

4.2 Anwendungsbeispiele aus der Bioinformatik

4.2.1 SVMs und die Regularisierung von Gen-Expressionsdaten

Brown, Grundy, Lin, Cristianini, Sugnet, Ares, Haussler (1999): Support Vector Machine Classification of Microarray Gene Expression Data

$m = 79$ cDNA-Microarrays mit $n = 2487$ Gene von *S. cerevisiae* aus 5 Klassen (Zitronensäure-Zyklus, Respirationskettenkomplex, ribosomale Proteine, Proteasome, Histone).

Aufgabe: entscheide für jedes Protein und jede Klasse ob's dazugehört. (Info in Hefe-Genom-Datenbank erhältlich)

Je 2/3 der Gene wurden für's Training verwendet, das übrige Drittel zum Testen.

Verglichene Methoden: Lineare Diskriminante, Parzen-Fenster (klassifiziere gemäß $\text{sgn} \sum y_i \exp(-\|x_i - x\|^2)$), Entscheidungsbäume, SVM (**Testsieger!**)

“Visual inspection of the raw data indicates that such classification should be possible”

$$X_i := \log \frac{\text{Exprlevel von Gen } x \text{ in Versuch } i}{\text{Exprlevel von Gen } x \text{ in Referenzzustand}}$$

Normalisierung:

$$\bar{X}_i := \frac{X_i}{\sqrt{\sum_{j=1}^m X_j^2}}$$

Fehlerfunktion: (Anzahl falsch Positive) + 2 x (Anzahl falsch negative)

SVM mit polynomialen Kernen verschiedener Grade und Gauß-Kernen
(Testsieger!)

Problem: Funktionale Genklassen enthalten sehr wenige Gene, die in den viele negativ-Beispielen untergehen

Lösungsansatz ähnlich zu Soft-Margin oder Ridge Regression: Kontrolliere Verhältnis falsch-positive/falsch-negative durch Training mit $K(x, x) + \lambda \frac{n^+}{n}$ bzw. $K(x, x) + \lambda \frac{n^-}{n}$, wobei n^+ und n^- die Anzahlen an positiv- und negativ-Beispielen sind.

Ridge Regression Sind Variablen korreliert, können zugeh. β_i sehr groß werden und sich gegenseitig aufheben. Eine Möglichkeit, das zu vermeiden:

$$\hat{\beta} = \arg \min_{\beta} \left\{ \sum_i \left(y_i - \beta_0 - \sum_j x_{ij} \beta_j \right)^2 + \lambda \sum_j \beta_j^2 \right\}$$

bzw.

$$\hat{\beta} = \arg \min_{\beta: \sum_i \beta_i^2 \leq s} \left\{ \sum_i \left(y_i - \beta_0 - \sum_j x_{ij} \beta_j \right)^2 \right\}$$

Achtung: Skalierungsabhängig, also Daten vorher normalisieren.

Lösung für Ridge Regression:

$$\hat{\beta} = (X^T X + \lambda I)^{-1} X^T y$$

(gewöhnliche lineare Regression erhält man daraus mit $\lambda = 0$)

Alternative Herleitung für Ridge Regression:

$\mathcal{N}(0, \tau^2)$ -Prior auf β_j

$$y_i \sim \mathcal{N}\left(\beta_0 + \sum_j x_{ij}\beta_j, \sigma^2\right)$$

Dann ist $\hat{\beta}$ Mittelwert der a-posteriori-Verteilung, wobei $\lambda = \tau^2/\sigma^2$.

Geometrische Interpretation der Ridge Regression:

Alle Hauptkomponenten werden geschrumpft, und je kürzer desto mehr.

Ist d_j der Eigenwert zur j -ten Hauptkomponente,

so schrumpfe mit Faktor $\frac{d_j^2}{d_j^2 + \lambda}$.

Ähnliche Methode: **LASSO** (least absolute shrinkage and selector operator)

$$\hat{\beta} = \arg \min_{\beta: \sum_i |\beta_i| \leq s} \left\{ \sum_i \left(y_i - \beta_0 - \sum_j x_{ij} \beta_j \right)^2 \right\}$$

nicht linear! einige Koeffizienten werden auf 0 gesetzt.

“kind of continuous subset selection”

(Hastie, Tibshirani, Friedman, 2001, *The Elements of Statistical Learning*)

S. Ramaswamy et al. (2001) Multiclass cancer diagnosis using tumor gene expression signatures. *PNAS* 18(26) 15149-15154

Affy-Microarrays von 218 Tumor-Proben aus 14 Klassen + 90 Proben gesundes Gewebe; 16063 Gene+ESTs

(Proben ausgewählt aus 314 Tumor- und 98 gesunden Proben nach Qualitätskontrolle)

Vorbehandlung der Daten mit Affymetrix-Software

Klassifikation nach jeder Tumorklasse mit linearer SVM, dann One-vs.-All (OVA): nimm die Klasse k für die $\langle \beta^{(k)}, x \rangle + \beta_0^{(k)}$ möglichst groß wird

(nach Minimierung von $\|\beta^{(k)}\|^2$ bzgl. $\forall i : y_i(\langle \beta^{(k)}, x_i \rangle + \beta_0^{(k)}) \geq 1$).

Kreuzvalidierung der Methode: von 144 Tumoren wurde jeweils einer ausgelassen, die SVM mit den restlichen trainiert, der eine musste dann vorhergesagt werden.

Außerdem wurde noch mit den 144 trainiert, um dann 54 andere zu klassifizieren.

Daraus geschätzte Korrektheit: 78%

Problem: β “enthält” alle Gene und ist daher schwer zu interpretieren

“Curse of Dimensionality”

“large p , small n problem” (p Gene, n Microarrays)

Recursive Feature Elimination: Gene, für die $|\beta_i|$ in unteren 10%-Bereich ist, werden eliminiert, dann wird OVA-SVM wiederholt; das ganze wird iteriert.

Ergebnis: Bei OVA-SVM je mehr Gene desto besser, aber 70% Genauigkeit ab ca. 20 Genen.

Hastie, Tibshirani, Eisen, Alizadeh, Levy, Staud, Chan, Botstein, Brown (2000) ‘Gene shaving’ as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology* 1(2)

1. $X :=$ Expressionsmatrix
2. $v :=$ erste Hauptkomponente von X
3. “Rasieren” die 10% Gene g mit den kleinsten $\langle g, v \rangle$ weg.
4. Wiederhole 2 und 3 bis nur noch 1 Gen übrig ist, seien $S_1 \subset \dots \subset S_k \subset \dots \subset S_N$ die dabei entstehenden Mengen, wobei S_k k Gene enthält; Bestimme das optimale \hat{k} mit der *gap statistik*.
5. Orthogonalisiere jede Zeile von X zum durchschnittlichen Gen $\bar{x}_{\hat{k}}$ in $S_{\hat{k}}$
6. Wiederhole 1 bis 5 mit den Orthogonalisierten Daten, um das zweitbeste Cluster zu finden. Iteriere weiter, um die M besten Cluster zu finden.

gap statistik

m Versuche, x_{ij} = Expression vom Gen i im Versuch j

\bar{x} = Mittelwert über alle x_{ij} mit $i \in S_k$ und $j \leq m$

\bar{x}_j = Mittelwert über alle x_{ij} mit $i \in S_k$ und festes j

$$V_W = \frac{1}{m} \sum_{j=1}^m \frac{1}{k} \sum_{i \in S_k} (x_{ij} - \bar{x}_j)^2 \quad \text{within variance}$$

$$V_B = \frac{1}{m} \sum_{j=1}^m (\bar{x}_j - \bar{x})^2 \quad \text{between variance}$$

$$V_T = \frac{1}{mk} \sum_{j=1}^m \sum_{i \in S_k} (x_{ij} - \bar{x})^2 = V_W + V_B \quad \text{total variance}$$

Bevorzuge Cluster mit hoher Varianz und hoher Kohärenz zwischen den Genen:

$$D_k := V_B/V_T$$

Permutiere nun die Einträge in jeder Zeile der Matrix X und berechne V_B/V_T .
Wiederhole das 1000 mal.

D_k^* := Mittelwert der 1000 Ergebnisse für V_B/V_T .

$$\text{Gap}(k) := D_k - D_k^*$$

$$\hat{k} := \arg \max_k \text{Gap}(k)$$

Hastie, Tibshirani, Botstein, Brown (2001) Supervised harvesting of expression trees. *Genome Biology* 2(1)

Quantitative Variable y (z.B. Lebenserwartung von Tumorpatienten) sei aus Genexpressionsdaten vorherzusagen.

1. Wende hierarchisches Clusterverfahren auf die p Gene an; seien c_1, \dots, c_{2p-1} alle auftretenden Cluster,

$$\bar{x}_{c_k} = (\bar{x}_{1,c_k}, \dots, \bar{x}_{n,c_k}) \text{ mit } \bar{x}_{i,c_k} = \sum_{j \in c_k} x_{ij} / |c_k|.$$

2. $\mathcal{M} := \{(1, \dots, 1)\}$ (Menge von Vektoren aus \mathbb{R}^n)

3. Wähle ein $a \in \mathcal{M}$ und ein c_k und setze

$\mathcal{M} := \mathcal{M} \cup \{(a_1 \cdot \bar{x}_{1,c_k}, \dots, a_n \cdot \bar{x}_{n,c_k})\}$, wobei a und c_k so gewählt werden, so dass

$$\hat{y} = \sum_{z \in \mathcal{M}} \beta_z \cdot z$$

bei optimaler Wahl der Koeffizienten β_z möglichst gut y vorhersagt, also z.B. $\sum_{i=1}^n (y_i - \hat{y}_i)^2$ minimiert.

4. Wiederhole m mal Schritt 3, wobei optimales m durch Kreuzvalidierung bestimmt wird.

Efron, Hastie, Johnstone, Tibshirani (2004) Least Angle Regression *Annals of Statistics* 32(2)

Aus einer Menge von Vektoren $x_1, x_2, \dots, x_p \in \mathbb{R}^n$ (z.B. Expressionswerte von p Genen aus n Versuchen) sind die auszuwählen, die eine Variable y möglichst gut linear vorhersagen, d.h. gesucht ist $m, a_1, \dots, a_m, \beta_0, \beta_1, \dots, \beta_m$, so dass

$$\hat{y}_i = \beta_0 + \sum_{k=1}^m \beta_k x_{a_k i}$$

nahe an y_i ist ein neues y_{n+1} gut aus $x_{a_1 n+1}, x_{a_2 n+1}, \dots, x_{a_m n+1}$ vorhersagen kann.

Ziel von **Least Angle Regression (LARS)** ist, dass einzelne x_i das Ergebnis nicht zu stark beeinflussen sollten, um somit robust gegenüber Ausreißern zu sein (Also dasselbe wie bei Ridge Regression, LASSO, Gene Shaving, Harvesting, ...).

LARS

1. Wähle als x_{a_1} das x_i , das am stärksten mit y korreliert ist.
2. Wähle β_0, γ_1 so, dass es eine Variable x_{a_2} gibt, so dass

$$y - (\beta_0 + \gamma_1 x_{a_1})$$

mit einem x_{a_2} genauso stark wie mit x_{a_1} korreliert ist.

3. Wähle γ_2 so, dass es eine Variable x_{a_3} gibt, so dass

$$y - (\beta_0 + \gamma_1 x_{a_1} + \gamma_2 (x_{a_1} / \|x_{a_1}\| + x_{a_2} / \|x_{a_2}\|))$$

mit einem x_{a_3} genauso stark wie mit x_{a_1} und x_{a_2} korreliert ist.

und so weiter. . . in Worten: Gehe jeweils nur so lange in die richtige Richtung bis es einen Vektor gibt, der genauso gut ist, und fahre dann mit der mittleren Richtung fort.

Segal, Dahlquist, Conklin (2003) Regression Approaches for Microarray Data Analysis. *Journal of Computational Biology* 10(6)

Vergleichen Harvesting, LASSO, LARS, SVMs an Beispieldatensatz für transgene Mäuse.

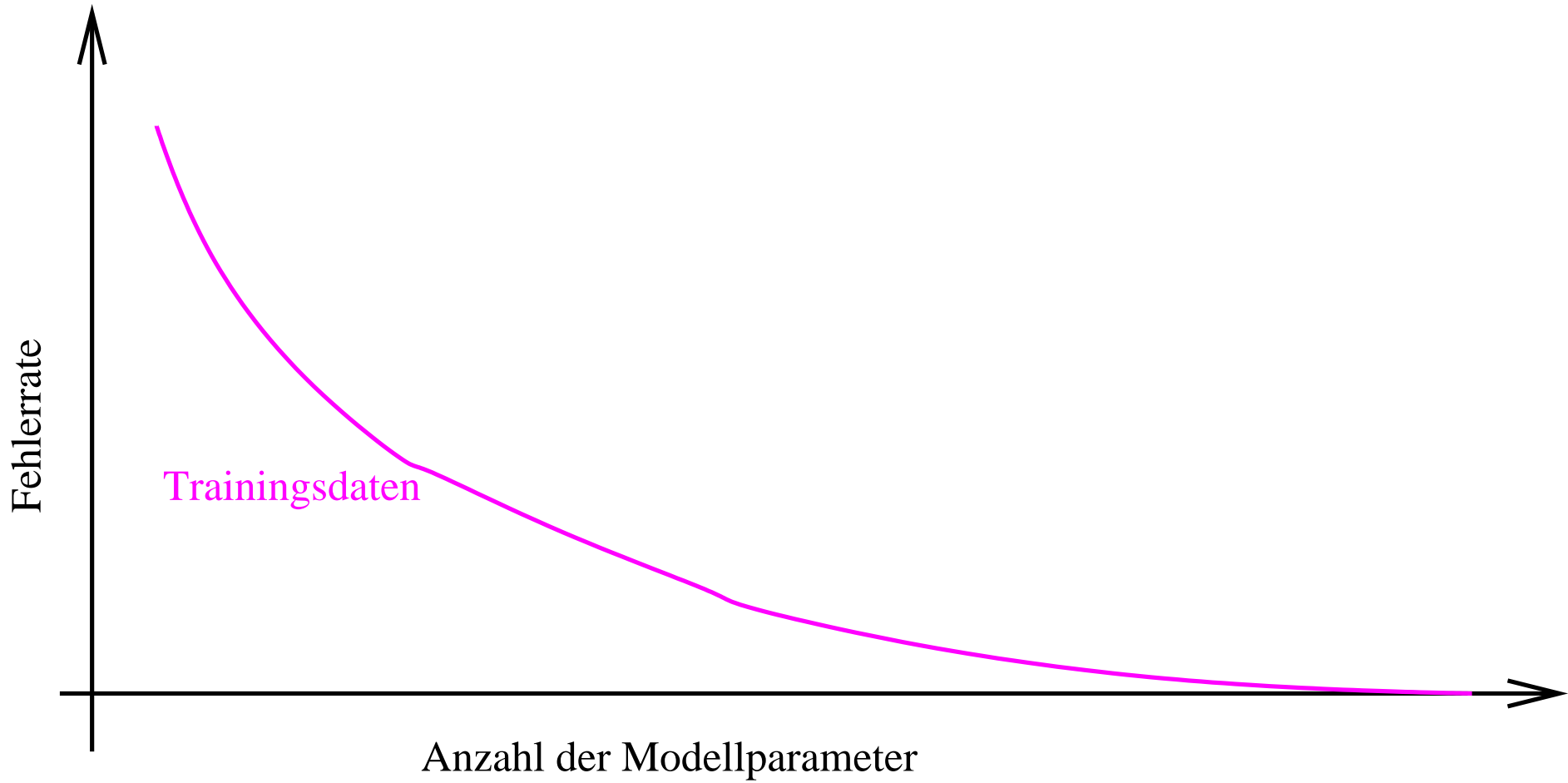
Wie hängt die Expression y des eingefügten Gens mit der Expression anderer Gene zusammen?

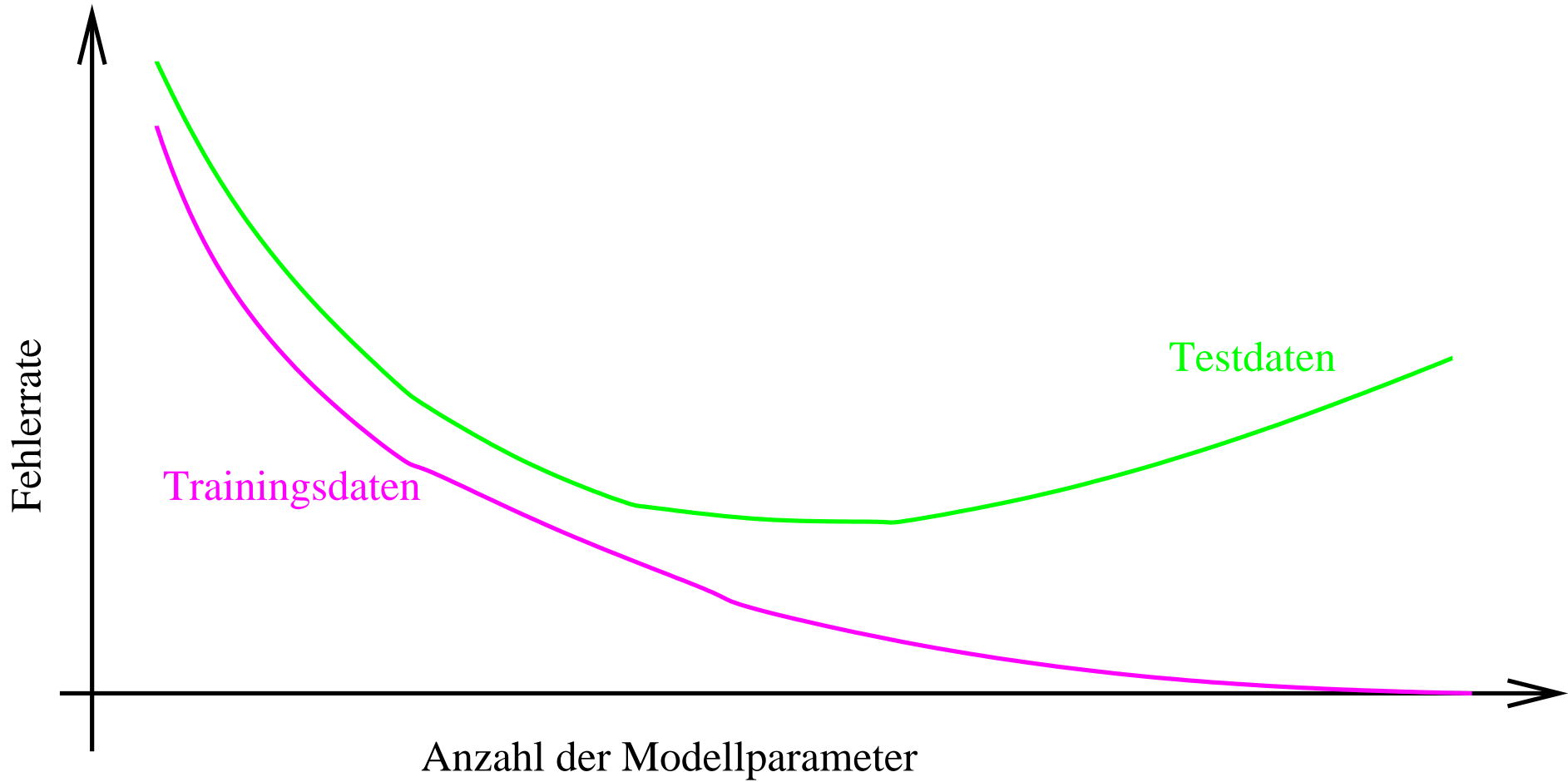
Ergebnisse:

Die Performance von Gene Harvesting hängt sehr davon ab, welches Cluster-Verfahren verwendet wird.

Direkte Anwendung von Gene Harvesting führte zu Artefakten, Varianten liefen besser, LASSO, LARS und lineare SVM noch besser.

Die eigentliche Idee von SVM ist allerdings bei Microarray-Analysen nicht relevant.





Wie gut das Modell M auf die Daten D passt, kann man mit der Likelihood des Modells

$$L_D(M) = W_{S_M}(D) \quad \text{oder auch} \quad \log L_D(M)$$

messen.

Wie gut das Modell M auf die Daten D passt, kann man mit der Likelihood des Modells

$$L_D(M) = W_{S_M}(D) \quad \text{oder auch} \quad \log L_D(M)$$

messen.

Je größer die Anzahl d der Parameter des Modells M , desto größer das Risiko, des "Overfittig".

Wie gut das Modell M auf die Daten D passt, kann man mit der Likelihood des Modells

$$L_D(M) = W_{S_M}(D) \quad \text{oder auch} \quad \log L_D(M)$$

messen.

Je größer die Anzahl d der Parameter des Modells M , desto größer das Risiko, des “Overfittig”.

Man kann den zu erwartenden **Klassifikationsfehler** schätzen durch **Akaikes Informations-Kriterium**:

$$\text{AIC} = -2 \log L_D(M) + 2d$$

Wie gut das Modell M auf die Daten D passt, kann man mit der Likelihood des Modells

$$L_D(M) = W_{S_M}(D) \quad \text{oder auch} \quad \log L_D(M)$$

messen.

Je größer die Anzahl d der Parameter des Modells M , desto größer das Risiko, des “Overfittig”.

Man kann den zu erwartenden **Klassifikationsfehler** schätzen durch **Akaikes Informations-Kriterium**:

$$\text{AIC} = -2 \log L_D(M) + 2d$$

Begründbar sinnvoll ist das zumindest unter Normalverteilungsannahmen.

Alternative: Bayes Informations-Kriterium

$$BIC = -2 \log L_D(M) + d \log n$$

n ist die Anzahl der unabhängigen Messungen.

Alternative: Bayes Informations-Kriterium

$$BIC = -2 \log L_D(M) + d \log n$$

n ist die Anzahl der unabhängigen Messungen.

Ansatz: alle Modelle sind *a priori* gleich wahrscheinlich.

Alternative: Bayes Informations-Kriterium

$$BIC = -2 \log L_D(M) + d \log n$$

n ist die Anzahl der unabhängigen Messungen.

Ansatz: alle Modelle sind *a priori* gleich wahrscheinlich.

Die *a posteriori* Wahrscheinlichkeit des Modells M ist

$$Ws(M | D) = \frac{Ws(M, D)}{Ws(D)} = \frac{Ws(D | M) \cdot Ws(M)}{Ws(D)}$$

Alternative: Bayes Informations-Kriterium

$$BIC = -2 \log L_D(M) + d \log n$$

n ist die Anzahl der unabhängigen Messungen.

Ansatz: alle Modelle sind *a priori* gleich wahrscheinlich.

Die *a posteriori* Wahrscheinlichkeit des Modells M ist

$$\text{Ws}(M | D) = \frac{\text{Ws}(M, D)}{\text{Ws}(D)} = \frac{\text{Ws}(D | M) \cdot \text{Ws}(M)}{\text{Ws}(D)}$$

unter gewissen Annahmen gilt:

$$\log \frac{\text{Ws}(M_1 | D)}{\text{Ws}(M_2 | D)} \approx (2 \log L_D(M_1) - d_1 \log n) - (2 \log L_D(M_2) - d_2 \log n)$$

Also: Wähle ein Modell mit möglichst geringem AIC oder BIC.

$$AIC = -2 \log L_D(M) + 2d$$

$$BIC = -2 \log L_D(M) + d \log n$$

Also: Wähle ein Modell mit möglichst geringem AIC oder BIC.

$$AIC = -2 \log L_D(M) + 2d$$

$$BIC = -2 \log L_D(M) + d \log n$$

Beobachtung: BIC bevorzugt im Vergleich zu AIC die einfacheren Modelle.

Noch eine Alternative: Likelihood-Quotienten Test (LRT)

Zum Vergleich zweier Modelle M_1 und M_2 , die “nested” sind, d.h. M_1 ist ein Spezialfall von M_2 , d.h. man kann M_1 aus M_2 erhalten, indem man bestimmte Parameter auf gewisse Werte festlegt.

Fast immer wird M_2 besser auf die Daten passen als M_1 , denn allgemein gilt $W_{S_{M_1}}(D) \leq W_{S_{M_2}}(D)$.

Noch eine Alternative: Likelihood-Quotienten Test (LRT)

Zum Vergleich zweier Modelle M_1 und M_2 , die “nested” sind, d.h. M_1 ist ein Spezialfall von M_2 , d.h. man kann M_1 aus M_2 erhalten, indem man bestimmte Parameter auf gewisse Werte festlegt.

Fast immer wird M_2 besser auf die Daten passen als M_1 , denn allgemein gilt $W_{S_{M_1}}(D) \leq W_{S_{M_2}}(D)$.

Signifikant besser???

Noch eine Alternative: Likelihood-Quotienten Test (LRT)

Zum Vergleich zweier Modelle M_1 und M_2 , die “nested” sind, d.h. M_1 ist ein Spezialfall von M_2 , d.h. man kann M_1 aus M_2 erhalten, indem man bestimmte Parameter auf gewisse Werte festlegt.

Fast immer wird M_2 besser auf die Daten passen als M_1 , denn allgemein gilt $W_{S_{M_1}}(D) \leq W_{S_{M_2}}(D)$.

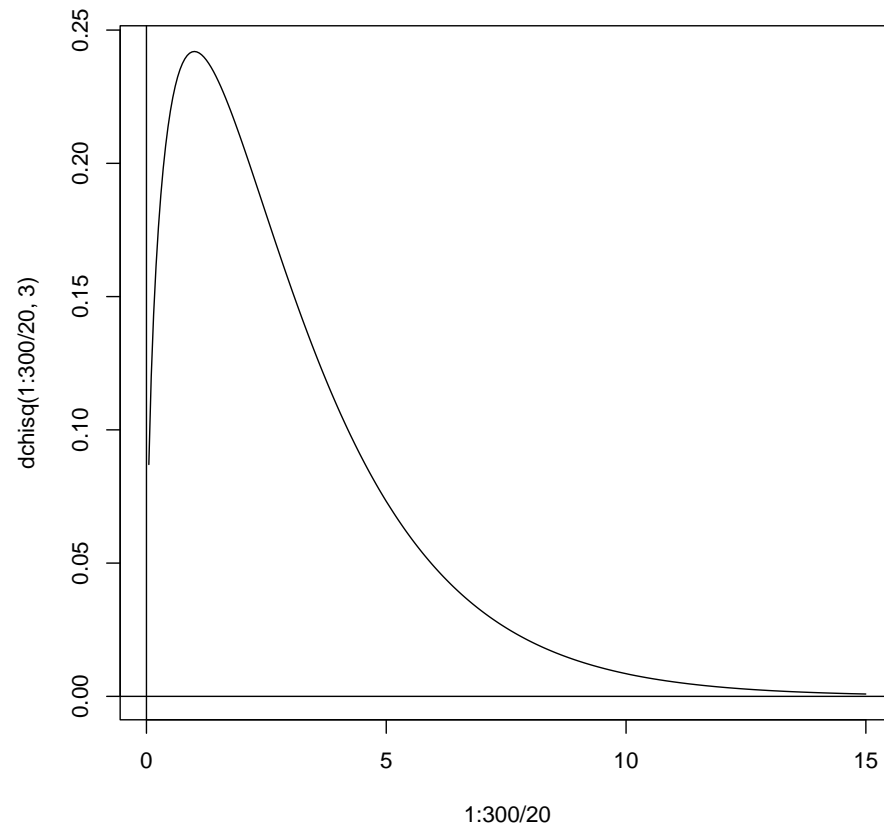
Signifikant besser???

Likelihood-Ratio-Test:

$$\text{Teststatistik: } \frac{L_D(M_2)}{L_D(M_1)} \quad (\text{mit } L_D(M) := W_{S_M}(D))$$

Für große Datensätze gilt

$$\mathcal{L}_{M_1} \left(2 \cdot \log \frac{L_D(M_2)}{L_D(M_1)} \right) \sim \chi^2_{\#\{\text{zusätzliche Parameter}\}}$$



ansonsten: Simulationsstudien....

Modellwahl sollte abhängen von...

- Komplexität des modellierten Systems, die sich in den Daten spiegelt
- Größe des Datensatzes
- **Fragestellung**

Vorschlag:

Probiere mehrere Modelle M_1, M_2, \dots, M_k aus, beginne mit sehr einfachen.

Sei $\hat{\theta}_i$ das Ergebnis bei Verwendung von M_i .

Überprüfe mit Simulationsstudien: Wenn $\hat{\theta}_i$ die Wahrheit wäre, wie gut könnte man dann mit M_j schätzen (auch bei $i = j$)?