

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:

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

Weitere Beispiele von Verlustfunktionen

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

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

$$y \log(1 + \exp(-f(\mathbf{x}, \mathcal{M}_i))) + (1 - y) \log(1 + \exp(\mathbf{x}_i^T \mathbf{w}))$$

- Perzeptron: $L[y, f(\mathbf{x}, \mathcal{M}_i)] = |-yf(\mathbf{x}, \mathcal{M}_i)|_+$
- Vapniks Hyperebenen: $L[y, f(\mathbf{x}, \mathcal{M}_i)] = |1 - yf(\mathbf{x}, \mathcal{M}_i)|_+$

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

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

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

- Dieser Schätzer ist unverzerrt (*unbiased*, erwartungstreu)
- Ein Schätzer für die Varianz des Schätzers mit quadratischer Kostenfunktion ist

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

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

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 $\mathbf{x}_i, y_i \in \text{Train}$

- 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

```
graph TD; A[Datensatz] --> B[Trainingsdaten]; A --> C[Testdaten]; D[Zum Trainieren des Modells] --> B; E[Zum Testen des Modells] --> C;
```

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 $k = 1, \dots, K$: 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

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

- Wir können nun schätzen mit dem mittleren Testfehler

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

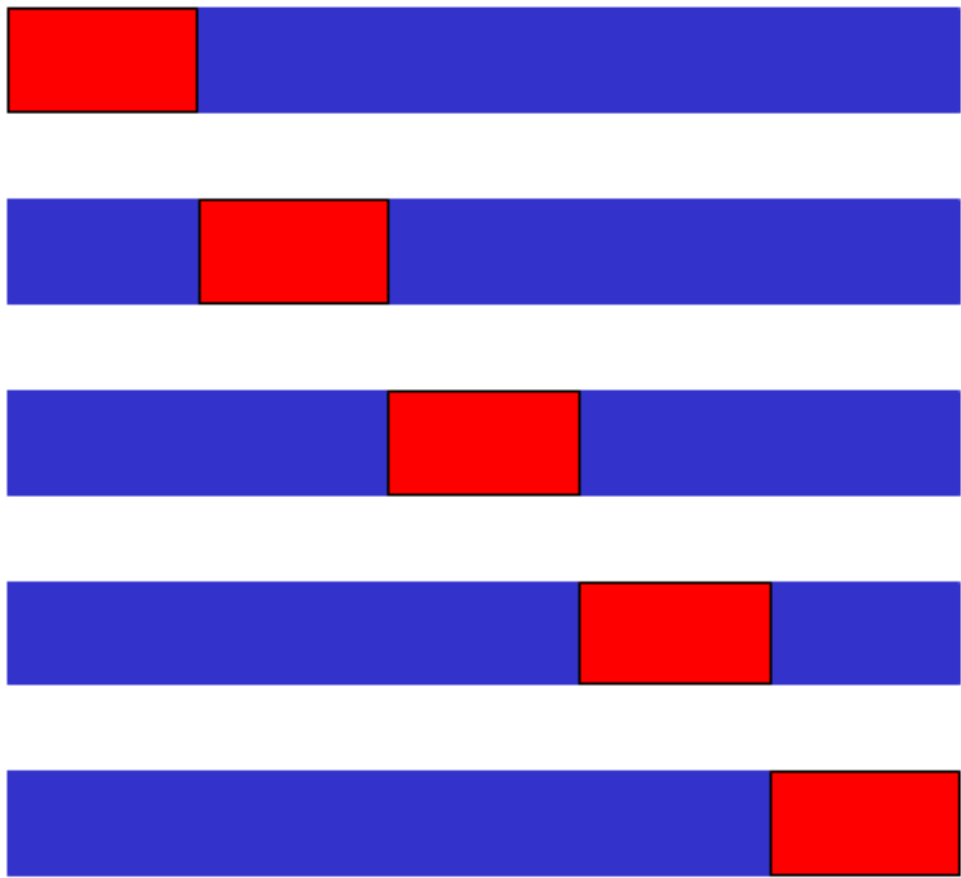
- Die Varianz des Schätzers kann man schätzen mit

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

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

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



5-Fache

Kreuzvalidierung:

Blau: Trainingsdaten

Rot: Testdaten

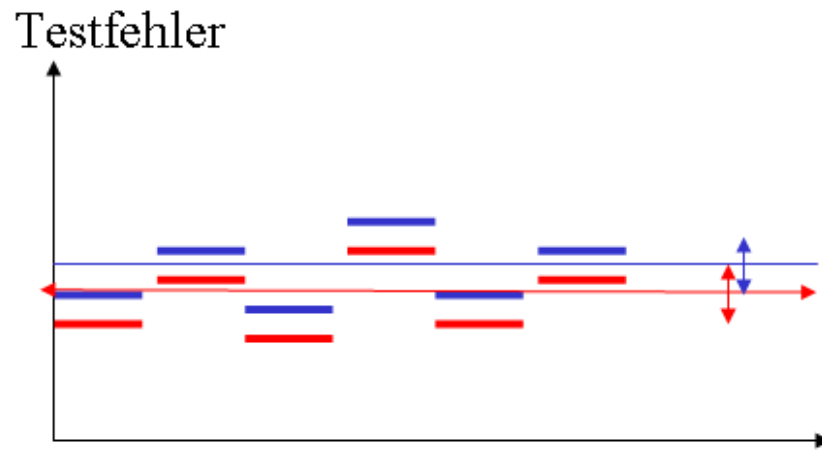
Gepaarte Tests

- 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

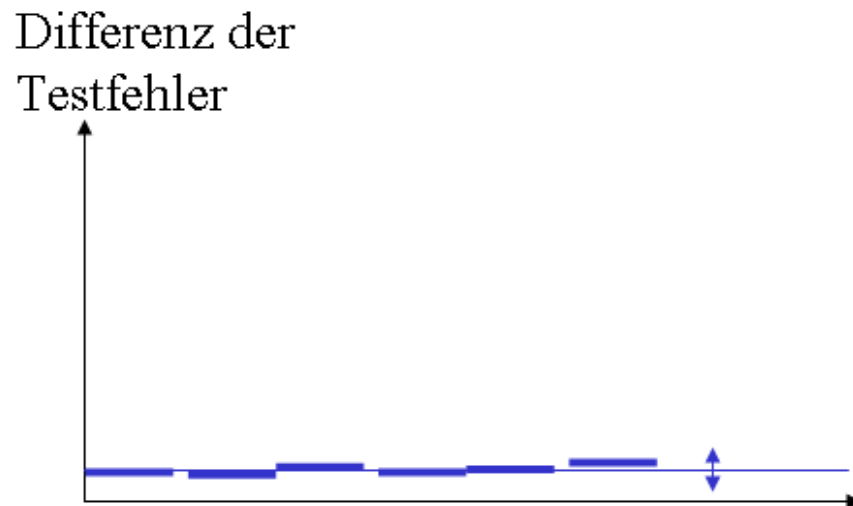
$$\text{MeanDiff}_{i,j} = \frac{1}{K} \sum_{k=1}^K J_k^{\text{Test}}(\mathcal{M}_i) - J_k^{\text{Test}}(\mathcal{M}_j)$$

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



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



- Untersucht man jedoch die Differenz der Performanz ist die bessere Performanz von Modellansatz 1 (rot) 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 Testfehler 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

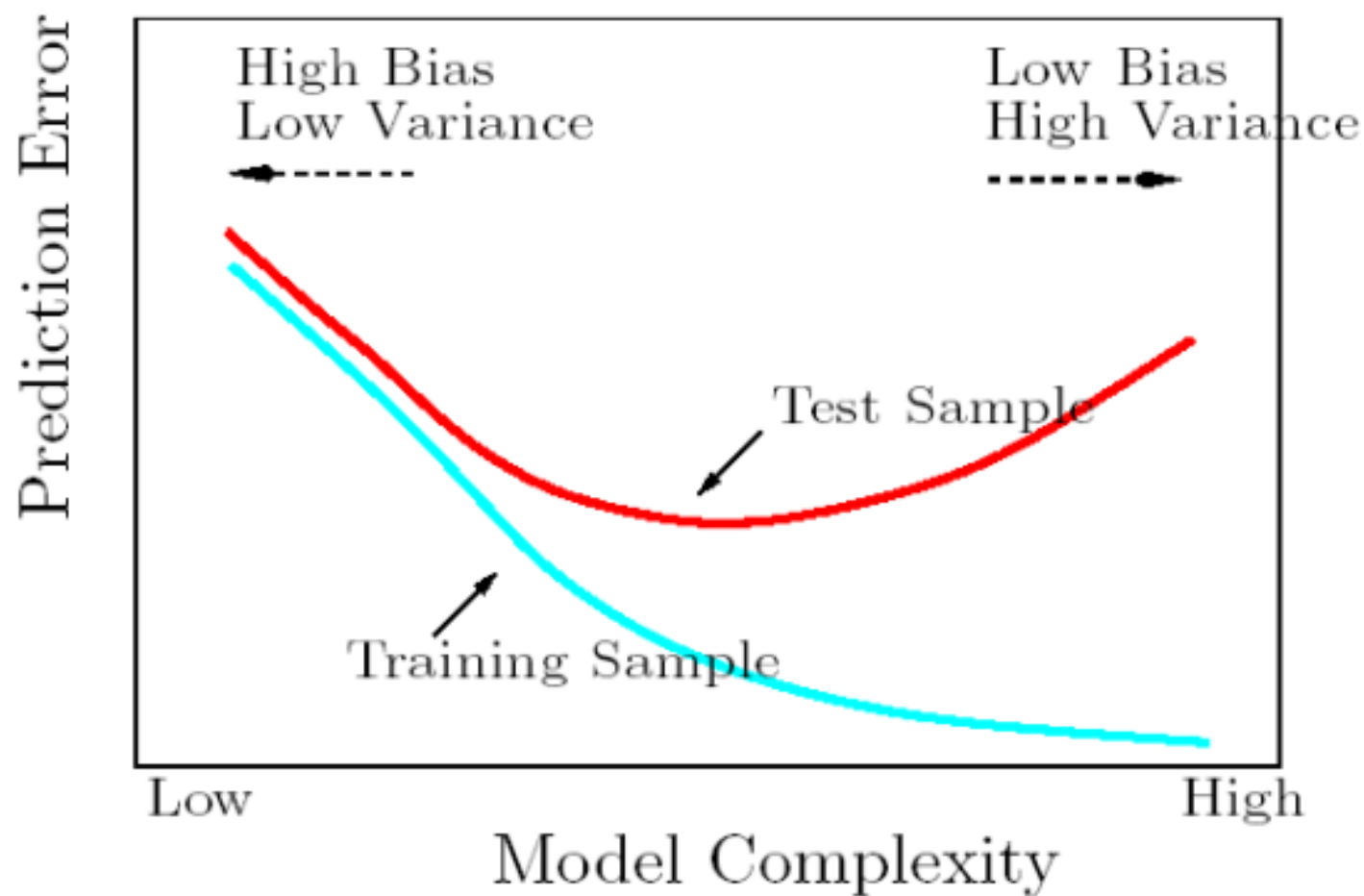


Figure 7.1: *Behavior of test sample and training sample error as the model complexity is varied.*

Wichtige Gütemaße für Klassifikation

- **Accuracy** (allg. Gütemaß): $P(pred = 1, y = 1) + P(pred = 0, y = 0)$
- **Recall** (Sensitivität, sensitivity, true positive rate): $P(pred = 1|y = 1)$ (Eine Suchmaschine sollten alle relevanten Dokumente finden; ein Feuermelder sollte jedes Feuer melden)
- **Precision** (Relevanz, Wirksamkeit, Genauigkeit) $P(y = 1|pred = 1)$ (Eine Suchmaschine sollten nur relevante Dokumente finden; wenn der Feuermelder anschlägt, sollte es auch ein Feuer geben)
- **Specificity** (Spezifität, true negative rate) $P(pred = 0|y = 0)$ (Wenn es kein Feuer gibt, sollte der Feuermelder nicht anschlagen)
- **F-measure** (F-Maß) $F = 2 \frac{Precision \times Recall}{Precision + Recall}$

Wichtige Gütemaße für Klassifikation (2)

Betrachte als Entscheidung die Funktion $\text{sign}(f(x) + \alpha)$ für $-\infty < \alpha < \infty$, d.h. wir variieren die Schwelle für die Klassifikation. Man kann Recall beliebig verbessern mit $\alpha \rightarrow -\infty$ bzw Precision beliebig verbessern mit $\alpha \rightarrow \infty$. Wichtig ist es, einen Anwendungsspezifischen optimalen Kompromiss zu treffen.

- ROC (Receiver operating characteristic). Für die ROC-Kurve wird α variiert und Recall (y-Achse) gegen (1-Specific) (x-Achse) geplottet. (1-Specific) wird auch als *false-positive rate* bezeichnet.
- AUC: Area under ROC curve. Numerische Integration der ROC Kurve. Ein perfekter Klassifikator ergibt $\text{AUC} = 1$, eine Zufallsklassifikation eine $\text{AUC} = 0.5$

Lerntheorien und theoretische Abschätzungen des Generalisierungsfehlers

Überblick: Statistische Theorien und Lerntheorien

VC-Theorie (Statistische Lerntheorie)

- Verteilungsfrei
- Worst-case Analyse
- *Vapnik*

PAC Lernen (probably approximate correct)

- Ähnlich zur VC-Theorie
- Berücksichtigt rechnerische Komplexität
- *Valiant*

Regularisierungstheorie

- Regularisierung: schlecht-gestellten Problemen -> gut-gestellte Probleme
- *Hadamard, Tikhonov*

Wahrscheinlichkeitslehre

- Beispiel: Bester linearer Schätzer
- Eigentlich nicht Statistik, aber führt zu einfachen Termen (Korrelationen), die geschätzt werden können

(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*

Robuste Statistik

- *Huber*

Stein Estimation

- Es gibt bessere Schätzer als ML-Schätzer
- *Stein* Schätzer

Prinzip der kleinsten Quadrate

- Gauss
- Entspricht Gauss'scher Likelihood

MDL – Theorie

- (minimum description length)
- Informationstheorie
 - *Rissanen, Wallace, Boulton*

Information Bottleneck

- *Tishby, Pereira, Bialek*

Empirical Bayes (technicality)

- Type II likelihood
- Evidence Framework

Objektive Bayes'sche Statistik

- Noninformative Priors (Jeffrey)
- Maximum Entropy Priors

Frequentistische Statistik

- Ablehnung eines Priors
- Dominierender statistischer Ansatz im 20-ten Jahrhundert
- *Fisher*
- *Pearson und Neyman*

Neyman-Pearson-Wald Entscheidungstheorie

- MinMax
- Bayes Optimal

Algorithmische Statistik

- Fokus auf Vorhersageleistung (nicht Parameterschätzung)
- *Breiman, Huber, Friedman*

- **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

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

- Uns interessiert nun

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

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

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

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

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

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

Bias-Varianz Zerlegung im Beispiel

- Im Beispiel:

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

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

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

- Für $N \rightarrow \infty$ 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

Kommentare

- Mit $E_D(R)$ kann man 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

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

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

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

Mallot's C_P

- Die Lösung ist also

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

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

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

- Somit kann man schreiben

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

- Diese Abschätzung wird C_P -Statistik genannt

Akaike's 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:

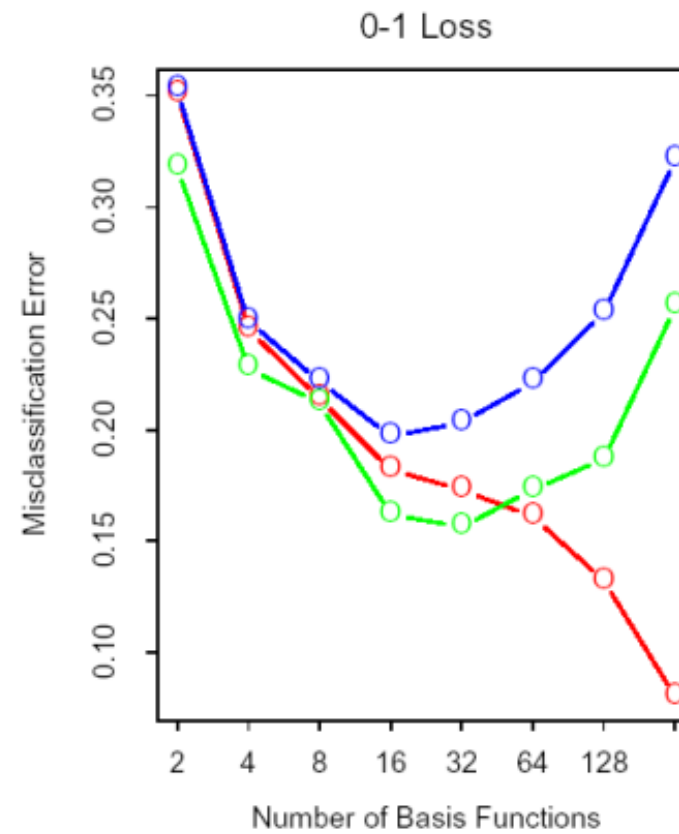
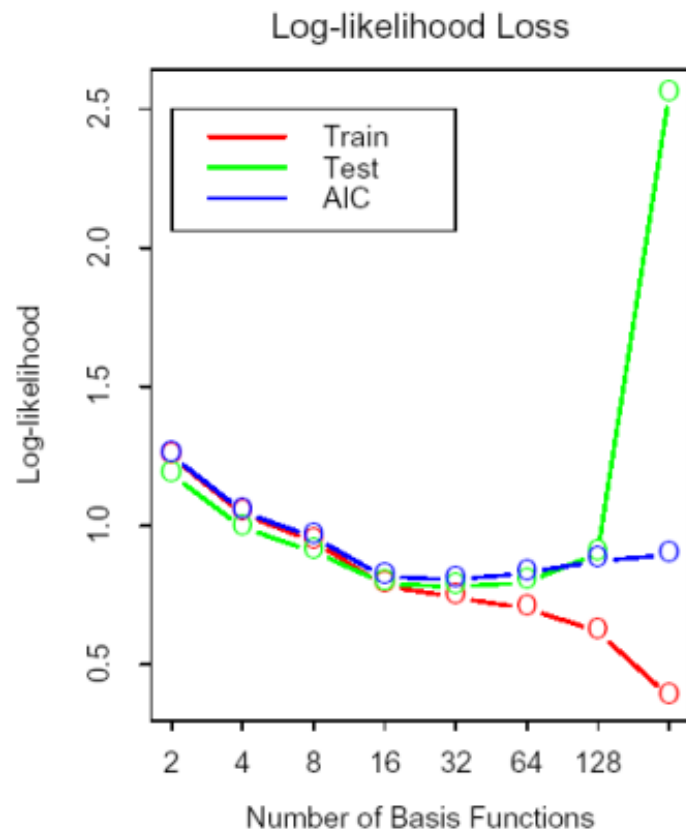
$$AIC = \frac{1}{\sigma^2} \frac{1}{N} \sum_{i=1}^N (y_i - f(\mathbf{x}_i, \mathbf{w}))^2 + 2 \frac{M}{N} = \frac{1}{\sigma^2} C_P$$

- 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 oft unpraktisch; eine Modellauswahl wird deshalb häufig betrachtet
- 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

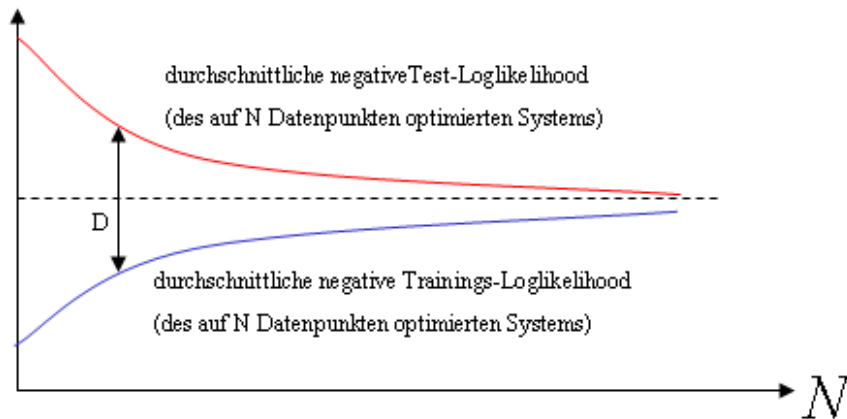
$$-\frac{1}{2N} BIC = \frac{1}{N} \log L - \frac{M}{N} \frac{1}{2} \log N$$

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 Trainingsdaten

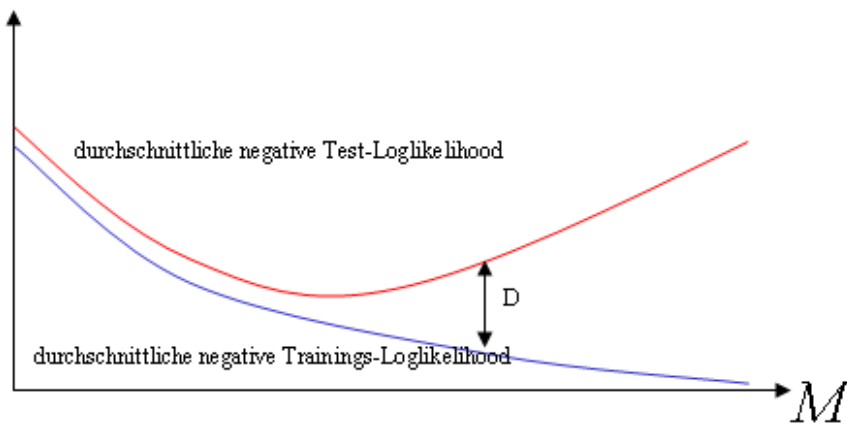
Vergleich: AIC und BIC



Schätzung von D :

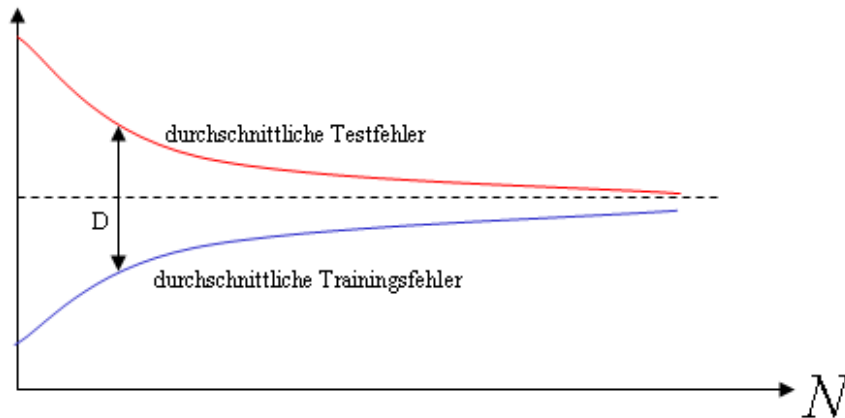
$$\text{AIC: } \hat{D} = \frac{M}{N}$$

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



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

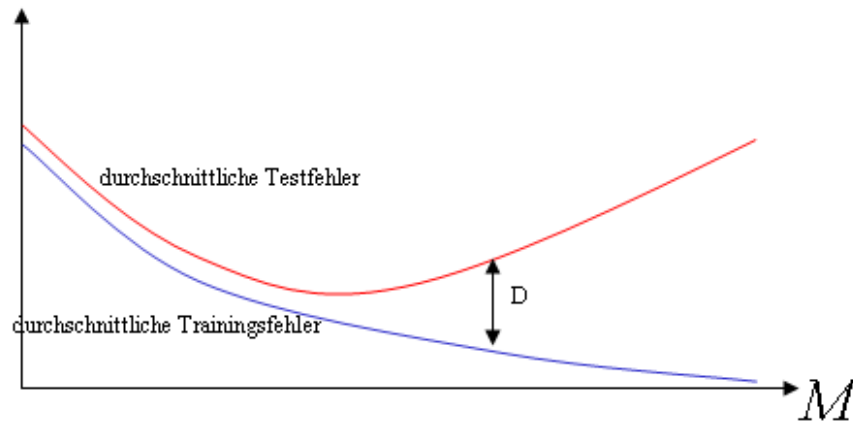
AIC und BIC: Spezialfall Gauß'sche Likelihood, quadratischer Fehler



Schätzung von D:

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

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



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 Kriterium identisch zum BIC Kriterium (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

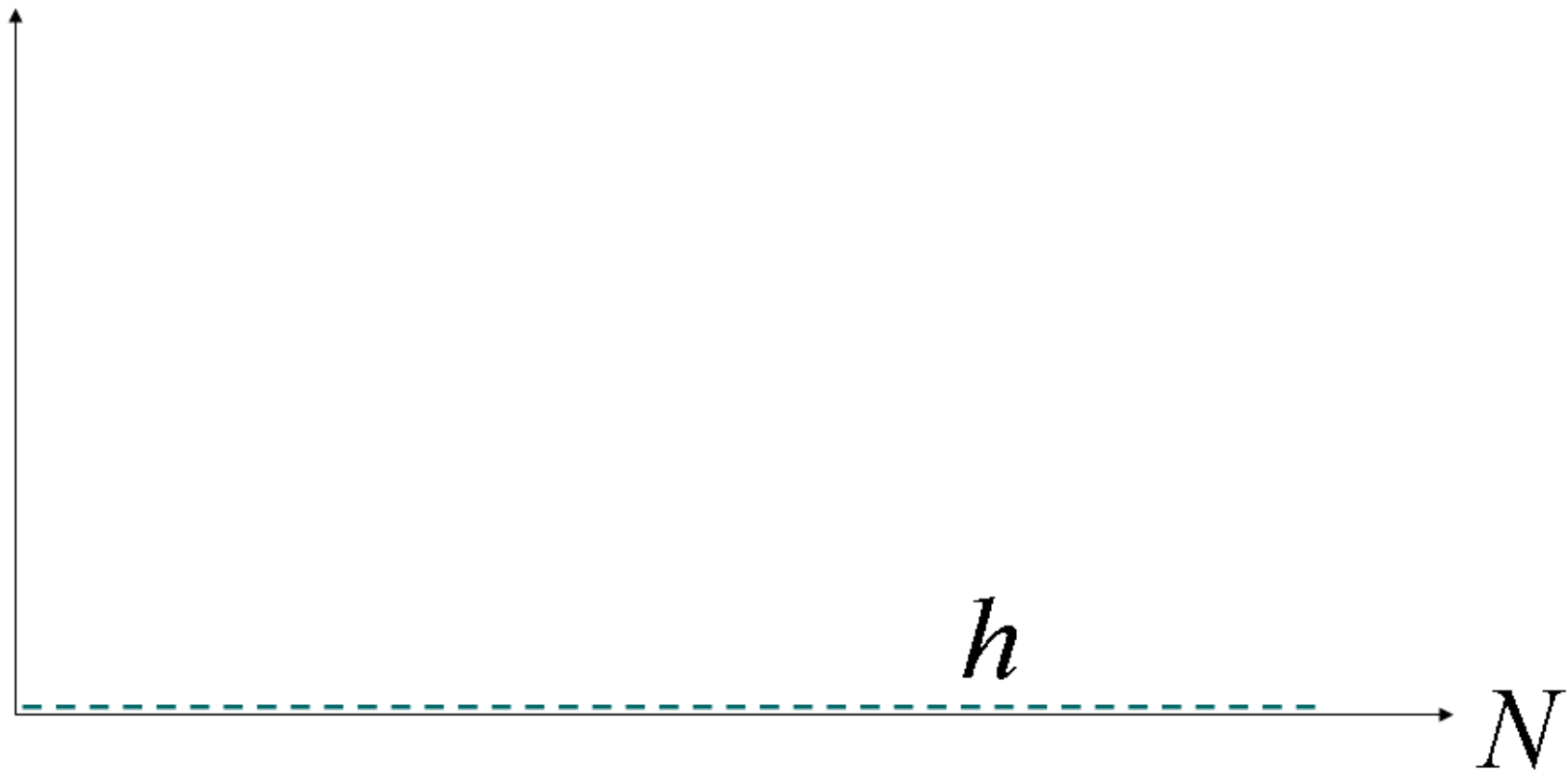
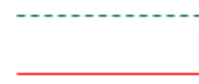
- Wie als erstes diskutiert, kann man über einen Testsatz oder über Kreuzvalidierung eine Aussage über den Generalisierungsfehler einer beliebigen Funktion machen
- Der frequentistische Ansatz beinhaltet einen Schätzer: d.h. man macht Aussagen über Funktionen, die z.B. den quadratischen Fehler im Training minimieren
- In der SLT betrachtet man **Funktionen f aus einer Funktionenklasse F** und interessiert sich für deren Generalisierungsfehler
- Beispiel: F ist die Klasse aller linearen Klassifikatoren mit $M - 1$ Merkmalen und f ist ein spezieller Klassifikator
- Die SLT macht eine Aussage über den **maximalen Unterschied zwischen $J^{\text{Train}}(f)$ (dem mittleren Trainingsfehler) und dem (mittleren) Generalisierungsfehler $R(f)$** ; dieser maximale Unterschied gilt dann für jede Funktion aus der Klasse, also nicht nur für die Funktion, die den Trainingsfehler minimiert
- **Wesentlich: Es wird nicht angenommen, dass die wahre Funktion h sich in F befindet**; h kann beliebig “böartig” sein (unstetig, ...).

- Im Beispiel muss also der wahre Klassifikator kein linearer Klassifikator sein
- In diesem Fall haben frequentistische und Bayes'sche Ansätze Probleme, sinnvolle Aussagen über die Generalisierungsfähigkeit zu machen
- Dies ist der Grund weshalb die Aussagen der VC-Theorie sehr konservativ sind! Es wird keinerlei Glattheitsannahme gemacht (Fluch der Dimensionalität!)



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

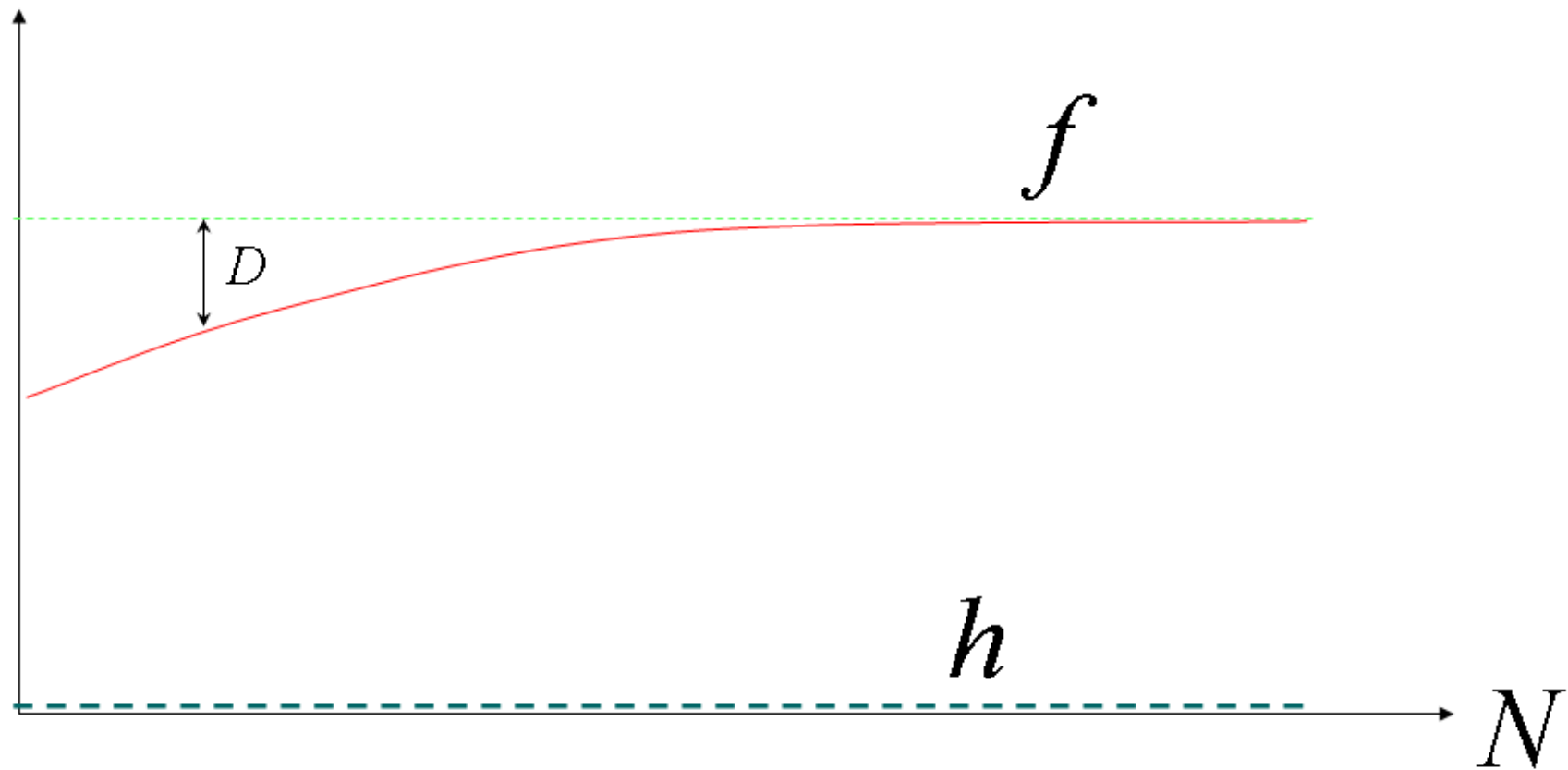
- Die VC-Theorie ist verteilungsfrei, das heißt sie macht keine Annahmen über eine zugrundeliegende Verteilung $P(\mathbf{x})$
- Die einzige wesentliche Annahme: Daten werden von einer *festen* Verteilung $P(\mathbf{x})$ generiert
- Zielgrößen werden von $h(\mathbf{x})$ generiert (im einfachsten Fall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

Generalisierungsfehler (gestrichelt)
Mittl. Trainingsfehler (durchgezogen)



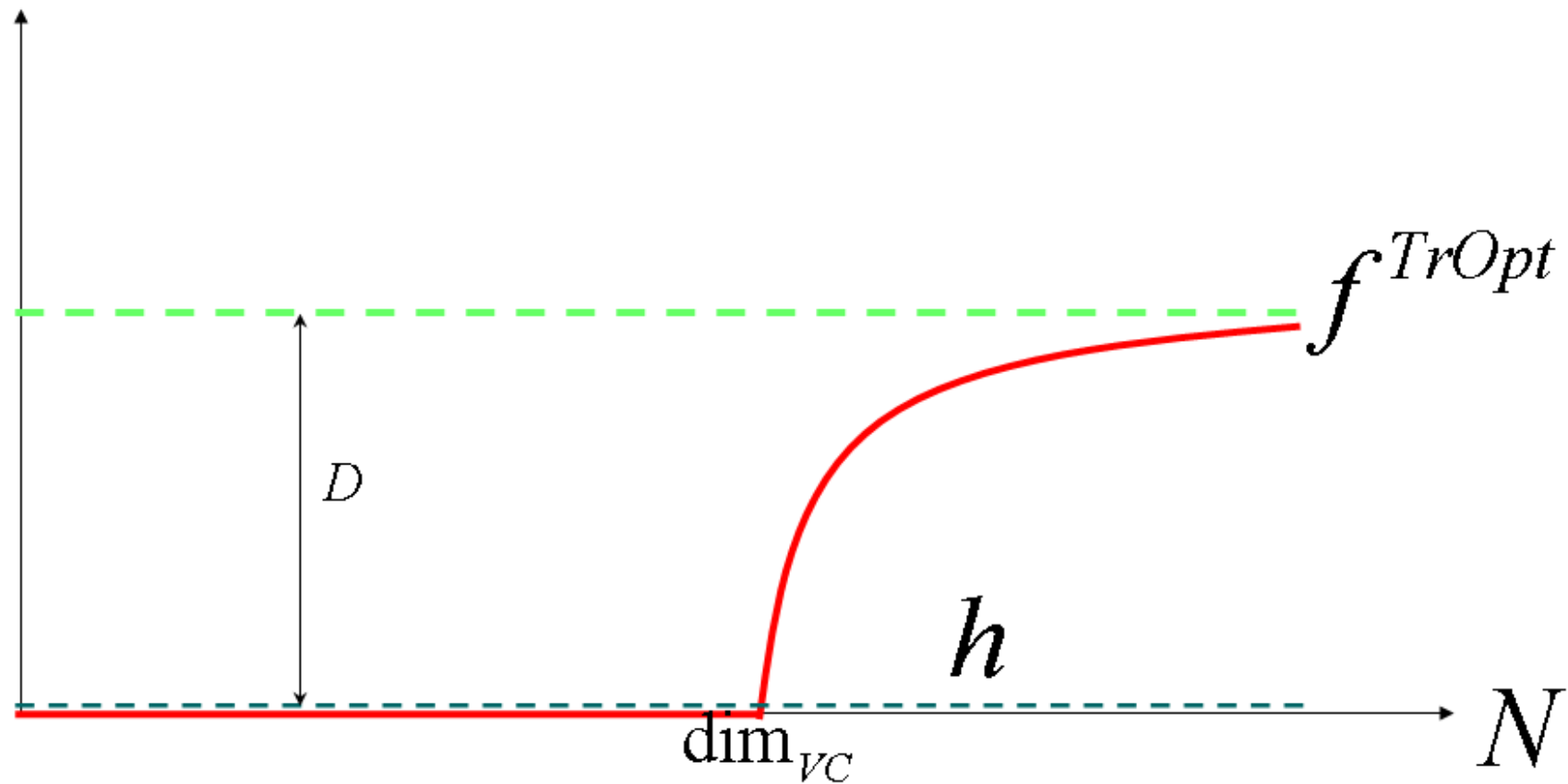
Die wahre Funktion h hat den kleinsten Generalisierungsfehler gleich 0;
der mittlere Trainingsfehler ist auch gleich 0

Generalisierungsfehler (gestrichelt) 
Mittl. Trainingsfehler (durchgezogen) 



Für eine beliebige feste Funktion f aus F ist der Generalisierungsfehler größer 0. Die VC Theorie berechnet eine obere Schranke für D , welche für jedes f aus F und für jedes h gilt.

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



Diese Funktion f sei die Funktion aus F , mit dem kleinsten Trainingsfehler: der Unterschied zwischen mittlerem Trainingsfehler und erwarteten Risiko ist groß

VC-Dimension und Generalisierung

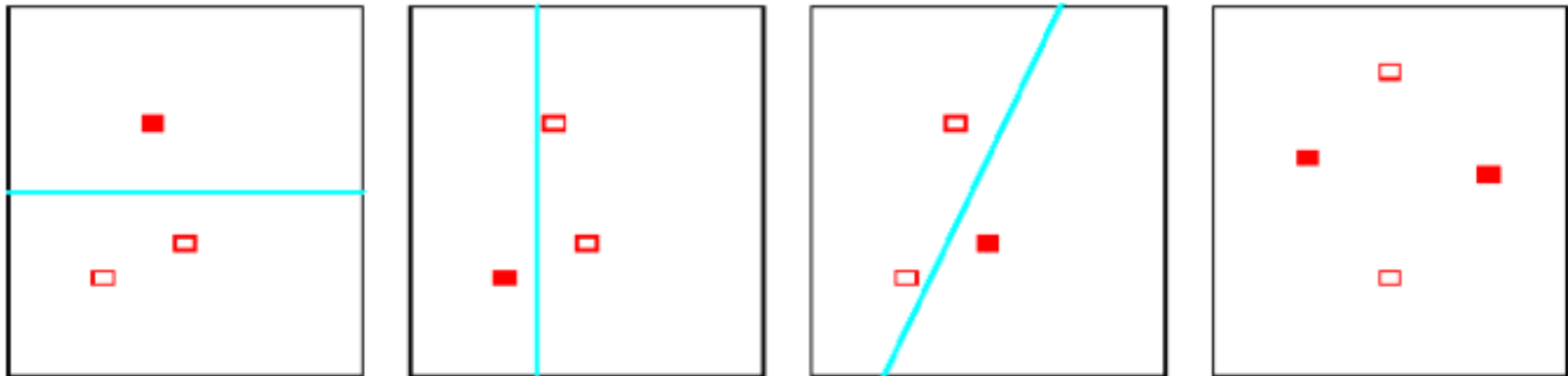
- dim_{VC} ist die VC-Dimension: eine gültige Generalisierung ist nur möglich, wenn $N > dim_{VC}$
- Für einen linearen Klassifikator ist $dim_{VC} = M$. Dies entspricht im Grunde auch dem klassischen Ergebnis, dass man mehr Trainingsdaten als freie Parameter haben sollte, um einen Klassifikator eindeutig zu definieren (ohne Regularisierung)
- Im Allgemeinen entspricht dim_{VC} der Mächtigkeit einer Funktionenklasse. In den folgenden Folien wird die VC-Dimension formal im Sinne der VC-Theorie definiert

VC-Dimension

- 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)



Selektion von Modellklassen

- Gegeben eine Modellklasse F werden (probabilistische) Schranken, für den Unterschied zwischen (mittlerem) Trainingsfehler und (mittlerem) Generalisierungsfehler berechnet
- 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

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

- Die technische Herausforderung besteht nun darin, gute Schranken zu finden. Im Anhang werden Schranken angegeben und es wird das wichtige Konzept der Konsistenz besprochen

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

- Es muss nur angenommen werden, dass $P(\mathbf{x})$ fest ist; die Natur kann “böartig” 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 Zielfunktion h kann beliebig böartig 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
- Die SLT ist sehr konservativ, da sie keinerlei Annahmen über die Komplexität von h macht; da oft eine Glattheitsannahme sinnvoll ist, sind die beobachteten Testfehler oft weit kleiner, als durch die VC-Theorie vorhergesagt! (siehe auch: No-Free-Lunch Theorem)
- Man kann einen Zusammenhang zum Fluch der Dimensionalität sehen: Wenn Funktionen beliebig komplex sein können, ist Generalisierung in hohen Dimensionen sehr problematisch und Abschätzungen der Generalisierungsfähigkeit sind zwangsläufig sehr konservativ!

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.
- In einem Bayesschen Ansatz kann man R ableiten, allerdings basiert dieser auf den a prior Annahmen. Man kann Verfahren zur Modellselektion wie BIC ableiten
- Ein wesentlicher “Trick” des Frequentistischen und des Bayesschen Ansatzes: da angenommen wird, dass die wahre Funktion sich durch einer der betrachteten Funktionen realisieren lässt, wird implizit die Komplexität der Zielfunktion eingeschränkt: d.h, es wird dadurch in der Regel impliziert, dass die Zielfunktion nicht beliebig “verrückt” sein kann!

- 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 böartig sein. Die Eingangsdaten müssen von einem beliebigen festen $P(x)$ generiert werden
- Die SLT ist sehr konservativ, da sie keinerlei Annahmen über die Komplexität von h macht; da oft eine Glattheitsannahme sinnvoll ist, sind die beobachteten Testfehler oft weit kleiner, als durch die VC-Theorie vorhergesagt!
- Während frequentistische und Bayes'sche Ansätze älter sind als das Maschinelle Lernen, stellt die SLT einen originären Beitrag dar

APPENDIX: VC-Theorie

Konsistenz

- Man verlangt Konsistenz: mit $N \rightarrow \infty$ soll nach dem Kriterium die Funktion mit dem geringsten mittleren Trainingsfehler optimal sein; wann konvergiert der mittlere Trainingsfehler zum tatsächlichen Fehler (Konsistenz des ERM (*empirical risk minimization*) Prinzips)?
- Damit das ERM Prinzip konsistent ist, ist es notwendig und hinreichend, dass der mittlere Trainingsfehler uniform wie folgt zum tatsächlichen mittleren Generalisierungsfehler konvergiert:
- Worst Case Analysis (MinMax) (one-sided uniform convergence)

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

(die gilt **für alle** $f : A \leq R(f) \leq B$ mit beliebigen Schranken A, B)

- In Worten: der Unterschied zwischen $J^{\text{Train}}(f)$ 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 Trainingsatz ein kleines $J^{\text{Train}}(f)$ 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 $J^{\text{Train}}(f)$ und dem Generalisierungsfehler $R(f)$
- Nach Vapnik folgt aus dem Theorem, dass eine Analyse der Konvergenz des ERM Prinzips eine Worst-Case Theorie sein muss (Vapniks Schlüsseltheorem) !

Schranken zur Generalisierungsfähigkeit

- Wir versuchen jetzt anzudeuten, wie man sinnvolle Schranken erhalten kann
- Gegeben eine Funktionenklasse und gegeben N , was ist die maximale Anzahl von Klassenzuordnungen (für mindestens eine Anordnung der Datenpunkten), die von mindestens einem Element der Funktionsklasse richtig klassifiziert werden kann. Diese Zahl ist die *growth function* $\Delta(N)$

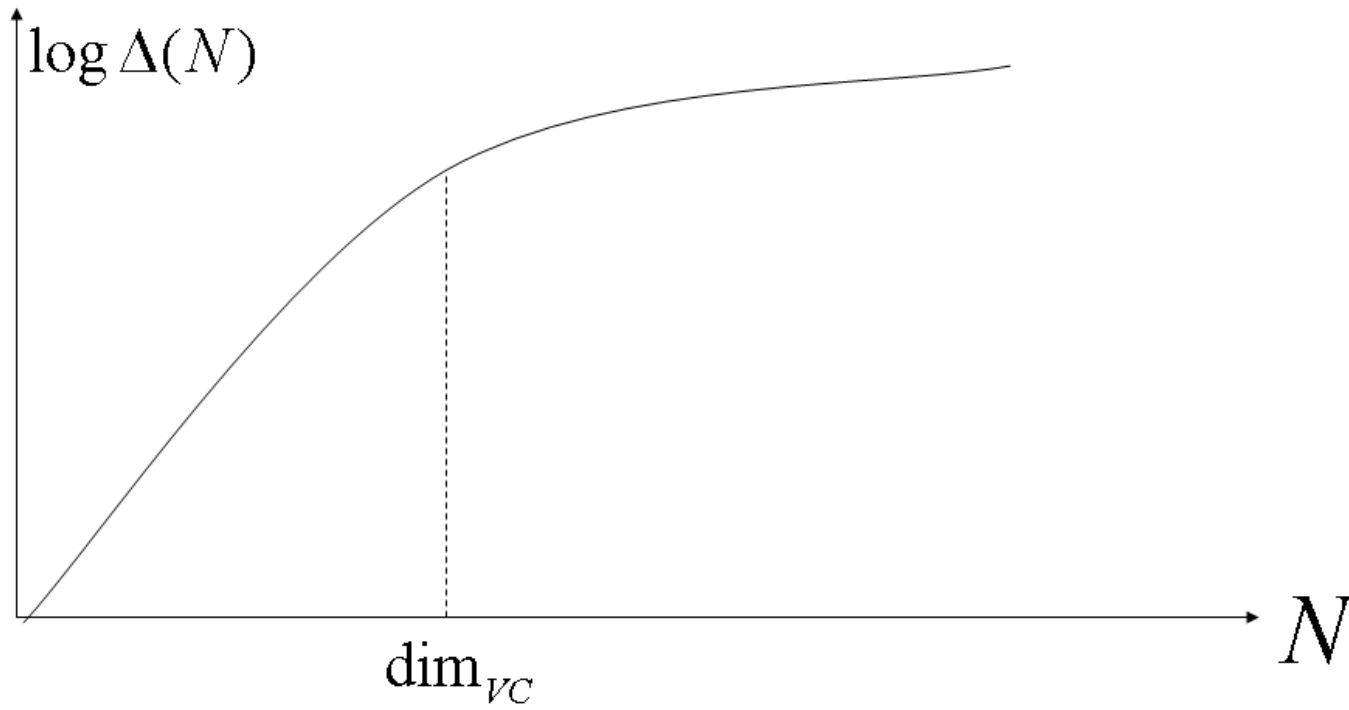
- Wir nun eine Schranke angeben. Sei N die Anzahl der Trainingsdaten, dann gilt,

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

- Die Growth-Function $\Delta(N)$ wächst zunächst exponentiell, bei endlicher VC-Dimension allerdings irgendwann nur noch polynomial, so dass der letzte Term dominiert
- Wenn N kleiner als die VC-Dimension ist, dann ist $\Delta(N) = 2^N$
- 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 Schranke für den Generalisierung möglich (aus Hertz, Krogh, Palmer: Introduction to the theory of neural computation)
- Für eine gegebene Funktionenklasse benötigt man, um Generalisierung zu erlangen, mindestens dim_{VC} Datenpunkte

Verlauf der Growth-Funktion



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!

APPENDIX: Minimum Description Length (MDL)

MDL: Modellannahmen

- Eine (typische) Codelänge für ein typisches Muster y in einem optimalen Code ist $-\log_2 P(y)$ (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 y .
 - Wir trainieren ein Modell und erhalten den Parametervektor \hat{w}
 - Wir übertragen zunächst \hat{w} mit erwarteter Codelänge $-\log_2 P(\hat{w})$ und dann die Daten mit erwarteter Codelänge $-P(y|\hat{w})$

– Die gesamte erwartete Codelänge (description length) ist somit

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

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

- Nach dem MDL (minimum description 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 äquivalent zur Bayesschen Modellauswahl, d.h. approximativ zu BIC.
- MDL hat eine längere Entwicklung hinter sich, die diese kurze Diskussion nur unzureichend widerspiegelt. 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 MDL ist äquivalent zu BIC.