Model fitting

Joachim Vandekerckhove

Winter 2025

Item Response Models

- The two-parameter logistic (2PL) model describes the probability of a correct response:

$$P(y|\theta, a, b_i) = \frac{1}{1 + e^{-a(\theta - b_i)}} \tag{1}$$

Item Response Models

- The two-parameter logistic (2PL) model describes the probability of a correct response:

$$P(y|\theta, a, b_i) = \frac{1}{1 + e^{-a(\theta - b_i)}} \tag{1}$$

- Parameters:

Item Response Models

- The two-parameter logistic (2PL) model describes the probability of a correct response:

$$P(y|\theta, a, b_i) = \frac{1}{1 + e^{-a(\theta - b_i)}} \tag{1}$$

- Parameters:
  - a: Discrimination (slope)

Item Response Models

- The two-parameter logistic (2PL) model describes the probability of a correct response:

$$P(y|\theta, a, b_i) = \frac{1}{1 + e^{-a(\theta - b_i)}} \tag{1}$$

- Parameters:
  - a: Discrimination (slope)
  - b_i: Difficulty (location) for item or condition i

Item Response Models

- The two-parameter logistic (2PL) model describes the probability of a correct response:

$$P(y|\theta, a, b_i) = \frac{1}{1 + e^{-a(\theta - b_i)}} \tag{1}$$

- Parameters:
  - a: Discrimination (slope)
  - b_i: Difficulty (location) for item or condition i
  - θ: Subject ability

Item Response Models

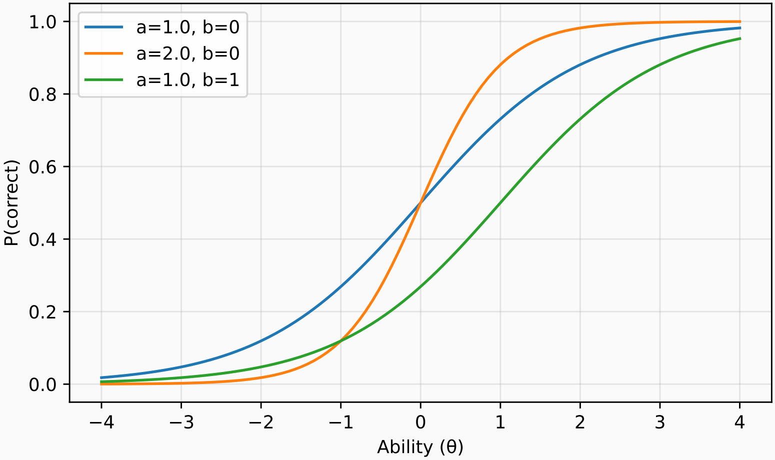
- The two-parameter logistic (2PL) model describes the probability of a correct response:

$$P(y|\theta, a, b_i) = \frac{1}{1 + e^{-a(\theta - b_i)}} \tag{1}$$

- Parameters:
  - a: Discrimination (slope)
  - b_i: Difficulty (location) for item or condition i
  - θ: Subject ability
- The three-parameter logistic (3PL) adds a guessing parameter c:

$$P(y|\theta, a, b_i, c) = c + (1 - c)\frac{1}{1 + e^{-a(\theta - b_i)}} \tag{2}$$

Visualizing the 2PL Model



Ordinary Least Squares

- Ordinary Least Squares (OLS) is a common estimation method:

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{N} (y_i - f(x_i, \beta))^2 \tag{3}$$

Ordinary Least Squares

- Ordinary Least Squares (OLS) is a common estimation method:

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{N} (y_i - f(x_i, \beta))^2 \tag{3}$$

- Properties:

Ordinary Least Squares

- Ordinary Least Squares (OLS) is a common estimation method:

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{N} (y_i - f(x_i, \beta))^2 \tag{3}$$

- Properties:
    - Minimizes squared differences between observed and predicted values ("loss")

Ordinary Least Squares

- Ordinary Least Squares (OLS) is a common estimation method:

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{N} (y_i - f(x_i, \beta))^2 \tag{3}$$

- Properties:
    - Minimizes squared differences between observed and predicted values ("loss")
    - Optimal when errors are normally distributed

Ordinary Least Squares

- Ordinary Least Squares (OLS) is a common estimation method:

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{N} (y_i - f(x_i, \beta))^2 \tag{3}$$

- Properties:
  - Minimizes squared differences between observed and predicted values ("loss")
  - Optimal when errors are normally distributed
  - Has closed-form solution for linear models

Ordinary Least Squares

- Ordinary Least Squares (OLS) is a common estimation method:

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{N} (y_i - f(x_i, \beta))^2 \tag{3}$$

- Properties:
    - Minimizes squared differences between observed and predicted values ("loss")
    - Optimal when errors are normally distributed
    - Has closed-form solution for linear models
    - Simple to implement and interpret

Squared Error Estimation

- Implementing OLS for the 2PL model, parameters are estimated by minimizing squared error:

$$\text{SSE}(\theta, a, \mathbf{b}, \mathbf{y}) = \sum_{i=1}^{N} (y_i - P(y_i|\theta, a, b_i))^2 \tag{4}$$

Squared Error Estimation

- Implementing OLS for the 2PL model, parameters are estimated by minimizing squared error:

$$\mathrm{SSE}(\theta, a, \mathbf{b}, \mathbf{y}) = \sum_{i=1}^{N} (y_i - P(y_i|\theta, a, b_i))^2 \tag{4}$$

- Where:

Squared Error Estimation

- Implementing OLS for the 2PL model, parameters are estimated by minimizing squared error:

$$\text{SSE}(\theta, a, \mathbf{b}, \mathbf{y}) = \sum_{i=1}^{N} (y_i - P(y_i|\theta, a, b_i))^2 \tag{4}$$

- Where:
  - y_i is the participant's observed response (0 or 1) to the i^{th} item of N items of varying difficulty b_i

Squared Error Estimation

- Implementing OLS for the 2PL model, parameters are estimated by minimizing squared error:

$$\mathrm{SSE}(\theta, a, \mathbf{b}, \mathbf{y}) = \sum_{i=1}^{N} (y_i - P(y_i|\theta, a, b_i))^2 \tag{4}$$

- Where:
  - y_i is the participant's observed response (0 or 1) to the i^{th} item of N items of varying difficulty b_i
  - $P(y_i|\theta, a, b_i)$ is the model prediction of y_i

Squared Error Estimation

- Implementing OLS for the 2PL model, parameters are estimated by minimizing squared error:

$$\text{SSE}(\theta, a, \mathbf{b}, \mathbf{y}) = \sum_{i=1}^{N} (y_i - P(y_i|\theta, a, b_i))^2 \tag{4}$$

- Where:
    - y_i is the participant's observed response (0 or 1) to the i^{th} item of N items of varying difficulty b_i
    - $P(y_i|\theta, a, b_i)$ is the model prediction of y_i
- This approach:

Squared Error Estimation

- Implementing OLS for the 2PL model, parameters are estimated by minimizing squared error:

$$\text{SSE}(\theta, a, \mathbf{b}, \mathbf{y}) = \sum_{i=1}^{N} (y_i - P(y_i|\theta, a, b_i))^2 \tag{4}$$

- Where:
  - y_i is the participant's observed response (0 or 1) to the i^{th} item of N items of varying difficulty b_i
  - $P(y_i|\theta, a, b_i)$ is the model prediction of y_i
- This approach:
  - Is intuitive and widely used in simple cases

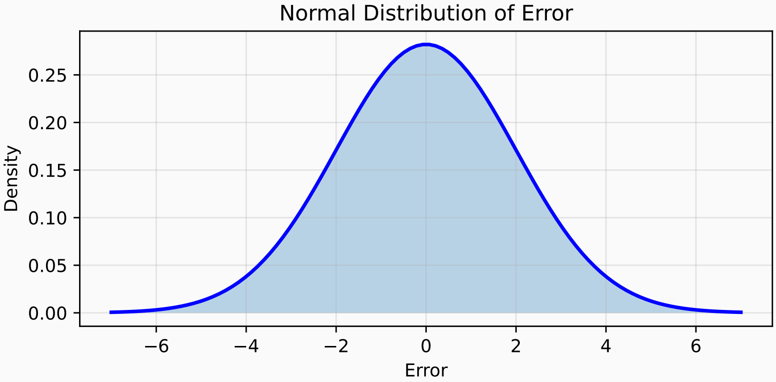
Squared Error Estimation

- Implementing OLS for the 2PL model, parameters are estimated by minimizing squared error:

$$\mathrm{SSE}(\theta, a, \mathbf{b}, \mathbf{y}) = \sum_{i=1}^{N} (y_i - P(y_i|\theta, a, b_i))^2 \tag{4}$$

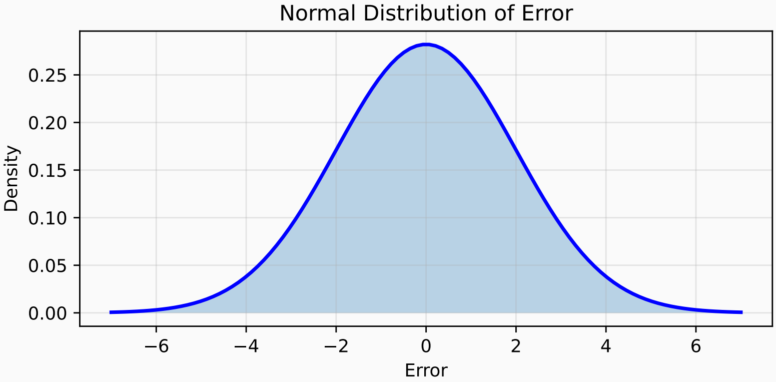
- Where:
    - y_i is the participant's observed response (0 or 1) to the i^{th} item of N items of varying difficulty b_i
    - $P(y_i|\theta, a, b_i)$ is the model prediction of y_i
- This approach:
    - Is intuitive and widely used in simple cases
    - Assumes normally distributed errors

Normality Assumption



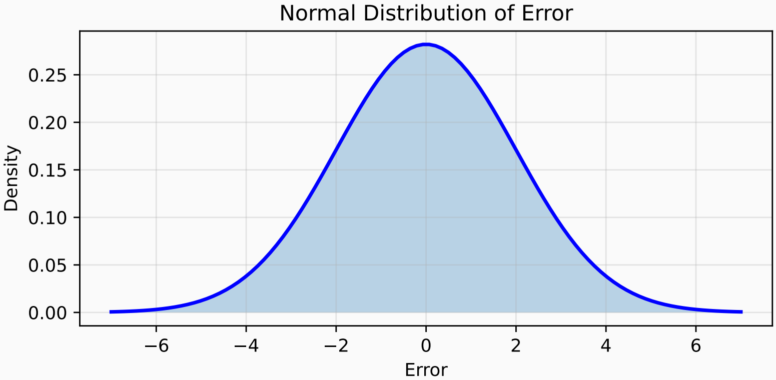
- May not be appropriate for non-normal errors

Normality Assumption



- May not be appropriate for non-normal errors
- Can give biased estimates for binary outcomes

Normality Assumption



- May not be appropriate for non-normal errors
- Can give biased estimates for binary outcomes
- Fine for a first pass, but not optimal for many applications

Maximum Likelihood Estimation

- For binary outcomes, likelihood is more appropriate:

$$L(\theta, a, \mathbf{b}, \mathbf{y}) = \prod_{i=1}^{N} P(y_i|\theta, a, b_i)^{y_i}(1 - P(y_i|\theta, a, b_i))^{1-y_i} \tag{5}$$

Maximum Likelihood Estimation

- For binary outcomes, likelihood is more appropriate:

$$L(\theta, a, \mathbf{b}, \mathbf{y}) = \prod_{i=1}^{N} P(y_i|\theta, a, b_i)^{y_i}(1 - P(y_i|\theta, a, b_i))^{1-y_i} \tag{5}$$

- Taking the log gives the log-likelihood function:

$$\ell(\theta, a, \mathbf{b}, \mathbf{y}) = \sum_{i=1}^{N} [y_i \log P(y_i|\theta, a, b_i) + (1 - y_i) \log(1 - P(y_i|\theta, a, b_i))] \tag{6}$$

Maximum Likelihood Estimation

- For binary outcomes, likelihood is more appropriate:

$$L(\theta, a, \mathbf{b}, \mathbf{y}) = \prod_{i=1}^{N} P(y_i|\theta, a, b_i)^{y_i}(1 - P(y_i|\theta, a, b_i))^{1-y_i} \quad (5)$$

- Taking the log gives the log-likelihood function:

$$\ell(\theta, a, \mathbf{b}, \mathbf{y}) = \sum_{i=1}^{N} [y_i \log P(y_i|\theta, a, b_i) + (1 - y_i) \log(1 - P(y_i|\theta, a, b_i))] \quad (6)$$

- Using the likelihood function instead of squared error appropriately accounts for the fact that some prediction errors are more severe than others

Regularization

- Regularization is a technique used to prevent overfitting in models

Regularization

- Regularization is a technique used to prevent overfitting in models
- It adds a penalty term to the loss function to discourage large parameter values

Regularization

- Regularization is a technique used to prevent overfitting in models
- It adds a penalty term to the loss function to discourage large parameter values
- This helps in cases where there is a lot of data and many parameters

Regularization

- Regularization is a technique used to prevent overfitting in models
- It adds a penalty term to the loss function to discourage large parameter values
- This helps in cases where there is a lot of data and many parameters
- Let's consider a 2PL model with predictors on item difficulty:

$$P(y|\theta, a, x, \beta) = \frac{1}{1 + e^{-a(\theta - f(x,\beta))}} \tag{7}$$

with $f(x, \beta) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k$

Regularization

- Regularization is a technique used to prevent overfitting in models
- It adds a penalty term to the loss function to discourage large parameter values
- This helps in cases where there is a lot of data and many parameters
- Let's consider a 2PL model with predictors on item difficulty:

$$P(y|\theta, a, x, \beta) = \frac{1}{1 + e^{-a(\theta - f(x,\beta))}} \tag{7}$$

with $f(x, \beta) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k$

  - Here, β is a vector of parameters for the predictors that replaces the item difficulty parameters **b**

Regularization

- Regularization is a technique used to prevent overfitting in models
- It adds a penalty term to the loss function to discourage large parameter values
- This helps in cases where there is a lot of data and many parameters
- Let's consider a 2PL model with predictors on item difficulty:

$$P(y|\theta, a, x, \beta) = \frac{1}{1 + e^{-a(\theta - f(x,\beta))}} \tag{7}$$

with $f(x, \beta) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k$

  - Here, β is a vector of parameters for the predictors that replaces the item difficulty parameters $\mathbf{b}$
  - Fewer predictors is better for model parsimony and interpretability

Regularization

- Regularization is a technique used to prevent overfitting in models
- It adds a penalty term to the loss function to discourage large parameter values
- This helps in cases where there is a lot of data and many parameters
- Let's consider a 2PL model with predictors on item difficulty:

$$P(y|\theta, a, x, \beta) = \frac{1}{1 + e^{-a(\theta - f(x,\beta))}} \tag{7}$$

with $f(x, \beta) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k$

- Here, β is a vector of parameters for the predictors that replaces the item difficulty parameters **b**
- Fewer predictors is better for model parsimony and interpretability
- Certainly, we need there to be fewer predictors than conditions, so $k < N$

LASSO and Ridge Regularization

$$\ell(\theta, a, x, \beta) = \sum_{i=1}^{N} [y_i \log P(y|\theta, a, x, \beta) + (1 - y_i) \log(1 - P(y|\theta, a, x, \beta))]$$

- LASSO (Least Absolute Shrinkage and Selection Operator) equation:

$$\ell_{\text{LASSO}}(\theta, a, x, \beta) = \ell(\theta, a, x, \beta) - \lambda \sum_{j=1}^{k} |\beta_j| \tag{8}$$

LASSO and Ridge Regularization

$$\ell(\theta, a, x, \beta) = \sum_{i=1}^{N} [y_i \log P(y|\theta, a, x, \beta) + (1 - y_i) \log(1 - P(y|\theta, a, x, \beta))]$$

- LASSO (Least Absolute Shrinkage and Selection Operator) equation:

$$\ell_{\text{LASSO}}(\theta, a, x, \beta) = \ell(\theta, a, x, \beta) - \lambda \sum_{j=1}^{k} |\beta_j| \tag{8}$$

- Ridge regularization equation:

$$\ell_{\text{Ridge}}(\theta, a, x, \beta) = \ell(\theta, a, x, \beta) - \lambda \sum_{j=1}^{k} \beta_j^2 \tag{9}$$

Interpretation of LASSO and Ridge Regularization

- LASSO:

Interpretation of LASSO and Ridge Regularization

- LASSO:
    - Shrinks the coefficients towards zero

Interpretation of LASSO and Ridge Regularization

- LASSO:
  - Shrinks the coefficients towards zero
  - Can set some coefficients to zero, performing “feature selection”

Interpretation of LASSO and Ridge Regularization

- LASSO:
  - Shrinks the coefficients towards zero
  - Can set some coefficients to zero, performing "feature selection"
  - Is equivalent to a Bayesian model with a Laplace prior on the coefficients

Interpretation of LASSO and Ridge Regularization

- LASSO:
  - Shrinks the coefficients towards zero
  - Can set some coefficients to zero, performing “feature selection”
  - Is equivalent to a Bayesian model with a Laplace prior on the coefficients
- Ridge:

Interpretation of LASSO and Ridge Regularization

- LASSO:
  - Shrinks the coefficients towards zero
  - Can set some coefficients to zero, performing "feature selection"
  - Is equivalent to a Bayesian model with a Laplace prior on the coefficients
- Ridge:
  - Shrinks the coefficients towards zero

Interpretation of LASSO and Ridge Regularization

- LASSO:
  - Shrinks the coefficients towards zero
  - Can set some coefficients to zero, performing "feature selection"
  - Is equivalent to a Bayesian model with a Laplace prior on the coefficients
- Ridge:
  - Shrinks the coefficients towards zero
  - Does not set any coefficients to zero

Interpretation of LASSO and Ridge Regularization

- LASSO:
    - Shrinks the coefficients towards zero
    - Can set some coefficients to zero, performing "feature selection"
    - Is equivalent to a Bayesian model with a Laplace prior on the coefficients
- Ridge:
    - Shrinks the coefficients towards zero
    - Does not set any coefficients to zero
    - Is equivalent to a Bayesian model with a Gaussian prior on the coefficients

Interpretation of LASSO and Ridge Regularization

- LASSO:
  - Shrinks the coefficients towards zero
  - Can set some coefficients to zero, performing "feature selection"
  - Is equivalent to a Bayesian model with a Laplace prior on the coefficients
- Ridge:
  - Shrinks the coefficients towards zero
  - Does not set any coefficients to zero
  - Is equivalent to a Bayesian model with a Gaussian prior on the coefficients
- Regularization is a form of Bayesian inference

Interpretation of LASSO and Ridge Regularization

- LASSO:
  - Shrinks the coefficients towards zero
  - Can set some coefficients to zero, performing "feature selection"
  - Is equivalent to a Bayesian model with a Laplace prior on the coefficients
- Ridge:
  - Shrinks the coefficients towards zero
  - Does not set any coefficients to zero
  - Is equivalent to a Bayesian model with a Gaussian prior on the coefficients
- Regularization is a form of Bayesian inference
- Can be combined with likelihoods or more basic loss functions (like squared error)

Conclusion

- The 2PL model is a close relative to Signal Detection Theory

Conclusion

- The 2PL model is a close relative to Signal Detection Theory
- You can fit the 2PL model using maximum likelihood estimation

Conclusion

- The 2PL model is a close relative to Signal Detection Theory
- You can fit the 2PL model using maximum likelihood estimation
- You can regularize the model using LASSO and Ridge regularization

Conclusion

- The 2PL model is a close relative to Signal Detection Theory
- You can fit the 2PL model using maximum likelihood estimation
- You can regularize the model using LASSO and Ridge regularization
- This is very similar to Bayesian parameter estimation

Conclusion

- The 2PL model is a close relative to Signal Detection Theory
- You can fit the 2PL model using maximum likelihood estimation
- You can regularize the model using LASSO and Ridge regularization
- This is very similar to Bayesian parameter estimation
- Soon, we will create python classes for:

Conclusion

- The 2PL model is a close relative to Signal Detection Theory
- You can fit the 2PL model using maximum likelihood estimation
- You can regularize the model using LASSO and Ridge regularization
- This is very similar to Bayesian parameter estimation
- Soon, we will create python classes for:
    - the 2PL model

Conclusion

- The 2PL model is a close relative to Signal Detection Theory
- You can fit the 2PL model using maximum likelihood estimation
- You can regularize the model using LASSO and Ridge regularization
- This is very similar to Bayesian parameter estimation
- Soon, we will create python classes for:
    - the 2PL model
    - the regularized 2PL model

Conclusion

- The 2PL model is a close relative to Signal Detection Theory
- You can fit the 2PL model using maximum likelihood estimation
- You can regularize the model using LASSO and Ridge regularization
- This is very similar to Bayesian parameter estimation
- Soon, we will create python classes for:
    - the 2PL model
    - the regularized 2PL model
- These classes will allow us to fit the model to data and perform inference

Conclusion

- The 2PL model is a close relative to Signal Detection Theory
- You can fit the 2PL model using maximum likelihood estimation
- You can regularize the model using LASSO and Ridge regularization
- This is very similar to Bayesian parameter estimation
- Soon, we will create python classes for:
    - the 2PL model
    - the regularized 2PL model
- These classes will allow us to fit the model to data and perform inference
- I will add a twist that is likely to confuse an AI assistant

Model fitting

Joachim Vandekerckhove

Winter 2025