Estimation of Item Response Models

Lecture #5

ICPSR Item Response Theory Workshop

The Big Picture of Estimation

ESTIMATOR = Maximum Likelihood;

↓

Mplus

↓

… answers …

Any questions?

Lecture Overview

- Estimation of IRT models
  - Marginal Maximum Likelihood
  - Limited Information
  - Markov Chain Monte Carlo

MARGINAL MAXIMUM LIKELIHOOD (FULL INFORMATION)

What all do we have to estimate?

- For example, a 7-item binary test and a 2-PL model, (assuming we fix Theta distribution to mean=0 and variance=1):
  - 7 item discriminations (a_i) and 7 item difficulties (b_i) = 14 parameters
- 7-item, 4-option test with graded response model (same Theta)?
  - 7 item discriminations (a_i) and 21 item difficulties (b_{ik}) = 28 parameters

- **Item effects** are modeled as **FIXED effects** (same over persons)
  - Thus, our inference is about THAT SPECIFIC ITEM

- What about the all the individual person **Thetas**?
  - Although we CAN estimate them, we don't HAVE TO
  - Thetas are modeled as **RANDOM effects** (one effect per person)
  - Thus, our inference is about the distribution of the latent trait in the population of persons (i.e., we care about the **variance** in that population, but not about the particular **individuals** per se)

Estimation: Items, then People

3 full-information item estimation methods:

- "**Full-information**" :: uses individual item responses
- 3 methods differ with respect to how they handle unknown thetas
- First, 2 less-used and older methods:
  - **"Conditional" ML** (treats Theta as known)
    - Uses total score as "Theta" (so can't include people with all 0's or all 1's)
    - Thus, is only possible within Rasch models (where total is sufficient for theta)
    - If Rasch model holds, estimators are consistent and efficient and can be treated like true likelihood values (i.e., can be used in model comparisons)
  - **"Joint" ML** (treats Theta as a fixed effect)
    - Iterates back and forth between persons and items (each as fixed effects) until item parameters don't change much – then calls it done (i.e., converged)
    - Many disadvantages: estimators are biased, inconsistent, with too small SEs and likelihoods that can't be used in model comparisons
    - More persons …more parameters to estimate….so bad gets worse

"Marginal" ML Estimation (with Numeric Integration)

- Gold standard of estimation (and used in Mplus)
- Relies on two assumptions of **independence**:
  - Item responses are independent after controlling for Theta: "local"
    - This means that the joint probability (likelihood) of two item responses is just the probability of each multiplied together
  - Persons are independent (no clustering or nesting)
    - You can add random effects to handle dependence, but then the assumption is "independent after controlling for random effects"
- Doesn't assume it knows the individual thetas, but does assume that it knows the theta *distribution* (usually standard normal)

"Marginal" ML Estimation

- Step 1: Select a set of **starting values** for all of the item parameters in your model
  - For example, a 7-item binary test and a 2-PL model, (assuming we fix Theta distribution to mean=0, variance=1):
    - 7 discriminations (a_i) and 7 difficulties (b_i) = 14 parameters
  - Starting values can be selected in any manner
    - Randomly or "Intelligently"
    - The closer the starting values are to the actual values of the parameters, the faster the estimation algorithm will converge
    - A good place to start might be the logit versions of the CTT item statistics: b_i = 1-p (for "difficulty"), a_i = item total correlation

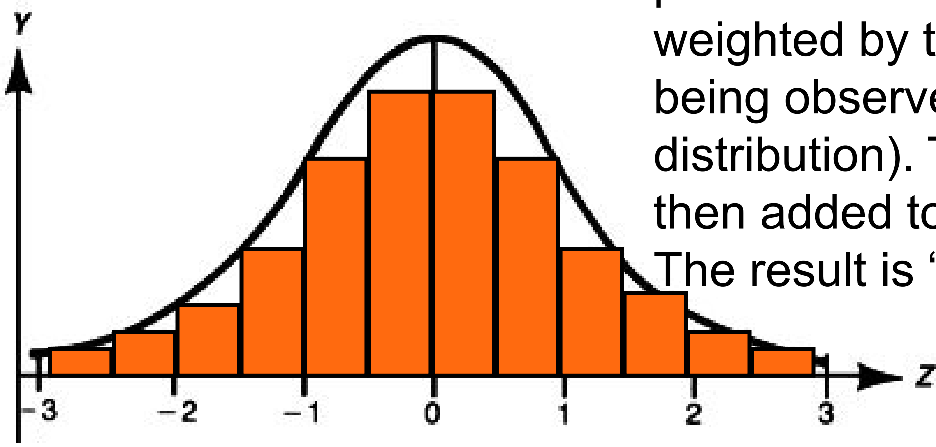
"Marginal" ML Estimation

- Step 2: Compute the **likelihood value** given by the *current* parameter values (using start values or updated values later on)
  - IRT model gives probability of response given item parameters & Theta
  - Likelihood is based on probability of each item response pattern
  - To get that "likelihood function" take each item probability and plug it in:
    - **Likelihood (all answers given parameters) = Product over items of: $p^y(1-p)^{1-y}$**

- But what about Theta? We don't have that yet… no problem!
- Computing the likelihood value for each set of possible parameters involves *removing* the individual Thetas from the equation
  - We don't know the Thetas, but we can guess their distribution (normal)
  - Accomplished by integrating across possible Thetas for each person
    - "Integration" is like summing the area under the curve
  - Integration is accomplished by quadrature – summing up rectangles that approximate the integral for each person

"Marginal" ML Estimation

- Step 2 (still): Divide the distribution into segments
  - → "Gaussian Quadrature" (# rectangles = # "quadrature points")
  - You can either divide the whole distribution into rectangles, or take the most likely section for each person and rectangle that
    - This is "adaptive quadrature" and is computationally more demanding, but gives more accurate results with fewer rectangles



The likelihood of each person's response pattern at each Theta rectangle is then weighted by that rectangle's probability of being observed (as given by the normal distribution). The weighted likelihoods are then added together across all rectangles. The result is "**numeric integration**"

Example of Numeric Integration

1. Start values for item parameters (for simplicity, assume 1.7a=1):
   - Item 1: mean = .73 :: logit = +1, so b = -1
   - Item 2: mean = .27 :: logit = -1, so b = +1
2. Compute likelihood for real data based on item parameters and plausible Thetas (-2,0,2) using IRT model: logit(y=1) = 1.7a(θ_s-b_i)

	Theta = -2	Logit	IF y=1 Prob	IF y=0 1-Prob	Likelihood if both y=1	Theta prob	Theta width	Product per Theta
Item 1 b = -1	(-2 - -1)	-1	0.27	0.73	0.0127548	0.05	2	0.001275
Item 2 b = +1	(-2 - 1)	-3	0.05	0.95				
	Theta = 0	**Logit**	**Prob**	**1-Prob**				
Item 1 b = -1	(0 - -1)	1	0.73	0.27	0.1966119	0.40	2	0.15729
Item 2 b = +1	(0 - 1)	-1	0.27	0.73				
	Theta = +2	**Logit**	**Prob**	**1-Prob**				
Item 1 b = -1	(2 - -1)	3	0.95	0.05	0.6963875	0.05	2	0.069639
Item 2 b = +1	(2 - 1)	1	0.73	0.27				
Overall Likelihood (Sum of Products over All Thetas):								**0.228204**

(then multiply over all people)

(repeat with new values of item parameters until find highest overall likelihood)

“Marginal” ML Estimation

- Step 3: Deciding if the algorithm should stop
  - Algorithm should stop once values of the likelihood ‘**don’t change much**’ from iteration to iteration
    - Change much = EOB (but typically a tiny number like 0.000001)
  - If values of the likelihood don’t change much, then the parameter values used in a given iteration are very close to the ones that will provide the maximum likelihood (the best answers)
  - However, if the change in likelihood value from the previous iteration to the current iteration is larger than your criterion, the algorithm continues… so onto Step 4….

“Marginal” ML Estimation

- Step 4: Choosing new parameter values
  - New values for the model parameters then get picked
  - Many methods are available to make the selection, ranging from algorithmic to downright trial-and-error:
    - Newton-Rhapson: (algorithmic) – uses the shape of the likelihood function to find parameter values closer to the maximum
      - May be harder to implement because shape of the likelihood function must be programmed, but Mplus uses a version of this
      - We’ll see an example of this approach when estimating Thetas
    - Grid search: (naïve) – tries all possible combinations of ranges of parameter values
      - Can take FOREVER
- Step 5: Repeat steps 2-5 until convergence…

Other "Marginal" ML Estimation Techniques

- The method just described is the traditional approach to marginal ML
  - Can be very useful and effective
  - Sometimes called 'Newton' or 'Quasi-Newton' depending on how new parameters are picked and how quadrature is accomplished

- Other methods exist
  - E-M algorithm (Expectation-Maximization Algorithm):
    - Avoids the integral problem by plugging in expected values of Theta at each step (no rectangles)
    - Computes most likely parameter values given a Theta
    - Used in BILOG/MULTILOG
    - Can be hard to do if the expectations are not easy to derive
    - Very slow to converge toward end of algorithm

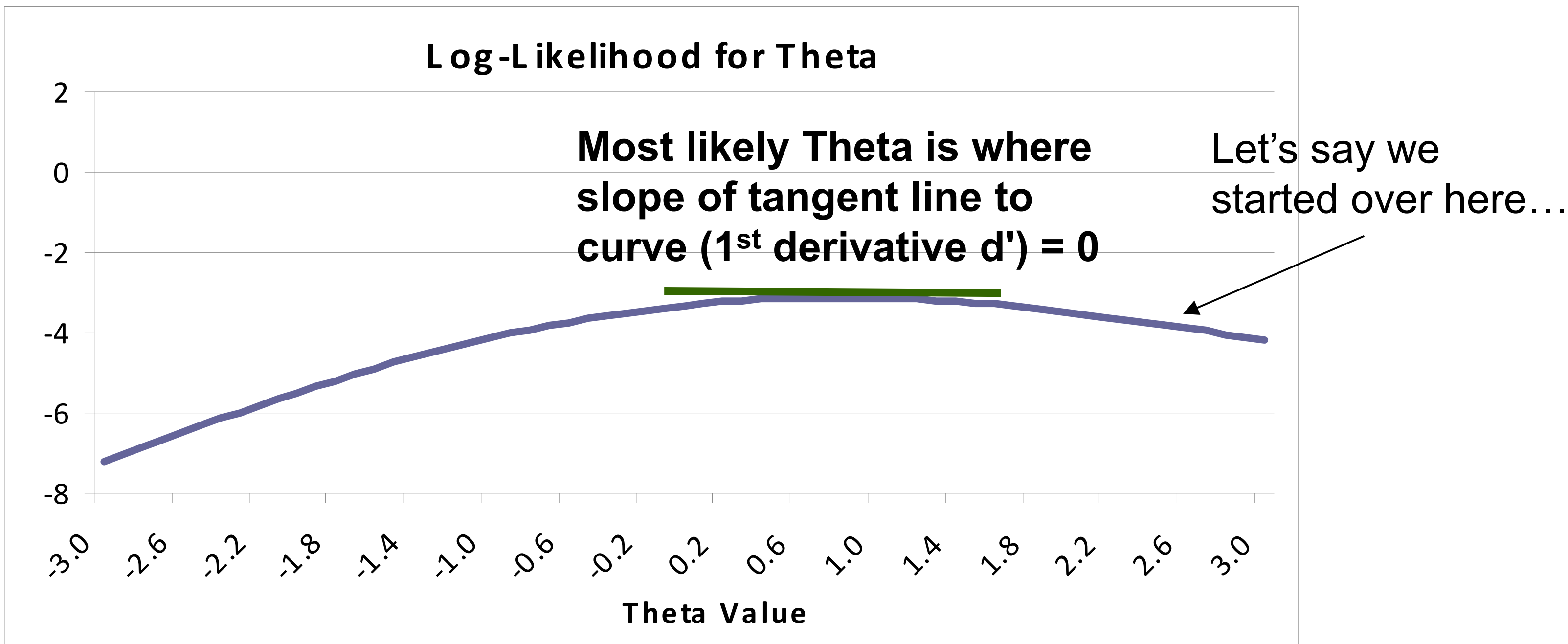
Once we have the items parameters, we need some Thetas…

- Let's say we are searching for Theta given observed responses to 5 items with known difficulty values, so we try out two possible Thetas
  - **Step 1**: Compute prob(Y) using IRT model given each possible Theta
    - b_1 = -2, θ = -1: Logit(Y=1) = (-1 − -2) = 1, so p = .73
    - b_5 = 2, θ = -1: Logit(Y=1) = (-1 − 2) = -3, so p = .05 :: 1-p = .95 (for Y=0)
  - **Step 2**: Multiple item probabilities together :: product = "likelihood"
    - Products get small fast, so can take the LN, then add them instead
  - **Step 3**: See which Theta has the highest likelihood (here, +2)
    - More quadrature points → better estimate of Theta
  - **Step 4**: Because people are independent, we can multiply all their response likelihoods together and solve all at once

Item	b	Y	Term	Value if… θ = -1	θ = +2
1	-2	1	p	0.73	0.98
2	-1	1	p	0.50	0.95
3	0	1	p	0.27	0.88
4	1	1	p	0.12	0.73
5	2	0	1-p	0.95	0.50
Product of values:				0.01	0.30

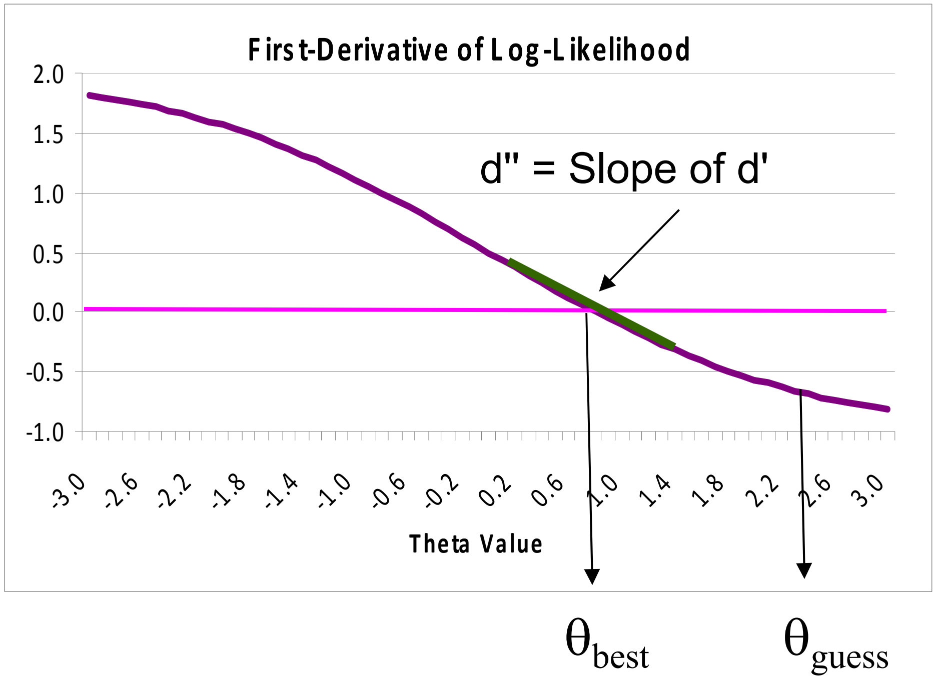
Theta Estimation via Newton Raphson

- We could calculate the likelihood over wide range of Thetas for each person and plot those likelihood values to see where the peak is…
  - But we have lives to lead, so we can solve it mathematically instead by finding where the slope of the likelihood function (the 1st derivative, d') = 0 (its peak)
- Step 1: Start with a guess of Theta, **calculate 1st derivative d'** at that point
  - Are we there (d' = 0) yet? Positive d' = too low, negative d' = too high



Theta Estimation via Newton Raphson

- Step 2: **Calculate the 2nd derivative** (slope of slope, d'') at that point
  - Tells us **how far off we are**, and is used to figure out how much to adjust by
  - d'' will always be negative as approach top, but d' can be positive or negative
- Calculate new guess of Theta: $\theta_{new} = \theta_{old} - (d'/d'')$
  - If (d'/d'') < 0 →Theta increases
    If (d'/d'') > 0 →Theta decreases
    If (d'/d'') = 0 then you are done
- **2nd derivative d'' also tells you how *good* of a peak you have**
  - Need to know where your best Theta is (at d'=0), as well as how precise it is (from d'')
  - If the function is flat, d'' will be smallish
  - **Want large d'' because 1/SQRT(d'') = Theta's SE**



Theta Estimation: ML with Help

- ML is used to come up with most likely Theta given observed response pattern and item parameters…

  …but can't estimate Theta if answers are all 0's or all 1's

- **Prior distributions** to the rescue!
  - Multiply likelihood function for Theta with prior distribution (usually we assume standard normal)
  - Contribution of the prior is minimized with increasing items, but allows us to get Thetas for all 0 or all 1 response patterns

- Note the implication of this for what Theta really is for each person:
  - **THETA IS A DISTRIBUTION, NOT A VALUE**
  - Although we can find the most likely value, we can't ignore its probabilistic nature or how good of an estimate it is (how peaked)
    - SE is constant for CFA factor scores, but SE is NOT constant for IRT Thetas
  - THIS IS WHY YOU SHOULD AVOID OUTPUTTING THETAS

Theta Estimation: 3 Methods

- **ML**: Maximum Likelihood Scoring
  - Uses just item parameters to come up with Thetas
  - Can’t estimate Theta if none or all are answered correctly
- **MAP**: Maximum a Posteriori Scoring
  - Combine ML estimate with a continuous normal prior distribution
  - Theta estimate is mode of combined posterior distribution
  - Theta will be regressed toward mean if reliability is low
  - Iterative procedure, so computationally intensive
- **EAP**: Expected A Posteriori Scoring
  - Combine ML estimate with a ‘rectangled’ normal prior distribution
  - Theta estimate is mean of combined posterior distribution
  - Non-iterative, easier computationally (and what Mplus does)

"IRT" vs. "Rasch": What Goes into Theta

- In Rasch models, **total score is a 'sufficient statistic'** for Theta
  - For example, given 5 items ordered in difficulty from easiest to hardest, each of these response patterns where **3/5 are correct** would yield the **same estimate of Theta**:

    1 1 1 0 0 (most consistent)
    0 1 1 1 0
    0 0 1 1 1
    1 0 1 0 1 (???)
    …. (and so forth)

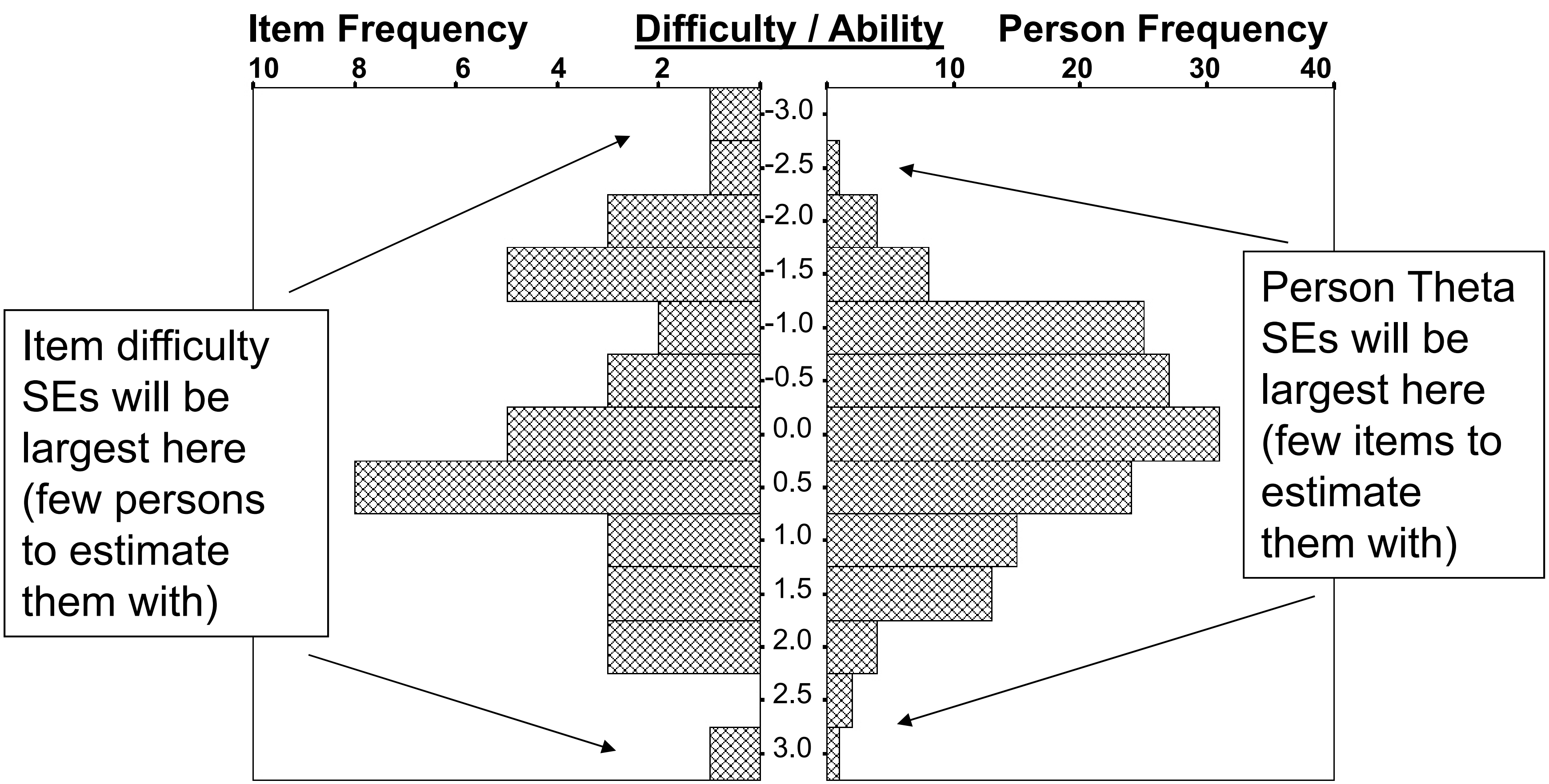
- In 2-PL models, items with higher discrimination (a_i) **count more** towards Theta (and SE will be lower for tests with higher a_i items)
  - It not only matters **how many** items you got correct, but **which ones**
  - Rasch people don't like this idea, because then ordering of persons on Theta is dependent on the item properties

Interpretation of Theta

- **Theta estimates are ‘sample-free’ and ‘scale-free’**
  - Theta estimate does not depend on who took the test with you
  - Theta estimate does not depend on which items were on the test
    - Assuming items all measure same thing, can get location on latent ability metric regardless of which *particular* items were given

- However: although the Theta estimate does not depend on the particular items, its ***standard error*** does
  - Extreme Thetas without many items of comparable difficulty will not be estimated that well :: large SE (flat likelihood)
  - Likewise, items of extreme difficulty without many persons of comparable ability will not be estimated that well :: large SE

Distribution of Item Difficulty & Person Ability



What Can Go Wrong with Model Comparisons under MML…

- **MML is a full-information estimator**
  - **It tries to reproduce the observed item response pattern, not a matrix!**
- Model DF is based on FULL response pattern:
  - DF = # possible observed patterns – # parameters – 1
  - So, for an example of 24 binary items in 1-PL Model:
    - Max DF = 2^24 – $\#a_i$ – $\#b_i$ – 1 =16777216 – 1 – 24 – 1 = **16777190!**
    - If some cells aren't observed (Mplus deletes them from the χ^2 calculation), then DF may be < Max DF, and thus χ^2 won't have the right distribution
- Pearson χ^2 based on classic formula: $(obs - exp)^2 / exp$
  - Not enough people to fill up all possible patterns
  - Other χ^2 given in output is "Likelihood Ratio" χ^2, calculated differently
  - **χ^2 generally won't work well for assessing absolute global fit in IRT**

What we can do in MML: Relative Fit via -2LL Comparisons

- **Nested models** can be compared with -2LL difference tests
  - Step 1: Calculate -2*difference of LL_{old} and LL_{new}
  - Step 2: Calculate difference in df_{old} and df_{new} (given as "# free parms")
  - Compare $-2LL_{diff}$ on $df = df_{diff}$ to χ^2 critical values (or excel CHIDIST)
  - Add 1 parameter? $-2LL_{diff} > 3.84$, add 2: $-2LL_{diff} > 5.99$…

- If **adding** a parameter, model fit gets **better** (LL up, -2LL down)
- If **removing** a parameter, model fit gets **worse** (LL down, -2LL up)
- AIC and BIC values (based off of -2LL) can be used to compare non-nested models (given same sample), smaller is better

- No trustable absolute global fit info available via MML for IRT (but help is on the way)
  - Can't fill up all possible response patterns in most samples

MML in Mplus vs. SAS NLMIXED

- MML with numerical integration can also be conducted using SAS PROC NLMIXED
  - Several recent articles with helpful code (see reference list)
  - De Boeck and Wilson (2004) cover regular IRT models and ‘explanatory’ IRT models in NLMIXED
  - The plus side: Because NLMIXED is a general estimation routine, you can pretty much do any model you want…
    - Get -2LL, AIC, and BIC to do model comparisons
    - See my examples for 1-PL and 2-PL versions of binary model, partial credit model, and graded response model
  - The downside: IT TAKES FOREVER
    - The estimation routines in NLMIXED are more general than Mplus
    - Limited to one level of random effects – can now use PROC GLIMMIX for more levels

Summary: MML for IRT Models

- Marginal ML with numeric integration for IRT models tries to find the parameters most likely *given the observed item response pattern*
  - Can provide IRT parameters on logit (default) or probit scales

- Because of the integration (rectangling Theta) required at each step of estimation, it will not be feasible to use MML for IRT models in small samples or for many factors at once

- MML usually cannot provide absolute fit information
  - Data are not summarized by a matrix – by response patterns instead
  - Usually not enough people to fill up all possible response patterns, so there's no valid basis for an absolute fit comparison
  - Nested models can have relative fit compared via difference in -2LL

- There is another game in town for IRT in Mplus, however…

LIMITED INFORMATION METHODS

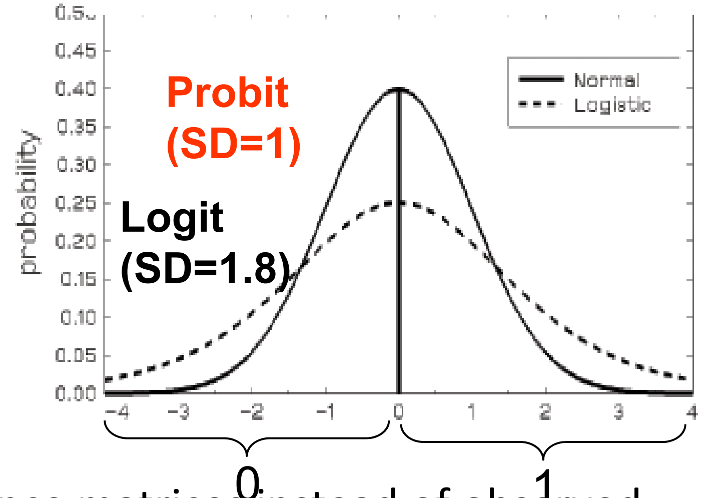
WLSMV Estimation

Actual Data

00	01
10	11

From the observed proportions we try to guess what the correlation would be from a bivariate version of this →

Distribution of Pretend Y*



- WLSMV uses "pretend" versions of covariance matrices instead of observed
  - Correlations among pretend continuous variables (Y*) that really underlie binary ("tetrachoric") or ordinal ("polychoric") responses (pretend Y* has Variance=1)
- The diagonal W "weight" part tries to emphasize reproducing latent variable correlations that are relatively well-determined more than those that aren't
  - The full weight matrix is of order z*z, where z is number of elements to estimate
  - The "diagonal" part means it only uses the *preciseness of the estimates themselves*, not the covariances among the "preciseness-es" (much easier)
- The "MV" corrects the χ^2 test for bias arising from this weighting process

More about WLSMV Estimation

- Works much better and faster than ML when you have small samples or many factors to estimate (no rectangling required)
- Does assume missing data are missing completely at random (whereas ML assumes missing at random)
- Because a covariance matrix (for the Y*) is used as the input data, we get absolute fit indices as in CFA
  - People tend not to be as strict with cut-off values, though

- Model coefficients will be on the **probit scale** instead of logit scale
- Two different model parameterizations are available via the **PARAMETERIZATION IS** option on the **ANALYSIS** command
  - “**Delta**” (default): variance of Y* = 1 = “marginal parameterization”
  - “**Theta**”: error variance = 1 instead = “conditional parameterization”
    - **WE WILL USE THIS ONE TO HELP SIMPLIFY CONVERSIONS**

MARKOV CHAIN MONTE CARLO ESTIMATION

MCMC Estimation

- Markov Chain Monte Carlo is a resampling technique that uses Bayes' theorem to provide estimates of model parameters
  - Both items and persons
- Not typically used for estimation in the conventional models since MML and LI methods typically work well
  - Sometimes used for more complicated models or data problems

MCMC Estimation

- The concept of MCMC estimation is to construct a set of random draws from the posterior distribution for each parameter being estimated

- **<u>The problem</u>**: typically difficult to draw random numbers from an arbitrary distribution without a known form

MCMC Estimation

- To simplify, a distribution from which we can easily draw samples is chosen, and random draws are taken from it

- These draws are either accepted or rejected as being plausible from the actual posterior distribution until we have retained enough draws to make inferences

MCMC Estimation

- The draws that are retained are then taken to be a sample from the posterior distribution

- Using the sample from the posterior, the point estimates can taken to be the mean or the mode of the posterior distribution

MCMC Estimation

- One nice feature:
  - One can incorporate estimates of uncertainty into parameter estimation.
    - E.g., standard errors of ability can be taken into account when estimating item parameters
    - Likewise, standard errors of item parameters can be taken into account when estimating ability parameters

MCMC Conclusion

- Method of simulating random draws from any theoretical multivariate distribution
  - Any posterior distribution

- Features of the theoretical distribution (i.e., mean, variance) are then estimated based on the random sample

- Easy to specify, but takes a **long** time
  - Very inefficient, but it almost always works

CONCLUDING REMARKS

Wrapping Up…

- ML estimation with numeric integration provides:
  - '**Best guess**' at to the value of each item and person parameter
  - **SE** that conveys the uncertainty of that prediction

- The '**best guesses**' for the model parameters do not depend on the sample:
  - Item estimates do not depend on the particular individuals that took the test
  - Person estimates do not depend on the particular items that were administered
  - Thus, model parameter estimates are sample-invariant

- The **SEs** for those model parameters DO depend on the sample
  - Item parameters will be estimated less precisely where there are fewer individuals
  - Person parameters will be estimated less precisely where there are fewer items

- Limited information methods provide more flexibility
  - Only available in Mplus…and only available for some models
  - Should only use if sample size to model complexity ratio is low (only if you have to)

- Markov Chain Monte Carlo methods are also flexible
  - Can be used for all sizes of data and model complexity
  - Not as easy to construct (Mplus has methods; WinBugs; R; Fortran…)