

Fakultät für Mathematik

FernUniversität Hagen

Ein Active-Set-Verfahren quadratischer Laufzeit zur Least-Squares-Approximation von Daten monotonen Ursprungs

Bachelorarbeit
(2., korr. Version vom 15.9.2014)

Daniel Altmann
Matrikel-Nummer: 8314942

Themensteller und Betreuer: Dr. rer. nat. Eugen Grycko,
Prof. Dr. Winfried Hochstättler

Inhaltsverzeichnis

1. Einleitung	4
2. Eine Motivation der monotonen Regression	6
2.1. Kleinste-Quadrate-Abstände von Punkten zu Mengen	6
2.2. Die Idee der Hüllmenge	7
2.3. Ein Anwendungsfall: Die Behandlung negativer Ausreißer	9
2.4. Ein Beispiel mit allgemeineren Anwendungen	13
3. Ein Algorithmus zur Berechnung der Regression	17
3.1. Die KKT-Bedingungen im Kontext konvexer Probleme	17
3.2. Linear beschränkte, quadratische Probleme (QPs)	20
3.3. Eine Vorstellung des Frameworks von Best und Chakravarti	21
3.4. Ein Divide-and-Conquer-Verfahren zur Lösung des MR-Problems	25
3.5. Der „Pool Adjacent Violators Algorithm“	39
4. Schlußwort	44
Anhang	46
A. Der MSE (Mean Squared Error)	47
Literatur	49

Wir verwenden in dieser Arbeit folgende Konventionen:

- \mathbb{N} sei die Menge der natürlichen Zahlen $\{0, 1, 2, \dots\}$.
- \mathbb{R} sei die Menge der reellen Zahlen.
- Vektoren werden in fett gedruckten Kleinbuchstaben notiert:
 $\mathbf{u}, \mathbf{v}, \mathbf{w}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\eta}$
- Vektorwertige Abbildungen und Matrizen werden ebenfalls fett gedruckt – speziell Matrizen zusätzlich in Großbuchstaben:
 $\mathbf{f}, \mathbf{g}, \mathbf{h}, \mathbf{A}, \mathbf{B}, \mathbf{C}$
- Skalare werden in Kleinbuchstaben notiert:
 a, b, c, μ, λ
- Mengen werden in Kalligraphieschrift notiert:
 $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{M}, \mathcal{N}$
- Ein Optimierungsproblem wird in Kalligraphieschrift und umgebenden Klammern notiert:
 $(\mathcal{P}), (\mathcal{QP})$

1. Einleitung

In dieser Arbeit werden wir einen Algorithmus zu Berechnung der monotonen Regression vorstellen, der von Herrn Günter Klützke stammt. In Kapitel 2 werden wir mögliche Anwendungen der monotonen Regression motivieren. In Kapitel 3 folgt eine nach didaktischen Gesichtspunkten aufgebaute Vorstellung des Algorithmus sowie ein Beweis seiner Korrektheit. Beide Abschnitte bedürfen im Hinblick auf die Aktualität einiger Anmerkungen.

Das Problem der monotonen Regression (*isotonic regression problem*) ist in mehreren Abwandlungen seit über 40 Jahren bekannt und wurde, in der Ausprägung wie wir sie hier vorstellen, algorithmisch erfolgreich gelöst. Daher gibt es entsprechend viele Anwendungen und wir verweisen exemplarisch auf die Changepoint-Analyse in [WWM01] bzw. die Analyse der Daten eines Massenspektrometers in [Kea06]. Es würde den Rahmen dieser Arbeit sprengen, darauf einzugehen. Wir haben einen experimentellen Zugang gewählt, um den Nutzen und einfache Anwendungen der monotonen Regression zu veranschaulichen. Entsprechend sind die Abschnitte unter Kapitel 2 als illustrative Beispiele zu verstehen und nicht als *state of the art*.

Es gibt tatsächlich einen $O(n)$ -Algorithmus, den sogenannten *pool adjacent violators algorithm*, für dieses Problem, siehe [GW84] für eine genaue Implementierung. Er ist auch heute verbreitet und wurde, z. B. im R-Iso-Package, implementiert. Im Hinblick auf die Laufzeit ist dieser besser als der von uns vorgestellte Klützke-Algorithmus. Wir finden jedoch, daß der Klützke-Algorithmus ein besonders leicht zu implementierendes Divide-and-Conquer-Verfahren darstellt und deswegen eine eigene Erwähnung verdient.

Man kann Algorithmen als Black Box (Ein-/Ausgabeverhalten) oder als White Box (*wie* wird verfahren) betrachten. Die letztere Sichtweise ist im allgemeinen etwas unbequemer, da die Motivation, sich eine Sache genauer anzuschauen, wohl häufig aus einem Mangel an Information herrührt, während die erste Perspektive bereits eine klare Priorität voraussetzt. Es kommt immer wieder vor, daß man über Tabellen stolpert, welche z. B. die Zeitkomplexität von Algorithmen im Hinblick auf ein bestimmtes Problem vergleichen – eine typische Black Box Perspektive, die man natürlich immer wieder einnehmen muß, um überhaupt zu einer Lösung zu gelangen. Daß diese Sicht nicht stets der Wahrheit letzter Schluß ist, zeigt ein Vergleich zwischen dem Dijkstra-Algorithmus und dem Bellman-Ford-Algorithmus, um nur ein Beispiel zu nennen.

Wir sind gegen Ende der Bearbeitungszeit auf eine Verallgemeinerung unseres Verfahrens gestoßen, siehe [LRS10], was wir im Schlußwort wieder aufgreifen werden. Solch eine Erfahrung zeigt deutlich, daß man auch eine alternative Herangehensweise nicht verwerfen sollte, nur weil sie sich in einer aktuellen Ausführung nicht behaupten kann. Vielmehr sollte man in einer

genauen Analyse herausfinden, welche Informationen ein bestimmter Algorithmus, abgesehen von der formalen Ausgabe, eigentlich berechnet.

Ich möchte mich an dieser Stelle bei meinen Betreuern Herrn Dr. Eugen Grycko und Herrn Professor Dr. Winfried Hochstättler für das schöne Thema und die freundliche und hilfreiche Betreuung bedanken.

2. Eine Motivation der monotonen Regression

2.1. Kleinste-Quadrate-Abstände von Punkten zu Mengen

In diesem Abschnitt wollen wir – aus einer etwas abstrakteren Sicht – ein Approximationsproblem vorstellen: Gegeben sei ein funktionaler Zusammenhang $\tilde{f} : \mathbb{R} \rightarrow \mathbb{R}$, welcher experimentell an den Stützstellen $x_1 \leq x_2 \leq \dots \leq x_n$ gemessen wurde und mit Meßfehlern behaftet ist. Die Meßwerte seien gegeben durch

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

Wir nehmen nun an, daß die Funktion \tilde{f} aus theoretischen Überlegungen heraus monoton sein muß und bis auf einen Parameter α durch einen Term T bestimmt ist, d. h.

$$\forall a, b \in \mathbb{R} : a \leq b \Rightarrow \tilde{f}(a) \leq \tilde{f}(b),$$

$$\exists \alpha \in \mathbb{R} \forall x \in \mathbb{R} : \tilde{f}(x) = T(x, \alpha).$$

Betrachtet man die Menge von Funktionen

$$\{f : \mathbb{R} \rightarrow \mathbb{R} \mid \exists \alpha \in \mathbb{R} \forall x \in \mathbb{R} : f(x) = T(x, \alpha)\}$$

nicht über \mathbb{R} , sondern über der Menge der Stützstellen $\{x_i \mid i \in \{1, \dots, n\}\}$, so erhält man die folgende Menge $\mathcal{F} \subseteq \mathbb{R}^n$:

$$\mathcal{F} = \{\mathbf{f} \in \mathbb{R}^n \mid \exists \alpha \in \mathbb{R} \forall x \in \{x_1, \dots, x_n\} : f_i = T(x_i, \alpha)\}.$$

Gesucht ist nun das Element $\mathbf{f} \in \mathcal{F}$, welches den gemessenen Datenvektor

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

am besten bezüglich der euklidischen 2-Norm $\| \cdot \|_2$ approximiert. Das wäre formal also

$$\begin{aligned} \min \mathbf{f} : & \quad \|\mathbf{f} - \mathbf{y}\|_2 \\ \text{u. d. N. :} & \quad \mathbf{f} \in \mathcal{F}. \end{aligned} \tag{2.1.0.1}$$

Das ist eine allgemeine Formulierung eines Regressionsproblems mit einem Parameter. Es sei darauf hingewiesen, daß die Eindeutigkeit des Minimums in dieser Form nicht gesichert ist, insbesondere falls die Menge \mathcal{F} nicht konvex ist. Für die Beispiele in diesem Text spielt das jedoch keine große Rolle.

2.2. Die Idee der Hüllmenge

Sowohl die Menge aller monotonen Funktionen als auch die Menge aller monotonen Vektoren $\mathcal{M} = \{\mathbf{v} | \mathbf{v} \in \mathbb{R}^n \text{ mit } i \leq j \Rightarrow v_i \leq v_j\}$ sind konvexe Mengen. Ein gemessener Datenvektor ist aufgrund von Meßfehlern im allgemeinen nicht mehr monoton, selbst wenn der zugrunde liegende Zusammenhang monoton ist. Hier wird, wie anfangs erwähnt, $\mathcal{F} \subseteq \mathcal{M}$ vorausgesetzt. Sei nun \mathbf{m} das Element aus \mathcal{M} , für welches $\|\mathbf{m} - \mathbf{y}\|_2$ minimal wird. Auch für beliebige Mengen sagen wir \mathbf{m} ist *minimal aus \mathcal{M} für \mathbf{y}* oder, sofern keine Verwechslungen zu befürchten sind, sprechen wir von \mathbf{m} als dem *minimalen Element* oder einfach dem *Minimum*. Das Auffinden von \mathbf{m} wird in der Literatur, z. B. in [BC90] oder [Sta], mit abweichenden Zusätzen, als *isotonic regression problem* bezeichnet. Dabei wird vor dem Hintergrund allgemeinerer Probleme unterschieden, welche Partialordnungsbeziehung zwischen den m_i vorliegen soll oder bezüglich welcher Norm minimiert wird. Hier wird nur der Fall der gewöhnlichen, euklidischen 2-Norm $\| \cdot \|_2$ im \mathbb{R}^n sowie die lineare, auch totale, Partialordnung der m_i als Nebenbedingung betrachtet. Also $m_1 \leq m_2 \leq \dots \leq m_n$. Das entspricht der Forderung nach Monotonie, da wir die Stützstellen x_i als geordnet annehmen. Dem Fachbegriff *isotonic regression problem with respect to a complete order* aus [BC90] weisen wir kurzerhand die Sprechweise *monotone Regression* oder das Kürzel *MR* zu, welches wir entsprechend verwenden. Wir sprechen also von der *MR-Approximation*, der *MR-Lösung* oder dem *MR-Problem*. Wir haben uns dabei von der *KQ*-Notation aus [FKPT01] inspirieren lassen. *KQ* steht dort für „Kleinste-Quadrate“. Hier besitzt \mathcal{M} eine besonders einfache, konvexe Struktur, und man darf darauf hoffen zu einem leichteren Problem zu kommen, indem man nicht (2.1.0.1) löst, sondern (man beachte: \mathbf{m} ist das Minimum aus \mathcal{M} für \mathbf{y})

$$\begin{aligned} \min \mathbf{f} : & \quad \|\mathbf{f} - \mathbf{m}\|_2 \\ \text{u. d. N. :} & \quad \mathbf{f} \in \mathcal{F}. \end{aligned} \tag{2.2.0.2}$$

Dies ist jedoch kein äquivalentes Problem. Im Hinblick auf die Zielfunktion dürfte dies unmittelbar klar sein, doch auch das minimierende Element kann ein gänzlich anderes sein – selbst wenn \mathcal{F} ebenfalls konvex ist. Um diesen Sachverhalt im Prinzip zu veranschaulichen, folgen zwei Beispiele.

Stellvertretend für die Menge \mathcal{M} betrachten wir die konvexe Menge

$$\mathcal{L} = \{\mathbf{v} | \mathbf{v} \in \mathbb{R}^2 \text{ mit } v_1 \leq 0\}$$

und ihre Teilmenge

$$\mathcal{T} = \{\mathbf{v} \in \mathbb{R}^2 \mid \exists \alpha \in [0, 2] : \mathbf{v} = \begin{pmatrix} -2 + \alpha \\ \alpha \end{pmatrix}\} \subseteq \mathcal{L}.$$

In Abbildung 2.1 gilt also $\mathcal{T} = [\mathbf{c} \mathbf{a}]$, wobei \mathcal{L} der linken Halbebene entspricht. Der Punkt

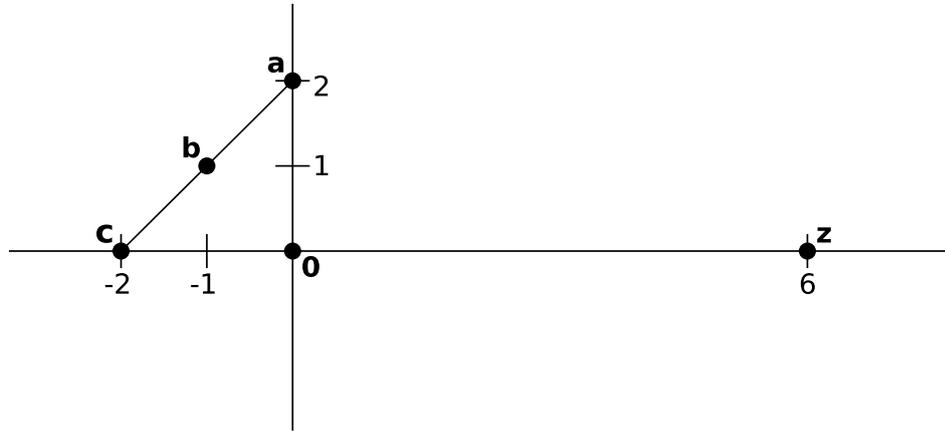


ABBILDUNG 2.1.: Beispiel zu Optimalpunkten, schlechter Fall

aus \mathcal{T} , der \mathbf{z} am nächsten liegt, ist \mathbf{a} . Das entspräche $\alpha = 2$ in der Parameterdarstellung. Der minimale Abstand zwischen der Menge \mathcal{L} und dem Punkt \mathbf{z} wird durch $\mathbf{0}$ realisiert. Versucht man den Minimalpunkt aus $\mathcal{T} = [\mathbf{c} \mathbf{a}]$ für \mathbf{z} durch das Problem

$$\begin{aligned} \min \mathbf{p} : & \quad \|\mathbf{p} - \mathbf{0}\|_2 \\ \text{u. d. N. :} & \quad \mathbf{p} \in \mathcal{T} \end{aligned} \tag{2.2.0.3}$$

anzunähern, so kommt man zu dem völlig anderen Ergebnis \mathbf{b} . Insbesondere in der Parameterdarstellung wird der Unterschied zwischen \mathbf{a} und \mathbf{b} deutlich: Während \mathbf{b} bei $\alpha = 1$ liegt, findet sich das wahre Minimum \mathbf{a} bei $\alpha = 2$.

Eine unkritische Anwendung der Idee in zwei Stufen zu approximieren – erst in einer vereinfachten Obermenge, dann im eigentlichen Suchraum – kann also zu schlechten Resultaten führen. Auch die Parameterdarstellung spielt hier eine Rolle. Hat man allerdings Grund zu der Annahme, daß die Menge \mathcal{M} in „Gestalt und Form“ gut geeignet ist, um Approximationen in \mathcal{T} zu erzeugen, so kann die gleiche Idee schöne Resultate produzieren, wie in Abbildung 2.2 zu sehen ist. Man betrachte hierzu \mathcal{M}' , innerhalb des gestricheltem Rands, als approximierende Menge mit Minimalpunkt \mathbf{o}' für \mathbf{Z} . Es ist einleuchtend, daß die Lösung für

$$\begin{aligned} \min \mathbf{p} : & \quad \|\mathbf{p} - \mathbf{o}'\|_2 \\ \text{u. d. N. :} & \quad \mathbf{p} \in \mathcal{T} \end{aligned} \tag{2.2.0.4}$$

in diesem Fall ziemlich nahe bei \mathbf{a} liegt.

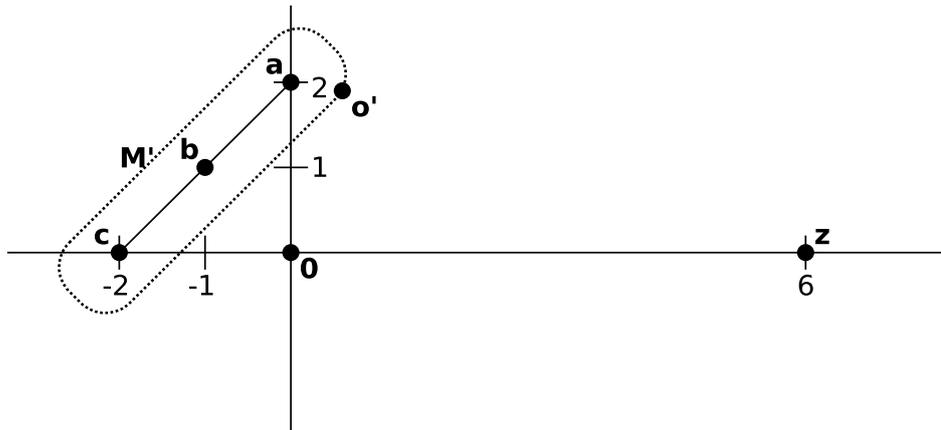


ABBILDUNG 2.2.: Beispiel zu Optimalpunkten, gutartiger Fall

2.3. Ein Anwendungsfall: Die Behandlung negativer Ausreißer

Es kommt in der Praxis häufig vor, daß man aus pragmatischen Gründen ein nichtlineares Regressionsproblem durch Logarithmierung zu einem linearen Problem macht, siehe z. B. [FKPT01]. In diesem Abschnitt soll vorgestellt werden, wie negative Ausreißer mithilfe der MR-Lösung „verschwinden“ und so einer Logarithmierung zugänglich gemacht werden.

Hierzu betrachte man folgendes Modell, das auf einen Vorschlag von Herrn Dr. Eugen Grycko hin umgesetzt wurde. Die Funktion $f(x) = x^\alpha$, wobei $\alpha > 0$ sei, werde an den Stützstellen x_1, \dots, x_n ausgewertet; wieder entstehen Meßfehler S_{x_i} . Diese werden als unabhängig und identisch normalverteilt angenommen:

$$S_{x_i} \sim \mathcal{N}(0, \sigma^2).$$

Die tatsächlichen Meßwerte seien also $\tilde{y}_i := \tilde{f}(x_i) = x_i^\alpha + S_{x_i}$ – eine fehlerfreie Messung ergäbe $y_i^{(\alpha)} = x_i^\alpha$, wobei der geklammerte, griechische Buchstabe über y hier weder als Potenz noch als Ableitung zu verstehen ist, sondern einfach als Parameter des funktionalen Zusammenhangs $f(x) = x^\alpha$. Der Parameter α ist in einer realen Anwendung unbekannt und soll durch kleinste Quadrate geschätzt werden. In diesem Sinne gesucht ist also das minimierende β für

$$\begin{aligned} \min \beta : & \quad \|\mathbf{y}^{(\beta)} - \tilde{\mathbf{y}}\|_2 \\ \text{u. d. N. :} & \quad \beta \in \mathbb{R}_+. \end{aligned} \tag{2.3.0.5}$$

Unser Experiment wurde mit $\alpha = 1.2$ durchgeführt. Jeder Datensatz wurde in Schritten von 0.001 zwischen 2.5 und $(3.5 - 0.001)$ bei einer Streuung von $\sigma = 3$ mittels normalverteilter Zufallszahlen erzeugt – jeder Datensatz besteht also aus 1000 Stützstellen und ebenso vielen Meßwerten. Wir greifen einen solchen heraus: In Abbildung 2.3 sieht man deutlich, wie einige Ausreißer im negativen Bereich liegen. Nach der monotonen Regression liegen in diesem Beispiel nur noch positive Werte vor, wie in Abbildung 2.4 zu sehen ist. Während die ursprüngliche

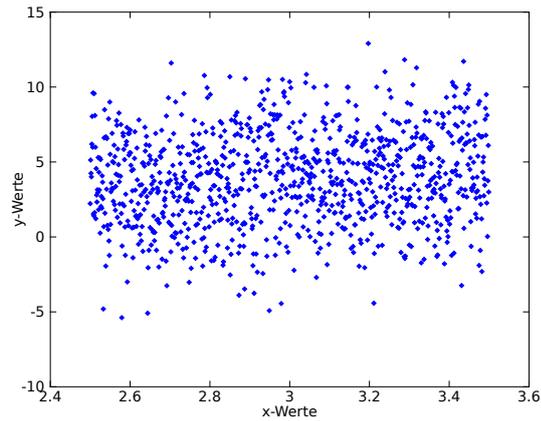


ABBILDUNG 2.3.: Beispieldatensatz mit negativen Ausreißern

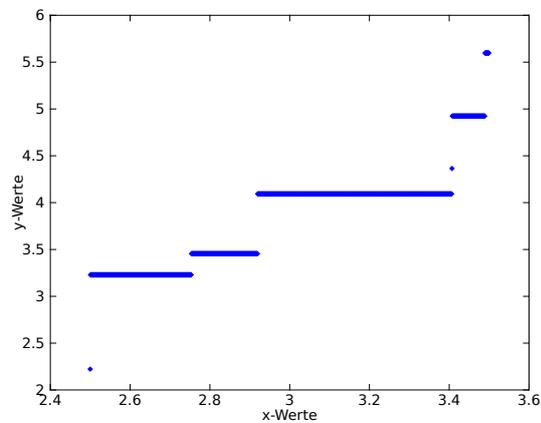


ABBILDUNG 2.4.: MR-Approximation ohne negative Ausreißer

Fehlergleichung $x_i^\beta = \tilde{y}_i$ lautet, und ein Logarithmieren bzw. Linearisieren wegen negativen Ausreißern nicht möglich war, lautet die Fehlergleichung nach der Regression $x_i^\beta = m_i$, wobei m die MR-Lösung für den Datensatz \tilde{y} darstellt. Diese Fehlergleichung kann man bei solchen Datensätzen logarithmieren, die nach der Regression keine negativen Ausreißer mehr enthalten. Sie ist dann in der Form

$$\beta \log(x_i) = \log(m_i)$$

sogar einer *linearen* Regression zugänglich, wenn man davon absieht, daß der Fehlerterm nun nicht mehr additiv und normalverteilt in die Daten eingeht. Sie liefert in unseren Tests trotzdem eine relativ gute Schätzung für α – erst recht im Vergleich zur naiven Idee alle negativen Werte zu vernachlässigen, worauf wir nochmals zurück kommen werden. Die vorgeschlagene Methode ist bei weitem nicht die einzige Methode, einen gestörten Datensatz „auszumitteln“; dazu könnten auch gleitende Durchschnitte verwendet werden, siehe z. B. [FKPT01], Kapitel 14 über Zeitreihen.

In unserem Experiment wurden 1259 Datensätze erzeugt. Bei 259 davon waren auch nach einer

MR-Approximation noch negative Werte vorhanden¹. Diese konnten demnach auch nicht logarithmiert werden. Bei einer Datensatzgröße von 1000 Punkten zwischen 2.5 und $(3.5 - 0.001)$, $\alpha = 1.2$ und $\sigma = 3$ überrascht es nicht, daß alle Datensätze die Monotoniebedingung verletzen und negative Werte besaßen. Die Datensätze hatten im Durchschnitt etwa 108.442 negative Ausreißer. Der empirische, quadrierte Bias unserer logarithmierten, linearen Kleinste-Quadrate-Schätzung auf monoton geglätteten Daten betrug etwa $7.173 \cdot 10^{-6}$, und die empirische Varianz betrug $5.914 \cdot 10^{-4}$. Das entspricht einem empirischen MSE von $5.986 \cdot 10^{-4}$. Das ist durchaus vergleichbar mit Bias und Varianz einer logarithmierten Kleinste-Quadrate-Schätzung, die nicht unter negativen Ausreißern leidet und auch nicht geglättet wurde. Im Hinblick auf solche Vergleiche werden wir im folgenden ein differenziertes Bild zeigen.

Diesem ersten Beispiel muß man Konstruiertheit zugestehen. Der Anteil an unbrauchbaren Datensätzen, die trotz Störung immer noch monoton sind, a priori keine negativen Werte enthalten oder auch nach der MR-Regression noch negative Werte enthalten, läßt sich durch geeignete Wahl von α , σ , Startpunkt und Dichte des Datensatzes gut anpassen. Für eine Simulation bedeutet das: Man müßte im ungünstigen Fall sehr viele Datensätze erzeugen, um überhaupt einen Vergleich durchführen zu können. Für die Praxis heißt das: Diese „nicht zugänglichen“ Datensätze können entsprechend häufig oder selten vorliegen. Man wird wohl von Anwendung zu Anwendung entscheiden müssen, ob diese Methode praktikabel ist oder nicht. Man kann aber zumindest dem Problem der negativen Werte *nach* der Regression dadurch entgegen wirken, indem man diese verwirft. Diese Werte treten naturgemäß am Anfang des Datensatzes auf. Gemessen an der naiven Idee, vorweg alle negativen Werte im Datensatz zu verwerfen, was bei bestimmten Parameterkombinationen ein beträchtlicher Teil sein kann, ist diese Vorgehensweise in bestimmten Bereichen besser, meistens ebenbürtig, in seltenen Fällen aber auch unterlegen.

Wir haben uns dazu entschlossen – anstelle einer detaillierten Auflistung von Bias, Stichprobenvarianz und eventuell Gauß- bzw. Binomialtests für ausgewählte Parameterbereiche – eine Art Überblicksgrafik zu erstellen, die wir für interessanter halten.

Der Plot in [Abbildung 2.6](#) entstand wie folgt: Wir haben für jede der beiden Methoden Datensätze mit 250 Stützstellen über ein Intervall der Länge von ca. 1 berechnet, bspw. mit den Stützstellen $1, 1.004, \dots, 2 - 0.004$. Die Parameter α und Startpunkt s wurden aus $\{0.1, 0.2, \dots, 10\}$ variiert; die Auflösung Rasters ist also 0.1. An jedem Punkt des Rasters wurden 100 Datensätze berechnet. Anschließend wurde verglichen, welche der beiden Schätzmethoden:

- „Naive Methode“: Weglassen aller negativen Werte im Datensatz; Logarithmierung; Lineare KQ-Schätzung.
- Berechnung der MR-Regression; Weglassen aller negativen Werte nach der Regression; Logarithmierung; Lineare KQ-Schätzung.

näher am wahren Wert α lag. Lag der naive Schätzwert näher an α , so wurde der Wert im

¹Tatsächlich lief die Simulation so lange, bis 1000 brauchbare Datensätze zur Verfügung standen.

Raster an der Stelle (α, s) um eins hochgezählt. Ergaben die Schätzungen den gleichen Wert, so wurde um 0.5 hochgezählt und lag die naive Schätzung weiter von α entfernt, so wurde nicht hochgezählt. Bei jeweils 100 Experimenten pro Parameterkombination sind also in jedem Punkt des Rasters Werte zwischen 0 und 100 möglich. Bei noch kleineren Werten für s bzw. größeren Werten für α ist es möglich, daß selbst ein kompletter Datensatz nicht ausreicht um echt positive Werte nach der monotonen Regression zu erhalten. Dunkelblau steht für einen Wert von 0, Grün für 50 und Dunkelrot für 100. Im rechten unteren Bereich von Abbildung 2.6 stehen nur 50er Werte. Zum Vergleich geben wir eine Farbmatrix, zu sehen in Abbildung 2.5, mit Werteverlauf von links nach rechts an. Für große α -Werte und große Startwerte gibt es

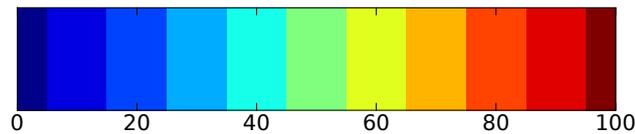


ABBILDUNG 2.5.: Farbwerte von 0 bis 100, grobe Unterteilung

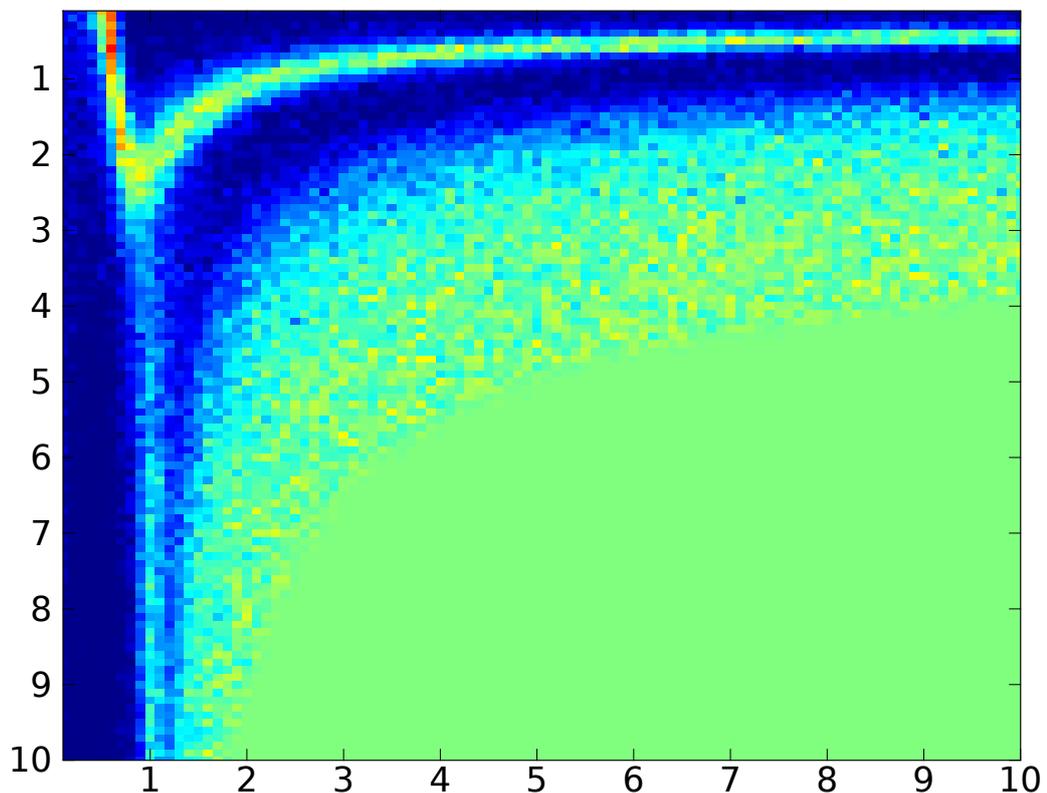


ABBILDUNG 2.6.: Farbraster für den Experimentausgang, vertikal α , horizontal s

zunehmend keine negativen Ausreißer mehr; ab einem gewissen Punkt kommen auch praktisch nur noch monotone Datensätze vor, also Datensätze, die zwar gestört, aber aufgrund des hohen

α -Wertes trotzdem monoton steigend sind. Es wundert also nicht, daß der untere, rechte Bereich des Rasters monochrom in Grün erscheint. Bemerkenswert jedoch ist die hakenförmige Struktur im oberen, linken Bereich, die sich nach rechts zieht sowie kleinere Inseln, in denen die naive Methode dominiert.

Es dürfte nicht schwer fallen, sich die Kurven für verschiedene α -Werte vorzustellen. Man kann intuitiv argumentieren, warum speziell kleine Stützstellen wichtig sind, wenn man es mit kleinen α -Werten zu tun hat. Wir werden davon aus Gründen der Sachlichkeit jedoch ablassen. Unser Fazit ist: Im Falle einer realen Anwendung sollte man eine Vorstellung davon haben, aus welchem Parameterbereich ein Datensatz stammt und es empfiehlt sich eigene Datensätze zu erzeugen, um zu testen, welche Schätzmethode die gewünschten Anforderungen wie Rechenzeit und Effektstärke in angemessen signifikanter Weise erfüllt. Auch das Problem, daß bestimmte Effekte erst ab einer gewissen Dimension beobachtbar werden, macht ein auf die konkrete Anwendung zugeschnittenes Experiment unumgänglich. In unserem Fall rührt diese Einsicht aus einem Versuch her, die Matrix aus Abbildung 2.6 mit größerer Genauigkeit, d. h. mit 1000 Datensätzen pro Punkt, zu berechnen. Wie sich heraus gestellt hat, sind Datensätze, die auch nach der Regression komplett im negativen Bereich liegen, zwar selten, sie kommen aber vor. In diesem Sinne zitieren wir aus [MGPS98], S. 162: „Experimental evaluation cannot be substituted when judging statistical procedures.“

Je nach Fähigkeit eines Mathematikers wird es wohl mehr oder weniger schwierig sein, ein Schätzverfahren, das in der Praxis durchaus einem spontanen Einfall entspringen kann, formal zu analysieren. Unser Eindruck ist auch, daß man erschwerte Bedingungen hat, wenn man nicht weiß welche „positive“ Eigenschaft eines Algorithmus man nachweisen möchte. Es spricht aus unserer Sicht einiges dafür, Indizien für einen solchen Prozeß im Experiment zu gewinnen.

2.4. Ein Beispiel mit allgemeineren Anwendungen

In dem Artikel [Sta] findet man viel nützliche Information, in welchem Kontext die MR-Approximation benutzt wird. Ein in unseren Augen besonders schönes Beispiel daraus, wollen wir hier, etwas ausgeschmückt, reproduzieren.

Hierzu stelle man sich vor, man würde eine bestimmte Spezies auf abgegrenztem Raum beobachten, und man hätte keine oder nur sehr unpraktikable Methoden, die Population direkt zu zählen. Stattdessen benutzt man z. B. Bewegungssensoren und betrachtet zu entsprechend engmaschig gewählten Zeitpunkten das Ereignis „Ausschlag“ bzw. „kein Ausschlag“. Es ist einleuchtend, daß die Sensoren um so häufiger das Ereignis „Ausschlag“ aufzeichnen werden, je größer die Population geworden ist; bei begrenztem Raum ist diese Frequenz also proportional zur Populationsgröße selber.

[Heu09] hat einige sehr schöne, einführende Beispiele zur logistischen Funktion und ihren Anwendungen. Die Modellierung und Lösung der logistischen Differentialgleichung haben wir

daraus übernommen (man vergleiche dort §1, den Abschnitt über logistisches Wachstum). Die alternative Formulierung findet man in gängigen Quellen; da jedoch nur ein Ausklammern nötig ist, verweisen wir nicht weiter darauf.

Die Modellierung der Differentialgleichung lautet also:

$$\frac{dP}{dt} = \lambda P(K - P).$$

Hier steht P für die Populationsgröße in „Anzahl der Lebewesen“; sie hängt von der Zeit t ab. λ ist eine positive Konstante, die mit dem Wachstum *zu tun* hat, und K beschreibt die Kapazität des Lebensraums. Nach einer Umformung, zu sehen in Gleichung (2.4.0.6), kann man den letzten Faktor naheliegender als Dämpfungsfaktor aus $[0, 1]$ interpretieren.

$$\frac{dP}{dt} = \lambda K P \left(1 - \frac{P}{K}\right) \quad (2.4.0.6)$$

Für kleines P ähnelt diese Gleichung einer linearen Differentialgleichung mit einem konstanten Koeffizienten λK und zeigt in diesem Bereich auch ähnliches Verhalten. Die Lösung der logistischen Differentialgleichung ist gegeben durch

$$P(t) = \frac{K}{1 + \left(\frac{K}{P_0} - 1\right) e^{-\lambda K t}}.$$

In der alternativen Formulierung sieht man etwas deutlicher, wie sich der Koeffizient λK in der Lösung niederschlägt: Er entspricht der Wachstumsrate. In unserem Beispiel wurde als Beobachtungszeitraum 10 gewählt, d. h. also $t \in [0, 10]$, $\lambda = 0.01$ und $K = 100$, woraus $\lambda K = 1$ folgt. Die Population zum Zeitpunkt 0 beträgt $P_0 = 1$ und es wurden Daten zu 10000 äquidistanten Zeitpunkten gemessen, also von 0 bis $(10 - 0.001)$. In Abbildung 2.7 sehen wir den idealisierten Kurvenverlauf, wie ihn die Differentialgleichung vorgeben würde. Die

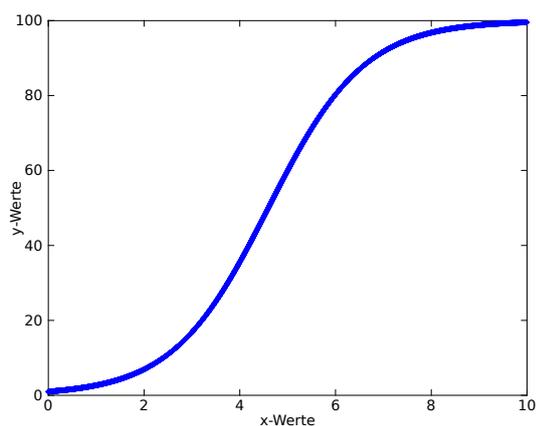


ABBILDUNG 2.7.: Kurve logistischen Wachstums

tatsächlich gemessenen Daten wurden wie folgt simuliert: Zu jedem Zeitpunkt t gilt für die Populationsgröße $P_0 \leq P(t) \leq K$. Interpretiert man den normierten Parameter

$$W(t) := \frac{P(t) - P_0}{K - P_0} \in [0, 1]$$

als Wahrscheinlichkeit zum Zeitpunkt t eine Bewegung im Lebensraum zu messen und eine Bewegung als $P(t) = K$ bzw. keine Bewegung als $P(t) = P_0$, so könnte eine mögliche Meßreihe wie in Abbildung 2.8 aussehen. Der Begriff *0-1-Messung* ist hier im Sinne „ganz

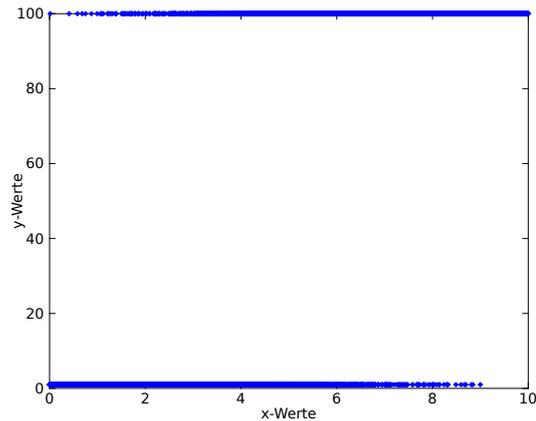


ABBILDUNG 2.8.: Datensatz einer 0-1-Messung der Population

oder gar nicht“ zu verstehen. Als Mensch erkennt man daraus recht wenig. Abgesehen von der offensichtlichen Beobachtung, daß die maximalen Meßwerte sich zu späteren Zeitpunkten häufen, kann man quantitativ recht wenig aussagen. Eine MR-Approximation hingegen läßt bereits das logistische Wachstum erahnen, wie in Abbildung 2.9 zu sehen ist.

Unsere experimentelle Erfahrung mit der MR-Methode steht im Einklang mit dem letzten, zusammenfassenden Absatz aus [Sta]: Man hat ein wirkungsvolles Mittel zur Hand, auch große Datensätze zu approximieren. Die Methode ist ziemlich allgemein und bedarf keiner genaueren Modellierung, wie sie bei parametrischen Methoden vorliegen muß. Es stehen schnelle und zugleich einfache Algorithmen zur Verfügung. Die Qualität der Ausgabe im Beispiel logistischen Wachstums leidet jedoch beträchtlich, wenn der Datensatz weniger Stützstellen enthält. Speziell im ersten Beispiel (Logarithmierung) kann man mit anderen Parametern beobachten, wie Ausreißer am Anfang des Samples zum Problem werden – diese bedürfen besonderer Aufmerksamkeit.

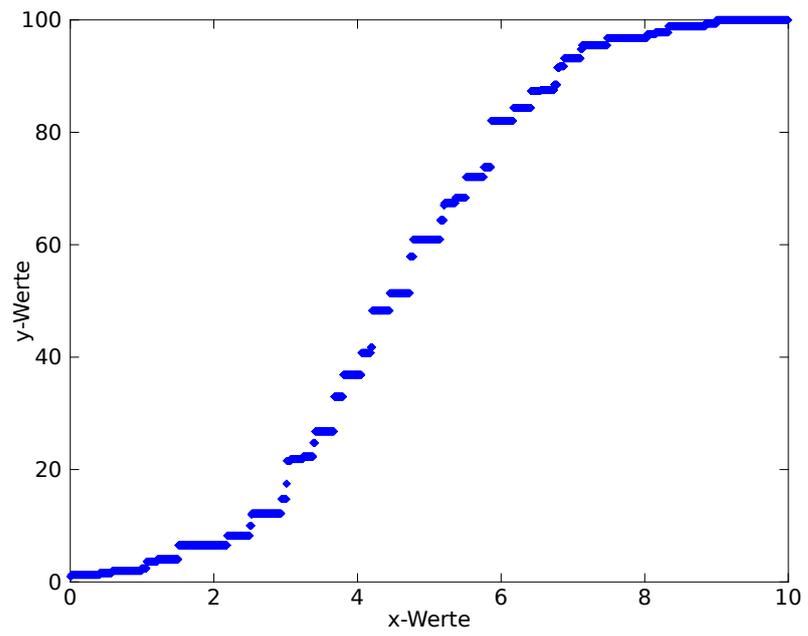


ABBILDUNG 2.9.: MR-Approximation der 0-1-Messung.

3. Ein Algorithmus zur Berechnung der Regression

Das Ziel dieses Abschnitts ist es, die Korrektheit eines Algorithmus zu beweisen, der das oben beschriebene MR-Problem in quadratischer Laufzeit löst. Er wurde uns von Herrn Günter Klützke zugetragen und findet sich als Pseudocode in Abschnitt 3.4.

Wir werden in den folgenden Unterabschnitten die Grundsteine für den Korrektheitsbeweis legen. Sie können auch als kompakte aber unvollständige Einführung in linear beschränkte, quadratische Programme gesehen werden. Sätze, Beweise und Notation entstammen überwiegend [Ree08] und [Jun99]. Andere Quellen werden an geeigneter Stelle erwähnt.

3.1. Die KKT-Bedingungen im Kontext konvexer Probleme

Im folgenden sei stets

$$\mathcal{V} = \mathbb{R}^n.$$

Wir legen die Funktionen $f : \mathcal{V} \rightarrow \mathbb{R}$, $\mathbf{h} : \mathcal{V} \rightarrow \mathbb{R}^m$ und $\mathbf{g} : \mathcal{V} \rightarrow \mathbb{R}^l$ zugrunde. Mit g_i bzw. h_j seien die Komponenten von \mathbf{g} und \mathbf{h} gemeint; die Indizes müssen sich in entsprechenden Grenzen bewegen. Alle Funktionen f, g_i, h_j werden als stetig differenzierbar angenommen; die Anwendung des *Nabla*-Operators ∇ ist also wohldefiniert. Wir betrachten folgendes Problem (\mathcal{P}):

$$\begin{aligned} (\mathcal{P}) : \quad & \min_{\mathbf{x}} : f(\mathbf{x}) \\ & \text{u. d. N.} : \quad h_j(\mathbf{x}) = 0 \quad (j = 1, \dots, m), \\ & \quad \quad g_i(\mathbf{x}) \leq 0 \quad (j = 1, \dots, l), \end{aligned} \tag{3.1.0.1}$$

Die Funktionen \mathbf{g} und \mathbf{h} dienen der Eingrenzung der Punktmenge, bezüglich der f minimiert werden soll. Man nennt f die *Zielfunktion*. Die Klausel „u.d.N.“ steht für „unter den Nebenbedingungen“. Dem aufmerksamen Leser fällt vielleicht auf, daß Gleichungsnebenbedingungen streng genommen nicht benötigt werden, zumal sie äquivalent durch Ungleichungsnebenbedingungen ersetzt werden können. Da sie konzeptionell und algorithmisch häufig anders behandelt werden, stellt man sie meist separat dar. Man kann die Gleichungen und Ungleichungen statt mit Indizes auch äquivalent und kürzer als $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ bzw. $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ aufschreiben. Dabei ist eine Ungleichung zwischen Vektoren komponentenweise zu verstehen. Die zugehörige Ordnungsrelation ist reflexiv, transitiv und antisymmetrisch – jedoch nicht trichotom. Gelegentlich findet man in

der Literatur bei den Nebenbedingungen noch die Klausel $\mathbf{x} \in \mathcal{X} \subseteq \mathcal{V}$, um noch allgemeinere Einschränkungen an \mathbf{x} zu zulassen. Davon werden wir jedoch keinen Gebrauch machen. Für die oben erwähnte Eingrenzung der Punktmenge hat sich folgende Terminologie etabliert:

Definition 3.1.1 (Zulässigkeit; Klassisches o.a. unrestringiertes Problem). Die Menge

$$\mathcal{Z}_P = \{\mathbf{x} \in \mathcal{V} \mid g(\mathbf{x}) \leq 0 \text{ und } h(\mathbf{x}) = 0\}$$

heißt *Zulässigkeitsbereich für (P)*; entsprechend heißt \mathbf{x} ein *zulässiger Punkt für (P)*, falls $\mathbf{x} \in \mathcal{Z}_P$.

Gilt $\mathcal{Z}_P = \mathbb{R}^n$, so heißt (P) *unrestringiert* oder nach [Jun99] auch *klassisch*.

Die folgenden, sogenannten Karush-Kuhn-Tucker-Bedingungen sind sehr wichtig und finden sich in vielen Büchern über Optimierung. Wir werden später den Satz 3.1.5 zitieren, der angibt unter welchen Anforderungen an f , \mathbf{g} und \mathbf{h} die KKT-Bedingungen das Optimum äquivalent charakterisieren. Zu den KKT-Bedingungen noch einige Vorbemerkungen. Normalerweise wird der Gradient einer Funktion $f : \mathcal{V} \rightarrow \mathbb{R}$ als Zeilenvektor angegeben:

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \dots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right).$$

Es gibt jedoch auch Autoren, die ihn als Spaltenvektor darstellen, und dieser Konvention wollen wir hier folgen. Der Vektorgradient hingegen, also die Jacobi-Matrix $\nabla \mathbf{F}$ einer Funktion $\mathbf{F} : \mathcal{V} \rightarrow \mathcal{V}$, hat weitgehend einheitlich die Gestalt

$$\nabla \mathbf{F}(\mathbf{x}) = \frac{\partial F_i(\mathbf{x})}{\partial x_j},$$

woran auch wir uns halten wollen. Die Verwendung desselben in den KKT-Bedingungen zielt jedoch nicht darauf ab, die lokale, lineare Approximation von \mathbf{F} in einem Punkt zu erreichen, sondern Punkte, aus dem von $\{\nabla F_i \mid i = 1 \dots n\}$ erzeugten Kegel bzw. Unterraum, als geeignete Linearkombination darzustellen. Deswegen bevorzugen wir $\nabla g(\mathbf{x})^T \boldsymbol{\mu}$ gegenüber $\boldsymbol{\mu}^T \nabla g(\mathbf{x})$ in der Formulierung der KKT-Bedingungen. In Vektorgleichungen stehen bei uns also stets *Spaltenvektoren*.

Definition 3.1.2 (\mathbf{x} ist ein KKT-Punkt). Seien f, g_i, h_j stetig differenzierbar von \mathcal{V} nach \mathbb{R} . $\mathbf{x} \in \mathbb{R}^n$ ist ein *KKT-Punkt* für (P) genau dann, wenn es Multiplikatoren $\boldsymbol{\lambda} \in \mathbb{R}^m$ und $\boldsymbol{\mu} \in \mathbb{R}^l$ gibt, so daß:

- $\mathbf{x} \in \mathcal{Z}_P$.
- Es gilt die folgende, *Gradientengleichung(en)* genannte Vektorgleichung:

$$\nabla f(\mathbf{x}) + \nabla g(\mathbf{x})^T \boldsymbol{\mu} + \nabla h(\mathbf{x})^T \boldsymbol{\lambda} = \mathbf{0},$$

bzw. mit Indizes:

$$\nabla f(\mathbf{x}) + \sum_{j=1}^m \lambda_j \nabla h_j(\mathbf{x}) + \sum_{i=1}^l \mu_i \nabla g_i(\mathbf{x}) = \mathbf{0}.$$

- $\boldsymbol{\mu} \geq 0$.
- $\langle \boldsymbol{\mu}, \mathbf{g}(\mathbf{x}) \rangle = 0$. Diese Bedingung bezeichnet man auch als *komplementären Schlupf*.

In Prosa lesen sich die KKT-Bedingungen wie folgt: \mathbf{x} muß zulässig sein, zusammen mit $\boldsymbol{\lambda}$ und positiven $\boldsymbol{\mu}$ die Gradientengleichung erfüllen, und $\boldsymbol{\mu}$ muß zusammen mit $\mathbf{g}(\mathbf{x})$ die Bedingung des komplementären Schlupfs erfüllen.

Setzt man die Vorzeichenbedingungen an $\boldsymbol{\mu}$ und $\mathbf{g}(\mathbf{x})$ als erfüllt voraus, so ist die Bedingung des komplementären Schlupfs äquivalent zu

$$\mu_i g_i(x) = 0 \quad (i = 1, \dots, l),$$

siehe z. B. [Ree08].

Definition 3.1.3. Eine Funktion $F : \mathcal{V} \rightarrow \mathbb{R}^k$ heißt *affin-linear*, g.d.w. es eine lineare Abbildung A und einen Vektor \mathbf{b} gibt, so daß

$$F(\mathbf{x}) = A(\mathbf{x}) + \mathbf{b}.$$

Sie heißt *konvex* genau dann, wenn $k = 1$ und

$$\forall \mathbf{x}, \mathbf{y} \forall \lambda \in [0, 1] : f(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1 - \lambda)f(\mathbf{y}).$$

Sie heißt *strikt konvex* genau dann, wenn $k = 1$ und

$$\forall \mathbf{x}, \mathbf{y}, \mathbf{x} \neq \mathbf{y} \forall \lambda \in (0, 1) : f(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}) < \lambda f(\mathbf{x}) + (1 - \lambda)f(\mathbf{y}).$$

Man spricht von einem *konvexen Problem*, wenn die Zielfunktion des Problems konvex ist.

Es sei jedoch darauf hingewiesen, daß selbst die Rahmenbedingung einer konvexen Zielfunktion auf einem konvexen Zulässigkeitsbereich im allgemeinen nicht die Existenz eines Minimums garantieren oder gar dessen Eindeutigkeit.

Definition 3.1.4. Im Kontext von (\mathcal{P}) nennen wir einen Punkt \mathbf{x} *optimal*, falls

$$\forall \mathbf{y} \in \mathcal{Z} : f(\mathbf{x}) \leq f(\mathbf{y}).$$

Der Punkt \mathbf{x} heißt *lokal optimal*, falls die obige Bedingung nur in einer offenen Teilmenge von \mathcal{Z} gilt.

Es läßt sich folgendes zeigen, siehe z. B. [Ree08]:

Satz 3.1.5. Ist f stetig differenzierbar und konvex, und sind sowohl \mathbf{g} als auch \mathbf{h} affin-lineare Funktionen, so ist \mathbf{x} genau dann eine optimale Lösung für (\mathcal{P}) , wenn es geeignete Multiplikatoren $\boldsymbol{\mu}$ und $\boldsymbol{\lambda}$ gibt, die \mathbf{x} zu einem KKT-Punkt machen.

3.2. Linear beschränkte, quadratische Probleme (QPs)

Im folgenden interessieren wir uns insbesondere für Probleme folgender Gestalt; sie werden *linear beschränkte, quadratische* Probleme genannt. Da für unsere Zwecke Polyeder als Zulässigkeitsbereiche ausreichen, verwenden wir einfach den kurzen Begriff „quadratisch“.

$$\begin{aligned}
 (\mathcal{QP}) \quad \min \mathbf{x} : & \quad \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{c}^T \mathbf{x} \\
 \text{u.d.N.} : & \quad \mathbf{C} \mathbf{x} - \mathbf{d} = \mathbf{0} \quad (\text{Gleichungsnebenbedingungen}) \\
 & \quad \mathbf{A} \mathbf{x} - \mathbf{b} \leq \mathbf{0} \quad (\text{Ungleichungsnebenbedingungen})
 \end{aligned}$$

Wobei \mathbf{Q} hier und im folgenden stets symmetrisch *und* positiv definit sei. Die Eigenschaft der positiven Definitheit entspricht der strikten Konvexität der Zielfunktion. Wir zitieren wieder aus [Ree08]:

Satz 3.2.1. *Ist \mathbf{Q} symmetrisch positiv definit und $\mathcal{Z}_{QP} \neq \{\}$, so besitzt (\mathcal{QP}) genau ein Optimum.*

Die folgenden Gedanken stammen aus [Ree08] bzw. [NW06] und sind dort als „Nullraum-Methode“ ausgewiesen.

Die Gleichungsnebenbedingungen von (\mathcal{QP}) drücken aus, daß \mathbf{x} aus einem durch \mathbf{C} und \mathbf{d} bestimmten, affinen Teilraum stammen muß. Hat man es nur mit solchen zu tun, so lassen sich diese, abgesehen von numerischen Schwierigkeiten, prinzipiell eliminieren, indem man durch folgende Beobachtungen zu einem neuen Problem gelangt:

\mathbf{N} sei die bijektive, lineare Einbettung von \mathbb{R}^k in den $\text{Kern}(\mathbf{C}) \subseteq \mathbb{R}^n$, mit $k \leq n$, wobei k die Dimension des Kerns sei. Der Vektor \mathbf{x}_s sei eine spezielle Lösung für $\mathbf{C}\mathbf{x}_s = \mathbf{d}$; will man dessen Existenz garantieren, kann man z. B. fordern, daß \mathbf{C} vollen Rang haben muß. Wir nehmen die Existenz von \mathbf{x}_s an – es gibt also für alle (Gleichungs-)zulässigen \mathbf{x} die Darstellung $\mathbf{x} = \mathbf{x}_s + \mathbf{N}\mathbf{v}$, wobei $\mathbf{v} \in \mathbb{R}^k$ gelten soll. Einsetzen in die Zielfunktion

$$\frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{c}^T \mathbf{x} = \frac{1}{2} (\mathbf{x}_s + \mathbf{N}\mathbf{v})^T \mathbf{Q} (\mathbf{x}_s + \mathbf{N}\mathbf{v}) + \mathbf{c}^T (\mathbf{x}_s + \mathbf{N}\mathbf{v})$$

führt auf die neue, quadratische Funktion

$$\frac{1}{2} \mathbf{v}^T (\mathbf{N}^T \mathbf{Q} \mathbf{N}) \mathbf{v} + (\mathbf{Q} \mathbf{x}_s + \mathbf{c})^T \mathbf{N} \mathbf{v} + \text{const.}$$

Wegen der Injektivität von \mathbf{N} ist $(\mathbf{N}^T \mathbf{Q} \mathbf{N})$ symmetrisch positiv definit, also insbesondere regulär, und man kann mit einem Kriterium der unrestringierten Optimierung das Optimum bestimmen. Wir führen fast wörtlich ein Kriterium aus [Jun99] an – hier sind Minima sinngemäß Optima:

Satz 3.2.2. Sei $f : V \rightarrow R$ in \mathbf{x} zwei mal stetig differenzierbar, es gelte $\nabla f(\mathbf{x}) = \mathbf{0}$, und $\mathbf{H}_f(\mathbf{x}) := \frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j}$ sei positiv definit. Dann ist \mathbf{x} ein lokales Minimum für f . Ferner gilt $\nabla f(\mathbf{x}) = \mathbf{0}$ in jedem lokalen Minimum.

Das führt in unserem Fall auf das eindeutig lösbare System $(\nabla_k(c_i n_{ij} v_j)) = c_i n_{ik} = (n^T)_{ki} c_i$

$$(\mathbf{N}^T \mathbf{Q} \mathbf{N}) \mathbf{v} = -\mathbf{N}^T (\mathbf{Q} \mathbf{x}_s + \mathbf{c}).$$

Wir sehen also: gäbe es keine Ungleichungsnebenbedingungen, so bricht das Minimierungsproblem unter Gleichungsnebenbedingungen auf ein klassisches Problem herunter. Dieser Gedanke erstreckt sich auch auf den Fall, daß die Zielfunktion ihr Minimum tatsächlich im Inneren des Zulässigkeitsbereichs annimmt. Zudem aber auch: wüßte man, welche der Ungleichungen im Optimum tatsächlich mit Gleichheit erfüllt sind, so könnte man das Problem vollständig auf ein klassisches reduzieren – und die gefundene Lösung mithilfe der KKT-Bedingungen einfach¹ als solche verifizieren. Diese vereinfachen sich aufgrund der linearen/quadratischen Gestalt des Problems. Durch einfaches Ableiten ergibt sich die Gradientengleichung zu

$$\mathbf{Q} \mathbf{x} + \mathbf{c} + \mathbf{C}^T \boldsymbol{\lambda} + \mathbf{A}^T \boldsymbol{\mu} = \mathbf{0}.$$

Die Idee, das Problem der Suche des Optimalpunkts auf die Suche der mit Gleichheit erfüllten Nebenbedingungen im Optimum abzuwälzen, wollen wir im folgenden Abschnitt weiter auf das MR-Problem spezialisieren.

3.3. Eine Vorstellung des Frameworks von Best und Chakravarti

Die Minimierung der Zielfunktion $\|\mathbf{x} - \mathbf{y}\|_2$ für das MR-Problem läßt sich über monotone Abbildungen umformen zu

$$\frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{c}^T \mathbf{x},$$

wobei in der letzte Zeile \mathbf{Q} und \mathbf{c} entsprechend

$$\begin{aligned} \mathbf{Q} &= \mathbf{1} \in \mathbb{R}^{n \times n} \\ \mathbf{c} &= -\mathbf{y} \in \mathbb{R}^n \end{aligned}$$

gewählt werden. Es sind also insbesondere die minimierenden Punkte identisch. Da es beim MR-Problem keine Gleichungsnebenbedingungen gibt, also auch keine \mathbf{C} -Matrix, beschreiben

¹Es ist bei gegebenem \mathbf{x} offensichtlich, daß es sich bei der Konstruktion der Multiplikatoren im schlimmsten Fall um ein lineares Optimierungsproblem handelt, daß mit dem Simplexalgorithmus lösbar ist. Sollte es nicht lösbar sein, so kann \mathbf{x} nach Satz 3.2.1 und 3.1.5 kein Optimum sein.

wir es der Bequemlichkeit halber als

$$\begin{aligned} \min \mathbf{x} : & \quad \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{c}^T \mathbf{x} \\ \text{u.d.N.} : & \quad \mathbf{A} \mathbf{x} \leq \mathbf{0} \end{aligned}$$

$$\mathbf{A} = \begin{pmatrix} 1 & -1 & 0 & 0 & \dots & 0 \\ 0 & 1 & -1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & -1 & 0 \\ 0 & \dots & 0 & 0 & 1 & -1 \end{pmatrix} \in \mathbb{R}^{(n-1) \times n}.$$

In der KKT-Gradientengleichung kommt also ebenfalls keine \mathbf{C} -Matrix mehr vor:

$$\mathbf{Q} \mathbf{x} + \mathbf{c} + \mathbf{A}^T \boldsymbol{\mu} = \mathbf{0}.$$

Sieht man genau hin, so erkennt man, daß der Bildraum von \mathbf{A}^T nichts anderes zu leisten vermag als Gewichte zwischen den einzelnen Vektorkomponenten zu verschieben. Der Bildraum von \mathbf{A}^T besteht einfach aus Vektoren, deren Quersumme 0 ist - anders ausgedrückt:

$$\text{Im}(\mathbf{A}^T) = \left\{ \mathbf{x} \in \mathcal{V}; \mid \mathbf{x} \circ \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = 0 \right\}.$$

Positive Multiplikatoren $\boldsymbol{\mu}$ reichen aus, wenn man nie in die falsche „Richtung“ verschieben muß.

Da \mathbf{Q} eine Diagonalmatrix ist und \mathbf{A}^T lediglich eine Nebendiagonale besitzt, kommt hinzu, daß das System der Gradientengleichungen entkoppelt, falls ein μ_i zu 0 gewählt werden kann; woher so eine Information stammen kann, werden wir später noch erklären.

Dieser Sachverhalt gestattet es, ein relativ allgemeines Active-Set-Verfahren, wie das aus [Ree08], auf unsere Bedürfnisse zuzuschneiden. Das Framework von [BC90] leistet genau das, und wir wollen es hier kurz vorstellen – alle Ideen und auch Teile der Notation stammen folglich aus [BC90].

Da man die optimale Lösung nicht kennt, also insbesondere auch nicht weiß, welche der Ungleichungen dort mit Gleichheit erfüllt sind, kann man versuchen Annahmen darüber zu machen, welche Ungleichungen dies sind. Diese Information wird vom sogenannten *active set* dargestellt. Dies ist auch bei allgemeinen Active-Set-Verfahren so. In unserem Fall wird das active set auch dahingehend interpretiert, daß Ungleichungen, die nicht im active set sind, als strikt erfüllt angenommen werden; wegen des komplementären Schlupfs also insbesondere der entsprechende Multiplikator zu 0 gesetzt werden darf. **Auf diese Art zerfällt das ursprüngliche KKT-System bei gegebenem active set in Teilsysteme.**

Das active set wird im folgenden durch eine Menge \mathcal{S} dargestellt, deren Elemente 2-Tupel sind. Angenommen eine Ungleichung wie $x_1 \leq x_2$ sei mit Gleichheit erfüllt, also $x_1 = x_2$, so gilt

$(1, 2) \in \mathcal{S}$. Den zugehörigen Multiplikator, eine Komponente von $\boldsymbol{\mu}$, bezeichnen wir je nach Kontext mit $\mu_{(1,2)}$ oder auch mit μ_1 . Die erstere Notation hebt hervor, auf welche Ungleichung sich der Multiplikator bezieht, die letztere hebt seine Wirkung in \mathbf{A}^T hervor.

Es folgen zwei Beispiele in Variablen. Der Eingabevektor sei y_1, \dots, y_6 . Das active set besteht aus den Ungleichungen für $x_1 \leq x_2, x_2 \leq x_3, x_4 \leq x_5, x_5 \leq x_6$ – drückt also die Gleichheit von x_1, x_2 und x_3 bzw. die Gleichheit von x_4, x_5 und x_6 aus. Wir schreiben hierfür $S = \{(1, 2), (2, 3), (4, 5), (5, 6)\}$.

Man beachte:

$$[(3, 4) \notin \mathcal{S}] \implies [\mu_{(3,4)} = \mu_3 = 0].$$

Das ursprüngliche KKT-System lautet hier (wir haben die μ_i bewußt auf die rechte Seite der Gleichungen geschrieben)

$$\begin{aligned} x_1 - y_1 &= -\mu_1 \\ x_2 - y_2 &= \mu_1 - \mu_2 \\ x_3 - y_3 &= \mu_2 - \mu_3 \\ x_4 - y_4 &= \mu_3 - \mu_4 \\ x_5 - y_5 &= \mu_4 - \mu_5 \\ x_6 - y_6 &= \mu_5 \\ \mu_i &\geq 0 \text{ für } i \in \{1 \dots 5\} \\ \mu_i(x_i - x_{i+1}) &= 0 \text{ für } i \in \{1 \dots 5\}. \end{aligned}$$

Die Bedingungen, die sich aus dem active set S ergeben, würden zusätzlich auf

$$\begin{aligned} x_1 &= x_2 = x_3 \\ x_4 &= x_5 = x_6 \end{aligned}$$

führen. Sie werden aber, wie gesagt, weitgehender interpretiert: Da $(3, 4) \notin S$ wird $\mu_{(3,4)} = \mu_3 = 0$ angenommen, und **die Forderung $\mu_i \geq 0$ für μ_1, μ_2 und μ_4, μ_5 wird fallen gelassen**, was auf die beiden, mit linearer Algebra lösbaren Subsysteme

$$\begin{aligned} x_1 - y_1 &= -\mu_1 \\ x_2 - y_2 &= \mu_1 - \mu_2 \\ x_3 - y_3 &= \mu_2 \\ x_1 &= x_2 = x_3 \end{aligned}$$

und

$$\begin{aligned}x_4 - y_4 &= -\mu_4 \\x_5 - y_5 &= \mu_4 - \mu_5 \\x_6 - y_6 &= \mu_5 \\x_4 &= x_5 = x_6\end{aligned}$$

führt. Diese Teilung legt es nahe, jedem active set eine Partition \mathcal{T} des Datenvektors bzw. seiner Indexmenge zu zuordnen. In unserem Fall entspricht das $\mathcal{T} = \{\{1, 2, 3\}, \{4, 5, 6\}\}$, wobei die Zahlen für Indizes der x_i bzw. y_i stehen.

Definition 3.3.1. Unter einer *Partition* \mathcal{T} einer Grundmenge \mathcal{G} verstehen wir eine Familie von disjunkten Teilmengen \mathcal{M}_i von \mathcal{G} , die \mathcal{G} überdecken.

Nun, zu welchem Ziel diese Betrachtungen? Die Addition eines solchen KKT-Teilsystems macht deutlich, daß

$$\begin{aligned}\frac{1}{3} \sum_{i=1}^3 y_i &= x_1 = x_2 = x_3 \\ \frac{1}{3} \sum_{i=4}^6 y_i &= x_4 = x_5 = x_6.\end{aligned}$$

Die entsprechenden Multiplikatoren μ_i lassen sich, eventuell aber unter Verletzung der Vorzeichenbedingungen, ebenfalls leicht berechnen. Der Sachverhalt läßt sich wie folgt ausnutzen: Ist die zusammengesetzte Lösung

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_3 \\ x_4 \\ \vdots \\ x_6 \end{pmatrix}$$

tatsächlich zulässig, d. h. $x_1 \leq x_2 \leq \dots \leq x_6$, und führen alle Teilsysteme auf positive Multiplikatoren, so kann man das KKT-Gesamtsystem mit \mathbf{x} und

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ 0 \\ \mu_4 \\ \mu_5 \end{pmatrix}$$

lösen, wobei in allgemeineren Fällen *alle* μ_i zu Null gewählt werden, die nicht zu Ungleichungen des active set gehören². D. h. man hätte in diesem Fall tatsächlich eine optimale Lösung \mathbf{x} .

²Über eben diese Komponenten macht selbst die Gesamtheit aller Teilsysteme keine Aussage. Lediglich die Interpretation des active set legt dies fest.

Die Algorithmen aus [BC90] generieren entweder eine Folge von aktiven Mengen $\mathcal{S}_0, \mathcal{S}_1, \dots$, denen man zulässige Zwischenlösungen $\mathbf{x}_0, \mathbf{x}_1 \dots$ zuordnen kann – diese werden *primale Verfahren* genannt und haben sogenannte *primal zulässige* Zwischenlösungen – oder sie generieren Zwischenlösungen mit stets positiven Multiplikatoren, diese werden *duale Verfahren* genannt und haben *dual zulässige* Zwischenlösungen. Sie können aus der Sicht eines Programmierers als Fixpunktiterationen verstanden werden, in der die primale Zulässigkeit (respektive die duale Zulässigkeit bei dualen Verfahren) eine Invariante darstellt und die duale (primale) Zulässigkeit äquivalent zur Abbruchbedingung ist. Dieser Sichtweise folgen auch die Korrektheitsbeweise aus [BC90], und wir werden den Korrektheitsbeweis des Klützke-Algorithmus in diesem Stil führen. Dabei werden wir ebenfalls das Konzept des active set \mathcal{S} gleichwertig durch das der Partition \mathcal{T} ersetzen.

3.4. Ein Divide-and-Conquer-Verfahren zur Lösung des MR-Problems

Auch in diesem Abschnitt orientiert sich die Notation und Vorgehensweise an [BC90]. Der Korrektheitsbeweis wurde bereits in [AGHK14] gebracht. Dieser Abschnitt ordnet die Resultate in das Framework von [BC90] ein.

Es folgt eine Reihe von Definitionen, die das notwendige Vokabular und elementare Sachverhalte für die folgenden Beweise bereitstellen.

Definition 3.4.1. Zu gegebener Partition \mathcal{T} von $\{1, \dots, n\}$ und einem $\mathcal{M} \in \mathcal{T}$, $\mathcal{M} = \{i, i + 1, \dots, k\}$, $i \leq k$ bezeichnen wir mit $\mathcal{U}_j(\mathcal{M}) = \{i, \dots, j\}$ und $\mathcal{O}_j(\mathcal{M}) = \{j + 1, \dots, k\}$ die entsprechenden *Unter-* und *Oberabschnitte* der Menge, falls $i \leq j < k$. Wenn aus dem Kontext klar ist, auf welche Menge Bezug genommen wird, so lassen wir den Parameter \mathcal{M} der Ober- bzw. Unterabschnitte weg. Wir bezeichnen sowohl das Paar $(j, j + 1)$, als auch j selber als *Teilungspunkt*. Mit

$$\text{mean}_y(\mathcal{M}) = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} y_i$$

bezeichnen wir das arithmetische Mittel aller Komponenten des Datenvektors \mathbf{y} , mit Indizes $i \in \mathcal{M}$. $|\mathcal{M}|$ bezeichnet die Kardinalität der Menge \mathcal{M} . Wenn \mathbf{y} aus dem Zusammenhang bekannt ist, schreiben wir auch kurz $\text{mean}(\mathcal{M})$. Zwei Mengen $\mathcal{M}, \mathcal{N} \in \mathcal{T}$ heißen *benachbart*, falls es geeignete $k \leq l \leq m$ gibt, so daß $\mathcal{M} = \{k, k + 1, \dots, l\}$ und $\mathcal{N} = \{l + 1, l + 2, \dots, m\}$ gilt. Wir schreiben hierfür

$$\mathcal{M} \triangleleft \mathcal{N}$$

und nennen \mathcal{M} den *Vorgänger* von \mathcal{N} . Notabene:

$$\mathcal{M} \triangleleft \mathcal{N}$$

$$\Leftrightarrow$$

$$\exists s : \mathcal{M} = \mathcal{U}_s(\mathcal{M} \cup \mathcal{N}) \text{ und } \mathcal{N} = \mathcal{O}_s(\mathcal{M} \cup \mathcal{N}).$$

Die beiden Mengen \mathcal{M} und \mathcal{N} müssen demnach selbst geordnet und „lückenlos“ sein. Die Relation \prec ist asymmetrisch, irreflexiv und nach unserer Definition nicht transitiv.

Zu einem gegebenen Vektor

$$\mathbf{z} = \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix} \in \mathbb{R}^n$$

bezeichnet

$$\mathbf{z}[u : v] := \begin{pmatrix} z_u \\ \vdots \\ z_v \end{pmatrix} \in \mathbb{R}^{v-u+1}$$

den entsprechenden *Teilvektor*, falls $1 \leq u \leq v \leq n$. Es gilt insbesondere $\mathbf{z}[u : v]_1 = z_u$. Anstelle von $(u : v)$ kann auch eine nichtleere Teilmenge von $\{1, \dots, n\}$ treten. Ein Beispiel wäre:

$$\begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} [\{1, 3, 5\}] = \begin{pmatrix} a \\ c \\ e \end{pmatrix}.$$

Wir werden den $[_]$ -Operator auf Vektoren linksassoziativ verwenden.

Seien \mathcal{V} und \mathcal{W} Partitionen über einer Grundmenge \mathcal{G} . \mathcal{W} heißt eine *Verfeinerung* von \mathcal{V} , falls es für jede Menge $\mathcal{N} \in \mathcal{V}$ eine Familie von Mengen $\mathcal{M}_i \in \mathcal{W}$ gibt, so daß

$$\mathcal{N} = \bigcup_{i \in I} \mathcal{M}_i.$$

Da \mathcal{V} und \mathcal{W} per definitionem Partitionen sind, wird \mathcal{N} disjunkt überdeckt.

Sei \mathcal{M} eine Menge. Mit $\mathcal{P}(\mathcal{M})$ bezeichnen wir die *Potenzmenge* von \mathcal{M} , mit $\mathcal{P}_e(\mathcal{M})$ die Menge aller *endlichen* Teilmengen von \mathcal{M} .

Das Ersetzungsaxiom, siehe z. B. [Ebb03], garantiert zu jeder Abbildung $f : \mathcal{G} \rightarrow \mathcal{H}$ eine Abbildung f^* , die auf den entsprechenden Potenzmengen operiert.

$$\begin{aligned} f^* : \mathcal{P}(\mathcal{G}) &\longrightarrow \mathcal{P}(\mathcal{H}) \\ \mathcal{M} &\longmapsto \{f(m) \mid m \in \mathcal{M}\} \end{aligned}$$

Der Operator $(_)^*$ wird auch *Potenzmengenfunctor* genannt, was wir sinngemäß aus [LB71] übersetzt haben.

Satz 3.4.2. *Seien \mathcal{V} und \mathcal{W} Partitionen über einer Grundmenge \mathcal{G} und $f : \mathcal{G} \rightarrow \mathcal{H}$ eine Bijektion. \mathcal{V} ist eine Verfeinerung von \mathcal{W} genau dann, wenn $f^{**}(\mathcal{V})$ eine Verfeinerung von $f^{**}(\mathcal{W})$ ist.*

Die folgende, rekursive Version des Klützke-Algorithmus 1 ist die erste uns bekannte. In Zeile 5 wird untersucht, an welcher Stelle der größte Unterschied zwischen linkem und rechtem „Teilmittel“ zu finden ist. Ist die entsprechende Differenz echt kleiner 0, eben diese Bedingung ist äquivalent zu

$$\text{mean}(\mathcal{U}_s) < \text{mean}(\mathcal{O}_s),$$

so wird der Eingabevektor y aufgeteilt (Zeile 8). Linke und rechte Hälfte der Eingabe können parallel optimiert werden.

Zur Suche des Teilungspunktes s : Der Operator $\arg \min(_)$ ist streng genommen nicht wohldefiniert. Wir verstehen ihn in so, daß unter allen Minima der bezüglich (\leq, \mathbb{N}) kleinste Minimalpunkt zurück gegeben wird. Im Falle $n = 1$ sei das Resultat s undefiniert (\perp) – die entsprechende Bedingung bei der if-Abfrage soll in diesem Fall mit „false“ interpretiert werden. Findet sich keine solche Differenz, so tritt Zeile 10 in Kraft und alle Komponenten von y werden durch das arithmetische Mittel ersetzt.

Der Operator $(_ \oplus _)$ in Zeile 8 verkettet zwei Vektoren wie folgt:

Definition 3.4.3. Sei

$$\mathbf{a} \in \mathbb{R}^n = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}$$

und

$$\mathbf{b} \in \mathbb{R}^m = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix},$$

so ist

$$\mathbf{a} \oplus \mathbf{b} = \begin{pmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_m \end{pmatrix} \in \mathbb{R}^{n+m}.$$

In dem Framework von [BC90] werden, wie oben erwähnt, die Algorithmen zur Lösung des MR-Problems als Fixpunktiterationen begriffen, die bestimmte Zwischenlösungen erzeugen. In der rekursiven Variante existieren diese Zwischenlösungen bzw. die zugehörigen Partitionen nur implizit auf dem Stack: Jeder Aufruf verwaltet seine eigenen lokalen Variablen, und man bräuchte schon eine Art Debugger³, um den gesamten Programmzustand zu erfassen. Wir haben uns darum entschlossen mit Algorithmus 2 eine iterative Version des Algorithmus zu geben, in der die Entstehung der Zwischenlösungen explizit sichtbar gemacht wird: Die Partitionen

³Oder man benutzt globale Variablen, die sicher nicht zur Vereinfachung beitragen würden.

Algorithmus 1 Klützke Algorithmus, rekursive Version

```
1: function RECAPPROX(y)  $\triangleright \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$ 
2:    $n \leftarrow \text{length}(\mathbf{y})$ 
3:    $\mathcal{M} \leftarrow \{1, \dots, n\}$ 
4:   init:
5:    $s \leftarrow \arg \min_{j=1 \dots (n-1)} (\text{mean}(\mathcal{U}_j(\mathcal{M})) - \text{mean}(\mathcal{O}_j(\mathcal{M})))$ 
6:   split?:
7:   if  $\text{mean}(\mathcal{U}_s) < \text{mean}(\mathcal{O}_s)$  then
8:     return RECAPPROX( $\mathbf{y}[1 : s]$ )  $\oplus$  RECAPPROX( $\mathbf{y}[s + 1 : n]$ )
9:   else
10:    return  $(\text{mean}_y(\mathcal{M}), \dots, \text{mean}_y(\mathcal{M}))^T \in \mathbb{R}^n$ 
```

\mathcal{T}_α^g und \mathcal{T}_ω^g ergeben zusammen immer eine Partition von $\{1, \dots, n\}$. In \mathcal{T}_α^g befinden sich die Mengen, in denen eine bevorstehende Teilung noch unklar ist. In \mathcal{T}_ω^g befinden sich solche, bezüglich denen die Eingabe gemittelt werden muß. Diese beiden Variablen ersetzen den Stack der rekursiven Version. Wird g hochgezählt, ist eine neue Zwischenlösung entstanden. **Wir werden in den folgenden Beweisen, mit Ausnahme der Laufzeitanalyse, stets auf die iterative Version Bezug nehmen.**

Wir geben noch eine Definition an, die es in einem mathematischen Sinne möglich macht von *dem* bestimmten Ausführungszeitpunkt g zu sprechen; die Auswahl in Zeile 7 würde in einer Implementierung randomisiert erfolgen, was wiederum einer nicht definierten Auswertungsreihenfolge der rekursiven Aufrufe in Algorithmus 1 entspricht. Die späteren Beweise sind von der konkreten Konstruktion für $\text{fix}(_)$ unabhängig; dadurch wird auch deutlich, warum sich der Algorithmus gut parallelisieren ließe.

Definition 3.4.4. Sei $\text{ord} : \mathcal{P}_e(\mathbb{N}) \rightarrow \mathbb{N}$ (irgend-) eine Bijektion. Die randomisierte Auswahl in Zeile 7 auf einer Partition \mathcal{T} von $\{1, \dots, n\}$, wobei n beliebig ist, kann man formal angeben durch

$$\text{fix}(\mathcal{T}) = \arg \min_{\mathcal{M} \in \mathcal{T}} (\text{ord}(\mathcal{M})).$$

Natürlich darf \mathcal{T} nicht leer sein. Zu gegebenem Eingabevektor \mathbf{y} bezeichnen wir mit t die Anzahl der Teilungen bis zur Terminierung; Satz 3.4.8 wird zeigen, warum t wohldefiniert ist. Die Variable g bewegt sich also im Bereich $0 \leq g \leq t$. Wir nennen g in diesem Zusammenhang einen *Zeitpunkt*. Mit \mathbf{x}^g bezeichnen wir die *Zwischenlösung*⁴, die der oben beschriebenen, zusammengesetzten Lösung aller KKT-Subsysteme der Partition $\mathcal{T}^g := \mathcal{T}_\alpha^g \cup \mathcal{T}_\omega^g$ entspricht. Die Ausgabe wird in den Beweisen also mit \mathbf{x}^t bezeichnet.

Algorithmus 2 Klützke Algorithmus, iterative Version

1: **procedure** SETAPPROX(\mathbf{y}) $\triangleright \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$

2: $n \leftarrow \text{length}(\mathbf{y})$

3: $\mathcal{T}_\alpha^0 \leftarrow \{\{1, \dots, n\}\}$

4: $\mathcal{T}_\omega^0 \leftarrow \{\}$

5: $g \leftarrow 0$

6: **repeat**

7: $\text{fix } \mathcal{M} \in \mathcal{T}_\alpha^g$

8: $\mathcal{M} = \{i, \dots, k\}$

9: *init:*

10: $s \leftarrow \arg \min_{j=i \dots (k-1)} (\text{mean}(\mathcal{U}_j(\mathcal{M})) - \text{mean}(\mathcal{O}_j(\mathcal{M})))$

11: *split?:*

12: **if** $\text{mean}(\mathcal{U}_s) < \text{mean}(\mathcal{O}_s)$ **then**

13: $\mathcal{T}_\alpha^{g+1} \leftarrow (\mathcal{T}_\alpha^g \setminus \{\mathcal{M}\}) \cup \{\mathcal{U}_s, \mathcal{O}_s\}$

14: $\mathcal{T}_\omega^{g+1} \leftarrow \mathcal{T}_\omega^g$

15: $g \leftarrow g + 1$

16: **else**

17: $\mathcal{T}_\alpha^g \leftarrow \mathcal{T}_\alpha^g \setminus \{\mathcal{M}\}$

18: $\mathcal{T}_\omega^g \leftarrow \mathcal{T}_\omega^g \cup \{\mathcal{M}\}$

19: **until** $\mathcal{T}_\alpha^g \neq \{\}$

20: Berechne \mathbf{x}^g entsprechend den KKT-Subsystemen von \mathcal{T}_ω^g ,

21: also $\forall \mathcal{M} \in \mathcal{T}_\omega^g : \mathbf{x}^g[\mathcal{M}] = (\text{mean}_y(\mathcal{M}), \dots, \text{mean}_y(\mathcal{M}))^\top$.

22: **return** \mathbf{x}^g

Wir bringen die folgenden Behauptungen ohne Beweis.

⁴Wir setzen g in den oberen Index und meinen damit keine, wie auch immer geartete *Potenz* von \mathbf{x} , sondern wir wollen im Unterschied zu den unten indizierten *Komponenten* eines Vektors auf ein *vektorwertiges* Element einer Folge hinweisen.

Behauptung 3.4.5. \mathcal{T}^{g+1} ist eine Verfeinerung von \mathcal{T}^g , dabei gilt $\mathcal{T}^{g+1} \neq \mathcal{T}^g$.

Behauptung 3.4.6. Die Menge aller Partitionen über einer endlichen Grundmenge \mathcal{G} ist selbst eine endliche Menge.

Behauptung 3.4.7. Die Verfeinerungsbeziehung ist reflexiv, transitiv und antisymmetrisch. Das zeichnet sie als Partialordnung aus.

Es folgt

Satz 3.4.8. Der Klützke-Algorithmus terminiert.

Der Begriff der Wohlordnung wird, speziell in seiner Anwendung bei Terminationsbeweisen, z. B. in [Ave95] erklärt. Einen exakteren Beweis, etwa über die Multimengenordnung, wollten wir dem Leser ersparen.

Beweis:

Aus Behauptung 3.4.6 und 3.4.7 folgt: Die Verfeinerungsbeziehung der Partitionen über $\{1, \dots, n\}$ ist eine Wohlordnung. Die Alternative in Zeile 16 kann nicht beliebig oft hintereinander ausgeführt werden, da \mathcal{T}_α^g als Teilmenge von $\mathcal{P}(\{1, \dots, n\})$ stets endlich ist. Daher terminiert der Klützke-Algorithmus. ■

Eine interessante Beobachtung ist, daß die Eindeutigkeit des Resultats hier aus der Eindeutigkeit des Optimums folgt. Das Konzept der Konfluenz, siehe z. B. [Ave95], würde es gestatten, die Eindeutigkeit auch unabhängig davon zu beweisen.

Wir werden im folgenden Satz 3.4.9 beweisen. Er sagt aus, daß alle Zwischenlösungen primal zulässig sind. Ferner zeigen wir, daß im Terminierungsfall $\mathcal{T}_\alpha^t = \{\}$ jedes KKT-Subsystem, entsprechend einem $\mathcal{M} \in \mathcal{T}_\omega^t$, positive Multiplikatoren besitzt, also eine dual zulässige Lösung und somit optimal ist. Streng genommen folgt erst daraus die Eindeutigkeit der Ausgabe bzw. die Unabhängigkeit von der konkreten Auswahlfunktion $\text{fix}(_)$.

Satz 3.4.9. Sei $\mathcal{T}^g := \mathcal{T}_\alpha^g \cup \mathcal{T}_\omega^g$ die erzeugte Folge von Partitionen und \mathbf{x}^g die entsprechende Folge von Zwischenlösungen, welche für alle g , $1 \leq g \leq t$ und für alle $\mathcal{M} \in \mathcal{T}^g$, $\mathcal{M} = \{j, \dots, l\}$, $j \leq l$ die folgenden KKT-Subsysteme

$$x_j^g - y_j = -\mu_j^g \quad (3.4.9.1a)$$

$$x_{j+1}^g - y_{j+1} = \mu_j^g - \mu_{j+1}^g \quad (3.4.9.1b)$$

$$\vdots \quad (3.4.9.1c)$$

$$x_l^g - y_l = \mu_{l-1}^g \quad (3.4.9.1d)$$

$$x_j^g = x_{j+1}^g = \dots = x_l^g \quad (3.4.9.1e)$$

$$\mu_l^g = 0 \text{ oder im Fall } l = n \text{ nicht vorhanden.} \quad (3.4.9.1f)$$

lösen. Dann gilt: \mathbf{x}^g ist zulässig für beliebige $0 \leq g \leq t$.

Die Gleichung (3.4.9.1e) impliziert, wie gesagt, $x_k = \text{mean}(\mathcal{M})$ für alle $j \leq k \leq l$. Was wir also zeigen wollen ist, daß für alle $\mathcal{M}, \mathcal{N} \in \mathcal{T}^g$ gilt:

$$\mathcal{M} \prec \mathcal{N} \implies \text{mean}(\mathcal{M}) \leq \text{mean}(\mathcal{N})$$

Dies ist klar, falls \mathcal{M} und \mathcal{N} unmittelbar nach der ersten Teilung erzeugt wurden. Die Frage ist, warum die steigenden Mittelwerte über den Verlauf des Algorithmus eine Invariante darstellen.

Definition 3.4.10. Wir nennen folgende Eigenschaft die *Invariante* des Klützke-Algorithmus:

Seien $\mathcal{T}^0 = \{\{1, \dots, n\}\}$, \mathbf{y} und s als der erste Teilungspunkt gegeben; also sei

$$\mathcal{T}^1 = \{\{1, \dots, s\}, \{s+1, \dots, n\}\}.$$

Für jede weitere Verfeinerung $\mathcal{T}^g = \{\mathcal{M}_1, \dots, \mathcal{M}_v, \mathcal{M}_{v+1}, \dots, \mathcal{M}_h\}$ von \mathcal{T}^1 mit $g > 1$, $\mathcal{M}_i \prec \mathcal{M}_{i+1}$ für $i \in \{1, \dots, h-1\}$ und $\sum_{i=1}^v |\mathcal{M}_i| = s$ gilt:

$$\text{mean}(\mathcal{M}_v) \leq \text{mean}(\mathcal{M}_{v+1}) \quad (3.4.10.1)$$

Behauptung 3.4.11. *Die Invariante gilt.*

Beweis:

Sei \mathbf{y} beliebig aber fix. Gibt es keinen Teilungspunkt, so ist die *Invariante* leer erfüllt.

Sei also s der erste Teilungspunkt. Jeder Partition $\mathcal{T} = \{\mathcal{M}_1, \dots, \mathcal{M}_h\}$ von $\{1, \dots, n\}$ kann man eine Partition des Eingabevektors zuordnen:

$$\mathbf{y} = \underbrace{(y_1, \dots, y_{\lambda_1})}_{\lambda_1 \text{ Einträge}} \underbrace{(y_{\lambda_1+1}, \dots, y_{\lambda_1+\lambda_2})}_{\lambda_2 \text{ Einträge}} \cdots \underbrace{(y_{n-\lambda_h+1}, \dots, y_n)}_{\lambda_h \text{ Einträge}}^T.$$

Es gilt also

$$|\mathcal{M}_i| = \lambda_i.$$

Des weiteren besitzt das arithmetische Mittel die Gewichtungseigenschaft

$$\text{mean}\left(\bigcup_{i=1}^h \mathcal{M}_i\right) = \frac{\sum_{i=1}^h |\mathcal{M}_i| \text{mean}(\mathcal{M}_i)}{\sum_{i=1}^h |\mathcal{M}_i|}. \quad (3.4.11.1)$$

Sei also

$$\begin{aligned} \mathcal{T}^0 &= \{\{1, \dots, n\}\}, \\ \mathcal{T}^1 &= \{\{1, \dots, s\}, \{s+1, \dots, n\}\}, \\ \mathcal{T}^g &= \{\mathcal{M}_1, \dots, \mathcal{M}_v, \mathcal{M}_{v+1}, \dots, \mathcal{M}_h\}, \end{aligned}$$

$g > 1$, $\mathcal{M}_i \triangleleft \mathcal{M}_{i+1}$ für $i \in \{1, \dots, h-1\}$ und $\sum_{i=1}^v |\mathcal{M}_i| = s$. Falls $g > 1$ nicht existiert, ist die *Invariante* wieder leer erfüllt. Wir benützen die Abkürzungen

$$\begin{aligned} M_s^l &= \text{mean}\left(\bigcup_{i=1}^v \mathcal{M}_i\right) = \frac{1}{s} \sum_{i=1}^s y_i, \\ M_s^r &= \text{mean}\left(\bigcup_{i=v+1}^h \mathcal{M}_i\right) = \frac{1}{n-s} \sum_{i=(n-s)+1}^n y_i, \\ z_j &= \text{mean}(\mathcal{M}_j) \end{aligned}$$

und verweisen nochmals auf die wichtige Eigenschaft (3.4.11.1) des arithmetischen Mittels. Zeile 10 des Klützke-Algorithmus garantiert uns, daß die Differenz $M_s^l - M_s^r$ minimal unter allen anderen ist, d. h. insbesondere im Fall $v \leq (h-2)$ gilt:

$$\begin{aligned} M_s^l - M_s^r &\leq M_{s+\lambda_{v+1}}^l - M_{s+\lambda_{v+1}}^r \\ &= \text{mean}\left(\bigcup_{i=1}^{v+1} \mathcal{M}_i\right) - \text{mean}\left(\bigcup_{i=v+2}^h \mathcal{M}_i\right) \\ &= \frac{sM_s^l + \lambda_{v+1}z_{v+1}}{s + \lambda_{v+1}} - \frac{(n-s)M_s^r - \lambda_{v+1}z_{v+1}}{(n-s) - \lambda_{v+1}} \end{aligned}$$

und im Fall $v \geq 2$:

$$\begin{aligned} M_s^l - M_s^r &\leq M_{s-\lambda_v}^l - M_{s-\lambda_v}^r \\ &= \text{mean}\left(\bigcup_{i=1}^{v-1} \mathcal{M}_i\right) - \text{mean}\left(\bigcup_{i=v}^h \mathcal{M}_i\right) \\ &= \frac{sM_s^l - \lambda_v z_v}{s - \lambda_v} - \frac{(n-s)M_s^r + \lambda_v z_v}{(n-s) + \lambda_v}. \end{aligned}$$

Liegen beide Fälle gleichzeitig vor, so können wir $z_v < z_{v+1}$ mit einem Lemma beweisen. Die Randfälle $v = 1$ und $v = h-1$ müssen getrennt betrachtet werden. Das Lemma und beide Beweise stammen aus [AGHK14]. Das verkürzende Lemma ist eine Idee von Herrn Professor Dr. Winfried Hochstättler, die wir dankbar angenommen haben.

Lemma 3.4.12. 1. Für $v \geq 2$ gilt:

$$\frac{M_s^l - z_v}{s - \lambda_v} + \frac{M_s^r - z_v}{n - s + \lambda_v} \geq 0 \quad (3.4.12.1)$$

2. Für $v \leq h-2$ gilt:

$$\frac{z_{v+1} - M_s^l}{s + \lambda_{v+1}} + \frac{z_{v+1} - M_s^r}{n - s - \lambda_{v+1}} \geq 0. \quad (3.4.12.2)$$

Beweis:

Wir erinnern daran, daß sich zum Zeitpunkt 0 die Differenz $M_s^l - M_s^r$ als minimal gegenüber allen anderen Differenzen heraus gestellt hat.

1. Fall:

$$\begin{aligned}
M_s^l - M_s^r &\leq M_{s-\lambda_v}^l - M_{s-\lambda_v}^r \\
&= \frac{sM_s^l - \lambda_v z_v}{s - \lambda_v} - \frac{(n-s)M_s^r + \lambda_v z_v}{n-s + \lambda_v} \\
&= M_s^l + \lambda_v \frac{M_s^l - z_v}{s - \lambda_v} - M_s^r + \lambda_v \frac{M_s^r - z_v}{n-s + \lambda_v} \\
\iff 0 &\leq \frac{M_s^l - z_v}{s - \lambda_v} + \frac{M_s^r - z_v}{n-s + \lambda_v}.
\end{aligned}$$

2. Fall:

$$\begin{aligned}
M_s^l - M_s^r &\leq M_{s+\lambda_{v+1}}^l - M_{s+\lambda_{v+1}}^r \\
&= \frac{sM_s^l + \lambda_{v+1} z_{v+1}}{s + \lambda_{v+1}} - \frac{(n-s)M_s^r - \lambda_{v+1} z_{v+1}}{n-s - \lambda_{v+1}} \\
&= M_s^l + \lambda_{v+1} \frac{z_{v+1} - M_s^l}{s + \lambda_{v+1}} - M_s^r + \lambda_{v+1} \frac{z_{v+1} - M_s^r}{n-s - \lambda_{v+1}} \\
\iff 0 &\leq \frac{z_{v+1} - M_s^l}{s + \lambda_{v+1}} + \frac{z_{v+1} - M_s^r}{n-s - \lambda_{v+1}}.
\end{aligned}$$

■

Wir nehmen also $2 \leq v \leq h-2$ an. Multipliziert man (3.4.12.1) und (3.4.12.2) jeweils mit dem gemeinsamen Nennern und addiert die Ungleichungen, so kommen wir auf:

$$\begin{aligned}
&(M_s^l - z_v)(n-s + \lambda_v) + (M_s^r - z_v)(s - \lambda_v) \\
&+ (z_{v+1} - M_s^l)(n-s - \lambda_{v+1}) + (z_{v+1} - M_s^r)(s + \lambda_{v+1}) \geq 0 \\
\iff &\underbrace{(\lambda_v + \lambda_{v+1})(M_s^l - M_s^r)}_{<0} + n(z_{v+1} - z_v) \geq 0.
\end{aligned}$$

Also gilt

$$(z_v - z_{v+1}) \leq (M_s^l - M_s^r) \frac{(\lambda_v + \lambda_{v+1})}{n} < 0.$$

Es bleiben die Fälle $v = 1$ bzw. $v = (h-1)$ zu zeigen. Sei also $v = 1$, was $z_v = M_s^l$ und $\lambda_v = s$

impliziert:

$$\begin{aligned}
z_v - M_s^r &\leq \frac{s z_v + \lambda_{v+1} z_{v+1}}{s + \lambda_{v+1}} - \frac{(n-s) M_s^r - \lambda_{v+1} z_{v+1}}{(n-s) - \lambda_{v+1}} \\
z_v - M_s^r &\leq z_v \frac{s}{s + \lambda_{v+1}} + z_{v+1} \left(\frac{\lambda_{v+1}}{s + \lambda_{v+1}} + \frac{\lambda_{v+1}}{(n-s) - \lambda_{v+1}} \right) \\
&\quad - M_s^r \left(\frac{n-s}{(n-s) - \lambda_{v+1}} \right) \\
z_v \left(\frac{\lambda_{v+1}}{s + \lambda_{v+1}} \right) &\leq - M_s^r \left(\frac{\lambda_{v+1}}{(n-s) - \lambda_{v+1}} \right) \\
&\quad + z_{v+1} \left(\frac{\lambda_{v+1} n}{(s + \lambda_{v+1})((n-s) - \lambda_{v+1})} \right) \\
z_v(n - (s + \lambda_{v+1})) - n z_{v+1} &\leq - M_s^r (s + \lambda_{v+1}) \\
n z_v - n z_{v+1} &\leq (z_v - M_s^r)(s + \lambda_{v+1}) \\
z_v - z_{v+1} &\leq \underbrace{(z_v - M_s^r)}_{< 0} \frac{(s + \lambda_{v+1})}{n} < 0
\end{aligned}$$

Im Fall $v = h - 1$ kann man $z_v - z_{v+1} \leq 0$ auf analoge Art beweisen.

Es ist möglich auf Lemma 3.4.12 zu verzichten und wie im Fall $v = 1$ einfach in die nahe liegende Ungleichung einzusetzen, was jedoch auf einen längeren Beweis führt. Sowohl das Lemma als auch der letzte Beweis sind von Dualität geprägt. Man kann die Beweise sozusagen „halbieren“, indem man den Isomorphismus

$$\begin{aligned}
\eta : \mathcal{V} &\longrightarrow \mathcal{V} \\
(v_1, \dots, v_n)^T &\longmapsto (-v_n, \dots, -v_1)^T
\end{aligned}$$

betrachtet. Dieser kann unter Umständen nützlich sein, doch hier würde die genaue Einführung der Notation letzten Endes keinen Platz sparen. ■

Behauptung 3.4.13. *Gilt die Invariante, so ist \mathbf{x}^g zulässig für alle $0 \leq g \leq t$.*

Die *Invariante* ist eine statische Eigenschaft. Sie sagt lediglich aus, daß Teilmengen, die sich links oder rechts des ersten Teilungspunktes bilden, steigende Mittelwerte haben. Der Beweis von Behauptung 3.4.13 wird zeigen, warum daraus die Zulässigkeit aller \mathbf{x}^g folgt.

Beweis:

Wir führen den Beweis durch vollständige Induktion nach g .

Die Fälle $g = 0$ und $g = 1$:

Im Fall $g = 0$ gibt es nur ein KKT-Subsystem. Aus $x_1 = \dots = x_n$ folgt trivialerweise $x_1 \leq \dots \leq x_n$. Im Fall $g = 1$ gibt es zwei KKT-Subsysteme, aus denen $x_1 = \dots = x_s$ und $x_{s+1} = \dots = x_n$ folgt. Zeile 12 garantiert dann $x_s \leq x_{s+1}$.

Der Fall (\mathbf{x}^g ist zulässig $\implies \mathbf{x}^{g+1}$ ist zulässig):

Sei nun $g > 1$, $\mathcal{T}^g = \{\mathcal{M}_1, \dots, \mathcal{M}_v, \dots, \mathcal{M}_h\}$ und $\mathcal{T}^{g+1} = \{\mathcal{M}_1, \dots, \mathcal{M}_{v_a}, \mathcal{M}_{v_b}, \dots, \mathcal{M}_h\}$. Beide Partitionen seien bezüglich \triangleleft geordnet, also insbesondere $\mathcal{M}_{v_a} = \mathcal{U}_s(\mathcal{M}_v)$ und $\mathcal{M}_{v_b} = \mathcal{O}_s(\mathcal{M}_v)$ für geeignetes s . Wegen der Bedingung in Zeile 12 gilt $\text{mean}(\mathcal{M}_{v_a}) \leq \text{mean}(\mathcal{M}_{v_b})$. Daraus folgt: $\mathbf{x}^{g+1}[\mathcal{M}_v]$ ist zulässig. Es bleibt also $\text{mean}(\mathcal{M}_{v-1}) \leq \text{mean}(\mathcal{M}_{v_a})$ und $\text{mean}(\mathcal{M}_{v_b}) \leq \text{mean}(\mathcal{M}_{v+1})$ zu zeigen, da die Zwischenlösung \mathbf{x}^g wegen der Induktionshypothese als zulässig vorausgesetzt werden darf und in den Komponenten aus $\mathcal{M}_i \neq \mathcal{M}_v$ mit \mathbf{x}^{g+1} übereinstimmt:

$$\forall \mathcal{M} \neq \mathcal{M}_v : \mathbf{x}^g[\mathcal{M}] = \mathbf{x}^{g+1}[\mathcal{M}].$$

Da \mathcal{T}^{g+1} eine Verfeinerung von \mathcal{T}^0 ist, gibt es Zeitpunkte $a, b < g$ und Mengen $\mathcal{X} \in \mathcal{T}^a$, $\mathcal{Y} \in \mathcal{T}^b$ mit $(\mathcal{M}_{v-1} \cup \mathcal{M}_v) \subseteq \mathcal{X}$ und $(\mathcal{M}_v \cup \mathcal{M}_{v+1}) \subseteq \mathcal{Y}$, zu denen \mathcal{M}_v und \mathcal{M}_{v+1} respektive \mathcal{M}_{v-1} und \mathcal{M}_v durch Teilungspunkte getrennt wurden. Wir führen hier nur den Beweis für

$$\text{mean}(\mathcal{M}_{v-1}) \leq \text{mean}(\mathcal{M}_{v_a})$$

aus. Der Fall $\text{mean}(\mathcal{M}_{v_b}) \leq \text{mean}(\mathcal{M}_{v+1})$ wird analog bewiesen.

Hat \mathcal{M}_v keinen Vorgänger, so ist nichts zu zeigen. Sei also

$$\mathcal{X} = \mathcal{M}_u \cup \dots \cup \mathcal{M}_{v-1} \cup \mathcal{M}_v \cup \dots \cup \mathcal{M}_w$$

wobei die \mathcal{M}_i in \triangleleft -totalgeordneter Reihenfolge aufgeführt sind. Die Abbildung m sei die monotone Bijektionen von \mathcal{X} auf $\{1, \dots, |\mathcal{X}|\}$. Ist das kleinste Element in \mathcal{M}_u etwa c , so ist

$$m(x) = x - c + 1. \tag{3.4.13.1}$$

Da

$$\{\mathcal{M}_u, \dots, \mathcal{M}_{v-1}, \mathcal{M}_{v_a}, \mathcal{M}_{v_b}, \dots, \mathcal{M}_w\}$$

eine Verfeinerung von $\{\mathcal{X}\}$ ist, so gilt wegen Satz 3.4.2, daß

$$\{m^*(\mathcal{M}_u), \dots, m^*(\mathcal{M}_{v-1}), m^*(\mathcal{M}_{v_a}), m^*(\mathcal{M}_{v_b}), \dots, m^*(\mathcal{M}_w)\}$$

eine Verfeinerung von $\mathcal{T}'^0 = m^{**}(\{\mathcal{X}\})$ ist. Die Anwendung der Invariante auf \mathcal{T}'^0 und den Vektor $\mathbf{y}' = \mathbf{y}[\mathcal{X}]$ führt auf

$$\text{mean}_{\mathbf{y}'}(m^*(\mathcal{M}_{v-1})) \leq \text{mean}_{\mathbf{y}'}(m^*(\mathcal{M}_{v_a}))$$

und damit zu

$$\text{mean}_{\mathbf{y}}(\mathcal{M}_{v-1}) \leq \text{mean}_{\mathbf{y}}(\mathcal{M}_{v_a}),$$

was zu zeigen war. Die Gleichheit der entsprechenden Mittel ist konzeptionell leicht zu verstehen.

Etwas aufwendig kann man jedoch auch formal zeigen, daß

$$\begin{aligned}
\text{mean}_y(\mathcal{M}_{v-1}) &= \frac{1}{|\mathcal{M}_{v-1}|} \sum_{i \in \mathcal{M}_{v-1}} y_i \\
&= \frac{1}{|\mathcal{M}_{v-1}|} \sum_{i=1}^{|\mathcal{M}_{v-1}|} \mathbf{y}[\mathcal{M}_{v-1}]_i \\
&= \frac{1}{|m^*(\mathcal{M}_{v-1})|} \sum_{i=1}^{|m^*(\mathcal{M}_{v-1})|} \mathbf{y}[\mathcal{M}_{v-1}]_i \\
&= \frac{1}{|m^*(\mathcal{M}_{v-1})|} \sum_{i=1}^{|m^*(\mathcal{M}_{v-1})|} \mathbf{y}[\mathcal{X}][m^*(\mathcal{M}_{v-1})]_i \\
&= \frac{1}{|m^*(\mathcal{M}_{v-1})|} \sum_{i \in m^*(\mathcal{M}_{v-1})} \mathbf{y}[\mathcal{X}]_i \\
&= \text{mean}_{y'}(m^*(\mathcal{M}_{v-1}))
\end{aligned}$$

bzw. die analoge Aussage für \mathcal{M}_{v_a} , was wir nicht ausführen. ■

Beweis von Satz 3.4.9:

Aus Behauptung 3.4.11 und 3.4.13 folgt die Gültigkeit von Satz 3.4.9. ■

Es geht nun nur noch darum, die duale Zulässigkeit von \mathbf{x}^t zu beweisen. Da im Terminationsfall $\mathcal{T}_\alpha = \{\}$ gilt, also alle $\mathcal{M} \in \mathcal{T}_\omega$ sind, verwenden wir, um der besseren Lesbarkeit willen, ab hier \mathbf{x} statt $\mathbf{x}^t[\mathcal{M}]$ und gehen analog bei $\boldsymbol{\mu}$ vor. Die folgende Behauptung findet sich, samt Induktionsbeweis und sogar etwas allgemeiner, in [BC90]. Dort wurde die Zielfunktion nicht umgeformt, weswegen die Multiplikatoren dort doppelt so groß sind. Die hier verwendete Form und ihr Beweis stammen bis auf das Vorzeichen der Multiplikatoren aus [AGHK14].

Behauptung 3.4.14. *Angenommen es gilt*

$$\{1, \dots, k\} = \mathcal{M} \in \mathcal{T}_\omega$$

also

$$(\text{mean}(\mathcal{U}_i) - \text{mean}(\mathcal{O}_i)) \geq 0.$$

für alle $i \in \{1, \dots, k-1\}$, so löst der Vektor $\boldsymbol{\mu}$ mit

$$\mu_i = \left(\sum_{j=1}^i y_j \right) - \frac{i}{k} \sum_{j=1}^k y_j \geq 0$$

die Gradientengleichung des KKT-Teilsystems für \mathcal{M} .

Beweis:

Wir verwenden Induktion nach i .

1. $i = 1$:

Wegen $x_1 = \frac{1}{k} \sum_{j=1}^k y_j$ liefert die erste Gradientengleichung im KKT-Teilsystem für \mathcal{M}

$$\mu_1 = y_1 - \frac{1}{k} \sum_{j=1}^k y_j.$$

2. Angenommen die Induktionshypothese gilt für i , so auch für $i + 1$, falls $i + 1 < (k - 1)$.

Wir betrachten die Gradientengleichung $i + 1$ im KKT-Teilsystem für \mathcal{M} :

$$\begin{aligned} \mu_{i+1} &= y_{i+1} - \frac{1}{k} \sum_{j=1}^k y_j + \mu_i \\ &= y_{i+1} - \frac{1}{k} \sum_{j=1}^k y_j + \sum_{j=1}^i y_j - \frac{i}{k} \sum_{j=1}^k y_j \\ &= \sum_{j=1}^{i+1} y_j - \frac{i+1}{k} \sum_{j=1}^k y_j. \end{aligned}$$

Die Gültigkeit der letzten Gleichung läßt sich ebenfalls nachrechnen:

$$\begin{aligned} x_k - y_k &= \frac{1}{k} \sum_{j=1}^k y_j - y_k \\ &= \frac{1}{k} \sum_{j=1}^{k-1} y_j - \frac{k-1}{k} y_k \\ &= \sum_{j=1}^{k-1} y_j - \frac{k-1}{k} \sum_{j=1}^k y_j \\ &= \mu_{k-1} \end{aligned}$$

Es bleibt der Aspekt

$$\mu_i \geq 0$$

zu beweisen:

$$\begin{aligned}
0 &\leq \text{mean}(\mathcal{U}_i) - \text{mean}(\mathcal{O}_i) \\
&= \frac{1}{i} \sum_{j=1}^i y_j - \frac{1}{k-i} \sum_{j=i+1}^k y_j \\
&= \frac{1}{i(k-i)} \left((k-i) \sum_{j=1}^i y_j - i \sum_{j=i+1}^k y_j \right) \\
&= \frac{1}{i(k-i)} \left(k \sum_{j=1}^i y_j - i \sum_{j=1}^k y_j \right) \\
&= \frac{k}{i(k-i)} \left(\sum_{j=1}^i y_j - \frac{i}{k} \sum_{j=1}^k y_j \right) \\
&= \frac{k}{i(k-i)} \mu_i
\end{aligned}$$

.

Die Einschränkung der Indizes für \mathcal{M} in der Form

$$\{1, \dots, k\} = \mathcal{M}$$

stellt keine Einschränkung der Allgemeinheit dar, aber sie macht den Beweis deutlich lesbarer. Da bereits alle \mathbf{x}^g , insbesondere auch \mathbf{x}^t , als (primal) zulässig erkannt wurden, folgt

Korollar 3.4.15. \mathbf{x}^t ist dual und primal zulässig, also optimal.

Wir schließen mit dem folgenden

Satz 3.4.16. Die Laufzeit des Klützke-Algorithmus in der rekursiven Version liegt in der Klasse $O(n^2)$.

Beweis:

Bei jedem rekursiven Aufruf auf einem Teilvektor der Größe n werden $n - 1$ Differenzen berechnet. Bildet man erst die Summen und mittelt anschließend die einzelnen Summanden so wären das $2(n - 1)$ Additionen, ebenso viele Multiplikationen und $n - 1$ Subtraktionen. Anschließend wird unter diesen Differenzen in $n - 2$ Vergleichen das Minimum ermittelt. Insgesamt Bedarf dieser Schritt also $O(n)$ Operationen. Der Terminierungsfall würde zum Abbruch des Verfahrens und einer weiteren Mittelung in $O(n)$ Schritten führen, will man die Ausgabe explizit erzeugen. Da es bei n Elementen höchstens $n - 1$ Teilungspunkte gibt, wird der Algorithmus höchstens $n - 1$ mal aufgerufen. Jeder dieser Aufrufe kann in $O(n)$ Operationen abgearbeitet werden.

$$cn * (n - 1) \in O(n^2)$$

3.5. Der „Pool Adjacent Violators Algorithm“

Es gibt mehrere Versionen des *pool adjacent violators algorithm*, den wir fortan einfach mit *PAV-Algorithmus* abkürzen werden. In [BC90] wird eine Version vorgestellt, die mit Mengen arbeitet. Sie unterscheidet sich von der hier beschriebenen Version lediglich in einer systematischeren Auswahl der zu untersuchenden Mengen. In der Arbeit [GW84] wird ausführlich dargestellt, wie man von der Version aus [BC90] zu einer Implementierung mit linearer Laufzeit gelangt. Die Kernidee ist, während der Ausführung nicht alle Komponenten des Eingabevektors zu verwalten, sondern lediglich die Positionen, Mittelwerte und Gewichte der einzelnen Blöcke. Die eigentliche Ausgabe wird erst am Ende konstruiert.

Wir werden hier eine Version des PAV-Algorithmus darstellen, wie sie die Originalarbeit [ABERS55] nahe legt. Unser Augenmerk liegt darauf, ein duales Verfahren didaktisch klug vorzustellen. Der Kontext im Original hat zwar weniger mit konvexer Optimierung und mehr mit Maximum-Likelihood-Schätzern zu tun, die prinzipielle Funktionsweise des Algorithmus läßt sich dort jedoch binnen der ersten 11 Zeilen verstehen. Des weiteren wird dort die Reihenfolge der Zusammenfassung offen gelassen, was konzeptionell eine gewisse Ähnlichkeit zu unserer „randomisierten“ Mengenversion des Klützke-Algorithmus darstellt.

In einer moderneren Arbeit wie [Kea06] wird diese Unabhängigkeit schließlich in einem parallelisierten Verfahren ausgenutzt. Die Unterteilung a priori funktioniert nicht ganz so elegant wie beim Klützke-Algorithmus, dafür aber effizient. Anthony J. Kearsley schrieb in dieser Arbeit, die eine mögliche Anwendung der monotonen Regression auf Daten von Massenspektrometern diskutiert:

„At present, it is difficult to foresee applications involving isotonic regressions on data sets so large that the absolute savings in time will warrant parallel computation. Perhaps that day will come; for now, our primary interest in parallelizing isotonic regression is for the pedagogical value of so doing. In our view, isotonic regression is a remarkably simple and elegant example of a problem for which mathematical theory virtually guarantees that parallelization will be beneficial.“

Wir wenden uns nun dem PAV-Algorithmus in Abbildung 3 zu. Zeile 5 kann man konzeptionell ähnlich verstehen wie die *fix*-Auswahl im Klützke-Algorithmus. Es gibt jedoch einige Unterschiede: Das Prädikat, das diese Auswahl einschränkt, besteht hier nicht aus einer simplen Einschränkung wie $\mathcal{M} \in \mathcal{T}_\alpha^g$ – es muß vielmehr

$$\mathcal{M}, \mathcal{N} \in \mathcal{T}^g : \mathcal{M} \triangleleft \mathcal{N}, \text{mean}(\mathcal{M}) \geq \text{mean}(\mathcal{N}) \quad (3.5.0.1)$$

gelten, und es wird nicht eine Menge ausgewählt, sondern zwei. Man könnte die Bedingung (3.5.0.1) auch algorithmisch explizit mit einer *if*-Abfrage erzwingen, allerdings scheint uns das Prädikat an dieser Stelle prägnanter.

Wir verwenden die Begriffe des Ausführungszeitpunkts $0 \leq g \leq t$ analog zum vorigen Abschnitt. Gleiches gilt für die Zwischenlösungen \mathbf{x}^g und den Zeitpunkt t der Terminierung.

Algorithmus 3 Pool Adjacent Violators Algorithm

1: **procedure** PAVA(\mathbf{y}) $\triangleright \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$

2: $n \leftarrow \text{length}(\mathbf{y})$

3: $\mathcal{T}^0 \leftarrow \{\{1\}, \dots, \{n\}\}$

4: $g \leftarrow 0$

5: **while** $\exists \mathcal{M}, \mathcal{N} \in \mathcal{T}^g : \mathcal{M} \triangleleft \mathcal{N}, \text{mean}(\mathcal{M}) \geq \text{mean}(\mathcal{N})$ **do**

6: $\mathcal{T}^{g+1} \leftarrow (\mathcal{T}^g \setminus \{\mathcal{M}, \mathcal{N}\}) \cup \{\mathcal{M} \cup \mathcal{N}\}$

7: $g \leftarrow g + 1$

8: Berechne \mathbf{x}^g entsprechend den KKT-Subsystemen von \mathcal{T}^g ,

9: also $\forall \mathcal{M} \in \mathcal{T}^g : \mathbf{x}^g[\mathcal{M}] = \text{mean}_y(\mathcal{M})$.

10: **return** \mathbf{x}^g

Die Eingabe wird wieder mit \mathbf{y} bezeichnet.

Wir beginnen mit einer abgewandelten, konkretisierten Aussage aus [BC90] und einem Korollar aus gleicher Quelle. Wie in Abschnitt 3.4 erwähnt, sind unsere Multiplikatoren anders normiert.

Lemma 3.5.1. *Innerhalb eines Blocks*

$$\mathcal{M} = \{m, \dots, n\}$$

gilt für die Multiplikatoren $\mu_s, s \in \{m, \dots, n-1\}$

$$\begin{aligned} \mu_s &= (s - m + 1) \left(\frac{1}{s - m + 1} \sum_{i=m}^s y_i - \frac{1}{n - m + 1} \sum_{i=m}^n y_i \right) \\ &= (s - m + 1) (\text{mean}(\mathcal{U}_s) - \text{mean}(\mathcal{M})) \end{aligned} \tag{3.5.1.1}$$

$$\begin{aligned} &= (n - s) \left(\frac{1}{n - m + 1} \sum_{i=m}^n y_i - \frac{1}{n - s} \sum_{i=s+1}^n y_i \right) \\ &= (n - s) (\text{mean}(\mathcal{M}) - \text{mean}(\mathcal{O}_s)) \end{aligned} \tag{3.5.1.2}$$

$$\begin{aligned} &= \frac{(s - m + 1)(n - s)}{n - m + 1} \left(\frac{1}{s - m + 1} \sum_{i=m}^s y_i - \frac{1}{n - s} \sum_{i=s+1}^n y_i \right) \\ &= \frac{(s - m + 1)(n - s)}{n - m + 1} (\text{mean}(\mathcal{U}_s) - \text{mean}(\mathcal{O}_s)) \end{aligned} \tag{3.5.1.3}$$

Beweis:

Die erste Gleichung erhält man, indem man einen geeigneten Isomorphismus, wie $m(_)$ aus (3.4.13.1), auf Behauptung 3.4.14 anwendet. Ferner kann man alle drei Terme leicht umformen zu

$$\sum_{i=m}^s y_i - \frac{s - m + 1}{n - m + 1} \sum_{i=m}^n y_i,$$

was deren Äquivalenz beweist. ■

Korollar 3.5.2. Sei \mathcal{T} eine Partition, $\mathcal{M} \in \mathcal{T}$,

$$\mathcal{M} = \{m, \dots, n\} \in \mathcal{T}$$

und $\mathbf{x}[\mathcal{M}]$ der zugehörige Teilvektor einer Zwischenlösung. Der Teilvektor $\mathbf{x}[\mathcal{M}]$ besitzt positive Multiplikatoren genau dann, wenn eine und damit jede der drei Differenzen aus (3.5.1.1), (3.5.1.2) bzw. (3.5.1.3) positiv sind. Wenn also eine der drei äquivalenten Bedingungen für alle $s \in \{m, \dots, n-1\}$ gilt:

$$0 \leq (\text{mean}(\mathcal{U}_s) - \text{mean}(\mathcal{M})) \tag{3.5.2.1a}$$

$$0 \leq (\text{mean}(\mathcal{M}) - \text{mean}(\mathcal{O}_s)) \tag{3.5.2.1b}$$

$$0 \leq (\text{mean}(\mathcal{U}_s) - \text{mean}(\mathcal{O}_s)) \tag{3.5.2.1c}$$

Die Gleichung (3.5.1.3) bzw. der Term aus (3.5.2.1c) sticht vor dem Hintergrund des Klützke-Algorithmus hervor, da es gerade diese Differenz ist, an deren Minimalität man den Teilungspunkt festmacht. Wenn man weitere Arbeiten, wie z. B. [BC90], liest, bekommt man den Eindruck, daß primale Verfahren beim Erzeugen neuer Mengen in der Partition darauf aufbauen, eine dieser drei Bedingungen maximal zu verletzen.

Das Ziel dieses Abschnittes ist es, den folgenden Satz zu beweisen.

Satz 3.5.3. Zu jedem Ausführungszeitpunkt g des PAV-Algorithmus, wobei $0 \leq g \leq t$ gilt, ist \mathbf{x}^g eine dual zulässige Zwischenlösung.

Dazu benötigen wir noch zwei Lemmata, die sinngemäß einem Beweis aus [BC90] entnommen sind. Der Beweis zum ersten Lemma ist eine Skizze, da es recht simpel ist.

Lemma 3.5.4. Sei $\mathcal{M} \triangleleft \mathcal{N}$ mit $\text{mean}(\mathcal{M}) \geq \text{mean}(\mathcal{N})$, so gilt:

$$\text{mean}(\mathcal{M}) \geq \text{mean}(\mathcal{M} \cup \mathcal{N}) \geq \text{mean}(\mathcal{N})$$

Beweis (Skizze):

Zuerst notiert man die Ungleichung der Voraussetzung als Summen mit entsprechenden Gewichten. Man multipliziert diese Ungleichung mit dem gemeinsamen Nenner der Summen und addiert anschließend die linke Summe mit dem neuen Gewicht der rechten Summe auf beide Seiten, um die erste Ungleichung zu beweisen. Um die zweite Ungleichung zu beweisen, verfährt man vice versa. ■

Lemma 3.5.5. Sei $\mathcal{M} \triangleleft \mathcal{N}$ und $\text{mean}(\mathcal{M}) \geq \text{mean}(\mathcal{N})$. Besitzen die Teilvektoren $x[\mathcal{M}]$ und $x[\mathcal{N}]$ positive Multiplikatoren, so auch $x[\mathcal{M} \cup \mathcal{N}]$.

Beweis:

Wir nehmen an, daß

$$\begin{aligned}\mathcal{M} &= \{m, \dots, n\} \\ \mathcal{N} &= \{n+1, \dots, k\} \\ \mathcal{M} \cup \mathcal{N} &= \{m, \dots, k\}\end{aligned}$$

gilt. Es ist zu zeigen, daß alle Multiplikatoren μ_s entsprechend einem Index s aus $\mathcal{M} \cup \mathcal{N} \setminus \{k\}$ positiv sind.

Wir führen eine Fallunterscheidung durch: Für einen Teilungspunkt s gilt entweder $s \in \mathcal{M} \setminus \{n\}$, oder $s \in \mathcal{N} \setminus \{k\}$, oder $s = n$. Im letztgenannten Fall folgt $\mu_n \geq 0$ aus der Voraussetzung des Lemmas und (3.5.2.1c).

Sei also $i \in \mathcal{M} \setminus \{n\}$. Wir betrachten $\{m, \dots, s\} = \mathcal{U}_s(\mathcal{M}) = \mathcal{U}_s(\mathcal{M} \cup \mathcal{N})$. Es gilt:

$$\begin{aligned}\text{mean}(\mathcal{U}_s(\mathcal{M})) & \\ &\geq \text{mean}(\mathcal{M}) \quad \text{nach (3.5.2.1a) angewandt auf } \mathcal{M} \\ &\geq \text{mean}(\mathcal{M} \cup \mathcal{N}) \quad \text{nach (3.5.4)}.\end{aligned}$$

Also folgt

$$\text{mean}(\mathcal{U}_s(\mathcal{M} \cup \mathcal{N})) \geq \text{mean}(\mathcal{M} \cup \mathcal{N})$$

und damit

$$\mu_s \geq 0$$

im Teilsystem $x[\mathcal{M} \cup \mathcal{N}]$ wegen (3.5.2.1a) angewandt auf $\mathcal{M} \cup \mathcal{N}$.

Für $i \in \mathcal{N} \setminus \{k\}$ folgert man analog

$$\begin{aligned}\text{mean}(\mathcal{O}_s(\mathcal{N})) & \\ &\leq \text{mean}(\mathcal{N}) \quad \text{nach (3.5.2.1b) angewandt auf } \mathcal{N} \\ &\leq \text{mean}(\mathcal{M} \cup \mathcal{N}) \quad \text{nach (3.5.4)}.\end{aligned}$$

Also folgt

$$\text{mean}(\mathcal{O}_s(\mathcal{M} \cup \mathcal{N})) \leq \text{mean}(\mathcal{M} \cup \mathcal{N})$$

und damit

$$\mu_s \geq 0$$

im Teilsystem $x[\mathcal{M} \cup \mathcal{N}]$ wegen (3.5.2.1b) angewandt auf $\mathcal{M} \cup \mathcal{N}$. ■

Beweis von Satz 3.5.3:

Wir führen den Beweis durch Induktion nach g . Im Fall $g = 0$ gilt $\mu_s = 0$ für alle Multiplikatoren.

Sei also \mathbf{x}^g zulässig, d. h. für alle $\mathcal{A} \in \mathcal{T}^g$ habe der Teilvektor $\mathbf{x}^g[\mathcal{A}]$ positive Multiplikatoren. Angenommen zum Zeitpunkt g werden die Mengen \mathcal{M} und \mathcal{N} in Zeile 5 gewählt. Die Bedingungen $\mathcal{M}, \mathcal{N} \in \mathcal{T}^g : \mathcal{M} \prec \mathcal{N}, \text{mean}(\mathcal{M}) \geq \text{mean}(\mathcal{N})$ garantieren, daß Lemma 3.5.5

anwendbar ist. Also besitzt $\mathbf{x}^{g+1}[\mathcal{M} \cup \mathcal{N}]$ positive Multiplikatoren. Für $\mathcal{M} \neq \mathcal{A} \neq \mathcal{N}$ besitzt $\mathbf{x}^{g+1}[\mathcal{A}] = \mathbf{x}^g[\mathcal{A}]$ positive Multiplikatoren nach Induktionsannahme. ■

Behauptung 3.5.6. *Der PAV-Algorithmus erzeugt eine primal zulässige Lösung \mathbf{x}^t , also das Optimum.*

Beweis:

Sind keine benachbarten Mengen mehr zu finden, welche die Bedingung $\text{mean}(\mathcal{M}) \geq \text{mean}(\mathcal{N})$ verletzen, so ist klar, daß der aktuellen Partition eine primal zulässige Zwischenlösung entspricht. Das ist gerade die Abbruchbedingung des Algorithmus. ■

4. Schlußwort

Gegen Ende unserer Arbeit haben wir einen interessanten Artikel gefunden, [LRS10], in dem ein Verfahren vorgestellt wird, das deutlich allgemeiner ist als unseres. Man kann den Klützke-Algorithmus in gewisser Weise als Spezialfall interpretieren. In besagter Arbeit wird das Konzept der Monotonie erweitert auf Funktionen von \mathbb{R}^n nach \mathbb{R} , was wir hier nicht näher erläutern wollen. Jeder Nebenbedingung wird eine Kante in einen Graphen zugeordnet. Anschließend betrachtet man „monotone Kantenschnitte“ auf diesem Graphen, um jeweils zwei Teilmengen bezüglich ihres Flusses zu vergleichen. Es wird anschließend der Schnitt ausgewählt, welcher die Differenz (4.0.6.1) maximiert. Die entsprechenden beiden Teilprobleme können dann getrennt gelöst werden. Es stellt sich heraus, daß dieser Vergleich auf ein sehr ähnliches Kriterium hinausläuft, wie wir es im Klützke-Algorithmus verwenden, wenn man es auf den Spezialfall eines Pfades überträgt. Wir zitieren Bedingung (3) aus [LRS10] für den besagten Schnitt c :

$$\max_{c \in \mathcal{C}_{\mathcal{V}}} \sum_{i \in \mathcal{V}_c^+} (y_i - \bar{y}_{\mathcal{V}}) - \sum_{i \in \mathcal{V}_c^-} (y_i - \bar{y}_{\mathcal{V}}) \quad (4.0.6.1)$$

Dabei sei

- \mathcal{C} die Menge aller monotonen Schnitte,
- c eine Menge von Kanten, die einen solchen Schnitt definiert,
- \mathcal{V} die Knotenmenge, die dem Teilproblem der aktuellen Rekursion entspricht,
- und \mathcal{V}_c^- bzw. \mathcal{V}_c^+ sind die beiden Knotenmengen, die unter bzw. über dem Schnitt entstehen.

Ferner ist mit

$$\bar{y}_{\mathcal{V}} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} y_i \quad (4.0.6.2)$$

ein arithmetisches Mittel gemeint. Im Falle eines Pfades aus n Knoten gibt es $n - 1$ mögliche Schnitte, die jeweils aus einer Kante bestehen. Wir formen die obige Gleichung um und werden Gleichung (3.5.1.3) in einem neuen, allgemeineren Gewand wieder erkennen:

$$\begin{aligned}
& \sum_{i \in \mathcal{V}_c^+} (y_i - \bar{y}_{\mathcal{V}}) - \sum_{i \in \mathcal{V}_c^-} (y_i - \bar{y}_{\mathcal{V}}) \\
&= \sum_{i \in \mathcal{V}_c^+} y_i - \frac{|\mathcal{V}_c^+|}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} y_i - \sum_{i \in \mathcal{V}_c^-} y_i + \frac{|\mathcal{V}_c^-|}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} y_i \\
&= \frac{|\mathcal{V}_c^-|}{|\mathcal{V}|} \sum_{i \in \mathcal{V}_c^+} y_i - \frac{|\mathcal{V}_c^+|}{|\mathcal{V}|} \sum_{i \in \mathcal{V}_c^-} y_i - \frac{|\mathcal{V}_c^+|}{|\mathcal{V}|} \sum_{i \in \mathcal{V}_c^-} y_i + \frac{|\mathcal{V}_c^-|}{|\mathcal{V}|} \sum_{i \in \mathcal{V}_c^+} y_i \\
&= 2 \frac{|\mathcal{V}_c^+| |\mathcal{V}_c^-|}{|\mathcal{V}|} \left(\frac{1}{|\mathcal{V}_c^+|} \sum_{i \in \mathcal{V}_c^+} y_i - \frac{1}{|\mathcal{V}_c^-|} \sum_{i \in \mathcal{V}_c^-} y_i \right) \\
&= 2 \frac{|\mathcal{V}_c^+| |\mathcal{V}_c^-|}{|\mathcal{V}|} (\bar{y}_{\mathcal{V}_c^+} - \bar{y}_{\mathcal{V}_c^-})
\end{aligned}$$

Das Maximierungsproblem (4.0.6.1) läuft in unserem Fall also darauf hinaus, in einem Teilproblem nach dem *minimalen* Multiplikator μ_i zu suchen – **man beachte die umgedrehten Vorzeichen**. Die rechte Klammer alleine entspricht Zeile 5 in Algorithmus 1. Wir haben diese abgewandelte Bedingung in unseren Quellcode eingepflegt und konnten im Endergebnis keine andere Partitionierung feststellen. Der berechnete Trennungspunkt in einer Rekursion ist jedoch nicht stets der gleiche. Ein Beispiel, daß sich auch per Hand gut nachvollziehen läßt, liefert der folgende Eingabevektor

$$\mathbf{y} = \begin{pmatrix} 0 \\ 6 \\ 18 \\ 30 \end{pmatrix}.$$

Ist man mit einem, aus der eigenen Perspektive heraus, neuen Problem konfrontiert, so kann man auf tendenziell zwei Arten damit umgehen: Erstens, man sucht sich einen pragmatischen Weg zum Ziel – eventuell sind die Werkzeuge dabei etwas improvisiert. Zweitens, man nimmt das Problem mit den dazu gedachten Werkzeugen fachgerecht unter die Lupe. Unsere Herangehensweise ist wohl ein Kompromiss zwischen diesen beiden.

Anhang

A. Der MSE (Mean Squared Error)

Dieser Abschnitt ist eine Art Kurzfassung der Abschnitte 9.2.1 und 9.2.2 aus [FKPT01].

Möchte man für eine bestimmte Zufallsvariable $X \sim V(\theta)$, den unbekannt Parameter θ der Verteilung schätzen, so kann man das Experiment n mal wiederholen und versuchen durch die empirische Verteilung der Stichprobe auf die Verteilung der Grundgesamtheit zu schließen. Aus theoretischer Sicht beobachtet man n Zufallsvariablen X_1, \dots, X_n , die unabhängig und identisch verteilt sind. Mithilfe einer sogenannten Schätzstatistik $T = g(X_1, \dots, X_n)$ läßt sich der Parameter θ mehr oder weniger genau bestimmen. Als Beispiel denke man sich hier eine Poissonverteilte Zufallsvariable $X \sim Po(\lambda)$ mit der Dichte $\rho_\lambda(k) = \frac{\lambda^k}{k!} e^{-\lambda}$ und dem Erwartungswert $\mathbb{E}[X] = \lambda$. Letzterer läßt sich beispielsweise mithilfe der Schätzstatistik $T = \frac{1}{n} \sum_{i=1}^n X_i$, was dem arithmetischen Mittel entspricht, approximieren. Im allgemeinen gibt es mehrere Schätzverfahren, die man miteinander vergleichen möchte. Im Falle der Poissonverteilung wäre es denkbar gewesen, λ^2 über die empirische Varianz $\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$, wobei \bar{X} das arithmetische Mittel bezeichnet, abzuschätzen. Naiv könnte man so über die Wurzel auf λ zurück rechnen – doch es stellt sich bei mehreren Verfahren stets die Frage, welche Vorgehensweise „besser“ ist. Um das zu präzisieren, hat sich in der Statistik der Begriff des MSE etabliert.

Definition A.0.7. Für gegebenes θ bezeichnet

$$MSE_\theta = \mathbb{E}[(T - \theta)^2]$$

der Erwartungswert der quadratischen Abweichung von T zu θ . Die Bezeichnung stammt aus dem Englischen *mean squared error*.

Definition A.0.8. Für gegebenes $X \sim V(\theta)$ bezeichnet

$$\text{Bias}_\theta(T) = \mathbb{E}_\theta[T] - \theta$$

die erwartete Abweichung der Schätzstatistik T vom wahren Wert θ , den *Bias* von T unter θ .

Der Bias gibt Aufschluß darüber, wie weit man im Mittel vom wahren Wert „fehlschätzt“. Dieser kann für unterschiedliche θ durchaus schwanken. Man denke z. B. an eine konstante

Schätzstatistik $T = g(X_1, \dots, X_n) = c$. Die Varianz $\mathbb{V}(T)$ von T hingegen sagt aus, wie weit die Schätzwerte streuen, unabhängig davon um *welchen* Wert. Tatsächlich gilt¹:

$$MSE = \mathbb{V}(T) + \text{Bias}(T)^2.$$

Der MSE ist eine theoriegestützte Eigenschaft von T . Zur Berechnung Bedarf es einer Kenntnis der Verteilung mitsamt dem Parameter θ . Ferner muß die Funktion g einer Analyse zugänglich sein.

Doch auch ohne Theorie kann man ein Schätzverfahren am Computer simulieren und anstelle der theoretischen Größen Erwartungswert und Varianz deren empirische Entsprechungen des arithmetischen Mittels und der Stichprobenvarianz setzen. So wurde in dieser Arbeit vorgegangen.

¹Wir lassen den Index fortan weg und behalten im Hinterkopf, daß die entsprechenden Größen auch von θ abhängen.

Literatur

- [ABERS55] M. Ayer, H.D. Brunk, G.M. Ewing, W.T. Reid und E. Silverman. »An empirical distribution function for sampling with incomplete information«. In: *Annals of Mathematical Statistics* 26.4 (1955), S. 641–647. DOI: [10.1214/aoms/1177728423](https://doi.org/10.1214/aoms/1177728423).
- [AGHK14] Daniel Altmann, Eugen Grycko, Winfried Hochstättler und Günter Klütze. »Monotone smoothing of noisy data«. In: *Preprint Server der FernUniversität Hagen* (2014). URL: <https://www.fernuni-hagen.de/MATHEMATIK/DMO/pubs/feudmo034-14.pdf>.
- [Ave95] Jürgen Avenhaus. *Reduktionssysteme*. Berlin, Heidelberg: Springer, 1995. ISBN: 3-540-58559-1.
- [BC90] Michael J. Best und Nilotpal Chakravarti. »Active set algorithms for isotonic regression; A unifying framework«. In: *Mathematical Programming* 47 (1990), S. 425–439. DOI: [10.1007/BF01580873](https://doi.org/10.1007/BF01580873).
- [Ebb03] Heinz-Dieter Ebbinghaus. *Einführung in die Mengenlehre*. 4. Aufl. Heidelberg, Berlin: Spektrum Akademischer Verlag, 2003. ISBN: 3-8274-1411-3. URL: <http://www.springer.com/springer+spektrum/mathematik/book/978-3-8274-1411-3>.
- [FKPT01] Ludwig Fahrmeir, Rita Künstler, Iris Pigeot und Gerhard Tutz. *Statistik - Der Weg zur Datenanalyse*. 3., verb. Aufl. Berlin, Heidelberg, New York u.a.: Springer, 2001. ISBN: 3-540-67826-3.
- [GW84] S.J. Grotzinger und C. Witzgall. »Projections onto order simplexes«. In: *Applied Mathematics and Optimization* 12.1 (1984), S. 247–270. DOI: [10.1007/BF01449044](https://doi.org/10.1007/BF01449044).
- [Heu09] Harro Heuser. *Gewöhnliche Differentialgleichungen - Einführung in Lehre und Gebrauch*. 6., akt. Aufl. Wiesbaden: Vieweg+Teubner, 2009. ISBN: 978-3-834-80705-2.
- [Jun99] Dieter Jungnickel. *Optimierungsmethoden - Eine Einführung*. Berlin, Heidelberg: Springer, 1999. ISBN: 3-540-66057-7.
- [Kea06] Anthony J. Kearsley. »Projections Onto Order Simplexes and Isotonic Regression«. In: *J. Res. Natl. Inst. Stand. Technol.* 111.2 (2006), S. 121–125. DOI: [10.6028/jres.111.011](https://doi.org/10.6028/jres.111.011).
- [LB71] Saunders Mac Lane und Garrett Birkhoff. *Algebra*. 6. Aufl. London: Collier-Macmillan Ltd, 1971.

- [LRS10] Ronny Luss, Saharon Rosset und Moni Shahar. »Decomposing Isotonic Regression for Efficiently Solving Large Problems«. In: *Advances in Neural Information Processing Systems 23*. Curran Associates, Inc., 2010, S. 1513–1521. URL: <http://papers.nips.cc/paper/4079-decomposing-isotonic-regression-for-efficiently-solving-large-problems.pdf>.
- [MGPS98] Otto Moeschlin, Eugen Grycko, Claudia Pohl und Frank Steinert. *Experimental Stochastics*. Berlin, Heidelberg, New York: Springer, 1998. ISBN: 3-540-14619-9.
- [NW06] Jorge Nocedal und Stephen J. Wright. *Numerical Optimization*. 2. Aufl. New York: Springer Science+Business Media, 2006. ISBN: 978-0-387-30303-1. DOI: [10.1007/978-0-387-40065-5](https://doi.org/10.1007/978-0-387-40065-5).
- [Ree08] Rembert Reemtsen. *Einführung in die nichtlineare Optimierung. Ein Kurs für die FernUniversität Hagen*. 2008.
- [Sta] *Isotonic Regression wiki page*. http://stat.wikia.com/wiki/Isotonic_regression. Accessed: 2014-05-13.
- [WWM01] Wei Biao Wu, Michael Woodroffe und Graciela Mentz. »Isotonic regression: Another look at the changepoint problem«. In: *Biometrika* 88.3 (2001), S. 793–804. DOI: [10.1093/biomet/88.3.793](https://doi.org/10.1093/biomet/88.3.793).

Versicherung

Name: Daniel Altmann
Matrikel-Nr.: 831 49 42
Fach: Mathematik
Modul: Abschlußarbeit

Hiermit versichere ich an Eides statt, daß ich die vorliegende Abschlußarbeit mit dem Thema

„Ein Active-Set-Verfahren quadratischer Laufzeit zur Least-Squares-Approximation von Daten monotonen Ursprungs“

selbständig und ohne Inanspruchnahme fremder Hilfe angefertigt habe. Ich habe dabei nur die angegebenen Quellen und Hilfsmittel verwendet und die aus diesen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen. Ich erkläre mich damit einverstanden, daß die Arbeit mit Hilfe eines Plagiatserkennungsdienstes auf enthaltene Plagiate überprüft wird.

Datum: _____ Unterschrift: _____