An approach for identifying salient repetition in multidimensional representations of polyphonic music

Jamie Forth and Geraint A. Wiggins
Goldsmiths, University of London,
London, SE14 6NW, UK
{j.forth,g.wiggins}@gold.ac.uk

February 18, 2009

Abstract

SIATEC is an algorithm for discovering patterns in multidimensional datasets (Meredith et al., 2002). This algorithm has been shown to be particularly useful for analysing musical works. However, in raw form, the results generated by SIATEC are large and difficult to interpret. We propose an approach, based on the generation of set-covers, which aims to identify particularly salient patterns that may be of musicological interest. Our method is capable of identifying principal musical themes in Bach Two-Part Inventions, and is able to offer a human analyst interesting insight into the structure of a musical work.

1 Introduction

This paper attempts to identify the repetition of perceptually salient patterns in symbolically represented music. A geometrical approach is adopted in which pieces of music are represented as multidimensional datasets. Following the work of Meredith et al. (2002, 2003) and Meredith (2006), we have implemented SIATEC, a pattern induction algorithm, and have conducted a series of use-case studies in order to investigate the properties of the generated results in terms of musicological value and perceptual salience. SIATEC is known to discover many more patterns than are typically of interest in any music analysis context, and the results can be difficult to interpret (Meredith et al., 2002, p. 340). We propose a post-processing step, similar in character to the NP-hard minimum weighted set cover problem (Karp, 1972), in which various heuristics can be employed in order to optimise the results in terms of specific music-analytic objectives.

Viewing this problem in the light of theoretical computer science is beneficial on several counts. Firstly, the knowledge that an exact solution cannot be computed in polynomial time assures us that we need to seek approximate solutions. Secondly, a great deal of research has been conducted in trying to establish methods for deriving solutions within acceptable bounds of approximation. The standpoint of computer science informs understanding of the nature of this problem, and provides examples of rigorously tested methods that may be applicable to our case. The inherent level of ambiguity in set-covering problems accords with the common situation in musical analysis whereby different interpretations of a work may be considered equally valid and correct. In order for an analyst to reach any firm conclusion, compromises must be made, which are often informed by conventions (heuristics) of music theory.

An applied aim of this research is to develop tools suitable for various music-analytic tasks. Within the field of musicology such tools may assist conventional score analysis, and may prove particularly useful for larger-scale corpus analysis. The latter overlaps with interests of music information retrieval, where such techniques may be applied in order to extract commonly occurring patterns as the basis for classification. A composer may also be able to gain inspiration by analysing a work in progress and arriving at a fresh perspective. More novel applications may be found in music psychology or artificial intelligence, where large collections of music could be analysed in order to derive data for training or testing models of musical behaviour. The work discussed in this paper has been conducted within a larger project investigating musical creativity. This wider project requires a considerable body of musical knowledge with which to train artificial musical agents in an attempt to simulate aspects of creative musical behaviour. In order to justify claims of potential utility, tools for automated musical analysis must be adaptable to individual tasks, which may involve handling varying degrees of ambiguity, and necessarily must scale to real-world musical problems.

2 Previous Work

The concept of a musical pattern entails repetition. The definition of `SIATEC` ensures the enumeration of *all* maximal repeated patterns (Meredith et al., 2002, pp. 331–333). A large number of these discovered patterns will usually prove to be of little interest from a musical or perceptual perspective, and this is one problem our heuristics must address. Yet a more complicated issue concerns the many *types* of salient repetitive patterns that may exist in a musical work. In other words, the kinds of patterns that are likely to be of interest, and the ways in which they are interesting, may vary considerably.

There is agreement amongst both musicologists and music psychologists as to the importance of repetition in music (see, for example, Lerdahl & Jackendoff, 1983; Nattiez, 1987; Krumhansl, 1997). One cross-cultural study based on fifty musical works found that 94 percent of all musical passages longer than a few seconds in duration were repeated at some point in the work (Huron, 2006, pp. 228–9). However, this result does not account for the role of repetition in music in its entirety, because repetition may exist in many forms beyond the exact repetition of musical events in sequence. For example, melodies may still be perceived as instances of the same basic melodic motif despite being transposed in pitch or transformed in time. Indeed, perceptual similarity may pertain for any individual listener under an arbitrary number of processes of elaboration and transformation. In the context of computational analysis, therefore, careful consideration must be given to the notion of pattern equality.

Much previous work in this area has concentrated on techniques for string matching. Despite successes within certain specialised tasks, notably concerning monophonic melodies, various limitations of string methods become apparent in the context of music (Lemström & Pienimäki, 2007).

An alternative approach to string matching exists in the form of geometrically-based algorithms. Within a geometrical framework, the individual note events of a piece of music correspond to single points in a multidimensional space.[1] A family of algorithms related to `SIA` (*Structure Induction Algorithm*) have been developed for pattern discovery and matching in multidimensional datasets (Meredith et al., 2002; Wiggins et al., 2002; Meredith et al., 2003; Meredith, 2006). The initial development of these techniques was motivated to a large extent for application to music, but are equally applicable in other domains where objects may be adequately represented in a multidimensional space.

Following Meredith et al. (2002, p. 328), we define a datapoint as a k-tuple of real numbers, and a pattern P or dataset D as a finite set of k-dimensional datapoints. We reserve the term *dataset* to refer to a complete set of datapoints we wish to process, for example, a piece of music, while *pattern* refers to a subset of the dataset. A translator is a vector that maps from one instance of a pattern to another within a dataset. More precisely, a vector t is a *translator* for P in D if and only if the translation of P by t is also a subset of D.

The basic `SIA` algorithm computes all the maximal repeated patterns in a dataset (Meredith et al., 2002, pp. 334–5). The algorithm finds the largest non-empty set of translatable datapoints for every positive translation possible within the dataset. Hence, each pattern discovered by `SIA` is called a *maximal translatable pattern* (MTP). The worst case running time of `SIA` is $O(kn^2 log_2 n)$.

An important extension to `SIA` is `SIATEC` (Meredith et al., 2002, pp. 335–8). `SIATEC` underlies both the approach to pattern discovery adopted in the present paper, as well as the closely related `COSIATEC` algorithm, which will be discussed below. Like `SIA`, the `SIATEC` algorithm enumerates all the maximal translatable patterns in a dataset, but also groups them into equivalence classes. A *translational equivalence class* (TEC) is represented as an ordered pair $\langle P, T(P, D)\rangle$, where P is a maximally translatable pattern, and $T(P, D)$ is the set of translators for P in the dataset D. The worst-case running time of `SIATEC` is $O(kn^3)$.

[1] A variation on this approach, based on sets of line segments in space, is discussed in Ukkonen et al. (2003)

Composition	Number of datapoints	Number of TECs
BWV 772	458	9035
BWV 773	634	11724
BWV 774	494	9882
BWV 775	443	9304
BWV 776	733	15978
BWV 777	547	17209
BWV 778	473	11103
BWV 779	598	11731
BWV 780	558	11995
BWV 781	439	9038
BWV 782	568	11306
BWV 783	685	15969
BWV 784	564	12250
BWV 785	592	16782
BWV 786	477	10407

Table 1: Number of datapoints (notes) and the number of discovered TECs in J. S. Bach's Two-Part Inventions

Even for small datasets, the raw output of `SIATEC` can quickly become unmanageably large, as can be seen in Table 1. Furthermore, the patterns are diverse in size and structure, and on the whole are not readily intuitive. It would be straightforward to rank the discovered patterns based on a set of criteria; for example, to sort by pattern size $|P|$, or the number of pattern repetitions $|T(P, D)|$. However, such a simplistic approach presents two particular difficulties. Firstly, the method does not lend itself to any principled means of deciding how many of the most highly ranked patterns should be selected as being representative of the repetition in the dataset. Secondly, this method would preclude the ability to make inter-pattern judgments, that is, for the value of one pattern to influence the value of another, due to combinatorial explosion.

`COSIATEC` is one method for automatically identifying a subset of 'interesting' patterns from amongst the many patterns discovered by `SIATEC` (Meredith et al., 2003; Meredith, 2006). `COSIATEC` is designed to generate compressed representations of datasets by representing them in terms of highly repetitious subsets. The algorithm first runs `SIATEC`, generating a list of $\langle P, T(P, D)\rangle$ pairs, and then selects the best pattern based on heuristics. The algorithm then removes from the original dataset all the datapoints that are members of the occurrences of the chosen pattern P. The process continues until all the datapoints have been removed from the dataset. The resulting set of patterns are collectively termed a *cover* (Wiggins et al., 2002). In this case, each datapoint is represented in a cover exactly once.

For each iteration of `COSIATEC`, the remaining patterns are evaluated according to the three heuristics of *coverage*, *compression ratio*, and *compactness*, and the most highly valued pattern is selected to become part of the resulting cover. These heuristics are also employed in the present work, and are defined below.

Although motivated by compression, `COSIATEC` has been shown to identify principal musical themes in pieces of music (Meredith et al., 2003; Meredith, 2006). This is explicable given the very nature of a musical theme, which will typically appear numerous times during a work, making it an ideal pattern for use in the encoding of a compressed representation. Therefore, `COSIATEC` offers a tidy solution to the difficulties of sifting through the output of `SIATEC`. The problem becomes one of generating optimal (finite) covers given particular heuristics. Furthermore, being a greedy algorithm, the generation of covers entails a degree of pattern co-dependency, since previously selected patterns will affect the outcome of later iterations.

3 An Alternative Selection Algorithm

The approach to pattern discovery in the present work follows a similar strategy to that of COSIATEC. We again formulate the problem as one of cover generation, but explore possibilities created by shifting the emphasis away from purely optimising compression. The foremost difference in this approach is that we only apply SIATEC once—to initially process the entire dataset. Furthermore, we evaluate only once the musicological value of each pattern discovered by SIATEC. Therefore, the musicological value of each pattern is determined within the same initial context, prior to selection. The rational being that, compared with COSIATEC, the selection process in this case more closely relates to the process of musical listening, since listeners perceive patterns in a musical work in the context of all the notes. The resulting values, or weightings, are adjusted during subsequent iterations of the selection process, but only by the single varying factor of the number of currently uncovered datapoints that are covered by a particular pattern. This process is described below as an example of a weighted set-cover problem.

Also explored is the effect of loosening the constraint requiring that each datapoint only be covered once. This enables us to consider datapoints as belonging to multiple patterns within a single cover, creating the potential to infer structural relationships between repeated elements. A similar strategy could be pursued within COSIATEC, but only within the context of each iteration as covered datapoints are removed at each stage.

3.1 Generating Set Covers

The approach taken in this paper for generating covers from musical patterns discovered by SIATEC can be described in terms of the widely known NP-hard set-covering problem. Cormen et al. (2001, pp. 1033–4) describe this problem:

> An instance $(X, \mathcal{F})$ of the ***set-covering problem*** consists of a finite set X and a family $\mathcal{F}$ of subsets of X, such that every element of X belongs to at least one subset in $\mathcal{F}$:
>
> $$X = \bigcup_{S \in \mathcal{F}} S. \tag{1}$$
>
> We say that a subset $S \in \mathcal{F}$ ***covers*** its elements. The problem is to find a minimum-size subset $\mathcal{C} \subseteq \mathcal{F}$ whose members cover all of X.
>
> $$X = \bigcup_{S \in \mathcal{C}} S. \tag{2}$$

In other words, the desired outcome of this optimisation problem is to find the smallest number of subsets in $\mathcal{F}$ that account for (cover) each element in X at least once.

The standard approach to set-cover utilises a greedy algorithm based on the heuristic of selecting at each stage the set S that covers the largest number of uncovered elements in X. Ties are broken randomly. This algorithm is the best known performing algorithm, achieving an approximation ratio of $(1 - o(1)) \ln n$ (Feige, 1998, p. 634).

In the context of SIATEC, X is equivalent to a dataset D. Similarly to the selection process of COSIATEC, we consider all instances of a pattern P in a dataset D as constituting a single subset of D. We use P_{TEC} in order to explicitly refer to a subset of D that is the union of each occurrence of P in D. Therefore, each subset $S \in \mathcal{F}$ is a P_{TEC}, and $\mathcal{F}$ is equivalent to the entire set of patterns (considering each as a P_{TEC}) discovered by SIATEC: $\mathcal{F} = \langle P_{\text{TEC}0}, P_{\text{TEC}1}, \ldots, P_{\text{TEC}n-1} \rangle$.

For our purpose, simply finding a minimum-sized subset $\mathcal{C} \subseteq \mathcal{F}$ does not adequately characterise the problem, since we require a means of specifying which patterns should be considered 'better' or 'worse' by the selection algorithm. Therefore, a more appropriate model for the problem is the equally well-known generalisation of the minimal set-cover problem: the minimum weighted set-cover problem (Chvatal, 1979). Typically, a greedy algorithm is also adopted, except that covers are selected in the order that minimises the ratio of cover weight to number of elements covered.

To place the SIATEC cover problem in this context, it is necessary to attach weighting values to each of the discovered patterns in $\mathcal{F}$. This step is similar to the use of heuristics in COSIATEC, except that in this case the values are calculated only once, prior to the actual selection process. The higher a pattern scores according to a heuristic, the more relevant it is considered to be to the analysis. The heuristics used to calculate these values are discussed in the following section.

Contrary to the more typical formation of weighted set-cover problems, the selection process in this case seeks to maximise: *weight*$(P_{\text{TEC}}) \times$ *cover-ratio*(P_{TEC}) where *weight* is the musicological value of the pattern P_{TEC} predetermined by a set of evaluation heuristics, and *cover-ratio* is the ratio between the number of uncovered datapoints in the dataset D that are members of the pattern P_{TEC}, to the total number of uncovered datapoints in D. The definition of cover-ratio here is based on the concept of *coverage* used in COSIATEC, which is defined as: 'the number of datapoints in the dataset that are members of occurrences of the pattern' (Meredith et al., 2003, p. 7). A minimum cover-ratio threshold, which must be exceeded for a pattern to become a member of the cover, has proved a useful parameter in the generation of set-covers. In practice, a minimum cover-ratio threshold of between five percent and twenty percent is the typical useful range. Higher values in this range are particularly useful in order to generate covers consisting of only a small number of patterns. High cover-ratio values may lead to not every datapoint in D being represented in the set-cover. However, this is not necessarily unsatisfactory, since not every single note in a piece of music may prove sufficiently salient to be accounted for by a typical listener, or even professional analysts.

Once it has been determined that a pattern should become a member of the set-cover, a final step is taken to determine whether a pattern should be considered a *primary* or *secondary* pattern. This step is simply intended to make the generated results more comprehensible for the human analyst, by attempting to group together 'similar' patterns. If a pattern is the first pattern to be selected, it is simply defined as primary. Each subsequently selected pattern is compared to each existing primary pattern in terms of the number of datapoints they commonly cover. This is in order to identify the primary pattern that is 'most similar' to the newly selected pattern, quantified in terms of overlapping coverage. If the proportion of commonly covered datapoints is greater than an arbitrarily defined threshold—50 percent in this case—then the newly selected pattern is declared a secondary pattern, and grouped together with the most similar primary pattern. If the newly selected pattern is not considered similar to any of the other primary patterns it is declared a primary pattern. Whether a pattern is defined as primary or secondary has no bearing on the actual selection process, it is purely a means of organising the selected patterns, as well as offering an estimation of the number of distinct musical ideas present in the work.

4 Pattern Evaluation Heuristics

As noted above, there may be many different forms of repetition in a piece of music. It is therefore necessary to establish an evaluation criterion in order to automate the extraction of the kinds of repetitions that are considered relevant to an analytical objective. The cover-ratio heuristic, introduced in the previous section, is one measure of a pattern's value, and is recomputed for every pattern for each iteration of the cover generation algorithm. Here we describe two further heuristics that are used to provide static, or absolute, measures of the value of a pattern.

Compression ratio is defined as 'the compression ratio that can be achieved by representing the set of points covered by all occurrences of a pattern by specifying just one occurrence of the pattern together with all the non-zero vectors by which the pattern is translatable within the dataset' (Meredith, 2006, p. 13). Compression ratio can, therefore, be calculated in terms of coverage:

$$\text{compression ratio} = \frac{\text{coverage}}{|P| + |T\,(P, D)| - 1} \tag{3}$$

Compression ratio is particularly useful for identifying large, non-overlapping patterns that have many occurrences in a dataset.

The second heuristic used is *compactness*, which is defined as 'the ratio of the number of points in the pattern to the number of points in the region spanned by the pattern' (Meredith, 2006, p. 13). This heuristic applies to each *occurrence* of a pattern P belonging a TEC. Therefore, unlike compression-ratio which generates a single value for a TEC, compactness produces $|T\,(P, D)|$ values for each TEC. In order to arrive at a single value for a TEC, since the selection algorithm generates covers by selecting P_{TEC} subsets of D, the obvious approach is to use either the mean or maximum compactness value as the TEC weighting value. From a musical perspective, selecting the maximum pattern compactness value to determine the weighting of a TEC can be justified on the principle that a significant musical theme will typically have at least one relatively prominent (compact) occurrence in a work.

As discussed in Meredith et al. (2003) and Meredith (2006), the definition of 'region', for example, as a segment, bounding box, or convex hull, impacts on the computed compactness value. For our purposes, we employ the notion of a region as a segment, but also take into account the musical voicing within a pattern. Therefore, all the notes that occur inclusively between the first note onset of a pattern and the final offset are considered to belong to the region (segment) spanned by the pattern. However, unlike previous work, we calculate the ratio using only those notes in the region that are also members of the voices present in the pattern. This decision is based on the assumption that notes belonging to the musical voices present in a pattern are more likely to influence its perceptual salience, compared with notes belonging to other musical voices. This assumption is consistent with empirical findings related to melodic streaming (Bregman, 1990, pp. 61–64). The present definition of compactness, relying to a certain degree on specific musical concepts, is less generic than the original geometrical definition. However, it has proved to be the most satisfactorily performing variant in our study. Furthermore, given that our testing dataset consists of the fifteen J. S. Bach Inventions, the strict two-part texture of the music gives additional credence to the utilisation of voicing information in the selection process.

Patterns are initially assigned a default weight of one, and each heuristic is implemented such that it returns a normalised weighting value of between zero and one. Weighting combination is multiplicative. Therefore, if a pattern is rated most highly by each heuristic, it would retain the final value of one. If a pattern scores zero for any of the heuristics the final rating would also be zero.

The values generated by heuristics can also be scaled in a variety of ways. Most relevant to the current analysis is that values can be adjusted using either a linear or exponential mapping, and can also be given explicit minimum and maximum thresholds beyond which patterns are assigned a weighting of zero. Setting a threshold for a heuristic is particularly useful as a means of excluding a subset of patterns prior to the cover generation process. The removal of redundant subsets $S \in \mathcal{F}$ is common in the literature (Caprara et al., 1998, p. 2). The density of coverage present in $\mathcal{F}$ plays a significant role in determining the bounds of approximation (Clarkson & Varadarajan, 2005), so from a practical standpoint, the prior reduction in the size of a set cover instance may lead to improved solutions.

5 Results

We have applied the cover generation algorithm and heuristics discussed above to J. S. Bach's Two-Part Inventions (BWV 772–786). Each piece was analysed using a range of different parameters for each heuristic. Individual pieces from this set: BWV 722, 724 and 725, are also the subject of analysis in Meredith et al. (2003) and Meredith (2006). Therefore, we focus attention on these three pieces in order to draw some comparison.

The covers generated by our system robustly included the same patterns as those discovered by `COSIATEC`. A minor deviation from this is in BWV 775. `COSIATEC` selects the opening two bars as constituting the most important pattern in work. Our system makes an almost identical selection, except that it does not include the very first note. Both interpretations are valid when considering the full score. There are indeed many occurrences of the pattern discovered by `COSIATEC`, however, there are also many additional occurrences where the first note is different. The difference between the two results is presumably a reflection of a different balance of emphasis given to the heuristics used to evaluate the patterns.

Figures 1 and 2 show a complete cover generated from BWV 772 using the following heuristic parameters:

- Cover-ratio (min: 0.2)
- Compression-ratio (min: 0.25, max 1.0)
- Compactness (min: 0.25, max 1.0)

There are 9035 TECs discovered by SIATEC in BWV 772. After calculating the weights of each TEC using compression-ratio and compactness, only 129 TECs have non-zero weights. This considerable reduction indicates that a very large proportion of the patterns discovered by SIATEC are not relevant to this particular analytical focus. Generating a cover with the relatively high cover-ratio threshold of 20 percent produces a cover consisting of only six patterns—three primary and three secondary. Setting a lower threshold in this case tends to increase the number of secondary patterns discovered.

Patterns 1 and 2, the first and second patterns discovered, are the inversion of the subject, and subject itself respectively. These patterns are the same as those discovered by COSIATEC, and are labeled as the subject of the work as analysed by Dreyfus (1996, p. 10). The two secondary patterns, 1.1 and 2.1, are both clearly subsets of their parent patterns. From an analytical perspective, the most interesting aspect of these patterns is how their individual pattern of occurrence *differs* from that of their parent, which may be described as an instance of *entanglement* (Wiggins et al., 2002). This is particularly apparent for pattern 2.1 from bar 16 until the end, where many instances of the pattern overlap. As well as highlighting the high density of this simple descending three-note quaver pattern at the end of the piece, the change in the translations of pattern 2.1 in relation to pattern 2 suggest some sort of developmental change to the primary pattern. In fact, this change corresponds to the note that immediately follows an occurrence of pattern 2, which in these closing bars forms an interval of a 2nd. All previous occurrences, except for the occurrence preceding bar 9, are followed by a larger interval, most commonly a 5th.

It cannot really be argued that pattern 1.2 is a perceptually salient pattern when considered as a single occurrence. However, when taking into account the larger pattern that is formed by the overlapping occurrences, an important pattern emerges (Figure 3). Pattern 1.2 is indicative of the gap-fill pattern that accompanies pattern 1 in bars 3–5 and 11–13 (where it appears in the treble voice), and is also embedded in the structure of pattern 1 itself.

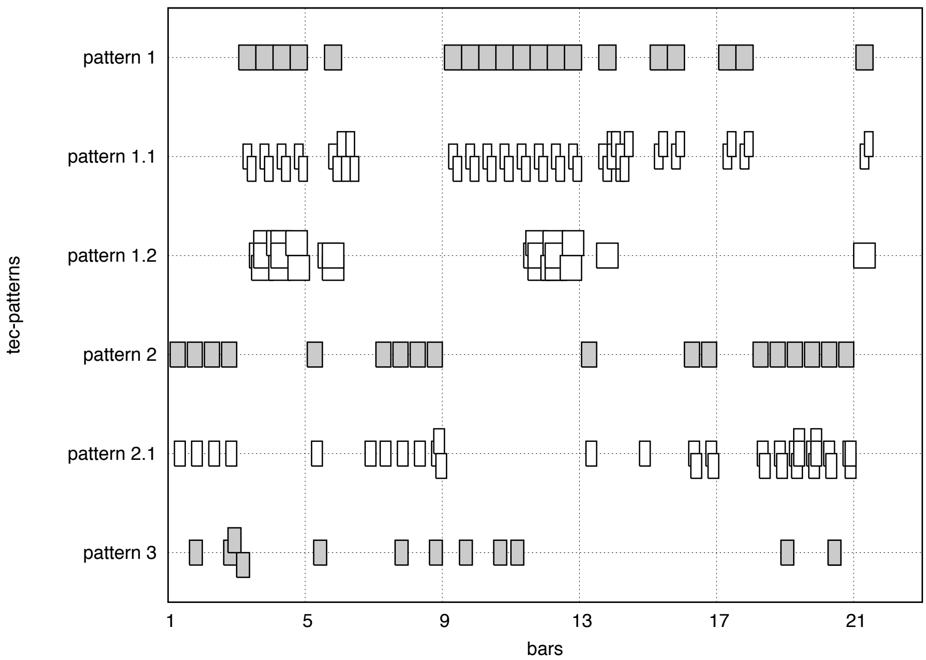

Figure 1: A schematic representation of the primary and secondary patterns selected from the SIATEC analysis of BWV 772. The filled boxes are primary patterns, the empty boxes are secondary patterns. Each box represents a pattern occurrence. To aid clarity, patterns that overlap are draw alternately above and below the line.

(a) Pattern 1: bar 3 (b) Pattern 2: bar 1 (c) Pattern 3: bar 1

Figure 2: The primary and secondary patterns selected from the SIATEC analysis of BWV 772.

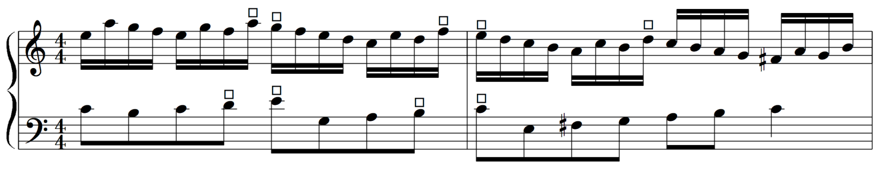

Figure 3: Bars 3–4 from BWV 772. Square boxes above notes signify the beginning of an occurrence of pattern 1.2.

6 Future Work

Applied examples from the literature present several variations on the greedy algorithm that have proved useful in particular domains, which may similarly be beneficial in our case. Marchiori & Steenbeek (1998) describe the `Enhanced Greedy` algorithm, which has a more sophisticated heuristic for breaking ties when adding new covering sets of equal size to the solution. At each iteration the algorithm also checks for (and possibly removes) sets that become 'nearly' redundant in the solution due to the addition of new sets. The `Iterated Enhanced Greedy` algorithm is also described, in which a subset of the currently best (smallest) cover is used as an initial partial solution for a further iteration of the algorithm. Another approach that would also warrant empirical investigation in this context is the multiple weighted set cover problem, which is a further generalisation of the basic set-cover problem where events must be covered a specified minimum number of times (Yang & Leung, 2005). Alternative approaches to the basic greedy algorithm, including approximate linear programming and exact branch and bound method, are discussed in Caprara et al. (1998).

7 Conclusion

An approach to automated musical analysis has been presented. The method is essentially a selection algorithm based on a set of heuristics that attempt to determine the quality of discovered patterns in terms of musical salience. The SIATEC algorithm is integral to the process, since it provides the initial set of discovered patterns from which the selection is made. Our method also owes much to the COSIATEC algorithm, which is also a means of selecting important patterns from the patterns discovered by SIATEC. The primary difference between our approach and COSIATEC is that our method is not based solely on the principle of optimising compression, but instead allows musicological principles to influence the outcome alongside information theoretic measures. As a result, we are able to select patterns that are deemed to be of musicological interest, but which may not lead to the generation of a complete or optimally compressed representation of the dataset, as is generated by COSIATEC. For example, we are able to select multiple patterns that share notes in common, but which have different patterns of occurrence within a piece. The ability to analyse the occurrences of closely related patterns within a work can provide interesting insight into the compositional treatment of thematic ideas.

There is still a great deal of work to be done in order to improve the quality and reliability of automated music analysis. However, even as it stands, our system opens up some very interesting possibilities for future work. Automated systems cannot currently hope to match the quality of analysis performed by professional musicologists, but do have an advantage of being able to process very large amounts of data. The ability to reliably isolate significant musical patterns and infer basic structural relationships between patterns from across a database of many thousands of pieces of music would create a rich source of musical knowledge, with exciting potential for a range of future research.

References

Bregman, A. S. (1990). *Auditory Scene Analysis: The Perceptual Organization of Sound*. Cambridge, MA, USA: The MIT Press.

Caprara, A., Fischetti, M., & Toth, P. (1998). Algorithms for the set covering problem. Technical report, Technical Report No. OR-98-3. DEIS-Operations Research Group, 1998.

Chvatal, V. (1979). A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, *4*(3), 233–235.

Clarkson, K. L. & Varadarajan, K. (2005). Improved approximation algorithms for geometric set cover. In *SCG '05: Proceedings of the twenty-first annual symposium on Computational geometry*, (pp. 135–141)., New York, NY, USA. ACM.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). *Introduction to Algorithms* (2nd ed.). Cambridge, MA, USA: The MIT Press.

Dreyfus, L. (1996). *Bach and the Patterns of Invention*. Cambridge, MA, USA: Harvard University Press.

Feige, U. (1998). A threshold of ln n for approximating set cover. *Journal of the ACM*, *45*(4), 634–652.

Huron, D. (2006). *Sweet Anticipation: Music and the psychology of expectation*. Cambridge, MA, USA: A Bradford Book MIT Press.

Karp, R. M. (1972). Reducibility among combinatorial problems. In R. Miller & J. Thatcher (Eds.), *Complexity of computer computations* (pp. 85–103). New York, NY, USA: Plenum Press.

Krumhansl, C. (1997). Effects of perceptual organization and musical form on melodic expectancies. In M. Leman (Ed.), *Music, Gestalt, and Computing: Studies in Cognitive and Systematic Musicology*, volume 1317 of *LNAI* (pp. 294–320). Heidelberg, Germany: Springer-Verlag.

Lemström, K. & Pienimäki, A. (2007). On comparing edit distance and geometric frameworks in content-based retrieval of symbolically encoded polyphonic music. *Musicæ Scientiæ*, *Discussion Forum 4A*, 135–152.

Lerdahl, F. & Jackendoff, R. (1983). *A Generative Theory of Tonal Music* (1999 ed.). Cambridge, MA, USA: MIT Press.

Marchiori, E. & Steenbeek, A. (1998). An iterated heuristic algorithm for the set covering problem. In Mehlhorn, K. (Ed.), *Proceedings of the Workshop on Algorithm Engineering*, (pp. 155–166).

Meredith, D. (2006). Point-set algorithms for pattern discovery and pattern matching in music. In Crawford, T. & Veltkamp, R. C. (Eds.), *Proceedings of the Dagstuhl Seminar on Content-Based Retrieval*, number 06171 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl.

Meredith, D., Lemström, K., & Wiggins, G. A. (2002). Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, *31*(4), 321–345.

Meredith, D., Lemström, K., & Wiggins, G. A. (2003). Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. In *Cambridge Music Processing Colloquium 2003*, Cambridge, UK. Department of Engineering, University of Cambridge.

Nattiez, J.-J. (1987). *Musicologie générale et sémiologue (Music and Discourse: Towards a Semiology of Music (1990), trans. by Carolyn Abbate).* Princeton NJ, USA: Princeton University Press.

Ukkonen, E., Lemström, K., & Mäkinen, V. (2003). Sweepline the music! In R. Klein, H.-W. Six, & L. Wegner (Eds.), *Computer Science in Perspective*, volume 2598 of *LNCS* (pp. 330–342). Berlin, Germany.: Springer-Verlag.

Wiggins, G. A., Lemström, K., & Meredith, D. (2002). Sia(m)ese: An algorithm for transposition invariant, polyphonic content-based music retrieval. In Fingerhut, M. (Ed.), *Proceedings of the 3rd International Conference on Music Information Retrieval*, (pp. 283–284).

Yang, J. & Leung, J. Y.-T. (2005). A generalization of the weighted set covering problem. *Naval Research Logistics*, *52*(2), 142–149.