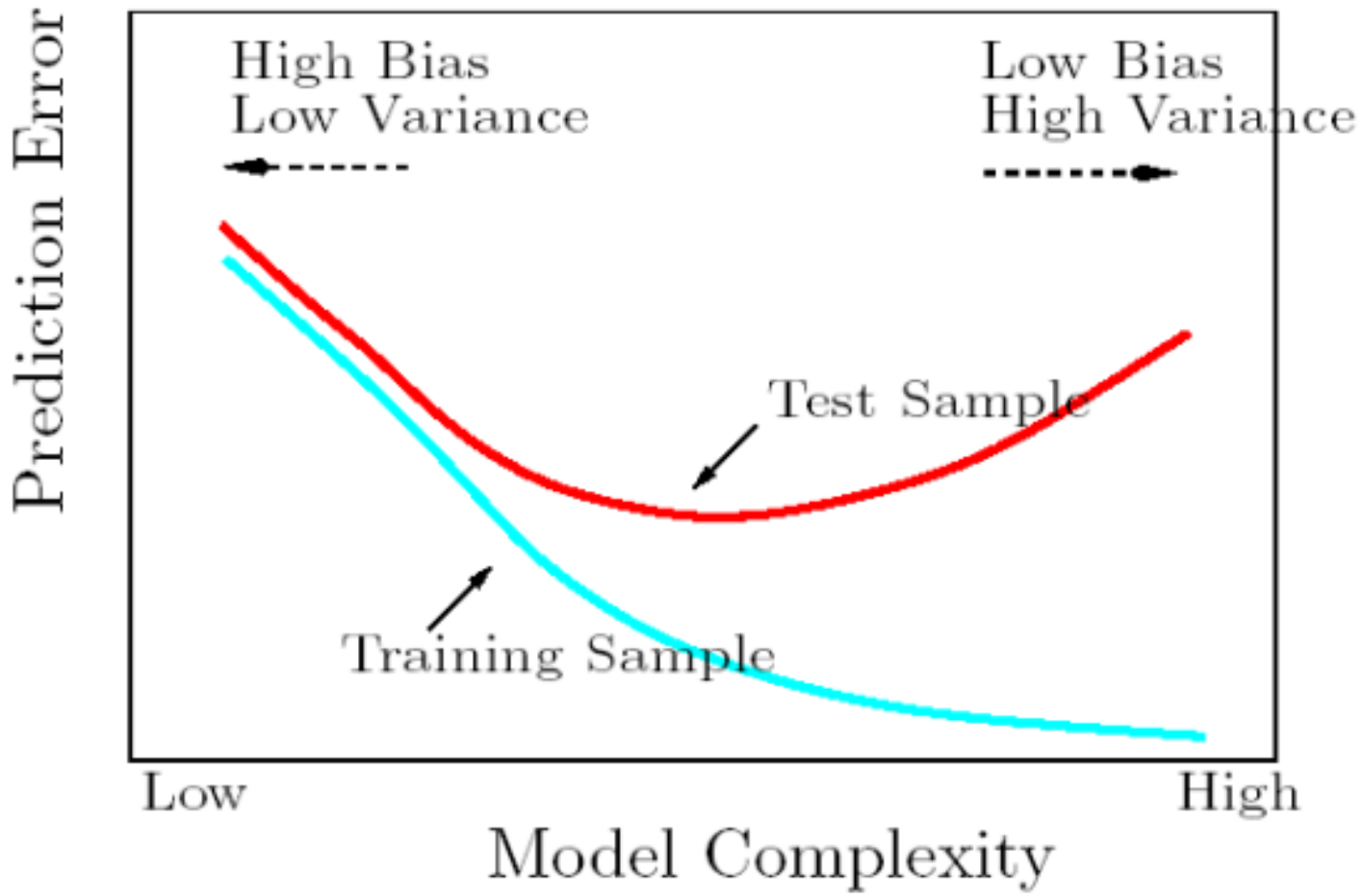
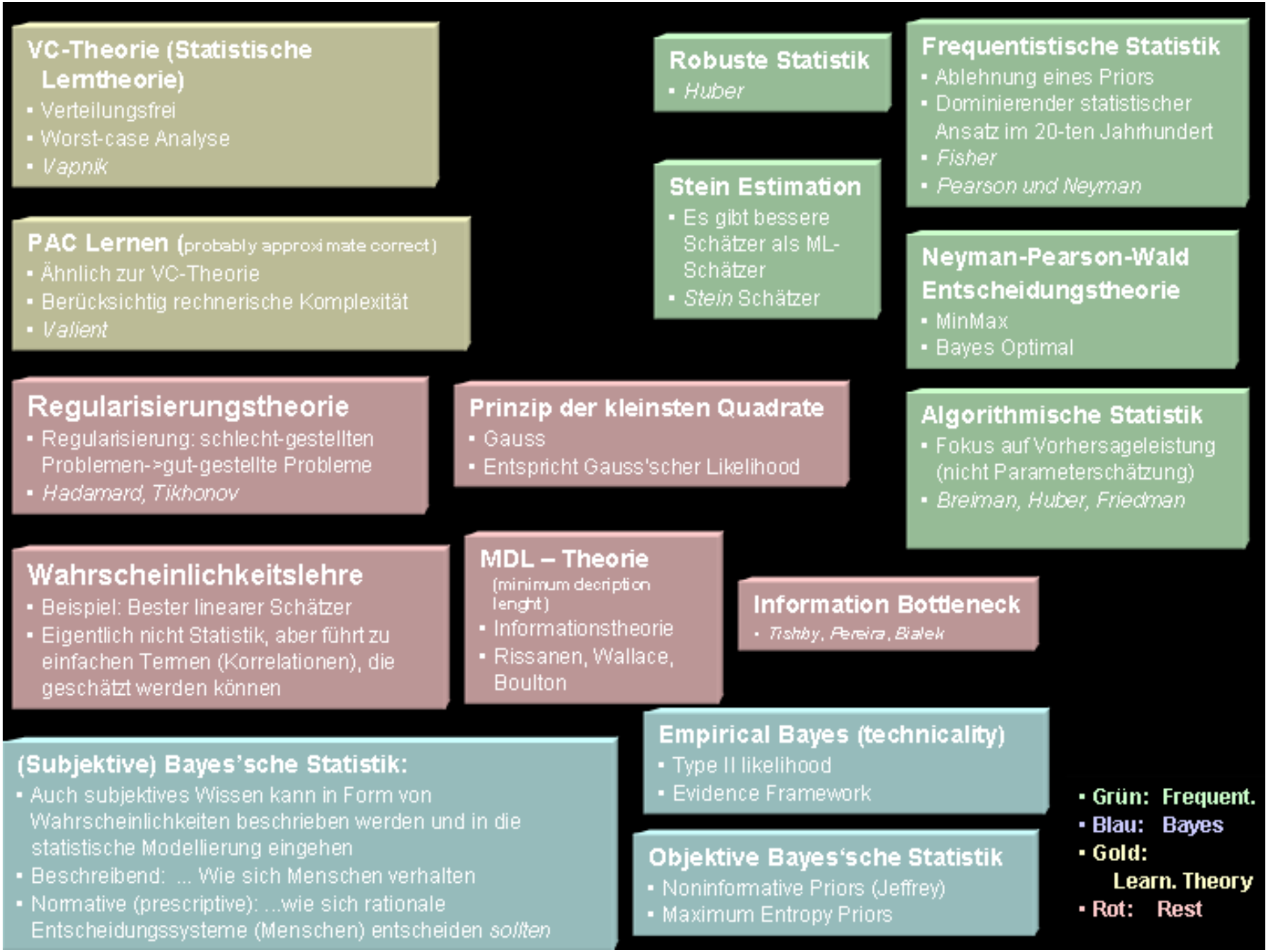


Figure 7.1: *Behavior of test sample and training sample error as the model complexity is varied.*

Lerntheorien und theoretische Abschätzungen des Generalisierungsfehlers

Überblick: Statistische Theorien und Lerntheorien


VC-Theorie (Statistische Lerntheorie)
• Verteilungsfrei
• Worst-case Analyse
• Vapnik
Robuste Statistik
• Huber
Frequentistische Statistik
• Ablehnung eines Priors
• Dominierender statistischer Ansatz im 20-ten Jahrhundert
• Fisher
• Pearson und Neyman
Stein Estimation
• Es gibt bessere Schätzer als ML-Schätzer
• Stein Schätzer
PAC Lernen (probably approximate correct)
• Ähnlich zur VC-Theorie
• Berücksichtig rechnerische Komplexität
• Valient
Neyman-Pearson-Wald Entscheidungstheorie
• MinMax
• Bayes Optimal
Regularisierungstheorie
• Regularisierung: schlecht-gestellten Problemen->gut-gestellte Probleme
• Hadamard, Tikhonov
Prinzip der kleinsten Quadrate
• Gauss
• Entspricht Gauss'scher Likelihood
Algorithmische Statistik
• Fokus auf Vorhersageleistung (nicht Parameterschätzung)
• Breiman, Huber, Friedman
Wahrscheinlichkeitslehre
• Beispiel: Bester linearer Schätzer
• Eigentlich nicht Statistik, aber führt zu einfachen Termen (Korrelationen), die geschätzt werden können
MDL – Theorie (minimum decription lenght)
• Informationstheorie
• Rissanen, Wallace, Boulton
Information Bottleneck
• Tishby, Pereira, Bialek
Empirical Bayes (technicality)
• Type II likelihood
• Evidence Framework
(Subjektive) Bayes'sche Statistik:
• Auch subjektives Wissen kann in Form von Wahrscheinlichkeiten beschrieben werden und in die statistische Modellierung eingehen
• Beschreibend: ... Wie sich Menschen verhalten
• Normative (prescriptive): ...wie sich rationale Entscheidungssysteme (Menschen) entscheiden sollten
Objektive Bayes'sche Statistik
• Noninformative Priors (Jeffrey)
• Maximum Entropy Priors
• Grün: Frequent.
• Blau: Bayes
• Gold: Learn. Theory
• Rot: Rest

Lerntheorien

- A: Klassische Frequentistische Verfahren
  - C_p Statistik
  - Akaikes Informationskriterium (AIC)
- B: Bayessche Verfahren
  - Striktes Bayes: ich muss mich niemals entscheiden: Mitteln anstatt auswählen
  - Bayessche Modellauswahl, Bayesian Information Criterion (BIC)
- C: Moderne Frequentistische Verfahren
  - Minimum Description Length (MDL) Prinzip (Appendix)
  - Statistische Lerntheorie (Vapnik-Chervonenkis (VC) Theorie)
- **Wir werden evaluieren, wie diese Theorien die Differenz zwischen Trainingsfehler und Testfehler abschätzen!**

A: Klassische Frequentistische Verfahren

Frequentistischer Ansatz

- Über den Testfehler erhielten wir einen unverzerrten Schätzer für das Risiko für ein beliebiges Modell mit einem beliebigen Parametervektor
- Im hier diskutierten frequentistischen Ansatz gehen wir von einem *definierten Schätzer* aus
- Betrachten wir einen Trainingsdatensatz D der Größe N; man geht von unendlich vielen Datensätzen der gleichen Größe aus und interessiert sich für den Erwartungswert des Generalisierungsfehlers, wenn man einen definierten Schätzer verwendet. Diesen Erwartungswert schreiben wir

$$E_D(R)$$

1-Dimensionales Beispiel

- Betrachten wir zunächst den 1-dimensionalen Fall. Daten werden generiert nach

$$x_i = \mu + \epsilon_i$$

  ϵ_i ist unabhängiges Rauschen mit Rauschvarianz σ^2

- Wir nehmen einen regularisierten Schätzer an

$$\widehat{\mu} = \frac{1}{N + \lambda} \sum_{i=1}^{N} x_i$$

- Uns interessiert nun

$$R(\widehat{\mu}) = E_x(\widehat{\mu} - x)^2 = \int (\widehat{\mu} - x)^2 P(x) dx$$

  Da $P(x)$ unbekannt ist, können wir den Erwartungswert nicht berechnen

Bias-Varianz Zerlegung

- Wir können allerdings Ausdrücke über den Erwartungswert von $R(\mu)$ herleiten

$$E_D(R) = E_D E_x(\hat{\mu} - x)^2 = E_D[(\hat{\mu} - E_D(\hat{\mu})) - (E_D(\hat{\mu}) - \mu) - E_x(\mu - x)]^2$$

- Man kann nun die quadratische Form ausmultiplizieren und kann dann zeigen, dass alle Wechselwirkungsterme Null sind. Somit erhält man die Zerlegung

$$E_D(\hat{\mu} - x)^2 = \mathsf{Bias}^2 + \mathsf{Var} + \mathsf{Rest}$$

$$\mathsf{Rest} = E_x(x - \mu)^2$$

$$\mathsf{Bias} = E_D(\hat{\mu}) - \mu$$

$$\mathsf{Var} = E_D[\hat{\mu} - E_D(\hat{\mu})]^2$$

Bias-Varianz Zerlegung im Beispiel

- Im Beispiel:

$$\text{Rest} = \sigma^2$$

$$\text{Bias} = \frac{-\lambda\mu}{N+\lambda}$$

$$\text{Var} = \frac{N\sigma^2}{(N+\lambda)^2}$$

- Für $N \to \infty$ gehen Bias und Varianz gegen Null
- Für ein großes λ ist die Varianz klein aber der Bias^2 gross. Für ein kleines λ ist die Varianz groß aber der Bias^2 klein. Dies ist das berühmte Bias-Varianz Dilemma
- Für ein endliches N gibt es ein optimales λ

Komentare

- Mit $E_D(R)$ kann mean leichter arbeiten als mit R
- Der Bias lässt sich nur schwer abschätzen, da natürlich μ unbekannt ist; daher wendet man die Theorie in erster Linie auf unverzerrte Schätzer an
- Rest und Var kann man Abschätzen, indem man unverzerrte Schätzer für σ^2 verwendet

Bias-Varianz Zerlegung: Funktionswert

- Man kann die gleiche Zerlegung ebenso auf einen Funktionswert $f(\mathbf{x})$ anwenden

$$E_D E_y(\hat{f}(x) - y)^2 = \text{Bias}^2 + \text{Var} + \text{Rest}$$

$$\text{Rest} = E_y(y - f(x))^2$$

$$\text{Bias} = E_D(\hat{f}(x)) - f(x)$$

$$\text{Var} = E_D[\hat{f}(x) - E_D(\hat{f}(x))]^2$$

Bias-Varianz Zerlegung für den Generalisierungsfehler

- Man kann die gleiche Zerlegung ebenso auf vektorielle Größen und damit auch Funktionen anwenden. Mit dem quadratischen Fehler als Kostenfunktion erhalten wir für den erwarteten Generalisierungsfehler

$$E_D(R) = \text{Bias}^2 + \text{Var} + \text{Rest}$$

$$\text{Rest} = \int (f(x) - y)^2 P(x, y) dx dy$$

$$\text{Bias}^2 = \int (E_D(\hat{f}(x)) - f(x))^2 P(x) dx$$

$$\text{Var} = \int E_D[\hat{f}(x) - E_D(\hat{f}(x))]^2 P(x) dx$$

Anwendung auf Lineare Modelle

- Wir nehmen an, dass Daten generiert werden nach

$$y_i = \phi(\mathbf{x}_i)\mathbf{w} + \epsilon_i$$

wobei: ϵ_i ist unabhängiges Rauschen mit Varianz σ^2

- Das Modell hat die gleiche Form; ein unverzerrter Schätzer ist

$$\hat{\mathbf{w}} = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{y}$$

mit

$$\mathsf{Cov}(\mathbf{w}) = \sigma^2(\Phi^T\Phi)^{-1}$$

Weiterhin nehmen wir an, dass $P(\mathbf{x})$ durch die empirische Verteilung auf den Trainingsdaten approximiert werden kann

$$\mathsf{Var} = \frac{1}{N}\mathbf{trace}[X\mathsf{Cov}(\mathbf{w})X^T] = \frac{1}{N}\sigma^2\mathbf{trace}[X(X^TX)^{-1}X^T] = \frac{M}{N}\sigma^2$$

Mallot's C_P

- Die Lösung ist also

$$E_D(R) \approx \frac{M}{N} + \sigma^2 = \sigma^2 \frac{M+N}{N}$$

- Nun ist σ^2 in der Regel unbekannt. Eine unverzerrte Schätzung ist

$$\hat{\sigma}^2 = \frac{N}{N-M} J^{\mathbf{Train}}$$

- Somit kann man schreiben

$$E_D(R) \approx \frac{M+N}{N-M} J^{\mathbf{Train}} = J^{\mathbf{Train}} + 2\frac{M}{N}\hat{\sigma}^2$$

- Diese Abschätzung C_P-Statistik genannt

Akaikes Information Criterion (AIC)

- Man erhält für Modelle, bei denen die Log-Likelihood

$$l = \log L = \sum_{i=1}^{N} \log P(y_i|\mathbf{x}_i, \mathbf{w})$$

optimiert wird Akaike's *Information Criterion* (minimiere:)

$$AIC = 2\left(-\frac{1}{N}\log L + \frac{M}{N}\right)$$

- $\frac{M}{N}$ ist eine Schätzung der Differenz zwischen mittlerer Trainings-Loglikelihood und mittlerer Test-Loglikelihood.

Kommentare zu AIC

- AIC ist äquivalent zu C_p für Gauss Rauschen mit bekannter Rauschvarianz:

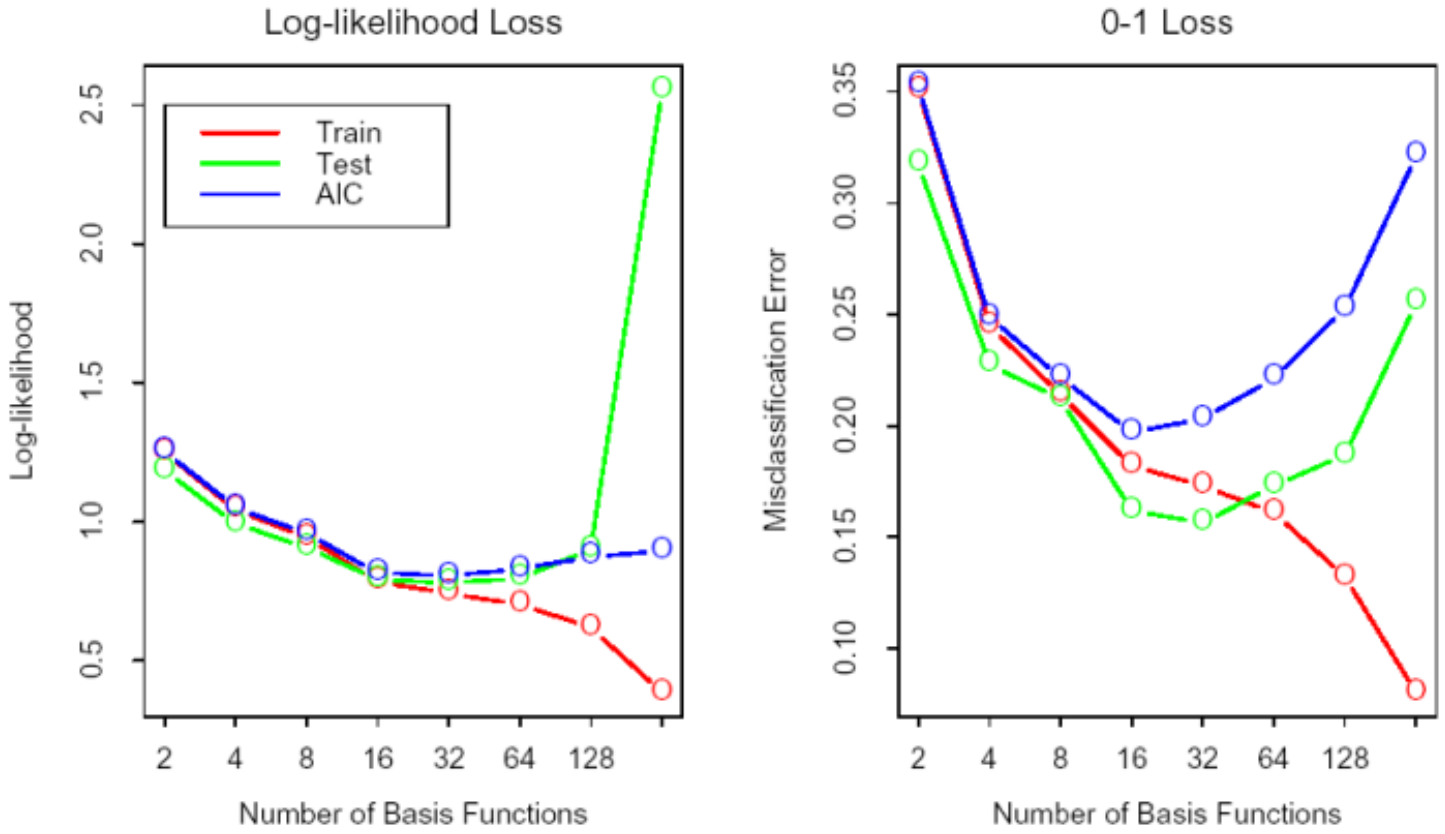
$$AIC = \frac{1}{\sigma^2}\frac{1}{N}\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i, \mathbf{w}))^2 + 2\frac{M}{N} = \frac{1}{\sigma^2}C_P$$

- Der Ausdruck

$$-AIC/2 = \frac{1}{N}\log L - \frac{M}{N}$$

schätzt die mittlere Log-Likelihood von neuen Daten ab, deren Eingangswerte mit den Trainingsdaten übereinstimmen (*in-sample*)

AIC für Likelihood Kostenfunktion und für 1/0 Kostenfunktion



Bayessche Ansätze

Die Bayessche Perspektive

- Zunächst benötigt der Bayessche Ansatz keine Modellauswahl
- Man spezifiziert a priori Wahrscheinlichkeiten für die Modelle,

$$P(\mathcal{M}_i)$$

- Eine a posteriori Vorhersage wird dann

$$P(y|\mathbf{x}) = \sum_i \int P(y|\mathbf{x}, \mathbf{w}, \mathcal{M}_i) P(\mathbf{w}|D, \mathcal{M}_i) P(\mathcal{M}_i|D) d\mathbf{w}$$

Generalisierungsfehler

- Für Modelle mit $y_i = f(\mathbf{x}_i) + \epsilon_i$ kann man als Generalisierungsfehler definieren

$$R = \int (y - E(y|x))^2 P(y|\mathbf{x}) P(x) dx dy$$

  und kann nach Abschätzung von $P(x)$ das Integral approximativ berechnen

- Allerdings würde man dennoch nach dem strengen Bayesschen Ansatz alle Modelle berücksichtigen und keine Auswahl treffen

- Die Abschätzung enthält natürlich die persönlichen *a priori* Annahmen des Modellierers

Bayessche Modellauswahl

- Der konsequente Bayessche Ansatz ist of unpraktisch und man betrachtet deshalb dennoch Modellauswahl
- A posteriori Modellwahrscheinlichkeit

$$P(\mathcal{M}|D) \propto P(\mathcal{M})P(D|\mathcal{M})$$

- Typischerweise nimmt man an, dass alle Modelle gleich-wahrscheinlich sind (a priori)
- Somit ist der entscheidende Term (marginal likelihood, evidence)

$$P(D|\mathcal{M}) = \int P(w|\mathcal{M})P(D|w)dw$$

Laplace Approximation der Marginal Likelihood

- $\log P(D|\mathcal{M})$ wird asymptotisch Gauß-förmig, allerdings ist das Integral nicht zu Eins normiert;
- Man behält nun nur die Terme, die von N abhängen. Dann erhält man

$$\log P(D|\mathcal{M}) \approx \log P(D|\hat{\mathbf{w}}_{MAP}, \mathcal{M}) - \frac{M}{2} \log N$$

Bayesian Information Criterion (BIC)

- BIC ist 2 Mal diesem Ausdruck (man ersetzt die MAP Parameterschätzung durch die ML-Parameterschätzung) (minimiere)

$$BIC = -2 \log L + M \log N$$

und die mittlere vorhergesagte Loglikelihood

$$-\frac{1}{2N} BIC = \frac{1}{N} \log L - \frac{M}{N} \frac{1}{2} \log N$$

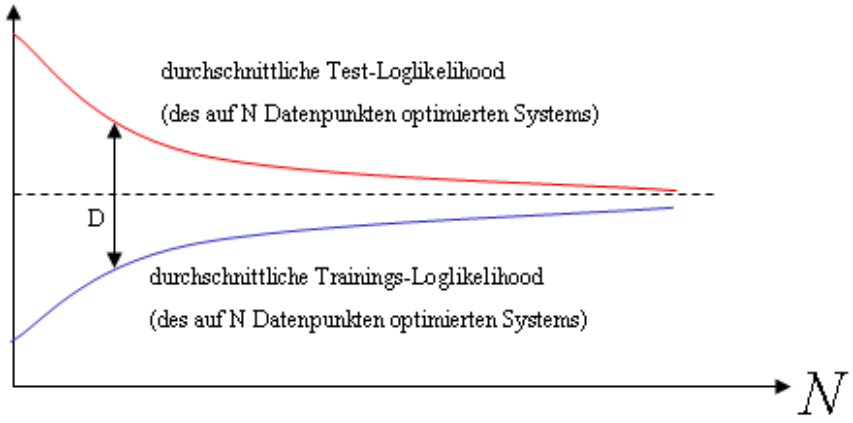
Vergleiche

$$-AIC/2 = \frac{1}{N} \log L - \frac{M}{N}$$

- $\frac{M}{N} \frac{1}{2} \log N$ ist eine Schätzung der Differenz zwischen mittlerer Trainings-Loglikelihood und mittlerer Test-Loglikelihood.

- Die BIC Korrektur ist um den Faktor $\frac{1}{2} \log N$ größer und verringert sich langsamer mit $(\log N)/N$ mit der Anzahl der Trainigsdaten

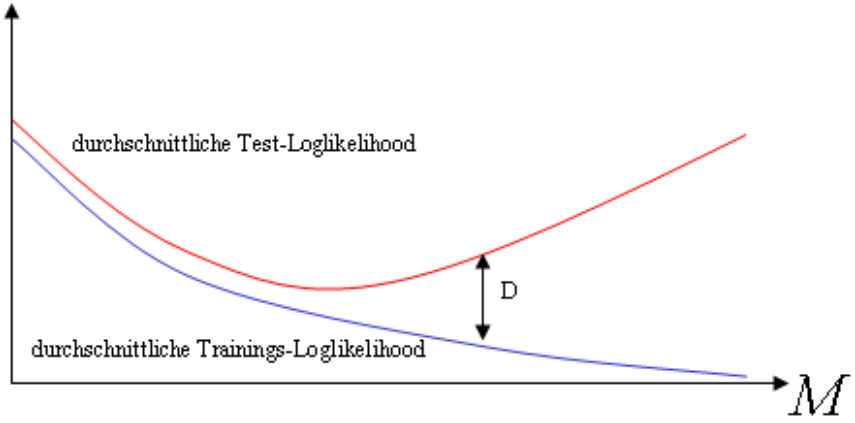
Vergleich: AIC und BIC



Schätzung von D:

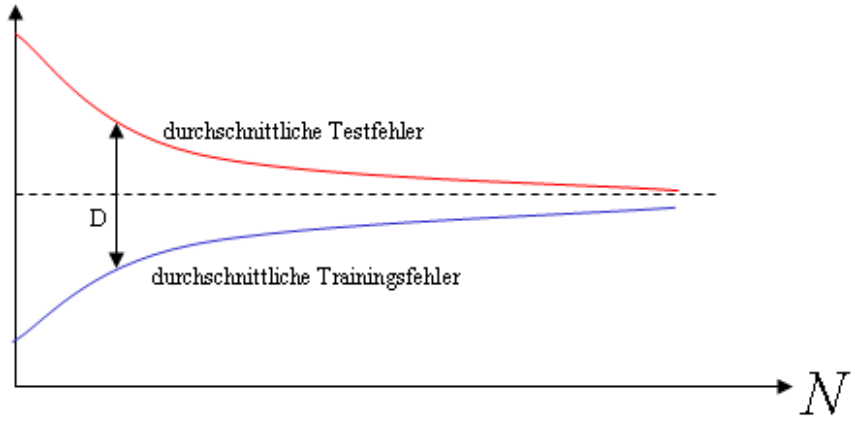
AIC: $\hat{D} = \frac{M}{N}$

BIC: $\hat{D} = \frac{M}{N}\frac{1}{2}\log N$



Mit einer zunehmenden Anzahl von Datenpunkten N verringert sich D (d.h. die Überanpassung), mit zunehmender Komplexität M erhöht sich D

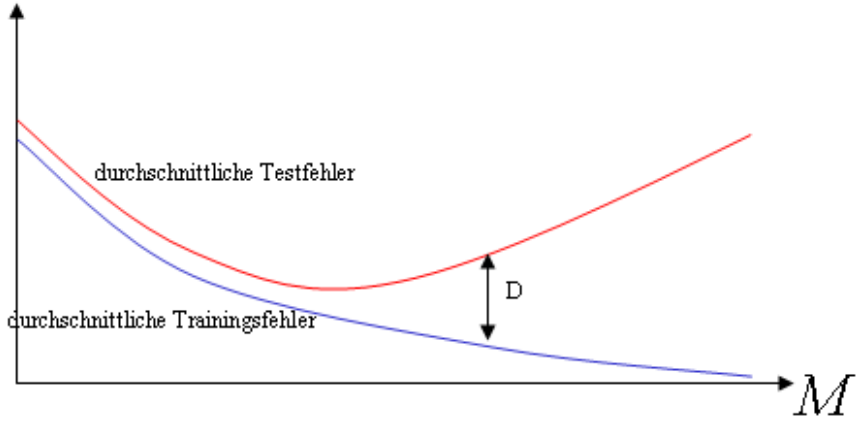
AIC und BIC: Spezialfall Gauß'sche Likelihood, quadratischer Fehler



Schätzung von D:

AIC: $\hat{D} = \frac{M}{N}\sigma^2$

BIC: $\hat{D} = \frac{M}{N}\frac{\sigma^2}{2}\log N$



Mit einer zunehmenden Anzahl von Datenpunkten N verringert sich D (d.h. die Überanpassung), mit zunehmender Komplexität M erhöht sich D

C: Moderne Frequentistische Verfahren

Minimum Description Length

- Basierend auf dem Konzept der algorithmischen Komplexität (Kolmogorov, Solomonoff, Chaitin)
- Auf Basis dieser Ideen: Rissanen (und Wallace, Boulton) führten das Prinzip der minimum description length (MDL) ein
- Unter einigen Vereinfachungen wird das MDL Kriterion identisch zum BIC Kriterion (siehe Appendix)

Statistische Lerntheorie

- Die Statistische Lerntheorie (Statistical Learning Theory, SLT) steht in der Tradition der russischen Mathematiker Andrey Kolmogorov und Valery Ivanovich Glivenko sowie des italienischen Mathematikers Francesco Paolo Cantelli
- Die Väter der SLT sind Vladimir Vapnik und Alexey Chervonenkis (daher auch der Begriff: VC-Theorie)
- Wird als Teil der *Computational Learning Theory* (COLT) angesehen, zu der auch das verwandte Konzept des PAC (*Probably approximately correct learning*) Lernens (Leslie Valiant) gehört

Basis der SLT

- Wie als erstes diskutiert, kann man über einen Testsatz oder über Kreuzvalidierung eine Aussage über den Generalisierungsfehler einer beliebigen Funktion machen
- Der frequentistische Ansatz beinhaltete einen Schätzer: d.h. man macht Aussagen über Funktionen, die z.B. den quadratischen Fehler im Training minimieren
- In der SLT betrachtet man Funktionen f aus einer Funktionenklasse F und interessiert sich für deren Generalisierungsfehler
- Die SLT macht eine Aussage über den maximalen Unterschied zwischen $J^{\mathbf{Train}}(f)$ und dem Generalisierungsfehler $R(f)$; dieser maximale Unterschied gilt dann für jede Funktion aus der Klasse, also nicht nur für die Funktion, die den Trainingsfehler minimiert
- Es wird nicht angenommen, dass die wahre Funktion h sich in F befindet; h kann beliebig "bösartig" sein (unstetig, ...)

Vapnik-Chervonenkis (VC-) Theorie (Statistical Learning Theory)

- Die VC-Theorie ist verteilungsfrei, das heißt sie macht keine Annahmen über eine zugrundeliegende Verteilung $P(\mathbf{x})$
- Die einzige wesentliche Annahme: Daten werden von einer *festen* Verteilung $P(\mathbf{x})$ generiert
- Zielgrößen werden von $h(\mathbf{x})$ generiert (im einfachsten Fall und hier ohne Rauschen und binär, $h(\mathbf{x}) \in \{-1, 1\}$); $h(\mathbf{x})$ muss nicht zur Funktionenklasse gehören

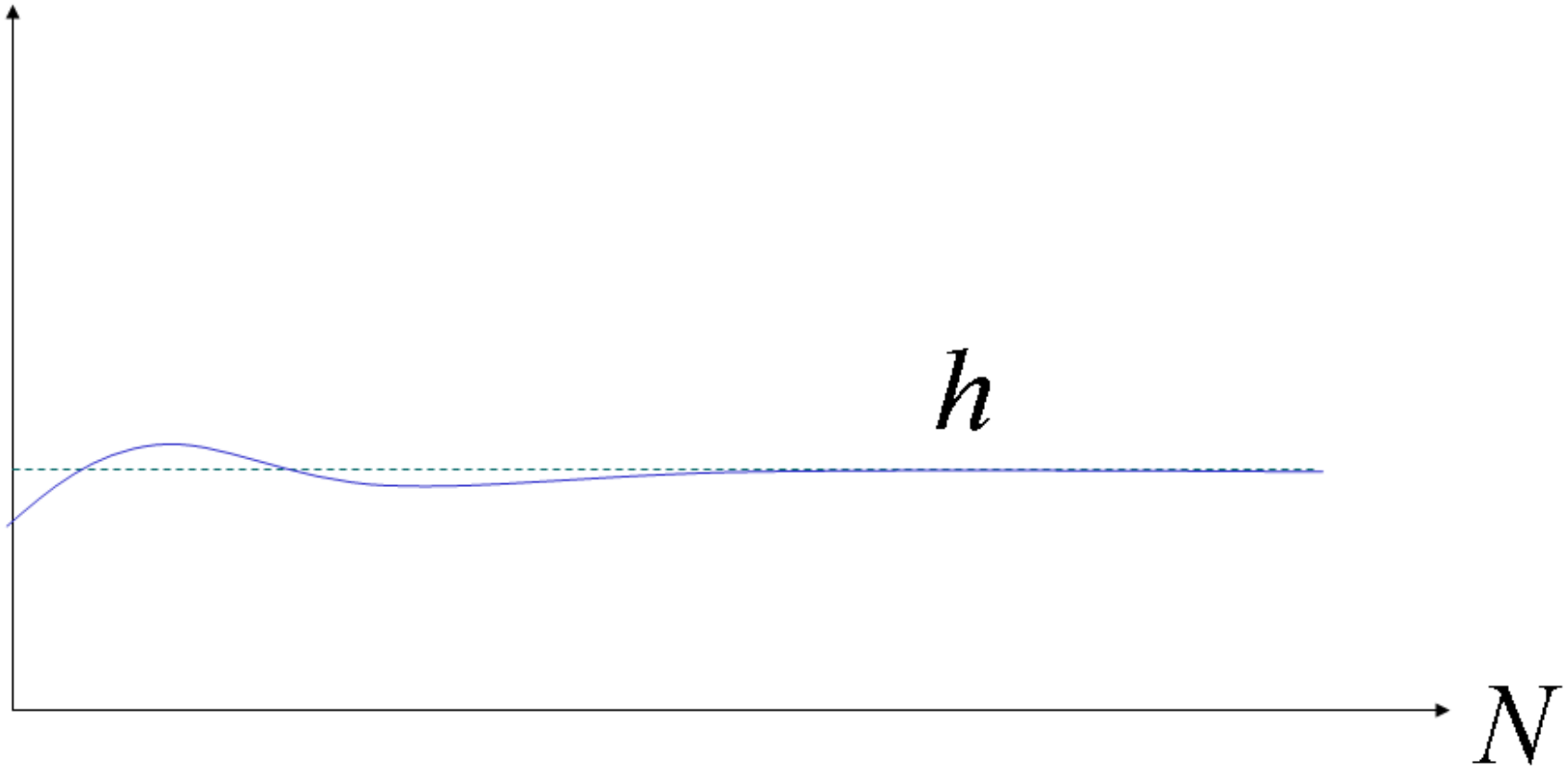
Konsistenz

- Man verlangt Konsistenz: mit $N \rightarrow \infty$ soll nach dem Kriterion die Funktion mit dem geringsten mittleren Trainingsfehler optimal sein
- Worst Case Analysis (MinMax) (one-sided uniform convergence)

$$\lim_{N \rightarrow \infty} P\left(\max_{f \in F} |R(f) - J^{\mathbf{Train}}(f)| > \epsilon\right) = 0, \forall \epsilon > 0$$

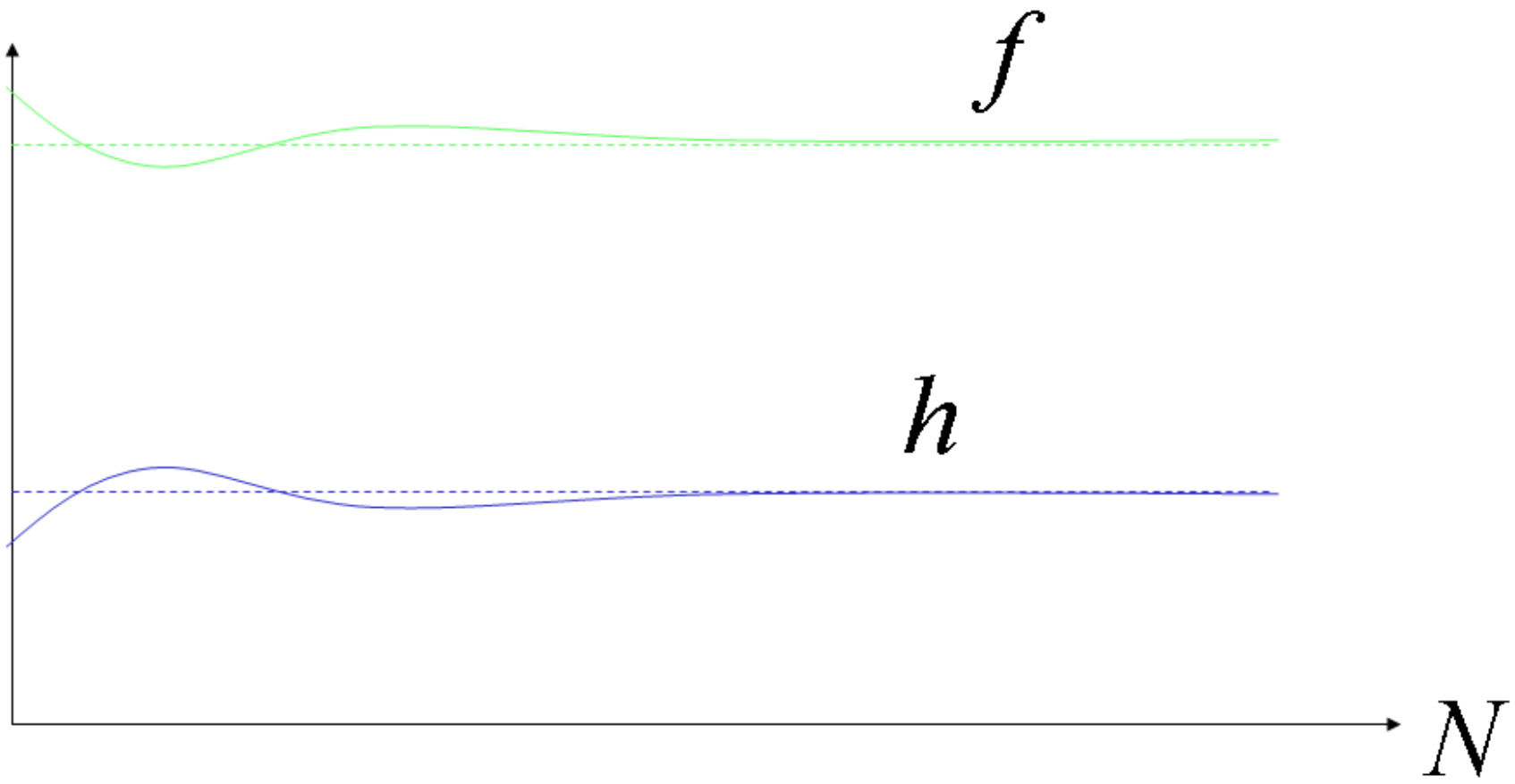
  (die gilt **für alle** $f : A \leq R(f) \leq B$ mit beliebigen Schranken A, B)
- In Worten: der Unterschied zwischen $J^{\mathbf{Train}}(f)$ und dem Generalisierungsfehler $R(f)$ geht gegen Null, für $N \rightarrow \infty$. Und dies gilt für alle Funktionen aus der Funktionenklasse. Beachte: die Differenz ist in der Regel am größten, für Funktionen, die im Trainingssatz ein kleines $J^{\mathbf{Train}}(f)$ besitzen
- Die Schranke ist unabhängig von den eigentlichen Trainingsdaten, solange diese nach $P(x)$ generiert wurden
- Die VC-Theorie berechnet nun verschiedene Schranken für den Unterschied zwischen $J^{\mathbf{Train}}(f)$ und dem Generalisierungsfehler $R(f)$

Generalisierungsfehler (gestrichelt) ----------
Mittl. Trainingsfehler (durchgezogen) ——————



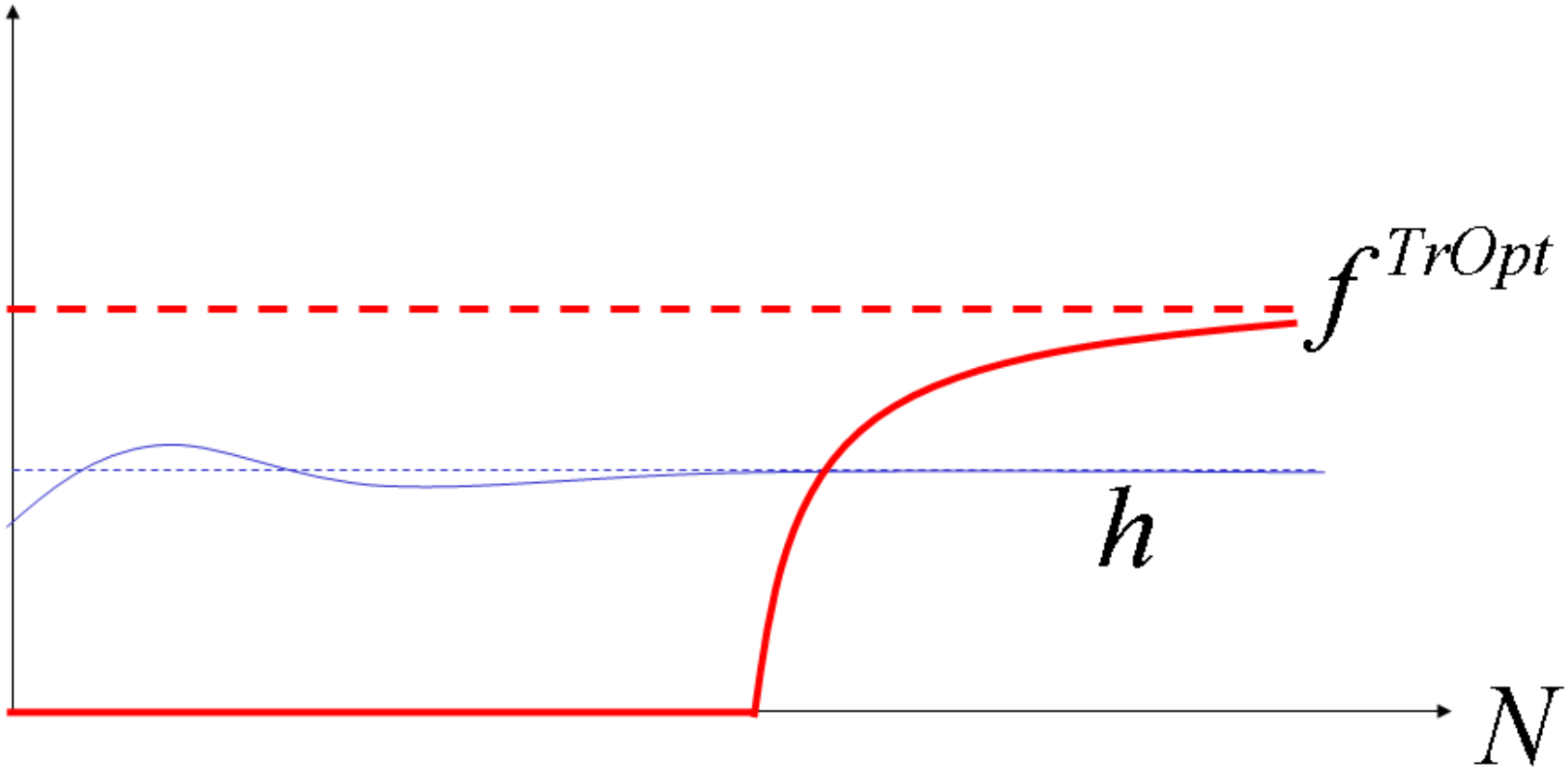
Die wahre Funktion *h* hat das kleinste erwartete Risiko; asymptotisch konvergiert der mittlerer Trainigsfehler zum erwartetes Risiko

Generalisierungsfehler (gestrichelt) ----------
Mittl. Trainingsfehler (durchgezogen) ——————



Für eine beliebige Funktion *f* ist das erwartete Risiko größer

Generalisierungsfehler (gestrichelt) ----------
Mittl. Trainingsfehler (durchgezogen) ——————



Diese Funktion *f* minimiert den Trainingsfehler;
der Unterschied zwischen mittlerem
Trainingsfehler und erwarteten Risiko ist groß

Selektion von Modellklassen

- Modellselektion: Angenommen, es ständen zwei Funktionsklassen F_1 und F_2 zur Verfügung. Seien f_1^{TrOpt} und f_2^{TrOpt} die Funktionen, die den Trainingsfehler für Funktionsklasse 1 bzw Funktionsklasse 2 minimieren. Man würde nun Funktionsklasse 1 auswählen, wenn

$$J^{\mathbf{Train}}(f_1^{TrOpt}) + \mathbf{Schranke}_1 < J^{\mathbf{Train}}(f_2^{TrOpt}) + \mathbf{Schranke}_2$$

- Die technische Herausforderung besteht nun darin, gute Schranken zu finden
- Vapnik argumentiert, dass nur eine *Worst-Case*-Analyse zu konsistenten nicht-trivialen Resultaten führt

VC-Dimension

- Wir versuchen jetzt anzudeuten, wie man sinnvolle Schranken erhalten kann
- Zunächst führen wir der Begriff "shatter" ein
- Eine Menge von Datenpunkten wird durch F ge-*shattered* (zerschmettert), wenn für jede beliebige Klassenzuordnung $\in \{1, -1\}$ es ein Mitglied der Klasse gibt, welches die Klassen richtig zuordnet
- Die VC-dimension einer Funktionsklasse dim_{VC} ist die maximale Anzahl von Datenpunkten, die ge-*shattered* werden kann, für mindestens eine Anordnung der Datenpunkte

VC-Dimension für die Klasse der linearen Klassifikatoren

- Für die Klasse der linearen Klassifikatoren ist die VC-Dimension gleich M, d.h. gleich der Anzahl der freien Parameter (Anzahl der Eingangsvariablen plus 1)

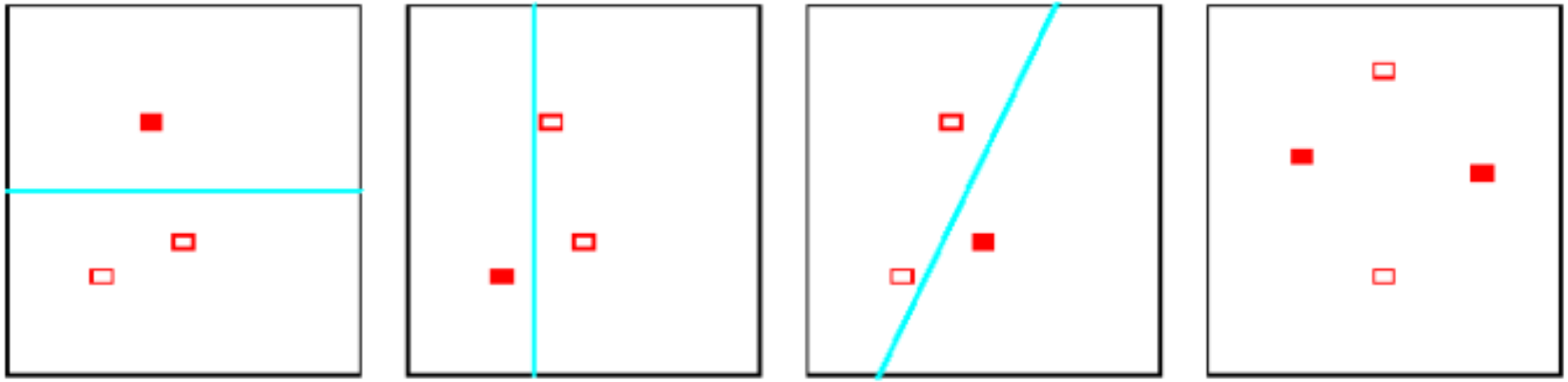


Illustration der VC-Dimension

- Vapnik argumentiert nun, dass man nur mit eingeschränkten Funktionsklassen etwas über den Generalisierungsfehler aussagen kann
- Gegeben sei ein Klassifikationsproblem mit zwei Eingängen und $N = 1000$: wenn ein linearer Klassifikator alle Muster richtig klassifiziert, dann hat er etwas "gelernt"
- Beachte: obwohl h beliebig "bösartig" sein kann, so müssen die Eingänge unabhängig nach $P(x)$ gezogen werden, d.h. die Wahrscheinlichkeit, dass die Daten nur durch Zufall so aussehen, als wenn sie durch einen linearen Klassifikator klassifiziert werden können, ist gering
- Alternativ können wir einen Klassifikator betrachten, der den Eingangsraum in 1 Millionen Gitterflächen partitioniert; Mit $N = 1000$ hat man praktisch noch nichts gelernt, da 99% aller Gitterflächen noch unbelegt sind ($dim_{VC} = 1$ Millionen $+1$)

Growth-Funktion

- Nun zur Frage: gegeben die VC-Dimension einer Funktionenklasse und gegeben N, was ist die maximale Anzahl von Klassenzuordnungen, die von mindestens einem Element der Funktionsklasse richtig klassifiziert werden kann. Diese Zahl ist die *growth function* $\Delta(N)$
- Wenn N kleiner als die VC-Dimension ist, sind dies alle $\Delta(N) = 2^N$ möglichen Zuordnungen
- Wenn N größer als die VC-Dimension ist, gilt als Schranke $\Delta(N) \leq N^{dim_{VC}} + 1$
- Wenn die VC-Dimension unendlich ist, wächst $\Delta(N)$ asymptotisch wie 2^N für alle N und es ist keine Generalisierung möglich (aus Hertz, Krogh, Palmer: Introduction to the theory of neural computation)

Eine Schranke

- Mit all diesen Definitionen können wir nun endlich eine Schranke angeben. Sei N die Anzahl der Trainingsdaten, dann gilt,

$$P\left(\max_{f\in F}\left|R(f)-J^{\mathbf{Train}}(f)\right|>\epsilon\right)\leq 4\ \Delta(2N)\ \exp\left(-\epsilon^2 N/8\right)$$

- Wenn man eine Aussage über den Generalisierungsfehler machen kann, sollte die Wahrscheinlichkeit, dass die Schranke größer als ϵ ist, gegen Null gehen, mit $N\longrightarrow\infty$
- Der letzte Term wird exponentiell mit N kleiner; wird ϵ größer gewählt, dann wird die Wahrscheinlichkeit ebenfalls kleiner
- Die Growth-Function $\Delta(N)$ wächst zunächst exponentiell, bei endlicher VC-Dimension allerdings irgendwann nur nur polynomial, so dass der letzte Term dominiert

Vapnik-Chervonenkis (VC-) Theorie: Vorteile und Nachteile

- Die Zielfunktion h kann beliebig bösartig sein; h geht nicht in die Schranke ein (beeinflusst aber natürlich den Trainingsfehler)
- h muss nicht durch eine Funktion in der Funktionsklasse realisiert werden können
- Im Wesentlichen wird die Schranke durch die Komplexität der betrachteten Funktionsklasse definiert
- Es muss nur angenommen werden, dass $P(\mathbf{x})$ fest ist; die Natur kann "bösartig" sein, in Bezug auf welche Verteilung $P(\mathbf{x})$ sie auswählt. Aus dieser Verteilung muss sie dann aber unabhängige Samples liefern (d.h., hier muss die Natur "neutral" sein)!
- Die VC-Dimension lässt sich für viele interessante Klassen von Funktionen nicht berechnen; nur weniger gute oder schlechte Grenzen sind verfügbar
- Als worst-case Theorie ist die Übertragbarkeit auf den *average case* nur begrenzt möglich

Zusammenfassung

- Man ist interessiert an der Abschätzung des Generalisierungsfehler $R(f)$
- Gegeben ein beliebiges f, unverzerrte Schätzer von $R(f)$ können über einen Testsatz oder über Kreuzvalidierung erzielt werden. Beides sind Verfahren, die in der Praxis Anwendung finden
- Frequentistische Ansätze machen Aussagen über $E_D(R)$ wobei der Erwartungswert über verschiedene Trainingsdatensätze der gleichen Größe N sind und ein bestimmter Schätzer (*Least-Squares*, *Maximum Likelihood*, ...) verwandt wird. Man nimmt in der Regel an, dass die Zielfunktion durch mindestens eines der Modelle realisiert werden kann
- In einem Bayesschen Ansatz kann man R ableiten, allerdings basiert dieser auf den a prior Annahmen. Man nimmt in der Regel an, dass die Zielfunktion durch mindestens eines der Modelle realisiert werden kann. Man kann Verfahren zur Modellselektion wie BIC ableiten

- SLT beschäftigt sich mit dem maximalen Unterschied zwischen mittlerem Trainingsfehler und $R(f)$ für Funktionen aus einer Funktionsklasse F. Die Zielfunktion muss nicht realisierbar sein durch ein Mitglied von F und kann beliebig bösartig sein. Die Eingangsdaten müssen von einem beliebigen festen $P(x)$ generiert werden

APPENDIX

Miss-spezifizierte Modelle

Miss-spezifizierte Modelle

- Der Likelihood-Ansatz (und ebenso der Bayes'sche Ansatz) nimmt an, dass sich das wahre Modell in der Klasse der betrachteten befindet
- Man kann jedoch zeigen, dass im Fall der Miss-Spezifikation der ML-Ansatz definierte und sinnvolle Ergebnisse liefert
- Betrachten wir als Abstand zwischen wahrer Verteilung $P(\mathbf{x})$ und approximativer Verteilung $P_\theta(\mathbf{x})$ mit Parametern θ den sogenannten Kullback-Leibler Abstand (KL-Divergenz)

$$KL(P\|P_\theta) = \int P(\mathbf{x}) \log \frac{P(\mathbf{x})}{P_\theta(\mathbf{x})} d\mathbf{x}$$

- Der Kullback-Leibler Abstand ist gleich Null, wenn beide Verteilungen gleich sind und ist ansonsten größer Null. Beachte, dass der KL-Abstand unsymmetrisch ist: $KL(P\|P_\theta) \neq KL(P_\theta\|P)$

- Approximiert man die wahre unbekannte Verteilung durch die Samples, erhält man die negative log-Likelihood

$$KL(P\|P_\theta) \approx -\frac{1}{N}\sum_{i=1}^{N} \log P_\theta(\mathbf{x}_i)$$

- Man kann nun zeigen, dass unter schwachen Regularitätsbedingungen ein Modell, welches die log-Likelihood maximiert asymptotisch zu Parametern konvergiert, so dass der Abstand zwischen wahren und approximativem Modell im Sinne der KL-Divergenz minimal ist

- Dies bedeutet, dass auch wenn die wahre Verteilung nicht in der Klasse der betrachteten Modelle ist, der ML-Ansatz sinnvolle Ergebnisse liefert!

Minimum Description Length (MDL)

MDL: Modellannahmen

- Eine (typische) Codelänge für ein typisches Muster y in einem optimalen Code ist $-\log_2 P(\mathbf{y})$ (Shannon)
- Wir wollen die Zielwerte der Trainingsdaten $\{y_i\}_{i=1}^N$ übertragen
- Naiver Ansatz: wir übertragen die Daten, die eine mittlere Codelänge $-\log_2 P(y)$ besitzen
- Modellansatz:
  - Sender und Empfänger kennen beide die Eingangsdaten und die priori Verteilung und die funktionelle Form der Likelihood; Ziel ist die effizienteste übertragung der Daten $\mathbf{y}$.
  - Wir trainieren ein Modell und erhalten den Parametervektor $\widehat{\mathbf{w}}$
  - Wir übertragen zunächst $\widehat{\mathbf{w}}$ mit erwarteter Codelänge $-\log_2 P(\widehat{\mathbf{w}})$ und dann die Daten mit erwarteter Codelänge $-P(\mathbf{y}|\widehat{\mathbf{w}})$

– Die gesamte erwartete Codelänge (description length) ist somit

$$-\log P(\hat{\mathbf{w}}) - \log P(D|\hat{\mathbf{w}})$$

welche typischerweise geringe ist als $-\log_2 P(y)$

- Nach dem MDL (minimum descriprion length Modell) Prinzip ist das Modell optimal, für welches MDL minimal ist
- Die DL kann angenähert werden zu (siehe Appendix)

$$E(DL) \approx -\log L(\hat{w}) - \log P(\hat{w}) \approx -\log L(\hat{w}) + \frac{M}{2} \log N$$

- Hier wird Rissanen's MDL Kriterium equivalent zur Bayesschen Modellauswahl, d.h. approximativ zu BIC.
- MDL hat eine längere Entwicklung hinter sich, die diese kurze Diskussion nur unzureichend wiederspiegelt. Für eine weitergehende Diskussion: www.gruenwald.nl: A tutorial introduction to the MDL principle.

MDL: Bezug zur Informationstheorie

- Ziel ist die (wiederholte) übertragung der Werte einer Zufallsvariablen $\mathbf{x}$ mit Verteilung $P(\mathbf{x})$
- Shannon's Theorem (Source Coding Theorem) sagt aus, dass die mittlere Codelänge (*description length*, DL) eines Codes größer oder gleich der Entropie ist

$$E(DL) \geq -\sum_{\mathbf{x}} P(\mathbf{x}) \log_2 P(\mathbf{x})$$

  $DL =$ Länge des binären Codes

- Ein optimaler Code würde die Gleichheit erfüllen (Shannon Limit) und würde dem Wert $\mathbf{x}$ die Länge $-\log_2 P(\mathbf{x})$ zuordnen
- Dies bedeutet, dass häufigere Muster einen kürzeren Code erhalten sollten
- Eine (typische) Codelänge für ein typisches Muster $\mathbf{x}$ ist $-\log_2 P(\mathbf{x})$

MDL: Modellannahmen

- Wir wollen die Zielwerte der Trainingsdaten $\{y_i\}_{i=1}^N$ übertragen
- Sender und Empfänger kennen beide die Eingangsdaten und die funktionelle Form von a priori Verteilung und Likelihood; Ziel ist die effizienteste übertragung der Daten $\mathbf{y}$.
- Wir übertragen erst den Parametervektor $\mathbf{w}$ mit $P(\mathbf{w})$
- ... und dann die Ausgänge mit $P(\mathbf{y}|\mathbf{w}, \mathbf{X}, \mathcal{M})$
- Wir gewinnen, da $P(\mathbf{y})$ ohne Regressionsmodell eine sehr viel kleinere Wahrscheinlichkeitsdichte besitzt wie mit Regressionsmodell und $P(\mathbf{y}|\mathbf{w}, \mathbf{X}, \mathcal{M})$

Rissanen's Minimum Description Length (Modellselektion)

- Betrachten wir nun ein Modell $\mathcal{M}$ mit a priori Parameter Verteilungen $P(\mathbf{w})$ und Likelihoods $P(D|\mathbf{w})$
- Angenommen, dass der Parameter Schätzer $\hat{\mathbf{w}}$ und die Likelihood $P(D|\hat{\mathbf{w}})$ typischen Werten entsprechen, so ist die typische Codelänge gleich

$$-\log P(\hat{\mathbf{w}}) - \log P(D|\hat{\mathbf{w}})$$

  Dies bedeutet, dass man für die effizienteste übertragung das Modell wählen sollte, für das diese Summe minimal ist

MDL und BIC

- Eine genauere Analyse berücksichtigt, dass eine ungenaue Kodierung von $\mathbf{w}$ äquivalent zu zusätzlichem Rauschen auf der Zielgröße ist
- Man kann argumentieren, dass der Parametervektor $\mathbf{w}$ in jeder Dimension nur mit $\sqrt{N}$ Bins pro Dimension übertragen werden muss. Dies bedeutet, dass bei mehr Daten man mit einer besseren Kodierung der Parameter gewinnt. Unter der Annahme von Uniformität ist der Komplexitätsterm

$$\log P(\mathbf{w}) \rightarrow \log(1/\sqrt{N})^M = -\frac{M}{2} \log N$$

und $\mathcal{M}DL$ ist äquivalent zu BIC.