Ludwig-Maximilians-Universität München
Institut für Statistik

Zur Variablenselektion bei fehlerbehafteten Daten: korrigierte Score-Funktionen in Kombination mit dem LASSO-Verfahren

Bachelorarbeit

Abgabetermin: 24. August 2018

Name: Quan Nguyen
Betreuer: Prof. Dr. Thomas Augustin

Abstract

Um die Beta-Koeffizienten in einem Generalisierten Linearen Modell (GLM) zu ermitteln wird üblicherweise die Nullstelle der Score-Funktion ermittelt. Sind die unabhängigen Variablen allerdings von Messfehlern betroffen, sind die Schätzungen für die Beta-Koeffizienten oftmals verzerrt. Eine Möglichkeit das auszugleichen sind korrigierte Score-Funktionen, deren bedingter Erwartungswert gegeben $\boldsymbol{Y}$ und $\boldsymbol{Z}$ der wahren Score-Funktion entspricht (Nakamura; 1990). Eine weitere Herausforderung bei der Modellschätzung ist die Variablenselektion. Ein bekanntes Verfahren zur Variablenselektion ist das sogenannte LASSO (Tibshirani; 1996). Im Rahmen dieser Arbeit wurde untersucht inwiefern sich die beiden Verfahren kombinieren lassen. Es stellte sich heraus, dass sich einige geläufige Verfahren zur Lösung des LASSO Problems in Zusammenhang mit den korrigierten Score-Funktionen nicht anwenden lassen. Ein von (Fan und Li; 2001) vorgeschlagener Algorithmus mit dem SCAD Strafterm, der auf die korrigierte Log-Likelihood-Funktion anwendbar ist, erwies sich als kompatibel und es wurden die Ergebnisse für die sogenannte naive und korrigierte Log-Likelihood-Funktion gegenübergestellt. In dieser Arbeit wurde speziell die Poisson-Verteilung betrachtet. Die Ergebnisse zeigen, dass die naive Log-Likelihood-Funktion mit dem SCAD Algorithmus keine brauchbaren Ergebnisse liefert unter Messfehlern. Der SCAD Algorithmus mit der korrigierten Log-Likelihood-Funktion liefert für verschiedene Messfehler geeignete Schätzungen und selektiert eine vernünftige Anzahl an richtigen Nullen, wodurch sich das kombinierte Verfahren bei Messfehlern anbietet.

Keywords : korrigierte Score, Messfehler, Variablenselektion, GLM, LASSO, konvexe Optimierung, SCAD

Inhaltsverzeichnis

1. Einleitung

Messfehler in Kovariablen haben drei Auswirkungen:

1. Sie verursachen Verzerrung bei der Parameterschätzung in statistischen Modellen.
2. Sie führen zu einem Verlust an Power bei der Erkennung von Zusammenhängen zwischen Variablen.
3. Sie verdecken die Eigenschaften der Daten, was die graphische Modellanalyse schwierig macht.

Die ersten beiden werden „Doppelschlag“ (Double Whammy) des Messfehlers genannt. Alle drei nennt man einen „Dreifachschlag“ (Triple Whammy). (Carroll et al.; 2006, S. 1) In dieser Arbeit wird vor allem die erste Eigenschaft, die Verzerrung von Parameterschätzungen, betrachtet.

In einem Generalisierten Linearen Modell (GLM) werden die Parameter üblicherweise über eine Maximum-Likelihood Schätzung ermittelt (Fahrmeir et al.; 2009, S. 223). Dabei wird das $\boldsymbol{\beta}$ ermittelt, welches die Score-Funktion $\boldsymbol{s(\beta, Z, Y)}$ gleich 0 setzt. Dabei sind die $\boldsymbol{Z}$ die Menge der unabhängigen Variablen und die $\boldsymbol{Y}$ die Menge der Zielvariablen, deren Verteilung von den $\boldsymbol{Z}$ abhängt. In vielen Fällen ist es allerdings so, dass die $\boldsymbol{Z}$ nicht vorliegen, sondern nur die mit einem Messfehler gemessenen/beobachteten Werte $\boldsymbol{X}$ von $\boldsymbol{Z}$. Messfehler können z.B. entstehen bei der Messung der Proteinaufnahme. Da die wahre Menge nicht direkt messbar ist wird ein Biomarker verwendet. Dadurch wird der wahre Wert mit einer zusätzlichen Variabilität erfasst. (Carroll et al.; 2006, Kap. 1)

Wenn man jetzt die $\boldsymbol{Z}$ durch die $\boldsymbol{X}$ ersetzt, erhält man die „naive“ Score-Funktion. Das Problem hierbei ist, dass die Schätzungen der naiven Score-Funktion nicht unbedingt konsistent sind. D.h. es gilt nicht immer $E(\boldsymbol{s(\beta_t, Z, Y)}|\boldsymbol{Y}) = 0$ und damit hält die Bedingung auch für ein geschätztes $\boldsymbol{\beta}_x$ nicht. (Nakamura; 1990, Kap. 2)

Eine Möglichkeit dagegen vorzugehen und unverzerrte Schätzungen zu erhalten sind korrigierte Score-Funktionen (Nakamura; 1990). Die Idee dahinter ist die „naive“ Score-Funktion mit einer Korrektur zu versehen, so dass der bedingte Erwartungswert gegeben

$\boldsymbol{Z}$ und $\boldsymbol{Y}$ der wahren Score-Funktion entspricht. Diese wiederum ist um den wahren Parameterwert zentriert, woraus folgt, dass die korrigierte Score-Funktion auch eine unverzerrte Score-Funktion ist. (Nakamura; 1990, Kap. 1)

Eine weitere Problematik, die häufig bei der Modellschätzung auftritt, ist die Anzahl p der Kovariablen, die man gerne reduzieren möchte. Zwei übliche Möglichkeiten damit umzugehen sind die Teilmengenselektion und die Ridge Regression. Die Teilmengenselektion liefert interpretierbare Modelle, kann aber extrem variabel sein, weil es ein diskreter Prozess ist. Regressoren werden entweder im Model behalten oder entfernt. Kleine Veränderungen in den Daten können in sehr unterschiedlichen Modellen resultieren und dies kann die Vorhersagegenauigkeit reduzieren. (Tibshirani; 1996, Kap. 1) Eine Teilmenge der Größe 2 muss dabei nicht unbedingt die Variable beinhalten, die in der Teilmenge der Größe 1 enthalten wäre (Hastie et al.; 2009, Kap. 3.3.1). Ridge Regression ist ein stetiger Prozess, der Variablen schrumpft und daher stabiler ist. Der Nachteil allerdings ist, dass keine Variablen selektiert werden, wodurch die Interpretierbarkeit leidet (Tibshirani; 1996).

Das LASSO-Verfahren, kurz für „least absolute shrinkage and selection operator“, schrumpft Koeffizienten und setzt dabei manche auf 0, was einer Variablenselektion entspricht. Es versucht also die guten Eigenschaften von der Teilmengenselektion und der Ridge Regression zu kombinieren. Das LASSO-Verfahren hilft somit bei zwei Schwierigkeiten:

1. Vorhersagegenauigkeit: Durch das Schrumpfen und auf 0 setzen von manchen Koeffizienten, wird die Vorhersagegenauigkeit eventuell verbessert. Die Varianz wird reduziert und im Gegenzug dafür wird die Verzerrung ein wenig vergrößert.

2. Interpretierbarkeit: Wenn man eine große Anzahl von Prädiktoren hat, möchte man oftmals ein kleineres Modell, welches nur die stärksten Effekte enthält.

(Tibshirani; 1996, Kap.1)

In dieser Arbeit soll versucht werden das LASSO-Verfahren auf die korrigierten Score-Funktionen anzuwenden. Ziel ist es Variablenselektion mit den für Messfehler korrigierten Koeffizienten zu betreiben, die Schrumpfung ist ein zusätzlicher Effekt. Dabei soll die Poisson-Verteilung betrachtet werden. Im Laufe der Arbeit wurde festgestellt, dass das Elastische Netz eine verbesserte Erweiterung des LASSO darstellt. Allerdings soll der Fokus trotzdem auf dem LASSO liegen, da in vielen Fällen das LASSO auf das Elastische Netz erweitert werden kann (Zou und Hastie; 2005). Um das LASSO Problem für eine

Poisson-Verteilung mit Messfehlerkorrektur zu lösen, wurden diverse nichtlineare, konvexe Optimierungsverfahren in Erwägung gezogen. Schlussendlich wurde ein Algorithmus mit dem SCAD Strafterm verwendet, welcher Eigenschaften des LASSO mitbringt (Fan und Li; 2001).

Am Anfang von Kapitel 2 wird das GLM kurz erklärt. In Kapitel 2.1 wird die Methodik der korrigierten Score-Funktionen erläutert. Kapitel 2.2 gibt einen Überblick über das LASSO inklusive weiterer Regularisierungsverfahren. Kapitel 2.3 stellt einige konvexe Optimierungsverfahren vor und erklärt die jeweilige Problematik in Bezug auf die korrigierte Score-Funktion. In Kapitel 2.3.5 wird der von (Fan und Li; 2001) vorgeschlagene Algorithmus mit dem SCAD Strafterm erläutert. Kapitel 3 erklärt die Anwendung dieses Algorithmus auf die naive (gewöhnliche) und korrigierte Log-Likelihood der Poisson-Verteilung anhand einer Simulation in R (R Core Team; 2018). Die Ergebnisse werden für die Werte $\sigma = 0/0, 1/0, 2/0, 3$ und $n = 100/500/1000$ dargestellt, jeweils für die naive und korrigierte Log-Likelihood-Funktion und gegenübergestellt.

2. Methodik

Wir wollen also in einem ersten Schritt die Beta-Koeffizienten eines Models schätzen und dabei gleichzeitig Messfehler in den unabhängigen Variablen berücksichtigen. Im zweiten Schritt bzw. gleichzeitig sollen diese Koeffizienten dann geschrumpft und selektiert werden. Für den ersten Schritt sind die korrigierten Score-Funktionen von (Nakamura; 1990) die Grundlage und im zweiten Schritt das LASSO-Verfahren von (Tibshirani; 1996).

Generalisiertes Lineares Modell

Das Modell für das die Beta-Koeffizienten geschätzt werden sollen ist ein Generalisiertes Lineares Modell (GLM), welches eine Erweiterung des klassischen linearen Modells ist. Im klassischen linearen Modell verwendet man zur Schätzung üblicherweise die Kleinste-Quadrate-Methode (KQ-Schätzer) (Fahrmeir et al.; 2009, S. 90-91) und im GLM greift man auf die Maximum-Likelihood-Methode zurück . Zudem lässt ein GLM auch eine andere Verteilungsannahme als die Normalverteilung für die Zielvariable zu. In einem GLM geht man davon aus, dass die Zielvariable $\boldsymbol{Y}$ für jede Beobachtung eine Verteilung der Form

$$f_Y(y;\theta,\phi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\} \tag{2.1}$$

annimmt. Hierbei sind a(.), b(.) und c(.) spezifische Funktionen, d.h. sie variieren je nach Verteilung. $y\theta$ und $b(\theta)$ sind dabei Funktionen, die den natürlichen Parameter θ beinhalten. Erwartungswert und Varianz lassen sich bestimmen durch die erste und zweite Ableitung des $b(\theta)$ nach dem natürlichen Parameter θ mit

$$E(y) = \mu = b'(\theta), \qquad Var(y) = b''(\theta)a(\phi).$$

(McCullagh und Nelder; 1989, Kap. 2) Desweiteren gilt für den Erwartungswert μ_i und den linearen Prädiktor $\eta_i = \boldsymbol{x}_i.\boldsymbol{\beta} = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}$ die Beziehung

$$\mu_i = h(\boldsymbol{\eta}_i) = h(\boldsymbol{x}_i.\boldsymbol{\beta}) \qquad \boldsymbol{\eta}_i = g(\mu_i),$$

wobei h eine eineindeutige und zweimal differenzierbare Responsefunktion ist und g die Umkehrfunktion $g = h^{-1}$ von h ist. Der natürliche Parameter θ_i hängt zudem über

$$\mu_i = b'(\theta_i), \qquad \mu_i = h(\boldsymbol{x_i}.\boldsymbol{\beta})$$

von $\boldsymbol{\beta}$ ab. (Fahrmeir et al.; 2009, S. 221)

Die Poisson-Verteilung wäre nach der Formel

$$f(y; \lambda_{pois}) = \exp(y * \log(\lambda_{pois}) - \lambda_{pois} - \log(y!)),$$

mit

$$\theta = \log(\lambda_{pois})$$

$$a(\phi) = 1$$

$$b(\theta) = b'(\theta) = b''(\theta) = \exp(\theta) = \lambda_{pois}$$

$$c(y, \phi) = -\log(y!).$$

D.h. die Varianz ist in der Poisson-Verteilung gleich dem Erwartungswert. (Fahrmeir et al.; 2009, S. 218-220)

2.1. Korrigierte Score zur Bestimmung von korrigierten Beta-Koeffizienten

Zu Beginn haben wir n Beobachtungen und zu jeder Beobachtung einen Zeilenvektor $\boldsymbol{z}_i\boldsymbol{.}, i \in \{1, ..., n\}$, mit den dazu erhobenen Kovariablen. Die Menge der Zeilenvektoren $\boldsymbol{z}_i\boldsymbol{.}$ wird notiert als $n \times p$ Matrix $\boldsymbol{Z}$. Die y_i sind Zufallsvariablen deren Verteilung von den $\boldsymbol{z}_i\boldsymbol{.}$ abhängt. Die Menge aller y_i wird in einem Vektor $\boldsymbol{Y}$ notiert. (McCullagh und Nelder; 1989, S. 26)

Klassisches Messfehlermodell

Die Zeilenvektoren $\boldsymbol{x}_i\boldsymbol{.}$ stellen die mit einem Messfehler $\boldsymbol{\epsilon}_i\boldsymbol{.}$ beobachteten/gemessenen Werte für die $\boldsymbol{z}_i\boldsymbol{.}$ dar. Hier gilt

$$\boldsymbol{x}_i\boldsymbol{.} = \boldsymbol{z}_i\boldsymbol{.} + \boldsymbol{\epsilon}_i\boldsymbol{.}, \qquad E(\boldsymbol{\epsilon}_i\boldsymbol{.}|\boldsymbol{z}_i\boldsymbol{.}) = 0$$

wobei die $\boldsymbol{\epsilon}_{i\bullet}$, gegeben die wahren Werte $\boldsymbol{z}_{i\bullet}$, normalverteilt sind mit Erwartungswert 0 und bekannter Varianzmatrix $\boldsymbol{\Lambda}$. Wir haben hier also ein klassisches additives Messfehler Modell in dem die Fehlerterme eine konstante Varianz haben und die $\boldsymbol{x}_{i\bullet}$ eine unverzerrte Messung der $\boldsymbol{z}_{i\bullet}$ sind. (Carroll et al.; 2006, S. 3)

Klassische Messfehler könnten z.B. vorliegen, wenn eine Quantität mit einem Gerät gemessen wird und wiederholte Messungen um den wahren Wert variieren (Heid et al.; 2004, S. 367). Ein konkretes Beispiel wäre die Proteinaufnahme, die aber üblicherweise nicht direkt beobachtbar ist. Stattdessen wird oftmals ein Biomarker der Proteinaufnahme, genannt Harnstickstoff, gemessen. Der Biomarker erfasst die wahre Proteinaufnahme mit einer zusätzlichen Variabilität. (Carroll et al.; 2006, Kap. 1.2)

Modell mit Berksonfehler

Das Gegenstück dazu wäre ein Modell mit Berksonfehlern. Hier gilt

$$\boldsymbol{z}_{i\bullet} = \boldsymbol{x}_{i\bullet} + \boldsymbol{\epsilon}_{i\bullet}, \qquad E(\boldsymbol{\epsilon}_{i\bullet}|\boldsymbol{x}_{i\bullet}) = 0$$

und der bedingte Erwartungswert der $\boldsymbol{\epsilon}_{i\bullet}$, gegeben $\boldsymbol{x}_{i\bullet}$, gleich 0 ist. Daraus folgt, dass der wahre Wert eine größere Variabilität vorweist, als der gemessene Wert. (Carroll et al.; 2006, S. 5)

Berksonfehler liegen z.B. vor wenn ein Gruppendurchschnitt einem Individuum zugewiesen wird, welches den Charakteristiken der Gruppe entspricht. Der Gruppendurchschnitt ist der gemessene Wert und der latente Wert des Individuums ist der wahre Wert. Ein konkretes Beispiel wäre die Nutzung einer Job-Expositions-Matrix statt der Nutzung von individuellen Expositionswerten. (Heid et al.; 2004, S. 367)

Unterschied Klassisch vs. Berkson

Im klassischen Modell ist der Fehler $\boldsymbol{\epsilon}_{i\bullet}$ unabhängig von den $\boldsymbol{z}_{i\bullet}$, d.h. $E(\boldsymbol{\epsilon}_{i\bullet}|\boldsymbol{z}_{i\bullet}) = E(\boldsymbol{\epsilon}_{i\bullet})$ oder zumindest $E(\boldsymbol{\epsilon}_{i\bullet}|\boldsymbol{z}_{i\bullet}) = 0$. Im Modell mit den Berksonfehlern ist der Fehler $\boldsymbol{\epsilon}_{i\bullet}$ unabhängig von den $\boldsymbol{x}_{i\bullet}$, d.h. $E(\boldsymbol{\epsilon}_{i\bullet}|\boldsymbol{x}_{i\bullet}) = E(\boldsymbol{\epsilon}_{i\bullet})$ oder zumindest $E(\boldsymbol{\epsilon}_{i\bullet}|\boldsymbol{x}_{i\bullet}) = 0$. Daraus folgt, dass die Varianz für die mit Messfehlern gemessenen Werte im klassischen Modell größer ist als die Varianz der wahren Werte und umgekehrt im Modell mit Berksonfehlern. Also

$$var(\boldsymbol{x}_{i\bullet}) > var(\boldsymbol{z}_{i\bullet})$$

für das klassische Modell und

$$var(\boldsymbol{z}_{i\bullet}) > var(\boldsymbol{x}_{i\bullet})$$

für das Modell mit Berksonfehlern. (Carroll et al.; 2006, Kap. 2.2.2)

Möchte man sich selbst überzeugen, dass man eine große Menge statistische Power hat trotz Messfehler, dann wird empfohlen den Messfehler als Berksonfehler zu sehen (Carroll et al.; 2006, S. 5). Dies kann allerdings dazu führen, dass die Power weit überschatzt wird, wenn der Messfehler in Wirklichkeit klassisch ist. (Carroll et al.; 2006, S. 6)

Die korrigierte Score-Funktion lässt sich im Falle der Normalverteilung auch auf multiplikative Messfehlern anwenden (Nakamura; 1990, S. 132). Es ist zudem möglich den Fall zu betrachten in dem manche Komponenten von $\boldsymbol{z}_{i\bullet}$ ohne Messfehler gemessen werden, allerdings hat dann die Varianzmatrix $\boldsymbol{\Lambda}$ nicht mehr vollen Rang (Buzas und Stefanski; 1995).

Matrixschreibweise

Die Menge der Vektoren $\boldsymbol{x}_{i\bullet}$ wird als $n \times p$ Matrix $\boldsymbol{X}$ bezeichnet. (Nakamura; 1990, Kap. 2) Die β_j, $j \in \{1, ..., p\}$, sind die Parameterwerte und werden in einem Spaltenvektor $\boldsymbol{\beta}$ der Länge p notiert. In Matrixnotation lässt sich das Ganze folgendermaßen darstellen:

$$\boldsymbol{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \boldsymbol{Z} = \begin{pmatrix} z_{11} & \cdots & z_{1p} \\ z_{21} & \ddots & \\ \vdots & & \\ z_{n1} & \cdots & z_{np} \end{pmatrix} = \begin{pmatrix} \boldsymbol{z}_{1\bullet} \\ \boldsymbol{z}_{2\bullet} \\ \vdots \\ \boldsymbol{z}_{n\bullet} \end{pmatrix}$$

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_{11} & \cdots & \epsilon_{1p} \\ \epsilon_{21} & \ddots & \\ \vdots & & \\ \epsilon_{n1} & \cdots & \epsilon_{np} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\epsilon}_{1\bullet} \\ \boldsymbol{\epsilon}_{2\bullet} \\ \vdots \\ \boldsymbol{\epsilon}_{n\bullet} \end{pmatrix}, \boldsymbol{\Lambda} = \begin{pmatrix} \sigma^2 & \cdots & 0 \\ 0 & \sigma^2 & \\ \vdots & & \ddots \\ 0 & \cdots & \sigma^2 \end{pmatrix}$$

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ x_{21} & \ddots & \\ \vdots & & \\ x_{n1} & \cdots & x_{np} \end{pmatrix} = \begin{pmatrix} z_{11} + \epsilon_{11} & \cdots & z_{1p} + \epsilon_{1p} \\ z_{21} + \epsilon_{21} & \ddots & \\ \vdots & & \\ z_{n1} + \epsilon_{n1} & \cdots & z_{np} + \epsilon_{np} \end{pmatrix} = \begin{pmatrix} \boldsymbol{z}_{1\cdot} + \boldsymbol{\epsilon}_{1\cdot} \\ \boldsymbol{z}_{2\cdot} + \boldsymbol{z}_{2\cdot} \\ \vdots \\ \boldsymbol{z}_{n\cdot} + \boldsymbol{\epsilon}_{n\cdot} \end{pmatrix}.$$

(Nakamura; 1990) verwendet die Notation $\boldsymbol{\beta^T x_i}$, wobei $\boldsymbol{\beta}^T$ ein Zeilenvektor und $\boldsymbol{x_i}$ ein Spaltenvektor eines Individuums ist. Für eine gleichbleibende Notation wird in dieser Arbeit $\boldsymbol{x_i \cdot \beta}$ verwendet, welches zum selben Ergebnis führt. Die für Messfehler korrigierten Schätzungen werden im Folgenden als $\boldsymbol{\beta^{cor}}$ bezeichnet. Wir bezeichnen den wahren Parameterwert mit $\boldsymbol{\beta_t}$.

Naive Score-Funktion

Üblicherweise werden die Beta-Koeffizienten über die Score-Funktion $\boldsymbol{s(\beta, Z, Y)}$ geschätzt und das $\boldsymbol{\beta_x}$, welches die Gleichung $\boldsymbol{s(\beta_x, Z, Y)} = 0$ erfüllt, wird als Maximum-Likelihood-Schätzung bezeichnet (Fahrmeir et al.; 2009, S. 223). In vielen Fällen ist $\boldsymbol{Z}$ allerdings nicht gegeben, sondern nur $\boldsymbol{X}$. Ersetzt man jetzt $\boldsymbol{Z}$ durch $\boldsymbol{X}$ und schätzt die Beta-Koeffizienten durch $\boldsymbol{s(\beta, X, Y)} = 0$, so nennt man dies eine „naive“ Score-Funktion. Im Allgemeinen gilt nicht immer, dass $E(\boldsymbol{s(\beta_t, X, Y)|Y}) = 0$, d.h. selbst mit dem wahren Parameterwert $\boldsymbol{\beta_t}$ ist der auf $\boldsymbol{Y}$ bedingte Erwartungswert der naiven Score-Funktion nicht immer 0. Somit ist das $\boldsymbol{\beta_x}$, für welches gilt $\boldsymbol{s(\beta_x, X, Y)} = 0$, nicht unbedingt ein konsistenter Schätzer. (Nakamura; 1990, Kap. 2)

Korrigierte Schätzung

Für konsistente Schätzer orientieren wir uns an Nakamura und definieren eine korrigierte Log-Likelihood-Funktion, eine korrigierte Score-Funktion und eine korrigierte Information. Mithilfe dieser Funktionen erhalten wir den korrigierten Schätzer $\boldsymbol{\beta^{cor}}$. Die Idee ist, dass die bedingte Verteilung der korrigierten Schätzung, gegeben die Variablen $\boldsymbol{Y}$ und $\boldsymbol{Z}$, um die Maximum-Likelihood-Schätzung zentriert ist. Diese wiederum ist um den wahren Parameterwert zentriert, womit wir einen unverzerrten Schätzer haben. (Nakamura; 1990, S. 127) Wir haben eine konvexe Teilmenge eines Parameterraums F, welche den wahren Parameterwert $\boldsymbol{\beta_t}$ beinhaltet und haben eine korrigierte Log-Likelihood-Funktion $\boldsymbol{l^{cor}(\beta, X, Y)}$, wenn der auf $\boldsymbol{Z}$ und $\boldsymbol{Y}$ bedingte Erwartungswert der wahren

Log-Likelihood-Funktion entspricht. D.h. :

$$E(l^{cor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})|\boldsymbol{Z}, \boldsymbol{Y}) = l(\boldsymbol{\beta}, \boldsymbol{Z}, \boldsymbol{Y}) \tag{2.2}$$

Wenn $l^{cor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})$ differenzierbar ist in F, so nennt man die Ableitung $\frac{\partial l^{cor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})}{\partial \boldsymbol{\beta}}$ eine korrigierte Score-Funktion $\boldsymbol{s^{cor}}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})$. Wenn $E(\ldots|\boldsymbol{Z}, \boldsymbol{Y})$ und $\partial\boldsymbol{\beta}$ austauschbar sind,

$$E(\boldsymbol{s^{cor}}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})|\boldsymbol{Z}, \boldsymbol{Y}) = \boldsymbol{s}(\boldsymbol{\beta}, \boldsymbol{Z}, \boldsymbol{Y}). \tag{2.3}$$

(Nakamura; 1990, Kap. 2). Die korrigierte Score-Funktion ist zugleich der Gradient der korrigierten Log-Likelihood-Funktion. Wenn $\boldsymbol{s^{cor}}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})$ differenzierbar ist in F, so nennt man die negative Ableitung $\frac{-\partial \boldsymbol{s^{cor}}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})}{\partial \boldsymbol{\beta}}$ eine korrigierte Information $\boldsymbol{I^{cor}}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})$. Wenn $E(\ldots|\boldsymbol{Z}, \boldsymbol{Y})$ und $\partial\boldsymbol{\beta}$ austauschbar sind,

$$E(\boldsymbol{I^{cor}}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})|\boldsymbol{Z}, \boldsymbol{Y}) = \boldsymbol{I}(\boldsymbol{\beta}, \boldsymbol{Z}, \boldsymbol{Y}). \tag{2.4}$$

Die korrigierte Information ist die negative Hesse-Matrix. Wir haben eine korrigierte Schätzung $\boldsymbol{\beta^{cor}}$, wenn $\boldsymbol{s^{cor}}(\boldsymbol{\beta^{cor}}, \boldsymbol{X}, \boldsymbol{Y}) = 0$ und $\boldsymbol{I^{cor}}(\boldsymbol{\beta^{cor}}, \boldsymbol{X}, \boldsymbol{Y})$ positiv definit ist. (Nakamura; 1990, Kap. 2) Die positive Definitheit der korrigierten Information ist äquivalent zur negativen Definitheit der Hesse-matrix und damit steht fest, dass es sich um ein Maximum handelt.

Der Vorteil wird deutlich, dass die Korrigierte Score-Funktion keine Annahme über die Verteilung von den nicht vorhandenen $\boldsymbol{Z}$ trifft (Buzas; 2009, Kap. 1). Da die $\boldsymbol{Z}$ unbekannt sind, ist die korrigierte Score-Funktion eine funktionelle Methode (Huang et al.; 2015) (Carroll et al.; 2006, Kap. 2.1). Das Gegenstück wäre eine strukturelle Methode, in der die $\boldsymbol{Z}$ als Zufallsvariable betrachtet werden (Carroll et al.; 2006, Kap 2.1). Falls alle Messfehler 0 sind, entsprechen die korrigierten Funktionen den naiven Funktionen.

Unverzerrtheit

Für den Erwartungswert von $E(\boldsymbol{s^{cor}}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})|\boldsymbol{Z}, \boldsymbol{Y})$ gegeben $\boldsymbol{Y}$ gilt

$$E(E(\boldsymbol{s^{cor}}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})|\boldsymbol{Z}, \boldsymbol{Y})|\boldsymbol{Y}) = E(\boldsymbol{s}(\boldsymbol{\beta}, \boldsymbol{Z}, \boldsymbol{Y})|\boldsymbol{Y}).$$

(Nakamura; 1990) Und somit $E(E(\boldsymbol{s^{cor}}(\boldsymbol{\beta_t}, \boldsymbol{X}, \boldsymbol{Y})|\boldsymbol{Z}, \boldsymbol{Y})|\boldsymbol{Y}) = 0$, was darauf hindeutet, dass die korrigierte Score-Funktion eine unverzerrte Score-Funktion ist (Stefanski

und Carroll; 1987). Allerdings ist nicht jede unverzerrte Score-Funktion unbedingt eine korrigierte Score-Funktion (Nakamura; 1990).

Newton-Raphson Verfahren

Da die Poisson-Verteilung betrachtet werden soll und die korrigierte Lösung nicht direkt berechenbar ist, wird die Nullstelle üblicherweise mit dem Newton-Raphson Verfahren ermittelt (Fahrmeir et al.; 2009, S. 473). Um den Startwert für die korrigierte Log-Likelihood-Funktion zu bekommen, lässt man das Verfahren mit einem Startwert von 0 mit der naiven Log-Likelihood-Funktion konvergieren. Die sich daraus ergebenden Werte, werden in das Newton-Raphson Verfahren mit der korrigierten Log-Likelihood-Funktion eingesetzt. (Nakamura; 1990, Kap. 6)

Wenn wir eine Funktion durch eine Taylorreihe erster Ordnung an der Stelle x_0 approximieren und gleich 0 setzen ergibt sich das allgemeine Newton-Raphson-Verfahren als

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) \stackrel{!}{=} 0$$

bzw. nach einigen Umformungen

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

mit x_0 als Startwert und x als neuer Startwert. Das Verfahren führt man so oft aus, bis eine frei wählbare Toleranzgrenze erreicht wurde. Für unseren speziellen, mehrdimensionalen Fall ergibt sich der iterative Algorithmus

$$\boldsymbol{\beta_{i+1}} = \boldsymbol{\beta_i} - \boldsymbol{H_{(l^{cor}(\beta_i, X, Y))}(\beta_i)^{-1} * s^{cor}(\beta_i, X, Y)}.$$

$f(x)$ ist die korrigierte Score-Funktion und $f'(x)$ ist die Hesse-Matrix. $\boldsymbol{\beta_{i+1}}$ ist der neue Wert für den Betavektor nach einer Iteration und $\boldsymbol{\beta_i}$ ist der Wert vom ersten/vorherigen Schritt. (Fahrmeir et al.; 2009, S. 473-475)

Invertierbarkeit der Hesse-Matrix

Es kann vorkommen, dass die Hesse-Matrix nicht invertierbar ist und somit keine Lösung berechnet werden kann. Nicht-Invertierbarkeit der Hesse-Matrix liegt normalerweise an einer Multikollinearität (Gill und King; 2004, S. 55). Um die Invertierbarkeit der Hesse-Matrix zu prüfen gibt es mehrere Möglichkeiten, z.B. sicherstellen, dass die Determinante

ungleich 0 ist oder dass alle Eigenwerte größer 0 sind. Je nach Vorliebe kann man die Gleichung auch umstellen und erhält

$$\boldsymbol{H}_{(l^{cor}(\boldsymbol{\beta_i},\boldsymbol{X},\boldsymbol{Y}))}(\boldsymbol{\beta_i}) * \boldsymbol{\Delta\beta_i} = -\boldsymbol{s^{cor}}(\boldsymbol{\beta_i}, \boldsymbol{X}, \boldsymbol{Y}) \tag{2.5}$$

mit

$$\boldsymbol{\Delta\beta_i} = \boldsymbol{\beta_{i+1}} - \boldsymbol{\beta_i}.$$

Wahl des Startwerts für das Newton-Raphson-Verfahren

Für die Gleichung $\boldsymbol{s^{cor}}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \boldsymbol{0}$ ist es möglich, dass mehrere Nullstellen vorhanden sind (Nakamura; 1990, Kap. 6). Nakamura führte Berechnungen dazu durch und kam zu dem Ergebnis, dass unpassende Schätzungen für $\boldsymbol{s^{cor}}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \boldsymbol{0}$ sowohl von den passenden, als auch von den naiven Schätzern weit entfernt liegen. Es ist also angemessen, den naiven Schätzer als Startwert zu nehmen in diesen Modellen. Das stimmt mit (Stefanski und Carroll; 1987) überein.

Beispiel korrigierte Log-Likelihood-Funktion Normal-Verteilung

Für eine Zufallsvariable $y_i = \boldsymbol{z_i.}\boldsymbol{\beta} + \boldsymbol{\delta_i.}$, in der die $\boldsymbol{\delta_i.}$ identisch und unabhängig normalverteilt sind mit Erwartungswert 0 und Varianz σ^2, ergibt sich die Log-Likelihood-Funktion als

$$l(\boldsymbol{\beta}, \boldsymbol{Z}, \boldsymbol{Y}) = -\frac{1}{2}n\log(2\pi) - n\log(\sigma) - \frac{1}{2}\sigma^{-2}\sum_{i=1}^{n}(y_i - \boldsymbol{z_i.}\boldsymbol{\beta})^2$$

und wir definieren unser $l^{cor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})$ als

$$l^{cor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = -\frac{1}{2}n\log(2\pi) - n\log(\sigma) - \frac{1}{2}\sigma^{-2}\sum_{i=1}^{n}((y_i - \boldsymbol{x_i.}\boldsymbol{\beta})^2 - \boldsymbol{\beta^T\Lambda\beta}).$$

Für die Herleitungen der korrigierten Funktionen mit normalverteiltem Messfehler gilt allgemein:

$$\boldsymbol{x_i.} = \boldsymbol{z_i.} + \boldsymbol{\epsilon_i.}$$

$$\boldsymbol{\theta}_i = \boldsymbol{z_i.}\boldsymbol{\beta} \tag{2.6}$$

$$Var(\boldsymbol{\epsilon_i.}) = E(\boldsymbol{\epsilon_i.}^2) - \underbrace{E(\boldsymbol{\epsilon_i.})^2}_{=0} = E(\boldsymbol{\epsilon_i.}^2) = \boldsymbol{\Lambda} \tag{2.7}$$

$$E(\exp(\boldsymbol{x}_i.\boldsymbol{\beta})|\boldsymbol{Z},\boldsymbol{Y}) = \exp(\boldsymbol{z}_i.\boldsymbol{\beta} + \xi) \tag{2.8}$$

mit

$$\xi = \frac{\boldsymbol{\beta^T \Lambda \beta}}{2}$$

$$E(\boldsymbol{x}_i. \exp(\boldsymbol{x}_i.\boldsymbol{\beta})|\boldsymbol{Z},\boldsymbol{Y}) = (\boldsymbol{z}_i.^{\boldsymbol{T}} + \boldsymbol{\Lambda\beta}) \exp(\boldsymbol{z}_i.\boldsymbol{\beta} + \xi) \tag{2.9}$$

$$E(\boldsymbol{x}_i.\boldsymbol{x}_i.^T \exp(\boldsymbol{x}_i.\boldsymbol{\beta})|\boldsymbol{Z},\boldsymbol{Y}) = (\boldsymbol{\Lambda} + (\boldsymbol{z}_i.^T + \boldsymbol{\Lambda\beta}) + (\boldsymbol{z}_i.^{\boldsymbol{T}} + \boldsymbol{\Lambda\beta})^{\boldsymbol{T}}) \exp(\boldsymbol{z}_i.\boldsymbol{\beta} + \xi). \tag{2.10}$$

2.6 ist die normale Annahme, solange keine Reparametrisierung nötig ist (Nakamura; 1990, Kap. 4.1). 2.7 ist der Verschiebungssatz (Fahrmeir et al.; 2016, S. 250) und 2.8 lässt sich über die Momenterzeugende Funktion bestimmen mit

$$E(\exp(\boldsymbol{x}_i.\boldsymbol{\beta})|\boldsymbol{Z},\boldsymbol{Y}) = \exp\left(\boldsymbol{\beta^T} \boldsymbol{z}_i.^T + \frac{\boldsymbol{\beta^T \Lambda \beta}}{2}\right)$$

(Buzas und Stefanski; 1995, S. 2). Die Bezeichnung scheint hier etwas verwirrend zu sein, das liegt daran, dass Nakamura eine andere Schreibweise verwendet hat. 2.9 und 2.10 sind die nächsthöheren Ableitungen nach $\boldsymbol{\beta}$ und helfen bei der Bestimmung von der korrigierten Score-Funktion und der korrigierten beobachteten Information. (Nakamura; 1990, S. 131)

Die Herleitung des auf $\boldsymbol{Z}$ und $\boldsymbol{Y}$ bedingten Erwartungswertes der korrigierten Log-Likelihood-Funktion für die Normalverteilung befindet sich auf der nächsten Seite.

$$
\begin{aligned}
E(l^{cor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})|\boldsymbol{Z}, \boldsymbol{Y}) &= E(-\frac{1}{2}n\log(2\pi) - n\log(\sigma) - \frac{1}{2}\sigma^{-2}\sum_{i=1}^{n}((y_i - \boldsymbol{x_i.}\boldsymbol{\beta})^2 - \boldsymbol{\beta^T}\boldsymbol{\Lambda}\boldsymbol{\beta})|\boldsymbol{Z}, \boldsymbol{Y}) \\
&= \underbrace{E(-\frac{1}{2}n\log(2\pi)|\boldsymbol{Z}, \boldsymbol{Y})}_{=-\frac{1}{2}n\log(2\pi)} - \underbrace{E(n\log(\sigma)|\boldsymbol{Z}, \boldsymbol{Y})}_{=n\log(\sigma)} - \\
&\quad E(\frac{1}{2}\sigma^{-2}\sum_{i=1}^{n}((y_i - (\boldsymbol{z_i.} + \boldsymbol{\epsilon_i.})\boldsymbol{\beta})^2 - \boldsymbol{\beta^T}\boldsymbol{\Lambda}\boldsymbol{\beta})|\boldsymbol{Z}, \boldsymbol{Y}) \\
&= -\frac{1}{2}n\log(2\pi) - n\log(\sigma) - \frac{1}{2}\sigma^{-2}E(\sum_{i=1}^{n}((y_i - (\boldsymbol{z_i.} + \boldsymbol{\epsilon_i.})\boldsymbol{\beta})^2 - \boldsymbol{\beta^T}\boldsymbol{\Lambda}\boldsymbol{\beta}|\boldsymbol{Z}, \boldsymbol{Y}) \\
&= -\frac{1}{2}n\log(2\pi) - n\log(\sigma)) - \frac{1}{2}\sigma^{-2}E(\sum_{i=1}^{n}(y_i^2 \\
&\quad - 2y_i(\boldsymbol{z_i.} + \boldsymbol{\epsilon_i.})\boldsymbol{\beta} + ((\boldsymbol{z_i.} + \boldsymbol{\epsilon_i.})\boldsymbol{\beta})^2 - \boldsymbol{\beta^T}\boldsymbol{\Lambda}\boldsymbol{\beta})|\boldsymbol{Z}, \boldsymbol{Y}) \\
&= -\frac{1}{2}n\log(2\pi) - n\log(\sigma)) - \frac{1}{2}\sigma^{-2}\sum_{i=1}^{n}(\underbrace{E(y_i^2|\boldsymbol{Z}, \boldsymbol{Y})}_{=y_i^2} \\
&\quad - E(2y_i(\boldsymbol{z_i.} + \boldsymbol{\epsilon_i.})\boldsymbol{\beta}|\boldsymbol{Z}, \boldsymbol{Y})) \\
&\quad + (\boldsymbol{E}((\boldsymbol{z_i.} + \boldsymbol{\epsilon_i.})\boldsymbol{\beta})^2|\boldsymbol{Z}, \boldsymbol{Y}) - \underbrace{\boldsymbol{E}(\boldsymbol{\beta^T}\boldsymbol{\Lambda}\boldsymbol{\beta}|\boldsymbol{Z}, \boldsymbol{Y})}_{=\boldsymbol{\beta^T}\boldsymbol{\Lambda}\boldsymbol{\beta}} \\
&= -\frac{1}{2}n\log(2\pi) - n\log(\sigma) - \frac{1}{2}\sigma^{-2}\sum_{i=1}^{n}(y_i^2 - E(2y_i(\boldsymbol{z_i.} + \boldsymbol{\epsilon_i.})\boldsymbol{\beta}|\boldsymbol{Z}, \boldsymbol{Y}) \\
&\quad + (\boldsymbol{E}((\boldsymbol{z_i.} + \boldsymbol{\epsilon_i.})\boldsymbol{\beta})^2|\boldsymbol{Z}, \boldsymbol{Y}) - \boldsymbol{\beta^T}\boldsymbol{\Lambda}\boldsymbol{\beta}) \\
&= -\frac{1}{2}n\log(2\pi) - n\log(\sigma) - \frac{1}{2}\sigma^{-2}\sum_{i=1}^{n}(y_i^2 - \underbrace{E(2y_i\boldsymbol{z_i.}\boldsymbol{\beta}|\boldsymbol{Z}, \boldsymbol{Y})}_{=2y_i\boldsymbol{z_i.}\boldsymbol{\beta}} \\
&\quad - \underbrace{E(2y_i\boldsymbol{\epsilon_i.}\boldsymbol{\beta}|\boldsymbol{Z}, \boldsymbol{Y})}_{=0} \\
&\quad + \underbrace{\boldsymbol{E}((\boldsymbol{z_i.}\boldsymbol{\beta})^2|\boldsymbol{Z}, \boldsymbol{Y})}_{=(\boldsymbol{z_i.}\boldsymbol{\beta})^2} + \underbrace{E(2\boldsymbol{z_i.}\boldsymbol{\epsilon_i.}\boldsymbol{\beta}|\boldsymbol{Z}, \boldsymbol{Y})}_{=0} + \underbrace{\boldsymbol{E}(\boldsymbol{\epsilon_i.}\boldsymbol{\beta})^2|\boldsymbol{Z}, \boldsymbol{Y})}_{=\boldsymbol{\beta^T}\boldsymbol{E}(\boldsymbol{\epsilon_i.}^2|\boldsymbol{Z}, \boldsymbol{Y})\boldsymbol{\beta}} - \boldsymbol{\beta^T}\boldsymbol{\Lambda}\boldsymbol{\beta}) \\
&= -\frac{1}{2}n\log(2\pi) - n\log(\sigma) - \frac{1}{2}\sigma^{-2}\sum_{i=1}^{n}(y_i^2 - 2y_i\boldsymbol{z_i.}\boldsymbol{\beta} \\
&\quad + (\boldsymbol{z_i.}\boldsymbol{\beta})^2 + \boldsymbol{\beta^T}\underbrace{\boldsymbol{E}(\boldsymbol{\epsilon_i.}^2|\boldsymbol{Z}, \boldsymbol{Y})}_{=\Lambda}\boldsymbol{\beta} - \boldsymbol{\beta^T}\boldsymbol{\Lambda}\boldsymbol{\beta}) \\
&= -\frac{1}{2}n\log(2\pi) - n\log(\sigma) - \frac{1}{2}\sigma^{-2}\sum_{i=1}^{n}(y_i - \boldsymbol{z_i.}\boldsymbol{\beta})^2 \\
&= l(\boldsymbol{\beta}, \boldsymbol{Z}, \boldsymbol{Y}),
\end{aligned}
$$

womit das $l^{cor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})$ nach Definition 2.2 auch wirklich eine korrigierte Log-Likelihood-Funktion ist und somit ein unverzerrter Schätzer der wahren Log-Likelihood-Funktion.

2.1.1. Korrigierte Score-Funktion Poisson-Verteilung

Für eine poisson-verteilte Zufallsvariable $\boldsymbol{Y}$ mit Erwartungswert $\exp(\boldsymbol{z_i.\beta})$ ergibt sich die korrigierte Log-Likelihood-Funktion als

$$l^{cor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{n}(-\exp(\boldsymbol{x_i.\beta} - \xi) + y_i\boldsymbol{x_i.\beta} - \log(y_i))) \tag{2.11}$$

Der Gradient/die dazugehörige korrigierte Score-Funktion ergibt sich als

$$\boldsymbol{s^{cor}}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{n}(y_i.\boldsymbol{x_i.}^T - (\boldsymbol{x_i.}^T - \boldsymbol{\Lambda\beta})\exp(\boldsymbol{x_i.\beta} - \xi)). \tag{2.12}$$

Und die negative Hesse-Matrix/korrigierte Information ergibt sich als

$$\boldsymbol{I^{cor}}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{n}((\boldsymbol{x_i.^T} - \boldsymbol{\Lambda\beta})(\boldsymbol{x_i.^T} - \boldsymbol{\Lambda\beta})^{\boldsymbol{T}} - \boldsymbol{\Lambda}) * \exp(\boldsymbol{x_i.\beta} - \xi). \tag{2.13}$$

(Nakamura; 1990, S. 132) Die Herleitungen sind analog zur Normal-Verteilung. Die korrigierten Funktionen für die Poisson-Verteilung sollen in den weiteren Schritten als Ausgangspunkt genommen werden.

2.2. Regularisierungssverfahren

LASSO-Verfahren

Wir haben nun eine Methode für die Schätzung von korrigierten Beta-Koeffizienten und wollen diese Variablen schrumpfen und selektieren. Dazu benutzen wir das LASSO-Verfahren, kurz für „least absolute shrinkage and selection operator“. Die LASSO Schätzung $(\alpha^{lasso}, \boldsymbol{\beta^{lasso}})$ in einem linearen Regressionsfall ist allgemein definiert als

$$(\alpha^{lasso}, \boldsymbol{\beta^{lasso}}) = argmin\left\{\sum_{i=1}^{N}(y_i - \alpha - \sum_{j=1}^{p}\beta_j x_{ij})^2\right\}, \qquad \sum_{j=1}^{p}|\beta_j| \leq t \tag{2.14}$$

(Tibshirani; 1996, S. 268). Das LASSO-Verfahren minimiert also die Residuenquadratsumme unter der Nebenbedingung, dass die Summe der Absolutbeträge der Beta-Koeffizienten kleiner ist als eine Konstante. Wir gehen von zentrierten Variablen aus, d.h. $\frac{\sum_{i=1}^{n} x_{ij}}{N} = 0$ und $\frac{\sum_{i=1}^{n} x_{ij}^2}{N} = 1$. Folglich fällt das α raus, da $\alpha^{lasso} = \bar{y}$ und $\bar{y} = 0$. Und die Designmatrix $\boldsymbol{X}$ bleibt $n \times p$ (Hastie et al.; 2009). Es wird also ein Modell ohne Intercept geschätzt (Hastie et al.; 2009, Kap. 3.4.2.). Die Berechnung der Lösung von 2.14 ist ein quadratisches Programmierungsproblem mit linearen Ungleichheitsbedingungen (Tibshirani; 1996, S. 267-268). Für ein ausreichend kleines t werden manche Koeffizienten genau auf 0 gesetzt. Das liegt an der Natur der Nebenbedingung (Hastie et al.; 2009, S. 69). Die Nebenbedingung ist auch bekannt als L1-Norm $||\beta||_1$. Die Gleichung 2.14 lässt sich auch in der Lagrange-Form schreiben als

$$(\boldsymbol{\beta^{lasso}}) = \underset{\beta}{argmin} \sum_{i=1}^{N} (y_i - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j|, \tag{2.15}$$

auf welcher der Fokus in dieser Arbeit liegen soll (Hastie et al.; 2009, S. 68). Hier ist das λ der Parameter, der die Stärke der Schrumpfung regularisiert. 2.14 und 2.15 sind äquivalent, da für ein gegebenes $\lambda \in [0, \infty)$ ein $t \geq 0$ existiert, so dass beide Probleme die selbe Lösung liefern und umgekehrt (Leng et al.; 2006, S. 1275). Die Konstante t kann hierbei weggelassen werden, da diese keine der interessierenden Variablen enthält.

Ridge Regression

Für die weiteren Methoden gehen wir wieder von standardisierten Beobachtungen aus und lassen den Intercept weg. Würde man die L2-Norm $||\beta||_2$ als Strafterm benutzen hätte man Ridge Regression in der Form

$$(\boldsymbol{\beta^{ridge}}) = \underset{\beta}{argmin} \sum_{i=1}^{N} (y_i - \sum_{j=1}^{p} \beta_j x_{ij})^2, \qquad \sum_{j=1}^{p} \beta_j^2 \leq t.$$

Hier werden die quadrierten Beta-Koeffizienten als Nebenbedingung für die Minimierung hergenommen. Der wesentliche Unterschied zum LASSO-Verfahren ist, dass Ridge Regression im Allgemeinen keine Koeffizienten auf 0 setzt. D.h. es wird keine Variablenselektion vorgenommen, sondern nur eine Schrumpfung der Koeffizienten (Tibshirani;

1996). Ridge Regression lässt sich ebenfalls in der Lagrange-Form darstellen als

$$(\boldsymbol{\beta^{ridge}}) = \underset{\beta}{argmin} \left\{ \sum_{i=1}^{N} (y_i - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}.$$

Elastisches Netz

Es ist möglich das LASSO-Verfahren und Ridge Regression „gleichzeitig“ anzuwenden. Man erhält dann das Elastische Netz Kriterium, welches sich darstellen lässt als

$$(\boldsymbol{\beta^{elastic}}) = argmin \sum_{i=1}^{N} (y_i - \sum_{j}^{p} \beta_j x_{ij})^2 + \lambda P_\alpha(\boldsymbol{\beta}), \tag{2.16}$$

wobei

$$\lambda P_\alpha(\boldsymbol{\beta}) = \lambda(\alpha \sum_{j=1}^{p} |\beta_j| + \frac{1}{2}(1-\alpha) \sum_{j=1}^{p} \beta_j^2)$$

der Strafterm des elastischen Netzes ist. (Friedman et al.; 2010, S. 3) Für $\alpha = 0$ erhält man einfache Ridge Regression und für $\alpha = 1$ erhält man das LASSO-Verfahren. (Zou und Hastie; 2005, S. 303) Für den Übergang von 0 bis 1 für α mit einem fixen λ geht das Verhalten der Lösung von Ridge-artig zu LASSO-artig. D.h. je größer α, desto mehr Koeffizienten sind auf 0 (Friedman et al.; 2010, S. 3) Der erste Term fördert ein sparsameres Modell, während der zweite Term eine Mittelung von hoch korrelierten Variablen fördert. Der Term α legt das Mischverhältnis von dem LASSO-Verfahren und der Ridge Regression fest. (Hastie et al.; 2009, S. 662-663)

Teilmengenselektion

Die Teilmengenselektion findet für jedes $k \in \{0, 1, 2, \ldots, p\}$ die Teilmenge der Größe k, die die kleinste Residuenquadratsumme aufweist. Ein effizienter Algorithmus, die Sprünge und Grenzen Prozedur, macht das möglich für ein p bis zu einer Größe von 40 (Furnival und Wilson; 2012) Eine Teilmenge der Größe 2 muss dabei nicht unbedingt die Variable beinhalten, die in der Teilmenge der Größe 1 gewesen wäre. Für die Wahl von k können verschiedene Kriterien verwendet werden (Hastie et al.; 2009, Kap. 3.3.1)
.

Eigenschaften der Regularisierungsverfahren

Die Verfahren kann man generalisieren durch einen Strafterm der Form $(\sum_{j=1}^{p} \beta_k^q)^{\frac{1}{q}}$, wobei Ridge Regression vorliegt bei $q = 2$ und LASSO bei $q = 1$. Teilmengenselektion korrespondiert zu q gegen 0, woraus folgt, dass das LASSO näher an der Teilmengenselektion liegt (Tibshirani; 2011, Kap. 1). (Tibshirani; 1996, Kap. 11) hat in Simulationsstudien das LASSO-Verfahren der Ridge Regression und der Teilmengenselektion gegenüber gestellt und die Vorteile in drei Szenarien begutachtet:

1. **wenige starke Effekte**: Subset Selection performte am Besten, LASSO nicht ganz so gut und Ridge Regression performte schlecht

2. **wenige bis moderate Anzahl von mäßigen Effekten**: LASSO performt am Besten, gefolgt von Ridge Regression und dann Subset Selection

3. **große Anzahl von kleinen Effekten**: Ridge Regression performte mit Abstand am Besten, gefolgt von LASSO und Subset Selection.

Die Ergebnisse beziehen sich auf Vorhersagegenauigkeit. (Tibshirani; 1996, S. 286). Man sieht zwar, dass keines der Verfahren die anderen beiden deutlich übertrifft, aber da Variablenselektion immer wichtiger wird in der modernen Datenanalyse, wirken die Teilmengenselektion und das LASSO attraktiver. (Zou und Hastie; 2005, S. 302)

Einschränkungen des LASSO

Trotz der vielen Vorteile kann das LASSO auch auf Schwierigkeiten stoßen. Gegeben sind die folgenden drei Szenarien:

1. Im Fall von $p > n$, selektiert das LASSO höchstens n Variablen, das liegt in der Natur eines konvexen Optimierungsproblems. Zudem ist das LASSO dann nicht genau definiert, falls die Beschränkung der L1-Norm größer gleich ist als ein bestimmter Wert.

2. Wenn eine Gruppe von Variablen existiert, in der die paarweise Korrelationen sehr groß sind, dann wählt das LASSO willkürlich eine Variable aus dieser Gruppe

3. Für gewöhnliche $n > p$ Situationen, in denen hohe Korrelationen zwischen den Prädiktoren bestehen, wurde beobachtet, dass die Ridge Regression besser performt als das LASSO.

(Zou und Hastie; 2005, S. 302) Ein Fall in dem diese Probleme auftreten könnten wäre das Genselektionsproblem. Ein Datensatz hat oftmals viele Prädiktoren und oft weniger als 100 Proben ($p > n$). Zudem haben die Gene mit dem selben „biologischen Pfad“ oft eine hohe Korrelation (Segal et al.; 2003). Ein optimales Verfahren sollte nun die trivialen Gene entfernen und eine ganze Gruppe inkludieren, sofern ein Gen aus der Gruppe selektiert wurde. Das LASSO ist nicht die beste Wahl für diese Aufgabe (Zou und Hastie; 2005, S. 302). Das von (Zou und Hastie; 2005) vorgeschlagene Elastische Netz kombiniert das LASSO mit der Ridge Regression und stellt somit eine Erweiterung zum LASSO-Verfahren dar. In Simulationsstudien stellten sie fest, dass das Elastische Netz bessere Vorhersagen liefert als das LASSO, besser unter Kollinearität performt und mehr Variablen beibehalten kann (Gruppierungseffekt) (Zou und Hastie; 2005, Kap. 5). Die Ergebnisse sprechen also für eine Überlegenheit des Elastischen Netzes gegenüber dem LASSO (Zou und Hastie; 2005, Kap. 7). In dieser Arbeit soll der Fokus trotzdem nur auf das LASSO gelegt werden, da das Problem der Minimierung des naiven elastischen Netzes äquivalent ist zum LASSO Optimierungsproblem. D.h. das Elastische Netz genießt ebenfalls die rechnerischen Vorteile des LASSO. (Zou und Hastie; 2005, S. 303) Falls also eine Lösung für das LASSO in diesem speziellen Fall gefunden wird, lässt sich in den meisten Fällen der Strafterm auf das elastische Netz erweitern.

LASSO für GLMs

Das LASSO-Verfahren soll nun auf GLMs angewendet werden. Hierbei wird eine Funktion maximiert, unter der Nebenbedingung $\sum_{j=1}^{p} |\beta_j| \leq t$. In diesem Fall maximieren wir die korrigierte Log-Likelihood-Funktion in der Lagrange-Form, also

$$\boldsymbol{\beta}^{\boldsymbol{cor,lasso}} = \underset{\beta}{argmax}(l^{cor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) - \lambda \sum_{j=1}^{p} |\beta_j|) \tag{2.17}$$

Äquivalent dazu kann man auch die negative, korrigierte Log-Likelihood-Funktion minimieren mit dem dazugehörigen Strafterm:

$$\boldsymbol{\beta}^{\boldsymbol{cor,lasso}} = \underset{\beta}{argmin}(-l^{cor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) + \lambda \sum_{j=1}^{p} |\beta_j|) \tag{2.18}$$

(Park und Hastie; 2007, S. 659).

Bestimmung des Regularisierungsparameter

(Tibshirani; 1996) schlägt in seinem Artikel drei Verfahren vor mit denen sich der LASSO Parameter t bestimmen lässt: Kreuzvalidierung, generalisierte Kreuzvalidierung und eine analytische, unverzerrte Risikoschätzung (Tibshirani; 1996, Kap. 4). In dieser Arbeit soll allerdings λ für GLMs bestimmt werden. Die Art λ zu bestimmen hängt dabei von dem jeweiligen Verfahren ab. (Friedman et al.; 2010) schlagen z.B. vor ein λ_{max} zu wählen, so dass $\boldsymbol{\beta} = \mathbf{0}$ ist, mit $N\alpha\lambda_{max} = max_l|\langle x_l, y\rangle|$. Dies ist so gewählt, da in ihrem Algorithmus $\boldsymbol{\beta}_j$ gleich 0 ist und bleibt, wenn $\frac{1}{N}|\langle x_j, y\rangle| < \lambda\alpha$. Ein λ_{min} wird bestimmt als $\lambda_{min} = \epsilon_{lambda}\lambda_{max}$ und dann werden U Werte von λ erstellt , die von λ_{max} bis λ_{min} abnehmen auf der Log Skala. Typische Werte sind $\epsilon_{lambda} = 0,001$ und $U = 100$. (Friedman et al.; 2010, Kap. 2.5)

2.3. Konvexe Optimierungsverfahren für GLMs

Um das Minimum von Gleichung 2.18 zu finden benötigen wir ein spezielles Verfahren, da die Ableitung für den Strafterm nicht direkt gegeben ist. In einem ersten Anlauf wurde versucht die Nullstelle mit einem pfadweisen Algorithmus, der den Koordinatenabstieg benutzt, zu finden. Der Algorithmus beruht auf dem allgemeinen Algorithmus von (Friedman et al.; 2010). In einem kritischen Schritt, der später genauer erklärt wird, ist eine Kennzahl nicht berechenbar. Die Methodik für das Verfahren soll bis zu diesem Schritt trotzdem erläutert werden.

2.3.1. Koordinatenabstieg

In R ist die Funktion 'glmnet'(`https://cran.r-project.org/web/packages/glmnet/glmnet.pdf`) vorhanden, die den Algorithmus mit dem Koordinatenabstieg durchführt, allerdings kann in dieser Funktion keine Formel eingegeben werden und somit kann die korrigierte Log-Likelihood-Funktion nicht direkt eingesetzt werden (R Core Team; 2018).

Für den Algorithmus des Koordinatenabstiegs möchten wir in jedem Schritt ein penalisiertes gewichtetes kleinste Quadrate Problem lösen. Dazu approximieren wir die korrigierte Log-Likelihood-Funktion durch eine Taylor-Reihe zweiter Ordnung an der Stelle $\boldsymbol{X\beta}$ und erhalten die quadratische Approximation

$$l_Q^{cor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) =$$

$$l^{cor}(\tilde{\boldsymbol{\beta}}, \boldsymbol{X}, \boldsymbol{Y}) + (\boldsymbol{X\beta} - \boldsymbol{X}\tilde{\boldsymbol{\beta}})^T \boldsymbol{\nabla} l^{cor}(\tilde{\boldsymbol{\beta}}, \boldsymbol{X}, \boldsymbol{Y}) + \frac{(\boldsymbol{X\beta} - \boldsymbol{X}\tilde{\boldsymbol{\beta}})^T \boldsymbol{H}_{l^{cor}(\tilde{\boldsymbol{\beta}}, \boldsymbol{X}, \boldsymbol{Y})}(\boldsymbol{X}\tilde{\boldsymbol{\beta}})(\boldsymbol{X\beta} - \boldsymbol{X}\tilde{\boldsymbol{\beta}})}{\mathbf{2}}.$$

Nach einigen Umformungen und $\tilde{\boldsymbol{\eta}} = \boldsymbol{X}\tilde{\boldsymbol{\beta}}$ erhält man

$$\frac{\mathbf{1}}{\mathbf{2}}(\boldsymbol{v}(\tilde{\boldsymbol{\eta}}) - \boldsymbol{X\beta})^T \boldsymbol{H}_{l^{cor}(\tilde{\boldsymbol{\beta}}, \boldsymbol{X}, \boldsymbol{Y})}(\boldsymbol{X}\tilde{\boldsymbol{\beta}})(\boldsymbol{v}(\tilde{\boldsymbol{\eta}}) - \boldsymbol{X\beta}) + \boldsymbol{C}(\tilde{\boldsymbol{\eta}}, \tilde{\boldsymbol{\beta}})$$

wobei

$$\boldsymbol{v}(\tilde{\boldsymbol{\eta}}) = \tilde{\boldsymbol{\eta}} - \boldsymbol{H}_{l^{cor}(\tilde{\boldsymbol{\beta}}, \boldsymbol{X}, \boldsymbol{Y})}(\boldsymbol{X}\tilde{\boldsymbol{\beta}})^{-1} \boldsymbol{\nabla} l^{cor}(\tilde{\boldsymbol{\beta}}, \boldsymbol{X}, \boldsymbol{Y})$$

und $\boldsymbol{C}(\tilde{\boldsymbol{\eta}}, \tilde{\boldsymbol{\beta}})$ hängt nicht von $\boldsymbol{\beta}$ ab. (Friedman et al.; 2010, S. 3-4)

Um die Berechnungen zu beschleunigen wird die Hesse-Matrix durch eine Diagonalmatrix mit den diagonalen Einträgen von der Hesse-Matrix ersetzt. Die diagonalen Einträge der Hesse-matrix werden im Folgenden mit $\boldsymbol{w}(\tilde{\boldsymbol{\eta}})_i$ notiert. Das funktioniert, da das optimale $\boldsymbol{\beta}$ ein fixer Punkt in dem Algorithmus bleibt und die Einträge außerhalb der Diagonalen verhältnismäßig klein sind (Hastie und Tibshirani; 1990, Kap. 8). Um jetzt nicht nach einem Maximum zu suchen, sondern nach einem Minimum, arbeiten wir mit der negativen quadratischen Approximation der korrigierten Log-Likelihood-Funktion und die zu minimierende Gleichung ergibt sich als

$$\boldsymbol{\beta}^{\boldsymbol{cor,lasso}} = \underset{\beta}{argmin}(-l_Q^{cor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) + \lambda \sum_j^p |\beta|). \tag{2.19}$$

D.h. jetzt soll ein penalisiertes Kleinste Quadrate Problem mit dem Koordinatenabstieg gelöst werden. (Friedman et al.; 2010, S. 8)

Algorithmus

Daraus ergibt sich der Algorithmus als:

1. $\tilde{\boldsymbol{\beta}}$ initialisieren und $\tilde{\boldsymbol{\eta}} = \boldsymbol{X}\tilde{\boldsymbol{\beta}}$ setzen.
2. Hesse-Matrix an der Stelle $\tilde{\boldsymbol{\eta}}$ und $\boldsymbol{v}(\tilde{\boldsymbol{\eta}})$ berechnen.
3. $\hat{\boldsymbol{\beta}}$ finden, so dass

$$\frac{1}{2n} \sum_{i=1}^{n} \boldsymbol{w}(\tilde{\boldsymbol{\eta}})_i (\boldsymbol{v}(\tilde{\boldsymbol{\eta}})_i - \boldsymbol{x_i.\beta})^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{2.20}$$

minimiert wird. (In dem Original Artikel nicht mit dem LASSO-Strafterm, sondern mit elastischen Netz Strafterm $\lambda P_\alpha(\beta)$) (Friedman et al.; 2010, S. 4)

4. Neue Werte einsetzen mit $\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}$ und $\tilde{\boldsymbol{\eta}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}$

5. Schritte 2-4 wiederholen bis zur Konvergenz von $\hat{\boldsymbol{\beta}}$.

Explizite Form

(Friedman et al.; 2010, S. 4) Wir möchten nun Gleichung 2.20, welche im Folgenden als $M(\beta)$ bezeichnet wird, minimieren und benutzen dafür den Koordinatenabstieg. Angenommen wir haben Schätzungen für $\tilde{\boldsymbol{\beta}}_l$, $l \neq k$ und möchten in Bezug auf $\boldsymbol{\beta}_k$ optimieren. Daraus bilden wir die Ableitung

$$\frac{\partial M(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_k} = -\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{w}(\tilde{\boldsymbol{\eta}})_i * x_{i,k}(\boldsymbol{v}(\tilde{\boldsymbol{\eta}})_i - \boldsymbol{x}_i.\boldsymbol{\beta}) + \lambda * sign(\beta_k), \tag{2.21}$$

wobei $\beta_k \neq 0$ und $sign(\beta_k)$ die Vorzeichenfunktion ist. $sign(\beta_k)$ ist 1 für $\beta_k > 0$, -1 für $\beta_k < 0$ und 0 für $\beta_k = 0$ (Friedman et al.; 2010, S. 4). Die Herleitung zu 2.21 findet sich in Friedman et al. (2007). Von 2.21 kommen wir nach einigen Rechnungen zu der Koordinatenlösung mit

$$\hat{\boldsymbol{\beta}}_k = \frac{S(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{w}(\tilde{\boldsymbol{\eta}})_i * x_{i,k}(\boldsymbol{v}(\tilde{\boldsymbol{\eta}})_i - \sum_{j\neq k} x_{i,j}\beta_j)), \lambda)}{\frac{1}{n}\sum_{i=1}^{p} \boldsymbol{w}(\tilde{\boldsymbol{\eta}})_i x_{ik}^2}. \tag{2.22}$$

Wobei

$$S(x,a) = sign(x)(|x| - a)_+ = \begin{cases} x - a & , x > 0 \quad \& \quad |x| > a \\ x + a & , x < 0 \quad \& \quad |x| > a \\ 0 & , a \geq |x| \end{cases}$$

die Softe-Schwellenwert-Funktion ist (Donoho und Johnstone; 1994). Und

$$\sum_{j\neq k} x_{i,j}\beta_j \tag{2.23}$$

ist der gefittete Wert ohne die Beiträge der x_{ik}.

Problematik des Koordinatenabstieg

Wie aus Gleichung 2.22 sichtbar wird, werden sowohl der Gradient, als auch die Hesse-Matrix an der Stelle $\tilde{\boldsymbol{\eta}} = \boldsymbol{X}\tilde{\boldsymbol{\beta}}$ benötigt. Allerdings enthält die korrigierte Score-Funktion der Poisson-Verteilung den Term $\xi = \frac{\boldsymbol{\beta^T \Lambda \beta}}{2}$, womit die Ableitung nach $\tilde{\boldsymbol{\eta}}$ nicht möglich ist. Folglich lassen sich auch die Werte $\boldsymbol{v}(\tilde{\boldsymbol{\eta}})_i$ nicht berechnen. Eine quadratische Appro-

ximation an der Stelle $\boldsymbol{\beta}$ wäre theoretisch möglich, allerdings ist die Hesse-Matrix dann $p \times p$ und nicht wie benötigt $i \times i$, womit sich der Algorithmus ebenfalls nicht ausführen lässt. Der naive Schätzer wäre nach $\tilde{\boldsymbol{\eta}}$ ableitbar, würde aber vom Thema wegführen, da dann die Korrektur für Messfehler fehlt.

2.3.2. Prädiktor-Verbesserer Methode

In einem zweiten Anlauf wurde versucht das Optimierungsproblem mit der Prädiktor-Verbesserer Methode zu lösen. Der Algorithmus wurde von (Park und Hastie; 2007) vorgeschlagen. Die Funktion ‚glmpath' (`https://cran.r-project.org/web/packages/glmpath/glmpath.pdf`) in R führt den Algorithmus aus, allerdings kann auch hier wieder keine Formel eingegeben werden (R Core Team; 2018).

Zu Beginn versuchen wir wieder die negative, korrigierte Log-Likelihood-Funktion

$$\boldsymbol{\beta}^{\boldsymbol{cor,lasso}}(\lambda) = \underset{\beta}{argmin}(-l^{cor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) + \lambda \sum_{j=1}^{p} |\beta_j|)$$

zu minimieren und somit die korrigierte und regularisierte Schätzung zu bekommen.

Die Methode bestimmt den ganzen Pfad der Koeffizientenschätzungen während λ variiert, um $\{\boldsymbol{\beta}^{\boldsymbol{cor,lasso}}(\lambda) : 0 < \lambda < \infty\}$ zu finden. Startwert für das λ ist $\lambda = \lambda_{max}$, wobei λ_{max} das größte λ ist, für welches $\boldsymbol{\beta}^{\boldsymbol{cor,lasso}}(\lambda)$ nicht 0 ist. Von dort aus wird eine Reihe von Lösungen berechnet, wobei jedes mal die Koeffizienten mit einem kleineren λ geschätzt werden, welches auf der vorherigen Schätzung basiert. (Park und Hastie; 2007, S. 660)

Folglich sind die Schrittlängen für keinen Parameter einheitlich, sondern sind von den Daten abhängig. Das macht die Methode flexibler und effizienter als z.B. den von (Rosset; 2004) vorgeschlagenen Algorithmus, der auf jede Verlust- und Strafterm Funktion, mit angemessenen Grenzen für die Definitionsbereiche und Ableitungen, angewendet werden kann. Der Algorithmus verändert λ und schätzt die neuen Koeffizienten durch eine Newton Iteration. Die λ haben also einen einheitlichen Abstand voneinander. Ein Beispiel für einen Algorithmus in dem die $||\beta||_1$ einen einheitlichen Abstand haben, kommt von (Zhao; 2004), die das „Boosted Lasso" vorschlagen, welches eine Variation der Koordinatenabstieg Methode verwendet. Der Algorithmus besteht aus einem Vorwärtsschritt und einem Rückwärtsschritt. Der Vorwärtsschritt ist ähnlich zum steigern und zur stufenweise vorwärts Anpassung. Der Rückwärtsschritt sorgt dafür, dass der Steigerungspfad sich dem LASSO-Pfad anähert. (Zhao; 2004)

In der Prädiktor-Verbesserer Methode wird das größte λ geschätzt, für welches sich das momentane aktive Set verändert und dann wird an diesem λ das neue Set berechnet. Jede Runde der Optimierung besteht aus drei Schritten : Schrittlänge des λ bestimmen, die dazugehörige Veränderung in den Koeffizienten vorhersagen und den Fehler in der vorherigen Vorhersage korrigieren. (Park und Hastie; 2007, S. 660)

Problemstellung

Wenn wir von der Darstellung einer Dichtefunktion als Exponentialfamilie

$$f_Y(y;\theta,\phi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\}$$

ausgehen und nach der Lösung für den natürlichen Parameter θ, und somit $\boldsymbol{\beta}$ suchen, dann fallen die Funktionen $a(.)$ und $c(.)$ raus. Das liegt daran, dass diese nicht den gewünschten Parameter enthalten und in den kommenden Schritten eine Ableitung nach $\boldsymbol{\beta}$ verwendet werden soll. Für ein GLM ergibt sich somit das Minimierungsobjekt mit L1-Penalisierung allgemein für GLMs als

$$l^{cor}_{predictor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = -\sum_{i=1}^{n}(y_i\theta(\boldsymbol{\beta})_i - b(\theta(\boldsymbol{\beta})_i)) + \lambda||\beta||_1, \tag{2.24}$$

wobei hier der Betavektor noch den Intercept β_0 enthält. (Park und Hastie; 2007, Kap. 2) Zusammen mit der korrigierten Log-Likelihood Funktion für die Poisson-Verteilung erhält man

$$l^{cor}_{predictor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{n}(\exp(\boldsymbol{x_i}.\boldsymbol{\beta} - \xi) - y_i\boldsymbol{x}_i.\boldsymbol{\beta}) + \lambda||\beta||_1$$

bzw. nach Umstellung

$$l^{cor}_{predictor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{n}(\underbrace{-y_i\boldsymbol{x}_i.\boldsymbol{\beta}}_{=y\theta} + \underbrace{\exp(\boldsymbol{x_i}.\boldsymbol{\beta} - \xi)}_{=b(\theta)}) + \lambda||\beta||_1 \tag{2.25}$$

Unter der Annahme, dass keine Komponente von $\boldsymbol{\beta}$ gleich 0 ist und abgeleitet in Bezug auf $\boldsymbol{\beta}$ definieren (Park und Hastie; 2007) eine allgemeine Funktion

$$H(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \frac{\partial l^{cor}_{predictor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})}{\partial \boldsymbol{\beta}} = -\boldsymbol{X^T W}(y-\mu)\frac{\partial \eta}{\partial \mu} + \lambda sign\begin{pmatrix} 0 \\ \boldsymbol{\beta} \end{pmatrix},$$

wobei $\boldsymbol{X}$ eine $n \times (p+1)$ Matrix inklusive einer Spalte von 1s ist, $\boldsymbol{W}$ ist eine Diagonal Matrix mit n Diagonalelementen $V^{-1}\frac{\partial \mu}{\partial \eta_i}^2$ und $(y-\mu)\frac{\partial \eta}{\partial \mu}$ ist ein Vektor mit n Elementen $(y_i - \mu_i)\frac{\partial \eta}{\partial \mu_i}$. (Park und Hastie; 2007, S. 661) Hier ist $V = Var(Y)$ die Varianz.

Problematik Prädiktor-Verbesserer Methode

Für den Erwartungswert und die Varianz in einem GLM mit gilt wie bereits erwähnt

$$E(y) = \mu = b'(\theta) \qquad Var(y) = b''(\theta)a(\phi),$$

mit $b'(\theta)$ als erste und $b''(\theta)$ als zweite Ableitung nach θ. (McCullagh und Nelder; 1989, S. 29)(Fahrmeir et al.; 2009, S. 218) Für die normale Poissonregression gilt:

$$E(y) = \exp(\theta) = b'(\theta) = b''(\theta) = Var(y) = \lambda.$$

(Fahrmeir et al.; 2009, S. 220). Allerdings ist im Falle der korrigierten Score-Funktion $b(\theta)$ nicht mehr $\exp(\theta)$, sondern $\exp(\theta - \xi)$ und für das „$b(\theta)$“ der korrigierten Log-Likelihood-Funktion lässt sich die Ableitung nach θ nicht bestimmen. Daraus folgt, dass sich für die Prädiktor-Verbesserer Methode $V^{-1}(\frac{\partial \mu}{\partial \eta})_i^2$ und $(y_- \mu_i)(\frac{\partial \eta}{\partial \mu})_i$ nicht bestimmen lassen. Für $V^{-1}(\frac{\partial \mu}{\partial \eta})_i^2$ muss $\mu = \frac{\partial \exp(\theta - \xi)}{\partial \theta} = \frac{\partial \exp(\boldsymbol{x}_i.\boldsymbol{\beta} - \frac{\boldsymbol{\beta^T \Lambda \beta}}{2})}{\partial \boldsymbol{x}_i.\boldsymbol{\beta}}$ gebildet werden, d.h. es liegt dieselbe Problematik vor wie beim Koordinatenabstieg und die Methode kann nicht genutzt werden.

2.3.3. Kleinste Winkel Regression

Ein weiteres bekanntes Verfahren um das LASSO Problem für GLMs zu lösen zu lösen ist eine Modifizierung der „Least Angle Regression“ (LARS) von (Efron et al.; 2004). Der Algorithmus ist für ein lineares Modell gedacht, dafür schlagen (Madigan und Ridgeway; 2004) eine Erweiterung für GLMS vor.

Problematik LARS

In der Erweiterung auf GLMs mit nichtlinearen Verlustfunktionen wird zuerst eine Form der Approximation benötigt. Im zweiten Schritt wird für ein bestimmtes α die Kovariable $\boldsymbol{x}_i.$ gesucht, welche die größte Verbesserung in $l(f(\boldsymbol{x}) + \boldsymbol{x}_i^T \alpha)$ bringt. $f(\boldsymbol{x})$ ist dabei eine

Funktion die von $\boldsymbol{x_{i\cdot}}$ abhängt. Um diese zu finden wird das Maximum gesucht mit

$$i^* = \underset{\text{i}}{argmax} \left| \frac{\partial}{\partial \alpha} l(f(\boldsymbol{x}) + \boldsymbol{x_{i\cdot}}\alpha) \right|_{\alpha=0}.$$

(Madigan und Ridgeway; 2004, S. 467) Hier stellt sich das Problem, dass die Korrektur ξ rausfallen würde, da diese nicht von den $\boldsymbol{x_{i\cdot}}$ abhängt. Somit wären die LASSO Schätzungen nicht für Messfehler korrigiert und die Methode fällt ebenfalls raus.

2.3.4. Innere-Punkte Methode

(Koh et al.; 2007) schlagen eine Innere-Punkte Methode vor für L1-regularisierte logistische Regression. Der Algorithmus lässt sich auch auf ein konvexes Minimierungsproblem mit L1-Penalisierung verallgemeinern. Dabei wird eine Verlustfunktion (negative Log-Likelihood Funktion)(Koh et al.; 2007, S. 1520) benötigt, die zweimal differenzierbar ist und dann ergibt sich das Kriterium als

$$min \frac{1}{n} \sum_{i=1}^{n} l_{inner}(z_i) + \lambda ||w||_1$$

unter der Bedingung

$$z_i = w^T \alpha_i + v b_i + c_i, \qquad i \in \{1, ..., n\}$$

Wobei l_{inner} die Verlustfunktion ist, w entspricht hier dem Betavektor ohne β_0 und v ist der Intercept. $c_i \in \boldsymbol{R}$, $a_i \in \boldsymbol{R}^n$ und $b_i \in \boldsymbol{R}$ sind Daten (Koh et al.; 2007, Kap. 8). Allerdings gilt jetzt für den „linearen Prädiktor“ im korrigierten Fall

$$\boldsymbol{\eta^{cor}} = \boldsymbol{x_{i\cdot}}\boldsymbol{\beta} - \frac{\boldsymbol{\beta^T \Lambda \beta}}{2}$$

und mit der Annahme, dass die Designmatrix als erste Zeile nur 1en enthält erhalten wir:

$$\boldsymbol{\eta^{cor}} = \beta_0 + \boldsymbol{x}_{i,(0)}\boldsymbol{\beta}_{(0)} - \frac{\beta_0 \sigma \beta_0}{2} - \frac{\boldsymbol{\beta_{(0)} \Lambda_{[,-1]} \beta_{(0)}}}{2}$$

Wobei das $\Lambda_{[,-1]}$ für die Varianz matrix ohne die erste Spalte steht und $\beta_{(0)}$ für den Betavektor ohne 0. Man sieht, dass der „korrigierte“ Intercept jetzt nicht in der Form vorliegt wie in z_i benötigt und die Methode kann nicht angewendet werden.

Nelder-Mead-Verfahren

Im fünften Anlauf wurde überlegt das Minimierungskriterium mit dem Nelder-Mead-Verfahren, auch bekannt als Downhill-Simplex Verfahren, zu lösen. (Nelder und Mead; 1965). Eine Anwendung auf einen simulierten Datensatz führte zu unbrauchbaren Ergebnissen. Die Idee wurde auf Grund von mangelnder Literatur nicht weiter verfolgt und die Ursache für die unbrauchbaren Ergebnisse konnte nicht ermittelt werden.

2.3.5. SCAD

Im sechsten Anlauf wurde versucht das Minimierungsproblem mit einem von (Fan und Li; 2001) vorgeschlagenen Algorithmus zu lösen. Der Algorithmus verwendet einen „glatt gestutzten, absolute Abweichung“ Strafterm („Smoothly Clipped Absolute Deviation Penalty“-SCAD). Der Strafterm is gegeben mit

$$pen'_\lambda(\theta) = \left\{ \mathbb{1}(\theta \leq \lambda) + \frac{(a\lambda - \theta)_+}{(a-1)\lambda} \mathbb{1}(\theta > \lambda) \right\} \tag{2.26}$$

mit

$$a > 2, \qquad \theta > 0$$

und $\mathbb{1}$ als Indikatorfunktion. (Fan und Li; 2001) behaupten, dass ein guter Strafterm drei Eigenschaften erfüllen sollte:

1. Unverzerrtheit: Der resultierende Schätzer ist beinahe unverzerrt, um unnötige Modellierungsverzerrung zu vermeiden.

2. Spärlichkeit: Der resultierende Schätzer ist eine Schwellenwertregel, die automatisch klein geschätzte Koeffizienten auf 0 setzt, um Modellkomplexität zu reduzieren.

3. Stetigkeit: Der resultierende Schätzer ist stetig in den Daten, um Instabilität in der Modellvorhersage zu vermeiden.

(Fan und Li; 2001, S. 1349) Zudem behaupten sie, dass die Strafterme der Ridge Regression, des LASSO und der Harten Schwellenwertfunktion diese drei Eigenschaften nicht gleichzeitig erfüllen. Das SCAD verbessere zumindest die Eigenschaften der Strafterme des LASSO und der Harten Schwellenwertfunktion. (Fan und Li; 2001, S. 1350-1351) Der SCAD Strafterm ist eine Modifizierung des LASSO Srafterms, welche größere Koeffizienten weniger streng schrumpft. Der zweite Term reduziert den Anteil der Schrumpfung

im LASSO für größere Werte von β, wobei keine Schrumpfung auftritt für $a \to \infty$. (Hastie et al.; 2009, S. 92)

Die Ausgangsstellung ist ähnlich wie in den anderen Verfahren. Wir möchten die negative Log-Likelihood-Funktion mit einem Strafterm minimieren (Fan und Li; 2001, Kap. 3.1).

$$\boldsymbol{\beta}_{pen}^{cor}(\lambda) = \underset{\beta}{argmin}(-l^{cor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) + n\sum_{j=1}^{p} pen_\lambda|\beta_j|)$$

Wir benötigen auch wieder eine quadratische Approximation, allerdings diesmal nicht an der Stelle $\boldsymbol{\eta}$, sondern an der Stelle $\boldsymbol{\beta}$. Die negative Log-Likelihood-Funktion wird im Folgenden als $l_L^{cor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})$ bezeichnet. Gegeben einem Startwert $\boldsymbol{\beta}_{Start}$, der nahe am Minimierer liegt, kann der SCAD Strafterm lokal approximiert werden mit einer quadratischen Funktion als

$$[pen_\lambda(|\beta_j|)]' = pen'_\lambda(|\beta_j|)sign(\beta_j) \approx \left\{\frac{pen'_\lambda(|\beta_{j0}|)}{|\beta_{j0}|}\right\}\beta_j,$$

für $\beta_j \neq 0$. Falls ein $\beta_{j,0}$ nahe an 0 liegt, dann setze $\hat{\beta}_j = 0$. Und weiterhin :

$$pen_\lambda(|\beta_j|) \approx pen_\lambda(|\beta_{j0}|) + \frac{1}{2}\left\{\frac{pen'_\lambda(|\beta_{j0}|)}{|\beta_{j0}|}\right\}(\beta_j^2 - \beta_{j,o}^2),$$

für $\beta_j \approx \beta_{j0}$. (Fan und Li; 2001, Kap. 3.3) D.h. das Minimierungsproblem reduziert sich auf ein quadratisches Minimierungsproblem und das Newton-Raphson Verfahren kann angewendet werden. Die quadratische Approximation des Kriteriums mit der korrigierten Log-Likelihood Funktion ergibt sich als

$$l_L^{cor}(\boldsymbol{\beta_0}, \boldsymbol{X}, \boldsymbol{Y}) + \nabla l_L^{cor}(\boldsymbol{\beta_0}, \boldsymbol{X}, \boldsymbol{Y})^T(\boldsymbol{\beta} - \boldsymbol{\beta_0}) + \frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta_0})^T$$
$$\nabla^2 l_L^{cor}(\boldsymbol{\beta_0}, \boldsymbol{X}, \boldsymbol{Y})(\boldsymbol{\beta} - \boldsymbol{\beta_0}) + \frac{1}{2}n\boldsymbol{\beta}^{\boldsymbol{T}}\Sigma_\lambda(\boldsymbol{\beta_0})\boldsymbol{\beta}.$$

Wobei

$$\nabla l_L^{cor}(\boldsymbol{\beta_0}, \boldsymbol{X}, \boldsymbol{Y}) = \frac{\partial l_L^{cor}(\boldsymbol{\beta_0}, \boldsymbol{X}, \boldsymbol{Y})}{\partial \boldsymbol{\beta}}, \tag{2.27}$$

$$\nabla^2 l_L^{cor}(\boldsymbol{\beta_0}, \boldsymbol{X}, \boldsymbol{Y}) = \frac{\partial^2 l_L^{cor}(\boldsymbol{\beta_0}, \boldsymbol{X}, \boldsymbol{Y})}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}^T}, \tag{2.28}$$

$$\Sigma_\lambda(\boldsymbol{\beta}_0) = diag\left\{\frac{pen'_\lambda(|\beta_{10}|)}{|\beta_{10}|}, \ldots, \frac{pen'_\lambda(|\beta_{p0}|)}{|\beta_{p0}|}\right\}. \tag{2.29}$$

(Fan und Li; 2001, S. 1354) Die Lösung für die quadratische Approximation ergibt sich

als

$$\boldsymbol{\beta}_1 =$$

$$\boldsymbol{\beta}_0 - \left\{\nabla^2 l_L^{cor}(\boldsymbol{\beta}_0, \boldsymbol{X}, \boldsymbol{Y}) + n\boldsymbol{\Sigma}_\lambda(\boldsymbol{\beta}_0)\right\}^{-1} \left\{\nabla l_L^{cor}(\boldsymbol{\beta}_0, \boldsymbol{X}, \boldsymbol{Y}) + n\boldsymbol{U}_\lambda(\boldsymbol{\beta}_0)\right\} \tag{2.30}$$

mit

$$\boldsymbol{U}_\lambda(\boldsymbol{\beta}_0) = \boldsymbol{\Sigma}_\lambda(\boldsymbol{\beta}_0)\boldsymbol{\beta}_0.$$

Wenn der Algorithmus konvergiert, erfüllt er die Bedingung

$$\frac{\partial l_L^{cor}(\hat{\boldsymbol{\beta}}_{\mathbf{0}}, \boldsymbol{X}, \boldsymbol{Y})}{\partial \beta_j} + n * pen'_\lambda(|\hat{\beta}_{j,0}|) * sign(\hat{\beta}_{j,0}) = 0$$

(Fan und Li; 2001, S. 1354). Wenn ein β_{j0} sehr nahe bei 0 liegt, dann soll $\hat{\beta}_j = 0$ gesetzt werden (Fan und Li; 2001, S. 1354)

Wahl der Regularisierungsparameter

(Fan und Li; 2001) schlagen zwei Methoden vor, um die Regularisierungsparameter λ und a zu bestimmen: fünffache Kreuzvalidierung und generalisierte Kreuzvalidierung. In Simulationen fanden sie heraus, dass $a = 3,7$ ein vernünftiger Wert ist, weswegen dieser Wert in der Simulation hergenommen wird. (Fan und Li; 2001, S. 1356) In dieser Arbeit soll kein optimales λ bestimmt werden, sondern es sollen die Ergebnisse für verschiedene Werte von λ dargestellt werden.

Vergleich zu anderen Verfahren

(Fan und Li; 2001) verglichen die Teilmengenselektion, die Ridge Regression, das LASSO und das SCAD untereinander. Das LASSO performte am Besten bei großem Rauschen und einer kleinen Stichprobengröße. Wenn das Rauschen geringer wird, performt das SCAD besser als das LASSO. Ridge Regression performte sehr schlecht. (Fan und Li; 2001, S. 1356)

3. Simulationen

Der Algorithmus mit dem SCAD Strafterm soll nun auf jeweils die naive und korrigierte Log-Likelihood-Funktion angewendet und dann gegenübergestellt werden. Für die Erzeugung der Daten und die Berechnung der einzelnen Schritte wurde das Programm R benutzt (R Core Team; 2018). Der Code wurde zusammen mit der Arbeit abgegeben. Die Ergebnisse sind reproduzierbar durch die set.seed Funktion.

Datenerzeugung Poisson-Verteilung

Für die Erstellung der Daten mit Poisson-Verteilung orientieren wir uns an (Nakamura; 1990). Wir erzeugen die $n \times p$ Matrix $\boldsymbol{Z}$. Eine Zeile $\boldsymbol{z_i.}$ von der Matrix $\boldsymbol{Z}$ gibt die Werte für ein Individuum an. Die Werte von $\boldsymbol{Z}$ werden aus der Gleichverteilung gezogen, so dass die Werte im Intervall $(0, 1)$ liegen und voneinander unabhängig sind.

Die $n \times p$ Matrix $\boldsymbol{X}$ erzeugen wir so, dass $\boldsymbol{x_i.} = \boldsymbol{z_i.} + \boldsymbol{\epsilon_i.}$, wobei $\boldsymbol{\epsilon_i.}$ eine normalverteilte Zufallsvariable mit Mittelwert $\mathbf{0}$ und Varianzmatrix $\boldsymbol{\Lambda}$ ist. Die Diagonaleinträge von $\boldsymbol{\Lambda}$ entsprechen der Varianz σ^2. Die Werte außerhalb der Diagonalen sind 0. Die Standardabweichung σ der Fehlerterme ist entweder 0, 0,1, 0,2 oder 0,3. Daraus folgt, dass alle Messfehler gleich 0 sind, wenn σ gleich 0 ist. Die Varianz ist die quadrierte Standardabweichung.

Wir erzeugen den Vektor $\boldsymbol{Y}$ der Länge n, so dass die y_i einer Poisson-Verteilung folgen mit Mittelwert $\exp(\boldsymbol{z_i.\beta})$. $\boldsymbol{\beta}_t$ ist ein Vektor der Länge p mit den wahren Parameterwerten $\boldsymbol{\beta_t} = (0, 0, 0, 0, 0, 0.5, 1, 2, 2, 2)^T$. Die Werte sind so gewählt, dass die Hälfte Nullen sind und die andere Hälfte aus einem Wert nahe 0, einem mittelgroßen Wert und 3 großen Werten besteht.

Der Parameter a aus dem SCAD Algorithmus ist 3.7 und λ soll bei 0.1 starten und in jedem Schritt um 0,1 erhöht werden. Für jede Auswertung wird ein Algorithmus 100 mal wiederholt (Anzahl Wiederholungen).

Kenngrößen für Poisson-Verteilung

Die negative korrigierte Log-Likelihood-Funktion für die Poisson-Verteilung ergibt sich als

$$l_L^{cor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{n}(\exp(\boldsymbol{x_i}.\boldsymbol{\beta} - \xi) - y_i \boldsymbol{x}_i.\boldsymbol{\beta} + \log(y_i)).$$

Die erste Ableitung nach $\boldsymbol{\beta}$ der negativen korrigierten Log-Likelihood-Funktion mit den Startwerten $\boldsymbol{\beta}_0$ ist der Gradient

$$\nabla l_L^{cor}(\boldsymbol{\beta}_0, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{n}(-y_i.\boldsymbol{x_i}. + (\boldsymbol{x}_i. - \boldsymbol{\Lambda\beta_0}) * \exp(\boldsymbol{x}_i.\boldsymbol{\beta_0} - \xi)).$$

Die zweite Ableitung der negativen korrigierten Log-Likelihood-Funktion nach $\boldsymbol{\beta}$ ergibt sich als

$$\nabla^2 l_L^{cor}(\boldsymbol{\beta}_0, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{n}((\boldsymbol{x}_i. - \boldsymbol{\Lambda\beta_0})(\boldsymbol{x_i}. - \boldsymbol{\Lambda\beta_0})^{\boldsymbol{T}} - \boldsymbol{\Lambda}) * \exp(\boldsymbol{x}_i.\boldsymbol{\beta_0} - \xi),$$

was der korrigierten Information entspricht.

Auswertung

Da die Simulationen von (Fan und Li; 2001) nach 30 Iterationen konvergierten, orientieren wir uns daran und nehmen 30 als Richtwert für die Anzahl an Newton Iterationen. Nach 30 Iterationen sollte ersichtlich sein, ob der Algorithmus konvergiert. Geschätzte Werte kleiner 0,01 werden auf 0 gesetzt. Dieser Schwellenwert ist willkürlich gewählt. Der Vergleich zwischen dem SCAD Algorithmus mit der naiven und der korrigierten Log-Likelihood-Funktion soll anhand von zwei Kriterien geschehen:

1. **Abweichung vom wahren Wert**: ähnlich wie bei Nakamura soll der Durchschnittswert der n geschätzten Werte nach 30 Iterationen mit dem wahren Wert $\boldsymbol{\beta}_t$ in Relation gesetzt werden. Nakamura stellte die Relation zum geschätzten Wert ohne Messfehler her (Nakamura; 1990, Kap. 5). Das ist hier auf Grund von Problemen mit der Invertierbarkeit für kleine λ Werte nicht möglich, darum wird der wahre Wert $\boldsymbol{\beta}_t$ als Vergleich genommen. Es werden nur die geschätzten Werte für $\boldsymbol{\beta}_t$ ungleich 0 verglichen. Hier werden die letzten fünf Einheiten aus dem wahren Betavektor betrachtet, da die größer als 0 sind. Hier soll überprüft werden, ob der

SCAD Algorithmus mit der korrigierten Log-Likelihood-Funktion die Koeffizienten präziser schätzt als der SCAD Algorithmus mit der naiven Log-Likelihood-Funktion. Für den Vergleich wird der Durchschnittswert aus den n Simulationen gebildet und durch den wahren Wert geteilt. Die Entfernung zum Wert 1 wird im Folgenden als (durchschnittliche) Abweichung bezeichnet.

2. **Ø Anzahl richtig und falsch geschätzter Nullen**: ähnlich zu (Fan und Li; 2001) soll auch die durchschnittliche Anzahl korrekt und falsch geschätzter Nullen abgebildet werden. Damit soll vor allem überprüft werden, ob der SCAD Algorithmus für die korrigierte Log-Likelihood-Funktion, unter Messfehlern, die korrekte Anzahl an Nullen eher richtig schätzt, als der SCAD Algorithmus für die naive Log-Likelihood-Funktion. (Fan und Li; 2001)

In den folgenden Auswertungen sollen die Ergebnisse für $\sigma = 0/0, 1/0, 2/0, 3$ dargestellt werden. N ist dabei jeweils 100/200/500. Die wahren Parameterwerte sind $\boldsymbol{\beta_t} = (0, 0, 0, 0, 0, 0.5, 1, 2, 2, 2)^T$.

In einigen Fällen kommt es vor, dass die Matrix singulär ist. Matrix singulär bedeutet, dass die Matrix $(\nabla^2 l^{cor}(\boldsymbol{\beta}_0, \boldsymbol{X}, \boldsymbol{Y}) + n\boldsymbol{\Sigma}_\lambda(\boldsymbol{\beta}_0))^{-1}$ nicht mehr invertierbar ist in R. Dazugehörige Fehlermeldung:

Error in solve.default(Hesse + n * Sigma) : system is computationally singular: reciprocal condition number = 2.45909e-18. Für diese Fälle ist kein Ergebnis vorhanden. Es wird das kleinste λ dargestellt für das eines der beiden Algorithmen zuerst ausführbar ist und das kleinste λ für welches der andere Algorithmus ausführbar ist.

3.1. Sigma = 0

Der Fall ohne Messfehler soll kurz betrachtet werden.

Sigma = 0, n = 100, p = 10, Wiederholungen = 100

Auswertung $\sigma = 0$, $n = 100$, $p = 10$, $Wiederholungen = 100$		
λ	Naive Loglikelihood	Korrigierte Loglikelihood
$0,1 - 2,1$	Matrix singulär	Matrix singulär
$2,2$	Abweichungen:	Abweichungen:
	0.85 0.96 1.00 1.00 1.00	0.85 0.96 1.00 1.00 1.00
	Ø Anzahl Korrekte Nullen = 4	Ø Anzahl Korrekte Nullen = 4
	Ø Falsche Nullen =0	Ø Falsche Nullen =0

Für $\sigma = 0$ lässt sich der SCAD Algorithmus ab einem λ von 2,2 für die naive und korrigierte Log-Likelihood-Funktion berechnen. Ohne Messfehler führen beide Durchläufe logischerweise zum selben Ergebnis, da die Matrix $\mathbf{\Lambda}$ nur aus Nullen besteht. Bei einem λ von 2,2 sind die 2er Werte des Betavektors sehr gut geschätzt und die Werte 0.5 und 1 ein wenig ungenauer mit den Abweichungen 0,85 und 0,96. Die durchschnittlich korrekt geschätzte Anzahl an Nullen ist 4 und die durchschnittliche Anzahl and falschen Nullen ist 0. Ab einem λ von ca. 80 werden alle Koeffizienten auf 0 geschätzt.

Sigma = 0, n = 500, p=10, Wiederholungen=100

Auswertung $\sigma = 0$, $n = 500$, $p = 10$, $Wiederholungen = 100$		
λ	Naive Loglikelihood	Korrigierte Loglikelihood
$0,1 - 2$	Matrix singulär	Matrix singulär
$2,1$	Abweichungen:	Abweichungen:
	0.88 0.96 1.00 1.00 1.00	0.88 0.96 1.00 1.00 1.00
	Ø Anzahl Korrekte Nullen = 5	Ø Anzahl Korrekte Nullen = 5
	Ø Falsche Nullen =0	Ø Falsche Nullen =0

Für $n = 500$ lässt sich der SCAD Algorithmus ab einem λ von 2,1 ausführen. Für ein λ von 2,1 sind die 2er Werte des Betavektors sehr gut geschätzt und die Werte 0,5 und 1 ein wenig schlechter mit den Abweichungen 0,12 und 0,04. Die Anzahl der durchschnittlich korrekt geschätzten Nullen ist 5 und die durchschnittlich falsch geschätzte Anzahl an Nullen ist 0.

Sigma = 0, n = 1000, p=10, Wiederholungen=100

Auswertung $\sigma = 0$, $n = 1000$, $p = 10$, $Wiederholungen = 100$		
λ	Naive Loglikelihood	Korrigierte Loglikelihood
$0,1 - 0,5$	Matrix singulär	Matrix singulär
$0,6$	Abweichungen:	Abweichungen:
	0.97 0.99 1.00 1.00 1.00	0.97 0.99 1.00 1.00 1.00
	Ø Anzahl Korrekte Nullen = 4,8	Ø Anzahl Korrekte Nullen = 4,8
	Ø Falsche Nullen =0	Ø Falsche Nullen =0

Für $\sigma = 0$ und $n = 1000$ lässt sich der Algorithmus ab $\lambda = 0,6$ ausführen. Die Abweichungen sind 0,03, 0,01, 0, 0 und 0. Die durchschnittliche Anzahl richtig geschätzter Nullen ist 4,8 und die durchschnittliche Anzahl falscher Nullen ist 0.

Auswertung Sigma = 0

Eine Erhöhung von n sorgte dafür, dass der SCAD Algorithmus bereits für kleinere λ ausführbar ist. Im Falle von $n = 100$ auf $n = 500$ verbesserte sich die durchschnittliche Anzahl richtig geschätzter Nullen. Im Falle von $n = 1000$ ist die durchschnittliche Anzahl richtig geschätzter Nullen geringer als in den Fällen $n = 100/500$ mit 4,8, allerdings ist hier auch der Strafterm geringer. Es wird ersichtlich, dass die Abweichungen bei den kleineren Werten des Betavektors auffälliger sind, was daran liegen könnte, dass diese näher an Null liegen und deshalb stärker penalisiert werden.

3.2. Sigma = 0,1

Jetzt soll der Fall mit $\sigma = 0,1$ betrachtet werden.

Sigma = 0,1, n = 100, p=10, Wiederholungen=100

Auswertung $\sigma = 0,1$, $n = 100$, $p = 10$, $Wiederholungen = 100$		
λ	Naive Loglikelihood	Korrigierte Loglikelihood
$0,1 - 6,6$	Matrix singulär	Matrix singulär
$6,7$	Abweichungen:	
	0.69 0.89 0.97 0.95 0.94	
	Ø Korrekte Nullen =1,8	
	Ø Falsche Nullen =0	
$9,1$	Abweichungen:	Abweichungen:
	0.55 0.83 0.97 0.95 0.95	0.40 0.78 1.02 1.00 0.99
	Ø Korrekte Nullen =2,1	Ø Korrekte Nullen =3,2
	Ø Falsche Nullen =0	Ø Falsche Nullen =0,01

Für $\sigma = 0,1$ und $n = 100$ ist der SCAD Algorithmus für die naive Log-Likelihood-Funktion ab $\lambda = 6,7$ ausführbar und liefert im Durchschnitt 1,8 richtige Nullen und 0 Falsche. Die Abweichungen sind 0,31, 0,11, 0,03, 0,05 und 0,06. Ab einem λ von 9,1 ist der SCAD Algorithmus auch für die korrigierte Log-Likelihood-Funktion ausführbar und liefert im Durchschnitt 3,2 richtig geschätzte Nullen und 0,01 Falsche. Die Abweichungen sind 0,6, 0,22, -0,02, 0 und 0,01. Im Vergleich dazu liefert der SCAD Algorithmus im Durchschnitt 2,1 richtig geschätzte Nullen und 0 Falsche. Die Abweichungen sind 0,45, 0,17, 0,03, 0,05 und 0,05.

Sigma = 0,1, n = 500, p=10, Wiederholungen=100

Auswertung $\sigma = 0,1$, $n = 500$, $p = 10$, $Wiederholungen = 100$		
λ	Naive Loglikelihood	Korrigierte Loglikelihood
$0,1 - 2,2$	Matrix singulär	Matrix singulär
$2,3$	Abweichungen:	
	0.98 0.96 0.95 0.94 0.95	
	Ø Korrekte Nullen =0,99	
	Ø Falsche Nullen =0	
$4,3$	Abweichungen:	Abweichungen:
	0.88 0.94 0.96 0.96 0.96	0.73 0.91 1.01 1.00 1.00
	Ø Korrekte Nullen =1,6	Ø Korrekte Nullen =4
	Ø Falsche Nullen =0	Ø Falsche Nullen =0

Für $\sigma = 0,1$ und $n = 500$ lässt sich der SCAD Algorithmus für die naive Log-Likelihood-Funktion ab einem λ von 2,3 ausführen und liefert im Durchschnitt 0,99 richtig geschätzte Nullen und 0 Falsche. Die Abweichungen sind 0,02, 0,04, 0,05, 0,06 und 0,05. Ab einem λ von 4,3 lässt sich der SCAD Algorithmus auch für die korrigierte Log-Likelihood-Funktion ausführen und liefert im Durchschnitt 4 richtig geschätzte Nullen und 0 Falsche. Die Abweichungen sind 0,27, 0,09, -0,01, 0 und 0. Im Vergleich dazu liefert der SCAD Algorithmus mit der naiven Log-Likelihood-Funktion im Durchschnitt 1,6 richtig geschätzte Nullen und 0 Falsche. Die Abweichungen sind 0,12, 0,06, 0,04, 0,04 und 0,04.

Sigma = 0,1, n = 1000, p=10, Wiederholungen=100

Auswertung $\sigma = 0,1$, $n = 1000$, $p = 10$, $Wiederholungen = 100$		
λ	Naive Loglikelihood	Korrigierte Loglikelihood
$0,1 - 1,9$	Matrix singulär	Matrix singulär
2	Abweichungen:	
	0.96 0.98 0.95 0.95 0.95	
	Ø Korrekte Nullen =1,1	
	Ø Falsche Nullen =0	
2,9	Abweichungen:	Abweichungen:
	0.93 0.98 0.96 0.96 0.96	0.79 0.96 1.00 1.00 1.00
	Ø Korrekte Nullen =1	Ø Korrekte Nullen =4,1
	Ø Falsche Nullen =0	Ø Falsche Nullen =0

Für $\sigma = 0,1$ und $n = 1000$ ist der SCAD Algorithmus für die naive Log-Likelihood-Funktion ab einem λ von 2 ausführbar und liefert im Durchschnitt 1,1 richtig geschätzte Nullen und 0 Falsche. Die Abweichungen sind 0,04, 0,02, 0,05, 0,05 und 0,05. Ab einem λ von 2,9 lässt sich der SCAD Algorithmus auch für die korrigierte Log-Likelihood-Funktion ausführen und liefert im Durchschnitt 4,1 richtig geschätzte Nullen und 0 Falsche. Die Abweichungen sind 0,21, 0,04, 0, 0 und 0. Im Vergleich dazu liefert der SCAD Algorithmus mit der naiven Log-Likelihood-Funktion im Durchschnitt 1 richtig geschätzte Nullen und 0 Falsche. Die Abweichungen sind 0,07, 0,02, 0,04, 0,04 und 0,04.

Auswertung Sigma = 0,1

Im Falle von $\sigma = 0,1$ ist das Problem mit der Invertierbarkeit größer als im Fall $\sigma = 0$, d.h. es werden größere Werte für λ benötigt damit der Algorithmus ausführbar ist. Dabei ist das Problem mit der Invertierbarkeit leichter zu handhaben für den SCAD Algorithmus mit der naiven Log-Likelihood-Funktion, da sich die Matrix schon für kleinere λ invertieren lässt. Eine Erhöhung von n verbesserte das Problem mit der Invertierbarkeit bei beiden Algorithmen.

Für gleich große λ lieferte der SCAD Algorithmus mit der korrigierten Log-Likelihood-Funktion durchschnittlich mehr richtig geschätzte Nullen und gleich viele Falsche im Vergleich zu dem SCAD Algorithmus mit der naiven Log-Likelihood-Funktion. Die 2er Werte im Betavektor wurden präziser geschätzt. Der 1er Wert ungefähr gleich und der 0,5 Wert ungenauer. Im SCAD Algorithmus mit der naiven Log-Likelihood-Funktion sind die Abweichungen größer als im Fall $\sigma = 0$ und gleichmäßiger. Die Werte wurden scheinbar gleichmäßig penalisiert, obwohl diese unterschiedlich groß sind.

3.3. Sigma = 0,2

Sigma = 0,2, n = 100, p=10, Wiederholungen=100

Auswertung $\sigma = 0,2$**,** $n = 100$**,** $p = 10$**,** $Wiederholungen = 100$		
λ	Naive Loglikelihood	Korrigierte Loglikelihood
$0,1 - 11,9$	Matrix singulär	Matrix singulär
12	Abweichungen:	
	0.65 0.80 0.88 0.83 0.82	
	Ø Korrekte Nullen =1,3	
	Ø Falsche Nullen =0	
21	Abweichungen:	Abweichungen:
	0.33 0.56 0.87 0.82 0.81	0.17 0.42 1.03 0.93 0.89
	Ø Korrekte Nullen = 1,8	Ø Korrekte Nullen = 3,2
	Ø Falsche Nullen =0,03	Ø Falsche Nullen = 0,28

Für $\sigma = 0,2$ und $n = 100$ ist der Algorithmus für die naive Log-Likelihood-Funktion ab einem λ von 12 ausführbar und liefert im Durchschnitt 1,3 richtig geschätzte Nullen und 0 Falsche. Die Abweichungen sind größer als in den Fällen $\sigma = 0/0,1$, was allerdings auch an der stärkeren Penalisierung liegen könnte. Ab einem λ von 21 ist der SCAD

Algorithmus auch für die korrigierte Log-Likelihood-Funktion ausführbar und liefert im Durchschnitt 3,2 richtig geschätzte Nullen und 0,28 Falsche. Die Abweichungen sind 0,83, 0,58, -0,03, 0,07 und 0,11. Im Vergleich dazu liefert der SCAD Algorithmus mit der naiven Log-Likelihood-Funktion im Durchschnitt 1,8 richtig geschätzte Nullen und 0,03 Falsche. Die Abweichungen sind 0,67, 0,44, 0,13, 0,18 und 0,19.

Sigma = 0,2, n = 500, p=10, Wiederholungen=100

Auswertung $\sigma = 0,2$, $n = 500$, $p = 10$, $Wiederholungen = 100$		
λ	Naive Loglikelihood	Korrigierte Loglikelihood
$0,1 - 2,4$	Matrix singulär	Matrix singulär
2,5	Abweichungen:	
	1.11 0.91 0.82 0.81 0.82	
	Ø Korrekte Nullen =0,37	
	Ø Falsche Nullen =0	
$8,9$	Abweichungen:	Abweichungen:
	0.87 0.87 0.86 0.86 0.86	0.42 0.77 1.03 1.01 1.01
	Ø Korrekte Nullen = 0,88	Ø Korrekte Nullen = 3,7
	Ø Falsche Nullen =0	Ø Falsche Nullen = 0,04

Für $\sigma = 0,2$ und $n = 500$ lässt sich der SCAD Algorithmus für die naive Log-Likelihood-Funktion ab einem λ von 2,5 ausführen und liefert im Durchschnitt 0,37 richtig geschätzte Nullen und 0 Falsche. Die Abweichungen sind -0,11, 0,09, 0,18, 0,19 und 0,18. Ab einem λ von 8,9 lässt sich der SCAD Algorithmus auch für die korrigierte Log-Likelihood-Funktion ausführen und liefert im Durchschnitt 3,7 richtig geschätzte Nullen und 0,04 Falsche. Die Abweichungen sind 0,58, 0,23, -0,03, -0,01 und -0,01. Im Vergleich dazu liefert der SCAD Algorithmus mit der naiven Log-Likelihood-Funktion im Durchschnitt 0,88 richtig geschätzte Nullen und 0 Falsche. Die Abweichungen sind 0,13, 0,13, 0,14, 0,14 und 0,14.

Sigma = 0,2, n = 1000, p=10, Wiederholungen=100

Auswertung $\sigma = 0,2$**,** $n = 1000$**,** $p = 10$**,** $Wiederholungen = 100$		
λ	Naive Loglikelihood	Korrigierte Loglikelihood
$0,1$	Abweichungen: 1.19 0.99 0.82 0.83 0.82 Ø Korrekte Nullen =0,03 Ø Falsche Nullen =0	Matrix singulär
$0,2 - 5,9$	Teilweise ausführbar, ähnliche Ergebnisse zu $\lambda = 0,1$	
6	Abweichungen: 0.96 0.93 0.84 0.85 0.85 Ø Korrekte Nullen = 0.45 Ø Falsche Nullen =0	Abweichungen: 0.52 0.91 1.01 1.01 1.01 Ø Korrekte Nullen = 3,8 Ø Falsche Nullen = 0,01

Für $\sigma = 0,2$ und $n = 1000$ ist der SCAD Algorithmus für $\lambda = 0,1$ bereits ausführbar und liefert im Durchschnitt 0,03 richtige Nullen und 0 Falsche. Die Abweichungen sind -0,19, 0,01, 0,18, 0,17 und 0,18. Danach ist der SCAD Algorithmus nur noch teilweise ausführbar für die naive Log-Likelihood-Funktion und liefert ähnliche Ergebnisse zu $\lambda = 0,1$. Ab einem λ von 6 ist der SCAD Algorithmus auch für die korrigierte Log-likelihood-Funktion ausführbar und liefert im Durchschnitt 3,8 richtig geschätzte Nullen und 0,01 Falsche. Die Abweichungen sind 0,48, 0,09, -0,01, -0,01 und -0,01. Im Vergleich dazu liefert der SCAD Algorithmus im Durchschnitt 0,45 richtig geschätzte Nullen und 0 Falsche. Die Abweichungen sind 0,04, 0,07, 0,16, 0,15 und 0,15.

Auswertung Sigma = 0,2

Im Falle von $\sigma = 0,2$ ist das Problem mit der Invertierbarkeit wiederum größer als im Fall $\sigma = 0,1$ und nochmals größer als im Fall $\sigma = 0$. Auch hier ist das Problem leichter zu handhaben mit dem SCAD Algorithmus für die naive Log-Likelihood-Funktion. Eine Erhöhung von n verbesserte das Problem bei beiden Algorithmen.

Für gleich große λ liefert der SCAD Algorithmus mit der korrigierten Log-Likelihood-Funktion im Durchschnitt mehr richtig geschätzte Nullen, aber teilweise auch mehr falsch geschätzte. Die Schätzungen für die 2er Werte im Betavektor sind genauer, für den 1er Wert ungefähr gleich und für den 0,5 Wert schlechter. Der SCAD Algorithmus mit der naiven Log-Likelihood-Funktion schätzt die 2er Werte im Durchschnitt zu niedrig ein und die Werte 0,5 und 1 zu hoch, wenn man bedenkt, dass der SCAD Strafterm größere Werte geringer bestrafen sollte.

Für $n = 1000$ und $\lambda = 0,1$ ist der SCAD Algorithmus für die naive Log-Likelihood-Funktion ausführbar und liefert trotz geringem Strafterm relativ hohe Abweichungen und weniger richtig geschätzte Nullen verglichen mit dem Fall $\sigma = 0$.

3.4. Sigma= 0,3

Sigma = 0,3, n = 100, p=10, Wiederholungen=100

Auswertung $\sigma = 0,3$**,** $n = 100$**,** $p = 10$**,** $Wiederholungen = 100$		
λ	Naive Loglikelihood	Korrigierte Loglikelihood
$0,1 - 15,9$	Ø Matrix singulär	Matrix singulär
16	Abweichungen:	
	0.67 0.71 0.76 0.70 0.70	
	Ø Korrekte Nullen =1,2	
	Ø Falsche Nullen =0,01	
31,9	Abweichungen:	Abweichungen:
	0.24 0.38 0.69 0.60 0.59	0.18 0.31 0.93 0.71 0.69
	Ø Korrekte Nullen = 2,1	Ø Korrekte Nullen = 3.1
	Ø Falsche Nullen =0.12	Ø Falsche Nullen = 0,51

Für $\sigma = 0,3$ und $n = 100$ ist der SCAD Algorithmus für die naive Log-Likelihood-Funktion ab einem λ von 16 ausführbar und liefert im Durchschnitt 1,2 richtig geschätzte Nullen und 0,01 Falsche. Die Abweichungen sind 0,33, 0,29, 0,24, 0,3 und 0,3. Ab einem λ von 31,9 ist der SCAD Algorithmus auch für die korrigierte Log-Likelihood-Funktion ausführbar und liefert im Durchschnitt 3,1 richtig geschätzte Nullen und 0,51 Falsche. Die Abweichungen sind 0,82, 0,69 0,07, 0,29 und 0,31. Im Vergleich dazu liefert der SCAD Algorithmus im Durchschnitt 2,1 richtig geschätzte Nullen und 0,12 Falsche. Die Abweichungen sind 0,76, 0,62, 0,31, 0,4 und 0,41.

Sigma = 0,3, n = 500, p=10, Wiederholungen=100

Auswertung $\sigma = 0,3$, $n = 500$, $p = 10$, $Wiederholungen = 100$		
λ	Naive Loglikelihood	Korrigierte Loglikelihood
$0,1$	Abweichungen:	Matrix singulär
	1.28 0.88 0.70 0.69 0.70	
	Ø Korrekte Nullen = 0,09	
	Ø Falsche Nullen =0	
$0,2$	Ergebnis ähnlich zu $\lambda = 0,1$	
$0,3 - 1,6$	Matrix singulär	
$1,7$	Abweichungen:	
	1.20 0.85 0.69 0.68 0.69	
	Ø Korrekte Nullen = 0,11	
	Ø Falsche Nullen =0	
$13,8$	Abweichungen:	Abweichungen:
	0.83 0.79 0.76 0.75 0.75	0.26 0.55 1.05 1.03 1.00
	Ø Korrekte Nullen = 0,68	Ø Korrekte Nullen = 3.7
	Ø Falsche Nullen =0	Ø Falsche Nullen = 0,26

Für $\sigma = 0,3$ und $n = 500$ ist der SCAD Algorithmus für die naive Log-Likelihood-Funktion ausführbar und liefert im Durchschnitt 0,09 richtig geschätzte Nullen und 0 Falsche. Die Abweichungen sind -0,28, 0,12, 0,3, 0,31 und 0,3. Für $\lambda = 0,2$ ist der Algorithmus auch ausführbar und dann nicht mehr bis $\lambda = 1,7$. Ab einem λ von 13,8 ist der SCAD Algorithmus für die korrigierte Log-Likelihood-Funktion ausführbar und liefert im Durchschnitt 3,7 richtig geschätzte Nullen und 0,26 Falsche. Die Abweichungen sind 0,74, 0,45, -0,05, -0,03 und 0. Im Vergleich dazu liefert der SCAD Algorithmus im Durchschnitt 0,68 richtig geschätzte Nullen und 0 Falsche. Die Abweichungen sind 0,17, 0,21, 0,24, 0,25 und 0,25.

Sigma = 0,3, n = 1000, p=10, Wiederholungen=100

Auswertung $\sigma = 0,3$, $n = 1000$, $p = 10$, $Wiederholungen = 100$		
λ	Naive Loglikelihood	Korrigierte Loglikelihood
$0,1$	Abweichungen: 1.23 0.92 0.70 0.71 0.70 Ø Korrekte Nullen = 0,01 Ø Falsche Nullen =0	Matrix singulär
$0,2 - 10,4$	Ergebnis ähnlich zu $\lambda = 0,1$	
$10,1$	Abweichungen: 0.95 0.84 0.74 0.75 0.74 Ø Korrekte Nullen = 0,24 Ø Falsche Nullen =0	Abweichungen: 0.30 0.76 1.02 1.02 1.03 Ø Korrekte Nullen = 3,6 Ø Falsche Nullen = 0,21

Für $\sigma = 0,3$ und $n = 1000$ lässt sich der SCAD Algorithmus ab einem λ von 0,1 ausführen und liefert im Durchschnitt 0,01 richtig geschätzte Nullen und 0 Falsche. Die Abweichungen sind -0,23, 0,08, 0,3, 0,29 und 0,3. Ab einem λ von 10,1 lässt sich auch der SCAD Algorithmus für die korrigierte Log-Likelihood-Funktion ausführen und liefert im Durchschnitt 3,6 richtig geschätzte Nullen und 0,21 Falsche. Die Abweichungen sind 0,7, 0,24, -0,02, -0,02 und -0,03. Im Vergleich dazu liefert der SCAD Algorithmus mit der naiven Log-Likelihood-Funktion im Durchschnitt 0,24 richtig geschätzte Nullen und 0 Falsche. Die Abweichungen sind 0,05, 0,16, 0,26, 0,25 und 0,26.

Auswertung Sigma = 0,3

Im Falle $\sigma = 0,3$ ist das Problem mit der Invertierbarkeit ebenfalls größer als im Fall $\sigma = 0,2$, größer als im Fall $\sigma = 0,1$ und nochmals größer als im Fall $\sigma = 0$. Das Problem ist ebenfalls leichter zu handhaben im SCAD Algorithmus für die naive Log-Likelihood-Funktion. Eine Erhöhung von n verbesserte das Problem bei beiden Algorithmen.

Für gleich große λ lieferte der SCAD Algorithmus mit der korrigierten Log-Likelihood-Funktion im Durchschnitt mehr richtig geschätzte Nullen und teilweise mehr falsche. Für $n = 500/1000$ schätzte der SCAD Algorithmus mit der korrigierten Log-Likelihood-Funktion die 2er Werte präziser und die Werte 0,5 und 1 ungenauer. Für den SCAD Algorithmus mit der naiven Log-Likelihood-Funktion sind die Schätzungen für die 2er Werte durchschnittlich zu niedrig und für die Werte 0,5 und 1 zu hoch. Eine Interpretation für die Abweichungen in $n = 100$ ist schwierig auf Grund des hohen λ Wertes.

Schlußbemerkung

Für die Fälle $\sigma = 0,2/0,3$ wurde teilweise der 0,5 und 1 Wert im Betavektor durchschnittlich zu hoch geschätzt und die 2er Werte zu niedrig für den SCAD Algorithmus mit naiver Log-Likelihood-Funktion. Im SCAD Algorithmus mit der korrigierten Log-Likelihood-Funktion wurden die Werte 0,5 und 1 niedriger geschätzt und die 2er Werte relativ genau, woraus man schließen kann, dass die SCAD Penalisierung wie vorhergesehen funktioniert für die korrigierte Log-Likelihood-Funktion und nicht richtig für die naive Log-Likelihood-Funktion.

In der Auswertung wurden noch die durchschnittlichen geschätzten Werte betrachtet (hier nicht abgebildet, aber im Code vorhanden). Dabei wurde festgestellt, dass der SCAD Algorithmus mit der korrigierten Log-Likelihood-Funktion teilweise den 0,5 Wert im Betavektor auf Null gesetzt hat, was die Anzahl an falsch geschätzten Nullen erklärt.

4. Fazit

Am Anfang dieser Arbeit wurde die Idee der korrigierten Score-Funktion erklärt. Korrigierte Score-Funktionen sind eine Möglichkeit Messfehler in den unabhängigen Variablen zu berücksichtigen und in dieser Arbeit wurde dafür speziell die Poisson-Verteilung betrachtet. Danach wurden Regularisierungsverfahren und ihre Eigenschaften vorgestellt. Das LASSO-Verfahren ist eine Methode Variablen zu selektieren und zu schrumpfen. In vielen Fällen bietet es sich an das Elastische Netz zu benutzen, welches eine Kombination aus Ridge Regression und LASSO-Verfahren ist (Zou und Hastie; 2005). Im Anschluss daran wurde versucht das LASSO-Problem für die korrigierte Score-Funktion zu lösen. (Tibshirani; 1996)(Nakamura; 1990). Dies ist im Falle der Poisson-Verteilung ein nichtlineares, konvexes Optimierungsproblem. Dabei traten Probleme mit gängigen Verfahren auf:

1. **Koordinatenabstieg** : Ableitung der korrgierten Score-Funktionen nach $\boldsymbol{\eta}$ war auf Grund des Korrekturterms ξ nicht möglich (Simon et al.; 2011).

2. **Prädiktor-Verbesserer Methode**: Ableitung nach $\boldsymbol{\mu}$ und $\boldsymbol{\eta}$ für die korrigierte Score-Funktion nicht möglich (Park und Hastie; 2007).

3. **Kleinste Winkel Regression**: Korrekturterm ξ fällt raus durch Ableitung nach α (Efron et al.; 2004).

4. **Innere-Punkte Methode**: „Linearer Prädiktor“ der korrigierten Score-Funktion liegt nicht in der passenden Form vor (Koh et al.; 2007).

5. **Nelder-Mead-Verfahren**: Keine Literatur in Zusammenhang mit dem LASSO gefunden (Nelder und Mead; 1965).

Anschließend wurde ein Algorithmus mit dem SCAD Strafterm verwendet (Fan und Li; 2001). Der SCAD Strafterm ist eine Modifizierung des LASSO Strafterms, verwendet eine quadratische Approximation an der Stelle $\boldsymbol{\beta}$ und arbeitet mit Newton Iterationen. Der SCAD Strafterm penalisiert große Koeffizienten weniger stark. Der SCAD Algorithmus wurde jeweils auf die naive und korrigierte Log-Likelihood-Funktion angewendet

und die Ergebnisse gegenübergestellt. Die Simulation führte den SCAD Algorithmus 30 mal aus, wiederholte dies 100 mal und bildete daraus die Kennwerte. Der Wert a war gleich 3,7 und für λ wurden mehrere Werte im Abstand von 0,1, beginnend bei 0,1, betrachtet. Bei der Auswertung trat teilweise das Problem auf, dass eine Matrix singulär wird und kein Ergebnis berechnet werden konnte. Die Ursache konnte im Rahmen dieser Arbeit nicht ermittelt werden. Für die Fälle in denen ein Ergebnis berechnet werden konnte wurde der Vergleich auf zwei Ebenen betrachtet: Die durchschnittliche Abweichung vom wahren Wert (Korrektur für Messfehler/Genauigkeit) und die durchschnittliche Anzahl richtig geschätzter Nullen (Variablenselektion). Dabei wurden die Fälle für verschiedene Standardabweichungen $\sigma = 0/0,1/0,2/0,3$ des Messfehlers und Stichprobengrößen $n = 100/500/1000$ betrachtet. Je größer σ, desto stärker die Auswirkung vom Messfehler. Eine Erhöhung von n verbessert allgemein die Genauigkeit und die durchschnittliche Anzahl richtig geschätzter Nullen bei beiden Algorithmen. Ab $\sigma = 0,1$ ist im Allgemeinen der SCAD Algorithmus für die naive Log-Likelihood-Funktion für kleinere Werte von λ schon ausführbar, wobei der SCAD Algorithmus für die korrigierte Log-Likelihood-Funktion meist größere Werte benötigt. Der wahre, simulierte Betavektor ist hier $\boldsymbol{\beta_t} = (0, 0, 0, 0, 0, 0.5, 1, 2, 2, 2)^T$. Der Vergleich findet statt für den kleinsten λ Wert, ab dem beide Algorithmen ausführbar sind.

Für $\sigma = 0$ sind die Ergebnisse für naive und korrigierte Log-Likelihood-Funktion gleich. Die Ergebnisse sind ähnlich zu den Simulationen von (Fan und Li; 2001).

Für $\sigma = 0,1$ ist das Problem mit der Invertierbarkeit größer als im Fall $\sigma = 0$, d.h. es werden größere Werte für λ benötigt um den Algorithmus auszuführen. Es zeigen sich stärkere, gleichmäßige Abweichungen für die naive Log-Likelihood-Funktion im Vergleich zu dem Fall $\sigma = 0$. Der SCAD Algorithmus mit der korrigierten Log-Likelihood-Funktion zeigt geringe Abweichungen für die 2er Werte des Betavektors und größere Abweichungen für die Werte 0,5 und 1. Der SCAD Algorithmus mit der korrigierten Log-Likelihood-Funktion schätzt im Durchschnitt mehr richtige Nullen als der SCAD Algorithmus mit der naiven Log-Likelihood-Funktion. Dies war für alle Stichprobengrößen n der Fall.

Für $\sigma = 0,2$ ist das Problem mit der Invertierbarkeit wiederum größer geworden im Vergleich zu den Fällen $\sigma = 0/0,1$. Die Ergebnisse für den SCAD Algorithmus mit der korrigierten Log-Likelihood-Funktion sind ähnlich zu $\sigma = 0,1$. Für den SCAD Algorithmus mit der naiven Log-Likelihood-Funktion zeigen sich stärkere, gleichmäßige Abweichungen und weniger durchschnittlich richtig geschätzte Nullen verglichen mit den Fällen $\sigma = 0/0,1$ und folglich auch weniger verglichen mit dem SCAD Algorithmus für die korrigierte Log-Likelihood-Funktion.

Für $\sigma = 0,3$ ist das Problem mit der Invertierbarkeit nochmals größer geworden im Vergleich zu den Fällen $\sigma = 0/0,1/0,2$. Die Ergebnisse für den SCAD Algorithmus mit der korrigierten Log-Likelihood-Funktion sind ähnlich zu $\sigma = 0,1/0,2$. Für den SCAD Algorithmus mit der naiven Log-Likelihood-Funktion zeigen sich wiederum stärkere, gleichmäßige Abweichungen und weniger durchschnittlich, richtig geschätzte Nullen verglichen mit den Fällen $\sigma = 0/0,1/0,2$.

Die Ergebnisse führen zu folgenden Überlegungen für die Variablen:

- σ : Je größer die Auswirkung des Messfehlers, desto größer die Verzerrungen und weniger richtig geschätzte Nullen im SCAD Algorithmus mit der naiven Log-Likelihood-Funktion. Der SCAD Algorithmus mit der korrigierten Log-Likelihood-Funktion schätzte die Koeffizienten für die verschiedenen σ relativ gleichmäßig, wobei die Abweichung größer war für die Werte 0,5 und 1 im Betavektor. Die gleichmäßigen Abweichungen im SCAD Algorithmus für die naive Log-Likelihood-Funktion könnten bedeuten, dass die SCAD Penalisierung hier nicht richtig funktioniert.

- n : eine Vergrößerung des Stichprobenumfangs verbessert das Problem mit der Invertierbarkeit und führt zu genaueren Ergebnissen.

Die Ergebnisse sprechen dafür, dass der SCAD Algorithmus mit der korrigierten Log-Likelihood-Funktion den SCAD Algorithmus mit der naiven Log-Likelihood-Funktion hinsichtlich Genauigkeit und richtige Selektion von Nullen dominiert. Auch unabhängig vom Vergleich zeigte der SCAD Algorithmus mit der korrigierten Log-Likelihood-Funktion, dass er die 2er Werte im Betavektor/große Werte für verschiedene σ genau schätzte, die Werte 0,5 und 1 im Betavektor/kleinere Werte stärker penalisierte und eine vernünftige Menge an Nullfaktoren richtig identifizierte.

Ausblick

In dieser Arbeit wurde gezeigt, dass der von (Fan und Li; 2001) vorgeschlagene Algorithmus mit dem SCAD Strafterm zusammen mit der korrigierten Log-Likelihood-Funktion von (Nakamura; 1990) eine geeignete Methode darstellt Koeffizienten zu schätzen und selektieren, falls die unabhängigen Variablen von additiven normalverteilten Messfehlern betroffen sind. Die kombinierte Methode wurde auf die Poisson-Verteilung angewendet. Interessant wäre es zu sehen, ob die kombinierte Methode auch für andere Verteilungen sinnvolle Ergebnisse liefert. Von Interesse wäre es ebenfalls zu untersuchen, wie es sich mit den Varianzen für die geschätzten Beta-Koeffizienten verhält.

Literaturverzeichnis

Buzas, J. S. (2009). A Note on Corrected Scores for Logistic Regression, *Statistics and Probability Letters* **79**(22): 2351–2358.

Buzas, J. S. und Stefanski, L. A. (1995). A Note on Corrected-Score Estimation, *Statistics and Probability Letters* **28**(1): 1–8.

Carroll, R. J., Ruppert, D., Stefanski, L. A. und Crainiceanu, C. (2006). *Measurement Error in Nonlinear Models: A Modern Perspective, Second Edition*, Monographs on Statistics and Applied Probability.

Donoho, D. L. und Johnstone, I. M. (1994). Adapting to Unknown Smoothness via Wavelet Shrinkage, *Journal of the American Statistical Association* **90**(432): 1200–1224.

Efron, B., Hastie, T. und Johnstone, I. (2004). Least Angle Regression, *The Annals of Statistics* **32**(2): 407–499.

Fahrmeir, L., Kneib, T. und Lang, S. (2009). *Regression: Modelle, Methoden und Anwendungen*, 2 edn, Springer-Verlag Berlin Heidelberg.

Fahrmeir, L., Künstler, R., Pigeot, I. und Tutz, G. (2016). *Statistik: Der Weg zur Datenanalyse*, 8 edn, Springer.

Fan, J. und Li, R. (2001). Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties, *Journal of the American Statistical Association* **96**(456): 1348–1360.

Friedman, J., Hastie, T. und Höfling, H. (2007). Pathwise Coordinate Optimization, *The Annals of Applied Statistics* **1**(2): 302–332.

Friedman, J., Hastie, T. und Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent, *Journal of Statistical Software* **33**(1): 1–22.

Furnival, G. M. und Wilson, R. W. (2012). Regressions by Leaps and Bounds, *Technometrics* **16**(4): 499–511.

Gill, J. und King, G. (2004). What to do when your Hessian is not invertible: Alternatives to Model Respecification in Nonlinear Estimation, *Sociological Methods and Research* **33**: 54–87.

Hastie, T. und Tibshirani, R. (1990). *Generalized Additive Models*, CRC Press.

Hastie, T., Tibshirani, R. und Friedman, J. (2009). *The Elements of Statistical Learning*, 2 edn, Springer-Verlag New York.

Heid, I., Küchenhoff, H., Miles, J., Kreienbrock, L. und Wichmann, H. (2004). Two Dimensions of Measurement Error: Classical and Berkson Error in residential Radon Exposure Assessment, *Journal of Exposure Analysis and Environmental Epidemiology* **14**(5): 365–377.

Huang, Y., Wen, C. und Hsu, Y. (2015). The Extensively Corrected Score for Measurement Error Models, *Scandinavian Journal of Statistics* **42**: 911–924.

Koh, K., Kim, S. und Boyd, S. (2007). An Interior-Point Method for Large-Scale l1-Regularized Logistic Regression, *Journal of Machine Learning Research* **8**: 1519–1555.

Leng, C., Lin, Y. und Wahba, G. (2006). A Note on the Lasso and related Procedures in Model Selection, *Statistica Sinica* **16**(4): 1273–1284.

Madigan, D. und Ridgeway, G. (2004). Discussion of "Least Angle Regression"by Efron et al., *The Annals of Statistics* **32**(2): 465–469.

McCullagh, P. und Nelder, J. (1989). *Generalized Linear Models*, 2 edn, CRC Press.

Nakamura, T. (1990). Corrected Score Function for Errors-in-Variables Models: Methodology and Application to Generalized Linear Models, *Biometrika* **77**(1): 127–137.

Nelder, J. A. und Mead, R. (1965). A Simplex Method for Function Minimization, *The Computer Journal* **7**: 308–313.

Park, M. Y. und Hastie, T. (2007). L1-Regularization Path Algorithm for Generalized Linear Models, *Journal of the Royal Statistical Society: Series B, Statistical Methodology* **69**(4): 659–677.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *https://www.R-project.org zuletzt aufgerufen 20.06.2018*

Rosset, S. (2004). Tracking Curved Regularized Optimization Solution Paths.
URL: *https://www.semanticscholar.org/paper/Tracking-Curved-Regularized-Optimization-Solution-Rosset/b15db51cebe5752e3f71ae657829e8d0e29ae3b0 zuletzt aufgerufen am 20.08.2018*

Segal, M. R., Dahlquist, K. und Conklin, B. R. (2003). Regression Approaches for Microarray Data Analysis, **10**(6): 961–980.

Simon, N., Friedman, J., Hastie, T. und Tibshirani, R. (2011). Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, *Journal of Statistical Software* **39**(5).

Stefanski, L. A. und Carroll, R. (1987). Conditional Scores and Optimal Scores for Generalized Linear Measurement- Error Models, **74**(4): 703–716.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso, *Journal of the Royal Statistical Society: Series B, Statistical Methodology* **58**(1): 267–288.

Tibshirani, R. (2011). Regression Shrinkage and Selection via the Lasso: a Retrospective, *Journal of the Royal Statistical Society: Series B, Statistical Methodology* **73**(3): 273–282.

Zhao, P. (2004). Boosted Lasso.
URL: *https://www.semanticscholar.org/paper/Boosted-Lasso-Zhao/97d9827922c929c8cd901211bd79df44cc92bd47 zuletzt aufgerufen am 20.08.2018*

Zou, H. und Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net, *Journal of the Royal Statistical Society: Series B, Statistical Methodology* **67**(2): 301–320.

A. Notation

Notation	Bedeutung
n	Stichprobenumfang
p	Anzahl Kovariablen
$\boldsymbol{z_i.}$	Zeilenvektor von unabhängigen Variablen z für Individuum i
$\boldsymbol{x_i.}$	Zeilenvektor von den für z gemessenen Werten x für Individuum i, mit Messfehler versehen
y_i	Wert der Zielvariable für Individuum i, Verteilung von $\boldsymbol{z_i.}$ abhängig
$\boldsymbol{Z}$	$n \times p$ Matrix mit den Zeilenvektoren $\boldsymbol{z_i.}$
$\boldsymbol{Y}$	Vektor der Länge n mit den Werten y_i
$\boldsymbol{X}$	$n \times p$ Matrix mit den Zeilenvektoren $\boldsymbol{x_i.}$
$l(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})$	naive Log-Likelihood-Funktion
$\boldsymbol{s}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})$	naive Score-Funktion
$\boldsymbol{I}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})$	naive Beobachtete Information
$\boldsymbol{\beta_t}$	Wahrer Parameterwert
$E(...\|\boldsymbol{Z}, \boldsymbol{Y})$	bedingter Erwartungswert von ... gegeben $\boldsymbol{Z}$ und $\boldsymbol{Y}$
$l^{cor}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})$	Korrigierte log likelihood bestehend aus x's und y's, wenn gilt : $\boldsymbol{E}(\boldsymbol{l^{cor}}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})\|\boldsymbol{Z}, \boldsymbol{Y}) = \boldsymbol{l}(\boldsymbol{\beta}, \boldsymbol{Z}, \boldsymbol{Y})$
$\boldsymbol{s^{cor}}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})$	Korrigierte Score Funktion bestehend aus x's und y's, wenn gilt : $\boldsymbol{E}(\boldsymbol{s^{cor}}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})\|\boldsymbol{Z}, \boldsymbol{Y}) = \boldsymbol{s}(\boldsymbol{\beta}, \boldsymbol{Z}, \boldsymbol{Y})$
$\boldsymbol{I^{cor}}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})$	Korrigierte Beobachtete Information bestehend aus x's und y's, wenn gilt: $\boldsymbol{E}(\boldsymbol{I^{cor}}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})\|\boldsymbol{Z}, \boldsymbol{Y}) = \boldsymbol{I}(\boldsymbol{\beta}, \boldsymbol{Z}, \boldsymbol{Y})$
$\boldsymbol{\beta^{cor}}$	Für Messfehler korrigierte Schätzung, wenn gilt: $\boldsymbol{s^{cor}}(\boldsymbol{\beta^{cor}}, \boldsymbol{X}, \boldsymbol{Y}) = 0$ und $\boldsymbol{I^{cor}}(\boldsymbol{\beta^{cor}}, \boldsymbol{X}, \boldsymbol{Y})$ positiv definit

$\lambda \sum \|\|\beta\|\|$	LASSO-Strafterm bestehend aus dem Regularisierungsparameter λ und der L1-Norm der βs.
$\boldsymbol{\beta_j^{cor,lasso}}$	korrigierte Lassoschätzung für das j-te β
S(x,a)	Softe-Schwellenwert-Funktion, definiert als $sign(x) * (\|x\| - a)$
$l_Q^{cor}(\boldsymbol{\beta, X, Y})$	Quadratische Approximation der korrigierten log likelihood durch eine Taylorreihe zweiter Ordnung
$\boldsymbol{\nabla l^{cor}(\beta, X, Y)}$	Gradient/erste Ableitung der korrigierten loglikelihood. => korrigierte Score Funktion
$\boldsymbol{H_{(l^{cor}(\beta,X,Y))}(\beta)}$	Hesse-Matrix/zweite Ableitung der korrigierten log likelihood an der Stelle $\beta = (\beta_1, ..., \beta_p)$ auch $\nabla^2 l^{cor}(\boldsymbol{\beta, X, Y})$ => negative korrigierte Information
$\boldsymbol{H_{(l^{cor}(\beta,X,Y))}(\beta)^{-1}}$	Inverse der Hesse-Matrix
$\boldsymbol{\eta}$	$= \boldsymbol{X\beta}$, linearer Prädiktor allgemein
$\boldsymbol{\tilde{\eta}}$	$= \boldsymbol{X\tilde{\beta}}$, geschätzter linearer Prädiktor im derzeitigen Rechenschritt
$\boldsymbol{v(\tilde{\eta})}$	$= \boldsymbol{\tilde{\eta} - H_{(l^{cor}(\beta,X,Y))}(\beta)^{-1} * \nabla(\tilde{\eta})}$
$w(\tilde{\boldsymbol{\eta}})_i$	i-ter diagonaler Eintrag der Hessematrix
$\boldsymbol{\xi}$	$= \frac{\boldsymbol{\beta^T \Lambda \beta}}{2}$
λ_{pois}	Erwartungswert und Varianz der Poisson-Verteilung
l_{inner}	Verlustfunktion in der Innere-Punkte Methode
$pen'_{lambda}(\theta)$	SCAD Strafterm
β_0	Intercept
$\boldsymbol{\beta}_0$	Startwerte/ Approximierungspunkt im Algorithmus mit SCAD Strafterm
β_{j0}	Die j-te Einheit aus $\boldsymbol{\beta}_0$
$\boldsymbol{\beta}_1$	Neuer Punkt nach einem Algorithmus mit SCAD Strafterm
$l_L^{cor}(\boldsymbol{\beta, X, Y})$	negative Log-Likelihood-Funktion

Erklärung zur Urheberschaft

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, wurden mit Angabe der Quellen kenntlich gemacht. Die vorliegende Arbeit ist, weder als Seminar noch als Bachelorarbeit, einer Prüfungsbehörde vorgelegt worden.

München, den 22. August 2018 Quan Nguyen