Covariates and distal outcomes of latent profiles and classes with the latents R package

Mohammed Saqr, University of Eastern Finland
Sonsoles López-Pernas, University of Eastern Finland

26 November 2026

Contents

Data 2
The measurement model 3
Classification error 3
Class means of an external variable 4
Predicting class membership 5
One-step estimation 5
Direct effects on indicators 6
Staged estimation 7
Reference 8
Assumptions 9
References 9

The distinction between one-step and three-step approaches concerns whether external variables are allowed to influence the estimation and definition of latent profiles. External variables are not included among the indicators used to identify the profiles but are introduced to examine their relationships with profile membership. These variables may function as predictors of profile membership, covariates, or distal outcomes.

In the three-step approach, the measurement model is estimated first, using only the indicators specified to define the profiles. Once the profiles have been identified, individuals are assigned to their most likely profile based on their posterior membership probabilities. In the subsequent step, external variables are incorporated into the analysis while correcting for the classification errors associated with this assignment. This correction is important because profile membership is latent and individuals cannot necessarily be classified with complete certainty. By accounting for this uncertainty, the three-step approach reduces the bias that would arise from treating assigned profile membership as if it were directly observed. Importantly, the original measurement model remains unchanged: the profile-specific means and other measurement parameters are not re-estimated in response to the external variables.

In contrast, the one-step approach incorporates external variables directly into the latent profile model, estimating the measurement and structural components simultaneously. Consequently, the external variables can influence the estimation of profile membership and, depending on the model specification, alter the profile characteristics and classification of individuals. Although this approach allows the relationships between latent profiles and external variables to be estimated jointly, it may produce a different profile solution from the model estimated using only the original measurement indicators (Bolck et al., 2004; Vermunt, 2010).

This distinction is particularly relevant in person-centred research, where profiles are interpreted as substantively meaningful subgroups characterized by distinct patterns across a set of indicators (Saqr et al., 2024; Scrucca et al., 2024). Researchers often seek to investigate the characteristics, antecedents, and consequences of these subgroups after establishing their measurement structure. In such cases, allowing external variables to influence profile estimation may change the very subgroups that the subsequent analyses are intended to investigate. The three-step approach addresses this concern by preserving the original profile solution while accounting for classification uncertainty. Staged estimation extends the same principle to group classes in multilevel models, keeping the profile solution fixed while the group classes are estimated. Thus, three-step methods are particularly useful when the objective is to investigate relationships with external variables without changing the substantive meaning of the previously identified profiles.

This guide demonstrates these approaches in a two-level latent profile model of course engagement, in which course enrolments are nested within students. The measurement model is first estimated from the engagement indicators alone, and its classification-error matrix is examined, since this matrix is the basis of every three-step correction. The previous grade before each enrolment is then related to the profiles in three ways: as a distal variable whose mean is compared across profiles with `three_step()`, using the BCH and maximum likelihood corrections and, for comparison, uncorrected modal assignment; as a predictor of profile membership with `r3step()`; and as a covariate estimated jointly with the profiles in a one-step model through `profile_covariates`. The guide then shows how a covariate can affect an indicator directly rather than through profile membership, a form of differential item functioning that can otherwise be mistaken for a difference between profiles, and concludes with staged estimation, in which the group classes are estimated with the profiles held fixed through `fit_staged()`.

Data

`course_engagement` has one row per course enrolment: 1,422 enrolments from 106 students. `student` identifies the group, and five activity measures (`browse`, `lectures`, `forum_read`, `forum_post`, `attendance`) are the indicators. `previous_grade` is the standardized grade from the course before each enrolment; it is not an indicator, so it can serve as an external variable.

```
library(latents)
```

The measurement model

`multilpa()` fits the profiles without the external variable, with `vars` for the indicators, `id` for the group, `n_profiles`, `n_group_classes`, `n_starts` and `seed`. The within-profile covariances are estimated (`covariance_model = "full"`), because the indicators are correlated within profiles (see the multilevel latent profile analysis vignette).

```
fit <- multilpa(course_engagement,
                vars = c("browse", "lectures", "forum_read", "forum_post",
                         "attendance"),
                id = "student", n_profiles = 2, n_group_classes = 2,
                covariance_model = "full",
                n_starts = 3, seed = 1)
fit
#> Two-level latent profile analysis: 2 profiles, 2 group classes
#> 1422 individuals in 106 groups; varying full residual covariance (VVV)
#> Log likelihood: -8313.111554 | AIC: 16712.223 | BIC (groups): 16826.751
#> Converged: TRUE | iterations: 12 | best log likelihood reached by 3 of 3 starts (3 converged)
#> Smallest standardized Gaussian eigenvalue: 0.271 (profiles, profile_2); Gaussian mass: 583.2; weight effect
#>
#> Class sizes   profile_1 59.0% (839)   profile_2 41.0% (583)
#> Group classes group_class_1 32.5% (34)   group_class_2 67.5% (72)
#>
#> Means
#>   indicator profile_1 profile_2
#>      browse     0.531    -0.764
#>    lectures     0.422    -0.607
#>  forum_read     0.613    -0.881
#>  forum_post     0.527    -0.758
#>  attendance     0.632    -0.908
#>
#> Variances and standard errors: get_results(x, "profiles").
#> Every other table: get_results(x, what = ), or get_results(x, "all").
```

Classification error

`get_results()` with `"classification_errors"` returns the classification-error matrix, with `level` for the observations or the groups. Each row gives the probability of each assigned class for cases in a true class; the three-step methods use it to correct for misclassification.

```
get_results(fit, "classification_errors", level = "individuals")
#>         level true_class assigned_class probability
#> 1 individuals          1              1      0.9815
#> 2 individuals          1              2      0.0185
#> 3 individuals          2              1      0.0321
#> 4 individuals          2              2      0.9679
```

Enrolments of the first profile are assigned to it with probability 0.982 and those of the second with probability 0.968, so the corrections below are small but not negligible.

Class means of an external variable

`three_step()` estimates the mean of an external variable in each profile, with `data` for the data frame and `outcome` for the variable. The default method, BCH, corrects for classification error (Bolck et al., 2004; Bakk & Vermunt, 2016); standard errors are clustered on groups.

```
three_step(fit, data = course_engagement, outcome = "previous_grade")
#> Distal outcome by class (bch, 95% CI)
#>
#> Class   Mean          95% CI     SE  Effective n
#> -----  -----  --------------  -----  -----------
#>     1   0.23  [ 0.16,  0.30]  0.035       803.34
#>     2  -0.33  [-0.40, -0.27]  0.035       548.42
```

Previous grades average 0.23 standard deviations above the mean before enrolments of the first profile and 0.33 below it before enrolments of the second. `contrast = "pairs"` returns each difference between profile means with a Benjamini-Hochberg adjusted p-value (Benjamini & Hochberg, 1995).

```
three_step(fit, data = course_engagement, outcome = "previous_grade",
           contrast = "pairs")
#> Distal outcome: class differences (bch, 95% CI)
#>
#> Class  vs  Difference          95% CI      p  p (BH)
#> -----  --  ----------  --------------  -----  ------
#>     2   1       -0.57  [-0.67, -0.47]  <.001   <.001
```

The difference is -0.57 standard deviations (95% interval -0.67 to -0.47).

Maximum likelihood is the other correction. With `method = "ml"`, the assigned profile becomes an indicator of the true profile whose classification errors are fixed at those of the measurement model, and the external variable is normal within each profile, with its own variance. The table adds that variance. When the variable is close to normal within profiles, ML is usually more efficient than BCH (Vermunt, 2010), although with classification as accurate as here the standard errors are nearly equal; when it is skewed, BCH is the more robust choice.

```
three_step(fit, data = course_engagement, outcome = "previous_grade",
           method = "ml")
#> Distal outcome by class (ml, varying variances, 95% CI)
#>
#> Class   Mean          95% CI     SE  Variance  Effective n
#> -----  -----  --------------  -----  --------  -----------
#>     1   0.23  [ 0.17,  0.30]  0.035      0.93       838.76
#>     2  -0.34  [-0.41, -0.27]  0.036      0.91       583.24
```

ML gives nearly the same means (0.23 and -0.34), with variances of 0.93 and 0.91 within the profiles.

`method = "modal"` shows the effect of ignoring classification error: it treats each case's most probable profile as known.

```
three_step(fit, data = course_engagement, outcome = "previous_grade",
           method = "modal")
#> Distal outcome by class (modal, 95% CI)
#>
#> Class   Mean          95% CI     SE  Effective n
```

```
#> -----  -----  --------------  -----  -----------
#>     1   0.22  [ 0.15,  0.29]  0.034          842
#>     2  -0.32  [-0.39, -0.25]  0.035          580
```

The modal means (0.22 and -0.32) are slightly closer together than the corrected ones, the attenuation that classification error produces; with classification as accurate as here, the difference is small.

Predicting class membership

`r3step()` regresses profile membership on a covariate, with `data` for the data frame and `covariates` for the predictors. It returns multinomial logit coefficients, corrected for classification error. `vcov_type = "robust"`, the default, clusters the standard errors on groups.

```
r3step(fit, data = course_engagement, covariates = "previous_grade",
       vcov_type = "robust")
#>        level   outcome          term estimate standard_error statistic  p_value p_adjusted
#> 1 individuals profile_1    (Intercept)    0.394        0.1362      2.89 3.80e-03         NA
#> 2 individuals profile_1 previous_grade    0.615        0.0612     10.05 9.55e-24   9.55e-24
#>   conf_low conf_high
#> 1    0.127     0.661
#> 2    0.495     0.735
```

A previous grade one standard deviation higher raises the log odds of the first, more active profile against the second by 0.615 (95% interval 0.495 to 0.735), an odds ratio of about 1.85.

One-step estimation

`multilpa()` with `profile_covariates` estimates the covariate together with the profiles, as a predictor of profile membership. `group_covariates` does the same for group-class membership and needs a variable that is constant within groups. Covariates must have no missing values; missing indicators are integrated out with `missing = "fiml"`.

```
one_step <- multilpa(course_engagement,
                     vars = c("browse", "lectures", "forum_read",
                              "forum_post", "attendance"),
                     id = "student", n_profiles = 2, n_group_classes = 2,
                     covariance_model = "full",
                     profile_covariates = "previous_grade", n_starts = 3,
                     seed = 1)
one_step
#> Multilevel LPA with covariates: 2 profiles, 2 group classes
#> Log likelihood -8295.460308; AIC 16678.921; BIC (groups) 16796.112; converged TRUE
#> Smallest standardized Gaussian eigenvalue: 0.2705 (profiles, profile_2); Gaussian mass: 582.2; weight effec
#>
#> Class sizes   profile_1 59.1% (840)   profile_2 40.9% (582)
#> Group classes group_class_1 31.8% (34)   group_class_2 68.2% (72)
#>
#> Means
#>   indicator profile_1 profile_2
#>      browse     0.530    -0.765
#>    lectures     0.424    -0.612
```

```
#>  forum_read     0.612    -0.882
#>  forum_post     0.525    -0.757
#>  attendance     0.631    -0.910
#>
#> Variances and standard errors: get_results(x, "profiles").
#> Every other table: get_results(x, what = ), or get_results(x, "all").
```

`get_results()` with `"coefficients"` returns the membership coefficients, and `parameter_inference()` the standard errors and intervals of every parameter, beginning with the measurement means.

```
get_results(one_step, "coefficients")
#>     level       outcome           term parameter estimate
#> 1 profile     profile_1  group_class_1     logit   -1.577
#> 2 profile     profile_1  group_class_2     logit    1.316
#> 3 profile     profile_1 previous_grade     logit    0.447
#> 4   group group_class_1    (Intercept)     logit   -0.764
head(parameter_inference(one_step), 10)
#>          level   outcome       term parameter estimate standard_error statistic   p_value
#> 1  measurement profile_1     browse      mean    0.530         0.0281      18.9  1.45e-79
#> 2  measurement profile_1   lectures      mean    0.424         0.0328      12.9  2.98e-38
#> 3  measurement profile_1 forum_read      mean    0.612         0.0257      23.8 5.55e-125
#> 4  measurement profile_1 forum_post      mean    0.525         0.0303      17.3  3.15e-67
#> 5  measurement profile_1 attendance      mean    0.631         0.0238      26.5 6.98e-155
#> 6  measurement profile_2     browse      mean   -0.765         0.0323     -23.7 4.29e-124
#> 7  measurement profile_2   lectures      mean   -0.612         0.0318     -19.2  2.87e-82
#> 8  measurement profile_2 forum_read      mean   -0.882         0.0273     -32.4 1.20e-229
#> 9  measurement profile_2 forum_post      mean   -0.757         0.0285     -26.6 2.16e-155
#> 10 measurement profile_2 attendance      mean   -0.910         0.0283     -32.1 3.91e-226
#>    p_adjusted conf_low conf_high
#> 1    1.45e-79    0.475     0.585
#> 2    2.98e-38    0.360     0.488
#> 3   5.55e-125    0.561     0.662
#> 4    3.15e-67    0.465     0.584
#> 5   6.98e-155    0.584     0.677
#> 6   4.29e-124   -0.828    -0.702
#> 7    2.87e-82   -0.674    -0.549
#> 8   1.20e-229   -0.936    -0.829
#> 9   2.16e-155   -0.813    -0.701
#> 10  3.91e-226   -0.965    -0.854
```

The one-step coefficient of previous grade is 0.447, smaller than the three-step coefficient, because it is conditional on the group class and the profiles are estimated together with it.

Direct effects on indicators

A covariate can also act on an indicator rather than on membership: students with better previous grades may attend more within every profile. Left out of the model, such an effect is read as a difference between the profiles (differential item functioning). `multilpa()` with `direct_effects` estimates it, one `"indicator ~ covariate"` term per effect. The covariate shifts the indicator's mean by the same amount in every profile, and the profile means are then the means at a covariate value of zero, here the average standardized grade. Direct effects are estimated in a profile model with a diagonal covariance structure (`EEI` or `VVI`), so the comparison below uses the diagonal model with and without them.

```
diagonal <- multilpa(course_engagement,
                     vars = c("browse", "lectures", "forum_read",
                              "forum_post", "attendance"),
                     id = "student", n_profiles = 2, n_group_classes = 2,
                     n_starts = 3, seed = 1)
direct <- multilpa(course_engagement,
                   vars = c("browse", "lectures", "forum_read",
                            "forum_post", "attendance"),
                   id = "student", n_profiles = 2, n_group_classes = 2,
                   direct_effects = c("attendance ~ previous_grade",
                                      "forum_post ~ previous_grade"),
                   n_starts = 3, seed = 1)
get_results(direct, "direct_effects")
#>    indicator      covariate estimate standard_error statistic p_value conf_low conf_high
#> 1 attendance previous_grade  0.03626         0.0178     2.040  0.0414  0.00142    0.0711
#> 2 forum_post previous_grade  0.00375         0.0209     0.179  0.8578 -0.03724    0.0447
compare_models(without = diagonal, direct = direct)
#> Model comparison
#>
#> Model   Profiles  Group classes  k   LogLik     BIC  ΔBIC  Weight     ICL  Entropy  Min AvePP  < 0.8  Group ent
#> ------- -------- ------------- -- -------- -------- ---- ------ -------- ------- --------- ----- ---
#> without       2             2 23 -8439.82 16986.89 0.00  0.930 17175.69    0.94     0.98     0        0.94
#> direct        2             2 25 -8437.74 16992.07 5.17  0.070 17188.94    0.94     0.98     0        0.94
```

Attendance rises by 0.036 per unit of previous grade within each profile (95% interval 0.001 to 0.071), and forum posting by 0.004 (interval -0.037 to 0.045). The effects are small against the profile differences, and BIC is lower without them (16986.9 against 16992.1), so the profile differences on these indicators are not produced by direct effects of previous grades.

Staged estimation

`fit_staged()` holds the profiles fixed while the group classes are estimated, with the arguments of `multilpa()` that define the profiles. It first fits the profiles with one group class, then estimates the group classes with the profile means and covariance matrices held fixed.

```
staged <- fit_staged(course_engagement,
                     vars = c("browse", "lectures", "forum_read",
                              "forum_post", "attendance"),
                     id = "student", n_profiles = 2, n_group_classes = 2,
                     covariance_model = "full",
                     n_starts = 3, seed = 1)
staged
#> Two-level latent profile analysis: 2 profiles, 2 group classes
#> 1422 individuals in 106 groups; varying full residual covariance (VVV)
#> Log likelihood: -8313.829076 | AIC: 16633.658 | BIC (groups): 16641.648
#> Converged: TRUE | iterations: 9 | best log likelihood reached by 2 of 3 starts (3 converged)
#> Held fixed, not estimated here: means, variances (first stage)
#> Parameters estimated here: 3; with the held measurement: 43
#> Smallest standardized Gaussian eigenvalue: 0.2696 (profiles, profile_2); Gaussian mass: 579.9; weight effec
#>
#> Class sizes   profile_1 59.2% (842)   profile_2 40.8% (580)
#> Group classes group_class_1 67.5% (72)   group_class_2 32.5% (34)
```

```
#>
#> Means
#>   indicator profile_1 profile_2
#>      browse     0.520    -0.761
#>    lectures     0.424    -0.621
#>  forum_read     0.605    -0.885
#>  forum_post     0.522    -0.764
#>  attendance     0.625    -0.914
#>
#> Variances and standard errors: get_results(x, "profiles").
#> Every other table: get_results(x, what = ), or get_results(x, "all").
```

`get_results()` with `"stages"` reports what each stage estimated. `parameters` counts the parameters estimated at a stage, and `parameters_with_measurement` adds those held fixed.

```
get_results(staged, "stages")
#>      stage group_classes        fixed log_likelihood parameters parameters_with_measurement
#> 1 measurement            1            <NA>         -8490        41                        41
#> 2  membership            2 means, variances         -8314         3                        43
#>   converged
#> 1      TRUE
#> 2      TRUE
```

The first stage estimates the 41 parameters of the profiles with one group class; the second estimates 3 membership parameters with the profiles held fixed.

Reference

The calls below list other functions for external variables and fixed parameters. Each comment states what the call returns.

```
get_results(fit, "classification_errors", level = "groups")  # group-level errors
get_results(fit, "bch_weights", level = "individuals")       # BCH weights per case
three_step(fit, data = course_engagement, outcome = "previous_grade",
           method = "proportional")        # weighting by posterior probabilities
three_step(fit, data = course_engagement, outcome = "previous_grade",
           method = "ml", variance = "equal")  # ML with one shared variance
three_step(fit, data = course_engagement, outcome = "previous_grade",
           vcov_type = "independent")      # standard errors without clustering
r3step(fit, data = course_engagement, covariates = "previous_grade",
       vcov_type = "observed")             # observed-information standard errors
measurement <- multilpa(course_engagement,
                        vars = c("browse", "lectures", "forum_read",
                                 "forum_post", "attendance"),
                        id = "student", n_profiles = 2,
                        n_group_classes = 1, covariance_model = "full",
                        seed = 1)  # a one-class measurement fit
fit_staged(course_engagement,
           vars = c("browse", "lectures", "forum_read", "forum_post",
                    "attendance"),
           id = "student", n_profiles = 2, n_group_classes = 2,
           covariance_model = "full",
```

```
          measurement = measurement)      # stage two from that fit
multilpa(course_engagement,
         vars = c("browse", "lectures", "forum_read", "forum_post",
                  "attendance"),
         id = "student", n_profiles = 2, n_group_classes = 2,
         covariance_model = "full",
         start = starting_values(fit, what = "measurement"),
         fixed = "variances")             # hold chosen parameters fixed
parameter_inference(staged)               # standard errors of the estimated stage
```

Assumptions

- Three-step methods assume classification errors are independent of the external variable given the true class, and treat the error matrix as known.
- The ML correction also assumes the external variable is normal within each class; BCH does not.
- One-step coefficients are conditional on the group class and can change the profiles, so they need not equal the three-step coefficients.
- Staged estimation does not carry first-stage uncertainty into the second stage, and its likelihood is not the joint maximum.
- All associations are observational.

References

Bakk, Z., & Vermunt, J. K. (2016). Robustness of stepwise latent class modeling with continuous distal outcomes. *Structural Equation Modeling*, 23, 20–31.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, 57(1), 289–300.

Bolck, A., Croon, M., & Hagenaars, J. (2004). Estimating latent structure models with categorical variables: One-step versus three-step estimators. *Political Analysis*, 12(1), 3–27.

Saqr, M., Vogelsmeier, L. V. D. E., & López-Pernas, S. (2024). Capturing where the learning process takes place: A person-specific and person-centered primer. *Learning and Individual Differences*, 113, 102492. https://doi.org/10.1016/j.lindif.2024.102492

Scrucca, L., Saqr, M., López-Pernas, S., & Murphy, K. (2024). An introduction and R tutorial to model-based clustering in education via latent profile analysis. In M. Saqr & S. López-Pernas (Eds.), *Learning analytics methods and tutorials: A practical guide using R* (pp. 285–317). Springer. https://doi.org/10.1007/978-3-031-54464-4_9

Vermunt, J. K. (2010). Latent class modeling with covariates: Two improved three-step approaches. *Political Analysis*, 18, 450–469.