Title

irt pcm — Partial credit model

Description

`irt pcm` fits partial credit models (PCMs) to ordinal items. In the PCM, items vary in their difficulty but share the same discrimination parameter.

`irt gpcm` fits generalized partial credit models (GPCMs) to ordinal items. In the GPCM, items vary in their difficulty and discrimination.

Quick start

PCM for ordinal items `o1` to `o5`

```
irt pcm o1-o5
```

Plot CCCs for `o1`

```
irtgraph icc o1
```

Menu

Statistics > IRT (item response theory)

Syntax

Partial credit model

irt pcm *varlist* [*if*] [*in*] [*weight*] [, *options*]

Generalized partial credit model

irt gpcm *varlist* [*if*] [*in*] [*weight*] [, *options*]

options	Description
Model	
listwise	drop observations with any missing items
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, cluster *clustvar*, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
notable	suppress coefficient table
noheader	suppress output header
display_options	control columns and column formats
Integration	
intmethod(*intmethod*)	integration method
intpoints(#)	set the number of integration points; default is intpoints(7)
Maximization	
maximize_options	control the maximization process; seldom used
startvalues(*svmethod*)	method for obtaining starting values
noestimate	do not fit the model; show starting values instead
dnumerical	use numerical derivative techniques
coeflegend	display legend instead of statistics

intmethod	Description
mvaghermite	mean–variance adaptive Gauss–Hermite quadrature; the default
mcaghermite	mode-curvature adaptive Gauss–Hermite quadrature
ghermite	nonadaptive Gauss–Hermite quadrature

bootstrap, by, jackknife, statsby, and svy are allowed; see **[U] 11.1.10 Prefix commands**.

Weights are not allowed with the bootstrap prefix; see **[R] bootstrap**.

vce() and weights are not allowed with the svy prefix; see **[SVY] svy**.

fweights, iweights, and pweights are allowed; see **[U] 11.1.6 weight**.

startvalues(), noestimate, dnumerical, and coeflegend do not appear in the dialog box.

See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

listwise handles missing values through listwise deletion, which means that the entire observation is omitted from the estimation sample if any of the items are missing for that observation. By default, all nonmissing items in an observation are included in the likelihood calculation; only missing items are excluded.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim), that are robust to some kinds of misspecification (robust), that allow for intragroup correlation (cluster *clustvar*), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [R] ***vce_option***.

Reporting

level(#); see [R] **estimation options**.

notable suppresses the estimation table, either at estimation or upon replay.

noheader suppresses the output header, either at estimation or upon replay.

display_options: noci, nopvalues, cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **estimation options**.

Integration

intmethod(*intmethod*) specifies the integration method to be used for computing the log likelihood. mvaghermite performs mean and variance adaptive Gauss–Hermite quadrature; mcaghermite performs mode and curvature adaptive Gauss–Hermite quadrature; and ghermite performs nonadaptive Gauss–Hermite quadrature.

The default integration method is mvaghermite.

intpoints(#) sets the number of integration points for quadrature. The default is intpoints(7), which means that seven quadrature points are used to compute the log likelihood.

The more integration points, the more accurate the approximation to the log likelihood. However, computation time increases with the number of integration points.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **maximize**. Those that require special mention for irt are listed below.

from() accepts a properly labeled vector of initial values or a list of coefficient names with values. A list of values is not allowed.

The following options are available with irt but are not shown in the dialog box:

startvalues() specifies how starting values are to be computed. Starting values specified in from() override the computed starting values.

startvalues(zero) specifies that all starting values be set to 0. This option is typically useful only when specified with the from() option.

`startvalues(constantonly)` builds on `startvalues(zero)` by fitting a constant-only model for each response to obtain estimates of intercept and cutpoint parameters.

`startvalues(fixedonly)` builds on `startvalues(constantonly)` by fitting a full fixed-effects model for each response variable to obtain estimates of coefficients along with intercept and cutpoint parameters. You can also add suboption `iterate(#)` to limit the number of iterations `irt` allows for fitting the fixed-effects model.

`startvalues(ivloadings)` builds on `startvalues(fixedonly)` by using instrumental-variable methods with the generalized residuals from the fixed-effects models to compute starting values for latent-variable loadings. This is the default behavior.

`noestimate` specifies that the model is not to be fit. Instead, starting values are to be shown (as modified by the above options if modifications were made), and they are to be shown using the `coeflegend` style of output. An important use of this option is before you have modified starting values at all; you can type the following:

```
. irt ..., ... noestimate
. matrix b = e(b)
. ... (modify elements of b) ...
. irt ..., ... from(b)
```

`dnumerical` specifies that during optimization, the gradient vector and Hessian matrix be computed using numerical techniques instead of analytical formulas. By default, `irt` uses analytical formulas for computing the gradient and Hessian for all integration methods.

`coeflegend`; see [R] **estimation options**.

Remarks and examples

stata.com

The following discussion is about how to use `irt` to fit PCMs and GPCMs to ordinal items. If you are new to the IRT features in Stata, we encourage you to read [IRT] **irt** first.

The PCM is used for ordered categorical responses. An item scored $0, 1, \dots, K$ is divided into K adjacent logits, and a positive response in category k implies a positive response to the categories preceding category k.

The probability of person j scoring in category k on item i is

$$\Pr(Y_{ij} = k|\theta_j) = \frac{\exp\{\sum_{t=1}^{k} a(\theta_j - b_{it})\}}{1 + \sum_{s=1}^{K} \exp\{\sum_{t=1}^{s} a(\theta_j - b_{it})\}} \qquad \theta_j \sim N(0,1)$$

where a represents the discrimination common to all items, b_{it} represents the difficulty that distinguishes outcome t from the other outcomes in item i, and θ_j is the latent trait of person j.

In a GPCM, each item has its own discrimination parameter.

The PCM was proposed by Masters (1982). The GPCM was proposed by Muraki (1992).

▷ Example 1: Fitting a PCM

To illustrate the PCM, we use the analogical reasoning data from de Ayala (2009). `alike.dta` contains eight questions, `v1` through `v8`, that ask how two things are alike, for example, “In what way are a dog and a lion alike?” Each response is graded as 0 (incorrect), 1 (partially correct), and 2 (correct). Here we list the first five observations.

```
. use http://www.stata-press.com/data/r14/alike
(Analogical reasoning data from de Ayala (2009))
. list in 1/5, nolabel
```

	v1	v2	v3	v4	v5	v6	v7	v8
1.	2	2	0	0	0	0	0	0
2.	2	0	2	1	2	1	0	0
3.	2	2	1	2	2	1	0	0
4.	2	2	2	1	2	0	0	0
5.	2	2	2	2	1	2	2	2

Looking across the first row, we see that the first respondent correctly solved items v1 and v2 and was incorrect on the remaining items.

We fit a PCM as follows:

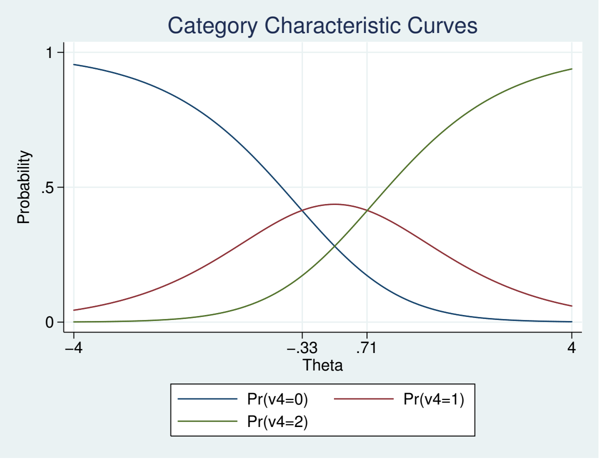
```
. irt pcm v1-v8
Fitting fixed-effects model:
Iteration 0:   log likelihood = -20869.947
Iteration 1:   log likelihood = -20869.947
Fitting full model:
Iteration 0:   log likelihood = -20048.975
Iteration 1:   log likelihood = -19814.317
Iteration 2:   log likelihood = -19678.395
Iteration 3:   log likelihood = -19678.271
Iteration 4:   log likelihood = -19678.271
Partial credit model                            Number of obs     =      2,941
Log likelihood = -19678.271
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Discrim	.8375472	.0194059	43.16	0.000	.7995124	.875582
v1						
Diff						
1 vs 0	-1.546962	.1128848	-13.70	0.000	-1.768212	-1.325712
2 vs 1	-2.463391	.0900475	-27.36	0.000	-2.639881	-2.286902
v2						
Diff						
1 vs 0	-.508318	.0803871	-6.32	0.000	-.6658738	-.3507622
2 vs 1	-1.592003	.0753361	-21.13	0.000	-1.739659	-1.444347
v3						
Diff						
1 vs 0	-1.242774	.0719814	-17.27	0.000	-1.383855	-1.101694
2 vs 1	-.2770088	.0562749	-4.92	0.000	-.3873056	-.166712
v4						
Diff						
1 vs 0	-.3337874	.0580143	-5.75	0.000	-.4474934	-.2200814
2 vs 1	.7146057	.0614175	11.64	0.000	.5942296	.8349819
v5						
Diff						
1 vs 0	1.89372	.0969163	19.54	0.000	1.703768	2.083672
2 vs 1	-1.454011	.0955847	-15.21	0.000	-1.641353	-1.266668
v6						
Diff						
1 vs 0	-.2165156	.052177	-4.15	0.000	-.3187806	-.1142506
2 vs 1	3.115386	.1146119	27.18	0.000	2.89075	3.340021
v7						
Diff						
1 vs 0	1.909344	.0834947	22.87	0.000	1.745698	2.072991
2 vs 1	-.0129814	.0832004	-0.16	0.876	-.1760511	.1500883
v8						
Diff						
1 vs 0	1.514291	.0685158	22.10	0.000	1.380003	1.64858
2 vs 1	1.63067	.0933511	17.47	0.000	1.447705	1.813635

The difficulties represent a point at which the two adjacent categories are equally likely. For item v4, a person with $\theta = -0.33$ is equally likely to answer incorrectly or to answer partially correct (labeled 1 vs 0). A person with $\theta = 0.71$ is equally likely to be partially correct or to be correct (labeled 2 vs 1).

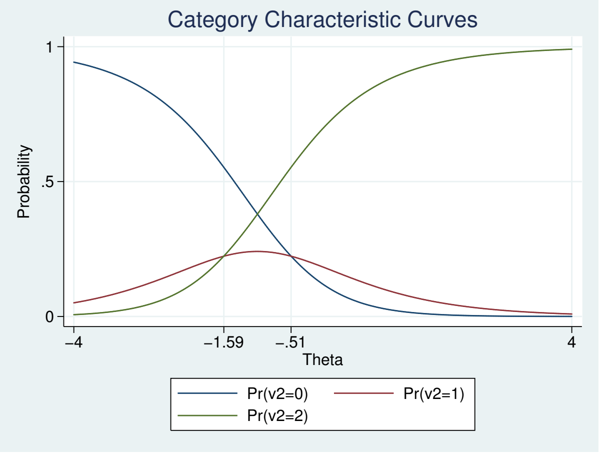
We can present this graphically using CCCs. The curves trace the probability of choosing each category as a function of θ using the estimated PCM parameters. Here we plot the probabilities for item v4 using irtgraph icc; see [IRT] **irtgraph icc** for details.

```
. irtgraph icc v4, xlabel(-4 -.33 .71 4, grid)
```



While the PCM is intended for items having ordered categorical responses, the model is parameterized as if the outcomes were nominal. Therefore, the difficulty parameters for a given item are not necessarily in an increasing order. For example, for item v2, the second difficulty parameter is -1.59 and is smaller than the first difficulty parameter, -0.51. This is called a reversal and indicates that the category with the reversed threshold is dominated by the other two categories. Here we show this situation graphically.

```
. irtgraph icc v2, xlabel(-4 -.51 -1.59 4, grid)
```



Notice that the probability of responding with a partially correct answer is never greater than both the probability of responding incorrectly and the probability of responding correctly. A reversal of the thresholds may indicate a potential problem with the item or with how raters graded the responses to the item. In our case, item v2 is primarily behaving like a binary item.

◁

Stored results

irt pcm and irt gpcm store the following in e():

Scalars	
e(N)	number of observations
e(k)	number of parameters
e(k_eq)	number of equations in e(b)
e(k_dv)	number of dependent variables
e(k_rc)	number of covariances
e(k_rs)	number of variances
e(irt_k_eq)	number of IRT model groups
e(k_items1)	number of items in first IRT model group
e(k_out#)	number of categories for the #th item, ordinal
e(ll)	log likelihood
e(N_clust)	number of clusters
e(n_quad)	number of integration points
e(rank)	rank of e(V)
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if target model converged, 0 otherwise
Macros	
e(cmd)	gsem
e(cmd2)	irt
e(cmdline)	command as typed
e(model1)	pcm or gpcm
e(items1)	names of items in first IRT model group
e(n_cuts1)	numlist of cuts in first IRT model group
e(depvar)	names of all item variables
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(clustvar)	name of cluster variable
e(family#)	family for the #th *item*
e(link#)	link for the #th *item*
e(intmethod)	integration method
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. Err.
e(opt)	type of optimization
e(which)	max or min; whether optimizer is to perform maximization or minimization
e(method)	estimation method: ml
e(ml_method)	type of ml method
e(user)	name of likelihood-evaluator program
e(technique)	maximization technique
e(datasignature)	the checksum
e(datasignaturevars)	variables used in calculation of checksum
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict
e(covariates)	list of covariates
e(footnote)	program used to implement the footnote display

Matrices

`e(_N)`	sample size for each item
`e(b)`	coefficient vector, slope-intercept parameterization
`e(b_pclass)`	parameter class
`e(out#)`	categories for the #th item, ordinal
`e(Cns)`	constraints matrix
`e(ilog)`	iteration log (up to 20 iterations)
`e(gradient)`	gradient vector
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance

Functions

`e(sample)`	marks estimation sample

Methods and formulas

Let Y_{ij} represent the (yet to be observed) outcome for item i from person j. Without loss of generality, we will assume all items take on the ordered categories, $k = 0, 1, \dots, K$.

Using the IRT parameterization, we see that the probability of person j with latent trait level θ_j (the latent trait) providing response k for item i is given by

$$\Pr(Y_{ij} = k | a_i, \mathbf{b}_i, \theta_j) = \frac{\exp\{\sum_{t=1}^{k} a_i(\theta_j - b_{it})\}}{1 + \sum_{s=1}^{K} \exp\{\sum_{t=1}^{s} a_i(\theta_j - b_{it})\}}$$

where a_i represents the discrimination for item i, $\mathbf{b}_i = (b_{i1}, \dots, b_{iK})$ represent the difficulties that distinguish the ordered categories of item i, and it is understood that

$$\Pr(Y_{ij} = 0 | a_i, \mathbf{b}_i, \theta_j) = \frac{1}{1 + \sum_{s=1}^{K} \exp\{\sum_{t=1}^{s} a_i(\theta_j - b_{it})\}}$$

`irt pcm` and `irt gpcm` fit the model using the slope-intercept form, so the probability for providing response k is parameterized as

$$\Pr(Y_{ij} = k | \alpha_i, \boldsymbol{\beta}_i, \theta_j) = \frac{\exp(k\alpha_i\theta_j + \beta_{ik})}{1 + \sum_{s=1}^{K} \exp(s\alpha_i\theta_j + \beta_{is})}$$

The transformation between these two parameterizations is

$$a_i = \alpha_i \qquad b_{ik} = -\frac{\beta_{ik} - \beta_{i,k-1}}{\alpha_i}$$

where $b_{i0} = 0$ and $\beta_{i0} = 0$. For `irt pcm`, the item discriminations a_i are constrained to be equal.

Let y_{ij} be the observed response for Y_{ij} and $p_{ij} = \Pr(Y_{ij} = y_{ij} | \alpha_i, \boldsymbol{\beta}_i, \theta_j)$. Conditional on θ_j, the item responses are assumed to be independent, so the conditional density for person j is given by

$$f(\mathbf{y}_j | \boldsymbol{B}, \theta_j) = \prod_{i=1}^{I} p_{ij}$$

where $\mathbf{y}_j = (y_{1j}, \dots, y_{Ij})$, $\boldsymbol{B} = (\alpha_1, \dots, \alpha_I, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_I)$, and I is the number of items.

Missing items are skipped over in the above product by default. When the `listwise` option is specified, persons with any missing items are dropped from the estimation sample.

The likelihood for person j is computed by integrating out the latent variable from the joint density

$$L_j(\boldsymbol{B}) = \int_{-\infty}^{\infty} f(\mathbf{y}_j|\boldsymbol{B}, \theta_j)\, \phi(\theta_j)\, d\theta_j$$

where $\phi(\cdot)$ is the density function for the standard normal distribution. The log likelihood for the estimation sample is simply the sum of the log likelihoods from the N persons in the estimation sample.

$$\log L(\boldsymbol{B}) = \sum_{j=1}^{N} \log L_j(\boldsymbol{B})$$

The integral in the formula for $L_j(\boldsymbol{B})$ is generally not tractable, so we must use numerical methods.

Gauss–Hermite quadrature and adaptive quadrature are documented in *Methods and formulas* of [IRT] **irt hybrid**.

References

de Ayala, R. J. 2009. *The Theory and Practice of Item Response Theory*. New York: Guilford Press.

Masters, G. N. 1982. A Rasch model for partial credit scoring. *Psychometrika* 47: 149–174.

Muraki, E. 1992. A generalized partial credit model: Application of an EM algorithm. *Applied Psychological Measurement* 16: 159–176.

Also see

[IRT] **irt pcm postestimation** — Postestimation tools for irt pcm

[IRT] **irt** — Introduction to IRT models

[IRT] **irt grm** — Graded response model

[IRT] **irt rsm** — Rating scale model

[SEM] **gsem** — Generalized structural equation model estimation command

[U] 20 Estimation and postestimation commands