CLCA for Hypothesis Testing in Surveys

Issues with Surveys

- Some information cannot be extracted/questioned in the straight-forward way
  - Example: race, attitudes, religious beliefs, etc
- Survey questionnaires should be validated to confirm that they measure the same construct
  - Example: Internal consistency approach

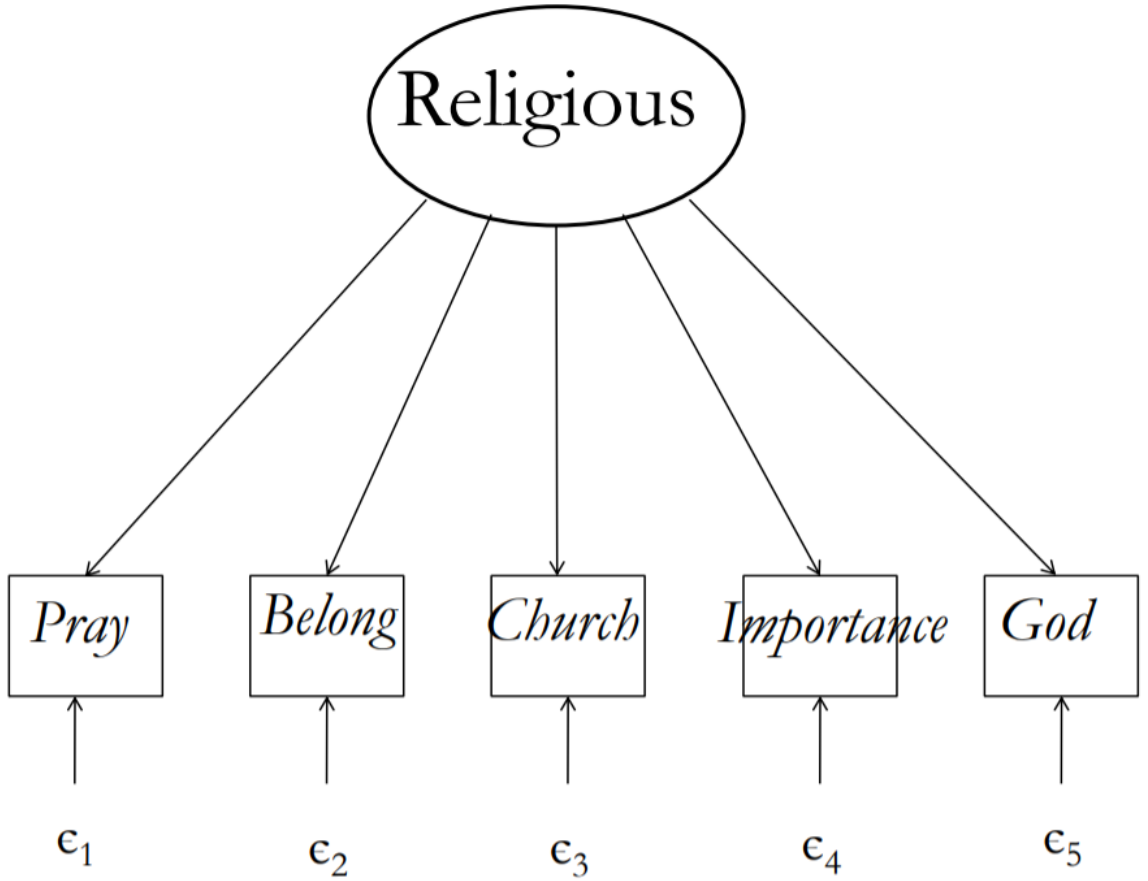
Concept Extraction Example

Concept: Religious Commitment

Questions on: Church attendance, denomination, praying, believe in God, importance of religion in one's life, etc

All questions are highly correlated -> There is a latent variable behind them

Concept Extraction Example

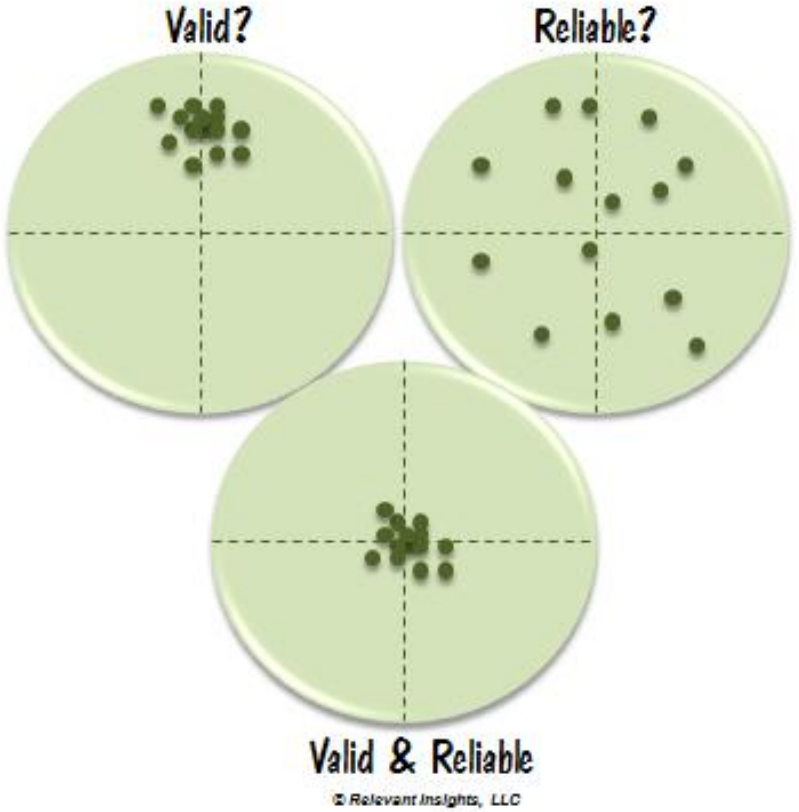


Internal Consistency

We want to ensure consistency of the questions which is the combination of

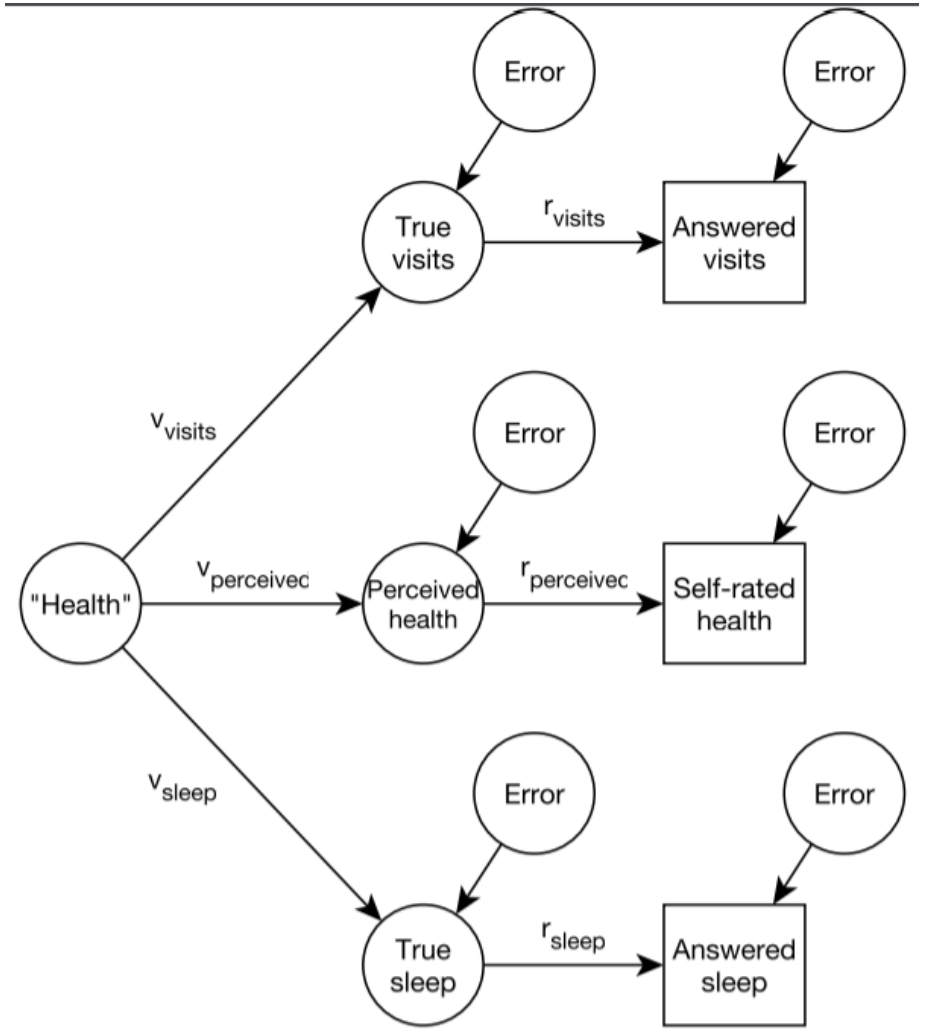
- validity – do questions measure what they are supposed to?
- reliability – do questions elicit the same information under the same conditions?

Validity and Reliability



Internal Consistency Example

- Concept: Health
- Questions: how many doctor's visits, how is your health in general, how well do you sleep



Estimations methods

Latent Concept: η	**Manifest measure:** y_i *Categorical*	*Continuous*
Categorical	Latent Class	Latent Profile
Continues	Latent Trait/ IRT	Factor Analysis

LCA

- **Assumption:** Observed co-variation between observed variables is due to unobserved, true variable. That is, observed values are conditionally independent given latent variables.
- **Model:**

$$f(Y_i;\pi_r) = \prod_{j=1}^{J}\prod_{k=1}^{K_j}(\pi_{jrk})^{Y_{ijk}}.$$

- where:
  - J – number of observed polytomous categorical variables
  - K_j is possible outcomes for individuals i = 1..N
  - Y_{ijk} is observed values of the J manifest variables such that Y_{ijk} = 1 if respondent *i* gives the *k*th response to the *j*th variable and 0 otherwise
  - R – number of latent classes
  - π_{jrk} denote the class-conditional probability that an observation in class *r* produces the *k*th outcome on the *j*th variable

LCA

Restrictions:

$$\sum_{k=1}^{K_j} \pi_{jrk} = 1 \qquad \sum_r p_r = 1$$

where p_r is the "prior" probabilities of latent class membership

Probability density function

$$\mathsf{P}(Y_i|\pi, p) = \sum_{r=1}^{R} p_r \prod_{j=1}^{J} \prod_{k=1}^{K_j} (\pi_{jrk})^{Y_{ijk}}$$

Posterior probability

$$\widehat{\mathsf{P}}(r_i|Y_i) = \frac{\hat{p}_r f(Y_i; \hat{\pi}_r)}{\sum_{q=1}^{R} \hat{p}_q f(Y_i; \hat{\pi}_q)}$$

LCA: Parameter estimation

Total number of parameters

$$R\sum_j(K_j - 1) + (R - 1)$$

If this number exceeds either the total number of observations, or one fewer than the total number of cells in the cross-classification table of the manifest variables, then the latent class model will be unidentified.

Estimation by maximization log-likelihood

$$\ln L = \sum_{i=1}^{N} \ln \sum_{r=1}^{R} p_r \prod_{j=1}^{J} \prod_{k=1}^{K_j} (\pi_{jrk})^{Y_{ijk}}$$

Might be done using EM-algorithm

CLCA

- CLCA is Conformation Latent Class Analysis
- It’s extension of LCA and aims to verify hypothesis based on provided data
- It assumes that all variable is binomial
- Currently there are no free software / library which provides this functionality

What are the hypothesis?

- In terms of CLCA hypothesis is a set of restrictions on π_{jr} and / or on p_r
- There are several types of them, which can be mixed:

1. $c_1 * \pi_{jr} + c_2 < c_3 * \pi_{j'r} + c_4$
2. $c_1 * \pi_{jr} + c_2 < c_3 * \pi_{jr'} + c_4$
3. $c_1 < \pi_{jr} < c_2$
4. $c_1 < p_r < c_2$
5. $c_1 * p_r + c_2 < c_3 * p_{r'} + c_4$

- It's possible to apply restrictions to π or p or both of them at the same time.

Example

- Consider last python developers survey, where we asked questions on frameworks, purpose of using python and additional technologies.
- The classes are pythonist specialization: web, data analysis, devOps, etc
- We have no idea on proportion of the classes but may say how they should answer the question

Example

- On preprocessing step each check-box or radio-button answer is turned into “Yes/No” question

- Examples of constrains:
  - $0.5 < \pi_{\text{django, django.dev}} < 1$
  - $\pi_{\text{other.web, django.dev}} < 0.05$
  - $\pi_{\text{for.data.analysis, django.dev}} < 0.5 * \pi_{\text{for.data.analysis, data.analyst}}$
  - $0 < \pi_{\text{docker, django.dev}} < 1$

Math behind CLCA

- Prior and posterior distributions, from which it's sampled, are truncated by the constraints
- Gibbs sampler is used to obtain value of π_{jr} and p_r
- When the model is computed, we do Gibbs sampling again based on provided posterior probabilities
- Answers are generated for each sample and then we estimate pseudo likelihood ratio (PLR1) between them and the sampled parameters
- PLR2 is also estimated for original data and sampled parameters
- Each time we compare PLR1 >= PLR2
- The share of positive comparisons is p-value. That is, if it's higher than 5% then the hypothesis is failed to be rejected.

Likelihood Ratio

- Likelihood Ratio is $LR(X,\varepsilon) = -2\ \sum_{p=1}^{P} N(x_p)\log[M(x_p|\varepsilon)\ /N(x_p)]$, where
  - X is our data, P denotes the number of different response vectors x_p, N is number of response vectors x_p observed in the data matrix for which LR is computed, ε is paramters
  - LR is rather sensitive to the presence of outliers in the data even if the set of restrictions H provide a good description of the response process
- We just want to know if the model holds for most of the sample without having to bother about a (relatively) small number of outliers.
- The latter is achieved using the pseudo likelihood ratio (PLR) test which is sensitive with respect to misspecifications of the restrictions and the number of latent classes, but is robust with respect to outliers

Pseudo Likelihood Ratio

- $PLR(X,\varepsilon) = -2\log\left(\frac{PL_H}{PL_M}\right) =$

$-2\sum_{j\neq j'}\sum_{v=0}^{1}\sum_{w=0}^{1} N\left(X_j = v, X_j' = w\right)\log[M(X_j = v, X_j' = w|\varepsilon)/N\left(X_j = v, X_j' = w\right),$

- where PL_H denotes the pseudo likelihood of CLCA, and PL_M the pseudo likelihood of the corresponding multinomial model:
  - $PL_H = \prod_{j\neq j'}\prod_{v=0}^{1}\prod_{w=0}^{1}\left[\frac{M\left(X_j = v, X_j' = w\middle|\varepsilon\right)}{N}\right]^{N\left(X_j=v,X_j'=w\right)}$
  - $PL_M = \prod_{j\neq j'}\prod_{v=0}^{1}\prod_{w=0}^{1}\left[\frac{N\left(X_j = v, X_j' = w\middle|\varepsilon\right)}{N}\right]^{N\left(X_j=v,X_j'=w\right)}$
  - $M\left(x_p\middle|\varepsilon\right) = N\sum_{q}^{Q} P_q\left(x_p\right)\omega_q$

Demo

Q&A