Tree plots of how classes split as their number increases with the latents R package

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

26 November 2026

Contents

Data and candidates 2
The tree at one number of group classes 3
Every tree 4
Comparing solutions at one level 7
Comparing two solutions 9
The tree of group classes 10
Summary of calls 11
References 11

Class enumeration estimates one mixture model for each number of classes. Each model is fitted from its own starting values, and the numbering of its classes is arbitrary, so class 2 of a three-class solution has no defined relation to class 2 of a four-class solution. The relation between two solutions is carried by the observations they classify. When the classes of the larger solution subdivide those of the smaller one, each class of the smaller solution passes its members to one or more classes of the larger, and the solutions are nested. When the larger solution redistributes observations across class boundaries, the classes of the two solutions overlap without nesting. A clustering tree represents these relations for a sequence of solutions (Zappia & Oshlack, 2018). Each row of the tree is one solution, divided into its classes in proportion to their size, and the band between two classes in adjacent rows is proportional to the observations they share.

In a mixture model each observation belongs to every class with a posterior probability. For two solutions A and B fitted to the same N observations, with posterior probabilities p_{ia} and q_{ib} and observation weights w_i, the shared posterior mass of class a of A and class b of B is

$$m_{ab} = \frac{\sum_{i=1}^{N} w_i \, p_{ia} \, q_{ib}}{\sum_{i=1}^{N} w_i}.$$

The masses form a joint distribution over the pairs of classes. Summed over the classes of B, they give the posterior share of class a, and summed over the classes of A, the posterior share of class b. The ratio of m_{ab} to the share of a is the part of class a that the larger solution places in class b. A tree built from posterior probabilities retains the uncertainty of the classification, which a tree built from modal assignments discards.

A two-level mixture model has two sequences of solutions (Vermunt, 2003). Observations are classified into classes, and the groups that contain them are classified into group classes by their distribution over those classes. The classes form a tree at each number of group classes, and the group classes form a tree at each number of classes. Solutions can also be compared at a single level, for example the four-class solutions estimated with different numbers of group classes, whose classes need not coincide.

Data and candidates

`student_esm` holds experience-sampling data from 100 university students (Neubauer & Schmiedek, 2024). At each prompt when they had not studied since the previous one, the students reported which of eight leisure activities they had done. The prompts are the observations and the students the groups. The candidates combine one to four classes of prompts with one to three group classes of students. Each candidate is fitted from twenty random starts because a latent class model of eight binary items has many local maxima, and with a tight convergence tolerance.

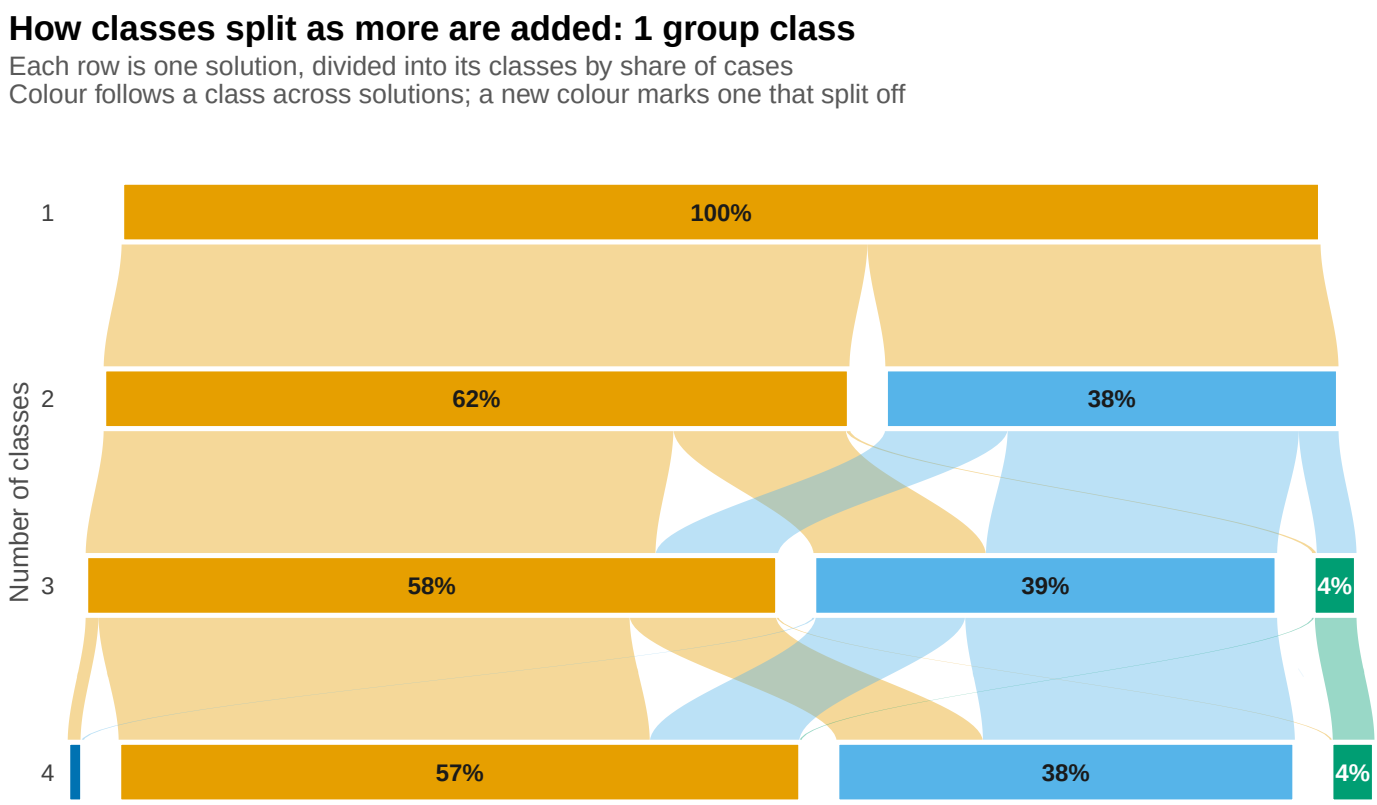
```
library(latents)
activities <- c("time_with_friends", "on_social_media", "tv_video_games",
                "listened_music", "sports", "walking", "reading",
                "part_time_job")
models <- enumerate_classes(student_esm, activities, "student",
                            n_profiles = 1:4, n_group_classes = 1:3,
                            categorical = activities, n_starts = 20,
                            tol = 1e-10, seed = 1)
```

The grid holds 12 candidates. Their classes of prompts are counted by `n_profiles`, the name the package uses at both levels of a two-level model. The 2 candidates with one class and several group classes could not be estimated. A group class is defined by its distribution over the classes, and with a single class every group has the same distribution. Every tree with two or more group classes therefore begins at two classes. The best solution of each candidate with one group class recurs in at least 3 of its starts (`n_best_replicated`), but 2 candidates with group classes reach theirs from a single start. The trees that involve group classes show how the plot reads, and their solutions would need more starts before they are interpreted.

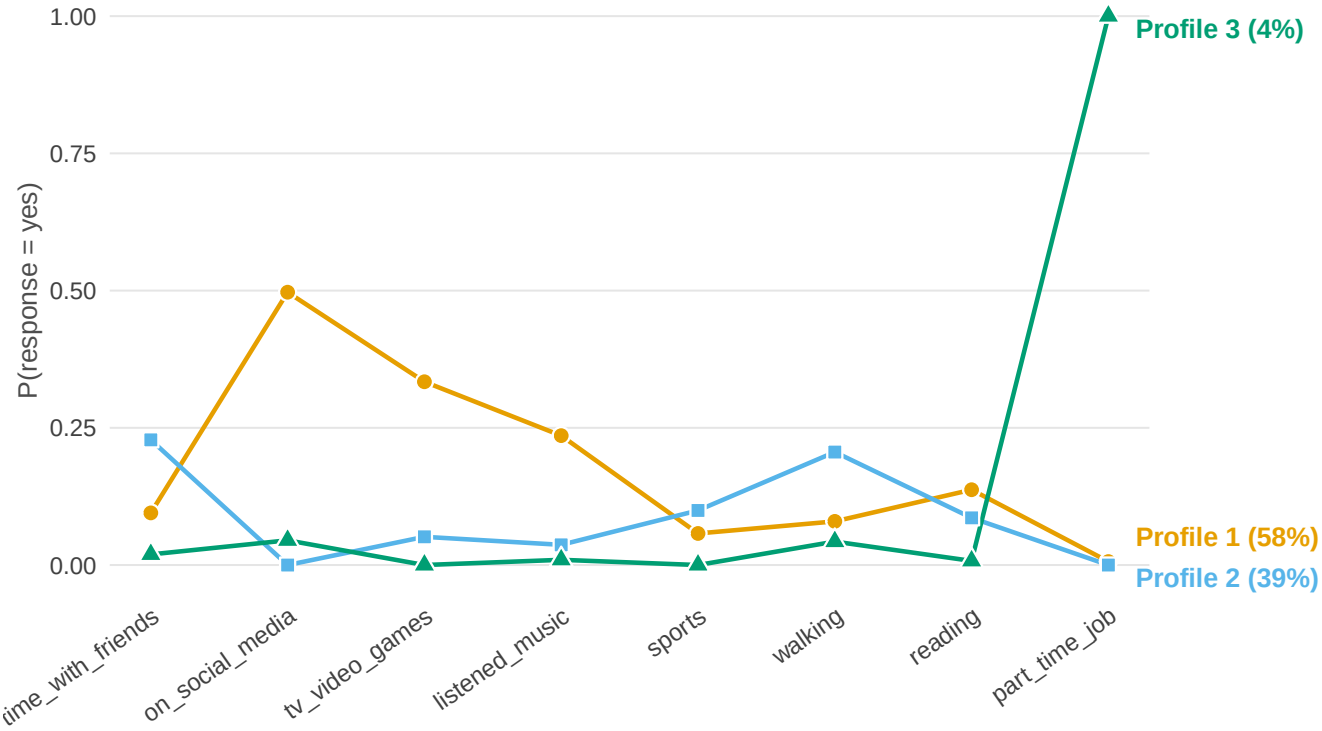
The tree at one number of group classes

At one group class the model is a single-level latent class model of the prompts. Its tree shows how the classes of prompts divide as classes are added.

```
plot_tree(models, n_group_classes = 1)
```




```
plot_responses(get_candidate(models, n_profiles = 3, n_group_classes = 1),
               intervals = FALSE)
```



The two-class solution separates prompts with media use (social media, television or games, music) from prompts with offline activities. In the three-class solution the media class keeps 76% of its posterior mass. The offline class divides into three parts. The largest, 64%, forms a class of time with friends and walking, 27% joins the media class, and the remainder forms a class of 4% of prompts in which the probability of a part-time job is 1.00. The part-time job class passes 96% of its mass to a single class of the four-class solution, so it persists as further classes are added.

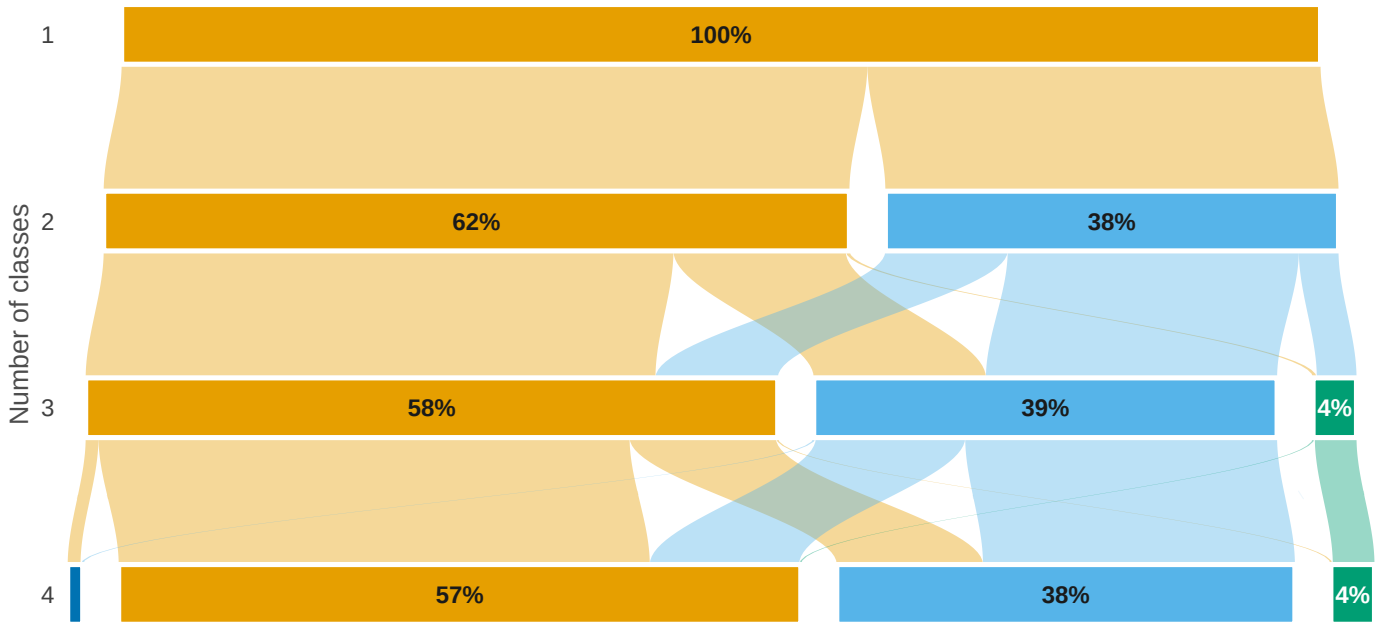
Every tree

Without a choice of group classes, each number of group classes has its own tree, returned as a separate plot. On an interactive screen device the plots are shown one at a time.

```
plot_tree(models)
```

How classes split as more are added: 1 group class

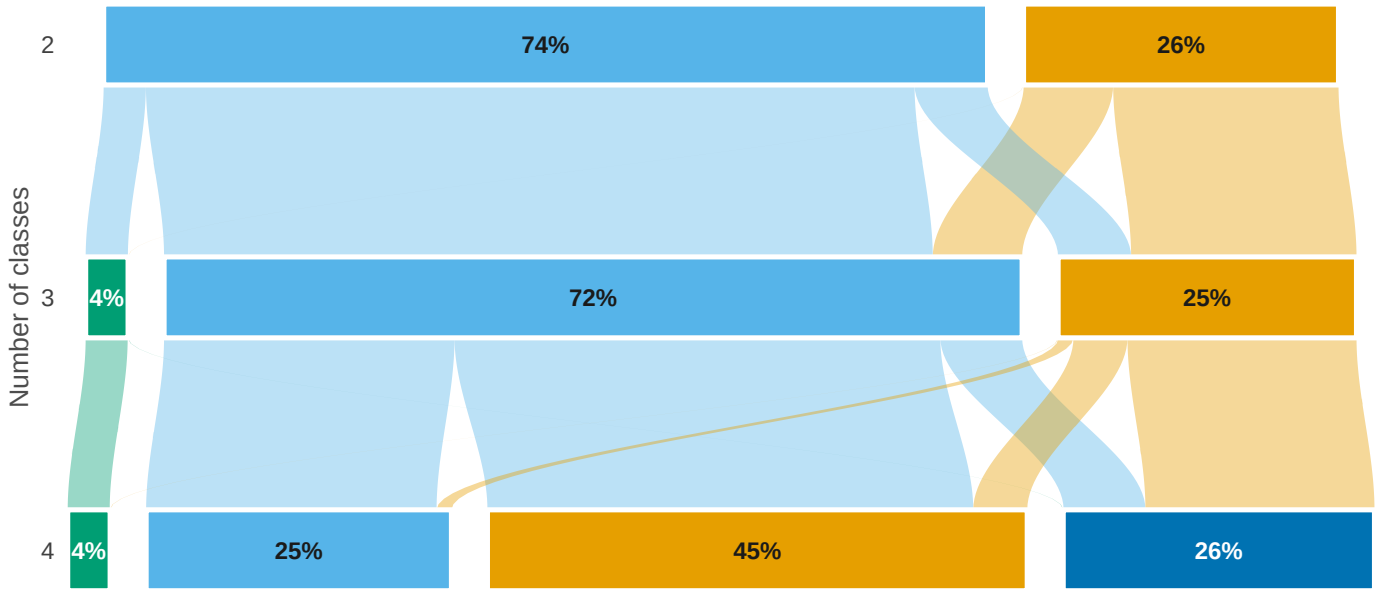
Each row is one solution, divided into its classes by share of cases
A colour marks the same class in every tree; a new colour marks one that split off



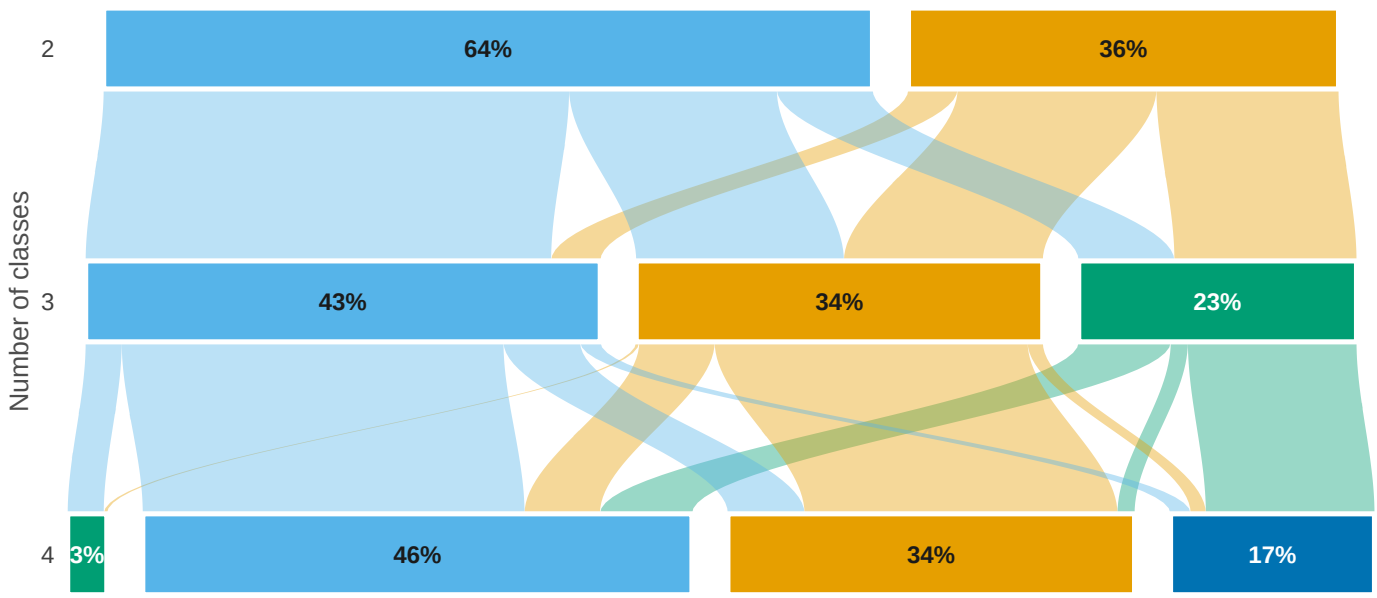
A band that forks: a class split in two (nested, stable solutions).
Bands that merge or cross: units shared between classes, by reassignment or by uncertain classification.

How classes split as more are added: 2 group classes

Each row is one solution, divided into its classes by share of cases
A colour marks the same class in every tree; a new colour marks one that split off



A band that forks: a class split in two (nested, stable solutions).
Bands that merge or cross: units shared between classes, by reassignment or by uncertain classification.

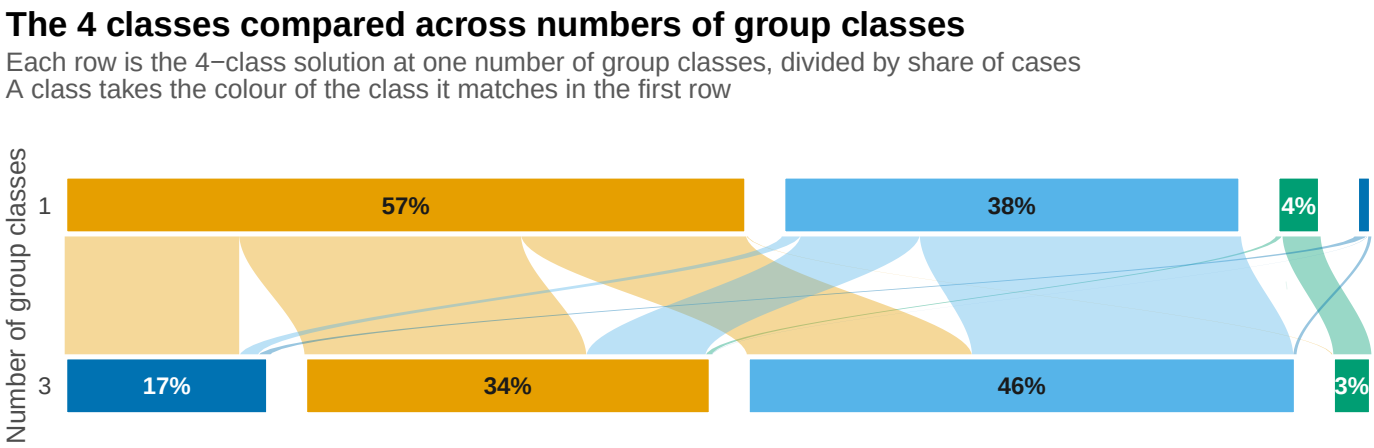


A later tree matches its classes to those of the first tree using a one-to-one assignment of posterior overlap at the same number of classes. The assignment maximizes total overlap for square matches up to eight classes; larger or rectangular matches use a greedy assignment. A class can therefore be matched to something other than its own largest-overlap class. Colours mark these assigned correspondences. `combine = TRUE` plots every tree as a panel of one figure.

Comparing solutions at one level

A single value of `n_profiles` selects one level of every tree. The rows of the plot are then the solutions with that number of classes, one for each number of group classes, and the bands show where the same prompts are classified in each.

```
plot_tree(models, n_profiles = 4, n_group_classes = c(1, 3))
```



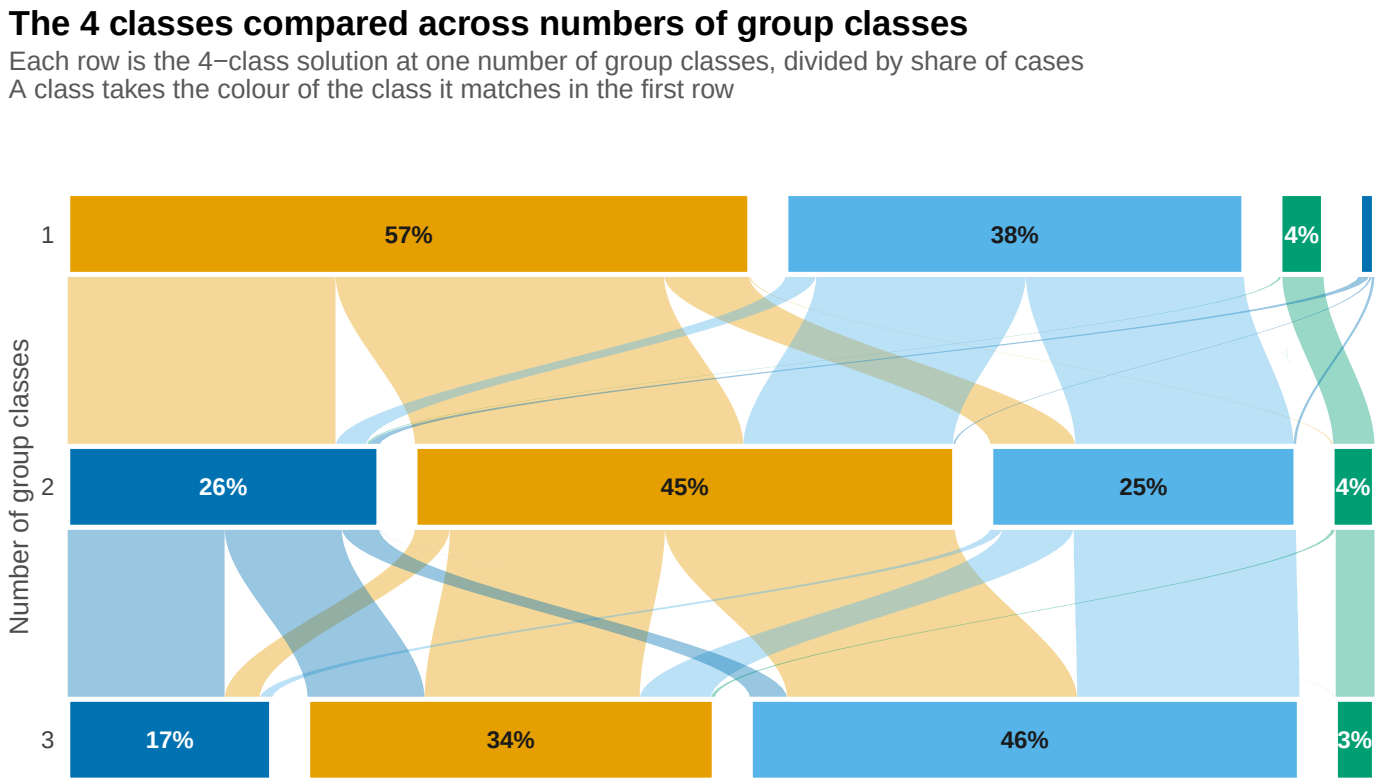
The values behind the plot are returned by `get_results()` with the same arguments, one row per pair of classes.

```
get_results(models, "tree", n_profiles = 4, n_group_classes = c(1, 3))
#>      tree     level  from_candidate from_class from_share    to_candidate to_class to_share
#> 1  classes4 individuals classes4_groups1       1    0.0364 classes4_groups3      1  0.4581
#> 2  classes4 individuals classes4_groups1       2    0.5690 classes4_groups3      1  0.4581
#> 3  classes4 individuals classes4_groups1       3    0.0125 classes4_groups3      1  0.4581
#> 4  classes4 individuals classes4_groups1       4    0.3821 classes4_groups3      1  0.4581
#> 5  classes4 individuals classes4_groups1       1    0.0364 classes4_groups3      2  0.3392
#> 6  classes4 individuals classes4_groups1       2    0.5690 classes4_groups3      2  0.3392
#> 7  classes4 individuals classes4_groups1       3    0.0125 classes4_groups3      2  0.3392
#> 8  classes4 individuals classes4_groups1       4    0.3821 classes4_groups3      2  0.3392
#> 9  classes4 individuals classes4_groups1       1    0.0364 classes4_groups3      3  0.1701
#> 10 classes4 individuals classes4_groups1       2    0.5690 classes4_groups3      3  0.1701
#> 11 classes4 individuals classes4_groups1       3    0.0125 classes4_groups3      3  0.1701
#> 12 classes4 individuals classes4_groups1       4    0.3821 classes4_groups3      3  0.1701
#> 13 classes4 individuals classes4_groups1       1    0.0364 classes4_groups3      4  0.0326
#> 14 classes4 individuals classes4_groups1       2    0.5690 classes4_groups3      4  0.0326
#> 15 classes4 individuals classes4_groups1       3    0.0125 classes4_groups3      4  0.0326
#> 16 classes4 individuals classes4_groups1       4    0.3821 classes4_groups3      4  0.0326
#>        mass share_of_from share_of_to
#> 1  3.18e-11      8.73e-10    6.94e-11
#> 2  1.88e-01      3.30e-01    4.09e-01
#> 3  2.62e-03      2.09e-01    5.72e-03
#> 4  2.68e-01      7.01e-01    5.85e-01
#> 5  4.66e-03      1.28e-01    1.37e-02
#> 6  2.35e-01      4.13e-01    6.93e-01
#> 7  4.30e-04      3.43e-02    1.27e-03
#> 8  9.91e-02      2.59e-01    2.92e-01
#> 9  1.27e-12      3.49e-11    7.47e-12
#> 10 1.46e-01      2.56e-01    8.56e-01
#> 11 9.48e-03      7.56e-01    5.57e-02
#> 12 1.51e-02      3.95e-02    8.87e-02
#> 13 3.17e-02      8.72e-01    9.73e-01
```

```
#> 14 8.85e-04      1.56e-03    2.71e-02
#> 15 1.26e-23      1.00e-21    3.85e-22
#> 16 2.82e-11      7.39e-11    8.66e-10
```

Three classes of the four-class solution with one group class place between 70% and 87% of their posterior mass in a single class of the solution with three group classes. The largest class, 57% of prompts, divides among 3 classes, none of which receives more than 41% of it. Omitting `n_group_classes` compares the same level across every number of group classes.

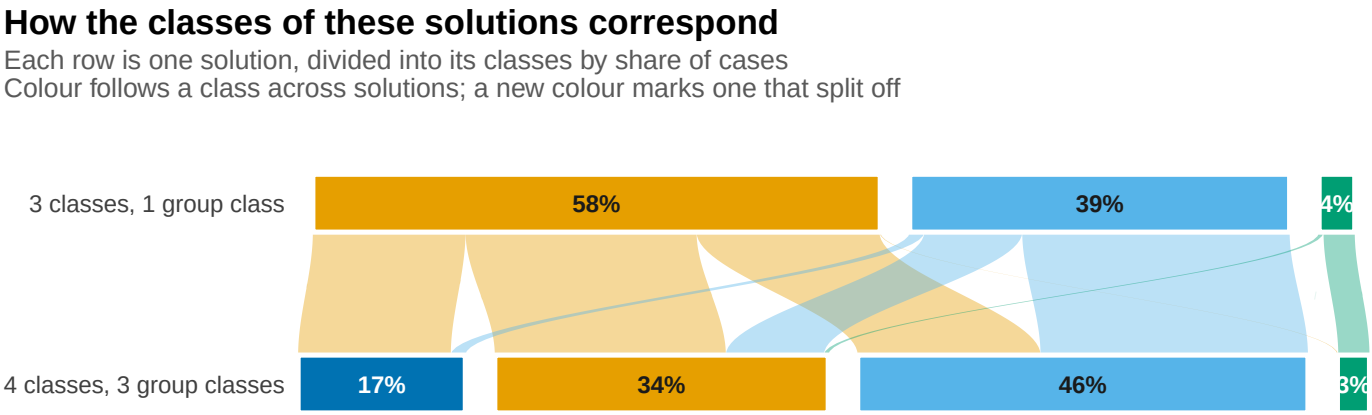
```
plot_tree(models, n_profiles = 4)
```



Comparing two solutions

Any two candidates are compared with `from` and `to`, each given by its number of classes and number of group classes.

```
plot_tree(models, from = c(3, 1), to = c(4, 3))
```

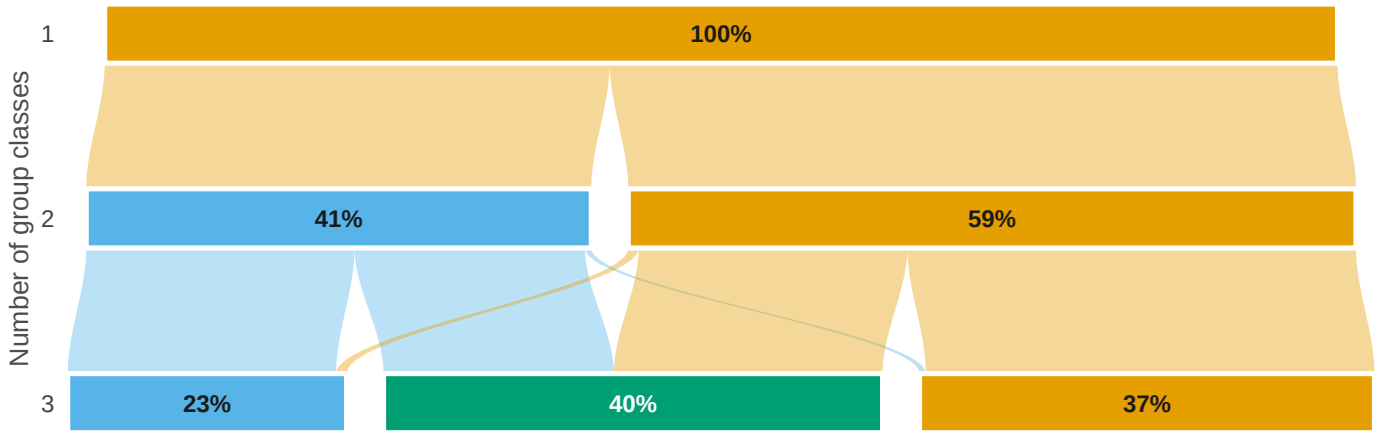


The candidates can also be given by name, in the order of the rows, with `candidates = c("classes3_groups1", "classes4_groups3")`.

The tree of group classes

With `level = "groups"` the units are the students and the classes are the group classes. A tree then runs down the numbers of group classes at one number of classes of prompts.

```
plot_tree(models, level = "groups", n_profiles = 4)
```



From two to three group classes, two group classes of the three-group-class solution each receive at least 96% of their students from a single group class of the two-group-class solution. The third draws its students from both, no more than 54% from either. The tree of group classes is read in the same way as the tree of classes. With `level = "groups"`, a single value of `n_group_classes` compares one level of group classes across numbers of classes, as `n_profiles` does for the classes.

Summary of calls

Question	Call
How do the classes divide as classes are added?	`plot_tree(models, n_group_classes = 1)`
The same for every number of group classes	`plot_tree(models)`, `combine = TRUE` for one figure
How do the solutions with four classes differ across group classes?	`plot_tree(models, n_profiles = 4, n_group_classes = c(1, 3))`
How do two chosen solutions correspond?	`plot_tree(models, from = c(3, 1), to = c(4, 3))`
How do the group classes divide?	`plot_tree(models, level = "groups", n_profiles = 4)`
The shared masses behind any of these	`get_results(models, "tree", ...)` with the same arguments

References

Neubauer, A. B., & Schmiedek, F. (2024). Approaching academic adjustment on multiple time scales. *Zeitschrift für Erziehungswissenschaft, 27*(1), 147–168. https://doi.org/10.1007/s11618-023-01182-8

Vermunt, J. K. (2003). Multilevel latent class models. *Sociological Methodology*, 33, 213–239.

Zappia, L., & Oshlack, A. (2018). Clustering trees: A visualization for evaluating clusterings at multiple resolutions. *GigaScience*, 7(7), giy083.