

Hierarchical (a.k.a. Multilevel) Modeling

Joachim Vandekerckhove

Spring 2025

Hierarchical modeling

A statistical framework for data with **dependencies** from **group structure**.

Hierarchical modeling

A statistical framework for data with **dependencies** from **group structure**.

- Examples: Students in classrooms, patients in hospitals, trials within participants, stimuli within conditions.
- Observations within the same group are typically correlated (non-independent).
- Standard methods ignore this structure, which can lead to biased estimates and incorrect inferences.

Pooling is a spectrum

Complete pooling (ignore structure)

Analyze all data together.

- × Underestimates errors.
- × Hides true group differences.
- × Doesn't quantify group variability.
- × Unlikely to generalize.

No pooling (separate analyses)

Analyze each group separately.

- × Ignores within-group similarities.
- × Inefficient.
- × Noisy estimates.
- × Impossible to generalize.

The compromise: Partial pooling

Hierarchical models provide a statistically principled compromise with *partial pooling*.

The compromise: Partial pooling

Hierarchical models provide a statistically principled compromise with *partial pooling*.

- Information is **adaptively shared** across units within populations.

The compromise: Partial pooling

Hierarchical models provide a statistically principled compromise with *partial pooling*.

- Information is **adaptively shared** across units within populations.
- Each unit contributes, weighted by its precision.

The compromise: Partial pooling

Hierarchical models provide a statistically principled compromise with *partial pooling*.

- Information is **adaptively shared** across units within populations.
- Each unit contributes, weighted by its precision.
- **“Borrow strength”**: Units within populations inform each other.

The compromise: Partial pooling

Hierarchical models provide a statistically principled compromise with *partial pooling*.

- Information is **adaptively shared** across units within populations.
- Each unit contributes, weighted by its precision.
- **“Borrow strength”**: Units within populations inform each other.
- Improves individual unit estimates (especially for noisy/small units) with good prior knowledge.

The compromise: Partial pooling

Hierarchical models provide a statistically principled compromise with *partial pooling*.

- Information is **adaptively shared** across units within populations.
- Each unit contributes, weighted by its precision.
- **“Borrow strength”**: Units within populations inform each other.
- Improves individual unit estimates (especially for noisy/small units) with good prior knowledge.
- Simultaneously estimates population-level effects **and** the extent of variation between participants.

Hierarchical model formulation

Let $y_{i|j}$ be outcome for observation i in group j , $x_{i|j}$ a predictor.

Hierarchical model formulation

Let $y_{i|j}$ be outcome for observation i in group j , $x_{i|j}$ a predictor.

- **Level 1 (within-group):**

$$y_{i|j} = \alpha_j + \beta x_{i|j} + \epsilon_{i|j}, \text{ with } \epsilon_{i|j} \sim \mathcal{N}(0, \sigma_y^2)$$

Intercept α_j varies by participant, while slope β is fixed (common).

Hierarchical model formulation

Let $y_{i|j}$ be outcome for observation i in group j , $x_{i|j}$ a predictor.

- **Level 1 (within-group):**

$$y_{i|j} = \alpha_j + \beta x_{i|j} + \epsilon_{i|j}, \text{ with } \epsilon_{i|j} \sim \mathcal{N}(0, \sigma_y^2)$$

Intercept α_j varies by participant, while slope β is fixed (common).

- **Level 2 (between-group):**

$$\alpha_j \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2)$$

The group intercepts (α_j) are drawn from a population distribution.

- μ_α : Population average intercept.
- σ_α^2 : Variance of intercepts across participants.

Hierarchical model formulation

Let $y_{i|j}$ be outcome for observation i in group j , $x_{i|j}$ a predictor.

- **Level 1 (within-group):**

$$y_{i|j} = \alpha_j + \beta x_{i|j} + \epsilon_{i|j}, \text{ with } \epsilon_{i|j} \sim \mathcal{N}(0, \sigma_y^2)$$

Intercept α_j varies by participant, while slope β is fixed (common).

- **Level 2 (between-group):**

$$\alpha_j \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2)$$

The group intercepts (α_j) are drawn from a population distribution.

- μ_α : Population average intercept.
- σ_α^2 : Variance of intercepts across participants.

This structure explicitly models the dependency within populations.

Hierarchical model structure

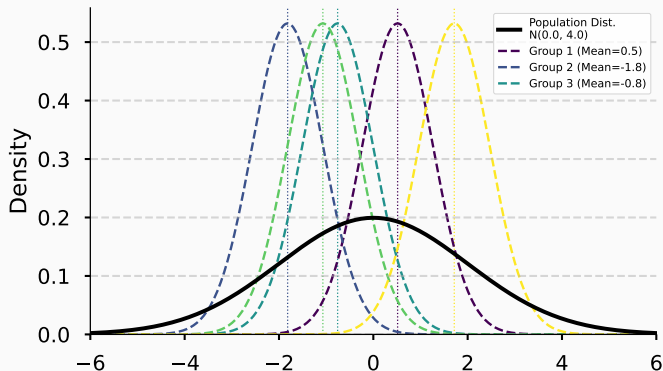


Figure 1: The population distribution (solid black line, representing $N(\mu_\alpha, \sigma_\alpha^2)$) describes the overall tendency for participant means. Person-specific distributions (dashed colored lines, representing $N(\alpha_j, \sigma_y^2)$ for different j) have means (α_j , marked by dotted vertical lines) drawn from the population distribution.

Hierarchical model formulation

Let $y_{i|j}$ be outcome for observation i in group j , $x_{i|j}$ a predictor.

$$y_{i|j} = \alpha_j + \beta x_{i|j} + \epsilon_{i|j}, \text{ with } \epsilon_{i|j} \sim \mathcal{N}(0, \sigma_y^2)$$

$$\alpha_j = \mu_\alpha + u_j, \text{ with } u_j \sim \mathcal{N}(0, \sigma_\alpha^2)$$

All together: $y_{i|j} = \mu_\alpha + \beta x_{i|j} + u_j + \epsilon_{i|j}$

This model has a global intercept (μ_α), a global slope (β), a participant-specific deviation (u_j), and a residual error ($\epsilon_{i|j}$).

- **Fixed effects**

Parameters that differ between participants and are estimated independently from one another – they do not share free parameters: $\alpha_1 \sim \mathcal{N}(0, 1), \alpha_2 \sim \mathcal{N}(0, 1), \dots$

- **Fixed effects**

Parameters that differ between participants and are estimated independently from one another – they do not share free parameters: $\alpha_1 \sim \mathcal{N}(0, 1), \alpha_2 \sim \mathcal{N}(0, 1), \dots$

- **Random effects**

Parameters that differ between participants but are estimated as an interconnected set – they are affected by shared parameters:

$$\alpha_1 \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2), \alpha_2 \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2), \dots$$

- **Fixed effects**

Parameters that differ between participants and are estimated independently from one another – they do not share free parameters: $\alpha_1 \sim \mathcal{N}(0, 1), \alpha_2 \sim \mathcal{N}(0, 1), \dots$

- **Random effects**

Parameters that differ between participants but are estimated as an interconnected set – they are affected by shared parameters:

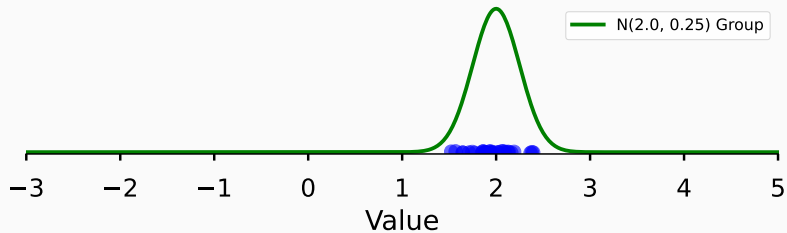
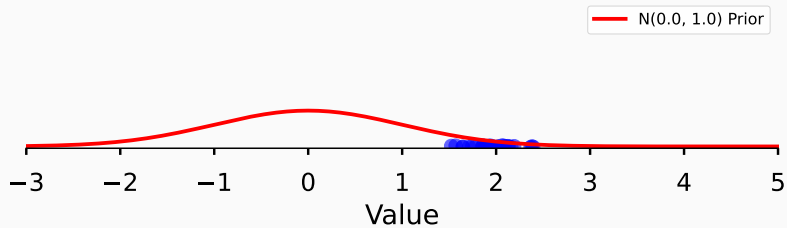
$$\alpha_1 \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2), \alpha_2 \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2), \dots$$

- **Variance components**

Parameters characterizing variability at different levels:

- Level 1: Residual variance σ_y^2 .
- Level 2: Random effect variances (σ_α^2) quantify the magnitude of inter-individual differences.

Fixed vs. random effects



Partial pooling and shrinkage

Estimates for person-specific parameters (e.g., $\hat{\alpha}_j$) balance two sources of information:

Partial pooling and shrinkage

Estimates for person-specific parameters (e.g., $\hat{\alpha}_j$) balance two sources of information:

1. From the person's own data (the **no-pooling estimate**, $\hat{\alpha}_{j,\text{no pool}}$).
2. From the overall population distribution (the **population mean estimate**, $\hat{\mu}_\alpha$).

Partial pooling and shrinkage

Estimates for person-specific parameters (e.g., $\hat{\alpha}_j$) balance two sources of information:

1. From the person's own data (the **no-pooling estimate**, $\hat{\alpha}_{j,\text{no pool}}$).
2. From the overall population distribution (the **population mean estimate**, $\hat{\mu}_\alpha$).

$$\hat{\alpha}_j \approx w_j \hat{\alpha}_{j,\text{no pool}} + (1 - w_j) \hat{\mu}_\alpha$$

Partial pooling and shrinkage

Estimates for person-specific parameters (e.g., $\hat{\alpha}_j$) balance two sources of information:

1. From the person's own data (the **no-pooling estimate**, $\hat{\alpha}_{j,\text{no pool}}$).
2. From the overall population distribution (the **population mean estimate**, $\hat{\mu}_\alpha$).

$$\hat{\alpha}_j \approx w_j \hat{\alpha}_{j,\text{no pool}} + (1 - w_j) \hat{\mu}_\alpha$$

The weight w_j (or shrinkage factor) depends on precision:

Partial pooling and shrinkage

Estimates for person-specific parameters (e.g., $\hat{\alpha}_j$) balance two sources of information:

1. From the person's own data (the **no-pooling estimate**, $\hat{\alpha}_{j,\text{no pool}}$).
2. From the overall population distribution (the **population mean estimate**, $\hat{\mu}_\alpha$).

$$\hat{\alpha}_j \approx w_j \hat{\alpha}_{j,\text{no pool}} + (1 - w_j) \hat{\mu}_\alpha$$

The weight w_j (or shrinkage factor) depends on precision:

$$w_j \approx \frac{\text{individual precision}}{\text{population precision} + \text{individual precision}} = \frac{1/\text{Var}(\hat{\alpha}_{j,\text{no pool}})}{1/\text{Var}(\hat{\alpha}_{j,\text{no pool}}) + 1/\sigma_\alpha^2}$$

Partial pooling and shrinkage

Estimates for person-specific parameters (e.g., $\hat{\alpha}_j$) balance two sources of information:

1. From the person's own data (the **no-pooling estimate**, $\hat{\alpha}_{j,\text{no pool}}$).
2. From the overall population distribution (the **population mean estimate**, $\hat{\mu}_\alpha$).

$$\hat{\alpha}_j \approx w_j \hat{\alpha}_{j,\text{no pool}} + (1 - w_j) \hat{\mu}_\alpha$$

The weight w_j (or shrinkage factor) depends on precision:

$$w_j \approx \frac{\text{individual precision}}{\text{population precision} + \text{individual precision}} = \frac{1/\text{Var}(\hat{\alpha}_{j,\text{no pool}})}{1/\text{Var}(\hat{\alpha}_{j,\text{no pool}}) + 1/\sigma_\alpha^2}$$

$\text{Var}(\hat{\alpha}_{j,\text{no pool}})$ depends on participant sample size n_j and within-participant variance σ_y^2 . Person-specific estimates are **shrunk** towards the population mean.

Adaptive shrinkage

The amount of shrinkage is automatically **adaptive** and data-dependent:

More shrinkage (towards $\hat{\mu}_\alpha$) when:

- Participant has **less data** / noisy estimate (large $\text{Var}(\hat{\alpha}_{j,\text{no pool}})$).
- Participants are very **similar** (small between-participant variance σ_α^2).

Adaptive shrinkage

The amount of shrinkage is automatically **adaptive** and data-dependent:

More shrinkage (towards $\hat{\mu}_\alpha$) when:

- Participant has **less data** / noisy estimate (large $\text{Var}(\hat{\alpha}_{j,\text{no pool}})$).
- Participants are very **similar** (small between-participant variance σ_α^2).

Less shrinkage (estimate closer to participant's own data) when:

- Participant has **more data** / precise estimate (small $\text{Var}(\hat{\alpha}_{j,\text{no pool}})$).
- Participants are very **dissimilar** (large between-participant variance σ_α^2).

Shrinkage visualization

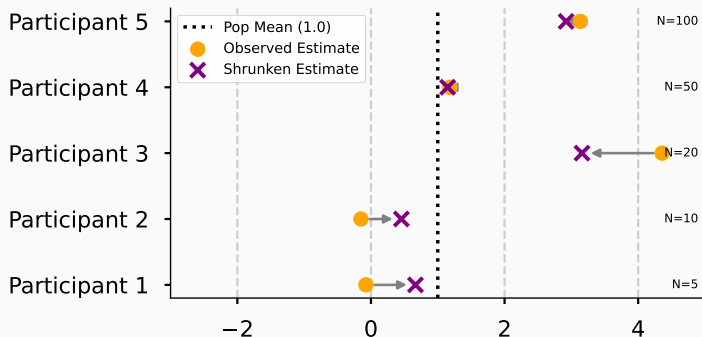


Figure 2: Observed group means (o) are pulled towards the estimated population mean (dashed line). The amount of shrinkage (length of gray arrow) is greater for groups with less data (smaller n_j), resulting in the shrunk estimates (x). The overall population mean estimate itself is informed by all participants.

Bayesian implementation

Bayesian methods provide a natural framework:

Bayesian implementation

Bayesian methods provide a natural framework:

- **Full probability model:** Specify the entire structure:
 - Level 1 likelihood: $P(\text{Data}|\text{Level 1 Params})$
 - Level 2 priors: $P(\text{Level 1 Params}|\text{Level 2 Hyperparams})$
 - Hyperpriors: $P(\text{Level 2 Hyperparams})$

Bayesian implementation

Bayesian methods provide a natural framework:

- **Full probability model:** Specify the entire structure:
 - Level 1 likelihood: $P(\text{Data}|\text{Level 1 Params})$
 - Level 2 priors: $P(\text{Level 1 Params}|\text{Level 2 Hyperparams})$
 - Hyperpriors: $P(\text{Level 2 Hyperparams})$
- **Coherent uncertainty:** Get full posterior distributions for **all** parameters (fixed effects, random effects, variance components), naturally propagating uncertainty.

Bayesian implementation

Bayesian methods provide a natural framework:

- **Full probability model:** Specify the entire structure:
 - Level 1 likelihood: $P(\text{Data}|\text{Level 1 Params})$
 - Level 2 priors: $P(\text{Level 1 Params}|\text{Level 2 Hyperparams})$
 - Hyperpriors: $P(\text{Level 2 Hyperparams})$
- **Coherent uncertainty:** Get full posterior distributions for **all** parameters (fixed effects, random effects, variance components), naturally propagating uncertainty.
- **Computation:** Modern MCMC (e.g., HMC/NUTS in Stan, PyMC) handles complex posteriors effectively.

Bayesian implementation

Bayesian methods provide a natural framework:

- **Full probability model:** Specify the entire structure:
 - Level 1 likelihood: $P(\text{Data}|\text{Level 1 Params})$
 - Level 2 priors: $P(\text{Level 1 Params}|\text{Level 2 Hyperparams})$
 - Hyperpriors: $P(\text{Level 2 Hyperparams})$
- **Coherent uncertainty:** Get full posterior distributions for **all** parameters (fixed effects, random effects, variance components), naturally propagating uncertainty.
- **Computation:** Modern MCMC (e.g., HMC/NUTS in Stan, PyMC) handles complex posteriors effectively.
- **Prior specification:** Requires care!
 - Fixed Effects / Means (μ_s): Often weakly informative (e.g., wide Normal).
 - Variance Components (σ_s): Crucial! Use weakly informative priors concentrated away from zero (e.g., Half-Normal, Half-Cauchy) to avoid issues.

Model checking and interpretation

Fitting is just the start! Rigorous checking is essential:

Model checking and interpretation

Fitting is just the start! Rigorous checking is essential:

- **MCMC convergence:** Check \hat{R} (should be ≈ 1.0), Effective Sample Size (ESS), trace plots.

Model checking and interpretation

Fitting is just the start! Rigorous checking is essential:

- **MCMC convergence:** Check \hat{R} (should be ≈ 1.0), Effective Sample Size (ESS), trace plots.
- **Prior sensitivity analysis:** Do results change much with different reasonable priors?

Model checking and interpretation

Fitting is just the start! Rigorous checking is essential:

- **MCMC convergence:** Check \hat{R} (should be ≈ 1.0), Effective Sample Size (ESS), trace plots.
- **Prior sensitivity analysis:** Do results change much with different reasonable priors?
- **Posterior predictive checks (PPCs):**
 - Simulate data from the fitted model.
 - Compare simulated data distributions to observed data. Mismatches indicate model problems.

Model checking and interpretation

Fitting is just the start! Rigorous checking is essential:

- **MCMC convergence:** Check \hat{R} (should be ≈ 1.0), Effective Sample Size (ESS), trace plots.
- **Prior sensitivity analysis:** Do results change much with different reasonable priors?
- **Posterior predictive checks (PPCs):**
 - Simulate data from the fitted model.
 - Compare simulated data distributions to observed data. Mismatches indicate model problems.

Interpretation: Focus on population parameters (μ s, fixed β s), magnitude of variation (σ s), and potentially shrunken group estimates (α_j s, β_j s), always with uncertainty (posterior distributions / intervals).

Crossed vs. nested random effects

- Nested: Students in classrooms, classrooms in schools.

$$y_{i|c|s} = \alpha_{c|s} + \beta x_{i|c,s} + \gamma_s + \epsilon_{i|c,s}$$

- Crossed: Participants respond to multiple stimuli (random effects for participant AND stimulus, not nested).

$$y_{ip} = \alpha_p + \gamma_i + \epsilon_{ip}$$

Parameter identification

- Identification: There exists a unique solution for the parameters given the data.

Parameter identification

- Identification: There exists a unique solution for the parameters given the data.
- Sometimes you can accidentally build a non-identified model.

Parameter identification

- Identification: There exists a unique solution for the parameters given the data.
- Sometimes you can accidentally build a non-identified model.
- Here is a common example:

$$y_{ip} = \alpha_p + \gamma_i + \epsilon_{ip} \quad \text{with} \quad \begin{cases} \alpha_p \sim \text{N}(\mu_\alpha, \sigma_\alpha^2) \\ \gamma_i \sim \text{N}(\mu_\gamma, \sigma_\gamma^2) \end{cases}$$

- If I increase μ_α by x and decrease μ_γ by x , the model makes identical predictions.
- This is a non-identified model.

Parameter identification

- Identification: There exists a unique solution for the parameters given the data.
- Sometimes you can accidentally build a non-identified model.
- Here is a common example:

$$y_{ip} = \alpha_p + \gamma_i + \epsilon_{ip} \quad \text{with} \quad \begin{cases} \alpha_p \sim \text{N}(\mu_\alpha, \sigma_\alpha^2) \\ \gamma_i \sim \text{N}(\mu_\gamma, \sigma_\gamma^2) \end{cases}$$

- If I increase μ_α by x and decrease μ_γ by x , the model makes identical predictions.
- This is a non-identified model.
- Fitting the model will go poorly – if there aren't convergence issues, you will see that the posterior distributions for μ_α and μ_γ are strongly negatively correlated.

Parameter identification

- Identification: There exists a unique solution for the parameters given the data.
- Sometimes you can accidentally build a non-identified model.
- Here is a common example:

$$y_{ip} = \alpha_p + \gamma_i + \epsilon_{ip} \quad \text{with} \quad \begin{cases} \alpha_p \sim \text{N}(\mu_\alpha, \sigma_\alpha^2) \\ \gamma_i \sim \text{N}(\mu_\gamma, \sigma_\gamma^2) \end{cases}$$

- If I increase μ_α by x and decrease μ_γ by x , the model makes identical predictions.
- This is a non-identified model.
- Fitting the model will go poorly – if there aren't convergence issues, you will see that the posterior distributions for μ_α and μ_γ are strongly negatively correlated.
- The solution is to add a **constraint**, for example $\mu_\alpha = 0$.

Generalization to cognitive modeling

$$\begin{aligned} y_{ip} &= \alpha_p + \gamma_i + \epsilon_{ip} \text{ with } \epsilon_{ip} \sim N(0, \sigma_{ip}^2) \\ \Leftrightarrow y_{ip} &\sim N(\alpha_p + \gamma_i, \sigma_{ip}^2) \end{aligned}$$

In cognitive modeling, data are rarely normally distributed.

How do we deal with this?

Lessons from psychometrics

- In psychometrics, we deal with binary data (e.g., correct/incorrect responses).

Lessons from psychometrics

- In psychometrics, we deal with binary data (e.g., correct/incorrect responses).
- A useful distribution for this is the **Bernoulli distribution**:

$$y_{ip} \sim \text{Bernoulli}(\pi_{ip})$$

Lessons from psychometrics

- In psychometrics, we deal with binary data (e.g., correct/incorrect responses).
- A useful distribution for this is the **Bernoulli distribution**:

$$y_{ip} \sim \text{Bernoulli}(\pi_{ip})$$

- Now we handle the **Bernoulli parameter** as a to-be-explained dependent variable:

$$\pi_{ip} = \alpha_p + \gamma_i \quad \Leftrightarrow \quad \pi_{ip} \sim \text{Bernoulli}(\alpha_p + \gamma_i)$$

Lessons from psychometrics

- In psychometrics, we deal with binary data (e.g., correct/incorrect responses).
- A useful distribution for this is the **Bernoulli distribution**:

$$y_{ip} \sim \text{Bernoulli}(\pi_{ip})$$

- Now we handle the **Bernoulli parameter** as a to-be-explained dependent variable:

$$\pi_{ip} = \alpha_p + \gamma_i \quad \Leftrightarrow \quad \pi_{ip} \sim \text{Bernoulli}(\alpha_p + \gamma_i)$$

- Psychometricians refer to α_p as the **person side** and γ_i as the **item side**.

Lessons from psychometrics

- In psychometrics, we deal with binary data (e.g., correct/incorrect responses).
- A useful distribution for this is the **Bernoulli distribution**:

$$y_{ip} \sim \text{Bernoulli}(\pi_{ip})$$

- Now we handle the **Bernoulli parameter** as a to-be-explained dependent variable:

$$\pi_{ip} = \alpha_p + \gamma_i \quad \Leftrightarrow \quad \pi_{ip} \sim \text{Bernoulli}(\alpha_p + \gamma_i)$$

- Psychometricians refer to α_p as the **person side** and γ_i as the **item side**.
- We can also handle either of these as a to-be-explained dependent variable:

$$\alpha_p = \beta_0 + \beta_1 z_p + \dots$$

$$\gamma_i = \zeta_0 + \zeta_1 q_i + \dots$$

See the pattern

First, in linear models (with normally distributed data), we did this decomposition:

$$y_{ip} \sim N(\mu_{ip}, \sigma_{ip}^2) \quad \Leftrightarrow \quad y_{ip} \sim N(\alpha_p + \gamma_i, \sigma_{ip}^2)$$

See the pattern

First, in linear models (with normally distributed data), we did this decomposition:

$$y_{ip} \sim N(\mu_{ip}, \sigma_{ip}^2) \quad \Leftrightarrow \quad y_{ip} \sim N(\alpha_p + \gamma_i, \sigma_{ip}^2)$$

Then, in psychometrics (with binary data), we did this decomposition:

$$y_{ip} \sim \text{Bernoulli}(\pi_{ip}) \quad \Leftrightarrow \quad y_{ip} \sim \text{Bernoulli}(\alpha_p + \gamma_i)$$

See the pattern

First, in linear models (with normally distributed data), we did this decomposition:

$$y_{ip} \sim N(\mu_{ip}, \sigma_{ip}^2) \quad \Leftrightarrow \quad y_{ip} \sim N(\alpha_p + \gamma_i, \sigma_{ip}^2)$$

Then, in psychometrics (with binary data), we did this decomposition:

$$y_{ip} \sim \text{Bernoulli}(\pi_{ip}) \quad \Leftrightarrow \quad y_{ip} \sim \text{Bernoulli}(\alpha_p + \gamma_i)$$

Now, in cognitive modeling (with complex distributions), we're going to do this:

$$y_{ip} \sim \text{Cognitive}(\psi_{ip}, \varsigma_{ip}) \quad \Leftrightarrow \quad y_{ip} \sim \text{Cognitive}(\alpha_p + \gamma_i, \varsigma_{ip})$$

See the pattern

First, in linear models (with normally distributed data), we did this decomposition:

$$y_{ip} \sim N(\mu_{ip}, \sigma_{ip}^2) \quad \Leftrightarrow \quad y_{ip} \sim N(\alpha_p + \gamma_i, \sigma_{ip}^2)$$

Then, in psychometrics (with binary data), we did this decomposition:

$$y_{ip} \sim \text{Bernoulli}(\pi_{ip}) \quad \Leftrightarrow \quad y_{ip} \sim \text{Bernoulli}(\alpha_p + \gamma_i)$$

Now, in cognitive modeling (with complex distributions), we're going to do this:

$$y_{ip} \sim \text{Cognitive}(\psi_{ip}, \varsigma_{ip}) \quad \Leftrightarrow \quad y_{ip} \sim \text{Cognitive}(\alpha_p + \gamma_i, \varsigma_{ip})$$

As we go, we'll declare some parameters to be focal (ψ_{ip}) and others nuisance (ς_{ip}).

Hierarchical cognitive models

This is just one way of decomposing model parameters:

$$y_{ip} \sim \text{Cognitive}(\psi_{ip}, s_{ip}) \quad \Leftrightarrow \quad y_{ip} \sim \text{Cognitive}(\alpha_p + \gamma_i, s_{ip})$$

Having a hierarchical structure is optional, but almost always the right call:

$$\alpha_p \sim \text{N}(\mu_\alpha, \sigma_\alpha^2)$$

Maybe $\mu_\alpha = \beta_0 + \beta_1 z_p + \dots$ really the world is your oyster at this point.

Putting it all together

We can now write a hierarchical model for any cognitive model.

Putting it all together

We can now write a hierarchical model for any cognitive model.

Say y_{ip} is person p 's data from condition i :

Putting it all together

We can now write a hierarchical model for any cognitive model.

Say y_{ip} is person p 's data from condition i : $y_{ip} = (\text{hit rate}_{ip}, \text{false alarm rate}_{ip})$

Putting it all together

We can now write a hierarchical model for any cognitive model.

Say y_{ip} is person p 's data from condition i : $y_{ip} = (\text{hit rate}_{ip}, \text{false alarm rate}_{ip})$

$$y_{ip} \sim \text{SDT}(d'_{ip}, c'_{ip})$$

$$d'_{ip} = \alpha_p + \gamma_i$$

$$\gamma_i = \zeta_0 + \zeta_1 \text{brightness}_i$$

$$\alpha_p \sim \text{N}(\mu_\alpha, \sigma_\alpha^2)$$

... where $\text{SDT}()$ is shorthand for the signal detection theory functions covered earlier.

Putting it all together

We can now write a hierarchical model for any cognitive model.

Say y_{ip} is person p 's data from condition i : $y_{ip} = (\text{hit rate}_{ip}, \text{false alarm rate}_{ip})$

$$y_{ip} \sim \text{SDT}(d'_{ip}, c'_{ip})$$

$$d'_{ip} = \alpha_p + \gamma_i$$

$$\gamma_i = \zeta_0 + \zeta_1 \text{brightness}_i$$

$$\alpha_p \sim \text{N}(\mu_\alpha, \sigma_\alpha^2)$$

... where $\text{SDT}()$ is shorthand for the signal detection theory functions covered earlier.

The rest is just a lot of typing.

Putting it all together

We can now write a hierarchical model for any cognitive model.

Say y_{ip} is person p 's data from condition i : $y_{ip} = (\text{hit rate}_{ip}, \text{false alarm rate}_{ip})$

$$y_{ip} \sim \text{SDT}(d'_{ip}, c'_{ip})$$

$$d'_{ip} = \alpha_p + \gamma_i$$

$$\gamma_i = \zeta_0 + \zeta_1 \text{brightness}_i$$

$$\alpha_p \sim \text{N}(\mu_\alpha, \sigma_\alpha^2)$$

... where $\text{SDT}()$ is shorthand for the signal detection theory functions covered earlier.

The rest is just a lot of typing.

This is an unidentified model, can you see why?

Assignment: Constrain an SDT model

Change how sensitivity (d') is modeled across conditions (k) in `conditional.py`.

Currently: conditional SDT model

- Independent d' per condition
- $d'_k \sim \mathcal{N}(0, 2^2)$
- Estimates an independent d'_k for each condition k
- Uses three free parameters to model d'_k

Desired: Condition **explanatory** SDT model

- Linear effect of difficulty C_k on d'_k
- $d'_k = \beta_0 + \beta_1 \times C_k$, with
 - $C = -1$ for 'Easy'
 - $C = 0$ for 'Medium'
 - $C = 1$ for 'Hard'
- Uses only two free parameters to model d'_k :
 - $\beta_0 \sim \mathcal{N}(0, 2^2)$
 - $\beta_1 \sim \mathcal{N}(0, 0.5^2)$
- Estimates intercept and slope; d'_k is derived.

Advanced assignment: Constrain a hierarchical SDT model

Change how group-level sensitivity (d') is modeled across conditions (k) in `full.py`.

In either case, at the individual level, parameters for person p in condition k come from a population distribution: $d'_{pk} \sim \mathcal{N}(\mu_k, \sigma_k^2)$, with $\sigma_k \sim \text{HalfNormal}(1)$.

Current full hierarchical model

- The parameters of the population distribution are independent per condition:
 - $\mu_k \sim \mathcal{N}(0, 2^2)$
- Estimates a separate mean μ_k for each condition.

Condition explanatory hierarchical model

- The mean of the d' population distribution is linear over conditions:
 - $\mu_k = \beta_0 + \beta_1 \times C_k$
- with
- $\beta_0 \sim \mathcal{N}(0, 2^2)$
 - $\beta_1 \sim \mathcal{N}(0, 0.5^2)$

Estimates intercept and slope; μ_k is derived.

Hierarchical (a.k.a. Multilevel) Modeling

Joachim Vandekerckhove

Spring 2025