# USE OF DECISION TREE TO DETECT GTTM GROUP BOUNDARIES


*Yuji Miura, Masatoshi Hamanaka*

University of Tsukuba

yuji@music.iit.tsukuba.ac.jp, hamanaka@iit.tsukuba.ac.jp

*Keiji Hirata*

NTT Communication Science Laboratories

hirata@brl.ntt.co.jp

*Satoshi Tojo*

Japan Advanced Institute of Science and Technology

tojo@jaist.ac.jp



## ABSTRACT

We describe σGTTM that combines the generative theory of tonal music (GTTM) and statistical learning. We previously devised exGTTM, which has accommodated the original GTTM to computer implementations. The exGTTM has adjustable parameters, these parameters have to be manually configured. Therefore, it is not perfectly suited for automation. To make complete automation possible, we combined statistical learning with GTTM to create σGTTM. To prevent the sparseness problem, we abstracted data properly by using the GTTM rules for analyzing musical structures. We use the abstracted data to construct a decision tree, which is a model of decisions and their possible consequences. With σGTTM, we can segment a melody automatically from the decision tree, dependent on conditional probability. Experimental results showed that σGTTM outperformed the baseline exGTTM.


## 1. INTRODUCTION

The goal of our research is to develop a music system that can be used to output musical expressions in accordance with the intention of a novice composer. Since current music sequencers operate only on surface structure of music such as notes and rests, they are not truly helpful for novice composers. However, if deeper musical structures, such as the melody, rhythm, and harmony, could be automatically analyzed on a computer, it would become possible for musical novices to produce advanced musical expressions. Our research has the goal of automating music structural analysis. For this, we have developed σGTTM, which combines a music theory called the generative theory of tonal music (GTTM) [5] with statistical learning. Here, we show that σGTTM can be used to segment a melody, the function that is the foundation of the structural analysis of music.

Our method statistically learns the relationship between an input score and an output melody segmentation. However, if we were to deal with a score as it is, the number of the combinations of training data would be very large. For example, the pitch, length, etc., of each note are described as absolute values. For this reason, it is possible that the pitches or lengths that are only slightly different will be learned as completely different data. This leads to a sparseness problem; the occurrence probabilities of combinations not in the training data are set to zero. To avoid combinations outside the training data being set to zero, we abstracted training data by using GTTM. By statistically learning the abstracted training data, the melody segmentation process can be automated.

## 2. RELATED WORKS

Music theory provides us with methodologies for analyzing and transcribing musical knowledge, experiences, and skills from a musician's way of thinking. Unlike other music theories [2, 6, 10], GTTM is based on structural rules which clarify the relation in parts of a piece and present the whole organization in detail. Thus, we regard the theory an adequate computational model of music.

GTTM is composed of four modules, each of which assigns a separate structural description to a listener's understanding of a piece of music. These four modules output a grouping structure, a metrical structure, a time-span tree, and a prolongational tree, respectively (Figure 1). The detection of local grouping boundaries in the grouping structure corresponds to melody segmentation. When potential local grouping boundaries are detected, grouping preference rules (GPRs 1, 2, and 3) are used for choosing the most appropriate one [5]. Although GPRs indicate the superiority of one structure over another, in practice it is known that there is a problem with conflict between rules.

Hamanaka developed exGTTM, an extended GTTM that is suitable for computer implementation [3]. exGTTM is premised on the data that any musical piece has more than one correct musical interpretation, and it uses parameters to eliminate as many ambiguities as possible. The automatic time-span tree analyzer (ATTA) [3] was also developed to solve the conflict between rules.

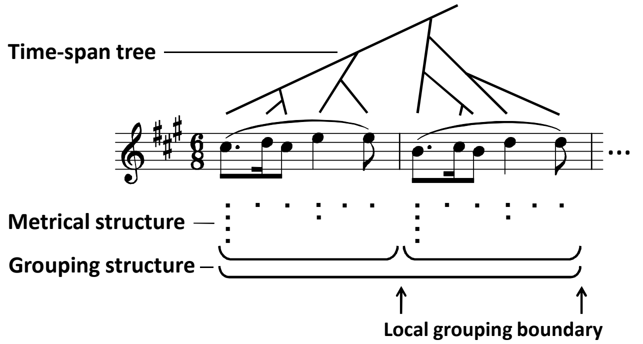


**Figure 1**. Grouping structure, metrical structure, and time-span tree.

However, ATTA needs a difficult manual configuration of fifteen parameters. The full automatic time-span tree analyzer (FATTA) can automatically estimate the optimal parameters on the basis of the stability of the time-span tree [4]. However, FATTA is not able to outperform the manual configuration of ATTA; because there are so many parameters and also many of them are dependent each other, we cannot optimize them easily.

Conventional melody segmentation methods such as Grouper of the Melisma Music Analyzer by Temperly [11], LBDM by Cambouropoulos [1] require the user to make manual adjustments to the parameters, and hence are not completely automatic. Although Temperly [12] has also employed probabilistic model, it was not applied for melody segmentation. The unsupervised learning model (IDyOM) proposed by Pearce et al. makes no use of rules of music theory with regard to melodic phrases, and it performed as well as Grouper, LBDM, and separate GPRs did [7]. However, as σGTTM learns all GPRs statistically and collectively, we expect that σGTTM performs better than a model that uses only statistical learning does.

## 3. USE OF DECISION TREE TO RESOLVE CONFLICT BETWEEN RULES

Using the abstracted training data, conflict between rules can be resolved by constructing a decision tree, which is a model of decisions and their possible consequences.

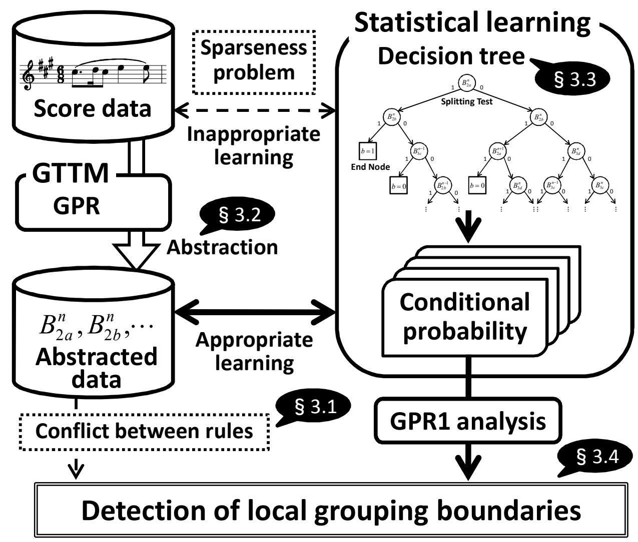


**Figure 2**. σGTTM: System for automatically detecting local grouping boundaries.

Moreover, melody segmentation can be automated by calculating the conditional probability that a local grouping boundary exists from the decision tree. The whole σGTTM system is shown in Figure 2.

Statistical learning requires training data in which GPR and the correct boundaries are applied to the score. We used the correct data for evaluating exGTTM as the training data [3]. We asked musicology experts to analyze a hundred 8-bar monophonic pieces of classical music. They were asked to analyze them in strict accordance with GTTM. Three other experts crosschecked these manually produced results.

### 3.1. Conflict between Rules

Because there is no strict order for applying GPRs, the conflict between rules often occurs when applying GPRs and results in ambiguities in analysis. An example analysis of the local grouping structure extracted from the training data is shown in Figure 3. In a state showing the distribution of the rule on a score, statistical learning was used to automatically detect the same local grouping

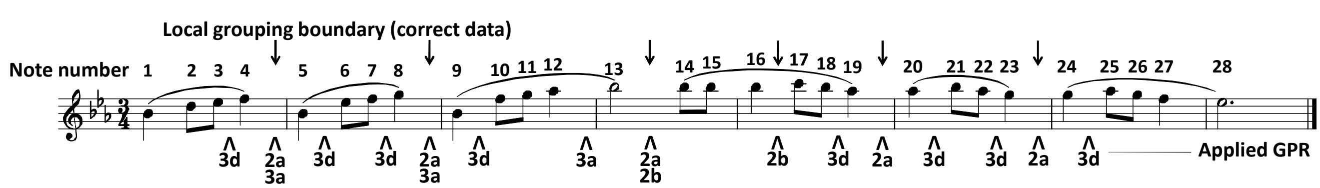


**Figure 3**. Example analysis of local grouping boundaries.

boundary as given by the correct data. However, detecting the boundary of a local grouping is difficult because of a conflict between rules. As shown in the analysis of Figure 3, GPR3d is applied between 18th and 19th note as the rule claims the greater change in duration in neighboring notes tends to be a boundary, while GPR2a is applied between 19th and 20th note which prioritizes the temporal proximity by slur and rest. However, the only one of them should be applied here to prohibit the 19th to be a singleton. Although GPR2a is applied in the example, the method of choosing which rule to apply is not clearly described in GTTM.

### 3.2. Abstraction of Training Data

A decision tree expresses important knowledge about a certain attribute in the data (target attribute) that need to be observed in the combination of the rules (conditional attribute). In a decision tree, one suitable rule is chosen out of a set of conditional attributes, and data is divided into two in accordance with the value of the rule. This division is called a splitting test. A decision tree is built by repeating these tests.

The data selected for constructing a decision tree depend on the purpose. Our purpose is resolving rule conflict between certain notes, and it is important to consider whether GPR is applied between the notes ($B^n_{2a}$, $B^n_{2b}$, $B^n_{3a}$, $B^n_{3b}$, $B^n_{3c}$, $B^n_{3d}$). If GPR1 is taken into consideration, application of the rule between the previous ($B^{n-1}_{2a}$, $B^{n-1}_{2b}$, $B^{n-1}_{3a}$, $B^{n-1}_{3b}$, $B^{n-1}_{3c}$, $B^{n-1}_{3d}$) and the following ($B^{n+1}_{2a}$, $B^{n+1}_{2b}$, $B^{n+1}_{3a}$, $B^{n+1}_{3b}$, $B^{n+1}_{3c}$, $B^{n+1}_{3d}$) pairs of notes is also important. The existence of these rules is defined as a conditional attribute. Furthermore, the existence of a local grouping boundary ($b$) is defined as target attribute. If a local grouping boundary is between notes, we assign a value of 1 to $b$, otherwise $b = 0$.

### 3.3. Construction of Decision Tree

The decision tree is constructed by using a decision tree generation algorithm, C4.5, derived by J. R. Quinlan [8]. The following algorithm generates a decision tree from a set $D$ of cases. The tree for $D$ has test $T$ (splitting test) as its root with one subtree for each outcome $T_i$ that is constructed by applying the same procedure recursively to the cases in $D_i$. A family of possible tests is examined, and one of them is chosen to maximize the value of various splitting criterion.

The splitting criterion used by C4.5 is the gain ratio. The number of classes is defined as $C$ and the proportion of cases in $D$ that belongs to the $j$th class is defined as $p(D, j)$. The uncertainty of the class to which a case in $D$ belongs is expressed as

$$Info(D) = -\sum_{j=1}^{C} p(D,j) \times \log_2(p(D,j)) \qquad (1)$$

and the corresponding information gained by a test $T$ with $k$ outcomes as

$$Gain(D,T) = Info(D) - \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times Info(D_i)\ . \qquad (2)$$

The information gained by a test is strongly affected by the number of outcomes. On the other hand, this split information

$$Split(D,T) = -\sum_{i=1}^{k} \frac{|D_i|}{|D|} \times \log_2\left(\frac{|D_i|}{|D|}\right) \qquad (3)$$

tends to increase with the number of outcomes of a test. The gain ratio criterion is used to assess the desirability of a test in terms of the ratio of its information gain to its split information. The gain ratio of every possible test is determined, and thus the split with maximum gain ratio is selected.

Furthermore, C4.5 prunes the initial tree, identifying subtrees that contribute little to predictive accuracy and replacing each by a leaf (end node).

### 3.4. Detection of Local Grouping Boundaries

Part of a constructed decision tree is shown in Figure 4. Each end node is labelled in accordance with a value with a large frequency of $b$ of the data classified into the node. Therefore, the adequacy of a boundary location for a combination of conditional attributes can be calculated. When this conditional probability is 50 percent or more, the system judges that there is a local grouping boundary ($b = 1$). When this conditional probability is less than 50 percent and when data on the combination of conditional attributes do not exist, the system judges that no local grouping boundary exists ($b = 0$). For example, in Figure 4, if it is $B^n_{2a} = 1$ and $B^n_{2b} = 1$, the local grouping boundary exists between $n$ notes.

The system then analyses each musical piece unit (Figure 5). When GPR1 is violated, that is, there are boundaries on both sides of one note, the local grouping boundary with the smaller probability is deleted.

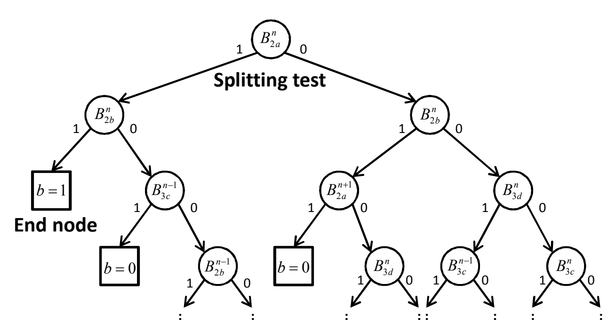


**Figure 4**. Part of constructed decision tree.

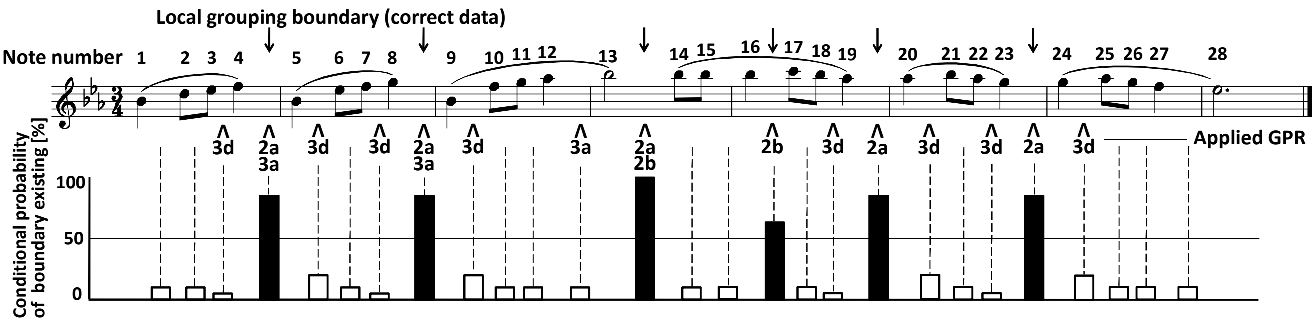


**Figure 5**. Analysis of each musical piece.

## 4. EXPERIMENTAL RESULTS

We evaluated the performance of the σGTTM system using the F-measure, which is given by the weighted harmonic mean of precision $P$ and recall $R$.

$$F_{measure} = 2 \times \frac{P \times R}{P + R} \quad (4)$$

$P$ is the number of correct boundaries divided by that of boundaries the system has suggested, while $R$ is the number of retrieved correct boundaries divided by that of all the truly correct boundaries.

We have evaluated the result by a 10-fold cross-validation technique. Experimental results show that the automatic local grouping boundary detection (σGTTM) outperformed the baseline ATTA (Table 1).

| | Precision $P$ | Recall $R$ | F-measure |
|---|---|---|---|
| σGTTM (new) | 0.764 | 0.630 | 0.690 |
| ATTA (previous) | 0.737 | 0.441 | 0.552 |

**Table 1**. Experimental results.

## 5. CONCLUSION

We have described a method of using statistical learning to automatically detect local grouping boundaries in GTTM. This method has adequately solved the conflict between rules by using a decision tree. Our experimental results have shown that the σGTTM system outperformed the baseline F-measure of ATTA. The σGTTM system will be exhibited on the Web as CGI[1]. Therefore, a benchmark with other melody segmentation methods is possible [9].

We plan to make it possible to construct a hierarchical grouping structure. Moreover, we plan to use statistical learning to implement other analyses of GTTM (metrical structure, time-span tree, and prolongational tree) on a computer. We can already interpret the conditional probability of boundary existing for each note as boundary strength, and thus, we plan to investigate whether the boundary strength is applicable to other research on the composition of music.

## 6. REFERENCES

[1] Cambouropoulos, E. "The Local Boundary Detection Model (LBDM) and its application in the study of expressive timing", in *Proceedings of the International Computer Music Conference*, 2001, pp. 290-293.

[2] Cooper, G. and Meyer, L.B. *The Rhythmic Structure of Music.* Chicago: The University of Chicago Press, 1960.

[3] Hamanaka, M., Hirata, K., and Tojo, S. "Implementing A Generative Theory of Tonal Music", in *Journal of New Music Research*, Vol. 35, No. 4, pp. 249-277, 2007.

[4] Hamanaka, M., Hirata, K., and Tojo, S. "FATTA: Full Automatic Time-span Tree Analyzer", in *Proceedings of the International Computer Music Conference*, 2007, pp. 153-156.

[5] Lerdahl, F. and Jackendoff, R. *A Generative Theory of Tonal Music.* Cambridge: The MIT Press, 1983.

[6] Narmour, E. *The Analysis and Cognition of Basic Melodic Structure.* Chicago: The University of Chicago Press, 1990.

[7] Pearce, M. T., Müllensiefen, D., Wiggins, G. A. "A comparison of statistical and rule-based models of melodic segmentation", in *Proceedings of the International Conference on Music Information Retrieval*, 2008, pp.89-94.

[8] Quinlan, J. R. *C4.5: Programs for Machine Learning.* Morgan Kaufman Publishers, 1993.

[9] Spevak, C., Thom, B., and Hothker, K. "Evaluating Melodic Segmentation", in *Proceedings of the International Conference on Music and Artificial Intelligence*, 2002, pp. 168-182.

[10] Temperly, D. *The Cognition of Basic Musical Structure.* Cambridge: The MIT Press, 2001.

[11] Temperly, D. The Melisma Music Analyzer. http://www.link.cs.cmu.edu/music-analysis/, 2003.

[12] Temperly, D. *Music and Probability.* Cambridge: The MIT Press, 2007.

[1]http://music.iit.tsukuba.ac.jp/sigmagttm.html