

zu durchschauen sind, dass sie von vornherein von den Personen ausgeschlossen werden.

In diesem Fall könnte es sein, dass die echte Ratewahrscheinlichkeit für eine Aufgabe sogar höher ist als $\frac{1}{3}$. Deshalb ist es oft interessanter, die Ratewahrscheinlichkeit nicht vorzugeben, oder für alle Aufgaben mit der gleichen Anzahl von Antwortalternativen dieselbe Ratewahrscheinlichkeit anzunehmen, sondern die Ratewahrscheinlichkeit γ_j für jede Aufgabe einzeln zu schätzen. Dies ist allerdings nur möglich, wenn genug Personen die Aufgaben beantwortet haben.

Genauso wie das 2PL-Modell ein Spezialfall des 3PL-Modells ist, bei dem die Ratewahrscheinlichkeit 0 ist, kann man auch ein zwei-parametriges Modell als Spezialfall des 3PL-Modells konstruieren, das eine positive Ratewahrscheinlichkeit zulässt, bei dem aber wieder alle Aufgaben dieselbe Trennschärfe haben, wie beim Rasch-Modell – also ein Rasch-Modell mit Rate-Parameter(n). 3PL- und 2PL-Modelle können mithilfe des marginalen Maximum-Likelihood-Ansatzes z.B. mit dem R-Paket `ltm` (Rizopoulos, 2011) geschätzt werden.

5.3 Modelle mit mehrstufigen Antwortkategorien

Bisher haben wir immer den Fall betrachtet, dass Personen eine Aufgabe lösen (kodiert mit 1) oder nicht lösen (kodiert mit 0), bzw. einer Aussage zustimmen (1) oder nicht zustimmen (0). Die folgenden Modelle stellen eine Erweiterung dieser möglichen Antwortkategorien dar: Nun soll es möglich sein, bei jeder Aufgabe z.B. zwischen 0 und 3 Punkten zu erhalten.

In einem Leistungstest kann man sich z.B. vorstellen, dass eine Person null Punkte erhält, wenn sie die Aufgabe gar nicht oder völlig falsch bearbeitet hat, einen Punkt, wenn sie einen Teil der Lösung richtig hat usw.; drei Punkte erhält die Person nur, wenn sie die Lösung komplett richtig ausgeführt hat. Auch bei Einstellungstests ist es üblich, die Personen auf einer ordinalen Skala eintragen zu lassen, inwieweit sie einer Aussage zustimmen (z.B. von „stimme überhaupt nicht zu“ bis „stimme voll zu“).

Die folgenden Modelle sind alle zur Auswertung von Tests mit solchen mehrstufigen Antwortkategorien konzipiert.

5.3.1 Das Partial-Credit-Modell

Beispielhaft soll hier das Partial-Credit-Modell von Masters (1982) ausführlicher vorgestellt werden, aber alle Modelle für mehrstufige Antwortkategorien folgen in ihrer Formulierung einem ähnlichen Prinzip: Anstatt einer gemeinsamen Modellgleichung, die die Lösungswahrscheinlichkeit für eine Person i und eine Aufgabe j beschreibt, braucht man jetzt, wo jede Aufgabe j mit einer von mehreren Antwortkategorien beantwortet werden kann, eine eigene Gleichung für jede Antwortkategorie c jeder Aufgabe j .

In unterschiedlichen Büchern und Artikeln wird das Partial-Credit-Modell oft mit unterschiedlichen Formeln dargestellt, die sich auf den ersten Blick nicht sonderlich ähnlich sehen. Deshalb werden im Folgenden die unterschiedlichen Darstellungsweisen zueinander in Beziehung gesetzt und erklärt. (Außerdem scheint das Partial-Credit-Modell Tippfehler geradezu magisch anzuziehen. In fast allen Büchern und Artikeln dazu habe ich mindestens einen Fehler in einer Formel gefunden – und natürlich umso mehr versucht, selber keine zu machen. Aber Menschen die Bücher schreiben sind eben auch nur Menschen, d.h. wenn Sie in unterschiedlichen Büchern und Artikeln Darstellungen des Partial-Credit-Modells entdecken, die sich nicht genau ineinander überführen lassen, liegt es vermutlich nicht an Ihnen...)

Die Herleitung des Partial-Credit-Modells nach Masters (1982) folgt der Idee, dass die Wahrscheinlichkeit des Übergangs von einer Antwortkategorie $c-1$ zur nächsten Antwortkategorie c mithilfe des Rasch-Modells beschrieben wird. Genauer gesagt beschreibt das Partial-Credit-Modell die bedingte Wahrscheinlichkeit dafür, dass die Antwort in die höhere Kategorie c fällt, unter der Bedingung, dass die Antwort in einer der beiden Kategorien $c-1$ oder c fällt, durch ein Rasch-Modell:

Für eine Aufgabe j mit den Antwortkategorien $0, \dots, m_j$ (bei der z.B. 0, 1, 2 oder 3 Punkte erzielt werden können, so dass es insgesamt $m_j + 1 = 4$ Kategorien gibt) ist die bedingte Wahrscheinlichkeit, dass die Antwort von Person i bei Aufgabe j in die Kategorie c fällt (unter der Bedingung, dass sie in eine der beiden Kategorien $c-1$ oder c fällt):

$$P(u_{ij} = c | u_{ij} \in \{c-1, c\}, \theta_i, \delta_{jc}) = \frac{e^{\theta_i - \delta_{jc}}}{1 + e^{\theta_i - \delta_{jc}}}$$

In dieser Formel gibt es einen eigenen Parameter δ_{jc} für jede Kategorie c der Aufgabe j . Ansonsten entspricht die Form der des Rasch-Modells aus den früheren Kapiteln.

Aus dieser bedingten Wahrscheinlichkeit leitete Masters (1982) die unbedingte Wahrscheinlichkeit dafür ab, dass die Antwort in die Kategorie c fällt:

$$P(u_{ij} = c | \theta_i, \delta_{j1}, \dots, \delta_{jm_j}) = \frac{e^{\sum_{k=0}^c (\theta_i - \delta_{jk})}}{\sum_{l=0}^{m_j} e^{\sum_{k=0}^l (\theta_i - \delta_{jk})}}$$

wobei zur Berechnung festgelegt wird, dass $\sum_{k=0}^0 (\theta_i - \delta_{jk}) = 0$ und entsprechend $\sum_{k=0}^l (\theta_i - \delta_{jk}) = \sum_{k=1}^l (\theta_i - \delta_{jk})$.

Diese Modellgleichung ist komplizierter als die für das einfache Rasch-Modell, weil über die verschiedenen Kategorien der Aufgabe Summen gebildet werden müssen, um die gesuchte Wahrscheinlichkeit zu berechnen. Außerdem gilt die Gleichung nicht für eine ganze Aufgabe, wie im Rasch-Modell, sondern nur für eine einzelne Kategorie einer Aufgabe.

Entsprechend gibt es im Partial-Credit-Modell auch nicht nur eine ICC pro Aufgabe, sondern es gibt so viele wie die Aufgabe Kategorien hat; diese Kurven nennt man daher auch „Category Characteristic Curves“ oder CCCs. Die

CCCs des Partial-Credit-Modells für eine Aufgabe mit vier Antwortkategorien und den entsprechenden Parametern δ_{jk} sind in Abbildung 5.3 dargestellt. Die einzelnen Kurven entsprechen dabei den Wahrscheinlichkeiten für die einzelnen Antwortkategorien: Von links nach rechts betrachtet ist die Wahrscheinlichkeit für die erste Antwortkategorie (0 Punkte) für niedrige Werte der Fähigkeit hoch und sinkt für höhere Werte der Fähigkeit ab. Im Gegensatz dazu sind die Wahrscheinlichkeiten für die mittleren Kategorien (1 und 2 Punkte) zunächst niedrig, bei dem der Kategorie entsprechenden Fähigkeitsniveau hoch und darüber hinaus wieder niedrig. Die Wahrscheinlichkeit für die letzte Kategorie (3 Punkte) hingegen ist zunächst niedrig und steigt für höhere Werte der Fähigkeit an.

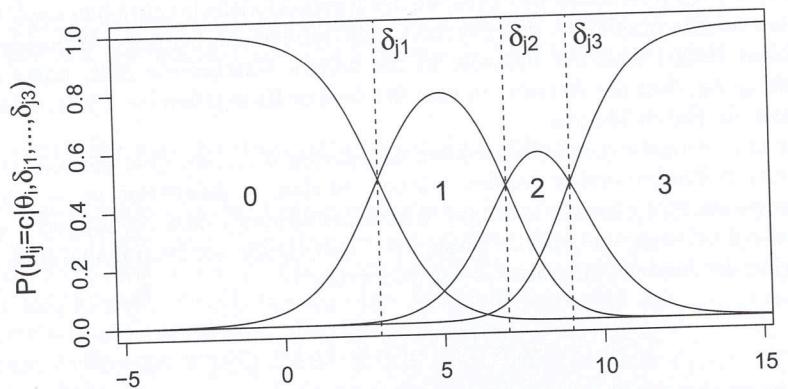


Abb. 5.3. CCCs für das Partial-Credit-Modell mit eingezeichneten δ -Parametern.

Für eine Person mit der Fähigkeit 5 kann man z.B. in Abbildung 5.3 ablesen, dass sie mit der höchsten Wahrscheinlichkeit in die zweite Kategorie fällt (d.h. 1 Punkt erzielt), während die Wahrscheinlichkeiten für die benachbarten Kategorien (0 und 2 Punkte) deutlich geringer sind und die Wahrscheinlichkeit für die höchste Kategorie (3 Punkte) fast bei 0 liegt. Die Wahrscheinlichkeiten für alle vier Kategorien zusammen addieren sich dabei für jedes Fähigkeitsniveau zum Wert 1 auf. Neben den Wahrscheinlichkeiten für die einzelnen Antwortkategorien sind in Abbildung 5.3 auch die Schwellenwerte auf der Fähigkeits-Skala (als gestrichelte Linien) eingezeichnet, die die Schnittpunkte zwischen den Kurven für die einzelnen Kategorien markieren: Eine Person mit der Fähigkeit 5 liegt z.B. zwischen den beiden Schwellenwerten, die die zweite Kategorie von der ersten und dritten Kategorie trennen. Für alle Personen zwischen diesen beiden

Schwellenwerten hat die zweite Kategorie die höchste Wahrscheinlichkeit für eine einzelne Kategorie.

Die Lage der Schwellenwerte entspricht genau den Parametern δ_{jk} aus der obigen Modellgleichung. Sie werden deshalb auch als Schwellenparameter bezeichnet. Dabei liegt z.B. δ_{j1} am Schnittpunkt zwischen Kategorie 0 und der nächst höheren Kategorie 1, δ_{j2} am Schnittpunkt zwischen Kategorie 1 und der nächst höheren Kategorie 2 etc., d.h. die Anzahl der Schwellenparameter ist um 1 kleiner als die Anzahl der Kategorien.

Die Schwellenparameter können in zwei andere Arten von Parametern umgerechnet werden, die dieselbe Information enthalten, aber anders zu interpretieren sind. (Die Umrechnung von einer Art von Parametern in eine andere nennt man Reparametrisierung. Dabei müssen die Parameterwerte nicht neu aus den Daten geschätzt werden, sondern es gibt eine eindeutige Rechenregel, mit der man zwischen den unterschiedlichen Parametrisierungen hin- und her-rechnen kann. Dadurch ändert sich aber auch das Aussehen der Modellgleichung, so dass man das Modell anhand der neuen Formel auf den ersten Blick manchmal nur schwer wiedererkennen kann.)

Eine häufig verwendete Parametrisierung des Partial-Credit-Modells ergibt sich aus der Darstellung mit den Schwellenparametern durch Aufsummieren von $\lambda_{jc} = \sum_{k=1}^c \delta_{jk}$:

$$\begin{aligned} P(u_{ij} = c | \theta_i, \delta_{j1}, \dots, \delta_{jm_j}) &= \frac{e^{\sum_{k=0}^c (\theta_i - \delta_{jk})}}{\sum_{l=0}^{m_j} e^{\sum_{k=0}^l (\theta_i - \delta_{jk})}} \\ &\stackrel{(1)}{=} \frac{e^{c \cdot \theta_i - \sum_{k=1}^c \delta_{jk}}}{\sum_{l=0}^{m_j} e^{l \cdot \theta_i - \sum_{k=1}^l \delta_{jk}}} \\ &\stackrel{(2)}{=} \frac{e^{c \cdot \theta_i - \lambda_{jc}}}{\sum_{l=0}^{m_j} e^{l \cdot \theta_i - \lambda_{jl}}} \end{aligned}$$

wobei $\lambda_{j0} = 0$. Die Darstellung der Modellgleichung in der letzten Zeile findet sich in vielen Büchern (wenn auch natürlich oft mit anderen griechischen Buchstaben anstelle von θ_i und λ_{jc}), weil die Formel weniger Summenzeichen enthält und dadurch einfacher aussieht. Zur Interpretation sind die kumulierten Schwellenparameter λ_{jc} aber nicht sonderlich gut geeignet.

Eine weitere mögliche Parametrisierung des Partial-Credit-Modells, die wieder relativ einfach zu interpretieren ist, ergibt sich aus den Schwellenparametern durch die Umrechnung in einen zentralen Lageparameter $\bar{\delta}_j = \sum_{k=1}^{m_j} \delta_{jk}/m_j$, dem Mittelwert der ursprünglichen Schwellenparameter, und den m_j Abweichungen der einzelnen Schwellenparameter von diesem Mittelwert, $\tau_{jk} = \bar{\delta}_j - \delta_{jk}$ (wobei von den m_j Differenzen τ_{jk} nur $m_j - 1$ frei variieren können):

⁽¹⁾ wegen $\sum_{k=0}^l (\theta_i - \delta_{jk}) = \sum_{k=1}^l (\theta_i - \delta_{jk})$ (wurde oben bei der Darstellung der Modellgleichung mit Schwellenparametern festgelegt) ist:

⁽²⁾ $\sum_{k=0}^c (\theta_i - \delta_{jk}) = \sum_{k=1}^c (\theta_i - \delta_{jk}) = \sum_{k=1}^c \theta_i - \sum_{k=1}^c \delta_{jk} = c \cdot \theta_i - \sum_{k=1}^c \delta_{jk}$

$$P(u_{ij} = c | \theta_i, \bar{\delta}_j, \tau_{j1}, \dots, \tau_{jm_j}) = \frac{e^{\sum_{k=0}^c (\theta_i - (\bar{\delta}_j - \tau_{jk}))}}{\sum_{l=0}^{m_j} e^{\sum_{k=0}^l (\theta_i - (\bar{\delta}_j - \tau_{jk}))}}$$

Die CCCs des Partial-Credit-Modells in der Darstellung mit dem zentralen Lageparameter $\bar{\delta}_j$ und den Differenzen τ_{jk} sind in Abbildung 5.4 dargestellt.

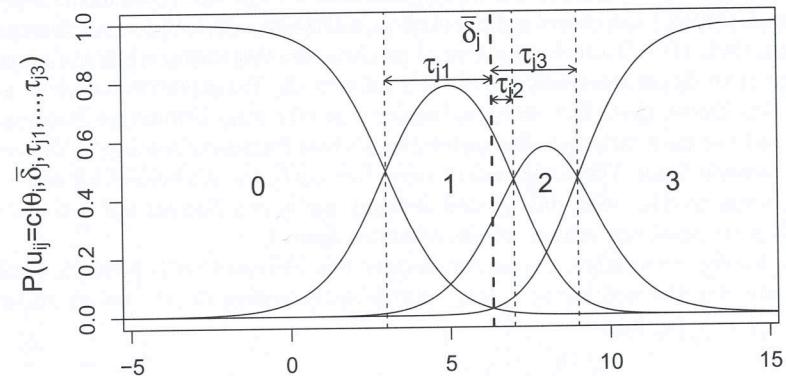


Abb. 5.4. CCCs für das Partial-Credit-Modell mit eingezeichneten $\bar{\delta}$ - und τ -Parametern.

Ein Vorteil dieser Parametrisierung ist, dass sich das Partial-Credit-Modell in dieser Form sehr gut mit dem Rating-Scale-Modell vergleichen lässt, das im nächsten Abschnitt vorgestellt wird. Ein weiterer (vermeintlicher) Vorteil ist, dass der Parameter $\bar{\delta}_j$ die zentrale Lage der Aufgabe auf dem Fähigkeits- bzw. Schwierigkeits-Kontinuum ausdrückt, und man damit die Lage der Aufgabe mit einem einzelnen Wert beschreiben kann. Genau darin liegt aber auch das Risiko für Fehlinterpretationen bei dieser Parametrisierung: Eine Aufgabe mit einem höheren $\bar{\delta}_j$ kann man nicht einfach als schwieriger bezeichnen als eine Aufgabe mit einem niedrigeren $\bar{\delta}_j$ (wie es im Rasch-Modell für Aufgaben mit unterschiedlicher Schwierigkeit β_j möglich ist), weil bei unterschiedlichen Aufgaben die Schwellenparameter zwischen den Kategorien unterschiedlich weit auseinanderliegen können, wie in Abbildung 5.5 dargestellt.

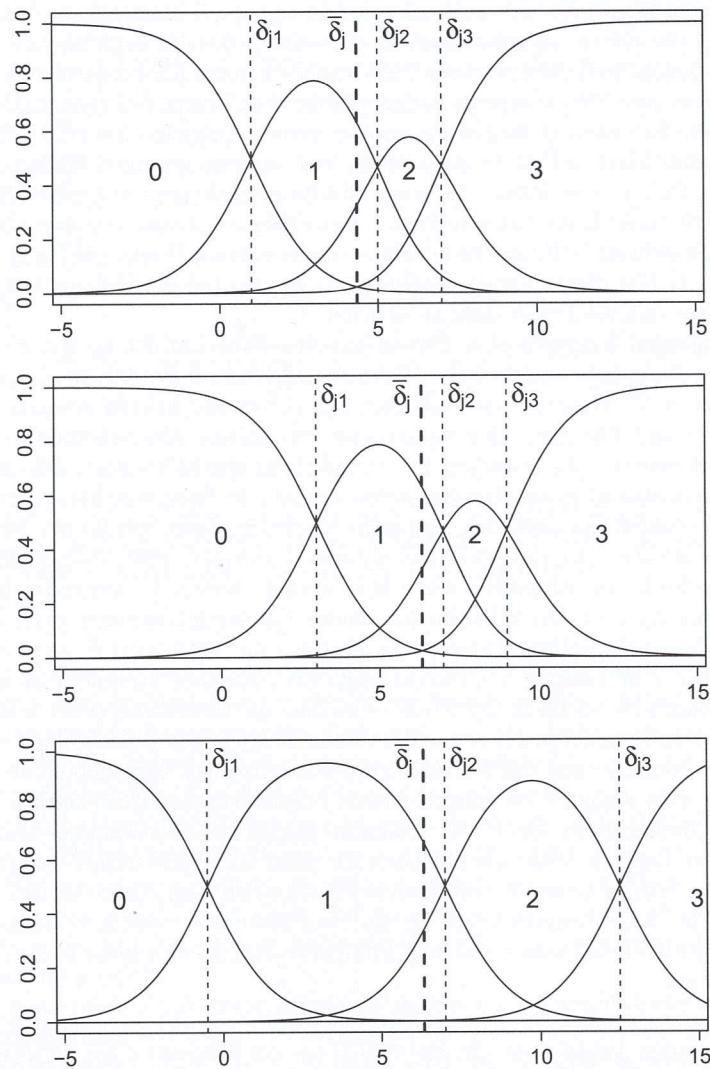


Abb. 5.5. CCCs für drei Aufgaben mit unterschiedlichen δ -Parametern: Vergleicht man die erste mit der zweiten Aufgabe, so liegen sowohl der zentrale Lageparameter $\bar{\delta}_j$ als auch jeder einzelne Schwellenparameter δ_{jk} bei der ersten Aufgabe weiter links als bei der zweiten Aufgabe. Die beiden Aufgaben lassen sich also in eine eindeutige Reihenfolge bringen. Vergleicht man hingegen die erste mit der dritten Aufgabe, so liegt der zentrale Lageparameter $\bar{\delta}_j$ zwar auch bei der dritten Aufgabe weiter rechts als bei der ersten Aufgabe, die Schwellenparameter zwischen den Kategorien liegen aber auch weiter auseinander. Dadurch lässt sich keine eindeutige Reihenfolge zwischen den beiden Aufgaben herstellen.

Auch formal kann man sich anhand von Abbildung 5.5 klarmachen, dass die erste und die dritte Aufgabe nicht in eine eindeutige Reihenfolge gebracht werden können, weil die erwartete Punktezahl^① je nach Fähigkeitsniveau mal für die eine, mal für die andere Aufgabe höher ist: Betrachtet man z.B. eine Person mit Fähigkeit 0, so hat sie bei der ersten Aufgabe eine relativ hohe Wahrscheinlichkeit, 0 Punkte zu erzielen, und nur eine geringe Wahrscheinlichkeit, 1 Punkt zu erzielen – die erwartete Punktezahl liegt also näher bei 0. Bei der dritten Aufgabe hat eine Person mit Fähigkeit 0 hingegen eine höhere Wahrscheinlichkeit, 1 Punkt zu erzielen – die erwartete Punktezahl liegt also näher bei 1. Für dieses Fähigkeitsniveau ist es also bei der dritten Aufgabe leichter, eine höhere Punktezahl zu erzielen.

Betrachtet man hingegen eine Person mit der Fähigkeit 10, so hat sie bei der ersten Aufgabe eine sehr hohe Wahrscheinlichkeit, 3 Punkte zu erzielen – die erwartete Punktezahl liegt also nahe bei 3. Bei der dritten Aufgabe hat eine Person mit Fähigkeit 10 hingegen eine viel höhere Wahrscheinlichkeit, 2 Punkte zu erzielen – die erwartete Punktezahl liegt also näher bei 2. Für dieses Fähigkeitsniveau ist es also bei der ersten Aufgabe leichter, eine höhere Punktezahl zu erzielen. Es lässt sich also nicht eindeutig sagen, welche der beiden Aufgaben leichter ist (wie formal bei Sijtsma & Hemker, 1998, dargestellt). Beim Vergleich von Aufgaben sollte man deshalb neben $\bar{\delta}_j$ immer auch die Differenzen τ_{jk} oder die Schwellenparameter δ_{jk} berücksichtigen (z.B. können sowohl die Schwellenparameter δ_{jk} als auch ihr Mittelwert $\bar{\delta}_j$ angegeben werden, wie in Abbildung 5.5, um die Lage einer Aufgabe zu beschreiben). Häufig kommt es jedoch in der Praxis vor, dass die Schätzungen der Schwellenparameter δ_{jk} nicht in der erwarteten Reihenfolge angeordnet sind, da diese durch die Formulierung des Partial-Credit-Modells nicht vorgegeben ist. Die CCCs für eine Aufgabe mit ungeordneten Schwellenparametern sind in Abbildung 5.6 dargestellt. Bei dieser Aufgabe erzielen nur sehr wenige Personen 1 Punkt, so dass die Wahrscheinlichkeit für diese Kategorie immer unterhalb der Wahrscheinlichkeiten für die anderen Kategorien liegt, und sich die Reihenfolge der Schwellenparameter δ_{j1} und δ_{j2} umkehrt – wodurch auch ihre Interpretation als Schwellen auf dem Fähigkeits-Kontinuum nicht mehr sinnvoll erscheint.

^① Die erwartete Punktezahl (d.h. der Erwartungswert) lässt sich einfach berechnen: Bei der ersten Aufgabe hat z.B. eine Person mit der Fähigkeit 0 (an der x-Achse ablesen) eine Wahrscheinlichkeit von ca. 73% (an der y-Achse ablesen – wobei man das an Abbildung 5.5 nicht so genau sehen kann wie es hier berechnet ist), 0 Punkte zu erzielen, eine Wahrscheinlichkeit von ca. 27%, 1 Punkt zu erzielen, und eine Wahrscheinlichkeit von je ca. 0%, 2 oder 3 Punkte zu erzielen. Die erwartete Punktezahl für eine Person mit der Fähigkeit 0 bei der ersten Aufgabe ist also $E(U_{i1}|\theta_i = 0) = 0, 73 \cdot 0 + 0, 27 \cdot 1 + 0 \cdot 2 + 0 \cdot 3 = 0, 27$, u.s.w.:

	Fähigkeit 0	Fähigkeit 10
Aufgabe 1	$E(U_{i1} \theta_i = 0) = 0, 27$	$E(U_{i1} \theta_i = 10) = 2, 95$
Aufgabe 3	$E(U_{i3} \theta_i = 0) = 0, 62$	$E(U_{i3} \theta_i = 10) = 2, 03$

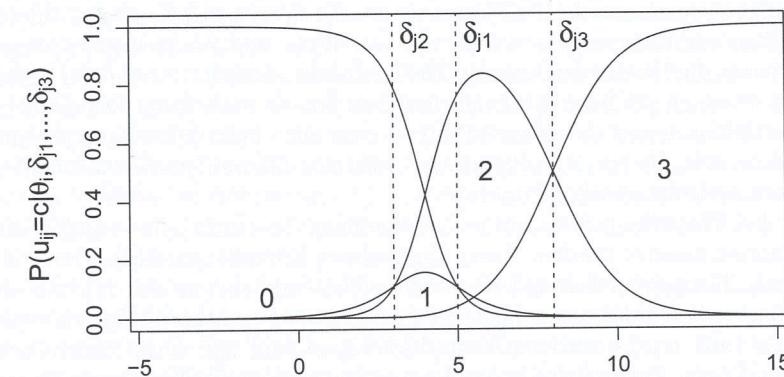


Abb. 5.6. CCCs für eine Aufgabe mit ungeordneten Schwellenparametern: δ_{j1} , der Schnittpunkt der Kurven für Kategorie 0 und Kategorie 1, liegt weiter rechts als δ_{j2} , der Schnittpunkt der Kurven für Kategorie 1 und Kategorie 2.

Über die formale und inhaltliche Bedeutung der ungeordneten Schwellenparameter gehen die Meinungen in der Fachwelt auseinander: Andrich (2010 und in früheren Arbeiten) wertet ihr Auftreten als Indiz dafür, dass die Antwortkategorien der betroffenen Aufgabe nicht in der intendierten, ordinalen Weise verwendet werden. Von diesem Standpunkt aus betrachtet, der von vielen anderen Autoren geteilt bzw. übernommen wurde (vgl. z.B. Rost, 2004), bietet das Partial-Credit-Modell also eine Möglichkeit, die intendierte Ordnung der Kategorien empirisch zu überprüfen. Adams, Wu und Wilson (2012) argumentieren jedoch gegen diese Sichtweise und führen zwei Definitionen von Ordnung an, die das Partial-Credit-Modell auch bei ungeordneten Schwellenparametern erfüllt.

Einig sind sich die Autoren allerdings darüber, dass ungeordnete Schwellenparameter darauf hinweisen bzw. daraus resultieren, dass die betroffenen Antwortkategorien in den Daten nur selten auftreten. Ursache hierfür ist oft eine ungeeignete Formulierung der Antwortkategorien bzw. des vorgegebenen Bewertungsschemas, wie z.B. Van Wyke und Andrich (2006) an der Beurteilung von geometrischen Zeichnungen von Schulkindern der dritten Klasse illustrieren:

Die Aufgabenstellung lautete, in einer Zeichnung mit mehreren Gänseblümchen einige davon mithilfe des Lineals zu einem Rechteck zu verbinden. Das Bewertungsschema sah vor, 0 Punkte zu vergeben, wenn die Antwort völlig unzureichend war, 1 Punkt, wenn zwar die richtigen Gänseblümchen ausgewählt wurden, das Rechteck aber schlecht gezeichnet war (z.B. mit ungeraden

oder nicht parallelen Seiten) und 2 Punkte, wenn das Rechteck komplett richtig eingezeichnet war. In der Praxis wurde die mittlere Kategorie (1 Punkt) allerdings wohl nie vergeben, weil, wie Van Wyke und Andrich (2006) argumentieren, die Kinder das Konzept des Rechtecks entweder verstanden hatten (dann machten sie beim Zeichnen mit dem Lineal auch keine Fehler mehr, und erzielten immer die vollen 2 Punkte) oder eben nicht (aber dann konnten sie auch nicht die für 1 Punkt nötigen richtigen Gänseblümchen auswählen, sondern erzielten gleich 0 Punkte).

Auch bei Einstellungstests kann es vorkommen, dass nicht alle vorgegebenen Kategorien genutzt werden. Wenn ungeordnete Schwellenparameter bzw. ungenutzte Kategorien vermieden werden sollen empfiehlt es sich also für die praktische Anwendung von mehrstufigen Antwortformaten besonders, einen Pretest (mit repräsentativen Versuchspersonen und ggf. auch Beurteilern) durchzuführen. Fallen dabei in der Auswertung mit dem Partial-Credit-Modell bereits ungeordnete Schwellenparameter auf, können die Antwortkategorien bzw. das Bewertungsschema noch entsprechend angepasst werden.

Wurden die Daten hingegen bereits erhoben, wird als pragmatische Lösung meist empfohlen, die betroffene Kategorie mit einer oder mehreren benachbarten Kategorien zusammenzufassen. Dazu müssen zunächst die Daten umkodiert, und dann das Partial-Credit-Modell neu geschätzt werden. Adams et al. (2012) raten allerdings vom routinemäßigen Zusammenlegen von Antwortkategorien nur aufgrund von ungeordneten Schwellenparametern ab.

Das Partial-Credit-Modell enthält das Rasch-Modell als Spezialfall mit zwei Antwortkategorien. Auch beim Partial-Credit-Modell haben alle Aufgaben dieselbe Trennschärfe und es existieren suffiziente Statistiken für alle Parameter, so dass es durch bedingte Maximum-Likelihood-Schätzung z.B. mit dem R-Paket **eRm** (Mair et al., 2011) geschätzt werden kann.

Eine Vielzahl von weiteren Modellen für mehrstufige Antwortkategorien wurde z.T. schon vor dem Partial-Credit-Modell vorgeschlagen. Die bekanntesten sollen im Folgenden noch kurz aufgeführt werden.

5.3.2 Das Rating-Scale-Modell

Das Rating-Scale-Modell von Andrich (1978) stellt einen Spezialfall des Partial-Credit-Modells dar, bei dem alle Aufgaben das gleiche Antwortformat aufweisen, d.h. insbesondere die gleiche Anzahl von Antwortkategorien.

Auch die Abstände zwischen den Schwellenparametern sind im Rating-Scale-Modell für alle Aufgaben gleich – im Gegensatz zum Partial-Credit-Modell, bei dem sich die Anzahl der Kategorien sowie deren Breite von Aufgabe zu Aufgabe unterscheiden können. Dies sieht man am besten in der Darstellung mit den Parametern $\bar{\delta}_j$ und τ_1, \dots, τ_m :

$$P(u_{ij} = c | \theta_i, \bar{\delta}_j, \tau_1, \dots, \tau_m) = \frac{e^{\sum_{k=0}^c (\theta_i - (\bar{\delta}_j - \tau_k))}}{\sum_{l=0}^m e^{\sum_{k=0}^l (\theta_i - (\bar{\delta}_j - \tau_k))}}$$

Jede Aufgabe j erhält dabei ihren eigenen zentralen Lageparameter $\bar{\delta}_j$. Sowohl die Anzahl der Kategorien $0, \dots, m$, als auch die Abweichungen der Schwellen vom zentralen Lageparameter τ_1, \dots, τ_m sind aber nun für alle Aufgaben gleich (und haben deshalb auch keinen Index mehr für die Aufgabe j , wie beim Partial-Credit-Modell auf S. 58 oben). D.h. die Aufgaben können je nach ihrem Wert für $\bar{\delta}_j$ weiter rechts oder weiter links liegen, die Abstände zwischen den Kategorien sind aber für jede Aufgabe gleich (wie bei den oberen beiden Aufgaben in Abbildung 5.5). In diesem Sinne lassen sich beim Rating-Scale-Modell die Aufgaben auch eindeutig bzgl. ihrer Schwierigkeit anordnen (Sijtsma & Hemker, 1998).

Auch im Rating-Scale-Modell kann es theoretisch passieren, dass sich die Reihenfolge der Schwellenparameter umkehrt, weil einzelne Kategorien nur selten genutzt werden. Da die Schätzung der Schwellenparameter (bzw. der Differenzen τ_1, \dots, τ_m) allerdings über alle Aufgaben hinweg erfolgt, wirkt sich eine Abweichung in einzelnen Aufgaben meist nicht merklich auf die aus allen Aufgaben geschätzten Schwellenparameter aus – wodurch ungeeignete Antwort- bzw. Bewertungsschemata ggf. unbemerkt bleiben können.

Die Entscheidung, ob ein Partial-Credit-Modell oder ein Rating-Scale-Modell zur Auswertung eines Tests verwendet werden soll, ist nicht immer ganz einfach. Wenn sich die Anzahl der Kategorien zwischen den Aufgaben unterscheidet, oder sich bei der Anpassung des Partial-Credit-Modells zeigt, dass die Breite der Kategorien von Aufgabe zu Aufgabe sehr unterschiedlich ist, sollte das Partial-Credit-Modell verwendet werden. Andererseits können inhaltliche Überlegungen für die Verwendung des Rating-Scale-Modells sprechen, z.B. wenn für alle Aufgaben dasselbe Antwortformat vorgegeben wurde (z.B. eine 5-stufige Likert-Skala mit denselben Labels von „stimme überhaupt nicht zu“ bis „stimme voll zu“).

Eine weitere empirische Entscheidungshilfe können Informationskriterien (wie das Akaike Information Criterion AIC und das Bayesian Information Criterion BIC; vgl. z.B. Tutz, 2000; Fahrmeir, Kneib & Lang, 2007) liefern, die sowohl die Güte der Anpassung an die Daten als auch die Anzahl der Parameter (d.h. der Komplexität des Modells) berücksichtigen. Oft kommt es dabei allerdings zu Widersprüchen zwischen dem AIC und dem BIC, weil das BIC die Anzahl der Parameter stärker bestraft und deshalb meist das sparsamere Modell bevorzugt (wie z.B. bei Baghaei, 2010).

Das Rating-Scale-Modell kann, wie das Partial-Credit-Modell, z.B. mit dem R-Paket **eRm** (Mair et al., 2011) geschätzt werden, das auch Informationskriterien zum Vergleich von Modellen ausgibt.

5.3.3 Das Graded-Response-Modell

Ein weiteres bekanntes Modell für mehrstufige Antwortkategorien ist das Graded-Response-Modell (Samejima, 1969). Bei diesem Modell handelt es sich um ein sogenanntes kumulatives Modell (vgl. Tutz, 2000). Modelliert wird nämlich die kumulierte Wahrscheinlichkeit dafür, dass eine Person mindestens

3

UNIDIMENSIONAL IRT WITH POLYTOMOUS ITEM RESPONSES

Item responses could be more than dichotomous outcomes. Responses for constructed-response partial credit items and Likert style responses are scored polytomously. Numbers assigned for different categorically ordered responses are typically 0, 1, 2, and 3 (or 1, 2, 3 and 4) for a four-option or four-level set of categories in an item. This chapter introduces the applications of the IRT models that handle polytomously scored item responses.

In a model for dichotomously scored item responses, the item response function (IRF) is the essential unit of modeling the item responses. In polytomous response modeling, the basic unit of modeling is at the category level, which is the category response function (CRF), or category probability function. Just as a dichotomous IRF can be graphically displayed using an item characteristic curve (ICC), a polytomous CRF can be graphically displayed using a category characteristic curve (CCC), or category probability curve. (There is one exception to this; the graded response model does utilize the cumulative category response probabilities. This is further explained in section 3.4.)

Four different polytomous IRT models will be discussed in this chapter: partial credit model (PCM), rating scale model (RSM), generalized partial credit model (GPCM), and graded response model (GRM). The nominal response model (NRM) is briefly presented at the end. The IRT packages used within the R software in this chapter are the “eRm,” “ltm,” and “mirt.” For the PCM, “eRm” and “mirt” were used. For the RSM, “mirt” was used. Both the “ltm” and “mirt” were used for the GPCM and the GRM. The NRM is discussed using “mirt.”

“eRm” has the capability to fit the RSM, but it was not considered here because the authors have experienced model estimation problems with warning messages when multiple data sets were tried. “ltm” has an option for specifying CCC of PCM, but it appears to fix the variance of the population ability distribution as one, which makes a more reduced model than traditional PCM in the marginal maximum likelihood (MML) estimation context which “ltm” employs.

3.1 Partial credit model (PCM) application

The partial credit model (PCM; Masters, 1982) is a Rasch family item response model that can handle item responses with more than two categories. The PCM is a direct extension of the dichotomous simple Rasch model, and it is classified as a divided-by-total model (Thissen & Steinberg, 1986). The CRF, and CCC, for category $k \in \{0, 1, 2, \dots, m_i\}$, i.e., responding in category k of an item with $m_i + 1$ possible category options, starting with 0, of the i th item has the following form.

$$P(X_{ikj} = k | \theta_j) = \frac{\exp \sum_{h=0}^k (\theta_j - b_{ih})}{\sum_{c=0}^{m_i} \exp \sum_{h=0}^c (\theta_j - b_{ih})}, \quad (3.1_1)$$

where X_{ikj} is the j th person's response in category k to the i th item. $P(X_{ikj} = k | \theta_j)$ is sometimes written as $P_{ikj}(\theta_j)$ for short. θ_j is the j th person's latent trait or ability score. The item parameter is b_{ih} ; this is the threshold parameter for the i th item in category k (where $h = 0, \dots, k$), hence it is sometimes simply denoted b_{ik} . Function b_{ih} is a point on the θ scale, where the CCC for the response in category k , $P_{ik}(\theta)$, intersects with the CCC for the response in category $k - 1$, $P_{i(k-1)}(\theta)$. Therefore, it is sometimes referred to as a category-intersection parameter to provide an easy way of remembering what the parameter represents. For the category $k = 0$, $b_{i(k=0)} = 0$ and $\theta_j - b_{i(k=0)}$ is defined as 0. As the earlier formula shows, the number of item parameters is m_i when an item has $m_i + 1$ number of categories. For instance, a four-option item has three threshold parameters: $b_{i(k=1)}$, $b_{i(k=2)}$, and $b_{i(k=3)}$.

Equation 3.1_1 is under the “threshold” parameterization. Another parameterization, called the “location plus deviation” parameterization, is sometimes used. The threshold parameter, or category-intersection parameter, b_{ik} , is sometimes decomposed into two components: an overall location parameter for an item plus the deviation parameter for a specific category, as presented in Equation 3.1_2.

$$b_{ik} = b_i + d_{ik}, \quad (3.1_2)$$

where b_i is the location parameter for item i , which is a measure of the overall difficulty of the item, and d_{ik} is the deviation of category k from the overall location of b_{ik} in reference to b_i (i.e., $d_{ik} = b_{ik} - b_i$). Under the location plus deviation parameterization, $\sum_k b_{ik} = 0$ is used typically as part of the model identification. An alternative model constraint for the PCM is to set the first threshold of the first item to 0, i.e., $b_{i(k=1)} = 0$. When the number of categories is two, then the PCM is reduced to the dichotomous simple Rasch model.

We want to caution readers that the terms to describe b_{ik} and d_{ik} vary. Some researchers call b_{ik} a “step” parameter and d_{ik} a “threshold” parameter (see Wright & Masters, 1982). We call d_{ik} the deviation parameter to convey clearly that d_{ik} is the difference between b_{ik} and b_i .

PCM calibration with the “eRm” package

```
install.packages("eRm") #3.1_1
library(eRm) #3.1_2
setwd("c:/mystudy/RIRT/") #3.1_3
ddd<-read.fwf("c3_pcm.dat", width=c(rep(1,5))) #3.1_4
names(ddd)<-paste("I", 1:5, sep="") #3.1_5
dim(ddd) #3.1_6
head(ddd) #3.1_7
summary(ddd) #3.1_8
apply(ddd, 2, table) #3.1_9
```

#3.1_1. The “eRm” package is installed onto the computer using “`install.packages()`.” This only needs to be done once on a computer system. See #2.1_1.

#3.1_2. The “eRm” package must be loaded into the current R session using the “`library()`” function. See #2.1_2.

#3.1_3. The working directory where data and R output files are stored is established using “`setwd()`.” For the PCM, the “`c3_pcm.dat`” file should be saved into the folder established as the working directory. See #2.1_3.

#3.1_4. Read the item response data file “`c3_pcm.dat`” into the R session using “`read.fwf()`” since the item responses are arranged with a fixed format style having no space between columns. The data are saved it as an R object called “`ddd`.” Each of the five columns represents an item and each row represents a person. See #2.2_1.

#3.1_5. Replace the default variable, or item, names by new names (“`I1`” through “`I5`”). See #2.1_8.

#3.1_6. The dimensions of “`ddd`” are providing using the “`dim()`” function. The number of rows and the number of columns are shown; in this application, 900 test-takers and 5 items. See #2.1_5.

#3.1_7. The first six rows of the data are displayed using “`head()`.” These data are polytomous with four category response options: 0, 1, 2, or 3.

#3.1_8. The “`summary()`” command produces descriptive statistics (the first quartile, second quartile [median], third quartile, minimum, maximum, and mean) for each item’s responses. The value of the mean for each item is the item easiness, or the item difficulty in classical test theory. For example, in a Likert style psychological test with response options, 0 = *Strongly Disagree*, 1 = *Disagree*, 2 = *Agree*, and 3 = *Strongly Agree*, the higher the item mean, or item easiness, the easier it was for test-takers to respond in a higher category to endorse, or agree with, the item statement.

```
> summary(ddd)
   I1      I2      I3      I4      I5
Min.   :0.000 Min.   :0.000 Min.   :0.000 Min.   :0.000 Min.   :0.000
1st Qu.:0.000 1st Qu.:1.000 1st Qu.:1.000 1st Qu.:0.000 1st Qu.:0.000
Median :1.000 Median :1.000 Median :2.000 Median :1.000 Median :2.000
Mean    :1.024 Mean    :1.484 Mean    :1.943 Mean    :1.397 Mean    :1.529
3rd Qu.:2.000 3rd Qu.:2.000 3rd Qu.:3.000 3rd Qu.:2.000 3rd Qu.:3.000
Max.   :3.000 Max.   :3.000 Max.   :3.000 Max.   :3.000 Max.   :3.000
```

Unfortunately, the “`summary()`” function in R does not include the standard deviation of the item responses. If the means and the standard deviations of items are desired, the “`apply()`” function can be useful for applying a specific function across rows or columns of a data set. The third argument is the operation, “`mean`” or “`sd`,” for the mean or standard deviation of the data, respectively. The second argument indicates if the operation is applied across the rows, “`1`,” or across the columns “`2`.”

```
apply(ddd, 2, mean) # For the mean calculation across columns
apply(ddd, 2, sd) # For the standard deviation calculation across columns
```

#3.1_9. To check that all categories were utilized for each item, i.e., checking items for any unused category (null category), the “`apply()`” command can be used. The first argument is the data set, the second argument is “`2`” to request the operation across columns, and the third argument applies the “`table`” function to each column of the data. If a table for a single item is desired, “`table(ddd$I1)`” can be used, here to request the table by the item name, `I1`.

```
> apply(ddd, 2, table)
   V1   V2   V3   V4   V5
0  353  181   92  242  236
1  274  281  171  224  209
2  171  259  333  269  198
3  102  179  304  165  257
```

If any item’s output does not display a frequency for all four categories, for this four-category example, the item and category may be problematic in the IRT analysis. All five items have no zero-frequency category in this application.

Before estimating an IRT model in R, we recommend that users rescore, or downcode, the data if null categories are found by removing the null category from the sequence of categories. Take, for example, an item that theoretically has five response categories: 0, 1, 2, 3, and 4. If categories 0 and 2 are never used, i.e., they are null categories, then the non-null categories of 1, 3, and 4 are rescored as 0, 1, and 2. Although this is a straightforward treatment of the null categories, this may not always be the most satisfactory solution. See Wilson and Masters (1993) for their proposal on handling null categories using a modified item parameterization for the PCM. Currently, no universally accepted approach for all polytomous response IRT models exists for the null categories. It is recommended that readers pay full attention in the early developments of a test to the item development, the optimal number of response options, or categories, and category descriptions for an item to mitigate the chance to observe this “null” category issue. When the null category is observed, readers may also be prompted to more closely evaluate the item and its categories for a better understanding of why there were no responses within a category.

Item parameter estimation

```
mod.pcm<-PCM(ddd, sum0 = F) #3.1_10
mod.pcm$conv #3.1_11
thresholds(mod.pcm)$threshtable$'1' #3.1_12
cbind(thresholds(mod.pcm)$threshpar, thresholds(mod.
pcm)$se.thresh) #3.1_13
```

#3.1_10. The PCM is fit to the “ddd” data, and item parameters are estimated using the “PCM()” function. The first threshold of the first item ($b_{1(k=1)}$) is constrained to be 0 for the model identification constraint by specifying the option “sum0 = F.” The estimation result is saved as an object called “mod.pcm.”

The “eRm” uses the conditional maximum likelihood (CML) estimation. A popular model constraint of the Rasch family model when the CML estimation method is to impose a constraint on the item parameters. As in the simple Rasch model, either the average of item parameters or the first item parameter of the first item is set to equal 0. For a straightforward understanding of the model identification imposed to the item parameters and the output, the authors prefer to use the constraint of the first threshold equal to 0 in the PCM application when using the “eRm” package. This allows the rest of the item parameters and person ability parameters to be estimated in reference to $b_{1(k=1)}$.

#3.1_11. The PCM convergence check result is displayed through “mod.pcm\$conv.” An output value of “1” indicates successful convergence, and no specific issue was detected under the default convergence criteria.

#3.1_12. The “thresholds()” command produces the parameter estimates of b_{ik} , which are the threshold parameter estimates and the overall item location, or difficulty. Specifying “\$threshtable\$'1'” provides the output as a matrix, rather than a list or other object type.

> thresholds(mod.pcm)\$threshtable\$'1'				
	Location	Threshold 1	Threshold 2	Threshold 3
I1	0.757580792	0.0000000	0.84016632	1.4325761
I2	0.062194359	-1.0091013	0.16114421	1.0345402
I3	-0.643925513	-1.5291272	-0.89319037	0.4905410
I4	0.241977890	-0.4118716	-0.05097397	1.1887793
I5	0.009683789	-0.4169371	0.10777690	0.3382115

The first numeric column is the overall “Location,” or overall item difficulty; the remaining “Threshold” columns are the threshold estimates (e.g., $\hat{b}_{1(k=1)}$, $\hat{b}_{1(k=2)}$, and $\hat{b}_{1(k=3)}$ for item 1). The first threshold of the first item is fixed as 0 for the model identification. The overall item difficulty of an item is defined as the average of all threshold parameters. The first item overall difficulty, or “Location,” is 0.7576, which is the average of the three threshold estimates (0.0000, 0.8402, and 1.4326).

If the first threshold of the first item happens to be the highest, then, because it is set to be 0, the other item thresholds will be estimated with a negative value. From the model-identification point of view, this is not a problem because the origin of the θ scale is arbitrary. However, some may find the negative item parameter estimates in all (or a majority of) items but the first item peculiar. In this case, the parameters could be rescaled by subtracting a constant from all item and person parameter estimates, which is the average of the overall location parameter estimates of all items. This sets the mean of the overall difficulty of items, or, simply speaking, the overall difficulty of the test, to 0.

For example, suppose that there is a three-item test and each item has four categories (0, 1, 2, and 3). Also, suppose that item threshold parameter estimates using the constraint of $b_{1(k=1)} = 0$ are 0, 0.4, 0.8 for the first item, -3.3, -2.3, -1.3 for the second item, and -2.2, -1.1, -0.9 for the third item. Then the overall item location, or difficulty parameter estimates, which are equal to the average of all threshold parameters, are 0.4, -2.3, and -1.4 for the first, second, and third items, respectively. To conduct the rescaling, the grand mean of the three overall item difficulty parameter estimates, -1.1, is subtracted from \hat{b}_{ik} . (Also, to place person ability on the same scale, $\hat{\theta}$, -1.1 is subtracted from all estimates.) These steps are illustrated here. This simple rescaling that sets the average of the overall item difficulties to be 0 may appeal more in some applications.

Typing “thresholds(mod.pcm)” or “thresholds(mod.pcm)\$threshtable” will also provide the same output as #3.1_12, which contains the item location and deviation parameters (b_i and d_{ik} , respectively). However, the current command in #3.1_12 produces the results as a matrix, which allows further operations when necessary, while these shorter commands do not. For example, suppose the current output is saved as an R object labeled as “rrr” as shown.

```
rrr<-thresholds(mod.pcm)$threshtable$'1'
mean(rrr[,1])
rrr-mean(rrr[,1])
```

If the average difficulty of a test is sought, the location parameter values are extracted (“rrr[,1]”) and the mean of them is calculated (“mean(rrr[,1])”). In this example, the mean is 0.0855. The aforementioned rescaling process that sets the overall difficulty of the test to 0 can be conducted with ease by implementing the third line as well, which subtracts the mean of all overall location parameters from the estimated item location and threshold parameters. The rescaled item parameter estimates are shown here.

> rrr-mean(rrr[,1])				
	Location	Threshold 1	Threshold 2	Threshold 3
I1	0.67207853	-0.08550226	0.75466405	1.3470738
I2	-0.02330790	-1.09460355	0.07564195	0.9490379
I3	-0.72942778	-1.61462948	-0.97869263	0.4050388
I4	0.15647563	-0.49737391	-0.13647623	1.1032770
I5	-0.07581847	-0.50243933	0.02227463	0.2527093

#3.1_13. The output for this command shows both item threshold parameter estimates (in the first numeric column) and their standard errors (in the second numeric column).

```
> cbind(thresholds(mod.pcm)$threshpar, thresholds(mod.pcm)$se.thresh)
      [,1]      [,2]
thresh beta I1.c1 0.00000000 0.0000000
thresh beta I1.c2 0.84016632 0.1576391
thresh beta I1.c3 1.43257606 0.1671750
thresh beta I2.c1 -1.00910129 0.1351391
thresh beta I2.c2 0.16114421 0.1264066
thresh beta I2.c3 1.03454016 0.1412999
...
thresh beta I5.c1 -0.41693707 0.1335477
thresh beta I5.c2 0.10777690 0.1356705
thresh beta I5.c3 0.33821154 0.1374057
```

Typing “summary(mod.pcm)” also produces the estimation results; however, they are for the original model parameterization used in the estimation in the “eRm” package and do not follow the usual PCM parameterization.

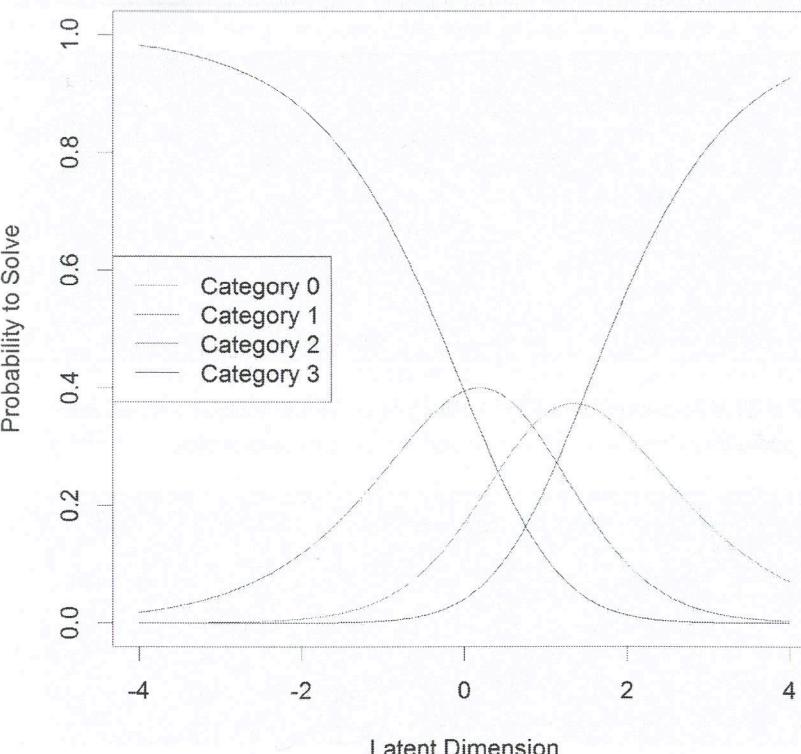
Category characteristic curve (CCC) plot

In the dichotomous response modeling, the IRF was displayed graphically with an ICC. In the polytomous response modeling, the response function for a category is plotted using a category characteristic curve (CCC). Sometimes a CCC is also called a “trace line,” although no universally accepted nomenclature exists.

```
plotICC(mod.pcm, 1) #3.1_14
```

#3.1_14. The CCC of a specified item can be requested using “plotICC()” with the desired item in the second argument. “plotICC(mod.pcm)” will provide a graphical user interface for all item CCCs, where users must press the return key to advance through the displays. A subset of items can be also selected by the command “plotICC(mod.pcm, 1:3),” for example, which provides the CCCs for items 1, 2, and 3. Unlike the simple Rasch case in the “eRm” package, there is no provision of empirical category characteristic curves compared with the model-based CCCs in the PCM application (shown in #2.1_18).

ICC Plot for Item I1



Person latent trait, or ability score estimation

```
p.pcm<-person.parameter(mod.pcm) #3.1_15
p.pcm #3.1_16
coef(p.pcm) #3.1_17
round(cbind(p.pcm$thetapar$NAgroup1, p.pcm$se.
theta$NAgroup1), 3) #3.1_18
plot(p.pcm) #3.1_19
```

#3.1_15 and #3.1_16. The “person.parameter()” command provides the maximum likelihood (ML) estimates for person latent trait or ability scores. There were five items and each item response ranged from 0 and 3. The minimum observed sum score is 0 and the maximum observed sum score is 15 (5×3). For the perfect score and all zero score, no finite ML estimate is possible. An interpolation method is used to provide finite ability estimates for those zero and perfect scores, and “NA” is reported for the “Std. Error.” In the Rasch family model application, the observed