

Hansohm, Jürgen

Working Paper

Algorithmus und Programm zur Bestimmung der monotonen Kleinst-Quadrate Lösung bei partiellen Präordnungen

Arbeitspapiere zur mathematischen Wirtschaftsforschung, No. 187

Provided in Cooperation with:

Universität Augsburg, Institut für Statistik und Mathematische Wirtschaftstheorie

Suggested Citation: Hansohm, Jürgen (2004) : Algorithmus und Programm zur Bestimmung der monotonen Kleinst-Quadrate Lösung bei partiellen Präordnungen, Arbeitspapiere zur mathematischen Wirtschaftsforschung, No. 187, Universität Augsburg, Institut für Statistik und Mathematische Wirtschaftstheorie, Augsburg

This Version is available at:

<https://hdl.handle.net/10419/22820>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

ARBEITSPAPIERE
ZUR MATHEMATISCHEN WIRTSCHAFTSFORSCHUNG

**Algorithmus und Programm
zur Bestimmung der monotonen Kleinst-Quadrate Lösung
bei partiellen Präordnungen**

Jürgen Hansohm

Heft 187/2004

Institut für Statistik und
Mathematische Wirtschaftstheorie
Universität Augsburg
D-86135 Augsburg
Tel. 0821/598-4150

Prof. Dr. Jürgen Hansohm
Fakultät WOW
Universität der Bundeswehr
Werner-Heisenberg-Weg
D-85579 Neubiberg
Tel. 089 / 6004-3542

Algorithmus und Programm zur Bestimmung der monotonen Kleinst-Quadrate Lösung bei partiellen Präordnungen

Jürgen Hansohm

Juergen.Hansohm@unibw-muenchen.de

<http://www.unibw-muenchen.de/campus/WOW/v1052/>

Abstract: Sei $A = \{1, \dots, n\}$ eine endliche Menge und $f : A \rightarrow \mathbf{R}$ gegeben. Im folgenden Beitrag werden Algorithmen zur Ermittlung der Kleinst-Quadrate Regression $g : A \rightarrow \mathbf{R}$ von f beschrieben, wobei g monoton bzgl. einer nicht notwendigerweise vollständigen Präordnung \preceq auf A ist. Neben den theoretischen Grundlagen und dem allgemeinen Fall werden auch die Spezialfälle der hierarchischen und vollständigen Präordnung beschrieben. Darüber hinaus werden Fehlerabschätzungen zum Optimum angegeben.

Sämtliche Algorithmen sind in dem Modul `PartOrder` implementiert, dessen Anwendung an einem Beispiel kurz beschrieben wird. Das Modul basiert auf dem .NET Framework und kann unter verschiedenen Programmiersprachen (Visual Basic, C++, C#, etc.) aufgerufen werden.

1 Einleitung

Das monotone Regressionsproblem tritt in vielen Fragestellungen auf, so bei der multidimensionalen Skalierung (Kruskal 1964a, b), der Behandlung qualitativer Daten in quantitativen Analysemethoden (Young 1975, 1981, Young et al. 1976, 1978), etc. Nicht immer sind hierbei die Präordnungen vollständig. Partielle Präordnungen treten z.B. bei komplexen Informationsniveaus auf (Hansohm 2003a), bei stratigraphischen Methoden in der Archäologie (Herzog 1993) oder bei isotonischen Schätzungen (Robertson et al. 1988)

2 Problemstellung

Sei im folgenden $A = \{1, \dots, n\}$ eine endliche Menge mit einer Präordnung¹⁾ \preceq auf A . Sei $f : A \rightarrow \mathbf{R}$, $w : A \rightarrow \mathbf{R}$ mit $w(a) > 0$ für alle $a \in A$, dann ist die Funktion $g : A \rightarrow \mathbf{R}$ gesucht, für die gilt:

¹⁾eine reflexive und transitive Relation

Monotoniebedingung :

$$a, b \in A, a \preceq b \Rightarrow g(a) \leq g(b) \quad (1)$$

und die

Kleinst-Quadrate Bedingung :

$$\sum_{a \in A} w(a)(f(a) - g(a))^2 \leq \sum_{a \in A} w(a)(f(a) - h(a))^2 \quad (2)$$

für alle Funktionen $h : A \rightarrow \mathbf{R}$, die die Monotoniebedingung erfüllen

Bemerkung 1 *Eine solche Funktion existiert immer und ist eindeutig (s. (5)).*

3 Theoretische Grundlagen

Die Menge $H = \{h : A \rightarrow \mathbf{R}\}$ ist mit

$$\langle r, s \rangle = \sum_{a \in A} w(a)r(a)s(a), \quad \|h\| = \sqrt{\langle h, h \rangle}, \quad r, s, h \in H \quad (3)$$

ein Hilbertraum und die Menge $K = \{k \in H \mid k \text{ erfüllt die Monotoniebedingung (1)}\}$ ist ein nichtleerer abgeschlossener konvexer Kegel in diesem Hilbertraum. Die gesuchte Lösung ist offensichtlich die Projektion von f auf diesen Kegel K , wobei die Projektion definiert ist als

$$P_C : H \rightarrow \mathcal{P}(H), \quad P_C(h) = \{\tilde{c} \in C \mid \|\tilde{c} - h\| \leq \|c - h\| \text{ für alle } c \in C \subseteq H\} \quad (4)$$

für alle nichtleeren abgeschlossenen Mengen C von H .

P_C hat mit $h, r, s \in H$ die folgenden Eigenschaften:

$$C \text{ konvex} \Rightarrow |P_C h| = 1 \quad (5)$$

$$C \text{ konvex} \Rightarrow \|P_C r - P_C s\| \leq \|r - s\| \quad (6)$$

$$C \text{ konvex} \Rightarrow (\langle h - z, c - z \rangle \leq 0 \text{ für alle } c \in C \Leftrightarrow z = P_C h) \text{ für alle } z \in C \quad (7)$$

$$C \text{ Kegel} \Rightarrow \langle h - z, z \rangle = 0 \text{ für alle } c \in C, z \in P_C h \quad (8)$$

Definition 1 Eine Teilmenge $I \subseteq A$ heißt **Ideal**, wenn gilt:

$$a \in I, b \in A, b \preceq a \Rightarrow b \in I$$

für alle $a \in I, b \in A$.

Definition 2 Die Abbildung $m : \mathcal{P}(A) \rightarrow \mathbf{IR}$ mit

$$m(C) = \frac{\sum_{c \in C} w(c)f(c)}{\sum_{c \in C} w(c)}$$

heißt **Durchschnittsfunktion** von C bzgl. f .

Satz 1 $(P_K f)(a)$ ist für $a \in \tilde{I}$ konstant gleich $m(\tilde{I})$, wenn \tilde{I} das Ideal mit minimalem $m(\cdot)$ ist und unter allen mit dieser Eigenschaft das größte²⁾.

Beweis: Da die Funktion $P_K f$ nur endlich viele Werte aufweist, ist leicht zu sehen, daß für jede nichtleere Menge $T = \{a \in A \mid (P_K f)(a) = \text{const}\}$ gilt:

$$(P_K f)(t) = m(T) \text{ für alle } t \in T$$

Sei nun $T_{\min} = \{a \in A \mid (P_K f)(a) = \min_{c \in A} (P_K f)(c)\}$. Wie man leicht erkennt, ist T_{\min} ein Ideal und für jedes Ideal $I \subseteq T_{\min}$ gilt $m(T_{\min}) \leq m(I)$ und für jedes Ideal $T_{\min} \subseteq I$ ebenfalls $m(T_{\min}) \leq m(I)$. Da sich $m(\tilde{I} \cup T_{\min})$ aus $m(T_{\min})$, $m(\tilde{I})$ und $-m(\tilde{I} \cap T_{\min})$ zusammensetzt, folgt $m(\tilde{I} \cup T_{\min}) = m(T_{\min})$ und nach Konstruktion von \tilde{I} : $T_{\min} \subseteq \tilde{I}$. $m(\cdot)$ ist eine strikte Cauchy Durchschnittsfunktion (Cauchy mean value function, Robertson et al. 1988, p. 24); d.h. für zwei disjunkte, nichtleere Mengen $R, S \subseteq A$ gilt:

$$m(R \cup S) \leq m(R) \Leftrightarrow m(S) \leq m(R \cup S)$$

Hieraus ergibt sich $m(\tilde{I} \setminus T_{\min}) = m(\tilde{I}) = m(T_{\min})$, ein Widerspruch zu $(P_K f)(\tilde{c}) > \min_{c \in A} (P_K f)(c)$ für $\tilde{c} \in \tilde{I} \setminus T_{\min}$, falls $\tilde{I} \setminus T_{\min} \neq \emptyset$. \square

Betrachtet man nun $A' = A \setminus I^*$ und wieder das Ideal I' von A' mit den obigen Eigenschaften, so erhält man sukzessive die Funktion $g = P_K f$ (siehe auch Brunk et al. (1957), Barlow et al. 1972, pp. 76, Robertson et al. 1988, pp. 24).

Diese Aussage liefert leider nur in Spezialfällen Ansätze für eine Berechnung. Im generellen Falle ist die Menge der zu untersuchenden Ideale praktisch zu groß. Für den generellen Fall wird deshalb ein Näherungsverfahren vorgeschlagen. Dessen Grundlagen werden im folgenden dargelegt.

Definition 3 Die zur abgeschlossenen, nichtleeren Menge $C \subset H$ ³⁾ **duale Menge** C^* ist definiert als die Menge aller linearen Funktionale auf H , die auf C nicht positiv sind; d.h.

$$C^* = \{u \in H \mid \langle u, c \rangle \leq 0 \text{ für alle } c \in C\}$$

²⁾Vereinigung und Durchschnitt von Idealen sind Ideale

³⁾um Verwechslungen mit H^* zu vermeiden

Es gilt:

$$C^* \text{ ist ein nichtleerer abgeschlossener konvexer Kegel} \quad (9)$$

$$C \text{ nichtleerer abgeschlossener konvexer Kegel} \Leftrightarrow (C^*)^* = C \quad (10)$$

$$C, \tilde{C} \text{ nichtleere abgeschlossene konvexe Kegel} \Rightarrow (\tilde{C} \subseteq C \Leftrightarrow C^* \subseteq \tilde{C}^*) \quad (11)$$

$$C_i \text{ nichtleere abgeschlossene konvexe Kegel} \Rightarrow (\cap C_i)^* = \sum_i C_i^* \quad (12)$$

Beweis: (9) ist trivial, (11) folgt aus (10). Zu (10): Da H reflexiv ist, ist jedes $c \in C$ ein lineares Funktional auf C^* und $\langle c, u \rangle \leq 0$ für alle $u \in C^* \Rightarrow C \subseteq (C^*)^*$. Sei umgekehrt $\tilde{c} \in (C^*)^*$. Aus (7) in Verbindung mit (8) folgt $\tilde{c} - P_C \tilde{c} \in C^*$ und damit $\langle \tilde{c} - P_C \tilde{c}, \tilde{c} - P_C \tilde{c} \rangle = \langle \tilde{c}, \tilde{c} - P_C \tilde{c} \rangle - \langle P_C \tilde{c}, \tilde{c} - P_C \tilde{c} \rangle = \langle \tilde{c}, \tilde{c} - P_C \tilde{c} \rangle \leq 0$. Also $\tilde{c} = P_C \tilde{c}$, bzw. $\tilde{c} \in C$. Zu (12): Sei $u = \sum_i u_i$ mit $u_i \in C_i^* \Rightarrow \langle u, c \rangle = \sum_i \langle u_i, c \rangle \leq 0$ für $c \in \cap C_i \Rightarrow u \in (\cap C_i)^* \Rightarrow \sum_i C_i^* \subseteq (\cap C_i)^*$. Ist umgekehrt $\tilde{c} \in (\sum_i C_i^*)^* \Rightarrow \langle u, \tilde{c} \rangle \leq 0$ und insbesondere $\langle u_i, \tilde{c} \rangle \leq 0$ für alle i . Somit ist $\tilde{c} \in \cap (C_i^*)^* = \cap C_i \Rightarrow (\sum_i C_i^*)^* \subseteq \cap C_i \Rightarrow \sum_i C_i^* = (\cap C_i)^*$ (11). \square

Die Bedeutung des dualen Kegels wird aus folgendem Satz deutlich:

Satz 2 Sei C ein nichtleerer abgeschlossener konvexer Kegel

$$g \in C, u \in C^*, g + u = f, \langle u, g \rangle = 0 \Rightarrow g = P_C f, u = P_{C^*} f \quad (13)$$

$$g = P_C f \Rightarrow u = f - g = P_{C^*} f, \langle u, g \rangle = 0 \quad (14)$$

$$u = P_{C^*} f \Rightarrow g = f - u = P_C f, \langle u, g \rangle = 0 \quad (15)$$

Beweis: Zu (13): $\langle u, c - g \rangle \leq 0$ für alle $c \in C \Rightarrow \langle f - g, c - g \rangle \leq 0$ und mit (7): $g = P_C f$. Aus (10) folgt $u = P_{C^*} f$. Zu (14): Aus (7) und (8) folgt $u = f - g \in C^*$ und nach (13) die Behauptung. (15) ergibt sich analog. \square

Definition 4 Zwei Elemente $a, b \in A$ heißen **äquivalent**, wenn $a \preceq b$ und $b \preceq a$ gilt, im Zeichen $a \sim b$.

Für äquivalente Elemente $a \sim b$ gilt offensichtlich $(P_K f)(a) = (P_K f)(b)$. Nach dem folgenden Satz können äquivalente Elemente vorher "eliminiert" werden; d.h. äquivalente Elemente werden sukzessiv zusammengelegt zu einem neuen Element mit entsprechend höherem Gewicht w (hierzu s.u.). Grundlage bildet der folgende

Satz 3 Sei g die optimale Lösung für f ; d.h. $\{g\} = P_K f$ (g erfüllt (1) und (2)) und $g(a) = g(b)$. Sei $T \subseteq H$ die Menge aller reellwertigen Funktionen auf A mit $t(a) = t(b)$ für $t \in T^4$. Dann gilt:

$$P_K f = P_{K \cap T} P_T f \quad (16)$$

⁴ T ist ein Teilraum von H und $K \cap T$ ein Kegel in T mit $g \in K \cap T$

Beweis: Sei $t_0 \in P_T f$. $P_T f$ ist einelementig $\Rightarrow P_T f = \{t_0\}$. Es gilt $\langle f - t_0, t \rangle = 0$ für alle $t \in T$.

Für alle $t \in T$ gilt:

$$\langle f - t_0, f - t_0 \rangle + \langle t_0 - t, t_0 - t \rangle = \langle f - t, f - t \rangle \quad (17)$$

denn

$$\begin{aligned} & \langle f - t, f - t \rangle \\ &= \langle f - t_0 + t_0 - t, f - t_0 + t_0 - t \rangle \\ &= \langle f - t_0, f - t_0 \rangle + \langle t_0 - t, t_0 - t \rangle + 2\langle f - t_0, t_0 - t \rangle \\ &= \langle f - t_0, f - t_0 \rangle + \langle t_0 - t, t_0 - t \rangle \end{aligned}$$

Da $\langle f - t_0, f - t_0 \rangle$ konstant ist, entspricht die Minimierung von $\langle f - t, f - t \rangle$ der Minimierung von $\langle t_0 - t, t_0 - t \rangle$ und umgekehrt. \square

4 Algorithmus

Sei zuerst der allgemeine Fall behandelt, die Spezialfälle erlauben i.a. effizientere Algorithmen.

Definition 5 $a \in A$ ist ein **unmittelbarer Vorgänger** von $b \in A$ bzgl. der Präordnung \preceq , wenn $a \preceq b$ und aus $a \preceq c \preceq b \Rightarrow c \sim a$ oder $c \sim b$ für alle $c \in A$ folgt. Entsprechend wird ein **unmittelbarer Nachfolger** definiert.

Sei im folgenden unterstellt, daß es keine äquivalenten Elemente in A gibt⁵⁾; d.h. daß aus $a \preceq b$ folgt $b \not\preceq a$ ($a, b \in A$). Alle Präferenzrelationen zwischen zwei Elementen $a, b \in A$, so vorhanden, seien durchnummeriert ($1: a \preceq b$, $2: a \preceq c$, $3: b \preceq d$, \dots , $m: x \preceq z$). Hierbei können Relationen, die sich aus der Transitivität ergeben (wie z.B. $a \preceq c$ bei $a \preceq b, b \preceq c$) weggelassen werden; d.h. es werden nur direkte Vorgänger-Nachfolgerbeziehungen berücksichtigt, dies ist aber nicht Bedingung. Sei $K_i = \{h \in H \mid h \text{ monoton (vgl. (1)) bezüglich der } i\text{-ten Relation}\}$ ($i = 1, \dots, m$). K_i ist offensichtlich ein nichtleerer, abgeschlossener, konvexer Kegel in H mit:

$$K = \bigcap_{i=1}^m K_i \quad (18)$$

Letzteres folgt aufgrund der Transitivität der Präferenzrelation \preceq auf A und \leq auf \mathbf{IR} . Um die Projektion auf K zu ermitteln, kann der Algorithmus von Dykstra angewendet werden (Dykstra 1983). Der Algorithmus benötigt nur die Projektionen auf K_i (hierzu s.u.).

⁵⁾weiter unten wird gezeigt, wie dies erreicht werden kann

Satz 4 Seien K_1, \dots, K_m nichtleere, abgeschlossene, konvexe Kegel im endlichdimensionalen Hilbertraum H und $f \in H$. Dann sei folgender Algorithmus definiert:

Start :

$$u_1^{(0)}, \dots, u_m^{(0)} = 0$$

Iteration :

$$u_i^{(k+1)} = P_{K_i^*} \left(f - \sum_{j < i} u_j^{(k+1)} - \sum_{j > i} u_j^{(k)} \right) \quad (i = 1, \dots, m)$$

Es gilt:

$$u_i^{(k)} \xrightarrow{k \rightarrow \infty} u_i^* \in K_i^* \quad (i = 1, \dots, m)$$

und mit $u^* = \sum_{i=1}^m u_i^*$, $K = \cap K_i$

$$u^* = P_{K^*} f, f - u^* = P_K f$$

Beweisskizze: Offensichtlich gilt:

$$\|f - \sum_{j \leq i} u_j^{(k+1)} - \sum_{j > i} u_j^{(k)}\| \leq \|f - \sum_{j < i} u_j^{(k+1)} - \sum_{j \geq i} u_j^{(k)}\| \leq \|f - 0\| \leq \|f\|$$

Damit ist klar, daß $u^{(k)} = \sum_i u_i^{(k)}$ in einer kompakten Menge enthalten ist, da H endlich dimensional ist. In einem etwas länglichen Beweis (s. Dykstra 1983) kann man zeigen, daß auch die $u_i^{(k)}$ ($i = 1, \dots, m$) beschränkt sind. Sei deshalb o.B.d.A. $(u_i^{(k_\nu)})_\nu$ eine für alle $i = 1, \dots, m$ konvergente Teilfolge und $u_i^{(k_\nu)} \xrightarrow{\nu \rightarrow \infty} u_i^*$ für alle $i = 1, \dots, m$. Aus der Abgeschlossenheit der K_i^* folgt $u_i^* \in K_i^* \Rightarrow u_i^* \in K^*$ (11) $\Rightarrow u^* \in K^*$ (9). Sei $\tilde{u} = \sum_i \tilde{u}_i \in K^*$ (12), dann gilt für alle $i = 1, \dots, m$:

$$\Rightarrow \langle f - \sum_{j < i} u_j^{(k_\nu+1)} - \sum_{j > i} u_j^{(k_\nu)} - u_i^{(k_\nu+1)}, \tilde{u}_i - u_i^{(k_\nu+1)} \rangle \leq 0$$

$\Rightarrow \langle f - u^*, \tilde{u}_i - u_i^* \rangle \leq 0 \Rightarrow \langle f - u^*, \tilde{u} - u^* \rangle \leq 0 \Rightarrow u^* = P_{K^*} f$ (7). Aus (15) folgt $f - u^* = P_K f$. Da $P_K f, P_{K^*} f$ einelementig sind, folgt die Konvergenz der Folge $(u_i^{(k)})_k$ für alle $i = 1, \dots, m$ und somit die Behauptung (s. Luenberger 1973). \square

5 Programm

Im Programm `PartOrder` - der Wurzelklasse im *namespace* `PartOrder` - werden zuerst alle äquivalenten Elemente zusammengefaßt. Dies geschieht durch den expliziten Aufruf der Methode `puttiestgether()`. Wenn für $a, b \in A$ gilt: $a \sim b$, so ist offensichtlich in der optimalen Lösung $(P_K f)(a) = (P_K f)(b)$. Nach Satz 3 kann dann b eliminiert und a durch a' ersetzt werden mit:

$$f(a') = (w(a)f(a) + w(b)f(b))/(w(a) + w(b))$$

$$w(a') = w(a) + w(b)$$

$$c \preceq a' \Leftrightarrow c \preceq a \text{ oder } c \preceq b \text{ für alle } c \in A$$

$$a' \preceq c \Leftrightarrow a \preceq c \text{ oder } b \preceq c \text{ für alle } c \in A$$

Dies ist offensichtlich die Projektion auf den Teilraum aller $h \in H$ mit $h(a) = h(b)$. Man beachte, daß durch das Zusammenfassen zweier äquivalenter Elemente neue äquivalente Elemente entstehen können, z.B. wenn $a \preceq c, c \preceq b$ und $a \sim b$.

Als zweiten Schritt wird der Algorithmus von Dykstra durchgeführt. Im Gegensatz zur von `PartOrder` abgeleiteten Klasse `ListPartOrder` verwendet `PartOrder` aus Effizienzgründen nicht die oben erwähnten Einzelrelationen zwischen zwei Elementen $a \preceq b$ als Basis für den Kegel, sondern für jedes Element $a \in A$ den Kegel, der auf

$$K_a = \{b \in A \mid b \text{ direkter Nachfolger von } a\} \cup \{a\}$$

monotonen Funktionale $h \in H$.

Die Projektion auf einen Kegel K_a ist durch einen einfachen Algorithmus ermittelbar, da es sich um eine Hierarchie handelt (s. Hansohm 2004). Um eine einfache Hierarchie, auch *simple tree* genannt, handelt es sich deshalb, weil die direkten Nachfolger von a nicht in Relation zueinander stehen⁶⁾.

Die Berechnung der Kleinst-Quadrate Lösung für diesen *simple tree* kann durch das Verfahren von Kruskal (Kruskal 1964a,b) für vollständige Präordnungen geschehen, wobei die vollständige Präordnung \preceq' auf A definiert ist durch:

1. $b, c \in K_a, b \neq a, c \neq a \Rightarrow b \preceq' a \Leftrightarrow f(b) \leq f(c)$
2. $b \in K_a, b \neq a \Rightarrow a \preceq' b$

Die Klasse `PartOrder` benötigt für die Angabe der partiellen Ordnung die Adjazenzmatrix der in Relation \preceq stehenden Paare (nur direkte Nachfolger) und erlaubt neben einfach zu implementierenden graphentheoretischen Algorithmen (z.B. Eliminierung schon transitiv gegebener Relationen) auch die Reduzierung der Elementzahl von A durch folgende weitere Überlegungen (vgl. Barlow, et al. (1972), pp.79, Theorem 2.6):

- Hat a^* mit $f(a^*) = \min\{f(a) \mid a \in A\}$ keinen Vorgänger, so ist $g(a^*) = f(a^*)$ und A kann um a^* reduziert werden.
- Hat a^* mit $f(a^*) = \max\{f(a) \mid a \in A\}$ keinen Nachfolger, so ist $g(a^*) = f(a^*)$ und A kann um a^* reduziert werden.
- Hat a^* mit $f(a^*) = \min\{f(a) \mid a \in A\}$ nur einen direkten Vorgänger, so kann a^* mit diesem zusammengelegt werden.

⁶⁾da es ja keine äquivalenten Elemente in A gibt

- Hat a^* mit $f(a^*) = \max\{f(a) \mid a \in A\}$ nur einen direkten Nachfolger, so kann a^* mit diesem zusammengelegt werden.
- Ist $f(b) > f(a)$ und $a \preceq b$ und a hat als direkten Nachfolger nur b und b hat als direkten Vorgänger nur a , so können a und b zusammengelegt werden⁷⁾.

6 Implementation und Spezialfälle

Die von `PartOrder` abgeleitete Klasse `ListPartOrder` benötigt wesentlich weniger Speicherplatz als `PartOrder`, da sie nur mit der Liste der oben beschriebenen Paarbeziehungen arbeitet. Allerdings ist die Menge der zu schneidenden Kegel damit höher. Analog zu `PartOrder` wird $u_i^{(k+1)} = P_{K_i^*}(f - \sum_{j<i} u_j^{(k+1)} - \sum_{j>i} u_j^{(k)})$ berechnet durch

$$f - \sum_{j<i} u_j^{(k+1)} - \sum_{j>i} u_j^{(k)} - P_{K_i}(f - \sum_{j<i} u_j^{(k+1)} - \sum_{j>i} u_j^{(k)})$$

und unterscheidet sich von $y = f - \sum_{j<i} u_j^{(k+1)} - \sum_{j>i} u_j^{(k)}$ also nur durch die beiden in der i -ten Relation liegenden Elemente. Diese beiden Werte sind demnach pro Kegel K_i nur zu speichern. Die Projektion von y auf den Kegel K_i sind elementar ausrechenbar, denn sei die i -te Relation $a \preceq b$, $K_i = \{h \in H \mid h \text{ monoton bzgl. } a \preceq b\}$ und $y(a) \leq y(b)$, so ist $P_{K_i}y = y$, anderenfalls:

$$(P_{K_i}y)(c) = \begin{cases} y(c) & \text{für } c \in A, c \neq a, c \neq b \\ m(\{a, b\}) & \text{sonst (s. Definition 2)} \end{cases}$$

6.1 Profile und Verbandsordnungen

Dijkstra (Robertson et al. 1988, p. 27, Theorem 1.4.6) selber schlägt seinen Algorithmus nur für folgende verbandsgeordnete (Dijkstra nennt diese matrixgeordnete) Präordnungen vor. Sei jedes Element $a \in A$ darstellbar als k -wertiger Vektor von Profilgrößen $p_i (i = 1 \dots, k)$, also $a = (p_1, \dots, p_k)$ und sei für jede dieser Profilgrößen p_i eine vollständige Präordnung \preceq_i definiert. Die Präordnung auf A ergibt sich dann durch

$$a = (p_1, \dots, p_k) \preceq b = (q_1, \dots, q_k) \Leftrightarrow p_i \preceq_i q_i \text{ für alle } i = 1, \dots, k$$

Diese Präordnung \preceq auf A definiert offensichtlich einen Verband.

Dijkstra verwendet als Kegel $K_i (i = 1, \dots, k)$ die Menge der Funktionale $h \in H$, die bzgl. der durch die Profilgrößen p_i gegebenen vollständigen Präordnung monoton

⁷⁾die Einschränkung, daß es sich um Elemente handeln muß, die *poolable* sind, kann nach Satz 3 fallengelassen werden

sind; d.h.

$$K_i = \{h \in H \mid a = (p_1, \dots, p_k) \preceq b = (q_1, \dots, q_k), p_i \preceq_i q_i \Rightarrow h(a) \leq h(b)\}$$

Offensichtlich ist auch hier wieder $K = \bigcap_{i=1}^k K_i$ und die $P_{K_i}h$ können mittels des Verfahrens von Kruskal (1964a, b) (s.u.) für vollständige Präordnungen ermittelt werden. Die Menge der zu schneidenden Kegel ist hierbei wesentlich kleiner als bei der paarweisen Betrachtung von `ListPartOrder` und auch gegenüber `PartOrder`.

Für diesen Spezialfall wurde keine Klasse implementiert.

6.2 Hierarchische Präordnungen

Hierarchische Präordnungen werden in der Klasse `HierOrder` behandelt. Diese Klasse leitet sich von der Klasse `PartOrder` ab. Die Klasse setzt nachstehend beschriebene Situation voraus. Die Menge A sei in nichtleere sogenannte "Tieblöcke" $T_k \subseteq A$, ($k = 1, \dots, m$) unterteilt; d.h.:

$$\bigcup_{k=1}^m T_k = A$$

$$T_r \cap T_s = \emptyset \text{ für } r \neq s, \text{ und } r, s = 1, \dots, m$$

Die Präordnung \preceq erfülle folgende Eigenschaften:

Tie-Eigenschaft :

Für jeden Tieblock T_k ($k = 1, \dots, m$) gelte entweder

$$a \not\preceq b \text{ und } b \not\preceq a \text{ für alle } a, b \in T_k \quad (19)$$

oder

$$a \preceq b \text{ und } b \preceq a \text{ (d.h. } a \sim b) \text{ für alle } a, b \in T_k \quad (20)$$

Hierarchie-Eigenschaft :

Für alle Tieblöcke T_r, T_s ($r, s = 1, \dots, m$) mit $r \neq s$ gelte entweder

$$a \prec b \text{ für alle } a \in T_r, b \in T_s \quad (21)$$

oder

$$b \prec a \text{ für alle } a \in T_r, b \in T_s \quad (22)$$

oder

$$\text{es existiert kein } c \in A \text{ mit } a \preceq c \text{ und } b \preceq c \text{ für alle } a \in T_r, b \in T_s \quad (23)$$

Eine solche Präordnung wird als **hierarchisch** bezeichnet.

Die Relation \preceq sei durch die Relation \preceq' ersetzt, die die Relation innerhalb der Tieblöcke verändert:

1. Sind $a, b \in T_k (k = 1, \dots, m)$ und gilt (19), so sei:

$$a \prec' b \Leftrightarrow f(a) < f(b)$$

Ist $f(a) = f(b)$, so sei $a \prec' b$ zufällig gewählt.

2. Gilt (20), so sei

$$a \prec' b \Leftrightarrow f(a) > f(b)$$

Ist $f(a) = f(b)$, so sei $a \prec' b$ zufällig gewählt.

3. Für Paare a, b aus verschiedenen Tieblöcken ($a \in T_r, b \in T_s, r \neq s (r, s = 1, \dots, m)$) sei

$$a \prec' b \Leftrightarrow a \prec b$$

Bemerkung 2 Die Relation \preceq' ist wieder eine hierarchische Präordnung mit der Eigenschaft, daß

$$a \preceq' b \text{ und } b \preceq' a \Rightarrow a = b, \text{ bzw. } a \sim' b \Leftrightarrow a = b \quad (24)$$

Eine Funktion $g : A \rightarrow \mathbf{IR}$, die die Monotoniebedingung bzgl. \preceq' (1) und die Kleinst-Quadrate Bedingung bzgl. f (2) erfüllt, erfüllt diese Bedingungen auch bzgl. \preceq ; d.h. $g = P_K f$. Dies ist mit Satz 1 leicht zu sehen.

Schritt 0 :

Setze $g = f$ und alle Elemente $a \in A$ als nicht markiert.

Solange noch nicht alle Elemente von A markiert sind, wiederhole die folgenden Schritte 1 und 2a oder 2b:

Schritt 1 :

Suche ein nichtmarkiertes Element $a^* \in A$ mit $f(a^*) \leq f(a)$ für alle $a \in A$ und markiere dieses Element a^* .

Schritt 2a :

Hat a^* keinen unmittelbaren nicht markierten Vorgänger, so setze für alle nicht markierten Elemente $a \in A$, die a^* als unmittelbaren Vorgänger haben, deren Vorgänger als nicht vorhanden ($= \emptyset$).

Schritt 2b :

Lege den eindeutigen nichtmarkierten unmittelbaren Vorgänger $b \in A$ von a^* mit a^* zusammen; d.h. $g(b) = \frac{g(b)w(b)+g(a^*)w(a^*)}{w(b)+w(a^*)}$ und $w(b) = w(b) + w(a)$ und alle nichtmarkierten Elemente $a \in A$, die a^* als unmittelbaren Vorgänger haben, bekommen als Vorgänger jetzt b .

Bemerkung 3 Die Konstruktion von \preceq' in Verbindung mit der Hierarchiebedingung garantiert, daß \preceq' antisymmetrisch ist (s. Bemerkung 2) und es damit entweder keinen oder einen eindeutigen unmittelbaren Vorgänger für jedes Element gibt.

Der Algorithmus basiert offensichtlich auf der 1. und 3. Regel zur Reduzierung von A (s.o.) und kann immer angewandt werden, da es bei einer Hierarchie immer einen eindeutigen Vorgänger oder überhaupt keinen Vorgänger gibt.

6.3 vollständige Präordnungen

Die Klasse `TotalOrder` leitet sich von `HierOrder` ab, da eine vollständige Präordnung \preceq auf A trivialerweise auch eine hierarchische Präordnung darstellt. Zusätzlich bietet diese Klasse auch noch das sehr effiziente Verfahren von Kruskal an, welches algorithmisch nachstehend beschrieben ist.

Sei $A = \{a_1, \dots, a_n\}$ nach der vollständigen Präordnung numeriert; d.h. $a_i \preceq a_{i+1}$ ($i = 1, \dots, n-1$). Tieblöcke seien wie bei der hierarchischen Präordnung behandelt. $g = P_K f$ wird nun wie folgt berechnet (vgl. Opitz 1980, p.132):

Schritt 0 :

$$i = 1, g(a_i) = f(a_i)$$

Schritt 1 :

Wenn $f(a_i) \leq f(a_{i+1})$, setze $g(a_{i+1}) = f(a_{i+1})$, anderenfalls setze $j = 0$ und erhöhe j sukzessive um 1 bis entweder $j = i-1$ oder $g(a_{i-j-1}) \leq m(\{a_{i-j}, \dots, a_i, a_{i+1}\})$. Setze dann $g(a_{i-j+k}) = m(\{a_{i-j}, \dots, a_i, a_{i+1}\})$ für $k = 0, \dots, j+1$.

Schritt 2 :

Wiederhole Schritt 1 für $i = 2, \dots, n-1$

7 Abschätzungen

`HierOrder` und `TotalOrder` benötigen keine Fehlerabschätzungen, da sie die exakte Lösung ermitteln. Die durch das Dykstra-Verfahren in `PartOrder` und `ListPartOrder` ermittelten Ergebnisse sind aber nur Näherungslösungen und erfüllen u.U. nicht einmal die Monotoniebedingung (1). Hier gelten folgende Abschätzungen zur eindeutigen Optimallösung.

Satz 5 Sei $f = g + u$, $g \in K$, $u \in K^*$

$$\Rightarrow \langle P_K f - g, P_K f - g \rangle \leq -\langle u, g \rangle$$

Beweis: Sei $u^* = f - P_K f \Rightarrow u^* \in K^*, \langle u^*, P_K f \rangle = 0$ (14) und damit $\langle P_K f - g, P_K f - g \rangle = \langle P_K f - f + f - g, P_K f - g \rangle = \langle u - u^*, P_K f - g \rangle = \langle u, P_K f - g \rangle - \langle u^*, P_K f - g \rangle = \langle u, P_K f \rangle - \langle u, g \rangle - \langle u^*, P_K f \rangle + \langle u^*, g \rangle = -\langle u, g \rangle + (\langle u, P_K f \rangle + \langle u^*, g \rangle) \leq -\langle u, g \rangle$. \square

Ist nach Abbruch der durch den Algorithmus durchgeführten Iterationen das erhaltene Ergebnis g monoton bzgl. \preceq (1), so ist $g \in K$ und der obige Satz kann zur Abschätzung herangezogen werden. Ist g nicht monoton, so kann daraus ein monotonen g' konstruiert werden und folgender Satz kommt zum Tragen.

Satz 6 Sei \preceq' eine vollständige Präordnung, die mit der partiellen Präordnung \preceq verträglich ist, d.h. $a \preceq b \Rightarrow a \preceq' b$ für alle $a, b \in A$ ⁸⁾ und $g' = P_{K'} f$ mit $K' = \{h \in H \mid h \text{ monoton bzgl. } \preceq'\}$

$$\Rightarrow \langle P_K f - g', P_K f - g' \rangle \leq \|g - g'\| \|f\|$$

wobei $f = g + u$, $u \in K^*$ sei.

Beweis: Sei $u^* = f - P_K f \Rightarrow u^* \in K^*, \langle u^*, P_K f \rangle = 0$ (14). Entsprechendes gilt mit $u' = f - g' \in (K')^*$. Damit folgt: $\langle P_K f - g', P_K f - g' \rangle = \langle P_K f - f + f - g', P_K f - g' \rangle = \langle u' - u^*, P_K f - g' \rangle = \langle u', P_K f - g' \rangle - \langle u^*, P_K f - g' \rangle = \langle u', P_K f \rangle - \langle u', g' \rangle - \langle u^*, P_K f \rangle + \langle u^*, g' \rangle = \langle u', P_K f \rangle + \langle u^*, g' \rangle \leq \langle u', P_K f \rangle$ da $u^* \in K^* \subseteq (K')^*$ (11). Es gilt: $\langle u', P_K f \rangle = \langle u + u' - u, P_K f \rangle = \langle u, P_K f \rangle + \langle u' - u, P_K f \rangle \leq \langle u' - u, P_K f \rangle$ da $u \in K^*$. $\langle u' - u, P_K f \rangle \leq \|u' - u\| \|P_K f\| \leq \|u' - u\| \|f\|$ (6). Mit $u' - u = g - g'$ folgt die Behauptung. \square

Algorithmisch ist g wieder das nach einer Anzahl von Iterationen erhaltene Abbruchergebnis und g' die mit dem Verfahren von Kruskal errechnete Lösung für \preceq' . Die Präordnung \preceq' kann durch fortgesetzte Minimumsbildung von g unter Berücksichtigung von \preceq leicht hergestellt werden, wenn A mit \preceq keine äquivalenten Elemente enthält.

8 Die Behandlung von fehlenden Werten

Keine der implementierten Klassen verfügt über die explizite Behandlung von fehlenden Werten von f , auch "missing values" genannt. Naheliegender wäre es, das Gewicht $w(a)$ für ein Element $a \in A$, für das kein Funktionswert $f(a)$ existiert, gleich null zu setzen. $w(a) = 0$ führt aber sowohl zu theoretischen Problemen ($\|\cdot\|$ ist dann keine Norm mehr) als auch zu algorithmischen, denn das Zusammenlegen von zwei "missing values" führt dann zu einem $\frac{0}{0}$ Ausdruck.

Es sei deshalb folgendes Vorgehen vorgeschlagen: Man wählt für die Gewichte von Elementen, für die kein Funktionswert existiert, eine sehr kleine Zahl > 0 ; z.B.

⁸⁾die Existenz ist nach Barlow et al. 1972, p. 86, Theorem 2.9 gesichert. Es gibt sogar eine solche vollständige Präordnung, deren monotone Regression die optimale Lösung bzgl. \preceq ergibt (v. Eeden 1958)

10^{-6} . Den Funktionswert setzt man einmal auf $\min\{f(a) \mid a \in A, f(a) \text{ existiert}\} - 1$, diese Funktion sei f_- genannt und einmal auf $\max\{f(a) \mid a \in A, f(a) \text{ existiert}\} + 1$, diese Funktion sei f_+ genannt.

Für diese Funktionen ermittelt man jeweils $P_K f_-$ und $P_K f_+$ mit den oben angegebenen Algorithmen. Wie man aus der Verallgemeinerung des Problems auf vektorwertige Funktionen f u.a. weiß (Hansohm 2004), ist P_K ein monotoner Operator, d.h. aus $f_- \leq f_+$ folgt $P_K f_- \leq P_K f_+$. Hierbei ist die Monotonie für Funktionen, wie üblich, elementweise definiert; d.h. $r \leq s$ für $r, s \in H$ ist definiert als $r(a) \leq s(a)$ für alle $a \in A$.

Dadurch erhält man für die "missing values" ein Intervall von mit der Präordnung \preceq konsistenten Werten. Allerdings erhält man u.U. auch ein Intervall für die Elemente $a \in A$, für die Werte von f existieren. In diesem Fall muß man das Gewicht für die Elemente, für die f nicht definiert ist, weiter reduzieren. Gehen diese Gewichte gegen 0, so ist klar, daß

$$(P_K f_-)(a) = (P_K f_+)(a)$$

für alle $a \in A$, für die f definiert ist. Nach Kosmol 1974 ist für $a \in A$, für die f nicht definiert ist, $(P_K f_-)(a)$ der kleinstmögliche Wert mit $w(a) = 0$, wenn dieser Wert größer gleich $\min\{f(a) \mid a \in A, f(a) \text{ existiert}\}$ ist, ansonsten kann der Wert $= -\infty$ gesetzt werden. Entsprechendes gilt für $(P_K f_+)(a)$.

9 Beispiel

Als Beispiel dienen stratigraphische Daten einer römischen Badeanlage mit $|A| = 993$ archäologischen Schichten, von denen 340 datiert sind. Bei diesen Datierungen handelt es sich um ein Intervall, von dem erstmal nur die untere Grenze verwendet werden soll. Für die 993 Elemente sind 4378 Paarbeziehungen \preceq definiert, davon 1122 Äquivalenzrelationen.

Berechnet man die monotone Regression nur über die 340 Datierungen ohne Präferenzrelationen über Elemente mit fehlenden Werten zu berücksichtigen, so erhält man schon nach 100 Iterationen ein zufriedenstellendes Ergebnis. Der Algorithmus ergibt nach Abbruch zwar keine monotone Lösung, der Fehler zur Monotonie beträgt aber nur 0,0005 - für historische Jahresdaten also praktisch gleich 0. Die daraus gebildete monotone Lösung (*s.o.*) hat zur wahren optimalen Lösung lediglich eine Abweichung von ca. 0,0002 im Normquadrat, kann also als praktisch exakt gelten.

Schließt man die Elemente mit fehlenden Werten mit ein und wählt für diese ein Gewicht von 10^{-6} , einen Wert von -1 ⁹⁾ und für die Elemente mit existierenden Werten das Gewicht $\frac{1}{\text{Intervallbreite}+1}$, so ergeben sich nicht ganz so befriedigende Werte. Nach 1000 Iterationen ergab sich nach 8,64 Sekunden auf einem Pentium 4 Rechner mit 3 GHz ein zwar gut zu interpretierendes Ergebnis, das aber keine

⁹⁾das Minimum der unteren Grenze der Datierungen ist gleich null

sinnvolle Abschätzung zum Optimum zuließ. Bei dieser Problemgröße stößt das Verfahren numerisch offensichtlich an seine Grenzen und liefert vergleichbar gute Ergebnisse wie oben erst nach 100000 Iterationen.

10 Literaturverzeichnis

- AYER, M., BRUNK, H.D., EWING, G.M., REID, W.T., SILVERMAN, E. (1955): An Empirical Distribution Function for Sampling with Incomplete Information. *Ann. Math. Statist.*, 26, 641–647
- BARLOW, R.E., BARTHOLOMEW, D.J., BREMNER, J.M., BRUNK, H.D. (1972): *Statistical Inference under Order Restrictions*. John Wiley & Sons, 1972, New York, London, Sydney, Toronto.
- BRUNK, H.D., EWING, G.M., UTZ, W.R. (1957): Minimizing Integrals in Certain Classes of Monotone Functions. *Pacific J. of Math.*, 7, 833–847
- DYKSTRA, R.L. (1983): An Algorithm for Restricted Least Squares Regression. *J. Amer. Statist. Assoc.* 78, 837–842
- DYKSTRA, R.L., BOYLE, J.P. (1987): An Algorithm for Least Squares Projections onto the Intersection of Translated, Convex Cones. *J. Statist. Plann. Inf.*, 11, 391–400
- HANSOHN, J. (2003a): Multivariate Analysis for Variables of Arbitrary Information Level. In: Schader, M., Gaul, W., Vichi, M. (Eds.): *Between Data Science and Applied Data Analysis*. Springer, Berlin, Heidelberg, New York, 226–234.
- HANSOHN, J. (2003b): Eine neue Methode zur Visualisierung qualitativer Daten. In: Geyer-Schulz, A., Taudes, A. (Eds.): *Informationswirtschaft: Ein Sektor mit Zukunft*. Lecture Notes in Informatics (LNI), P-33, Gesellschaft für Informatik (GI), 2003, 285–294.
- HANSOHN, J. (2004): Monotone Regression on Partially Preordered Sets. *GfKI Tagungsband Dortmund*
- HERZOG, I. (1993): Computer-aided Harris Matrix Generation. *Practices of Archaeological Stratigraphy*, 1993, 201–217
- KOSMOL, P. (1974): *Optimierung konvexer Funktionen mit Stabilitäts-Betrachtungen*. Universität Kiel, 1974
- KRUSKAL, J.B. (1964a): Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis. *Psychometrika*, 29, 1964, 1–27
- KRUSKAL, J.B. (1964b): Nonmetric Multidimensional Scaling: A Numerical Method. *Psychometrika*, 29, 1964, 115–129
- LUENBERGER, D.G. (1973): *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, 1973

- OPITZ, O. (1980): *Numerische Taxonomie*. Grundwissen der Ökonomik, UTB, Gustav Fischer, Stuttgart, New York
- ROBERTSON, T., WRIGHT, F.T., DYKSTRA, R.L. (1988): *Order Restricted Statistical Interference*. John Wiley & Sons, 1988, New York, Brisbane, Toronto, Singapore.
- VAN EEDEN (1958): *Testing and Estimating Ordered Parameters of Probability Distributions*. Doctoral Dissertation, University Amsterdam, 1958
- YOUNG, F.W. (1975): Methods for Describing Ordinal Data with Cardinal Models. *Journal of Mathematical Psychology*, 12, 1975, 416–536.
- YOUNG, F. W., DE LEEUW, J., TAKANE, Y. (1976): Regression with Qualitative and Quantitative Variables: An Alternating Least Squares Method with Optimal Scaling Features. *Psychometrika*, 41, 4, 1976, 505–529.
- YOUNG, F.W., TAKANE, Y., DE LEEUW, J. (1978): The Principal Components of Mixed Measurement Level Multivariate Data: An Alternating Least Squares Method with Optimal Scaling Features. *Psychometrika*, 43, 1978, 279–281.
- YOUNG, F.W. (1981): Quantitative Analysis of Qualitative Data. *Psychometrika*, 46, 4, 1981, p. 357–388.