Multinomial Process Trees: An Evaluation of Strategies for Model Optimization

Julia Abels [1]

Examiner: Dr. Marco RAGNI
Adviser: Nicolas RIESTERER

January 9, 2019

Abstract

The paper provides insight about the optimization of the structure of cognitive Multinomial Process Tree (MPT) models. These cognitive models are very important tools of evaluation in Cognitive Science and Psychology. The purpose of these models is to formalize assumptions about mental processing structures. But which components represent an effective and good predictive performance? There exist no theories to find the optimal structure of a MPT that optimizes performance. This article investigates three perspectives on how parameters and outcome categories of MPTs influence performance of the models. All results are empirically demonstrated.

1. Introduction

Multinomial process trees are important tools for stochastic analysis and research of categorical data in cognitive science. They measure different cognitive processes which arise in mental representation research and test whether psychological assumptions are true or not. The models can be used in many research fields of cognitive psychology, like presented in Hilbig et al. (2009). An example domain for MPTs is memory recognition. Here, a participant learns a number of vocabularies. Later, a mix of learned and unknown vocabularies is presented and the participant is asked to recognize which words were previously shown. In other words, the participant is asked to perform a classification into new and old items. In this experiment, the participant answers with yes for "I learned the vocabulary" and no for the contrary. This experiment allows for an investigation of how memory recollection is handled by human mind. Researchers try to understand how the mind handles and studies new information. Based on knowledge of the structure of mental processes, researchers are able to create i.e. "power law of practice and forgetting" (Ebbinghaus, Hermann).

An interesting research challenge for computer science is to optimize the structure of MPTs. To this end, optimization is understood as optimizing the maximum likelihood estimator (MLE, Heck et al., 2018). The higher the MLE value, the better the fit of the model to given experimental data.

Therefore, the purpose of this work is to research which structures of an MPT improve the MLE. So far, there are no theories for MPTs which summarize the influence of a modification of the structure on performance. However, since the model structure and the parameter assignment can be manipulated, it is interesting to investigate which changes have an influence on the estimator. Based on this knowledge, theories about model compositions that improve performance can be developed. This will allow assumptions to be made about good and bad tree structures in the future.

The composition of this article starts with related work. In this section, a short overview about the standard trees and the most popular MPT models is given. The next topic addresses the practical foundation of optimizing MPTs. To support these optimization strategies, the following section presents empirical results of model fits. The last section presents an interpretation of the empirical results, the problems that arise and consequential conclusion of the project.

2. Related Work

An MPT is a binary tree and consists of categories and parameters. More precisely, the tree has two possibilities at every node, which is represented by a parameter, and ends in a category. The different categories $\mathcal{C} = \{C_1, C_2, ..., C_n\}$

[1] Department of Cognitive Computation, University of Freiburg, Freiburg, Germany. Correspondence to: Julia Abels <julia.abels@neptun.uni-freiburg.de>.

This work was done as a bachelorproject.

illustrate all outcomes that can occur in an experiment. The parameters $\theta = \{\theta_1, \theta_2, ..., \theta_n\}$ represent decision possibilities. For example in memory research, if a participant can not remember a word of the presented set of terms in a recognition task and tries to guess if the word was learned, then the parameter at this pattern expresses the estimator for guessing the correct way and against probability for guessing the wrong way.
For calculating the MLE for a model, the model needs an estimate for each parameter, which indicates the probability that this branch of the tree for the path will arrive. Moreover, all category probabilities depend on the parameters that point to them. To calculate a path for a category, all parameters on which the category depends are required. Thus a path for one category $c_l \in C_k$ is given by

$$c_l = \prod_{i=1}^{n} \theta_i (1 - \theta_i) \tag{1}$$

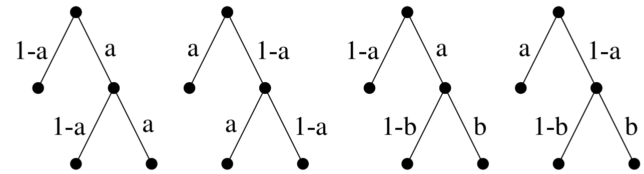
The parameter θ_i represents the probability of success of the process i on this path. Since it is possible to have same category at the end of more than one path, the paths have to be summed up, to compute the overall probability of ending up in category $C_k \in \mathcal{C}$:

$$C_k = \sum_{l=1}^{L} c_l = \sum_{l=1}^{L} \prod_{i=1}^{n} \theta_i (1 - \theta_i) \tag{2}$$

for the path $l = 1, 2, 3, ..., L$.
In addition, the estimator of all categories must be equivalent to one:

$$\sum_{n=1}^{N} C_n = 1 \tag{3}$$

In other words, multinomial process trees represent multinomial distributions.
Famous models are for example the one-high threshold model (1HTM, Hilbig et al., 2009), the two-high threshold model (2HTM, Hilbig et al., 2009) and the two-high threshold model of monitoring (2HTSM, Heck et al., 2018). The 1HTM consists of two parameter and two categories. Thereby the parameters stand for knowing the true answer and guessing the answer randomly. Both categories are vocabulary of old or new item. The 2HTM has, for both cases (new and old items), a certainty parameter and one guessing parameter. Hence, this model consists of three parameters and two categories in total. The 2HTSM considers an expansion of the yes-no recognition experiment. Thereby participants will learn vocabulary from two different sources and later they are asked to assign the items to their original category of source A, source B or New. This makes the model quite complex due to higher number of categories and parameters per source. For more information about MPTs, I recommend the Hilbig et al. (2009) paper.



(a) Two possibilities with one parameter for three categories

(b) Two possibilities with two parameter for three categories

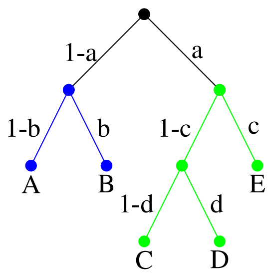
Figure 1. First sketches of models with different parameters for three categories.

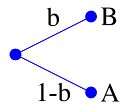
3. Optimizations

To find out how the tree structure can influence MLE performance, start with drawing trees with different number of categories. In order not to make the trees too complex, I started by drawing different models for two to five categories. Thereby I depicted for each category class different quantities of parameters and then different structures. As a result, there are as many possible models for each case of quantity as parameters for one category class. To better understand the procedure, in Figure 1 is an example with three categories.
Figure 1 illustrates all possibilities for three categories with the corresponding parameter number. The quality of the tree structure is examined by how the performance of MLE changes when the number of parameters and then the partial structures of the model is changed. Figure 1a shows two possibilities of a model with three categories and one parameter. The difference between the two models in Figure 1a is that the lower subtree has been swapped, which changes the structure. For the first tree, another subtree is attached to the parameter a and for the second tree to the other parameter $1-a$. So you have two possibilities to represent a model for one parameter and three categories. Figure 1b present two possibilities with two parameters for three categories. This example shows that new trees are created by increasing the number of parameters and exchanging individual subtrees.
In order to establish theories about performance of MLE, the different models of a category number class are examined under the following aspects: The first aspect is how the MLE changes if the number of parameters increases or decreases. Secondly, I analyse how the estimator is affected if the structure is modified. At last, I analyse if the estimator changes if the categories at the end of a path are swapped.

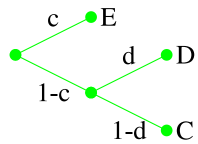
3.1. Creation of Theories

I started with solid number of categories to postulate a theory about the influence of parameter numbers or modifications of the tree of structure. To study how the structure of MPTs can be composed, it is helpful to create a non-trivial "toy-example" like in Figure 2. Figure 2 depicts a possible





(a) First Subtree



(b) Second Subtree

Figure 2. Tree with five categories and four parameters. Split this tree in two subtrees Figure 2a and Figure 2b. On basis of splitting, research how the MLE is modified. Futhermore, the structure of Tree Figure 2b is comparable with the 1HTM (Hilbig et al., 2009). The representation of the tree in the BMPT grammar looks as follows: $abABcdCDE$

tree with five categories. This tree can now be used to investigate of which partial models the tree is made up of and, if the model is split in possible subtrees, how many subtrees exists. The resulting subtrees can then be compared with other existing models. If you look at Figure 2 there exist equivalences of the structures between Figure 2b and the 1HMT. But the 1HTM has only two categories against model of Figure 2b with three categories. With the MLE, researchers want to compare the performance of models with the same number of categories. Thus a direct comparison is not interesting of these two models. For this reason, we first check and observe how small components perform individual.

I simplified the procedure of creating different trees by relying on the context-free language of BMPT (Purdy & Batchelder, 2009). Through the recursive construction of MPTs, it is possible to describe a tree with a formal language. With the context-free language it is easier to depict all different tree structures by iterating through all possibilities.

To take for example the set of categories $C = \{1, 2, 3\}$ and the set of parameters $\Theta = \{a, b, c\}$. Define as grammar

$$G_{BMPT} = (\{x\}, T, R, x) \tag{4}$$

where $\{x\}$ is a variable, $T = C \cup \Theta$ is the set of terminals, R is a set of production rules with:

$$x \rightarrow c \qquad \text{with } c \in C \tag{5}$$

$$x \rightarrow \theta x x \qquad \text{with } \theta \in \Theta \tag{6}$$

and x is the start symbol. For example, create a tree with three categories and two parameters on basis of G_{BMPT} proceed as follows:

$$x \rightarrow axx \rightarrow a1x \rightarrow a1bxx \rightarrow a1b2x \rightarrow a1b23 \tag{7}$$

Now use the grammar G_{BMPT} to create trees with different numbers of parameters and categories. Thereby iterate through all constellations from one to three parameters or more. Some of the resulting constellations are:

$$a1a23 \tag{8}$$

$$a1a32 \tag{9}$$

$$a3a12 \tag{10}$$

$$a1b23 \tag{11}$$

$$ac12b23 \tag{12}$$

3.1.1. Increase and Decrease Amount of Parameters

The first aspect to be investigated is how the performance of MLE behaves when the number of parameters changes. Thereby, the structure of the tree is not modified. The cases are considered how the estimator behaves when the number of parameters is lower and higher than the number of categories. Under the aspect that with more parameters a more exact fit to the data can be achieved, the performance of MLE should be better with more parameters. Above all, it is very interesting to observe how the estimator performs if the model has exactly one parameter less than the number of categories.

3.1.2. Change Position of Classes

The second case to be investigated is whether the categories at the end of a path have an impact on MLE. Thereby, it will be observed how the performance changes if only the categories at the end of the paths of a model are modified. So how does the performance change if for example the two categories of 1HTM permute, or if anything changes at all. Since many paths end in the same category for models that already exist (like 2HTM), it is interesting to examine the aspect of whether categories at all have an influence on performance.

3.1.3. Modify Structure

Finally, the aspect of how much influence the modification of the tree structure on the performance of MLE has is studied. After observing the parameters and categories, the last case is to consider what happens when subtrees are changed. So it is now examined how the performance changes by the composition of single subtrees. The number of categories and the number of parameters are not changed. Only subtrees are modified, so that the depth of the individual paths changes in the model. Since balanced trees can be calculated better, because of the trees has less depth, the hypothesis that a balanced tree achieve a better performance of MLE than a tree with the depth of parameter number n is established here.

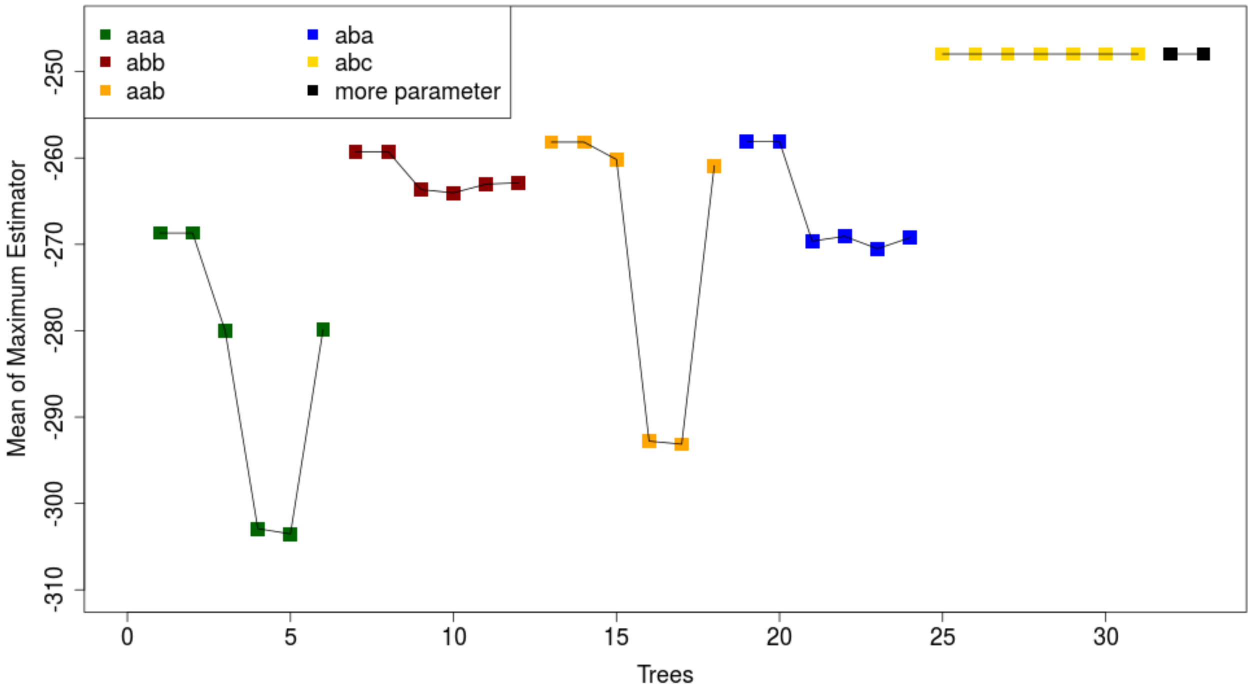


Figure 3. Results of 600 different fits (table has 600 rows) for four categories. The legend shows the parameter combination for four categories, where each parameter stands for one level in the tree. In each parameter combination category the first datapoint is the MLE value of an balanced tree. The second datapoint depicts the mean of MLE value if the categories have been shifted. The rest of the parameter category represents MLE values of different depth possibilities of the trees.

4. Results

The computations of the MLE and plots of solutions are made with the programming language R. With the aid of the programming language Python version 3.6, the trees were created in text files.
First of all, all tree constellations of a category are generated according to the BMPT grammar (Purdy & Batchelder, 2009) in a text file. This means, all possibilities of constellations, number of parameters and permutation of categories are listed in the text file.
For R there exists the module MPTinR (Singmann & Kellen, 2013) to work with MPTs. This module supports the evaluation of MLE for MPTs. So that the module can work with the trees, they have to be converted from the grammar to the `easy` format (see Singmann & Kellen, 2013, section 4). To learn more about the module, it is worth reading the paper by Singmann & Kellen (2013).
Next a table with random values to calculate MLE of the different trees constellations is created. Colums represent numbers of observations for the respective category, rows represent different datasets to use for fitting the data. So that one can make an empirical statement about the estimator, many rows, i.e. datasets, are fitted on a tree simultaneously and the arithmetic mean is taken from the results. The results are represented in plots and explained in following sections.

4.1. Plots

To examine the statements of section three, we need to study how the MLEs behave and change with four and five categories. A tree with three categories does not have so many constellation possibilities. For this reason, the results of three categories are not sufficiently informative, as can be seen in Appendix A (Figure 9). As the examination of two categories with only one parameter does not produce any interesting results, this case does not have to be considered. In Figure 3 and Figure 5, the different colors depict different parameter constellations. For every parameter constellation permute through the parameter to create all possible structures for a tree with BMPT. Thereby create only new constellations so that in the end there are no constellations where only parameters are swapped.
In each parameter combination category, depict the first datapoint the MLE value of an balanced tree. After, the datapoints illustrate the estimator values of trees with

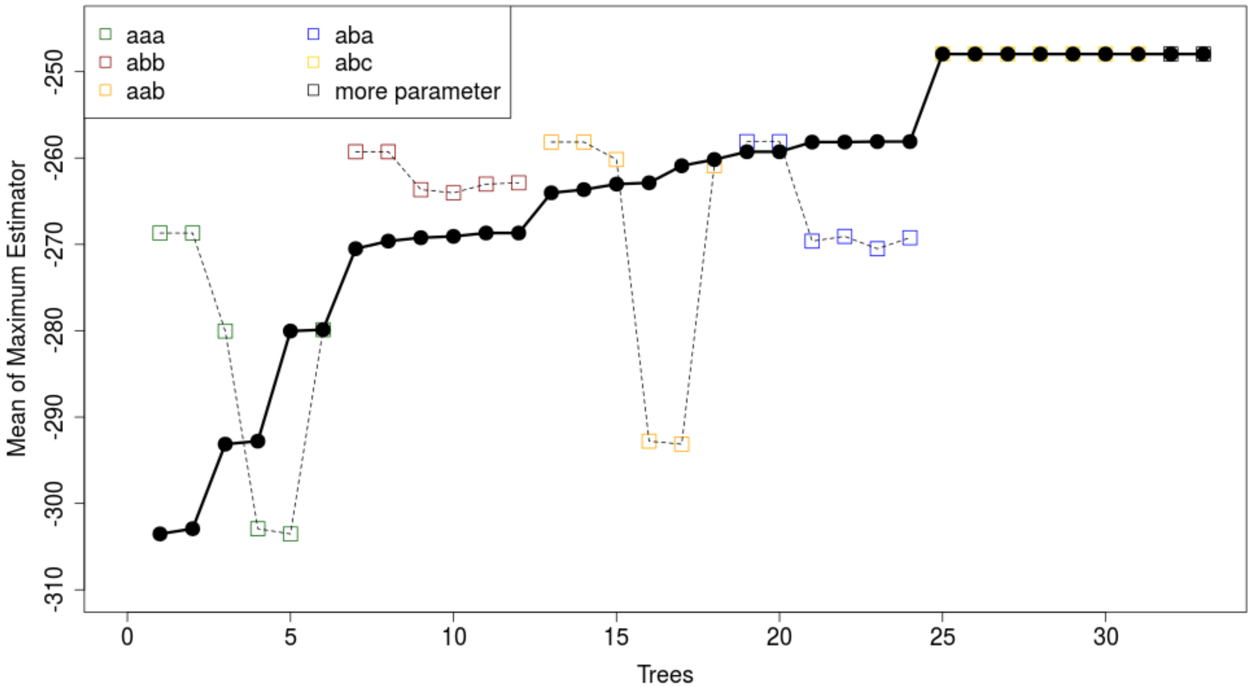


Figure 4. Sorted datapoints of four categories. The black points represent each datapoint of Figure 3 in ascending row sorted.

different depth.

4.1.1. Alter number of parameter

First, we consider the case that the number of parameters is equal or greater than the number of categories. It is noticeable in Figure 3 that above a certain number of parameters the estimator remains unchanged. This can be seen in the graphic by the combination abc and $more\ parameter$. No change of the MLE is discernible here. Even if the structure is enlarged by adding more subtrees without changing the number of categories, the MLE does not change.

On the other hand, if the number of parameter is reduced a change in the estimator can be observed, as the Figure 3 shows. That gives the impression that the performance of MLE depends on the number of parameters.

As can be seen in Figure 5, it is more difficult to recognize a pattern with five categories. However, it can be seen here again that there is no change in MLE values if the number of categories is one greater than number parameters. In addition, you can see that if the number of parameters is smaller than the number of categories, the estimator shows worse results than MLE of number of parameter equal to number of categories.

4.1.2. Swap categories

After decreasing number of parameter, change the categories at the end. In each parameter combination category, the second datapoint illustrates the value of MLE with swapped categories. In other words, the first and the second datapoint in each combination depict an equivalent structure. Only the categories at the end of the trees are swapped. As can be seen in the Figure 3, the value of MLE does not change. The results remain static if only the categories are permuted. Since no influence can be identified, the case does not need to be considered for five categories.

4.1.3. Swap Subtrees

The last aspect to be investigated is the influence of the change in structure. It examines what happens when subtrees are swapped. In other words, how the estimator changes when the depth of the tree is changed. For example, three trees with two different parameters and four categories of the parameter category abb can be represented in the BMPT grammar as follows:

$$ab12b34 \tag{13}$$

$$abb1234 \tag{14}$$

$$a1b2b34 \tag{15}$$

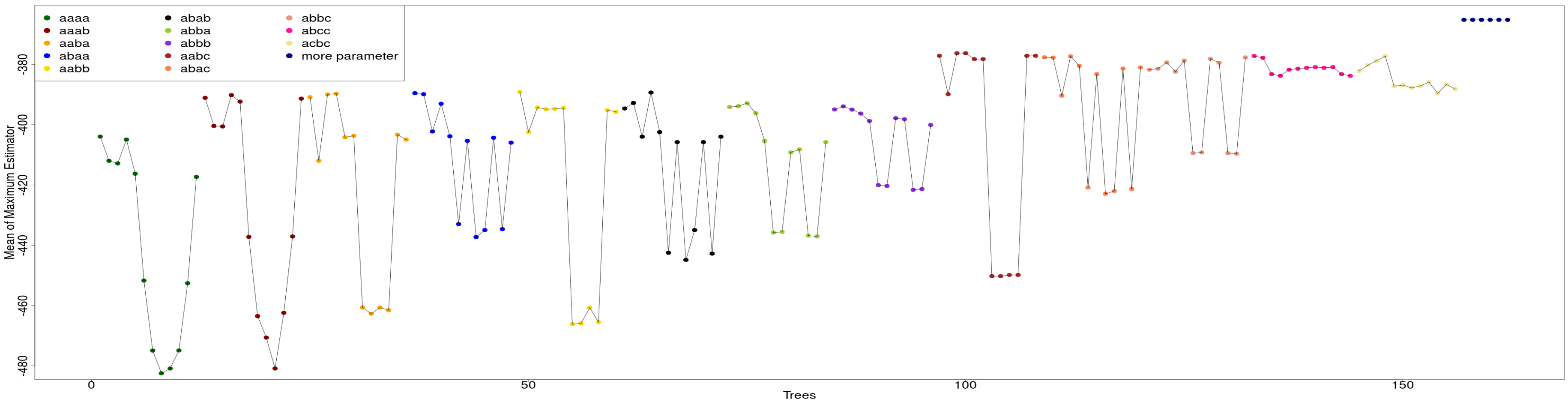


Figure 5. Results of 150 different fits (table has 150 rows) for five categories. The legend shows the parameter all permute combination for five categories. Again each parameter stands for one level in the tree. In each parameter combination category the first datapoint is the MLE value of an balanced tree. Now the change of the categories is not examined again, since this did not result in any influence on four categories. The rest of the parameter category represents MLE values of different depth possibilities of the trees.

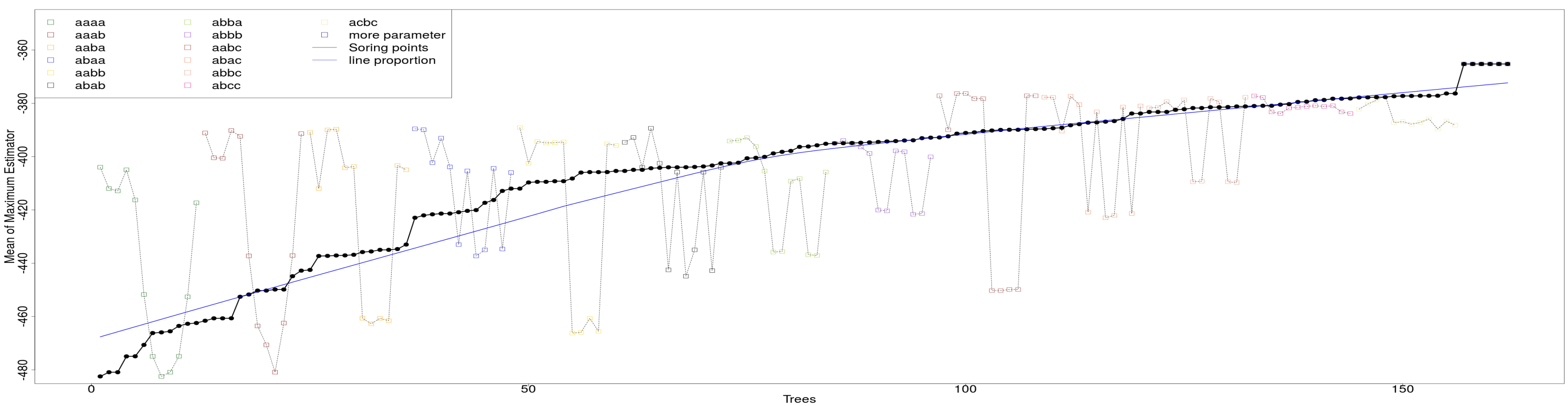


Figure 6. Sorted datapoints of five categories. The black points represent each datapoint of Figure 5 in ascending row sorted.

As can be see in Figure 3 and Figure 5, the different depths of the tree have an influence on the MLE value. This difference can be observed in the performance of the individual parameter categories.
In Figure 3, it can be seen that the first datapoint represents the best MLE value. One could conclude from this that a balanced tree performs better than a tree with a certain depth.
It can also be observed that it seems like the manner of the estimator performance behave equal at some pattern. So if you take a closer look to the parameter combination and how MLE performs if the tree composition has a depth of three, then by some combinations the performance becomes very bad. Figure 3 illustrates this with the parameter combination of aaa and aab. From this one can conclude that for four categories these combinations are not good combinations for a MPT. Because of small changes in the structure, the tree performs worse. The remaining combinations show that changes in the structure do not have a great influence. Thus, it cannot happen with these parameter combinations that the tree performs much worse by small changes.
In Figure 5, it seems more difficult to recognize a pattern. But also the behaviour of the performance seems to repeat itself here. So you can see again that with some combinations the performance varies quite strongly. This reinforces the assumption that some parameter combinations are not good choices for creating a tree.
On the other hand, we can see that not in each parameter combination, the first datapoint has the best results. Therefore, not every parameter combination performs best by a balanced tree. Accordingly, it could be sometimes efficiently to create a tree with a higher depth.
The whole behaviour of the parameter combinations seems to converge to one value here. Thus the jump in performance becomes smaller and smaller with more different parameter combinations. Consequently, a modification in the structure of a tree with more different parameters does not lead to a far from worse performance.

4.2. Modify Results

As can be seen in the results presented in subsection 4.1, the MLE value for four and five categories depends on some combinations, which results in jumpes in performance of MLE. That is why it is interesting to play with the results of the estimator to see how the performance of MLE behave on the whole. In this section, the MLE performance in dependence of the parameter category combination are no longer studied. Instead, it is examined how the individual values perform overall.

4.2.1. Sort Results

Firstly we focus on the behaviour of all MLE values. Accordingly, sort all datapoints and research whether the MLE performance is discretely or continuously distributed. In case of continuity, the points should indicate a functional distribution and, in the other case, clusters would be recognizable in the performance.
In Figure 4, it is recognizable that the datapoints suggest to follow an inverted logarithmic function. In any case, no clusters are recognizable. In addition, it is noticeable that many datapoints are in the interval $[-270, -260]$. So quite a lot of tree compositions perform similarly. However, there are also some tree constellations that perform very bad.
With five categories, as can be seen in Figure 6, it is more obvious that all values suggest to follow a negative logarithm distribution. The blue line in the figure illustrates the proportion of the sorted values. It also indicates that the distribution of the datapoints is approaching the negative logarithm function. By five categories, many tree compositions perform in the range of $[-420, -380]$. Thus, the range of trees which performance similarly has increased. It is also important to mention that in both figures the MLE values converge.

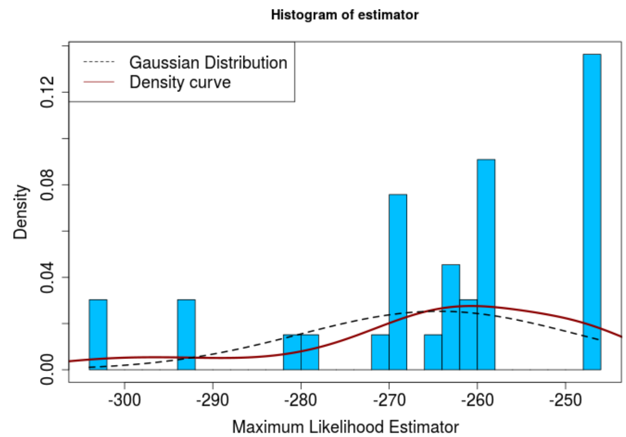


Figure 7. Histogram of four categories. Thereby each balk represents occurrence of datapoints at certain MLE value. To analyse distribution of the occurrence, illustrate gaussian and density curve distribution also.

4.2.2. Histogram of Results

Another possibility to check how the MLE values are distributed is to illustrate the values in a histogram like in Figure 7 and Figure 8. The histogram can be used to determine whether the performance of the trees is following a specific distribution. The output of four and five categories is depicted in Figure 7 and Figure 8. In both figures the Gaussian distribution as well as the density curve is illustrated. Thus, a direct comparison is possible. The functions were created on the basis of the frequency distributions.
In the same way the histogram can be used to recognize which values are represented more frequently in the data

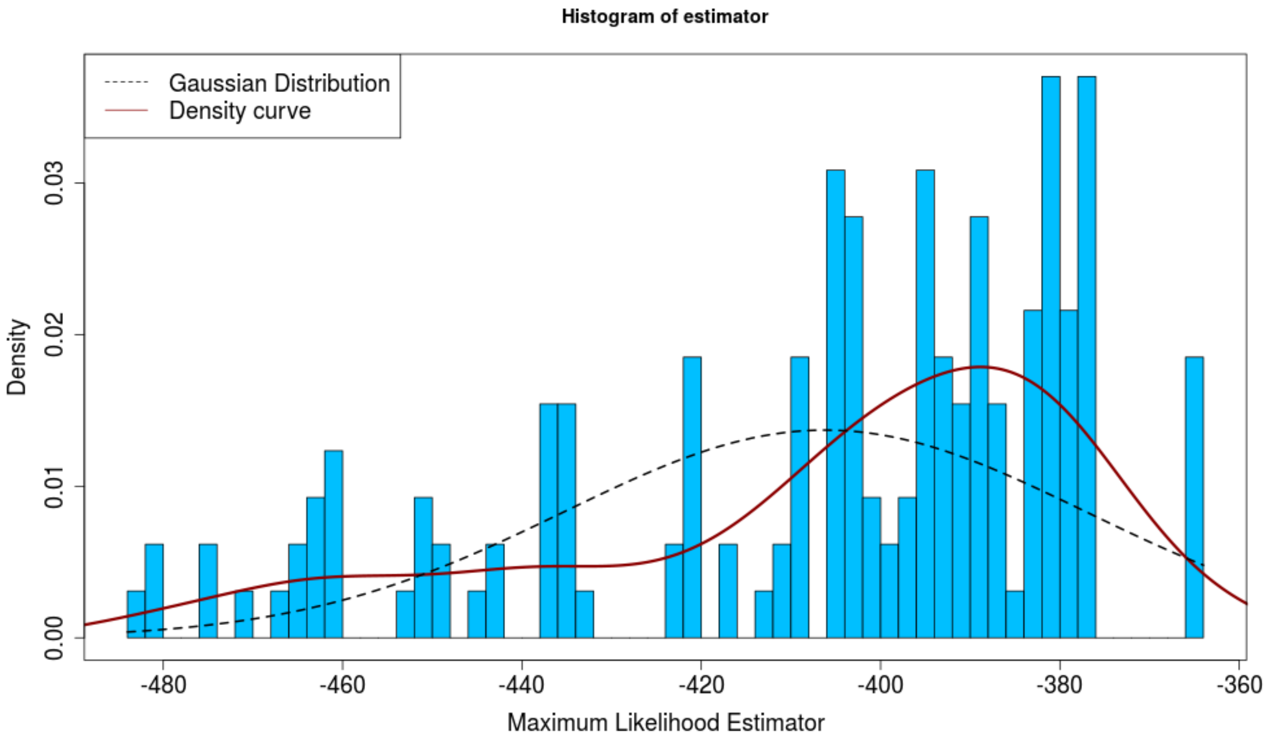


Figure 8. Histogram of five Categories. Thereby each balk represents occurrence of datapoints at certain MLE value. To analyse distribution of the occurrence, illustrate gaussian and density curve distribution also.

and how big the gap between some values is. In other words, how quickly can performance deteriorate due to minor changes.

Firstly, we analyze the histogram of four categories. In Figure 7, it can be seen that Gaussian and density curve are similar but not equivalent. So it could be that the two functions approach each other.

The last bar is the largest. That is because of trees with higher number of parameter than number of categories all achieve peak performance. It is also evident, that MLE values converge to this maximum likelihood value.

As can be seen in Figure 4, in the interval between -270 and -280, most values are represented. That means many tree compositions seem to be perform in this range again. In the figure you can also see that outliers of worse tree constellations are not often represented. Because of the broad spectrum of outlier incidents, there are greater gaps. Due to the greater distance between these values, the performance is worse faster.

Next, we compare how the MLE performance behaves in the case of five categories in Figure 8. Now the figure shows that the Gaussian and density curve are more diverse. This means that no statement about the Gaussian distribution of the MLE performance is possible.

Furthermore in the case of five categories, more values are represented in the histogram and less gaps between the bad performing trees appear. A comparison of Figure 7 and Figure 8 shows that that the occupancy of the values is much denser in Figure 8. Due to this density it is now more difficult to get a much worse performance through small changes. More tree compositions perform worse, so that it is difficult to make an general statement about bad parameter combinations and tree depth. In addition, some small gaps exists between the bad performance, which still makes it possible to get into worse values faster.

In conclusion, it seems like it is possible to make an assumption which numbers and constellations of parameters support a better performance for multinomial process trees.

5. Discussion

In this project, new theories for optimizing the performance of MPT were developed. Thereby, three different aspects were studied.

Firstly, how the performance changes, if the number of parameter is modified. The result is that a tree with a small number of parameter performs worse than a tree with more parameters. In more detail, a tree achieves the best performance with one parameter less than number of categories. Along with it, the performance stagnates as soon as at least

this number of parameters is given. This means that the performance converges. The interpretation of this behaviour can be tracked back to systems of linear equations if the equations have an infinite set of solutions.
Secondly, the influence of categories at the end of the paths was studied. This resulted in the observation that the modification of categories does not influence the model's performance. A reason for this could be that only the classes and not the structure are altered. Thus, only the name of the paths change which has no effect on a mathematical level. In other words, the path itself is not dependent on the class assignment.
The third aspect that was studied is how the performance is influenced by swapping the subtrees. It can be seen that by some parameter combinations the modification of the structure can have a high influence on the performance. This means that if the structure was changed by swapping a subtree, the performance becomes worse. It is noticeable that if the parameter combination has more different parameter, the performance of MLE is generally better. That means for the BMPT grammar, if parameters of the same name follow each other, the performance of MLE is worse. For example, the parameter combination of aba achieve better results of MLE than aab. However, in most cases, balanced trees achieve the best performance. In other cases, performance of MLE becomes worse with small modifications.
Finally, the distribution of performance was considered. The result is that the performance is negative logarithm distributed. This means that there are no models that will achieve the same results. However, no Gaussian distribution applies.

5.1. Problems

The first problem was that the estimator does not change if the number of parameter is increased. In the case that the number of parameters corresponds to the number of categories the performance of MLE stagnates.
This leads to the assumption that more parameters than categories are absolutely unnecessary. The explanation for this observation is that if the number of parameters is higher than the number of categories, the parameters have no unique estimates. If we compare it with systems of linear equations we see that if we have more unknown parameters than equations, the results depends on one unknown parameter. Consequently, if we reduce this to the problem with MPT, we can make the assumption for this case that parameters depend on the category. As a result, the MLE does not change if we have a higher number of parameters than categories.
Secondly, if the number of parameters is reduced, the values of MLE change. This makes it difficult to make an assumption which tree structure is better and which is worse. The problem is that if the trees are tested with two different datasets, the results are completely different. In addition, it is not possible to make general assumption about which tree composition achieves better performance of MLE. This problem can be handled if the trees are fitted many times. The calculated results are then averaged. With the mean of all fits it is possible to make a generally statement which structures of the trees optimize MLE. On the basis of static results, it is now possible to make an emprically proven statement.
Thirdly, after the computation of 1000 fits for four categories started, a new problem showed up. The first plot of this project should have the mean of each tree of 1000 cycles, but RStudio can not handle such a big datastructure. Thus, the plots for the categories have the means with fewer cycles. During the computation of 300, 500 and 700 cycles, I noticed that the results in the plots looks quite the same. That means that there must exists regulations of the tree structures which makes it possible to make an assumption about if a model performs well.

5.2. Conclusion

By and large, when creating an MPT, the best performance is achieved if the number of parameters is set to one less than the number of categories. It is valid:

$$numberParameter = numberCategory - 1 \tag{16}$$

In addition, when using many parameters nothing will be changed in performance. As a consequence the tree composition becomes more complex.

References

Heck, Daniel W., Arnold, Nina R., and Arnold, Denis. Treebugs: An r package for hierarchical multinomial-processing-tree modeling. *Behavior Research Methods*, 50:264–284, 2018.

Hilbig, Benjamin E, Moshagen, Morten, Nadarevic, Lena, Erdfelder, Edgar, Auer, Tina-Sarah, and Aßfalg, André. Multinomial processing tree models: A review of the literature. *Journal of Psychology*, 217:108–124, 2009.

Hu, Xiangen and Batchelder, William H. The statistical analysis of general processing tree models with the em algorithm. *Psychometrika*, 59:21–47, 1994.

Knapp, Bethany R. and Batchelder, William H. Representing parametric order constraints in multi-trial applications of multinomial processing tree models. *Journal of Mathematical Psychology*, 48:215–229, 2004.

Navarro, Daniel J. and Myung, In Jae. Model evaluation and selection. 2004.

Purdy, Brendan P. and Batchelder, William H. A context-free language for binary multinomial processing tree models. *Journal of Mathematical Psychology*, 53:547–561, 2009.

Riefer, David M. and Batchelder, William H. Multinomial modeling and the measurement of cognitive processes. *Psychological Review*, 95:318–339, 1988.

Singmann, Henrik and Kellen, David. Mptinr: Analysis of multinomial processing tree models in r. *Behavior Research Methods*, 45:560–75, 2013.

Wickelmaier, Florian and Zeileis, Achim. Using recursive partitioning to account for parameter heterogeneity in multinomial processing tree models. *Behavior Research Methods*, 50:1217–1233, 2018.

A. Appendix

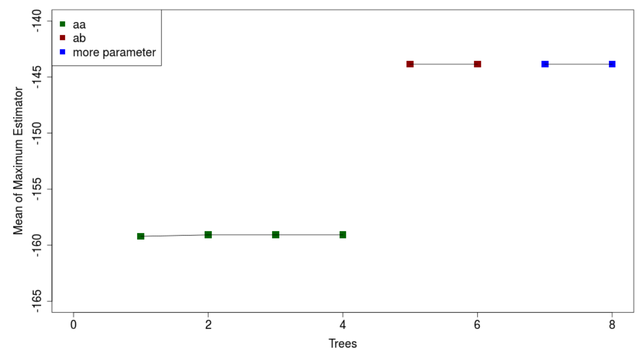


Figure 9. Results of 700 differents fits.

The purpose of this section is to give a short overview about the results of three categories. Since the results are not very varied, three categories are not very relevant for analysis. As you can see in the Figure 9, it is possible to study only five relevant cases. Thus three categories do not have a strong significance for the statements of section three. However, again it is illustrated that the performance converge and a comparison of the performance of MLE with one and two parameters depict a difference. The modification of the tree depth result in no difference between balanced tree and tree with depth of two.