Title

stata.com

irt hybrid — Hybrid IRT models

Description

`irt hybrid` fits IRT models to combinations of binary, ordinal, and nominal items.

Quick start

1PL model for binary items `b1` to `b5` and 2PL model for binary items `b6` to `b10`

```
irt hybrid (1pl b1-b5) (2pl b6-b10)
```

Plot ICCs for each item

```
irtgraph icc
```

GRM for ordinal items `o1`, `o2`, and `o3`, 3PL model for binary items `b1` and `b2`, and NRM for nominal items `n1` to `n5`

```
irt hybrid (grm o1 o2 o3) (3pl b1 b2) (nrm n1-n5)
```

Plot CCCs for `o1`

```
irtgraph icc o1
```

Menu

Statistics > IRT (item response theory)

Syntax

```
irt hybrid (model varlist1 [, mopts]) (model varlist2 [, mopts]) [...]
    [if] [in] [weight] [, options]
```

model	Description
`1pl`	One-parameter logistic model
`2pl`	Two-parameter logistic model
`3pl`	Three-parameter logistic model
`grm`	Graded response model
`pcm`	Partial credit model
`gpcm`	Generalized partial credit model
`rsm`	Rating scale model
`nrm`	Nominal response model

mopts	Description
`cns(`*spec*`)`	apply specified parameter constraints
`sepguessing`	estimate a separate pseudoguessing parameter for each item; allowed only with a `3pl` model
`gsepguessing`	estimate separate pseudoguessing parameters for each group; allowed only with a group `3pl` model

options	Description
`group(`*varname*`)`	fit model for different groups
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
`estmetric`	show parameter estimates in the estimation metric
`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`, `collect`, `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`, `estmetric`, `dnumerical`, and `coeflegend` do not appear in the dialog box.
See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

mopts

`cns(`*spec*`)` constrains item parameters to a fixed value or constrains two or more parameters to be equal; see [IRT] **irt constraints** for details.

`sepguessing` specifies that a separate pseudoguessing parameter be estimated for each item. This option is allowed only with a `3pl` model; see [IRT] **irt 3pl** for details.

`gsepguessing` specifies that separate pseudoguessing parameters be estimated for each group. This option is allowed only with a group `3pl` model.

Options

`group(`*varname*`)` specifies that the model be fit separately for the different values of *varname*; see [IRT] **irt, group( )** for details.

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)
```

estmetric displays parameter estimates in the slope-intercept metric that is used for estimation.

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 hybrid IRT models. In a hybrid model, one can fit different IRT models to subsets of items and perform a single calibration for the whole instrument. If you are new to the IRT features in Stata, we encourage you to read [IRT] **irt** first. If you are interested in the details of a specific IRT model, we refer you to the following.

Binary response models

`irt 1pl`	One-parameter logistic model
`irt 2pl`	Two-parameter logistic model
`irt 3pl`	Three-parameter logistic model

Categorical response models

`irt grm`	Graded response model
`irt nrm`	Nominal response model
`irt pcm`	Partial credit model
`irt rsm`	Rating scale model

▷ Example 1: Combining an NRM and a PCM within a single instrument

In example 1 of [IRT] **irt nrm**, we applied a NRM to the physical science test data from de Ayala (2009). The last item is in fact an open-ended question scored on a scale of 1 to 4; thus, a PCM may be more appropriate for this item.

We fit an NRM to items q1–q3 and a PCM to item q4 as follows:

```
. use https://www.stata-press.com/data/r17/science
(Physical science data from de Ayala (2009))
. irt hybrid (nrm q1-q3) (pcm q4)
Fitting fixed-effects model:
Iteration 0:   log likelihood = -9256.1514
Iteration 1:   log likelihood = -9256.1514
Fitting full model:
Iteration 0:   log likelihood = -9383.3438  (not concave)
Iteration 1:   log likelihood = -9251.6343  (not concave)
Iteration 2:   log likelihood = -9200.7879
Iteration 3:   log likelihood = -9183.2288
Iteration 4:   log likelihood = -9169.5042
Iteration 5:   log likelihood = -9168.4031
Iteration 6:   log likelihood = -9168.2855
Iteration 7:   log likelihood = -9168.2854
Hybrid IRT model                                Number of obs = 1,799
Log likelihood = -9168.2854
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
nrm						
q1						
Discrim						
2 vs 1	.4646853	.1344318	3.46	0.001	.2012039	.7281667
3 vs 1	-.7170683	.1835435	-3.91	0.000	-1.076807	-.3573297
4 vs 1	-.5973805	.1601828	-3.73	0.000	-.9113331	-.2834279
Diff						
2 vs 1	-.7692399	.3102269	-2.48	0.013	-1.377274	-.1612063
3 vs 1	-1.137519	.2353162	-4.83	0.000	-1.59873	-.6763076
4 vs 1	-.5488615	.1512123	-3.63	0.000	-.845232	-.2524909
q2						
Discrim						
2 vs 1	-.1940623	.2058631	-0.94	0.346	-.5975465	.2094219
3 vs 1	.5554902	.2001945	2.77	0.006	.1631162	.9478641
4 vs 1	-.0926769	.1957145	-0.47	0.636	-.4762702	.2909165
Diff						
2 vs 1	5.986206	6.587602	0.91	0.364	-6.925257	18.89767
3 vs 1	-3.529391	1.256803	-2.81	0.005	-5.99268	-1.066102
4 vs 1	17.80318	37.97769	0.47	0.639	-56.63173	92.23809
q3						
Discrim						
2 vs 1	-.2375187	.2286107	-1.04	0.299	-.6855875	.2105501
3 vs 1	.701352	.2196849	3.19	0.001	.2707775	1.131927
4 vs 1	1.274855	.2460035	5.18	0.000	.7926971	1.757013
Diff						
2 vs 1	1.128287	1.746527	0.65	0.518	-2.294843	4.551418
3 vs 1	-1.824874	.4596278	-3.97	0.000	-2.725728	-.9240203
4 vs 1	-1.131441	.1882566	-6.01	0.000	-1.500417	-.7624645

```
pcm
------------------------------------------------------------------------------
q4
     Discrim |   .3300808   .0526185     6.27   0.000     .2269504    .4332111
        Diff |
     2 vs 1 |  -2.056822    .322035    -6.39   0.000    -2.687999   -1.425644
     3 vs 2 |  -.2976236   .1923535    -1.55   0.122    -.6746294    .0793823
     4 vs 3 |   .8472731   .2048051     4.14   0.000     .4458626    1.248684
------------------------------------------------------------------------------
```

Note how the NRM and PCM are separated, so it is easy to tell which parameters correspond to which model. Because the PCM is nested in the NRM, we could perform a likelihood-ratio test to see whether our model is preferable to a pure NRM model; see example 1 in [IRT] **irt** and [R] **lrtest** for more information.

◁

▷ Example 2: The 3PL model revisited

In example 1 of [IRT] **irt 3pl**, we used the mathematics and science data from De Boeck and Wilson (2004) to fit a 3PL model where the pseudoguessing parameter was constrained to be the same across items `q1`–`q9`. We mentioned that model identification problems can occur when one tries to estimate a separate guessing parameter for each item. In this example, we show how to deal with identification problems by constraining some pseudoguessing parameters to zero and fitting a 3PL model with separate guessing parameters to the remaining items.

We first fit a full 3PL model to items `q1`–`q9`, where each item has its own pseudoguessing parameter. Because the corresponding fixed-effects model is not identified, we limit the number of iterations `irt` spends fitting the fixed-effects model to 5.

```
. use https://www.stata-press.com/data/r17/masc1
(Data from De Boeck & Wilson (2004))

. irt 3pl q1-q9, sepguessing startvalues(iterate(5))

Fitting fixed-effects model:

Iteration 0:   log likelihood = -5322.8824
Iteration 1:   log likelihood = -4291.3914  (not concave)
Iteration 2:   log likelihood = -4270.0005  (not concave)
Iteration 3:   log likelihood = -4269.7927  (not concave)
Iteration 4:   log likelihood = -4269.7825  (not concave)
Iteration 5:   log likelihood = -4269.7825  (not concave)

Fitting full model:

Iteration 0:   log likelihood = -4227.4731  (not concave)
Iteration 1:   log likelihood = -4188.8074  (not concave)
Iteration 2:   log likelihood =  -4134.829  (not concave)
Iteration 3:   log likelihood = -4121.9664  (not concave)
Iteration 4:   log likelihood =  -4120.161  (not concave)
Iteration 5:   log likelihood =   -4119.33
Iteration 6:   log likelihood = -4118.0626
Iteration 7:   log likelihood = -4117.0488
Iteration 8:   log likelihood = -4115.6541
Iteration 9:   log likelihood = -4115.4168
Iteration 10:  log likelihood = -4114.5522
Iteration 11:  log likelihood = -4114.3738
Iteration 12:  log likelihood = -4114.1039
Iteration 13:  log likelihood = -4113.9668
Iteration 14:  log likelihood = -4113.9036
Iteration 15:  log likelihood = -4113.7972  (not concave)
Iteration 16:  log likelihood = -4113.7712
Iteration 17:  log likelihood = -4113.7505
Iteration 18:  log likelihood = -4113.7226
```

```
Iteration 19:  log likelihood = -4113.7178
Iteration 20:  log likelihood = -4113.7089
Iteration 21:  log likelihood = -4113.7033
Iteration 22:  log likelihood = -4113.6975
Iteration 23:  log likelihood = -4113.6964
Iteration 24:  log likelihood = -4113.6948
Iteration 25:  log likelihood = -4113.6939
Iteration 26:  log likelihood = -4113.6936
Iteration 27:  log likelihood = -4113.6934
Iteration 28:  log likelihood = -4113.6934
Iteration 29:  log likelihood = -4113.6933
```

Three-parameter logistic model — Number of obs = 800

Log likelihood = -4113.6933

		Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
q1							
	Discrim	2.412093	.8686457	2.78	0.005	.709579	4.114608
	Diff	-.0919186	.2094281	-0.44	0.661	-.5023902	.318553
	Guess	.2054622	.1067005			-.003667	.4145914
q2							
	Discrim	.6595413	.1142254	5.77	0.000	.4356637	.883419
	Diff	-.14912	.1209748	-1.23	0.218	-.3862263	.0879862
	Guess	6.49e-06	.004595			-.0089996	.0090126
q3							
	Discrim	1.002138	.1672076	5.99	0.000	.6744172	1.329859
	Diff	-1.61108	.218446	-7.38	0.000	-2.039226	-1.182934
	Guess	9.89e-08	.0004223			-.0008275	.0008277
q4							
	Discrim	1.32466	.5435057	2.44	0.015	.2594084	2.389911
	Diff	.811461	.2349897	3.45	0.001	.3508897	1.272032
	Guess	.1921895	.0961314			.0037755	.3806035
q5							
	Discrim	.8519931	.1450072	5.88	0.000	.5677843	1.136202
	Diff	1.653157	.244456	6.76	0.000	1.174032	2.132282
	Guess	1.27e-08	.0001261			-.0002471	.0002471
q6							
	Discrim	2.160352	.8886889	2.43	0.015	.4185542	3.90215
	Diff	.8876168	.1201165	7.39	0.000	.6521929	1.123041
	Guess	.1725436	.0486371			.0772168	.2678705
q7							
	Discrim	.9442741	2.196661	0.43	0.667	-3.361102	5.24965
	Diff	2.599643	1.80189	1.44	0.149	-.9319954	6.131282
	Guess	.1862207	.2136893			-.2326026	.605044
q8							
	Discrim	1.477403	.2581921	5.72	0.000	.9713561	1.983451
	Diff	-1.664011	.1868776	-8.90	0.000	-2.030284	-1.297737
	Guess	1.33e-09	.0000219			-.0000429	.0000429
q9							
	Discrim	.6233966	.1200201	5.19	0.000	.3881615	.8586317
	Diff	-1.536892	.2867222	-5.36	0.000	-2.098858	-.9749272
	Guess	2.22e-08	.0001789			-.0003506	.0003506

We see that the pseudoguessing parameters for items q2, q3, q5, q8, and q9 are very close to zero. This suggests that we could fit a 2PL model to these five items and a full 3PL model with separate guessing parameters to the remaining four items.

```
. irt hybrid (2pl q2 q3 q5 q8 q9) (3pl q1 q4 q6 q7, sepg), startval(iter(5))
Fitting fixed-effects model:
Iteration 0:   log likelihood = -4846.1954
Iteration 1:   log likelihood = -4274.9988  (not concave)
Iteration 2:   log likelihood = -4269.8038  (not concave)
Iteration 3:   log likelihood = -4269.7889  (not concave)
Iteration 4:   log likelihood = -4269.7825  (not concave)
Iteration 5:   log likelihood = -4269.7825  (not concave)
Fitting full model:
Iteration 0:   log likelihood =   -4237.32  (not concave)
Iteration 1:   log likelihood = -4156.6562
Iteration 2:   log likelihood = -4122.4275
Iteration 3:   log likelihood = -4115.0165
Iteration 4:   log likelihood = -4113.7357
Iteration 5:   log likelihood = -4113.7317
Iteration 6:   log likelihood = -4113.7155  (not concave)
Iteration 7:   log likelihood = -4113.7153
Iteration 8:   log likelihood = -4113.7124
Iteration 9:   log likelihood = -4113.7041
Iteration 10:  log likelihood = -4113.6966
Iteration 11:  log likelihood = -4113.6965
Iteration 12:  log likelihood = -4113.6938
Iteration 13:  log likelihood =  -4113.694
Iteration 14:  log likelihood = -4113.6933
Iteration 15:  log likelihood = -4113.6933
Hybrid IRT model                                        Number of obs = 800
Log likelihood = -4113.6933
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
2pl						
q2						
Discrim	.6595289	.1141777	5.78	0.000	.4357448	.8833131
Diff	-.1491663	.1200093	-1.24	0.214	-.3843802	.0860477
q3						
Discrim	1.002143	.1672015	5.99	0.000	.6744346	1.329852
Diff	-1.611069	.2184352	-7.38	0.000	-2.039194	-1.182944
q5						
Discrim	.8519284	.1449869	5.88	0.000	.5677594	1.136097
Diff	1.65315	.2444541	6.76	0.000	1.174029	2.132271
q8						
Discrim	1.477406	.2581479	5.72	0.000	.9714453	1.983366
Diff	-1.664009	.1868519	-8.91	0.000	-2.030232	-1.297786
q9						
Discrim	.6233934	.1200188	5.19	0.000	.3881608	.858626
Diff	-1.536899	.286721	-5.36	0.000	-2.098862	-.9749362

3pl

q1						
Discrim	2.4122	.868651	2.78	0.005	.7096758	4.114725
Diff	-.0918963	.2094073	-0.44	0.661	-.502327	.3185344
Guess	.2054735	.1066903			-.0036357	.4145828
q4						
Discrim	1.324646	.5435556	2.44	0.015	.2592968	2.389995
Diff	.8114595	.2349801	3.45	0.001	.350907	1.272012
Guess	.1921879	.0961354			.0037661	.3806098
q6						
Discrim	2.160387	.8880769	2.43	0.015	.4197886	3.900986
Diff	.8876139	.120097	7.39	0.000	.652228	1.123
Guess	.1725445	.0485985			.0772932	.2677957
q7						
Discrim	.9441699	2.16823	0.44	0.663	-3.305483	5.193823
Diff	2.599752	1.781284	1.46	0.144	-.8915	6.091004
Guess	.1862199	.2109529			-.2272402	.5996801

Looking at the output, we see that the guessing parameters for `q1`, `q4`, `q6`, and `q7` are similar. So we could further simplify this model by constraining the pseudoguessing parameter to be the same for the 3PL items. We could use a series of likelihood-ratio tests to choose among the competing models; see example 1 in [IRT] **irt** and [R] **lrtest** for more information.

◁

Stored results

`irt` stores 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 equations
`e(k_items#)`	number of items in #th IRT equation
`e(sepguess#)`	1 if #th IRT model contains a separate pseudoguessing parameter
`e(k_cat#)`	number of categories for the #th item, ordinal
`e(k_out#)`	number of outcomes for the #th item, nominal
`e(ll)`	log likelihood
`e(N_clust)`	number of clusters
`e(N_groups)`	number of groups
`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(model#)`	name of IRT model for the #th equation
`e(items#)`	names of items in #th IRT equation

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(groupvar)	name of group 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)	parameter vector
e(b_pclass)	parameter class
e(cat#)	categories for the #th item, ordinal
e(out#)	outcomes for the #th item, nominal
e(Cns)	constraints matrix
e(ilog)	iteration log (up to 20 iterations)
e(gradient)	gradient vector
e(V)	covariance matrix of the estimators
e(V_modelbased)	model-based variance
e(groupvalue)	vector of group values in e(groupvar)
e(nobs)	vector with number of observations per group

Functions

e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices

r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

Methods and formulas are presented under the following headings:

The likelihood
Groups
Gauss–Hermite quadrature
Adaptive quadrature

The likelihood

Let y_{ij} be the observed outcome for item i from person j. Define $p_{ij} = \Pr(Y_{ij} = y_{ij}|\boldsymbol{B}_i, \theta_j)$, where Y_{ij} represents the (yet to be observed) outcome, $\boldsymbol{B}_i$ contains parameters for item i, and θ_j is the ability (the latent trait) of person j. The functional form of p_{ij} and the parameters that go into $\boldsymbol{B}_i$ depend on the choice of IRT model for item i.

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 $\mathbf{y}_j = (y_{1j}, \ldots, y_{Ij})$, $\boldsymbol{B} = (\boldsymbol{B}_1, \ldots, \boldsymbol{B}_I)$, 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.

Groups

When the `group()` option is specified, each group has its own model parameters. The collection of model parameters is

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \\ \vdots \\ \boldsymbol{B}_G \end{pmatrix}$$

where G is the number of groups.

The overall log likelihood is

$$\log L(\boldsymbol{B}) = \sum_{g=1}^{G} \log L(\boldsymbol{B}_g)$$

Gauss–Hermite quadrature

The integral of a function multiplied by the kernel of the standard normal distribution can be approximated using Gauss–Hermite quadrature (GHQ). For Q-point GHQ, let the abscissa and weight pairs be denoted by (x_q^*, w_q^*), $q = 1, \ldots, Q$. The GHQ approximation is then

$$\int_{-\infty}^{\infty} f(x) \exp(-x^2)\, dx \approx \sum_{q=1}^{Q} w_q^* f(x_q^*)$$

Using the standard normal distribution yields the approximation

$$\int_{-\infty}^{\infty} f(x)\phi(x)\, dx \approx \sum_{q=1}^{Q} w_q f(x_q)$$

where $x_q = \sqrt{2}x_q^*$ and $w_q = w_q^*/\sqrt{\pi}$. The GHQ approximation to the likelihood for person j is

$$L_j^{\mathrm{GHQ}}(\boldsymbol{B}) = \sum_{q=1}^{Q} w_q f(\mathbf{y}_j|\boldsymbol{B}, x_q)$$

Adaptive quadrature

This section sets the stage for mean–variance adaptive Gauss–Hermite quadrature (MVAGHQ) and mode-curvature adaptive Gauss–Hermite quadrature (MCAGHQ).

If we fix the item variables and the model parameters, we see that the posterior density for θ_j is proportional to

$$\phi(\theta_j) f(\mathbf{y}_j|\boldsymbol{B}, \theta_j)$$

It is reasonable to assume that this posterior density can be approximated by a normal density with mean μ_j and variance τ_j. Instead of using the prior density of θ_j as the weighting distribution in the integral, we can use our approximation for the posterior density,

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

The likelihood is then approximated with

$$L_j^*(\boldsymbol{B}) = \sum_{q=1}^{Q} \omega_q f(\mathbf{y}_j|\boldsymbol{B}, \xi_q)$$

where ξ_q and the ω_q are functions of x_q and w_q and the adaptive parameters μ_j and τ_j.

For MVAGHQ, μ_j is the posterior mean, and τ_j is the posterior variance of θ_j. They are computed iteratively by updating the posterior moments by using the MVAGHQ approximation, starting with a zero mean and unit variance.

For MCAGHQ, μ_j is the posterior mode for θ_j, and τ_j is the curvature at the mode. They are computed by optimizing the joint density with respect to θ_j.

References

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

De Boeck, P., and M. Wilson, ed. 2004. *Explanatory Item Response Models: A Generalized Linear and Nonlinear Approach*. New York: Springer.

Also see

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

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

[IRT] **irt 1pl** — One-parameter logistic model

[IRT] **irt 2pl** — Two-parameter logistic model

[IRT] **irt 3pl** — Three-parameter logistic model

[IRT] **irt constraints** — Specifying constraints

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

[IRT] **irt nrm** — Nominal response model

[IRT] **irt pcm** — Partial credit model

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

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

[SVY] **svy estimation** — Estimation commands for survey data

[U] 20 Estimation and postestimation commands