To Split or to Mix? Tree vs. Mixture Models for Detecting Subgroups

Hannah Frick, *Universität Innsbruck*, Hannah.Frick@uibk.ac.at
Carolin Strobl, *Universität Zürich*, Carolin.Strobl@psychologie.uzh.ch
Achim Zeileis, *Universität Innsbruck*, Achim.Zeileis@uibk.ac.at

Abstract. A basic assumption of many statistical models is that the same set of model parameters holds for the entire sample. However, different parameters may hold in subgroups (or clusters) which may or may not be explained by additional covariates. Finite mixture models are a common technique for detecting such clusters and additional covariates (if available) can be included as concomitant variables. Another approach that relies on covariates for detecting the clusters are model-based trees. These recursively partition the data by splits along the covariates and fit one model for each of the resulting subgroups. Both approaches are presented in a unifying framework and their relative (dis)advantages for (a) detecting the presence of clusters and (b) recovering the grouping structure are assessed in a simulation study, varying both the parameter differences between the clusters and their association with the covariates.

Keywords. finite mixture model, model-based clustering, model-based recursive partitioning

1 Introduction

A basic assumption of many statistical models is that its set of parameters applies to all observations. However, subgroups may exist for which different sets of parameters hold, e.g., the relationship between some response and regressors might be different for younger and older individuals. If the breakpoint which separates "younger" and "older" were known, parameter stability can simply be assessed by checking for parameter differences between these two specific subgroups. However, if the breakpoint is unknown or if there is a smooth transition between "young" and "old", the subgroups can be still be detected in a data-driven way. Either a finite mixture model [5] can be employed, possibly using age as a concomitant variable [2] to model smooth transitions between clusters. Alternatively, model-based recursive partitioning [9] can capture the difference by one or more splits in the partitioning variable age, yielding a tree structure (similar to classification and regression trees, [1]) where each leaf is associated with a parametric model.

Given their shared goal of establishing subgroups to capture parameter instabilities across subgroups, how do these mixture models and model-based trees compare? A unifying framework for both methods is presented and their relative (dis)advantages for (a) detecting the presence of clusters with different parameters and (b) recovering the underlying grouping structure are assessed in a simulation study.

2 Theory

Although both mixture models and model-based trees can be applied to general parametric models estimated by means of the maximum likelihood (ML) principle, we focus on linear regression here because it is the most simple and commonly applied model:

$$y_i = x_i^\top \beta + \varepsilon_i \tag{1}$$

with response y_i, regressor vector x_i, and errors ε_i for observations $i = 1, \ldots, n$. The unknown vector of regression coefficients β can be estimated by least squares, which yields the same estimates as ML estimation under the assumption of independent normal errors with variance σ^2. In the latter case the log-likelihood is given by $\sum_{i=1}^n \log \phi(y_i; x_i^\top \beta, \sigma^2)$ where $\phi(\cdot)$ denotes the probability density function of the normal distribution.

Within this framework, both trees and mixture models can assess whether the same coefficient vector β holds for all n observations and, if parameter stability is violated, simultaneously find clusters/subgroups and estimate the associated cluster-specific coefficients. Further covariates z_i can be used as concomitant or partitioning variables, respectively, to establish these clusters.

Finite mixture models

Finite mixture models assume that the data stem from K different subgroups with unknown subgroup membership and subgroup-specific parameters $\beta_{(k)}$ and $\sigma_{(k)}$ $(k = 1, \ldots, K)$. The full mixture model is a weighted sum over these separate models (or components):

$$f(y_i; x_i, z_i, \beta_{(1)}, \sigma_{(1)}, \ldots, \beta_{(K)}, \sigma_{(K)}) = \sum_{k=1}^{K} \pi_k(z_i) \cdot \phi(y_i; x_i^\top \beta_{(k)}, \sigma^2_{(k)}). \tag{2}$$

The component weights may depend on the additional covariates z_i through a concomitant variable model [2], typically a multinomial logit model

$$\pi_k(z_i) = \frac{\exp(z_i^\top \alpha_{(k)})}{\sum_{g=1}^{K} \exp(z_i^\top \alpha_{(g)})} \tag{3}$$

with component-specific coefficients $\alpha_{(k)}$. For identifiability, one group (typically the first) is used as a reference group and the coefficients of this group are set to zero: $\alpha_{(1)} = 0$. This also includes the special case without concomitant variables, where $z_i = 1$ is just an intercept yielding component-specific weights $\pi_k(z_i) = \pi_{(k)}$.

Given the number of subgroups K, all parameters in the mixture model are typically estimated simultaneously by ML using the expectation-maximization (EM) algorithm. To choose the number of subgroups K, the mixture model is typically fitted for $K = 1, 2, \ldots$ and then the best-fitting model is selected by some information criterion. Here, we employ the Bayesian Information Criterion (BIC).

Model-based recursive partitioning

Model-based recursive partitioning [9] can also detect subgroups for which different model parameters hold. These subgroups are separated by sample splits in the covariates z_i used for partitioning. The algorithm performs the following steps:

1. Estimate the model parameters in the current subgroup.
2. Test parameter stability along each partitioning variable z_{ij}.
3. If any instability is found, split the sample along the variable z_{ij*} with the highest instability. Choose the breakpoint with the highest improvement in model fit.
4. Repeat 2–4 on the resulting subsamples until no further instability is found.

Here, we only briefly outline how these parameter instability tests work and refer to [8] for the theoretical details. The basic idea is that the scores, i.e., the derivative of $\log \phi(\cdot)$ with respect to the parameters, evaluated at the estimated coefficients behave similar to least squares residuals: They sum to zero and if the model fits well they should fluctuate randomly around zero. However, if the parameters change along one of the partitioning variables z_{ij}, there should be systematic departures from zero. Such departures *along* a covariate can be captured by a cumulative sum of the scores (ordered by the covariate) and aggregated to a test statistic, e.g., summing the absolute or squared cumulative deviations. Summing the scores along a categorical partitioning variable then leads to a statistic that has an asymptotic χ^2 distribution while aggregation along numeric partitioning variables can be done in a way that yields a maximally-selected score (or Lagrange multiplier) test. In either case, the p-value p_j can be obtained for each ordering along z_{ij} without having to reestimate the model. The p-values are then Bonferroni-adjusted to account for testing along multiple orderings and partitioning continues until there is no further significant instability (here at the 5% level).

Since each split can be expressed through an indicator function $I(\cdot)$ (for going left or right), each branch of the tree can be represented as a product of such indicator functions. Therefore, the model-based tree induced by recursive partitioning is in fact also a model of type (2), albeit with rather different weights:

$$\pi_k(z_i) = \prod_{j=1}^{J_k} I(s_{(j|k)} \cdot z_{i(j|k)} > b_{(j|k)}) \tag{4}$$

where $z_{(j|k)}$ denotes the j-th partitioning variable for terminal node k, $b_{(j|k)}$ is the associated breakpoint, $s_{(j|k)} \in \{-1, 1\}$ the sign (signaling splitting to the left or right), and J_k the number of splits leading up to node k.

Differences and similarities

While both methods are based on the same linear regression model and aim at detecting subgroups with stable parameters, certain differences arise:

- Because K is fixed for each mixture model, it is based on model selection via an information criterion whereas the selection of K through a tree is based on significance tests.

- Covariates z are optional for mixture models and latent subgroups can be estimated. For a tree, those covariates are required. Furthermore, if no covariates associated with the subgroups are available, the groups cannot be detected.
- The concomitant model (3) assumes a smooth, monotonic transition between subgroups. The sample splits of a tree (4) represent abrupt shifts, multiple splits in a covariate are able to represent a non-monotonic transition. While variable selection is inherent to trees, it requires an additional step for mixtures models.
- Trees yield a hard clustering and mixtures a probabilistic clustering of the observations.

To investigate how the aforementioned differences between the two methods affect their ability to detect parameter instability, a simulation study is conducted, which is described in the next section.

3 Simulation study

To determine how well mixture models and model-based trees detect parameter instability, two basic questions are asked. First, is any instability found at all? Second, if so, are the correct subgroups recovered? These two aspects are potentially influenced by several factors: How does the relationship between the response y and the regressors x differ between the subgroups and how strongly does it differ? If there are any additional covariates z available, how and how strongly are those covariates connected to the subgroups? In general, we expect the following:

- Given the covariates z are associated strongly enough with the subgroups, trees are able to detect smaller differences in $\beta_{(k)}$ than mixtures because they employ a significance test for each parameter rather than an information criterion for full sets of parameters. In contrast, mixtures are more suitable to detect subgroups if they are only loosely associated with the covariates z, as long as the differences in $\beta_{(k)}$ are strong enough.
- If the association between covariates and subgroups is smooth and monotonic, mixtures are more suitable to detect the subgroups whereas trees are more suitable if the association is characterized by abrupt shifts and possibly non-monotonic.
- If several covariates determine the subgroups simultaneously, the mixture is more suitable, whereas trees are more suitable if z includes several noise variables unconnected to the subgroups.

Motivated by these considerations, the simulation design is explained in the next section.

Simulation design

A single regressor x and four additional covariates $z_1, \ldots, z_4$ are drawn from a uniform distribution on $[-1, 1]$. The response y is computed with errors drawn from a standard normal distribution. Two subgroups of equal size are simulated. How they differ is governed by the form of β – either in their *intercept*, *slope*, or *both* – and the magnitude of their differences is governed by the simulation parameter κ (Table 1). How the covariates are connected to the subgroups is governed by the form of $\pi_k(z)$ – either via a logistic or a step function – and the strength

	Label	Details
Coefficients	intercept	$\beta_{(1)} = (\kappa, 0)^\top \quad \beta_{(2)} = (-\kappa, 0)^\top$
	slope	$\beta_{(1)} = (0, \kappa)^\top \quad \beta_{(2)} = (0, -\kappa)^\top$
	both	$\beta_{(1)} = (\kappa, -\kappa)^\top \quad \beta_{(2)} = (-\kappa, \kappa)^\top$
Covariates	axis1	logistic with $\alpha_{(2)} = (\exp(\nu), 0, 0, 0)^\top$
	diagonal	logistic with $\alpha_{(2)} = (\exp(\nu), -\exp(\nu), 0, 0)^\top$
	double step	tree with $\pi_2(z) = I(z_1 > -0.5)I(z_1 < 0.5)$

Table 1. Simulation scenarios for regression coefficients and covariates.

of this association is governed by simulation parameter ν. Here, three scenarios for $\pi_k(z)$ are considered: a smooth logistic transition along z_1, a smooth logistic transition along z_1 and z_2 simultaneously, and a sharp transition along z_1 with two breakpoints (labeled *axis1*, *diagonal*, and *double step*, respectively). The corresponding parameter vector of the logistic function and the breakpoints can be found in Table 1. The simulation parameters cover the following ranges: $\kappa \in \{0, 0.05, \ldots, 1\}$ and $\nu \in \{-1, -0.5, \ldots, 2\}$. Note that $\beta_{(1)}$ and $\beta_{(2)}$ are identical if $\kappa = 0$ and thus only one subgroup is simulated. Each coefficient scenario is combined with every covariate scenario and the sample size $n \in \{200, 500, 1000\}$ is varied. The covariates z_3 and z_4 are always noise variables and thus either included or excluded in z. For each of these conditions, 500 datasets are drawn and three methods applied: a model-based tree, a plain mixture, and a mixture with concomitant variables. Both mixtures are fitted with $K = \{1, \ldots, 4\}$ and $\hat{K}$ selected via BIC. For all computations, the R system for statistical computing [7] is used along with the add-on packages **partykit** [4] and **flexmix** [3].

Outcome assessment

To address the first question of whether or not any instability is found, the *hit rate* is computed: This is the rate of selecting more than one subgroup – this corresponds to splitting at least once for a tree and to selecting $\hat{K} > 1$ for a mixture. To address the second question if the right subgroups are found, the estimated clustering is compared to the true clustering. Many external cluster indices such as the Rand index favor a "perfect match", i.e., splitting one true subgroup into several estimated subgroups (as might be unavoidable in a tree) is penalized by the index. Cramér's coefficient is invariant against such departures from a perfect match [6] and thus employed here.

Simulation results

Exemplary results are shown for the scenario with differences in *both* coefficients with $n = 200$ observations, without the noise variables z_3 and z_4, and the *double step* scenario as well as the logistic scenarios *axis1* and *diagonal* with three levels $\nu = \{-1, 0, 1\}$ of separation between subgroups. For the *double step* scenario, the hit rate for detecting instability is depicted in the left panel of Figure 1. The tree clearly outperforms both mixtures. For the logistic scenarios *axis1* and *diagonal*, the hit rates are depicted in Figure 2. If the covariates are only weakly associated with the subgroups ($\nu = -1$, left column), the tree is unable to detect the subgroups,

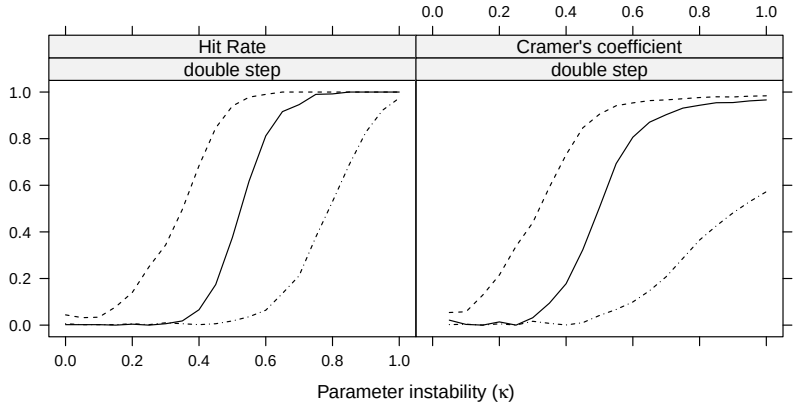

Figure 1. Hit Rate (left panel) and Cramér's coefficient (right panel) for the *double step* covariate scenario. Line type: dashed for tree, solid for mixture, dot-dashed for plain mixture.

regardless of how strongly they differ in their regression coefficients. Both mixtures are able to detect instability beyond a threshold of $\kappa = 0.7$ and reach hit rates of almost 1. For a medium association ($\nu = 0$), the tree is able to detect smaller differences in the regression coefficients than the mixtures but for larger differences both mixtures equally outperform the tree. If the association is strong ($\nu = 1$), the tree outperforms both mixtures. The mixture with concomitants in turn outperforms the plain mixture which is per definition invariant to (changes in) the association between covariates and subgroups. Interestingly, the tree performs rather similarly for the scenarios *axis1* and *diagonal*, indicating that approximation through sequential splits works rather well. For $\kappa = 0$ only one true subgroup exists and mixtures nearly always select $\hat{K} = 1$ while trees incorrectly select $\hat{K} > 1$ subgroups only in less than 5% of the cases (which is the significance level employed in the parameter stability tests).

The corresponding recovery of the subgroups as measured by Cramér's coefficient is depicted in the right panel of Figure 1 for the *double step* scenario. Similar to the detection of instability, the tree outperforms both mixtures. For the logistic scenarios, the Cramér's coefficients are shown in Figure 3. For low and medium levels of association between covariates and subgroups, both mixtures outperform the tree for stronger instabilities. For a medium level of association ($\nu = 0$) and small instabilities, the tree's advantage in detecting instabilities translates into an advantage of also uncovering the correct subgroups. However for a stronger association ($\nu = 1$), the mixture with concomitants recovers the true subgroups better than the other two methods once the hit rates are similar across methods. Despite its good hit rates, the tree never exceeds a Cramér's coefficient of about 0.6. This is the case regardless of how strong the regression coefficients differ, indicating that the tree's ability to uncover the correct subgroups is limited by the (relative) weakness of association between covariates and subgroups. For an even stronger association ($\nu > 1$, not depicted here), the tree recovers the subgroups as well as the concomitant mixture in the *axis1* scenario but fails to do so in the *diagonal* scenario.

For larger numbers of observations ($n = 500$ or 1000) and the other two coefficients scenarios (*intercept* and *slope*), results are similar to those shown here, just being generally more pronounced. Including two additional noise variables z_3 and z_4 affected both the tree and the concomitant mixture, with hit rates dropping slightly stronger for the mixture.

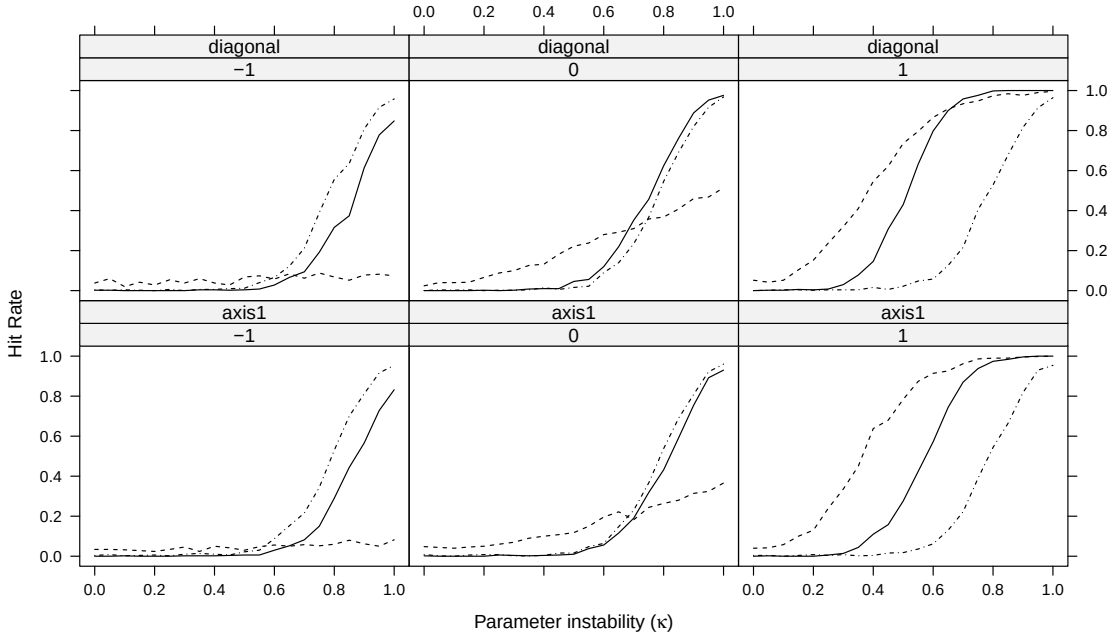

Figure 2. Hit Rate for the logistic covariate scenarios for three levels of $\nu \in \{-1, 0, 1\}$. Line type: dashed for tree, solid for mixture, dot-dashed for plain mixture.

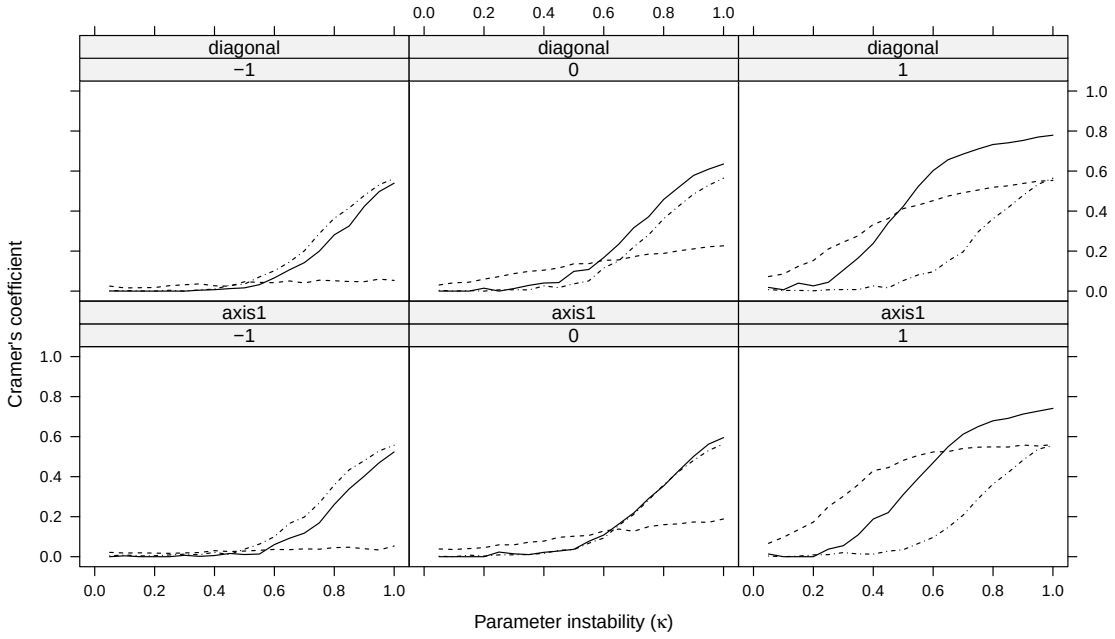

Figure 3. Cramér's coefficient for the logistic covariate scenarios for three levels of $\nu \in \{-1, 0, 1\}$. Line type: dashed for tree, solid for mixture, dot-dashed for plain mixture.

4 Discussion

Both methods are suitable to detect parameter instability (or lack thereof) and recover the subgroups (if any). Which method is more suitable depends largely on the association between the subgroups and covariates as well as how strongly the subgroups differ in their respective parameter vectors. If the association between subgroups and covariates is strong, the tree is able to detect smaller differences in the parameters than the mixtures. The approximation of a smooth transition between classes through sample splits works rather well. In addition, the tree can represent a non-monotonic association which the mixtures cannot. If the association between subgroups and covariates is weak but the difference in the parameters reasonably strong, mixture models are more suitable than the tree. Mixture models are also capable of detecting latent subgroups without any association to covariates. It would be interesting to investigate whether these relationships also apply to situations with more subgroups and mixtures with higher numbers of components. Further questions for further research include the assessment of subgroup recovery on a test set rather than in-sample and variable selection for the concomitant models of mixtures, which could also be accomplished with an information criterion.

In summary, both methods have their relative advantages and thus are more suitable to detect parameter instability and uncover subgroups in different situations. As the exact structure is unknown in practice, we suggest using both methods to gain better insight into the data.

Bibliography

[1] Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. (1984) *Classification and Regression Trees.* Wadsworth, California.

[2] Dayton, C.M. and Macready, G. (1988) *Concomitant-variable latent-class models.* Journal of the American Statistical Association, **83**(401):173–178.

[3] Grün, B. and Leisch, F. (2008) *FlexMix version 2: Finite mixtures with concomitant variables and varying and constant parameters.* Journal of Statistical Software, **28**(4):1–35.

[4] Hothorn, T. and Zeileis, A. (2014) *partykit: A modular toolkit for recursive partytioning in R.* Working Paper 2014-10, Research Platform eeecon, Universität Innsbruck.

[5] McLachlan, G. and Peel. D. (2000) *Finite Mixture Models.* John Wiley & Sons, New York.

[6] Mirkin, B. (2001) *Eleven ways to look at chi-squared coefficients for contingency tables.* The American Statistician, **55**(2):111–120.

[7] R Core Team (2013) *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.

[8] Zeileis, A. and Hornik, K. (2007) *Generalized M-fluctuation tests for parameter instability.* Statistica Neerlandica, **61**(4):488–508.

[9] Zeileis, A., Hothorn, T. and Hornik, K. (2008) *Model-based recursive partitioning.* Journal of Computational and Graphical Statistics, **17**(2):492–514.