Seminararbeit

Computationale Wende, Paradigmenwechsel oder Sturm im Wasserglas?

Ronert Obst

26. Februar 2013

Inhaltsverzeichnis

1 Einführung

> *A very large share of the advances of statistical theory during the next eighth [decade] are going to depend upon the computer in an essential way.*
>
> – Tukey (1965)

Die Computationale Wende hat enorme Veränderungen in der Statistik hervorgebracht. Ich möchte nicht nur den Wandel in der Praxis beschreiben, welchen es in fast jedem Gebiet durch den Computer gab, sondern auch den Paradigmenwechsel, welcher durch diese Veränderung hervorgerufen wurde. In der Statistik hat man es mit „schmutzigen“ Daten zu tun. Anders als in der Mathematik, die nach ewig gültigen Sachverhalten strebt, werden in der Statistik bestmögliche Modelle als Annäherung an die unbekannte Realität gesucht. Wie Box und Draper (1987) es formuliert haben:

> *all models are wrong, but some are useful.*

Deshalb gibt es in der Statistik eine Wechselwirkung zwischen der Theorie und dem praktisch Machbaren, soll heißen: zwischen der theoretischen und computationalen Statistik. Das Internet, die Entschlüsselung des menschlichen Genoms und andere Fortschritte, die erst durch den Computer möglich wurden, haben auch ganz neue Datensätze und Anforderungen für Statistiker hervorgebracht. Da die theoretische Entwicklung der Statistik immer auch von den Anforderungen an sie getrieben wurde, hat die Computationale Wende auf diese Weise auch indirekt das Paradigma der Statistik verschoben.

Die Computationale Wende hat mit den *Markov Chain Monte Carlo* (MCMC) Methoden begonnen, weshalb ich die Seminararbeit auch mit diesem Thema beginnen möchte. MCMC Methoden dienen dazu numerisch Integrale zu approximieren, die analytisch nicht leicht bestimmbar sind. Sie wurden Rahmen des Manhattan Projekts in Los Alamos in den 1940ern von den Wissenschaftlern um John von Neumann, vor allem Stanislaw Ulam, entwickelt (Watnik 2011). Zufallszahlen am Computer generieren zu können, war entscheidend für die Anwendung der MCMC Methoden und John von Neumann hat auf diesem Gebiet ebenfalls einen wichtigen Beitrag geliefert (Neumann 1951).

In den 1950ern waren Computer viel zu teuer und daher wenig verbreitet, um MCMC Methoden zum Durchbruch in der „mainstream“ Statistik zu verhelfen. Dies sollte erst Anfang der 1990er mit dem Paper von Gelfand und Smith (1990) und der weiten Verbreitung von preiswerten und leistungsstarken Computern geschehen – der sogenannten „zweiten MCMC Revolution“ (C. Robert und Casella 2011). Dieser Durchbruch verhalf auch der Bayesschen Statistik zu ihrem Aufstieg in den „mainstream“ der Statistik, da nun viele Bayessche Modelle erst praktikabel wurden. Dies hatte einen enormen Anstieg an theoretischen Arbeiten auf diesem Gebiet zur Folge, den es ohne die Computationale Wende nicht gegeben hätte. Sicherlich ein Gebiet der Statistik, wo die Computationale Wende ihren größten Einfluss auf das vorherrschende Paradigma hatte.

Im Dezember 1966 sponsorte die Royal Statistical Society die erste Konferenz zur computationalen Statistik, die seitdem regelmäßig stattfindet. Auch waren Computer, zumindest an den Universitäten, mittlerweile für die Forschung verfügbar (Gentle, Mori und Härdle 2004). 1967 erschien dann auch das erste Buch über computationale Statistik, Hemmerle (1967). Eine wichtige Entwicklung aus dieser Zeit waren *Simulationsstudien*, die ich als Nächstes betrachten werde. Die Idee hinter diesen Verfahren ist es, Inferenz auch dann betreiben zu können,

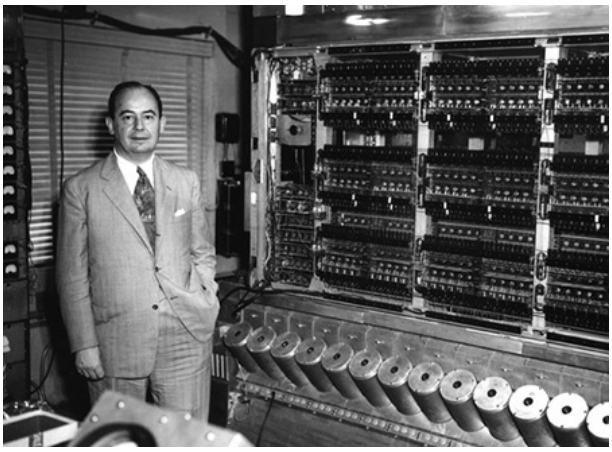

Abbildung 1: John von Neumann in Los Alamos

wenn ein analytisches Ergebnis nicht praktikabel ist. Man simuliert dabei Zufallszahlen aus einer Verteilung und ersetzt die analytische Inferenz durch ihre empirischen Analoga. Eng verwandt mit diesem Ansatz sind die so genannten *Resampling-Verfahren*, die auf dem selben Prinzip beruhen, aber anstatt aus einer Verteilung zu ziehen, werden hierbei Daten aus einer Stichprobe wiederholt perturbiert. Das Ziel ist dabei, Inferenz über ein Modell oder eine Statistik zu betreiben, wenn analytische Berechnungen sehr kompliziert oder unmöglich sind.

Als Nächstes werden wir uns anschauen, wie statistische Software, insbesondere das R-Projekt (R Core Team 2012), die Statistik beeinflusst hat. Statistische Software bestimmt letztendlich, welche Verfahren in der Praxis genutzt werden. Gerade *interaktive* Programmiersprachen wie R haben aber auch auf subtilere Art und Weise beeinflusst, wie wir Statistik betreiben. Nicht nur, dass komplexere, nichtlineare Modelle genutzt werden können, sondern es ist auch möglich wesentlich mehr Modelle auszuprobieren. Die explorative Analyse von Daten, wie sie Tukey (1970) vorschlug, ist mittlerweile Routine für Statistiker.

Machine Learning etablierte sich in den 1980ern als parallele Disziplin zur Datenanalyse. Sie warf alle alten Paradigmen über Bord und fokussierte sich komplett auf die neuen Möglichkeiten, die einem Computer zur Datenanalyse boten. Ihr einziges Ziel ist es, möglichst akkurate out-of-sample Vorhersagen zu generieren, ohne restriktive Annahmen über die Daten zu treffen. Dabei wird die Rechenkraft modernster Computersysteme genutzt, um algorithmisch funktionale Zusammenhänge zu finden, oftmals in hohen Dimensionen. Mittlerweile ist die Disziplin erwachsen geworden und produziert viele theoretisch fundierte Arbeiten. Die Statistik und das Machine Learning befruchten sich dabei gegenseitig und es wird interessant zu beobachten, inwieweit die beiden Disziplinen konvergieren werden.

Als Letztes werden wir uns die Frage stellen, ob zur Zeit eine neue Computationale Wende stattfindet. Die Größe einiger Datensätze die vom Internet, von Genomanalysen und Teilchenbeschleunigern wie dem LHC – um nur einige zu nennen – erzeugt werden, wachsen um einiges schneller als die Rechenkraft unserer Computer nach dem Mooreschen Gesetz (Gantz und Reinsel 2011). Es entsteht ein immer größerer Bedarf, dass statistische Verfahren *parallelisierbar* werden, um diese Datenmengen zu bewältigen. Wird dieser Umstand auch die theoretische Entwicklung beeinflussen?

2 Markov Chain Monte Carlo und Bayessche Statistik

2.1 Die Anfänge: Das Manhattan Projekt

Stanislaw Ulam kam die Idee zur Monte Carlo Methode beim Spielen von Solitär in 1946 in Los Alamos (Watnik 2011). Die Gewinnwahrscheinlichkeit eines Solitär-Spieles zu bestimmen, ist analytisch unlösbar. Die Idee von Ulam war, dass wenn eine große Anzahl an Spielen simuliert werden könnte, lässt sich die Gewinnwahrscheinlichkeit approximativ ermitteln. Generell ist die Idee hinter dem Monte Carlo Verfahren, analytisch schwer bestimmbare Integrale durch stochastische Simulation zu approximieren. Der skurrile Name entstand (und ist haften geblieben), da Ulam einen spielenden Onkel hatte, der sich immer Geld von ihm lieh, wenn er „nach Monte Carlo ging“ (Metropolis 1987a). Markov Chain bzw. Markov-Kette kommt daher, dass die simulierten Zustände nicht unabhängig sind, sondern eine Markov-Kette bilden. Enrico Fermi arbeitete mit dieser Methode bereits in den 1930ern ohne Computer während seiner schlaflosen Nächte. Er publizierte aber nie, da er lieber seine Kollegen in Rom mit seinen Ergebnissen verblüffte (Watnik 2011).

Derartige Simulationstechniken waren Statistikern schon länger bekannt, fanden aber keine Anwendung, da deren Implementierung zu aufwendig ohne Computer war. Ulam´s Einfall war in diesem Sinne keine neue Idee, aber er war der Erste, der das Verfahren an einem digitalen Computer implementieren konnte und es damit praktisch nutzbar machte (Metropolis 1987b). Er hatte aber auch das Glück, Zugang zu dem Ersten digitalen Computer überhaupt zu haben, dem ENIAC (Electronic Numerical Integrator And Computer).

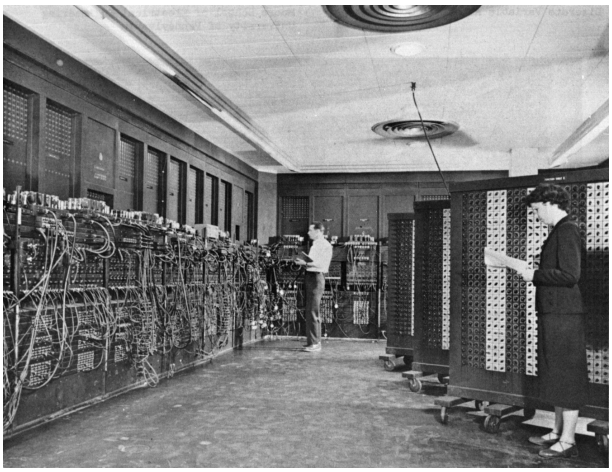

Abbildung 2: ENIAC in Los Alamos

Um ein stochastisches Verfahren wie Monte Carlo Simulati-

on anwenden zu können, braucht man erstmal ein stochastisches Element – Zufallszahlen. Neumann (1951) leistete auch auf diesem Gebiet Pionierarbeit. Computer sind von ihrem Wesen her rein deterministisch. Wenn man es sich näher überlegt, ist es überhaupt nicht trivial ein Programm zu erstellen, welches Zufallszahlen ziehen kann. Es werden deshalb deterministische Algorithmen verwendet, um Zufallszahlen zu *simulieren*. Die damit gezogenen Zufallszahlen sind zwar deterministisch – verhalten sich aber wie zufällig gezogene Zahlen. Neumann (1951) realisierte, dass wenn

1. Unabhängig und identisch verteilte Zufallszahlen U aus einer Gleichverteilung auf dem Intervall $[0,1]$ simuliert werden können,

2. lassen sich über die Inverse Verteilungsfunktion

$$X = F^{-1}(U) \stackrel{\text{def}}{=} \min\{x \mid F(x) \geq U\}$$

Zufallszahlen X aus einer beliebigen Verteilungsfunktion F simulieren.

Neumann (1951) unterscheidet zwei Möglichkeiten, Zufallszahlen aus einer Gleichverteilung zu ziehen:

1. Physische Quellen (z.B. radioaktiver Zerfall oder kosmische Hintergrundstrahlung);

2. Arithmetische Berechnung (ein rekursiver Algorithmus erzeugt eine Folge deterministischer Zufallszahlen mit den gleichen Eigenschaften wie eine Folge unabhängig identisch verteilter Zufallsvariablen).

Physische Quellen ergaben sich aber als problematisch (Gentle, Mori und Härdle 2004, S. 37), da

- sie wesentlich schwieriger zu betreiben sind;
- kosten mehr;
- langsamer sind;
- nicht reproduzierbar sind.

Deshalb entschied man sich für die zweite Möglichkeit und implementierte die so genannten *Multiplikativen Kongruenzegeneratoren*, die in ihrer einfachsten Form (1. Ordnung) auf der rekursiven Beziehung

$$x_{i+1} = ax_i \mod m$$

basieren. Seien $a, m \in \mathbb{N}$. Dann kann gezeigt werden, dass wenn

$$a^{m-1} = 1 \mod m, \; a^k \neq 1 \mod m, \; \forall 0 < k < m-1$$

und x_0 kein Vielfaches von m ist, die Folge $x_1, \ldots, x_{m-1}$ eine Permutation der Zahlen $\{1, \ldots, m-1\}$ ist (Müller 2011). Mitte der 1950er Jahren hatten die meisten Computer einen Zufallsgenerator eingebaut. Nur leider benutzten alle die selbe FORTRAN Bibliothek, die einen festen Startwert vorgab. So beruhten fast alle Simulationen zu dieser Zeit auf der selben Folge von Pseudozufallszahlen (Teichroew 1965).

2.2 Die „Zweite Markov Chain Monte Carlo Revolution“ und die Bayessche Statistik

MCMC Methoden haben erst Anfang der 1990er mit dem Paper von Gelfand und Smith (1990) im „mainstream“ der Statistik Fuß gefasst – die „zweite Markov Chain Monte Carlo Revolution“ (C. Robert und Casella 2011). MCMC Methoden wurden bereits in dem Paper von Hastings (1970) in ihrer jetzigen Grundform publiziert. Ein Mangel an computationalen Resourcen (man Denke an Computer in den 1970ern) und schwer zu lernende Programmiersprachen wie FORTRAN, waren sicher Gründe, weshalb der Durchbruch von MCMC und damit der Bayesschen Statistik erst in den 1990ern kam.

In den folgenden Jahren gab es geradezu eine Explosion an Forschungsarbeit in MCMC Methoden und der Bayesschen Statistik. 1991 erschien bereits die erste Software die MCMC Methoden zur Inferenz in der Bayesschen Statistik implementierte: BUGS (**B**ayesian inference **U**sing **G**ibbs **S**ampling) (C. Robert und Casella 2011). Viele Probleme, die davor praktisch unmöglich erschienen, waren auf einmal lösbar. Insbesondere war nun Inferenz in Bayessche Modelle mit nicht-konjugierter Prioriverteilung möglich. Aber auch Bayessche Variablenselektion (George und McCulloch 1993) und Modellselektion (Madigan und Raftery 1994) wurden populär.

Andrieu, Doucet und C. P. Robert (2004, S. 121) beschreiben den Einfluss der Computationalen Wende auf die Bayessche Statistik:

> *The prodigious advances made by Bayesian analysis in methodological and applied directions during the previous decade have been made possible only by advances of the same scale in computing abilities with, at the forefront, Markov chain Monte Carlo methods […] Many things happened in Bayesian analysis because of MCMC and, conversly many features of MCMC are only there because of Bayesian analysis! We think the current state of Bayesian analysis would not have been reached without MCMC techniques and also that the upward surge in the level of complexity of the models analyzed by Bayesian methods contributed to the very fast improvement in MCMC methods.*

Die Bayessche Statistik ist das Paradebeispiel dafür, wie sich das Paradigma der Statistik durch die technische Entwicklung der Computer wandelt. C. Robert und Casella (2011, S. 110) fassen dies noch einmal zusammen:

> *The impact of Gibbs sampling and MCMC was to change our entire method of thinking and attacking problems, representing a paradigm shift (Kuhn 1996).*

Gibbs-Sampling

MCMC Methoden sind mittlerweile so zahlreich und komplex, dass ich hier beispielhaft nur auf einen einfacheren (aber wichtigen) Speziallfall eingehen möchte: *Gibbs sampling*. Für andere MCMC Methoden verweise ich auf C. Robert und Casel-

la (2010). Gibbs-Sampling wurde ursprünglich von Physikern verwendet um *Gibbs random fields* zu untersuchen und wurde naheliegenderweise nach dem Physiker Josiah Willard Gibbs benannt (S. Geman und D. Geman 1984). Das bereits erwähnte Paper von Gelfand und Smith (1990) brachte dann endlich den Durchbruch in der Statistik. Die folgende Erklärung des Gibbs-Samplers richtet sich nach Casella und George (1992).

Der Gibbs-Sampler ist eine Methode, um indirekt Zufallszahlen aus einer unbekannten Verteilung zu ziehen, basierend auf elementaren Eigenschaften von Markov-Ketten. Man nehme an, uns ist eine gemeinsame Dichte $f(x, y_1, \dots, y_p)$ gegeben und wir möchten z.B. den Mittlerwert oder die Varianz der marginalen Dichte

$$f(x) = \int \dots \int f(x, y_1, \dots, y_p)\, dy_1 \dots dy_p$$

bestimmen. Man könnte die marginale Dichte analytisch oder numerisch bestimmen, was aber oft zu schwierig ist. In diesen Fällen kann man den Gibbs-Sampler verwenden, um $f(x)$ zu bestimmen.

Anstatt $f(x)$ direkt zu berechnen, erlaubt es uns der Gibbs-Sampler Zufallszahlen $X_1, \dots, X_m \sim f(x)$ zu ziehen. Asymptotisch konvergieren die so errechneten empirischen Momente gegen die theoretischen. Man betrachte z.B. den empirischen Mittelwert

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} X_i = \int_{-\infty}^{\infty} x f(x)\, dx = \mathbb{E}\left[X\right].$$

Seien nun (X,Y) Zufallsvariablen und wir möchten aus der marginalen Dichte $f(x)$ Zufallszahlen ziehen. Gibbs-Sampling funktioniert, in dem wir abwechselnd aus den vollständig bedingten Dichten

$$X'_j \sim f(x \,|\, Y'_j = y'_j)$$
$$Y'_{j+1} \sim f(y \,|\, X'_j = x'_j),$$

gegeben einem Startwert $Y'_0 = y'_0$, ziehen. Die so erstellte *Gibbs Folge*

$$X'_0, X'_1, X'_2, \dots, X'_n \tag{1}$$

konvergiert gegen $f(x)$ (die echte Marginalverteilung) mit $n \to \infty$. $X'_n = x'_n$ sind somit Zufallszahlen aus $f(x)$.

Weshalb konvergiert der Gibbs-Sampler gegen die marginale Verteilung? Nehmen wir als Beispiel eine multinomialverteilte 2×2 Tafel. Nehmen wir weiter an, X und Y sind beide marginal Bernoulli-verteilt mit gemeinsamer Verteilung

		X	
		0	1
Y	0	p_1	p_2
	1	p_3	p_4

mit

$$p_i \geq 0,\ p_1 + p_2 + p_3 + p_4 = 1.$$

Die gemeinsamen Wahrscheinlichkeiten ergeben

$$\begin{bmatrix} f_{x,y}(0,0) & f_{x,y}(1,0) \\ f_{x,y}(0,1) & f_{x,y}(1,1) \end{bmatrix} = \begin{bmatrix} p_1 & p_2 \\ p_3 & p_4. \end{bmatrix}$$

Hier ergibt die marginale Dichte von x

$$f_x = \begin{bmatrix} f_x(0) & f_x(1) \end{bmatrix} = \begin{bmatrix} p_1 + p_3 & p_2 + p_4 \end{bmatrix} \tag{2}$$

eine Bernoulliverteilung mit Erfolgswahrscheinlichkeit $p_2 + p_4$. Alle bedingten Wahrscheinlichkeiten lassen sich in zwei Matrizen darstellen

$$A_{y\,|\,x} = \begin{bmatrix} \frac{p_1}{p_1+p_3} & \frac{p_3}{p_1+p_3} \\ \frac{p_2}{p_2+p_4} & \frac{p_4}{p_2+p_4} \end{bmatrix}$$

und

$$A_{x\,|\,y} = \begin{bmatrix} \frac{p_1}{p_1+p_2} & \frac{p_2}{p_1+p_2} \\ \frac{p_3}{p_3+p_4} & \frac{p_4}{p_3+p_4} \end{bmatrix},$$

mit $A_{y\,|\,x}$, den bedingten Wahrscheinlichkeiten von Y gegeben $X = x$ und $A_{y\,|\,x}$, den bedingten Wahrscheinlichkeiten von X gegeben $Y = y$. Wenn wir nun die marginale Verteilung von X bestimmen wollen, interessiert uns die Folge von X' aus Gleichung 1. Um von $X'_0 \to X'_1$ zu kommen, muss man über Y'_1 gehen, also die Folge $X'_0 \to Y'_1 \to X'_1$. Dann bildet $X'_0 \to X'_1$ eine Markov-Kette mit Übergangswahrscheinlichkeit

$$\mathbb{P}\left[X'_1 = x_1 \,|\, X'_0 = x_0\right] = \sum_y \mathbb{P}\left[X'_1 = x_1 \,|\, Y'_1 = y\right] \times \mathbb{P}\left[Y'_1 = y \,|\, X'_0 = x_0\right].$$

Die Übergangsmatrix von der Folge X', $A_{x\,|\,x}$ ist

$$A_{x\,|\,x} = A_{y\,|\,x} A_{x\,|\,y},$$

so dass sich die Wahrscheinlichkeitsverteilung jedes Elementes X'_k der Folge über $\left(A_{x\,|\,x}\right)^k$ ausrechnen lässt. Sei

$$f_k = \begin{bmatrix} f_k(0) & f_k(1) \end{bmatrix}$$

die marginale Verteilung von X'_k, dann ist für jedes k,

$$f_k = f_0 A^k_{x\,|\,x} = \left(f_0 A^{k-1}_{x\,|\,x}\right) A_{x\,|\,x} = f_{k-1} A_{x\,|\,x}. \tag{3}$$

Solange alle Elemente von $A_{x\,|\,x}$ positiv sind, impliziert Gleichung 3 dass für einen beliebigen Startwert $f_0 \in [0,1]$,

$$f_k \overset{k\to\infty}{\longrightarrow} f$$

konvergiert und

$$f A_{x\,|\,x} = f \tag{4}$$

erfüllt. Wenn die Gibbs Folge konvergiert, muss das f welches Bedingung 4 erfüllt, die marginale Verteilung von X sein. Es ist leicht zu überprüfen, dass f_x aus Gleichung 2 die Bedingung 4 erfüllt:

$$f_x A_{x\,|\,x} = f_x A_{y\,|\,x} A_{x\,|\,y} = f_x.$$

Mit $k \to \infty$ konvergiert die Verteilung von X'_k gegen die von f_x, so dass man mit einem großen k mit Gibbs-Sampling Zufallszahlen aus der marginalen Verteilung f_x ziehen kann.

3 Simulation und Resampling

3.1 Simulation

Ein weiterer Effekt der Forschungsarbeit in Los Alamos und der Verfügbarkeit von digitalen Computern an den Universitäten war das zahlreiche Aufkommen von Simulationsstudien um bisher analytisch unlösbare Probleme empirisch zu knacken (Teichroew 1965). Die Idee analytisch schwer zu bestimmende Größen durch ihre empirischen Analoga zu ersetzen war nicht neu, aber durch die neuen Methoden lange Folgen an Pseudozufallszahlen mit digitalen Computern zu erstellen und mit der inversen Verteilungsfunktion damit aus beliebigen Verteilungen zu ziehen, machte diese Methoden erst wirklich praktikabel.

Die erste große systematische Simulationsstudie in der Statistik war die Princeton Robustness Study um Tukey in 1970 im Rahmen eines Seminars über Robuste Statistik, woraus ein Buch entstand (Andrews u. a. 1972). Diese Studie hatte eine Lawine an weiteren Simulationsstudien zur Folge (Stigler 2010). 1972 wurde daraufhin das *Journal of Statistical Computation and Simulation* gegründet. Seitdem sind Simulationsstudien ein fester Bestandteil statistischer Publikationen.

3.2 Kreuzvalidierung

Kreuzvalidierung entstand, wie die Simulation, vor der Computationalen Wende, wurde aber durch sie erst brauchbar. Die Motivation der Kreuzvalidierung ist es, den *Testfehler*, also den out-of-sample Vorhersagefehler, zu schätzen (der Fehler, der realistischerweise zu erwarten wäre, wenn man ein Modell auf neue Daten zur Vorhersage anwendet). Die Modellwahl ist meistens das Ziel. Das Prinzip der Kreuzvalidierung ist in Algorithmus 1 skizziert. Gebräuchliche Werte für K sind 5 und

Algorithmus 1 : K-fache Kreuzvalidierung (Stone 1974)

Man teile den Datensatz in K etwa gleich große Teile auf
for k *in* $1 : K$ **do**
1. Schätze das Modell auf alle bis auf den $k$ten Teil, also auf $K - 1$ Teile
2. Berechne den Vorhersagefehler für die Vorhersage des $k$ten Teils

end
Berechne den durchschnittlichen Vorhersagefehler der K Teile

10. Um die Variabilität der Selektion der K Teile zu beseitigen, ist es sinnvoll die K-fache Kreuzvalidierung mit zufälliger Wahl der Split-Punkte mehrfach zu wiederholen und die Ergebnisse zu mitteln. Kreuzvalidierung ist eine häufig genutzte Alternative zu den Informationskriterien und ist auch dann anwendbar, wenn Informationskriterien nicht definiert sind.

3.3 Jackknife

1958 führte Tukey den sogenannten *Jackknife* ein. Ein Jackknife ist ein Klappmesser, was symbolisieren soll, dass diese Methode in allen erdenklichen Situationen und Notfällen nutzbar ist (Miller 1964). Die Idee hinter dem Jackknife ist es, Standardfehler und Bias einer Statistik zu schätzen, wenn diese analytisch nicht leicht zu bestimmen sind. Folgende Einführung stammt aus Efron und R. J. Tibshirani (1994).

Sei $\boldsymbol{X} = (x_1, x_2, \ldots, x_n)$ eine Stichprobe und $\hat{\theta} = s(\boldsymbol{X})$ die uns interessierende Statistik. Wir wollen nun den Bias und die Varianz von $\hat{\theta}$ bestimmen. Der Jackknife berechnet die gewünschte Statistik n-mal und lässt dabei jedes mal genau *einen* Wert aus der Original-Stichprobe aus. Die Jackknife-Stichproben sehen also so aus:

$$\boldsymbol{X}_{(i)} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$$

für $i = 1, \ldots, n$. Sei

$$\hat{\theta}_{(i)} = s(\boldsymbol{X}_{(i)})$$

die $i$te Jackknife-Schätzung von $\hat{\theta}$. Dann ist der Jackknife-Schätzer vom Bias definiert als

$$\widehat{\text{bias}}_{\text{jack}} = (n-1)(\hat{\theta}_{(\cdot)} - \hat{\theta})$$

mit

$$\hat{\theta}_{(\cdot)} = \sum_{i=1}^{n} \hat{\theta}_{(i)}/n.$$

Der Jackknife-Schätzer für den Standardfehler ist definiert als

$$\widehat{\text{se}}_{\text{jack}} = \left[\frac{n-1}{n} \sum_{i=1}^{n} (\hat{\theta}_{(\cdot)} - \hat{\theta})^2 \right]^{\frac{1}{2}}.$$

Der Jackknife wurde de-facto von dem Bootstrap abgelöst, den wir uns als nächstes anschauen werden.

3.4 Bootstrap

Efron (1979) hat den *Bootstrap* als Generalisierung des Jackknife-Prinzipes eingeführt. Bootstrap bedeutet Stiefelriemen, und ist eine amerikanische Redewendung, analog zu dem Baron Münchhausen der „sich selbst am Schopf aus dem Sumpf zieht." Die Methode beruht auf wiederholtem Ziehen mit Zurücklegen (*Resampling*) aus den beobachteten Daten mit dem Ziel, wie beim Jackknife, den Bias, die Varianz oder die Verteilung einer Statistik zu schätzen, die analytisch nicht bestimmbar ist. Folgende Darstellung des Boostraps stammt ebenfalls aus Efron und R. J. Tibshirani (1994).

Sei $\boldsymbol{X} = (x_1, x_2, \ldots, x_n)$ eine Stichprobe und $\hat{\theta} = s(\boldsymbol{X})$ wieder die uns interessierende Statistik. Wir möchten nun beispielhaft den Standardfehler von $\hat{\theta}$ mittels Bootstrap bestimmen. Eine *Bootstrap-Stichprobe* erhält man, indem man n mal mit Zurücklegen aus den beobachteten Daten zieht. Also z.B. mit $n = 7$ könnte eine Bootstrap-Stichprobe so aussehen:

$$\boldsymbol{X}^* = (x_5, x_7, x_5, x_4, x_7, x_3, x_1).$$

Algorithmus 2 : Bootstrap-Algorithmus zur Berechnung des Standardfehlers (Efron und R. J. Tibshirani 1994)

for b *in* $1 : B$ **do**

1. Ziehe n mal mit Zurücklegen aus $\boldsymbol{X}$ um eine Bootstrap-Stichprobe $\boldsymbol{X}^{*\boldsymbol{b}}$ zu erhalten
2. Berechne die Statistik mit der Bootstrap Stichprobe

$$\hat{\theta}^*(b) = s(\boldsymbol{X}^{*\boldsymbol{b}})$$

end

Schätze den Standardfehler $\text{se}_F(\hat{\theta})$ mit der Standardabweichung der B Bootstrap-Stichproben

$$\widehat{\text{se}}_B = \left\{ \sum_{b=1}^{B} [\hat{\theta}^*(b) - \hat{\theta}^*(\cdot)]^2/(B-1) \right\}^{1/2},$$

mit

$$\hat{\theta}^*(\cdot) = \sum_{b=1}^{B} \hat{\theta}^*(b)/B$$

Algorithmus 2 stellt schematisch die Berechnung des Standardfehlers von $\hat{\theta}$, $\widehat{\text{se}}_B$ mit B Bootstrap-Stichproben dar.

Wie haben Resampling-Techniken wie der Bootstrap das Paradigma der Statistik verändert? Efron (2000):

> *From a pre-World War II standpoint, our current computational abilities are effectively infinite, at least in terms of answering many common questions that arise in statistical practice. And no, this has not spelled the end of statistical theory—though it certainly has changed (for the better, in my opionion) what constitutes a good question and a good answer.*

4 Statistische Software: Das R Projekt

4.1 Entstehungs- und Erfolgsgeschichte

> *[...] we are going to reach a position we should have reached long ago. We are going, if I have to build it myself, to have a programming system—a "language" if you like—with all that that implies, suited to the needs of data analysis. This will be planned to handle numbers in organized patterns of very different shapes, to apply a wide variety of data-analytical operations to make new patterns from old, to carry out the oddest sequences of apparentlyunrelated operations, to provide a wide variety of outputs, to automatically store all time-expensive intermediate results on "disk" until the user decides whether or not he will want to do something else with them, and to do all this, and much more, easily.*
>
> – Tukey (1962, S. 28)

R (R Core Team 2012) – die Standard-Programmiersprache in der akademischen Welt für Statistik – wurde ursprünglich in den 1990ern von Ross Ihaka und Robert Gentleman an der Universität von Auckland geschrieben. Der Vorläufer von R war S, eine *interaktive* Programmiersprache zur Datenanalyse die in den 1970ern von John Chambers in den Bell Laboratories entwickelt wurde. Mit interaktiven Programmiersprachen kann der Nutzer direkt interagieren, ohne erst ein Programm kompilieren zu müssen. Dies geht zu Lasten der Geschwindigkeit, ermöglicht aber dem Nutzer z.B. einen Befehl wie `plot(x, y)` in eine Konsole einzugeben, um einen Plot zu erzeugen. Bis dahin musste man sehr schwierige Programmiersprachen wie FORTRAN lernen, um Datenanalyse zu betreiben oder neue Algorithmen zu implementieren. Alleine das Einlesen von Datensätzen erforderte eine Menge Code. S und später R, haben die Entwicklung und Nutzung statistischer Software entscheidend vereinfacht. Mittlerweile wird R als Open-Source-Projekt vom „R Core Team" weiterentwickelt. Die Geschichte von R kann man wie folgt zusammenfassen (Quelle: Hyndman 2012):

- 1976: S wurde in den Bell Laboratories entwickelt
- 1980: S wurde das erste Mal außerhalb von Bell verwendet
- 1988: S-PLUS
- 1997: CRAN mit 12 Paketen gestartet
- 2000: R 1.0.0 erschienen

Ein wesentlicher Grund für den Erfolg von R ist CRAN, ein *Paketsystem* auf welchem *Pakete* frei zu Verfügung gestellt werden. Jeder kann auf CRAN Pakete hochladen – solange sie dem Qualitätsstandard genügen. Pakete sind Erweiterungen der Basisfunktionalität einer Software und beinhalten meistens Funktionen für ein spezifisches Verfahren. Es ist mittlerweile üblich, dass eine Referenzimplementierung eines neuen statistischen Verfahrens auf CRAN zu Verfügung gestellt wird. Auf diese Weise kann jeder ein neues Verfahren in der Praxis testen und sich den Quellcode anschauen. So verbreiten sich neue Verfahren schneller und es entsteht ein positiver Rückkopplungseffekt, da Interessierte durch die Referenzimplementierung das neue Verfahren besser und schneller verstehen können. Dies hat oftmals Verbesserungen bereits bestehender Verfahren zur Folge oder bietet Inspiration für neue Methoden. CRAN ermöglicht auch Praktikern neueste statistische Methoden zu verwenden, ohne den oftmals jahrelangen Weg über kommerzielle Anbieter gehen zu müssen. Durch den offenen Quellcode ist es viel leichter herauszufinden, wie eine Methode implementiert wurde – und wenn gewünscht – sie an die eigenen Bedürfnisse anzupassen.

4.2 Wie hat R die Statistik verändert?

Interaktive Programmiersprachen wie R fördern die explorative Analyse von Daten wie sie Tukey (1970) in seiner Pionierarbeit „Exploratory Data Analysis" vorschlug. Tukey (1970) definiert die explorative Analyse von Daten als *„[…] detective work*

– numerical detective work – or counting detective work – or graphical detective work.“ Es ist in R sehr leicht mit Daten zu spielen, sie kennen zu lernen und sie zu visualisieren. Mead und Stern (1973) schreiben: „*Some of the things we do have been based on convenience more than on anything else, and what is convenient when you have a computer is not what was convenient before you had it. For example, it used to be very unpopular, as has been mentioned, to draw diagrams.*“ Mit dem R Paket `ggplot2` (Wickham 2009) lässt sich z.B. mit nur 6 Zeilen Code eine Publikationsfähige Grafik erstellen (Beispiel aus Chang (2012)):

```
library(ggplot2)
library(gcookbook)
sps <- ggplot(heightweight, aes(x=ageYear, y=heightIn, colour=sex)) +
  geom_point() +
  scale_colour_brewer(palette="Set1")
sps + geom_smooth()
```

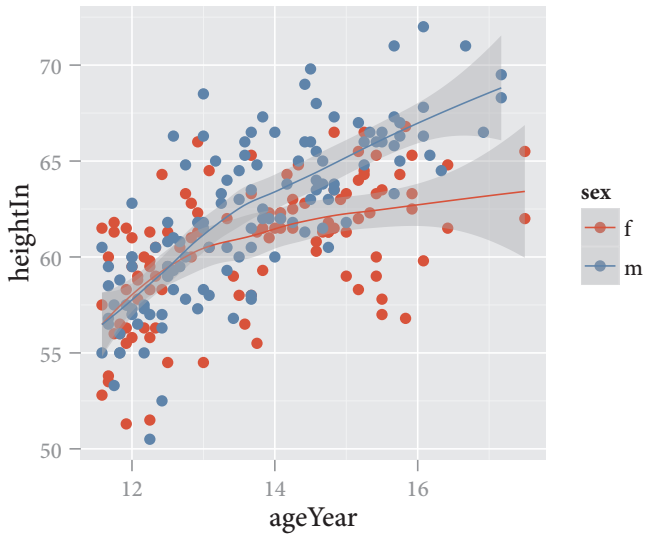

Abbildung 3: Beispiel `ggplot2`

Tukey (1965) sagt über Computergrafiken:

> *In exploration they are going to be the data analyst's greatest single resource.*

Die Leichtigkeit mit der man Ideen in interaktiven Programmiersprachen wie R umsetzen kann, hat dazu geführt, dass in der Praxis viel mehr Modelle und Visualisierung ausprobiert werden. Ripley (2005) erwähnt allerdings ermahnend, man sollte sich nicht dazu verleiten lassen, Denken mit Ausprobieren zu ersetzen.

Neben der explorativen Analyse bietet R mit seinem Paketsystem, nicht nur die Möglichkeit eine Vielzahl an Verfahren zu nutzen, sondern durch seine open-source Lizenz diese auch für seinen eigenen Bedarf zu modifizieren. Ich habe neulich erst selbst diese Möglichkeit wertschätzen gelernt. Im Rahmen meines Consultingprojektes entschied ich mich, ein neues Bayessches Verfahren zu verwenden, zu dem es noch keine Software gab. Hätte ich dieses von Grund auf selbst implementieren müssen, hätte es jeglichen zeitlichen Rahmen und Aufwand gesprengt und ich hätte eine andere, weniger passende Methode verwenden müssen. Ich konnte aber ein bereits bestehendes R-Paket so modifizieren, dass ich mit moderatem Zeitaufwand diese neue und vielversprechende Methode verwenden konnte. Diese Freiheit, Software zu modifizieren, hat vielleicht nicht das theoretische Paradigma der Statistik verschoben, aber das Praktische ungemein. Wenige Statistiker konnten in den 60ern realistischerweise in FORTRAN neue Verfahren implementieren ohne erheblichen Lern- und Zeitaufwand (normalerweise wurde dies Informatikern überlassen). Mit geschlossener, kommerzieller Software ist man darauf angewiesen, dass der Hersteller das Verfahren so implementiert (wenn überhaupt), wie man es nutzen möchte. Man war in der Praxis in den Methoden gefangen, welche als Software zur Verfügung standen. R hat uns davon befreit. Wie Ripley es formulierte:

> *It is statistical software that has revolutionized the way we approach data analysis, replacing the calculators—mechanical, electrical, electronic—used by earlier generations.*
>
> – Ripley (2005)

5 Machine Learning

5.1 Die zwei Kulturen

> *There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown.*
>
> – Breiman (2001)

Machine Learning ist eine junge Disziplin der Datenanalyse, die vollständig auf computerintensiven algorithmischen Methoden basiert. Sie ist sozusagen aus der Computationalen Wende entstanden. Statistik und Machine Learning haben das gleiche Ziel – von Daten zu lernen – unterscheiden sich aber in ihrer Herangehensweise, ihren Intentionen, ihrer Kultur, ihren Konventionen und ihrer Geschichte (Wasserman 2013). Die folgenden Gedanken und Diagramme zu den zwei unterschiedlichen Herangehensweisen an die Analyse von Daten stammen von Breiman (2001). Man stelle sich vor, die Daten werden von einer „black box“ generiert. Der Kovariablenvektor x geht als Input hinein und heraus kommt der Responsevektor y. Innerhalb dieser „black box“ verbindet die Natur die Kovariablen mit der Response. Die Datenanalyse hat zwei Ziele:

Vorhersage Zukünftige Werte der Response vorhersagen.

Information Informationen über die Natur der Verbindung zwischen Response und Kovariablen aus den Daten gewinnen.

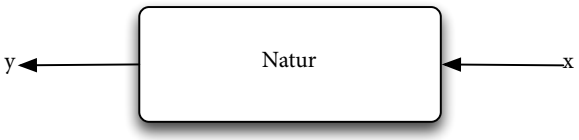

Abbildung 4: Die Natur verbindet die Kovariablen mit der Response

„The Data Modeling Culture“

Die Statistik nimmt an, dass sich ein stochastisches Modell innerhalb der „black box“ befindet. Ein gewöhnliches „data model“ nimmt an, dass der Response aus unabhängigen Ziehungen aus einem Modell

$$\text{Response Variable} = f(\text{Kovariablen, zufällige Fehler, Parameter})$$

besteht. Die Parameter werden aus den Daten geschätzt. Mit dem Modell werden Informationen über die Verbindung zwischen Response und Kovariablen gewonnen und Vorhersagen generiert. Die „black box“ wird z.B. wie in Abbildung 5 ausgefüllt. Die Informationen werden über Hypothesentests auf die

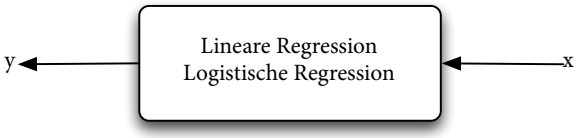

Abbildung 5: Beziehung zwischen Response und Kovariablen in der „data modeling“ Kultur

Signifikanz der Kovariablen gewonnen. Die Daten müssen aber oft strikte Annahmen bezüglich des datengenerierenden Prozesses erfüllen, damit die Hypothesentests gültig sind. Inferenz wird daher über das Modell und nicht die „wahre“ Natur betrieben, was zu falschen Schlussfolgerungen führen kann. Auch ist die funktionale Form von f durch die Modellannahmen eingeschränkt.

„The Algorithmic Modeling Culture“

In der „algorithmic modeling„ Kultur dagegen wird angenommen, dass das Innenleben der „black box“ komplex und unbekannt ist. Es gilt eine Funktion $f(x)$ zu finden, ein Algorithmus, der als Input die Kovariablen x nimmt und Vorhersagen für den Response y generiert. Ihre „black box“ sieht wie in Abbildung 6 aus. Beim Machine Learning wird der Schwerpunkt auf die Vorhersage und nicht auf Informationsgewinnung gelegt. Es werden keine Hypothesentests bezüglich der Kovariablen verwendet, sondern Modelle werden alleine anhand der *out-of-sample* Vorhersagegüte mithilfe von Resampling verglichen. Diese informelle Herangehensweise – ohne jegliche Annahmen über die Daten zu treffen – ermöglicht es, beliebige funktionale Zusammenhänge sehr flexibel zu schätzen.

Breiman (2001) erwähnt eine wichtige Tatsache nicht. Arbeitet man mit einem „algorithmic model“, geht dabei die Mög-

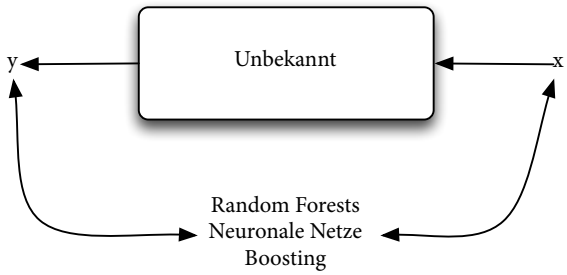

Abbildung 6: Herangehensweise der „algorithmic modeling“ Kultur

lichkeit verloren, formale Hypothesentests über die Signifikanz der Zusammenhänge durchzuführen. In der Medizin und den Naturwissenschaften sind die Informationen über Zusammenhänge oft wichtiger als Vorhersagen der Response. Cox (Breiman 2001, S. 216) sagt in diesem Zusammenhang: „*Professor Breiman takes data as his starting point. I would prefer to start with an issue, a question or a scientific hypothesis.*“ Auch Efron kritisiert den Mangel an Inferenz von dem „black box“ Ansatz (Breiman 2001, S. 219): „*The whole point of science is to open up black boxes, understand their insides, and build better boxes […].*“ Wenn es ausschließlich um die Vorhersage und nicht um Informationsgewinn geht, haben die flexiblen Ansätze des Machine Learning mittlerweile einen Vorsprung. Mit computerintensiven Machine Learning Modellen wie *deep neural networks* lassen sich beliebige Funktionen approximieren.

Für die meisten Statistiker bestehen Daten aus Zahlen. Neuartige Datensätze wie Webseiten, Netzewerke, Sprache, Bilder und Videos hat die Statistik bislang noch so gut wie gar nicht erfasst. Diese Datensätze bilden einen immer größeren Anteil der gesamten Daten auf der Welt und werden komplett dem Machine Learning überlassen, obwohl die Statistik einen wertvollen Beitrag liefern kann. Wasserman (2013, S. 10) dazu: „*This comes back to education. If our students can't analyze giant datasets like millions of twitter feeds or millions of web pages then other people will analyze those data. We will end up with a small cut of the pie.*“ Es wird interessant zu beobachten, inwieweit diese beiden Disziplinen der Datenanalyse konvergieren oder divergieren werden. Eine gewisse Konvergenz der Paradigmen ist bereits zu erkennen, z.B. in J. Friedman, Hastie und R. Tibshirani (2000). Auch in den zwei wichtigsten Machine Learning Journals (JMLR und NIPS) finden sich immer mehr statistische Themen. Andererseits etablieren sich momentan Masterstudiengänge in Machine Learning und Data Science, getrieben durch die Nachfrage aus der Industrie.

Machine Learning hat ein neues, eigenständiges Paradigma zur Datenanalyse geschaffen, basierend auf den neuen Möglichkeiten der Computationalen Wende. Ich teile nicht die Ansicht von Breiman (2001) und Wasserman (2013), dass Machine Learning dabei ist, die Statistik obsolet zu machen. Die Wahl zwischen einem „algorithmic model“ und einem „data model“ bestimmt viel mehr das Ziel der Analyse. Geht es um Inferenz – das Gewinnen von Informationen über die Natur aus den Daten und das Verstehen von dem zu Grunde liegenden Prozess – bietet Machine Learning (noch) keine Alternative zu der Statis-

tik. Geht es um die reine Prognosegüte unter realistischem Umständen, dem Testfehler, haben die flexiblen Modelle des Machine Learnings einen Vorteil. Beide Disziplinen beeinflussen und befruchten sich bereits, könnten aber noch mehr voneinander profitieren.

5.2 Ensembles und das „ISLE Framework“: Bagging & Boosting

Importance Sampling Learning Ensembles (ISLE) (J. H. Friedman 2003) – eine Klasse von Machine Learning Algorithmen wozu *Bagging* (Breiman 1996) und Boosting (Bühlmann und Hothorn 2007) gehören – haben einen besonders großen Einfluss auf die Statistik ausgeübt, weshalb ich diese Verfahren hier kurz anreißen möchte. Man betrachte nur einige der Artikel, die in Statistik Journals zu diesen Themen erschienen sind:

- J. Friedman, T. Hastie und R. Tibshirani (2000). „Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors)“. In: *The Annals of Statistics* 28.2, S. 337–407
- P. Bühlmann und B. Yu (2003). „Boosting With the L 2 Loss“. In: *Journal of the American Statistical Association* 98.462, S. 324–339
- P. Bühlmann und T. Hothorn (Nov. 2007). „Boosting Algorithms: Regularization, Prediction and Model Fitting“. In: *Statistical Science* 22.4, S. 477–505
- P. Bühlmann und B. Yu (2002). „Analyzing bagging“. In: *The Annals of Statistics* 30.4, S. 927–961

Das ISLE Framework ermöglicht es, auf Resampling basierende Machine Learning Algorithmen (wie z.B. L_2Boosting oder Random Forests) als Spezialfälle eines einzigen Algorithmus darzustellen (Giovanni Seni 2010).

Boosting

Schapire und Freund – die Erfinder des ersten Boosting Algorithmus *AdaBoost* – beschreiben Ihren Ansatz wie folgt (Schapire und Freund 2012): „*[…] boosting, an approach to machine learning based on the idea of creating a highly accurate prediction rule by combining many relatively weak and inaccurate rules.*“ Dieses Prinzip lässt sich mithilfe einer Analogie zu der bekannten Fernsehsendung „Wer Wird Millionär“ erläutern. Dort gibt es einen Publikumsjoker, den ein Kandidat ziehen kann, wenn er eine Antwort nicht weiß. Jedes Publikumsmitglied stimmt daraufhin ab, welche Antwort er für richtig hält. Dem Kandidaten wird dann das Ergebnis der Abstimmung gezeigt. Die Mehrheit hat fast immer recht – selbst bei äußerst schweren Fragen. Dabei sitzen im Publikum auch keine klügeren Menschen als in der Allgemeinheit zu finden sind. Wenn man mit dem Joker nur per Zufallswahl eine einzige Person aus dem Publikum befragen könnte, wäre dieser Joker meistens wertlos. Wenn aber alle Stimmen aggregiert werden, stimmt die Antwort fast immer. Dieses Prinzip, sogenannte „schwache Entscheidungsregeln“ zu kombinieren, um eine bessere zu erhalten, versucht Boosting zu nutzen, um bessere Modelle zu konstruieren. Diese Entscheidungsregeln, z.B.

- Lineare komponentenweise KQ-Schätzer;
- B-Splines; oder
- Bäume,

werden zu hunderten oder gar tausenden zu einem *Ensemble* (das Publikum) kombiniert. Um eine Vorhersage für den Response zu generieren, findet im Fall kategorialer Daten eine Mehrheitswahl aller Entscheidungsregeln statt und im Fall stetiger Daten wird der Mittelwert aller Vorhersagen genommen (Publikum stimmt ab). Der Boosting Algorithmus lernt etwas über die Daten durch wiederholtes Anwenden der Entscheidungsregeln auf die Daten. Wendet man aber die Entscheidungsregeln immer wieder auf den selben Response an, wird nichts Interessantes passieren. Der Boosting Algorithmus muss den Response so manipulieren, dass er etwas Neues in jeder Iteration über die Daten lernen kann. Dies geschieht dadurch, dass die Residuen der vorherigen Iteration zur neuen Response gemacht werden. So bekommen schwer vorherzusagende Beobachtungen ein höheres Gewicht. Normalerweise wird dabei ei-

Algorithmus 3 : L_2Boosting (Bühlmann und Yu 2003)

Initialisierung: $\widehat{F}_0 \equiv 0$
for m *in* $1 : m_{stop}$ **do**

1. Residuen $U_i = Y_i - \widehat{F}_{m-1}(X_i)$, $i = 1, \dots, n$ berechnen
2. Base-Learner auf aktuelle Residuen fitten $\widehat{f}_m(\cdot)$
3. $\widehat{F}_m(\cdot) = \widehat{F}_{m-1}(\cdot) + \nu \widehat{f}_m(\cdot)$ aktualisieren

end

ne feste Schrittlänge ν gewählt (z.B. $\nu = 0.1$ oder $\nu = 0.01$) und über die „Stop-Iteration“ m_{stop} optimiert. Die Attraktivität von Boosting besteht in der gleichzeitigen Regularisierung (durch $\nu < 1$) und Variablenselektion (durch m_{stop}).

Bagging

Bagging steht für „**b**ootstrap **agg**regat**ing**“ und wurde von Breiman (1996) eingeführt um die Varianz eines Modells zu reduzieren. Die Idee dahinter ist, dass man wie bei Boosting schwache Entscheidungsregeln zu einem Ensemble kombiniert. Diesmal werden aber nicht die Residuen der vorherigen Iteration, wie bei Boosting, herangezogen, um den Datensatz zu perturbieren, sondern es wird in jeder Iteration eine Bootstrap Stichprobe des Originaldatensatzes gezogen und darauf die Entscheidungsregel angewandt. Alle so konstruierten Entscheidungsregeln werden wie bei Boosting zu einem Ensemble kombiniert. Es wird über die Anzahl der Bootstrap-Stichproben optimiert. Bagging funktioniert besonders gut für Schätzmethoden mit hoher Varianz (kleine Änderungen in den Daten bewirken große Änderungen in der Schätzung), wie z.B. Klassifikations- oder Regressionsbäume, da durch das Perturbieren der Daten die Varianz reduziert wird. Stabile Schätzme-

thoden können aber durch Bagging verschlechtert werden, da auch der Bias durch die Perturbation erhöht wird.

Für B Bootstrap-Stichproben sieht der Bagging-Algorithmus wie folgt aus:

Algorithmus 4 : Bagging (Breiman 1996)

for i *in* $1 : B$ **do**

1. Ziehe ein Bootstrap-Sample aus den Daten
2. Berechne den Boostrap-Schätzer der Entscheidungsregel nach dem plug-in Prinzip

end

Aggregiere alle B Bootstrap-Schätzer zu einem Ensemble

6 Die neue Computationale Wende: Big Data

Bahnt sich eine „neue" Computationale Wende an? Die Größe der Datensätze wächst momentan schneller als das Moore'sche Gesetz (Gantz und Reinsel 2011). Die Entschlüsselung des menschlichen Genoms, riesige Teilchenbeschleuniger wie der LHC in CERN, das Internet und andere Quellen erzeugen mittlerweile Terra- und Petabyte große Datensätze. Die meiste statistische Software und statistischen Algorithmen können bereits Gigabyte große Datensätze nicht mehr in realistischer Zeit bearbeiten. Sollte die Statistik diese Datensätze ignorieren oder sich nur auf kleine Subsamples beschränken?

Der Flaschenhals liegt in der iterativen Natur der meisten statistischen Verfahren, so dass diese meistens nur auf einem Prozessorkern bearbeitet werden können. Eine neue Herausforderung für die Statistik wird es sein, neue *parallelisierbare* Verfahren zu entwickeln, oder alte so anzupassen, dass sie parallelisierbar werden und somit diese riesigen Datensätze auswerten können. Google hat z.b. 2012 ein experimentelles Neuronales Netz auf 1.000 Computern und 16.000 Prozessorkernen eingerichtet. Dieser Bedarf an statistischen Verfahren, die auf einer großen Menge an Computern parallel berechnet werden können, wird in den kommenden Jahren das Paradigma der Statistik wieder ein bisschen verschieben. Allerdings ermahnt uns Ripley (2005):

> *Just because we can now apply simple methods to large datasets should not of itself encourage doing so.*

7 Fazit

> *One thing seems certain: any statistician who seeks to influence the development of our subject in the next 150 years must become involved with computers.*
>
> – Nelder (1984)

War nun die Computationale Wende ein Sturm im Wasserglas? Ich denke nicht – die Computationale Wende hat die Entwicklung der statistischen Theorie fundamental beeinflusst und tut dies immer noch. Statistik zielt letztendlich immer darauf, Datenanalyse zur betreiben und ist damit von den ihr zur Verfügung stehenden Werkzeugen abhängig. Computer sind seit der zweiten Hälfte des 20. Jahrhunderts dieses Werkzeug.

Der Fortschritt in der Bayesschen Statistik in den letzten zwei Dekaden, wurde durch den Fortschritt in den computationalen Resourcen möglich gemacht. Ohne die computerintensiven MCMC Methoden, wäre heute die Bayessche Statistik nicht da, wo sie ist und damit auch das gesamte Paradigma der Statistik. Simulation und Resampling haben analytisch nicht zu lösende Probleme greifbar gemacht und die Fragestellungen in der Statistik verändert.

Statistische Software, insbesondere R, hat dazu beigetragen, dass Tukey´s explorative Analyse Routine ist und wir ohne große Mühe Visualisierungen eines Datensatzes erstellen können. Vor der Computationalen Wende war es unpraktikabel mehrere Modelle zu schätzen – heute ist dies in Minuten möglich. Open-source Software wie R hat uns die Freiheit gegeben, herauszufinden, wie ein Verfahren implementiert wurde und es an unsere Bedürfnisse anzupassen.

Die Computationale Wende hat eine eigene Disziplin zur Datenanalyse hervorgebracht: das Machine Learning. Diese ist mittlerweile eine erwachsene Disziplin, die wertvolle Beiträge zur statistischen Theorie beisteuert und damit das statistische Paradigma beeinflusst. Ich glaube nicht, dass die Veränderungen in der Statistik bereits abgeschlossen sind. Eine immer mehr vernetzte Welt und die Wissenschaft erzeugen zunehemend größere Datensätze, die schneller als die verfügbare Rechenkapazität eines einzelnen Computers wachsen. Es wird nötig sein, Verfahren zu entwickeln, die diese Datensätze auswerten können. Die Statistik wird vielleicht immer mehr mit der Informatik zusammenwachsen, was man schon etwas an der aufkommenden „data science" Bewegung erkennen kann.

Ein kurioser Umstand, der mir bei den Nachforschungen für diese Arbeit aufgefallen ist, beschreibt Efron (2000):

> *There is some sort of law working here whereby statistical methodology always expands to strain the current limits of computation.*

Literatur

Andrews, D. F. u. a. (Feb. 1972). *Robust Estimates of Location: Survey and Advances*. Princeton: Princeton University Press.

Andrieu, C., A. Doucet und C. P. Robert (2004). „Computational advances for and from Bayesian analysis“. In: *Statistical Science* 19.1, S. 118–127.

Becker, R. A. (1994). „A Brief History of S“. In: *AT&T Bell Laboratories*.

Box, G. E. P. und N. R. Draper (Jan. 1987). *Empirical model-building and response surfaces*. John Wiley & Sons Inc.

Breiman, L. (1996). „Bagging predictors“. In: *Machine learning* 24.2, S. 123–140.

— (2001). „Statistical modeling: The two cultures (with comments and a rejoinder by the author)“. In: *Statistical Science* 16.3, S. 199–231.

Bühlmann, P. und B. Yu (2002). „Analyzing bagging“. In: *The Annals of Statistics* 30.4, S. 927–961.

— (2003). „Boosting With the L 2 Loss“. In: *Journal of the American Statistical Association* 98.462, S. 324–339.

Bühlmann, P. und T. Hothorn (Nov. 2007). „Boosting Algorithms: Regularization, Prediction and Model Fitting“. In: *Statistical Science* 22.4, S. 477–505.

Casella, G. und E. I. George (1992). „Explaining the Gibbs sampler“. In: *American Statistician* 46.3, S. 167–174.

Chang, W. (Dez. 2012). *R Graphics Cookbook*. O'Reilly Media.

Efron, B. (1979). „Bootstrap methods: another look at the jackknife“. In: *The Annals of Statistics* 7.1, S. 1–26.

— (2000). „The bootstrap and modern statistics“. In: *Journal of the American Statistical Association* 95.452, S. 1293–1296.

Efron, B. und R. J. Tibshirani (Mai 1994). *An Introduction to the Bootstrap (Chapman & Hall/CRC Monographs on Statistics & Applied Probability)*. 1. Aufl. Chapman und Hall/CRC.

Friedman, J. H. (2003). „Importance sampled learning ensembles“. In: *Journal of Machine Learning …*.

Friedman, J., T. Hastie und R. Tibshirani (2000). „Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors)“. In: *The Annals of Statistics* 28.2, S. 337–407.

Gantz, J. und D. Reinsel (2011). *Extracting Value from Chaos*. URL: http://germany.emc.com/collateral/analyst-reports/idc-extracting-value-from-chaos-ar.pdf (besucht am 24. 01. 2013).

Gelfand, A. E. und A. F. M. Smith (1990). „Sampling-based approaches to calculating marginal densities“. In: *Journal of the American Statistical Association* 85.410, S. 398–409.

Geman, S. und D. Geman (1984). „Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images“. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 6, S. 721–741.

Gentle, J. E., Y. Mori und W. K. Härdle (2004). *Handbook of computational statistics*. Springer Berlin etc.

George, E. I. und R. E. McCulloch (1993). „Variable selection via Gibbs sampling“. In: *Journal of the American Statistical Association* 88.423, S. 881–889.

Giovanni Seni, J. E. (Mai 2010). „Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions (Synthesis Lectures on Data Mining and Knowledge Discovery)“. In: S. 1–127.

Hastings, W. K. (1970). „Monte Carlo sampling methods using Markov chains and their applications“. In: *Biometrika* 57.1, S. 97–109.

Hemmerle, W. (1967). *Statistical computations on a digital computer*. Blaisdell book in computer science. Blaisdell Pub. Co.

Hyndman, R. J. (Nov. 2012). „simpleR“. In: *Melbourne R Users' Group*. Melbourne, S. 1–81.

Kuhn, T. S. (1996). *The structure of scientific revolutions*. Bd. 2. University of Chicago press.

Madigan, D. und A. E. Raftery (1994). „Model selection and accounting for model uncertainty in graphical models using Occam's window“. In: *Journal of the American Statistical Association* 89.428, S. 1535–1546.

Mead, R. und R. D. Stern (1973). „The Use of a Computer in the Teaching of Statistics“. In: *Journal of the Royal Statistical Society. Series A (General)*, S. 191–225.

Metropolis, N. (1987a). „The beginning of the Monte Carlo method“. In: *Los Alamos Science* 15.584, S. 125–130.

Metropolis, N. (1987b). *The Los Alamos experience, 1943-1954*. Techn. Ber. Los Alamos National Lab., NM (USA).

Miller Jr, R. G. (1964). „A trustworthy jackknife“. In: *The Annals of Mathematical Statistics*, S. 1594–1605.

Müller, G. (2011). *Computerintensive Methoden Einheit 4: Zufallszahlen*. URL: http://www.statistik.lmu.de/institut/ag/leisch/teaching/cim1112/folien/cim4-4.pdf (besucht am 22. 01. 2013).

Nelder, J. A. (1984). „Present Position and Potential Developments: Some Personal Views: Statistical Computing“. In: *Journal of the Royal Statistical Society. Series A (General)*, S. 151–160.

Neumann, J. von (1951). „Various techniques used in connection with random digits“. In: *NBS Applied Math Series* 12.36-38, S. 1.

R Core Team (2012). *R: A Language and Environment for Statistical Computing*. ISBN 3-900051-07-0. R Foundation for Statistical Computing. Vienna, Austria. URL: http://www.R-project.org/.

Ripley, B. D. (Feb. 2005). „How computing has changed statistics“. In: *Celebrating Statistics: Papers in Honour of Sir David Cox on His 80th Birthday*. Hrsg. von A. C. Davison, Y. Dodge und N. Wermuth. Oxford University Press, S. 197–211.

Robert, C. und G. Casella (Nov. 2010). *Monte Carlo Statistical Methods*. Springer.

— (Feb. 2011). „A Short History of Markov Chain Monte Carlo: Subjective Recollections from Incomplete Data“. In: *Statistical Science* 26.1, S. 102–115.

Schapire, R. E. und Y. Freund (Mai 2012). *Boosting*. Foundations and Algorithms. MIT Press.

Stigler, S. M. (Nov. 2010). „The Changing History of Robustness“. In: *The American Statistician* 64.4, S. 277–281.

Stone, M. (1974). „Cross-validatory choice and assessment of statistical predictions“. In: *Journal of the Royal Statistical Society. Series B (Methodological)*, S. 111–147.
Teichroew, D. (1965). „A history of distribution sampling prior to the era of the computer and its relevance to simulation“. In: *Journal of the American Statistical Association* 60.309, S. 27–49.
Tukey, J. W. (1962). „The future of data analysis“. In: *The Annals of Mathematical Statistics* 33.1, S. 1–67.
— (1965). „The technical tools of statistics“. In: *American Statistician* 19.2, S. 23–28.
— (1970). „Exploratory Data Analysis (limited preliminary edition)“. In: *Ann Arbor*.
Wasserman, L. (Feb. 2013). „Rise of the Machines“. In: *Noch nicht erschienen*, S. 1–12.
Watnik, M. (Jan. 2011). „Early Computational Statistics“. In: *Journal of Computational and Graphical Statistics* 20.4, S. 811–817.
Wickham, H. (Aug. 2009). *ggplot2: Elegant Graphics for Data Analysis (Use R!)* 2nd Printing. Springer.