Tom Collins
www.tomcollinsresearch.net

ISMIR Review Examples

May 8, 2015
Last updated April 29, 2019

Introduction

This file contains a summary of the 2019 International Society for Music Information Retrieval (ISMIR) Conference review form, two submissions to the 2013 ISMIR Conference, one review for each submission, and the published version of each paper.[1] **The review for Paper A is too short, the review for Paper B is useful but on the long side. Reviewers are encouraged to adhere to the Goldilocks principle and produce something in between!**

A few questions and factors to consider when reviewing:

1. Does the abstract set out one or more clearly defined objectives?

2. Do you find the objective(s) interesting?

3. Does the paper achieve its stated objective(s)? In particular, please be wary of papers that state X as their objective, evaluate X using methodology Y, but Y doesn't constitute an *appropriate* evaluation of X.

4. If the paper uses technique Z to address problem W, does it reference existing work that addresses W via techniques other than Z? Strong papers will contain a comparative evaluation—or at least discussion—of multiple different techniques for addressing the problem. Weak papers often assume that only Z has ever been or could ever be used.

5. Please be courteous in your review, even if you find the paper unclear or unpalatable. For some submitting to ISMIR, it will be their first experience of the peer-review process, so there is no harm in following up criticisms with suggested amendments or improvements. If you find the research interesting but it's lacking requisite maturity, then it's fine to say something like "This project has potential, but to include the submission in the ISMIR proceedings in its current state would do more harm than good".

6. Consider whether the paper makes good use of the supplementary materials option, and/or (anonymised) links to demos or other sources of information. The blind review process is not an excuse for a paper to base significant claims on inaccessible, external sources.

7. It is not standard policy to sign reviews (it's a blind review process, after all). If you've spent a lot of time reading and reviewing a paper, however, and you want to add your name at the bottom of a review, I'm not going to remove it. I think it encourages a higher standard of reviewing. Beware that there is a negative side to signing reviews: if, ultimately, the paper is rejected, the authors may fairly or unfairly associate your name with this outcome.

Thank you for your time, and happy reviewing!

[1] Thank you to the authors of these papers for agreeing to include their submitted versions in this document.

The Review Form

The review form for the 2019 ISMIR conference contains the following statements/questions:

1. I am an expert in the topic of the paper.
2. The title and abstract reflect the content of the paper.
3. The paper discusses, cites and compares with all relevant related work.
4. The writing and language are clear and structured in a logical manner.
5. The references are well formatted.
6. The topic of the paper is relevant to the ISMIR community.
7. The content is scientifically correct.
8. The paper provides novel methods, findings or results.
9. The paper will have a large influence/impact on the future of the ISMIR community.
10. The paper provides all the necessary details or material to reproduce the results described in the paper.
11. The paper provides reusable insights (i.e. the capacity to gain an accurate and deep understanding). Such insights may go beyond the scope of the paper, domain or application, in order to build up consistent knowledge across the MIR community.
12. Please explain your assessment of reusable insights in the paper.
13. Write ONE line with the main take-home message from the paper (hidden from authors).
14. This paper is of award-winning quality (hidden from authors).
15. If yes, please explain why it should be awarded (hidden from authors).
16. Overall evaluation
17. Main review and comments for the authors
18. Confidential comments for the program committee (hidden from authors)

I tend to write detailed comments offline (Q17, maybe Q18), and then go to complete the online form. These two questions plus Q12 are important for substantiating the ratings that you give for the statements in Q1-11: if your detailed review contains no mention of *novelty* (Q8), for instance, then the Program Committee may not pay much attention to your rating for the *novelty* statement. Q1-11 have answer options "Strongly disagree", "Disagree", "Agree", and "Strongly agree". Q16 has the option "Strong reject", "Weak reject", "'Weak accept", and "Strong accept".

In case it is not clear, Q18 is resevrved for writing confidential comments to the Program Committee (the authors will not see these comments). This is where you can vent about a very badly written paper, or alternatively use Q14 and 15 to praise a very well written paper. Remember, your detailed comments to the authors should always remain courteous and measured.

Paper A: Submission...

SIARCT-CFP: IMPROVING PRECISION AND THE DISCOVERY OF INEXACT MUSICAL PATTERNS IN POINT-SET REPRESENTATIONS

First author
Affiliation1
author1@ismir.edu

Second author
Retain these fake authors in
submission to preserve the formatting

Third author
Affiliation3
author3@ismir.edu

ABSTRACT

The geometric approach to intra-opus pattern discovery (in which notes are represented as points in pitch-time space in order to discover repeated patterns within a piece of music) shows promise particularly for polyphonic music, but has attracted some criticism because: (1) the approach extends to a limited number of inexact repetition types only; (2) typically geometric pattern discovery algorithms have poor precision, returning many false-positives. This paper describes and evaluates a solution to the *inexactness problem* where algorithms for pattern discovery and inexact pattern matching are integrated for the first time. Two complementary solutions are proposed and assessed for the *precision problem*, one involving categorisation (hence reduction) of output patterns, and the second involving a new algorithm that calculates the difference between consecutive point pairs only, rather than all point pairs.

1. INTRODUCTION

The discovery of repeated patterns within a piece of music is an activity that manifests itself in a range of disciplines. In music psychology, for example, listeners' emotional responses to a piece exhibit distinctive behaviour at the beginning of repeated sections [11]. In music analysis, an awareness of the locations of motifs, themes, and sections, and their relation to one another, is a prerequisite for writing about the construction of a piece [3]. Last but not least, in music computing, algorithmic pattern discovery can be used to define compressed representations [13] (e.g., the numeric pitch sequence 67, 68, 67, 69, 69, 66, 67, 66, 68, 68 can be encoded as 67, 68, 67, 69, 69, and a translation operation "-1") and can act as a guide for the algorithmic generation of new music [9]. In the interests of supporting these multiple manifestations, it is important that the field of music information retrieval continues to develop and refine algorithms for the discovery of repeated patterns, and continues to evaluate these against each other and human-annotated ground truths.

There are two main representations in use for discovering repeated patterns within a piece of music (hereafter *intra-opus discovery* [8]): (1) *viewpoints* [9] involve encoding multiple aspects of the music as strings of symbols (such as the numeric pitches mentioned above, or durations, intervals between notes, etc.). This approach has been applied mainly to monophonic music; (2) the *geometric approach* [14] involves converting each note to a point in pitch-time space (see the pitch-time pairs in Figures 1A and B). Higher-dimensional spaces are also possible (e.g., including dimensions for duration or staff number). The geometric approach is well-suited to handling polyphonic music, where few attempts have been made to apply viewpoints. This paper focuses on the geometric approach; specifically, ontime and *morphetic pitch number* [14] (C4 = C♯4 = 60, D♭4 = D♮4 = D♯4 = 61, E♭4 = E4 = 62, etc.).

Before getting into more details of related work, it is helpful to distinguish the terms *pattern matching* and *pattern discovery*. Typically in pattern matching, there is a short musical query and a longer piece (or pieces) of music, and the aim is to match the query to more or less exact instances in the piece(s) [2, 16]. In intra-opus pattern discovery there is no query, just a single piece of music, and the requirement to discover motifs, themes, and sections that are repeated within the piece [8, 14]. (One could say that the purpose of a pattern discovery algorithm is to *create* analytically interesting but hitherto unknown queries.) Pattern discovery and pattern matching have been discussed in the same papers [13], but nobody to our knowledge has integrated discovery and inexact matching components in one algorithm before. This full integration is one of the contributions of the current work, and the other consists of two complementary methods for improving the precision of pattern discovery algorithms. The paper is organised around describing and evaluating components of a new algorithm called SIARCT-CFP, beginning at the end of the acronym with "FP" for fingerprinting, then "C" for categorisation, and finally SIARCT, which stands for Structure Induction Algorithm for r superdiagonals and Compactness Trawler, which has been defined before [5] and for which a Matlab implementation has been released. [1]

2. THE INEXACTNESS PROBLEM

In reviewing the Structure Induction Algorithm (SIA) and other geometric pattern discovery algorithms (see [14] or [7] for details), Lartillot and Toiviainen noted that "this

[1] A URL for the code will go here after review.

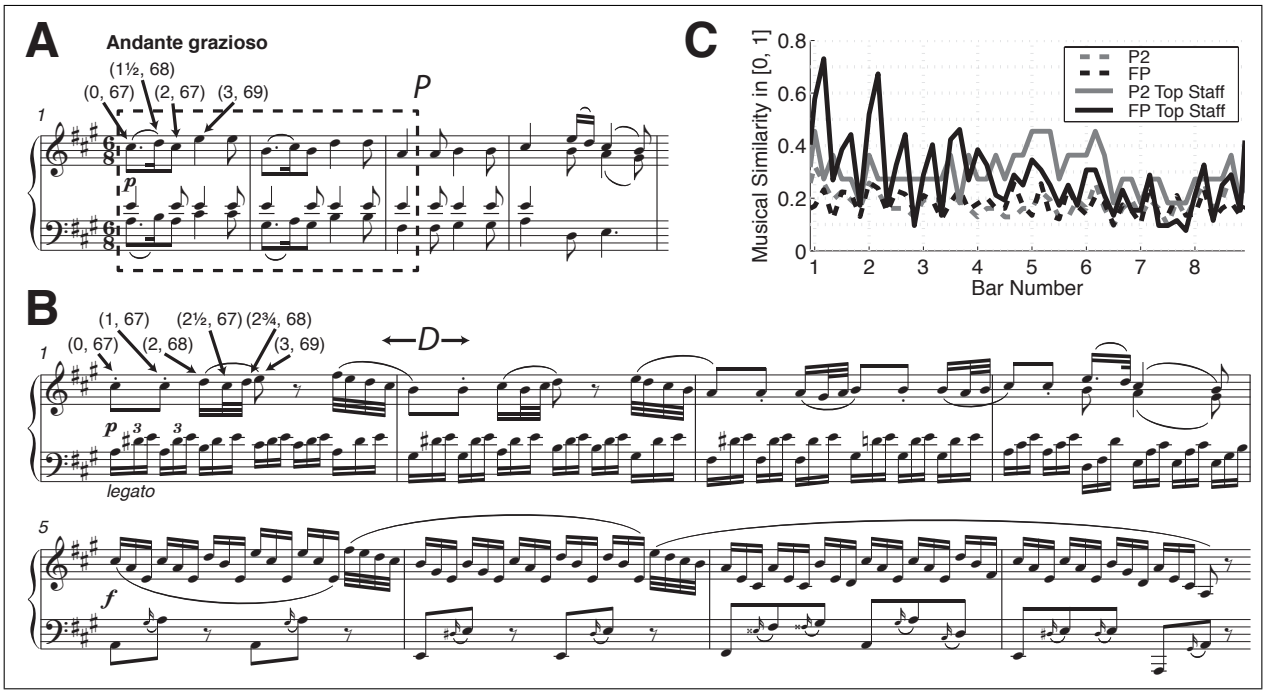

Figure 1. (A) Bars 1-4 of the Theme from the first movement of the Piano Sonata in A major K.331 by Wolfgang Amadeus Mozart (1756-1791). Labels give the ontime and morphetic pitch of the indicated note, and the box contains the top-rated pattern output by SIARCT; (B) Bars 1-8 of Variation II from the same movement; (C) Symbolic musical similarity of the pattern in (A) to the passage in (B), for two algorithms applied separately to the full texture and top staff only.

geometrical strategy did not apply to melodic repetitions that presented rhythmic variations" [10, pp. 290-291]. If the passage in Figure 1B were appended to the passage in Figure 1A for instance and SIA applied to the single resulting point set, there would be little in the output to suggest that the first two bars of Figure 1B contain a variation of the bounded pattern P in Figure 1A. The points $\{(0, 67), (3, 69), (6, 66), (9, 68), (12, 65)\}$ would appear in the same output maximal translatable pattern (MTP, [14]), as they are translated by the same amount in Figure 1B, but intervening points in the bounded pattern are not.

The pattern matching algorithm P2 [16] struggles with rhythmic variation also: for a given pattern P and a larger point set D, it returns all vector-frequency pairs $(\mathbf{w}, m)$ such that $m \geq 1$ points of P occur translated by $\mathbf{w}$ in D. We implemented P2 and used it to match P (from Figure 1A) to partial occurrences in D (Figure 1B). A summary of the output is plotted in Figure 1C, for both full-texture versions of P and D and a restriction to the right hand only (dashed and solid lines respectively). The maximal frequency M for pairs $(\mathbf{w}_1, m_1)_{i \in \{1,2,\ldots,s\}}$ corresponding to each quaver-note ontime in D is plotted, normalised by the number of points in P, to give a measure of the symbolic musical similarity of P to D over time. While there are local maxima in the grey lines at bars 1, 5, and 6 (in the latter case because P2 is transposition-invariant and there is a transposed pattern within P), in general they have a relatively small range, reflecting P2's struggle to distinguish genuine rhythmic variation from less related material.

Subsequent work on geometric pattern matching improves upon P2 in terms of capturing rhythmic variation, by representing durations as line segments [12, 16], by using the Hausdorff metric [15], or by converting to a tonal space representation [1]. A recent *fingerprinting* (FP) approach [2] has the advantage of not relying on durational information, and has options for transposition, time-shift, and scale-factor invariance, as well as tolerance for the percentage by which the inter-onset interval of a pair of notes is permitted to differ, compared to a corresponding note pair in the original. The output of FP is a time series $S = S_t : t \in T$, where the set T of successive time points may or may not be uniformly spaced. The magnitude of S_t, called the *matching score*, indicates the extent to which an occurrence of the query begins at time t. In the transposition-invariant version, calculation of the matching score time series begins by creating fingerprint tokens

$$[y_j - y_i,\ x_j - x_i],\ \ t, \tag{1}$$

for locally constrained combinations of successive ontime-pitch pairs $(x_i, y_i), (x_j, y_j)$, in both a query pattern P and the larger point set D. The pair in brackets in (1) is the hash key, and $t = x_i$ is a time stamp. A scatter plot of the time stamps of matching hash keys for P and D can be used to identify regions of high similarity, which appear as approximately diagonal lines. The matching score is calculated by applying an affine transformation to the scatter plot and binning (for details, see [2, 17]).

An implementation of the FP algorithm was used to match exact/inexact occurrences of P from Figure 1A to D in Figure 1B, and the results are plotted in Figure 1C as black lines. It can be seen that FP outperforms P2 at distinguishing the rhythmic variation in bars 1-2 of Figure 1B.

The use of locally constrained combinations of ontime-pitch pairs, rather than one candidate translation vector applied to all points in P, is what enables the FP algorithm to find a stronger match than P2.

Progress has been made in geometric pattern *matching* techniques, but Lartillot and Toiviainen's [10] criticism of the *discovery* approach still stands, as nobody to our knowledge has integrated an inexact matching technique within a pattern discovery approach. We do so now, according to the following steps:

1. Run SIARCT [5] on a given point set D, returning patterns $P_1, P_2, \ldots, P_M$, each of which has at least one translationally exact repetition (i.e., two occurrences) in D;

2. For $i = 1, 2, \ldots, M$, run the FP algorithm [2] on P_i and D, returning time points $t_1^{P_i}, t_2^{P_i}, \ldots, t_m^{P_i}$ at which there may be further exact/inexact occurrences of P_i, according to whether the value at $t_j^{P_i}$ is greater than some *similarity threshold* $c \in [0, 1)$.

Underlying this integration of pattern discovery and pattern matching is the following assumption, which we call the *translationally exact once* (TEO) hypothesis:

> If a piece of music contains multiple inexact occurrences of a perceptually salient or analytically interesting pattern, then for some majority subset of the pattern (i.e., a subset containing at least half of the points), there exists at least one translationally exact repetition (i.e., at least two occurrences).

If the discovery algorithm outputs a majority subset, then the matching algorithm may be relied upon to output further exact/inexact occurrences.

As a case study, the new algorithm defined above (called SIARCT-CFP) was run on the Nocturne in E major op.62 no.2 by Frédéric Chopin (1810-1849).[2] This is a sensible choice of piece, as there are multiple varied repetitions of the opening four-bar theme (c.f. Figures 2B and C for instance). Fourteen patterns were output in total, one of which Q is bounded in Figure 2A, and occurs translated exactly twice (Figure 2D, bar 27, and Figure 2E, bar 58). It can be seen that these occurrences are rated as identical or very similar to Q, with normalised matching scores of 1 and .963 respectively. The time series output by the FP has mean .267 and standard deviation .160, suggesting that the occurrence in Figure 2C is barely distinguishable from other unrelated material. This makes sense, as although the contour and durations of the melody are the same as in Q, the pitch intervals are different (see arrows) and so is the accompaniment. We note, however, that the FP algorithm could be extended further to incorporate contour (up, down, same), as well as other viewpoints [9], because of its use of locally constrained comparisons.

[2] The first part of the algorithm, SIAR, ran with parameter $r = 1$. Second, the compactness trawler (CT) ran with compactness threshold $a = 4/5$, cardinality threshold 10, and lexicographic region type [7]. Third, the categorising and fingerprinting (CFP) ran with similarity threshold $c = 1/2$.

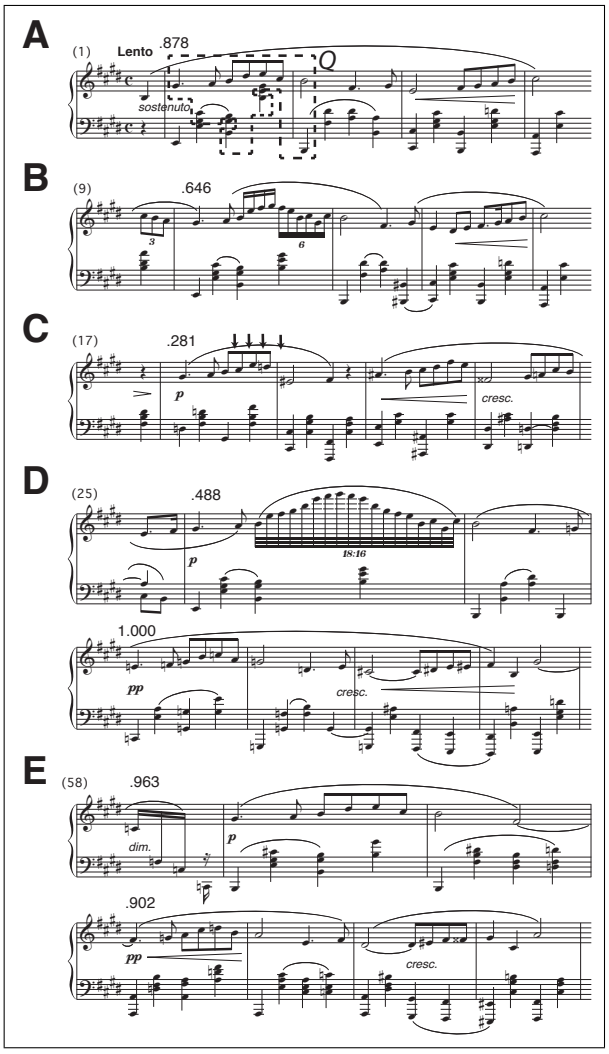

Figure 2. Excerpts from the Nocturne in E major op.62 no.2 by Chopin. Dashed lines in (A) bound a pattern Q discovered by SIARCT, which is used to match other inexact occurrences, with degree of exactness indicated in the figure by numbers in [0, 1]. Pedalling omitted for clarity.

3. THE PRECISION PROBLEM

3.1 Categorisation by Pattern Matching

Now that we have integrated some inexact pattern matching techniques into our pattern discovery approach, it is possible to employ them for the purposes of categorisation, based on the idea that P2 [16] or FP [2] can be used to compare two discovered patterns P_i and P_j in exactly the same way as if $P_i = P$ was a query and $P_j = D$ was a point set (or vice versa, as the measures are symmetric).

The second "C" in SIARCT-CFP stands for a **c**ategorisation process, which will be described now. The purpose of categorisation is to reduce an overwhelming amount of information (e.g., output patterns) to a more manageable number of exemplars. Here *categorisation* does not mean classifying patterns into an accepted/interesting category versus a rejected/uninteresting category; rather it means grouping similar patterns and representing each group with one exemplar pattern. Our motivation for categorising the

output of SIARCT is to improve its precision: while the precision and recall of pattern discovery algorithms has been shown to benefit from compactness trawling, the precision is still quite poor [7]. For example, SIARCT outputs 76 patterns when run on Chopin's op.62 no.2, which can be reduced to fourteen patterns by using the following categorisation process:

1. Let $\mathcal{P} = \{P_1, P_2, \ldots, P_M\}$ be the patterns output by SIARCT or some other algorithm, sorted descending by a rating of perceived pattern importance [5] (or some other ordering). Let $J = \{1, 2, \ldots, M\}$ index the patterns that are uncategorised currently;

2. Let $i = \min(J)$ and calculate the maximum normalised matching scores $s(P_i, P_j)$ for each $j \in J$ such that $j > i$;

3. For each similarity score $s(P_i, P_j)$ that is greater than some specifiable similarity threshold $c \in [0, 1)$, place pattern P_j in the category for which P_i is the exemplar, and remove j from J;

4. Repeat steps 2 and 3 until either J has one element k, in which case define P_k to be an exemplar with category membership P_k, or otherwise J is empty;

5. For the purposes of algorithm evaluation, return only the exemplars $P_{i(1)}, P_{i(2)}, \ldots, P_{i(m)}$.

Depending on the choice of c, $m \ll M$. The categorisation process can be visualised with two similarity matrices (Figure 3). The matrix in Figure 3A contains the maximum normalised matching scores for each pair of 76 output patterns for Chopin's op.62 no.2, ordered as in step 1 above. The matrix in Figure 3B is a permutation of Figure 3A, showing the categorised patterns ($c = .5$) in their fourteen categories, bounded by white squares. The third square from top-left in Figure 3B represents the category for which Q in Figure 2A is the exemplar. The more than fivefold ($5.43 \approx 76/14$) reduction in output achieved by pattern-matching categorisation may well improve precision: as discussed, the theme annotated in Figure 2A survives the categorisation process, and so do all of the repetitions in this piece lasting four or more bars (results not shown). Pattern-matching categorisation also constitutes a novel and interesting use of the FP algorithm [2]. It should be noted that choosing too high a value of c could lead to over-reduction and filtering out of analytically interesting patterns. For instance, the first three squares in Figure 3B show considerable variegation, suggesting that some interesting subcategories may be overlooked.

3.2 Consecutive Points and Conjugate Patterns

The final novel contribution of this paper is to evaluate the SIARCT pattern discovery algorithm [5] against a collection of music containing repeated sections, and to compare its performance (especially precision) to SIA [14] and SIAR [5]. SIA outputs thousands of patterns for Chopin's op.62 no.2 (and other pieces of music [7]), so it is necessary to develop a more parsimonious pattern discovery

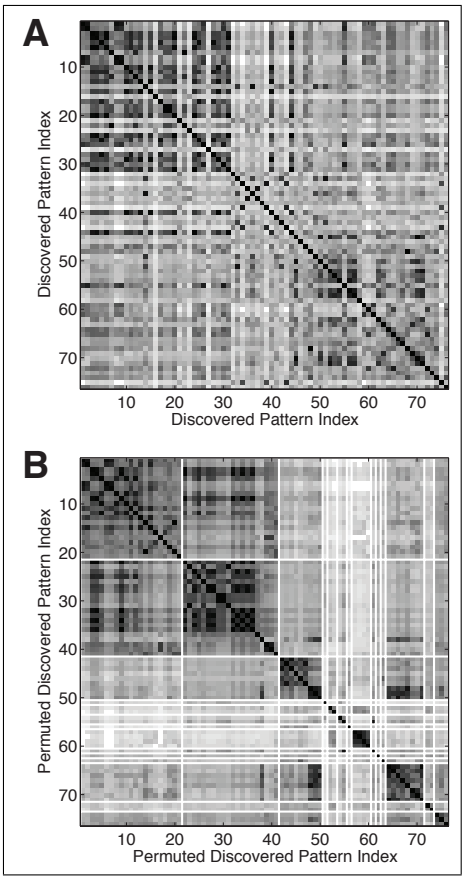

Figure 3. (A) Pairwise symbolic musical similarities (ranging from white for dissimilar to black for identical) for 76 patterns discovered by SIARCT in Chopin's op.62 no.2, ordered by a rating formula for perceived salience; (B) Permutation of the above matrix, with white lines indicating the results of categorising into fourteen groups.

algorithm for use as input to the categorisation and fingerprinting components described above (e.g., SIARCT outputs only 76 patterns for Chopin's op.62 no.2).

It has long been thought that in order to discover repeated patterns within a geometric representation D of a piece, it is necessary to calculate the difference between each pair of n points (giving $n[n-1]/2$ calculations in total), as in SIA [14]. Unlike SIA, the first step of SIARCT is to calculate the difference between consecutive pairs of points only ($n - 1$ calculations). Some exhaustive pairwise comparisons are still made in the second step, but for non-overlapping and typically small subsets of D, meaning that the total number of difference calculations performed by SIARCT is far less than $n[n-1]/2$, in all but one degenerate case. [3] The third step of SIARCT makes use of a concept known as *conjugate patterns* [5]: if a pattern containing l points occurs m times in a point set, then there exists in the same point set a pattern consisting of m points that occurs l times. The fourth step calculates MTPs for each vector in a list L. As a consequence of manipulating conjugate patterns, the vectors corresponding to repeated sections should be at or near the top of L. So for this step

[3] Please see [5] for the algorithmic details.

we could: (1) distribute each MTP calculation to parallel processors, and/or; (2) output MTPs dynamically for the user to browse, whilst calculation of the remaining MTPs continues. The main claim is that SIARCT will have much smaller output than SIA, with minimal or no negative impact on its performance as measured by precision, recall, and robust versions of these metrics [4]. The compactness trawler (CT) part of SIARCT is exactly as in [7], so is not addressed again here.

SIA, SIAR, and SIARCT were run on point-set representations of movements by Ludwig van Beethoven (1770-1827) and Chopin listed in Figure 4A. SIARCT ran with compactness threshold $a = 1$, and points threshold $b = 50$. This means that only patterns containing 50 points or more were returned, and they had to have maximal compactness of 1. The parameter values make sense in terms of trying to discover repeated sections. To make the evaluation fair, we also filtered the results of SIA and SIAR, returning only those patterns that contained 50 points or more. In the results, these versions of SIA and SIAR are referred to as SIA (50+) and SIAR (50+).

3.3 Evaluation Results

Figure 4B shows the log of the total number of patterns output by each algorithm for each movement/piece. It supports the claim that SIAR has a much smaller output than SIA. It is difficult to see from Figure 4B, but the same observation applies to the filtered versions of each algorithm, SIAR (50+) and SIA (50+). The number of patterns output by SIARCT is several orders of magnitude less than that of any other algorithm. Figure 4C and Figure 4E show that compared with SIA's performance, SIAR is not negatively impacted by restricting calculations to consecutive pairs of points. The establishment precision and establishment recall are higher for SIAR than for SIA across all pieces.

Overall, the most effective algorithm is SIARCT (see Figure 4C and Figure 4E). For half of the pieces, it discovers all ground truth patterns exactly (Figure 4F). When SIARCT fails to discover a ground truth pattern exactly, often this is due to a difference between the repeated section as written in the score, and the repeated pattern as heard in a performance. For instance, in the fourth movement of Beethoven's op.7, bars 65–70 are marked as a repeated section, and this is included in the ground truth. The repeated notes extend beyond these bars in both directions, however, creating a longer repeated pattern in a performance. SIARCT discovers the latter, 'performed' pattern, which reduces exact precision and recall. The more robust *establishment* metrics are not much reduced (e.g., see Figure 4E), and arguably discovering the performed pattern is preferable from a music-perceptual point of view.

4. DISCUSSION AND FUTURE WORK

This paper identifies two valid reasons why the geometric approach to intra-opus pattern discovery has attracted some criticism—namely (1) the approach extends to a limited number of inexact repetition types only, and (2) typically geometric pattern discovery algorithms are imprecise, returning many false-positives results. A new algorithm called SIARCT-CFP was then described and evaluated component by component, in an attempt to address these criticisms. It is the first geometric pattern discovery algorithm to fully integrate an inexact pattern matching component (the fingerprinting algorithm of [2]), and this matching component was shown to be effective for retrieving more or less exact occurrences of themes in pieces by Mozart and Chopin. The comparison of the FP algorithm [2] to a baseline pattern matching algorithm P2 [16] demonstrated that the former was superior for a particular example. In general it may be preferable to have two or more pattern-matching algorithms at one's disposal, however, as the number of variation techniques is large, and trying to account for them all with one algorithm will likely produce false-positive matches.

The precision metrics were of particular interest to us in the comparative evaluation of SIARCT [5], SIAR [5], and SIA [14], as it had been claimed that SIARCT could achieve superior levels of precision compared to SIA and SIAR, without harming recall. This claim was supported by the evaluation results, although in future work it will be necessary to see if similar results are achieved for ground truths containing shorter patterns than repeated sections.

Our *translationally exact once* (TEO) hypothesis (see Section 2) was borne out in the case study of Chopin's op.62 no.2, where Q (Figure 2A) occurred translated exactly twice, at bars 27 (Figure 2D) and 58 (Figure 2E). The contents of Q were sufficient for use as a query to retrieve less exact versions such as in bars 9 (Figure 2B) and 25 (Figure 2D). For the case study of the Theme section and Variation II from Mozart's K331, SIARCT was able to discover perceptually salient patterns such as P in Figure 1A, which recurs in bars 5-7 of the Theme section (not shown). As the TEO hypothesis holds in both cases, it would be worthwhile in future work to try to find counterexample pieces, as this will help to refine and improve our underlying assumptions and ensuing algorithms. Future work will also attempt to show users/developers the differences between themes and partial matches, and to identify variation techniques (triplets, minore, etc.) automatically.

5. ACKNOWLEDGEMENTS

To be included after review.

6. REFERENCES

[1] T. Ahonen, K. Lemstr om, and S. Linkola: "Compression-based similarity measures in symbolic, polyphonic music," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 91–96, 2011.

[2] A. Arzt, S. Böck, and G. Widmer: "Fast identification of piece and score position via symbolic fingerprinting," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 433–438, 2012.

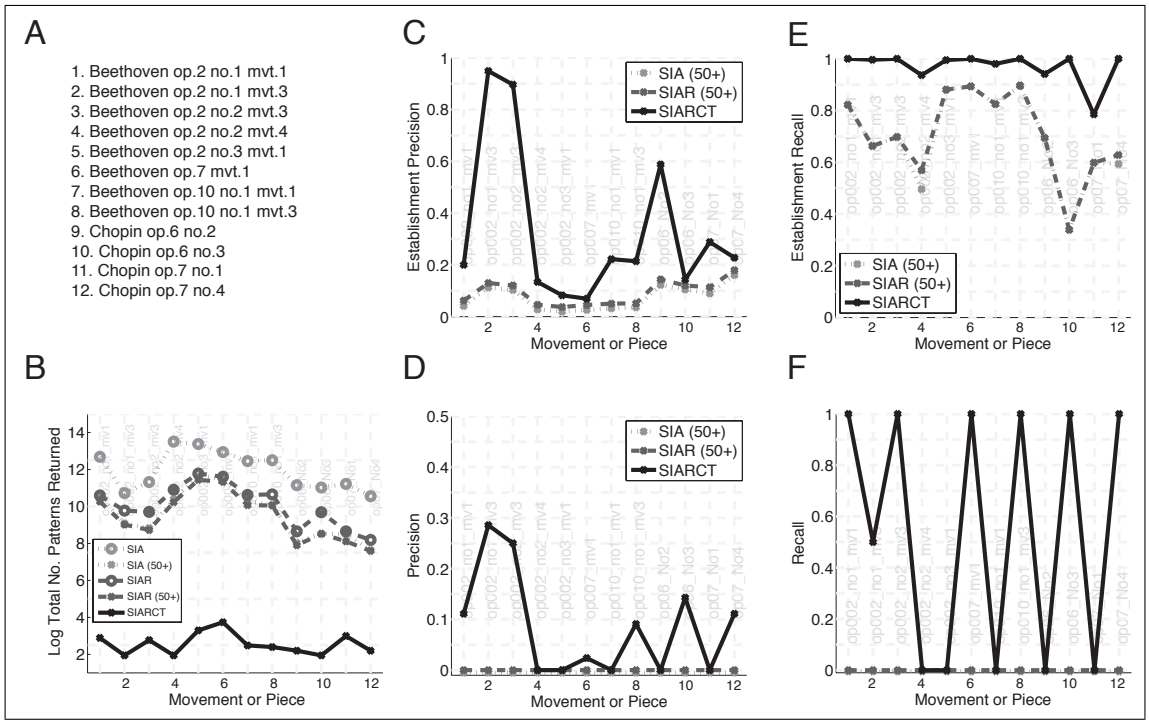

Figure 4. Evaluation metrics for three algorithms, run on eight movements by Beethoven and four pieces by Chopin.

[3] I. Bent and A. Pople: "Analysis," *The new Grove dictionary of music and musicians*, Macmillan, London, 2001.

[4] T. Collins: "Discovery of repeated themes and sections," Retrieved 4th May 2013, from http://www.music-ir.org/mirex/wiki/2013: Discovery_of_Repeated_Themes_%26_Sections

[5] T. Collins: *Improved methods for pattern discovery in music, with applications in automated stylistic composition*, PhD thesis, Faculty of Mathematics, Computing and Technology, The Open University, 2011.

[6] T. Collins, R. Laney, A. Willis, and P. Garthwaite: "Modeling pattern importance in Chopin's mazurkas," *Music Perception*, Vol. 28, No. 4, pp. 387–414, 2011.

[7] T. Collins, J. Thurlow, R. Laney, A. Willis, and P. Garthwaite: "A Comparative Evaluation of Algorithms for Discovering Translational Patterns in Baroque Keyboard Works," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 3–8, 2010.

[8] D. Conklin and C. Anagnostopoulou: "Representation and discovery of multiple viewpoint patterns," *Proceedings of the International Computer Music Conference*, pp. 479–485, 2001.

[9] D. Conklin and I. Witten: "Multiple viewpoint systems for music prediction," *Journal of New Music Research*, Vol. 24, No. 1, pp. 51-73, 1995.

[10] O. Lartillot and P. Toiviainen: "Motivic matching strategies for automated pattern extraction," *Musicae Scientiae*, Discussion Forum 4A, pp. 281–314, 2007.

[11] S. Livingstone, C. Palmer, and E. Schubert: "Emotional response to musical repetition," *Emotion*, Vol. 12, No. 3, pp. 552–567, 2012.

[12] A. Lubiw and L. Tanur: "Pattern matching in polyphonic music as a weighted geometric translation problem," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 154–161, 2004.

[13] D. Meredith: "Point-set algorithms for pattern discovery and pattern matching in music," *Proceedings of the Dagstuhl Seminar on Content-Based Retrieval*, 2006.

[14] D. Meredith, K. Lemström, and G. Wiggins: "Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music," *Journal of New Music Research*, Vol. 31, No. 4, pp. 321–345, 2002.

[15] C. Romming and E. Selfridge-Field: "Algorithms for polyphonic music retrieval: the Hausdorff metric and geometric hashing," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 457–462, 2007.

[16] E. Ukkonen, K. Lemström, and V. Mäkinen: "Geometric algorithms for transposition invariant content-based music retrieval," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 193–199, 2003.

[17] A. Wang: "An industrial strength audio search algorithm," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 7–13, 2003.

Paper A: Review

This paper presents some extensions to methods for pattern discovery within musical works. It is well written and accessible to the novice (me). I really have very little criticism, the algorithmic steps are well laid out, and the prose quite clear. As far as interest, this is fairly far from my area, but the examples were compelling, and I found the paper interesting overall. Methodologies appear sound, aspects of the algorithm are explored and evaluated separately. It should provide interest to those working in this area.

Paper A: Published Version...

SIARCT-CFP: IMPROVING PRECISION AND THE DISCOVERY OF INEXACT MUSICAL PATTERNS IN POINT-SET REPRESENTATIONS

Tom Collins, Andreas Arzt, Sebastian Flossmann, and Gerhard Widmer
Department of Computational Perception, Johannes Kepler University Linz
{tom.collins, andreas.arzt, sebastian.flossmann, gerhard.widmer}@jku.at

ABSTRACT

The geometric approach to intra-opus pattern discovery (in which notes are represented as points in pitch-time space in order to discover repeated patterns within a piece of music) shows promise particularly for polyphonic music, but has attracted some criticism because: (1) the approach extends to a limited number of inexact repetition types only; (2) typically geometric pattern discovery algorithms have poor precision, returning many false positives. This paper describes and evaluates a solution to the *inexactness problem* where algorithms for pattern discovery and inexact pattern matching are integrated for the first time. Two complementary solutions are proposed and assessed for the *precision problem*, one involving categorisation (hence reduction) of output patterns, and the second involving a new algorithm that calculates the difference between consecutive point pairs, rather than all point pairs.

1. INTRODUCTION

The discovery of repeated patterns within a piece of music is an activity that manifests itself in a range of disciplines. In music psychology, for example, listeners' emotional responses to a piece exhibit distinctive behaviour at the beginning of repeated sections [11]. In music analysis, an awareness of the locations of motifs, themes, and sections, and their relation to one another, is a prerequisite for writing about the construction of a piece [3]. Last but not least, in music computing, algorithmic pattern discovery can be used to define compressed representations [13] (e.g., the numeric pitch sequence 67, 68, 67, 69, 69, 66, 67, 66, 68, 68 can be encoded as 67, 68, 67, 69, 69, and a translation operation "-1") and can act as a guide for the algorithmic generation of new music [9]. In the interests of supporting these multiple manifestations, it is important that the field of music information retrieval continues to develop and refine algorithms for the discovery of repeated patterns, and continues to evaluate these against each other and human-annotated ground truths.

There are two main representations in use for discovering repeated patterns within a piece of music (hereafter *intra-opus discovery* [8]): (1) *viewpoints* [9] involve encoding multiple aspects of the music as strings of symbols (such as the numeric pitches mentioned above, or durations, intervals between notes, etc.). This approach has been applied mainly to monophonic music; (2) the *geometric approach* [14] involves converting each note to a point in pitch-time space (see the pitch-time pairs in Figures 1A and B). Higher-dimensional spaces are also possible (e.g., including dimensions for duration or staff number). The geometric approach is well-suited to handling polyphonic music, where few attempts have been made to apply viewpoints. This paper focuses on the geometric approach; specifically, ontime and *morphetic pitch number* [14] (C4 = C♯4 = 60, D♭4 = D♮4 = D♯4 = 61, E♭4 = E4 = 62, etc.).

Before getting into more details of related work, it is helpful to distinguish the terms *pattern matching* and *pattern discovery*. Typically in pattern matching, there is a short musical query and a longer piece (or pieces) of music, and the aim is to match the query to more or less exact instances in the piece(s) [2, 17]. In intra-opus pattern discovery there is no query, just a single piece of music, and the requirement to discover motifs, themes, and sections that are repeated within the piece [8, 14]. (One could say that the purpose of a pattern discovery algorithm is to *create* analytically interesting but hitherto unknown queries.) Pattern discovery and pattern matching have been discussed in the same papers [13], but nobody to our knowledge has integrated discovery and *inexact* matching components in one algorithm before. This full integration is one of the contributions of the current work, and the other consists of two complementary methods for improving the precision of pattern discovery algorithms. The paper is organised around describing and evaluating components of a new algorithm called SIARCT-CFP, beginning at the end of the acronym with "FP" for fingerprinting, then "C" for categorisation, and finally SIARCT, which stands for Structure Induction Algorithm for r superdiagonals and Compactness Trawler, which has been defined before [5] and for which a Matlab implementation has been released. [1]

2. THE INEXACTNESS PROBLEM

In reviewing the Structure Induction Algorithm (SIA) and other geometric pattern discovery algorithms (see [14] or [7] for details), Lartillot and Toiviainen noted that "this ge-

[1] http://www.tomcollinsresearch.net

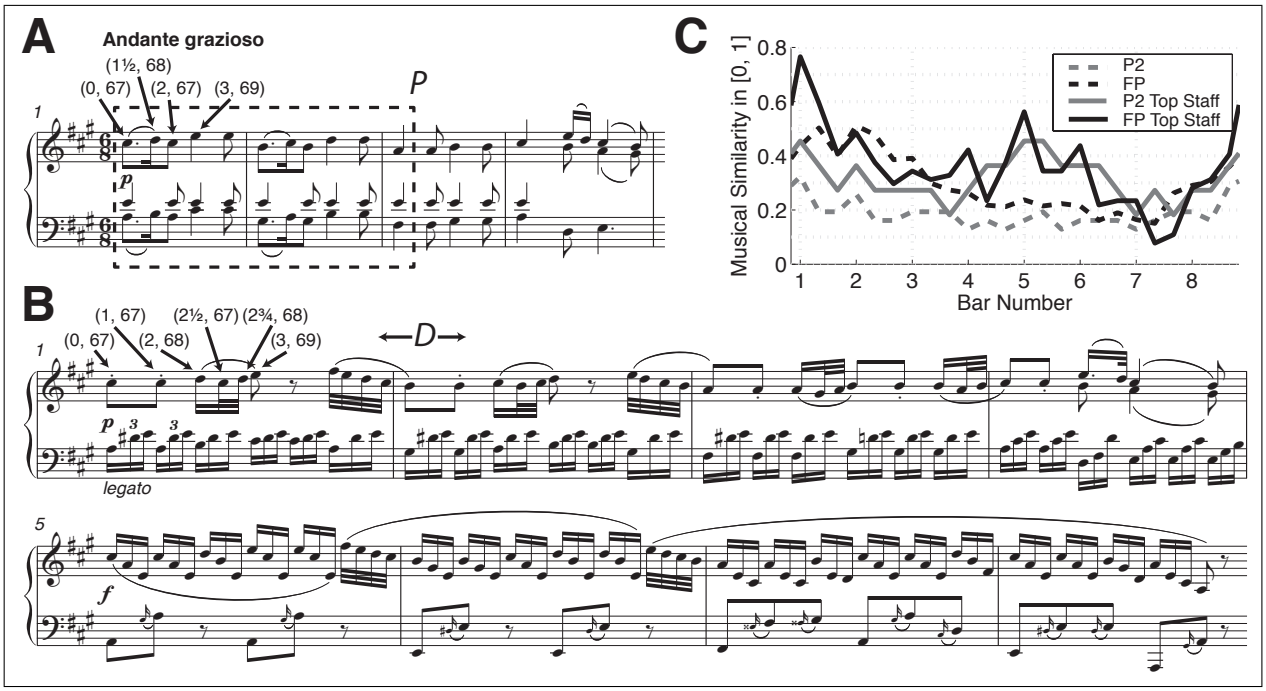

Figure 1. (A) Bars 1-4 of the Theme from the first movement of Piano Sonata no.11 in A major K331 by Wolfgang Amadeus Mozart (1756–1791). Labels give the ontime and morphetic pitch of the indicated note, and the box contains the top-rated pattern output by SIARCT; (B) Bars 1-8 of Variation II from the same movement; (C) Symbolic musical similarity of the pattern in (A) to the passage in (B), for two algorithms applied separately to the full texture and top staff only.

ometrical strategy did not apply to melodic repetitions that presented rhythmic variations" [10, pp. 290-291]. To illustrate this problem we use a theme by Mozart, from "one of the most overanalyzed pieces in the history of music theory" [15, p. 160]. We are not particularly interested in adding to discussions of the structure of the theme itself, rather in the relation of the theme to a subsequent variation. If the passage in Figure 1B were appended to the passage in Figure 1A and SIA applied to the single resulting point set, there would be little in the output to suggest that the first two bars of Figure 1B contain a variation on the bounded pattern P in Figure 1A. The points $\{(0, 67), (3, 69), (6, 66), (9, 68), (12, 65)\}$ would appear in the same output maximal translatable pattern (MTP, [14]), as they occur under the same translation in Figure 1B, but intervening points in the bounded pattern do not.

The pattern matching algorithm P2 [17] struggles with rhythmic variation also: for a given pattern P and a larger point set D, it returns all vector-frequency pairs $(\mathbf{w}, m)$ such that $m \geq 1$ points of P occur translated by $\mathbf{w}$ in D. We implemented P2 and used it to match P (from Figure 1A) to partial occurrences in D (Figure 1B). A summary of the output is plotted in Figure 1C, for both full-texture versions of P and D and a restriction to the right hand only (dashed and solid lines respectively). The maximal frequency M for pairs $(\mathbf{w}_1, m_1)_{i \in \{1,2,\ldots,s\}}$ corresponding to each crotchet-note ontime in D is plotted, normalised by the number of points in P, to give a measure of the symbolic musical similarity of P to D over time. While there are local maxima in the grey lines at bars 1, 2, and 5 (in the second case because P2 is transposition-invariant and there is a transposed pattern within P), in general they have a relatively small range, reflecting P2's struggle to distinguish genuine rhythmic variation from less related material.

Subsequent work on geometric pattern matching improves upon P2 in terms of capturing rhythmic variation, by representing durations as line segments [12, 17], by using the Hausdorff metric [16], or by converting to a tonal space representation [1]. A recent *fingerprinting* (FP) approach [2] has the advantage of not relying on durational information, and has options for transposition, time-shift, and scale-factor invariance, as well as tolerance for the amount by which the inter-onset interval of a pair of notes is permitted to differ, compared to a corresponding note pair in the original. The output of FP is a time series $S = S_t : t \in T$, where the set T of successive time points may or may not be uniformly spaced. The magnitude of S_t, called the *matching score*, indicates the extent to which an occurrence of the query begins at time t. In the transposition-invariant version, calculation of the matching score time series begins by creating fingerprint tokens

$$[y_j - y_i,\ x_j - x_i],\ \ t, \tag{1}$$

for locally constrained combinations of successive ontime-pitch pairs $(x_i, y_i), (x_j, y_j)$, in both a query pattern P and the larger point set D. The pair in brackets in (1) is the hash key, and $t = x_i$ is a time stamp. A scatter plot of the time stamps of matching hash keys for P and D can be used to identify regions of high similarity, which appear as approximately diagonal lines. The matching score is calculated by applying an affine transformation to the scatter plot and binning (for details, see [2, 18]).

An implementation of the FP algorithm was used to match exact/inexact occurrences of P from Figure 1A to D in Figure 1B, and the results are plotted in Figure 1C as black lines. It can be seen that FP outperforms P2 at distinguishing the rhythmic variation in bars 1-2 of Figure 1B. The use of locally constrained combinations of ontime-pitch pairs, rather than one candidate translation vector applied to all points in P, is what enables the FP algorithm to find a stronger match than P2.

Progress has been made in geometric pattern *matching* techniques, but Lartillot and Toiviainen's [10] criticism of the *discovery* approach still stands, as nobody to our knowledge has integrated an inexact matching technique within a pattern discovery approach. We do so now, according to the following steps, which define the "FP" part of SIARCT-CFP:

1. Let $P_1, P_2, \ldots, P_M$ be the output of a pattern discovery algorithm, each P_i having at least one translationally exact repetition (two occurrences) in D;

2. For $i = 1, 2, \ldots, M$, run the FP algorithm [2] on P_i and D, returning time points $t_1^{P_i}, t_2^{P_i}, \ldots, t_m^{P_i}$ at which there may be further exact/inexact occurrences of P_i, according to whether the value at $t_j^{P_i}$ is greater than some *similarity threshold* $c \in [0, 1)$.

Underlying this integration of pattern discovery and pattern matching is the following assumption, which we call the *translationally exact once* (TEO) hypothesis:

> If a piece of music contains multiple inexact occurrences of a perceptually salient or analytically interesting pattern, then for some majority subset of the pattern (i.e., a subset containing at least half of the points), there exists at least one translationally exact repetition (i.e., at least two occurrences).

If the discovery algorithm outputs such a majority subset, then the matching algorithm may be relied upon to output further exact/inexact occurrences of the pattern.

As a case study, the new algorithm SIARCT-CFP was run on the Nocturne in E major op.62 no.2 by Frédéric Chopin (1810–1849). [2] This is a sensible choice of piece, as it contains multiple variations of the opening theme (c.f. Figures 2B and D for instance). Fourteen patterns were output in total, one of which Q is bounded in Figure 2A, and occurs translated three times (bars 27–28, 58–59, and 60–61). These occurrences are rated as very similar to Q, with normalised matching scores close or equal to 1. The time series output by the FP has mean .264 and standard deviation .173, suggesting that the occurrence in Figure 2C is not distinguishable from other unrelated material. This makes sense, as although the contour and rhythm of the melody are as in Q, the pitch intervals are different (see arrows) and so is the accompaniment. We note, however, that

[2] The first part of the algorithm, SIAR, ran with parameter $r = 1$. Second, the compactness trawler (CT) ran with compactness threshold $a = 4/5$, cardinality threshold 10, and lexicographic region type [7]. Third, the categorising and fingerprinting (CFP) ran with similarity threshold $c = 1/2$.

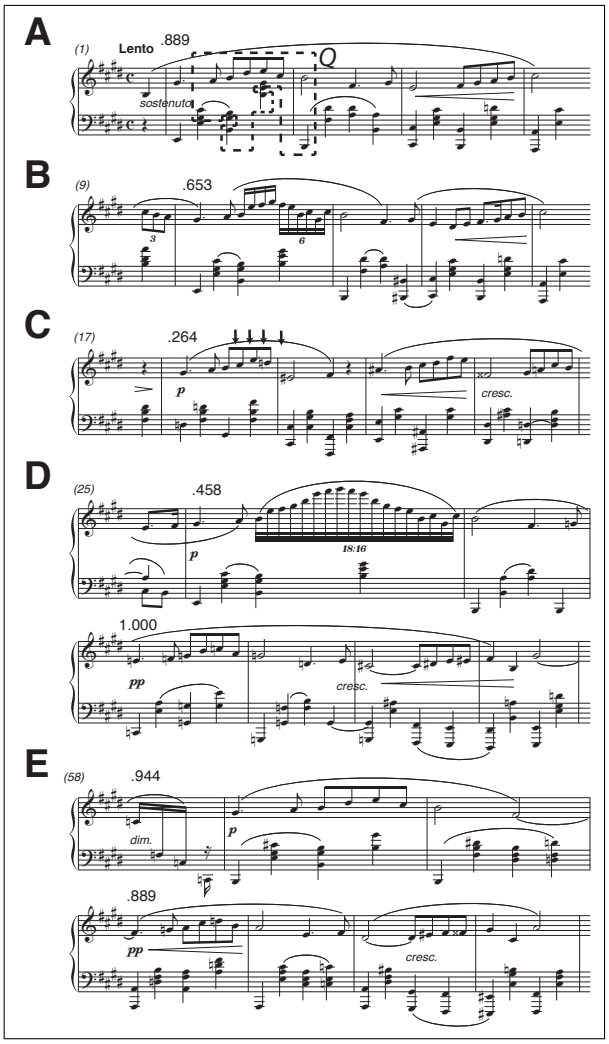

Figure 2. Excerpts from the Nocturne in E major op.62 no.2 by Chopin. Dashed lines in (A) bound a pattern Q discovered by SIARCT, which is used to match other inexact occurrences, with degree of exactness indicated in the figure by numbers in [0, 1]. Pedalling omitted for clarity.

the FP algorithm could be extended further to incorporate contour (up, down, same), as well as other viewpoints [9], because of its use of locally constrained comparisons.

3. THE PRECISION PROBLEM

3.1 Categorisation by Pattern Matching

Now that we have integrated some inexact pattern matching techniques into our pattern discovery approach, it is possible to employ them for the purposes of categorisation, based on the idea that P2 [17] or FP [2] can be used to compare two discovered patterns P_i and P_j in exactly the same way as if $P_i = P$ was a query and $P_j = D$ was a point set (or vice versa, as the measures are symmetric).

The second "C" in SIARCT-CFP stands for a **c**ategorisation process, which will be described now. The purpose of categorisation is to reduce an overwhelming amount of information (e.g., output patterns) to a more manageable number of exemplars. Here *categorisation* does not mean

classifying patterns into an accepted/interesting category versus a rejected/uninteresting category; rather it means grouping similar patterns and representing each group with one exemplar pattern. Our motivation for categorising the output of SIARCT is to improve its precision: while the precision and recall of pattern discovery algorithms has been shown to benefit from compactness trawling, the precision is still quite poor [7]. For example, SIARCT outputs 76 patterns when run on Chopin's op.62 no.2, which can be reduced to fourteen patterns by using the following categorisation process:

1. Let $P_1, P_2, \ldots, P_M$ be the output of a pattern discovery algorithm, sorted descending by a rating of perceived pattern importance [6], or some other ordering. Let $J = \{1, 2, \ldots, M\}$ index the patterns that are uncategorised currently;
2. For the most important uncategorised pattern, index $i = \min(J)$, calculate the maximum normalised matching scores $s(P_i, P_j)$ for each $j \in J$, $j \neq i$;
3. For each similarity score $s(P_i, P_j)$ that is greater than some specifiable similarity threshold $c \in [0, 1)$, place pattern P_j in the category for which P_i is the exemplar, and remove j from J;
4. Repeat steps 2 and 3 until either J has one element k, in which case define P_k to be an exemplar with category membership P_k, or otherwise J is empty;
5. For the purposes of algorithm evaluation, return only the exemplars $P_{i(1)}, P_{i(2)}, \ldots, P_{i(m)}$.

Depending on the choice of c, $m \ll M$. The categorisation process can be visualised with two similarity matrices (Figure 3). The matrix in Figure 3A contains the maximum normalised matching scores for each pair of 76 output patterns for Chopin's op.62 no.2, ordered as in step 1 above. The matrix in Figure 3B is a permutation of 3A, showing the categorised patterns ($c = .5$) in their fourteen categories, bounded by white squares. The fourth square from top-left in Figure 3B represents the category for which Q in Figure 2A is the exemplar. The fivefold ($5.43 \approx 76/14$) reduction in output achieved by pattern-matching categorisation may well improve precision: as discussed, the theme annotated in Figure 2A survives the categorisation process, and so do all of the repetitions in this piece lasting four or more bars (results not shown). Pattern-matching categorisation also constitutes a novel and interesting use of the FP algorithm [2]. It should be noted that choosing too low a value for c could lead to over-reduction and filtering out of analytically interesting patterns. For instance, the first two squares in Figure 3B show considerable variegation, suggesting that some interesting subcategories may be overlooked.

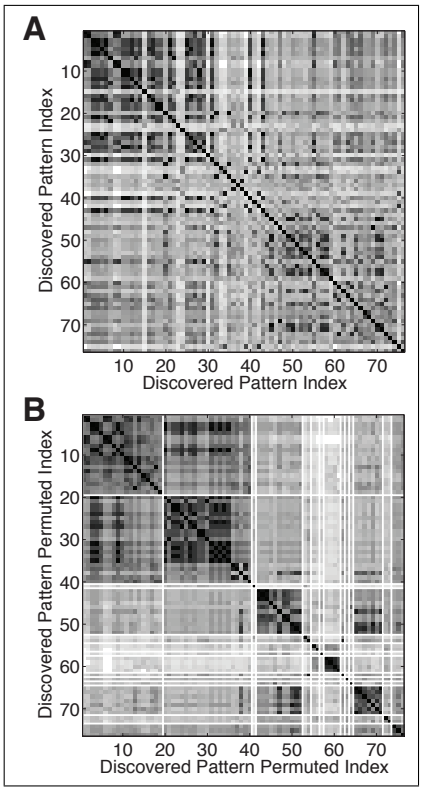

Figure 3. (A) Pairwise symbolic musical similarities (ranging from white for dissimilar to black for identical) for 76 patterns discovered by SIARCT in Chopin's op.62 no.2, ordered by a rating formula for perceived salience; (B) Permutation of the above matrix, with white lines indicating the results of categorising into fourteen groups.

3.2 Consecutive Points and Conjugate Patterns

The final novel contribution of this paper is to evaluate the SIARCT pattern discovery algorithm [5] against a collection of music containing repeated sections, and to compare its performance (especially precision) to SIA [14] and SIAR [5]. SIA outputs thousands of patterns for Chopin's op.62 no.2 (and other pieces of music [7]), so it is necessary to develop a more parsimonious pattern discovery algorithm for use as input to the categorisation and fingerprinting components described above (e.g., SIARCT outputs only 76 patterns for Chopin's op.62 no.2).

It has long been thought that in order to discover repeated patterns within a geometric representation D of a piece, it is necessary to calculate the difference between each pair of n points ($n[n-1]/2$ calculations in total), as in SIA [14]. Unlike SIA, the first step of SIARCT is to calculate the difference between consecutive pairs of points only ($n - 1$ calculations). Some exhaustive pairwise comparisons are still made in the second step, but for small, non-overlapping subsets of D, meaning that the total number of difference calculations performed by SIARCT is far less than $n[n-1]/2$, in all but one degenerate case. [3] The third step of SIARCT makes use of a concept known as *conjugate patterns* [5]: if a pattern containing l points occurs m times in a point set, then there exists in the same point set a pattern consisting of m points that occurs l times. The fourth step calculates MTPs for each vector in a list L. As a consequence of manipulating conjugate patterns, the vectors corresponding to repeated sections should be at or near

[3] Please see [5] for the algorithmic details.

the top of L. So for this step we could: (1) distribute each MTP calculation to parallel processors, and/or; (2) output MTPs dynamically for the user to browse, whilst calculation of the remaining MTPs continues. The main claim is that SIARCT will have much smaller output than SIA, with minimal or no negative impact on its performance as measured by precision, recall, and robust versions of these metrics [4]. The compactness trawler (CT) part of SIARCT is exactly as in [7], so is not addressed again here.

SIA, SIAR, and SIARCT were run on point-set representations of movements by Ludwig van Beethoven (1770–1827) and Chopin listed in Figure 4A. SIARCT ran with compactness threshold $a = 1$, and points threshold $b = 50$. This means that only patterns containing 50 points or more were returned, and they had to have maximal compactness of 1. The parameter values make sense in terms of trying to discover repeated sections. To make the evaluation fair, we also filtered the results of SIA and SIAR, returning only those patterns that contained 50 points or more. In the results, these versions of SIA and SIAR are referred to as SIA (50+) and SIAR (50+).

3.3 Evaluation Results

Figure 4B shows the log of the total number of patterns output by each algorithm for each movement/piece. It supports the claim that SIAR has a much smaller output than SIA. It is difficult to see from Figure 4B, but the same observation applies to the filtered versions of each algorithm, SIAR (50+) and SIA (50+). The number of patterns output by SIARCT is several orders of magnitude less than that of any other algorithm. Figure 4C and Figure 4E show that compared with SIA's performance, SIAR is not negatively impacted by restricting calculations to consecutive pairs of points. The establishment precision and establishment recall for SIAR and SIA are comparable across all pieces.

Overall, the most effective algorithm is SIARCT (see Figure 4C and Figure 4E). For half of the pieces, it discovers all ground truth patterns exactly (Figure 4F). When SIARCT fails to discover a ground truth pattern exactly, often this is due to a difference between the repeated section as written in the score, and the repeated pattern as heard in a performance. For instance, in the fourth movement of Beethoven's op.7, bars 65–70 are marked as a repeated section, and this is included in the ground truth. The repeated notes extend beyond these bars in both directions, however, creating a longer repeated pattern in a performance. SIARCT discovers the latter, *performed* pattern, which reduces exact precision and recall. The more robust *establishment* metrics are not much reduced (e.g., see Figure 4E), and arguably discovering the performed pattern is preferable from a music-perceptual point of view.

4. DISCUSSION AND FUTURE WORK

This paper identifies two valid reasons why the geometric approach to intra-opus pattern discovery has attracted some criticism—namely (1) the approach extends to a limited number of inexact repetition types only, and (2) typically geometric pattern discovery algorithms are imprecise, returning many false positives results. A new algorithm called SIARCT-CFP has been described and evaluated component-wise, in an attempt to address these criticisms. It is the first geometric pattern discovery algorithm to fully integrate an inexact pattern matching component (the fingerprinting algorithm of [2]), and this matching component was shown to be effective for retrieving inexact occurrences of themes in pieces by Mozart and Chopin. The comparison of the FP algorithm [2] to a baseline pattern matching algorithm P2 [17] demonstrated that the former was superior for a particular example. In general it may be preferable to have two or more pattern matchers at one's disposal, however, as the number of variation techniques is large, and trying to account for them all with one algorithm will likely produce false positive matches.

The precision metrics were of particular interest to us in the comparative evaluation of SIARCT [5], SIAR [5], and SIA [14], as we claimed that SIARCT could achieve levels of precision comparable to SIA and SIAR, without harming recall. This claim was supported by the evaluation results, although in future work it will be necessary to see if similar results are achieved for ground truths containing shorter patterns than repeated sections.

Our *translationally exact once* (TEO) hypothesis (see Section 2) was borne out in the case study of Chopin's op.62 no.2, where Q (Figure 2A) occurred exactly under translation (bars 27–28, 58–59, and 60–61), and its contents were sufficient for use as a query to retrieve less exact versions such as in bars 9 (Figure 2B) and 25 (Figure 2D). For the case study of the Theme section and Variation II from Mozart's K331, SIARCT was able to discover perceptually salient patterns such as P in Figure 1A, which recurs in bars 5-7 of the Theme section (not shown). As the TEO hypothesis holds in both cases, future work should focus on finding counterexample pieces, as this will help to refine and improve our underlying assumptions and ensuing algorithms. Future work will also attempt to show users/developers the differences between themes and partial matches, and to identify variation techniques (triplets, minore, etc.) automatically.

5. ACKNOWLEDGEMENTS

This paper benefited from the use of Kern Scores, and helpful discussions with David Meredith. We would like to thank four anonymous reviewers for their comments. This work is supported by the Austrian Science Fund (FWF), grants Z159 and TRP 109.

6. REFERENCES

[1] T. Ahonen, K. Lemström, and S. Linkola: "Compression-based similarity measures in symbolic, polyphonic music," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 91–96, 2011.

[2] A. Arzt, S. Böck, and G. Widmer: "Fast identification of piece and score position via symbolic fingerprint-

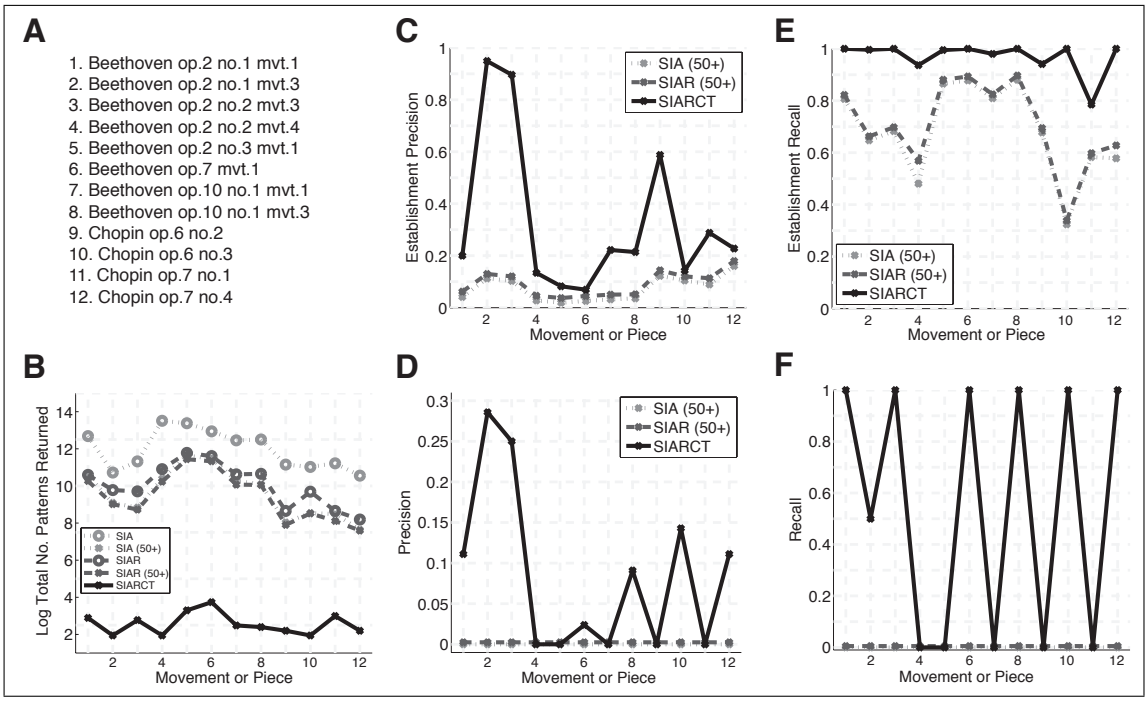

Figure 4. Evaluation metrics for three algorithms, run on eight movements by Beethoven and four pieces by Chopin.

ing," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 433–438, 2012.

[3] I. Bent and A. Pople: "Analysis," *The new Grove dictionary of music and musicians*, Macmillan, London, 2001.

[4] T. Collins: "Discovery of repeated themes and sections," Retrieved 4th May 2013, from http://www.music-ir.org/mirex/wiki/2013: Discovery_of_Repeated_Themes_%26_Sections

[5] T. Collins: *Improved methods for pattern discovery in music, with applications in automated stylistic composition*, PhD thesis, Faculty of Mathematics, Computing and Technology, The Open University, 2011.

[6] T. Collins, R. Laney, A. Willis, and P. Garthwaite: "Modeling pattern importance in Chopin's mazurkas," *Music Perception*, Vol. 28, No. 4, pp. 387–414, 2011.

[7] T. Collins, J. Thurlow, R. Laney, A. Willis, and P. Garthwaite: "A Comparative Evaluation of Algorithms for Discovering Translational Patterns in Baroque Keyboard Works," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 3–8, 2010.

[8] D. Conklin and C. Anagnostopoulou: "Representation and discovery of multiple viewpoint patterns," *Proceedings of the International Computer Music Conference*, pp. 479–485, 2001.

[9] D. Conklin and I. Witten: "Multiple viewpoint systems for music prediction," *Journal of New Music Research*, Vol. 24, No. 1, pp. 51-73, 1995.

[10] O. Lartillot and P. Toiviainen: "Motivic matching strategies for automated pattern extraction," *Musicae Scientiae*, Discussion Forum 4A, pp. 281–314, 2007.

[11] S. Livingstone, C. Palmer, and E. Schubert: "Emotional response to musical repetition," *Emotion*, Vol. 12, No. 3, pp. 552–567, 2012.

[12] A. Lubiw and L. Tanur: "Pattern matching in polyphonic music as a weighted geometric translation problem," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 154–161, 2004.

[13] D. Meredith: "Point-set algorithms for pattern discovery and pattern matching in music," *Proceedings of the Dagstuhl Seminar on Content-Based Retrieval*, 2006.

[14] D. Meredith, K. Lemström, and G. Wiggins: "Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music," *Journal of New Music Research*, Vol. 31, No. 4, pp. 321–345, 2002.

[15] E. Narmour: "Some major theoretical problems concerning the concept of hierarchy in the analysis of tonal music," *Music Perception*, Vol. 1, No. 1, pp. 129–199, 1983.

[16] C. Romming and E. Selfridge-Field: "Algorithms for polyphonic music retrieval: the Hausdorff metric and geometric hashing," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 457–462, 2007.

[17] E. Ukkonen, K. Lemström, and V. Mäkinen: "Geometric algorithms for transposition invariant content-based music retrieval," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 193–199, 2003.

[18] A. Wang: "An industrial strength audio search algorithm," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 7–13, 2003.

Paper B: Submission...

LEARNING DISTRIBUTED REPRESENTATIONS FOR MULTIPLE-VIEWPOINT MUSIC PREDICTION

First author
Affiliation1
author1@ismir.edu

Second author
Retain these fake authors in submission to preserve the formatting

Third author
Affiliation3
author3@ismir.edu

ABSTRACT

The analysis of sequential patterns is important for extracting information from music owing to its fundamentally temporal nature. In this paper, we present a distributed model for music prediction based on the Restricted Boltzmann Machine (RBM). This model is first used for predicting the next pitch in a given pitch sequence, and then adapted to also make use of information in sequences of note-durations from monophonic melodies, on the same task. In the process, we also propose an efficient way of representing this additional information that takes advantage of the RBM's structure. Results show that this RBM-based model performs very competitively with previously evaluated n-gram models and also outperforms them in certain cases. It is able to make use of information present in longer sequences more effectively than n-gram models, while also scaling gracefully in the number of free parameters required. This makes the present approach very promising for improved music prediction. The results obtained here are also in agreement with previously demonstrated success of distributed models for natural language.

1. INTRODUCTION

Whether or not computational models can capture elements of style and structure in tonal music is a question that has received much attention in computer music research. It started with an interest in Markov models for music analysis in the late 1940s and 1950s [2]. The years that followed saw a further diversification of the models applied to style-oriented music generation, as well as their musical applications. Some of the notable approaches that emerged are based on rule-discovery [7], neural networks [15], genetic algorithms [4], and context-free grammars [11].

Among the different approaches adopted for predicting music, we focus here on two that have received long-standing attention over the past 2-3 decades. The first are Markov models, which have been applied to a number of musical research tasks, including computer-assisted composition [2], chorale harmonization [1], musical style analysis [6], and as a collaborative musical instrument [16]. On the other hand, connectionist approaches have also proven to be effective in music. Neural networks have been successfully applied in the past for Jazz solo generation [19], rhythm analysis [20], and music composition [15]. While their applications are fairly similar, the way in which information is encoded in these two classes of models is fundamentally different, with the former relying on frequency counts of sequences occurring in the data, and the latter learning distributed representations of sequences using networks of interconnected units with simple non-linear activation functions.

In this paper, we present a model for music prediction based on one such distributed model - the Restricted Boltzmann Machine (RBM) [18]. The following are some reasons for this choice. Firstly, the inherent non-linearity of the RBM makes it a suitable candidate for learning complex patterns in data, such as those occurring in musical sequences. There exist efficient algorithms for training these models [5, 9]. The RBM, with its straightforward extensibility to deep networks [10], has become a vital building block for creating models that are capable of learning representations of data at multiple levels of abstraction. Moreover, problems related to insufficient data and computing power no longer pose limitations on the use of such distributed models as was the case in the past.

The model described here first analyses fixed-length sequences of musical pitch, and compares favourably to n-gram models that were previously evaluated with a prediction task on a corpus of monophonic MIDI melodies [17]. Its ability to make use of information in longer sequences more effectively is in agreement with previous work in statistical language modelling [3]. The structure of the proposed model ensures that it scales only linearly with the length of sequences to be learned, and also with the number of symbols in data. This pitch-only version of the model is then adapted to also make use of sequences of note-durations in the melodies. In the process, we also propose an efficient way to represent this additional information that takes advantage of the RBM's structure and thus, limits model complexity. We demonstrate an improvement in results by combining the two models using ideas proposed in multiple viewpoints for music prediction [6] using a simple arithmetic mean of their individual probability estimates. A Matlab implementation of the model, along with scripts used to generate the results in this paper, are available upon request.

The remainder of this paper is organized as follows. The next section introduces music prediction, and multiple viewpoint systems as a framework for music prediction with an analogy to natural language. Section 3 explains the RBM and its discriminative interpretation which make up the basis for the prediction model proposed in this paper. This is followed by a description of the model itself, in Section 4. An evaluation of the the model and its comparison with previously evaluated n-gram models is presented Section 5, followed by discussion on possible directions for future research in Section 6.

2. MUSIC PREDICTION WITH MULTIPLE-VIEWPOINT SYSTEMS

In order to explain music prediction with multiple viewpoints, the analogy to natural language is used here. In statistical language modelling, the goal is to build a model that can estimate the joint probability distribution of sequences of words occurring in a language L. A statistical language model (SLM) can be represented by the conditional probability of the next word w_T given all the previous ones $[w_1, \ldots, w_{(T-1)}]$ (written here as $w_1^{(T-1)}$), since

$$P(w_1^T) = \prod_{t=1}^{T} P(w_t | w_1^{(t-1)}) \,. \tag{1}$$

The most commonly used SLMs are *n-gram* models, which rely on the simplifying assumption that the probability of a certain word in a sequence depends only on the immediately preceding $(n-1)$ words [14]. This is known as the Markov assumption, and reduces (2) to

$$P(w_1^T) = \prod_{t=1}^{T} P(w_t | w_{(t-n+1)}^{(t-1)}) \,. \tag{2}$$

Following this explanation, musical styles can be interpreted as vast and complex languages [6]. The goal in music prediction is to predict the continuation s_T of a sequence of *musical events* $s_1^{(T-1)}$ of a *musical language* S. For each prediction, context information is obtained from a subsequence of events $s_{(T-n+1)}^{(T-1)}$ immediately preceding s_T. A musical event can be any directly observable or derived musical facet such as pitch, note duration, inter-onset interval, or a combination of two or more of such facets. In music prediction, much in the same way as statistical language modelling, one is concerned with the following conditional distribution $p(s_T | s_1^{(T-1)})$, or under the Markov property $p(s_T | s_{(T-n+1)}^{(T-1)})$.

The framework of multiple-viewpoint systems for music prediction [6] was developed with the aim to extend statistical modelling techniques to domains, where events have an internal structure and are richly representable by exploiting this structure. One such domain is music, where a musical piece can be expressed in terms of various musical dimensions such as pitch, duration, scale degree, etc. The framework of multiple viewpoints aims at exploiting information contained in these different dimensions, while at the same time keeping a check on the dimensionality of the models using these features. We limit our explanation of multiple-viewpoint systems in this section to the case of monophonic music, which is the focus of this paper.

As noted previously, a musical event s refers to the occurrence of a note in a melody, without reference to any particular property it may represent such as pitch, IOI, scale-degree, etc. The set of all representable events (the *event space*) is denoted by the symbol ξ, and the set of all sequences of events by ξ^*. A *type* is an abstract property of events, such as pitch, scale-degree or duration. For every type τ, there exists an associated partial function $\mathbf{\Psi}_\tau$ which maps sequences of events in ξ^* to elements of type τ. The set of all syntactically valid elements of type τ (its domain) is denoted by $[\tau]$, and the set of all sequences representable using elements of type τ by $[\tau]^*$.

A *viewpoint* comprises 1) a partial function $\mathbf{\Psi}_\tau : \xi^* \rightharpoonup [\tau]$, and 2) a context model of sequences in $[\tau]^*$. The set of all viewpoint types is categorized into *basic viewpoints* and *derived viewpoints*, of which the latter tend to represent more abstract properties. In the present work, we are interested in the prediction of the basic type *pitch*, which refers to musical pitch. A *linked viewpoint* is used to model correlations between any basic types in an event. Originally, it was proposed to achieve this through the use of a *product type*, which represents the space made up of the Cartesian product of the constituent basic types. Later in section 4, we propose an efficient alternative to this that is facilitated by the distributed model presented in this work.

A *multiple-viewpoint system* (MVS) is a set of models, each of which is trained on sequences of one *type*, whose individual predictions are combined together in some way to influence the prediction of the next event in a given event sequence. Given a context s_{t-n+1}^{t-1} and an event s_t, each viewpoint τ in an MVS must compute the probability $p_\tau(s_t | s_{t-n+1}^{t-1})$. It is first necessary to convert the sequence s_{t-n+1}^{t-1} in ξ^* to a string in $[\tau]$. Each viewpoint then predicts a distribution over $[\tau]$. Finally, these individual predictions are (if required) mapped to the basic type τ_b that is to be predicted, and combined using weighting schemes or ranked lists for inference over individual viewpoint systems. For example, $\{\tau_1, \tau_3\}$, $\{\tau_2, \tau_1 \otimes \tau_3\}$, and $\{\tau_2, \tau_1 \otimes \tau_2, \tau_3\}$ are multiple viewpoint systems. While originally, n-gram models were proposed to be used with the multiple viewpoints framework, we demonstrate how a distributed model, such as the Restricted Boltzmann Machine used here, can serve as more efficient and scalable alternative for improved prediction performance.

3. RESTRICTED BOLTZMANN MACHINES

The Restricted Boltzmann Machine (RBM) [18] is an undirected, bipartite graph consisting of a set of r visible units $\mathbf{v}$ and a set of q hidden units $\mathbf{h}$. These make up the visible and hidden layers of the RBM respectively. The two layers are fully connected to each other, but there exist no connections between any two hidden units, or any two visible units. Additionally, each of the hidden and visible units is connected to its respective bias unit that always has the

value 1. The edge between the j^{th} visible node and the

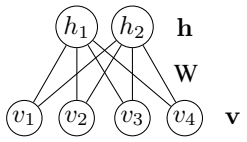

Figure 1. A simple Restricted Boltzmann Machine with four visible, two hidden, and no bias units.

i^{th} hidden node is associated with a weight w_{ji}. All these weights are together represented in an $q \times r$ matrix $\mathbf{W}$. The weights of connections between visible units and their respective biases is denoted by $\mathbf{b}$, an r-dimensional vector. Likewise, for the hidden units there is an q-dimensional hidden bias $\mathbf{c}$. The RBM is fully characterized by the three parameters $\mathbf{W}$, $\mathbf{b}$ and $\mathbf{c}$. Figure 1 shows a simple RBM with four visible and two hidden units, and the bias unit ignored to better illustrate its bipartite structure. Owing to this structure, the RBM exhibits the property of conditional independence between the nodes of each of the hidden and visible layers of the RBM when those of the other layer are given. That is

$$p(\mathbf{h}|\mathbf{v}) = \prod_{j=1}^{q} p(h_j|\mathbf{v}) \tag{3}$$

$$p(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{r} p(v_i|\mathbf{h}) \; . \tag{4}$$

The values of the binary stochastic units in the hidden layer given the visible layer (and vice versa) are given by the logistic sigmoid function as $p(h_j = 1|\mathbf{v}) = \sigma(c_j + W_{j\cdot}\mathbf{v})$, and $p(v_i = 1|\mathbf{h}) = \sigma(b_i + W'_{i\cdot}\mathbf{h})$ respectively.

The RBM is a special case of the Boltzmann Machine, which is an energy-based model for representing probability distributions [13]. In such energy-based models, probability is expressed in terms of an energy function. In the case of the RBM, this function is expressed as

$$Energy(\mathbf{v}, \mathbf{h}) = -\mathbf{b}^\top \mathbf{v} - \mathbf{c}^\top \mathbf{h} - \mathbf{h}^\top \mathbf{W} \mathbf{v} \; . \tag{5}$$

Learning in energy-based models can be carried out in a *generative* fashion, by updating the weights and biases in order to minimize the overall energy of the system with respect to the training data. This amounts to maximizing the log-likelihood function of the joint probability distribution $p(\mathbf{v})$, which is given by

$$p(\mathbf{v}) = \frac{e^{-FreeEnergy(\mathbf{v})}}{Z}, \tag{6}$$

with $Z = \sum_{\mathbf{v}} e^{-FreeEnergy(\mathbf{v})}$, i.e.,

$$FreeEnergy(\mathbf{v}) = -\log \sum_{\mathbf{h}} e^{-Energy(\mathbf{v},\mathbf{h})} \; . \tag{7}$$

While computing the exact gradient of the log-likelihood function for $p(\mathbf{v})$ is not tractable, an approximation of this gradient called the Contrastive Divergence gradient has been found to be a successful update rule for training RBMs [5, 9]. With the contrastive divergence update, the RBM can be trained efficiently using gradient descent.

The RBM described above models the joint probability $p(\mathbf{v})$ of the set of visible units $\mathbf{v}$. For music prediction (Section 2), one has to model a conditional distribution of the form $p(y|\mathbf{x})$. It has been demonstrated in [12] how an RBM can be used for a *discriminative* task such as classification, where it is known as a Discriminative Restricted Boltzmann Machine (DRBM). The posterior class probability distribution of the DRBM has the form

$$p(\mathbf{y} = \mathbf{e}_c|\mathbf{x}) = \sum_{\mathbf{h}} p(\mathbf{y} = \mathbf{e}_c, \mathbf{h}|\mathbf{x}) \tag{8}$$

$$= \frac{e^{-FreeEnergy(\mathbf{x},\mathbf{e_c})}}{\sum_{c'=1\ldots C} e^{-FreeEnergy(\mathbf{x},\mathbf{e_{c'}})}} \tag{9}$$

where $\mathbf{x}$ is the input vector, and $\mathbf{y}$ is a vector that is a *1-of-C* representation of the class (also known as *one-hot* encoding), with C being the number of classes. If $\mathbf{x}$ belongs to a class c, then $\mathbf{y} = \mathbf{e}_c$, where $\mathbf{e}_c$ is a vector with all values set to 0 except at position c. With respect to the general RBM, $\mathbf{x}$ and $\mathbf{y}$ together make up the visible layer $\mathbf{v}$.

Assuming a training set $\mathcal{D}_{train} = \{(\mathbf{x}_i, y_i)\}$ where $\mathbf{x}_i$ and $y_i \in \{1, \ldots, C\}$ are the i-th input vector and target class respectively, to train a DRBM generatively involves minimizing the negative log-likelihood

$$\mathcal{L}_{gen}(\mathcal{D}_{train}) = -\sum_{i=1}^{|\mathcal{D}_{train}|} \log p(\mathbf{x}_i, \mathbf{y}_i) \; . \tag{10}$$

The RBM, used in a discriminative manner forms the basis of the prediction model described in the next section.

4. A DISTRIBUTED MODEL FOR USE WITH MULTIPLE VIEWPOINTS

The prediction model we present in this paper models the conditional distribution $p(s_t|s_{(t-n+1)}^{(t-1)})$. The model places no restrictions on the *type* associated with events in the context $s_{(t-n+1)}^{(n-1)}$ (*source type*), or the predicted event s_t (*target type*). In the simplest case, both are the same. In the case where they are different, the performance of the model depends on how informative the source types are of the target type. In the present work, we demonstrate this model with two cases where (1) both the source and target viewpoint types are musical pitch, and (2) the source viewpoints include pitch and duration with the target viewpoint as pitch. For each monophonic melody (in the MIDI format) in a given dataset, sequences of the relevant source and target types are first extracted using the MIDI Toolbox [8]. These values are encoded as binary 1-of-$|\tau|$ vectors, where $|\tau|$ is the size of the domain of type τ. In the case where more than one source viewpoints exist, their corresponding vectors are simply concatenated. Such an idea is similar to that of the linked viewpoint type proposed in [6]. There are however, two important distinctions between the two. Firstly, the source and target viewpoints must be identical in the case of the n-gram models

originally proposed for use with multiple-viewpoint systems, whereas this is not mandatory for the RBM-based model. Secondly, a linked viewpoint between two arbitrary types τ_1 and τ_2 of domain sizes $|\tau_1|$ and $|\tau_2|$ respectively, would have a domain of size $|\tau_1| \times |\tau_2|$. Thus, for sequences of length n, the number of free parameters to be estimated are $(|\tau_1| \times |\tau_2|)^n$. In contrast, the number to be estimated in case of the RBM-based model, with q hidden units, is $q \times ((n-1) \times (|\tau_1| + |\tau_2|) + |\tau_3|)$, where τ_3 is the target type. Thus, the complexity of the latter does not increase exponentially as in the case of the former. At the moment, the model only handles a single target type. It is also to be noted that the nodes of the visible layer in the RBM represent a sequence of n events, thus making the number of visible units of the underlying RBM $(n-1) \times (|\tau_1| + |\tau_2|) + |\tau_3|$. Such a model only scales linearly with the length of the learned sequences, as well as the domain size of each of the involved viewpoint types. Its structure is depicted in Figure 2.

It is worth mentioning here that this RBM-based model adopts some features of the *neural probabilistic language model* described in [3]. Both these models are similar in the manner in which sequences (words in the case of one, and musical pitch in the other) are presented as input to their respective networks. Being the first to propose such a framework for predicting word sequences, [3] suggested the use of feed-forward, or recurrent neural networks. In contrast, an energy-minimization approach is adopted here to train an RBM-based model. Also in [3], an additional stage of dimensionality reduction was also performed to learn distributed word vectors in order to deal with a very large vocabulary of words. This step was not found to be strictly necessary in the present scenario where there are relatively much fewer musical pitches and note-durations, and is left to be explored in the future.

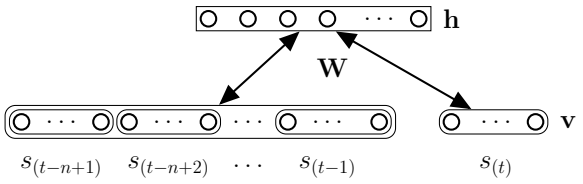

Figure 2. The structure of the prediction model. The set of nodes in the visible layer grouped together on the left make up the context $s^{(t-1)}_{(t-n+1)}$ of the source type(s). The set of nodes to the far right correspond to the target type.

To train the model generatively, a sequence $s^t_{(t-n+1)}$ is clamped to all the nodes in the visible layer. Training is done using the first instantiation of the Contrastive Divergence learning algorithm (CD-1). This simply means that the model parameters are updated after a single step of Gibbs sampling [9]. During prediction, the probability of each of the possible pitch values in the prediction space is determined using (9). Note that the distribution generated in this manner does not require any kind of smoothing operation for unseen sequences, as in the case of n-gram models. While it was found to be necessary in [17], that an empirical evaluation of different smoothing techniques be carried out in order to establish the most reliable one, such a need does not arise in the case of the RBM-based model.

5. EVALUATION

In order to evaluate the proposed model in an application independent manner, we refer to a recent study on n-gram models for music prediction [17]. There, entropy – a common metric in statistical language modelling – was used to evaluate the music prediction models. Given a probability mass function $p(s \in S) = p(\chi = s)$ of a random variable χ distributed over a discrete alphabet $S = \{s_1, \ldots, s_k\}$ such that the individual probabilities are independent and sum to one, the entropy $H(p)$ is defined as

$$H(p) = -\sum_{s \in S} p(s) \log_2 p(s) \; . \tag{11}$$

The value of entropy, with reference to a prediction model, indicates how *uncertain* it is about its predictions. A higher value of entropy reflects greater uncertainty. In practice, one rarely knows the true probability distribution of the stochastic process and uses a model to approximate the probabilities in (11). *Cross entropy* is a quantity which represents the divergence between the entropy calculated from the estimated probabilities and the source model. It is computed over all the sequences of length n in the test data $\mathcal{D}_{test}$, as

$$H_c(p_{mod}, \mathcal{D}_{test}) = -\frac{1}{|\mathcal{D}_{test}|} \sum_{s_1^n \in \mathcal{D}_{test}} \log_2 p_{mod}(s_n | s_1^{(n-1)}) \tag{12}$$

where p_{mod} is the probability assigned by the model to the last pitch in the sequence, given its preceding context. Cross entropy approaches the true entropy of the sequence as the number of test samples ($|\mathcal{D}_{test}|$) increases.

Evaluation is carried out on a corpus of monophonic MIDI melodies that cover a range of musical styles. This is a collection of 8 datasets, previously used to evaluate n-gram models for music prediction [17]. It consists of 152 Canadian folk melodies, 185 Bach chorales, 91 Alsatian folk melodies, 119 Yugoslavian folk melodies, 93 Swiss folk melodies, 104 Austrian folk melodies, 213 German folk melodies, and 237 Chinese folk melodies and a total of $54,308$ events (8893 bigrams, 8708 trigrams of musical events). Also in [17] two different models were evaluated, both individually and in combination. The first of these is a Long-Term Model (LTM), that is governed by structure and statistics induced from a large corpus of sequences from the same genre. And the other is a Short-Term Model (STM) which relies on structure and statistics particular to the melody being predicted. The notion of LTM and STM was introduced in [6]. The RBM-based model deals only with long-term effects that are induced from a corpus, and is thus compared with the performance of the LTM.

A 10-fold cross-validation technique is used for evaluation, where each fold contains a hold-out test set $\mathcal{D}_{test}$

and the remaining data $\mathcal{D}_{train}$ for learning. An internal 3-fold cross-validation is performed within $\mathcal{D}_{train}$ for model selection. The best training hyper-parameters are chosen based on the validation error in this step, and a model with this set of hyper-parameters is then trained on the entire $\mathcal{D}_{train}$ and evaluated on $\mathcal{D}_{test}$. Training is done using mini-batch gradient descent with a batch size of 100 samples. The RBM was trained over 300 epochs. During model selection, the learning rate λ was varied linearly between 0.02 and 0.1 in steps of 0.04 and the weight-regularization parameter w_{cost} between 0.0002 and 0.001 in steps of 0.0004. The momentum parameter ν, was set to 0.5 during the first five epochs and then increased to 0.9 for the rest of the training.

In this paper, we carry out three types of evaluation. The first measures the information content of pitch-only version of the proposed model in terms of entropy and compares it with a similar evaluation carried out on n-gram models for music prediction in [17]. It was observed that the RBM-based model compares favourably with the best of the n-gram models owing to the fact that it makes use of information in longer sequences. In the second, we compare a variant of the model with source types as pitch and duration, and target type as pitch to its n-gram model based linked viewpoint counterpart. And last, we combine these two models using mixture of experts and demonstrate how this can further improve the model performance in comparison to the individual models.

The first evaluation is carried out separately for each of the individual datasets. Event-sequence length n is varied in the range $[2, 9]$, and it was found that this range of values was sufficient for the RBM-based model to outperform the n-gram models on all but one of the datasets considered here (Bach chorales). Due to limited space, only the average information content values over all the datasets are presented in Table 1. A similar trend was observed for each individual dataset in the corpus.

Overall, the RBM-based model performs very competitively with respect to the best n-gram model-based LTMs evaluated in [17]. This is illustrated in Table 1. Here, *RBM-n* refers to a model which is trained on pitch sequences of length n. The column *Markov* corresponds to the LTM with a global order bound 2 (3-gram), which was found to do best among those with a limited order bound. And *Markov** refers to the LTM with an unbounded order. The RBM-based model, at one point or another performs on par with, or better than the best LTM. The value of n where a model performs better than the unbounded order Markov model happens to be different on different datasets, and typically occurs at $n = 3$ or $n = 4$. These differences, it is believed, might be attributed to the information dynamics of the different musical styles considered here. The value of $n = 6$ with average cross entropy 2.737, and $n = 8$ with average cross entropy 2.734 can be considered here as the best cases. It is to be noted that, although the case of $n = 8$ has a slightly lower cross entropy value, this has actually increased a little on some of the datasets when compared to $n = 6$. Another observation, along the same lines, is that the RBM-based model seems to show signs of deterioration in performance for $n > 6$. We suspect that the performance of these longer context models can be improved with a search over a larger hyper-parameter space, but leave this to be explored in the future. A similar trend was observed in the case of the Markov models in [17], but at a much lower global order bound of 3. This indicates the RBM-based model is able to make information available in longer event sequences.

In the second evaluation, we compared te information content of the single and multiple source type models (pitch and pitch with duration respectively) using the same target type (pitch), on the Bach chorale subset of the corpus. The results are shown in Table 2. One can conclude, from the lack of significant improvements of the multiple source type model over the single source type model, that this was not a very good choice of viewpoints to link, given the dataset. This is the case with n-gram models too. However, on an average the RBM-based model maintains its performance, with marginal improvements in the case of some values of sequence length n. The relative improvements of both classes of models with the addition of the duration type, in Table 2, indicate that the representation for multiple source types proposed in Section 4 as an alternative to the linked viewpoints may indeed be effective.

n	2	3	4	5
$RBM(single)$	2.732	2.554	2.510	2.465
$RBM(multiple)$	2.715	2.552	2.513	2.464
$RBM(combination)$	2.686	2.525	2.489	2.451
$Markov(single)$	2.737	2.565	2.505	2.473
$Markov(linked)$	2.761	2.562	2.522	2.502

Table 2. Information content values of the single and multiple source type RBM-based models, their combination, and respective n-gram model counterparts over a range of sequence lengths n.

To illustrate the application of the proposed RBM-based model to multiple viewpoints for music prediction, we combine the single and multiple source type models using mixture-of-experts. This simply involves taking the arithmetic mean of the distributions each of the two models predicts for pitch. The results of this are also illustrated in the third row of Table 2. While the improvement cannot be considered significant in some of the cases, there is reason to pursue this further in the future.

6. CONCLUSIONS & FUTURE WORK

In this paper, we present a distributed model based on the Restricted Boltzmann Machine for music prediction. It is demonstrated how such models can be more efficient and scalable alternatives to n-gram models for simultaneously modelling sequences in multiple musical dimensions. The proposed model is evaluated in comparison with n-gram models and is found to compare favourably with them. It

Markov	Markov*	RBM-2	RBM-3	RBM-4	RBM-5	RBM-6	RBM-7	RBM-8	RBM-9
2.948	2.878	3.059	2.894	2.815	2.771	2.737	2.743	2.734	2.751

Table 1. Comparison of prediction cross entropies in the pitch-only case between n-gram models with a global order bound of 2 (*Markov*), unlimited order (*Markov**) and RBM-based models with n ranging between 2 and 9 (*RBM-2* to *RBM-9* respectively). Overall, the prediction performance of the latter improves until RBM-6, followed by signs of deterioration in some of the datasets.

also makes use of information in longer event sequences than n-gram models. A simple application of the use of the proposed RBM-based model with multiple viewpoints for music prediction framework is also presented here.

In the future, we are interested to further study the performance of the proposed RBM-based model when combined with other musical dimensions such as interval, scale-degree, etc., and to explore more effective ways of combining models. The present model itself can be potentially extended into a deep network, as demonstrated in [9], which is expected to improve its performance. It would be interesting to see how the model can be made to predict multiple target types simultaneously. Also of interest are the applications of this model to MIR tasks such as melodic phrase segmentation and melody classification, and to aid music transcription systems.

7. REFERENCES

[1] Moray Allan and Christopher KI Williams. Harmonising chorales by probabilistic inference. *Advances in neural information processing systems*, 17:25–32, 2004.

[2] Charles Ames. The Markov Process as a Compositional Model: A Survey and Tutorial. *Leonardo*, 22(2):175–187, 1989.

[3] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

[4] John Biles. Genjam: A genetic algorithm for generating jazz solos. In *Proceedings of the International Computer Music Conference*, pages 131–131, 1994.

[5] Miguel A Carreira-Perpinan and Geoffrey E Hinton. On contrastive divergence learning. In *Artificial Intelligence and Statistics*, volume 2005, page 17, 2005.

[6] Darrell Conklin and Ian H Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73, 1995.

[7] David Cope. *Experiments in musical intelligence*, volume 12. AR Editions Madison, WI, 1996.

[8] Tuomas Eerola and Petri Toiviainen. Mir in matlab: The midi toolbox. In *Proceedings of the International Conference on Music Information Retrieval*, pages 22–27. Universitat Pompeu Fabra Barcelona, 2004.

[9] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[10] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18:1527–1554, 2006.

[11] Robert M Keller and David R Morrison. A Grammatical Approach to Automatic Improvisation. In *Sound and Music Computing Conference*, pages 11–13, 2007.

[12] Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted Boltzmann machines. In *International Conference on Machine Learning (ICML)*, pages 536–543. ACM Press, 2008.

[13] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting Structured Data*, 2006.

[14] Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[15] Michael C Mozer. Connectionist music composition based on melodic, stylistic and psychophysical constraints. *Music and connectionism*, pages 195–211, 1991.

[16] Francois Pachet. The continuator: Musical interaction with style. *Journal of New Music Research*, 32(3):333–341, 2003.

[17] Marcus Pearce and Geraint Wiggins. Improved methods for statistical modelling of monophonic music. *Journal of New Music Research*, 33(4):367–385, 2004.

[18] Paul Smolensky. Parallel distributed processing: explorations in the microstructure of cognition, vol. 1. chapter Information processing in dynamical systems: foundations of harmony theory, pages 194–281. MIT Press, Cambridge, MA, USA, 1986.

[19] Petri Toiviainen. Modeling the target-note technique of bebop-style jazz improvisation: An artificial neural network approach. *Music Perception*, pages 399–413, 1995.

[20] Tillman Weyde and Klaus Dalinghaus. Design and optimization of neuro-fuzzy-based recognition of musical rhythm patterns. *Journal of Smart Engineering Design*, 5(2):67–79, 2003.

Paper B: Review

This paper describes a variant on an existing method for predicting pitches, one after another, of a previously unseen melody, using distributions calculated from a database of training melodies. The new aspect of this work is the Restricted Boltzmann Machine (RBM), which replaces the Prediction by Partial Match (PPM) scheme used by Pearce and Wiggins (2004) and Conklin and Witten (1995) for calculation of the probability distributions. The RBM is claimed to be able to take into account greater temporal dependencies than PPM, hence 'distributed'. The link between the calculated probability distributions and pitch prediction involves selecting the pitch associated with the maximum probability in the distribution, at each point in time. In this way, the probability and entropy of pitches in a previously unseen test melody can be calculated, with low cross entropy (the chosen evaluation metric) taken to mean that both 1) the combination of musical dimensions employed to calculate the distributions, and 2) the calculation scheme, were appropriate for capturing aspects of the training music's structure and style. It is also possible to generate melodies based on the training data, again by selecting the pitch associated with the maximum probability in the distribution at each point in time.

Availability of software and data

1. The authors should be given credit for mentioning that a Matlab implementation of the model is available upon request.

Motivation for MIR

2. With regards benefit to the MIR community, I agree with the authors that interesting potential applications of the current work include melody segmentation, classification, and music transcription systems. You should mention these three applications in the introduction rather than the final sentence, to motivate why this paper was submitted to ISMIR, and not to one of the more suitable outlets for work on modeling musical style (e.g., International Conference on Computational Creativity). The transcription application seems a bit tenuous, given there already exist robust monophonic transcription systems. If improving polyphonic transcription is the aim instead, how will your system be adapted to polyphonic input? Telling the MIR community that they need this kind of work in order to improve audio engineering or machine learning solutions is a hard sell, but one that I back. The paper would be improved by following through with an evaluation of one of these potential applications, or for instance showing the incorrect output of a transcription algorithm and the corrections suggested with the current approach.

Selection of viewpoints

3. The current paper, Pearce and Wiggins (2004), and Conklin and Witten (1995) operate on the pitch of melodies; melodies that may be in any one of several keys. Please can you clarify whether melodies in the current paper are transposed prior to running the model? Does leaving the melodies untransposed mean that the MIDI note number sequence (64, 65) would be analysed as one in the same thing, whether the current melody is in C major, F major, etc.? This would lead to 'smearing' of well known scale-degree dependencies and calculated probabilities. For instance, in F major chorale melodies, 64 leads to 65 more often than to any other MIDI note number, but this dependency may be masked by conflating the analysis with C major melodies, where 64 leads to 62 most often, say. I appreciate that some of Conklin and Witten's (1995) viewpoints (e.g., intfref seqint) are transposition-invariant. As far as I can tell, yours are not, and I see this as a fairly serious musical issue.

4. In future work you acknowledge the need to apply the model to other viewpoint combinations. Previous work has established some promising combinations (Conklin and Witten, 1995; Pearce and Wiggins, 2007). Please clarify why one of the most successful combinations from either of the above was not included in the present work also.

Modeling sectional repetition, and nested repetition

5. 'In music, what happens in measure 5 may directly influence what happens in measure 55, without necessarily affecting any of the intervening material' (Cope, 2005, p. 98). You are allowing for a slightly larger temporal distribution of dependency than Pearce and Wiggins (2004) and Conklin and Witten (1995), but nowhere near as much as that alluded to by Cope. Even in your shorter (than 55 bars) melodies there will be sectional repetitions, in the Bach chorales at least (some marked with repeat signs, others not). In terms of alterations to the paper, you do mention extensibility to deep networks, but I think you need to link it to the music-modeling shortcoming that this extension would aim to address (e.g., the Cope quotation). You may be interested in Collins (2011, PhD thesis, Improved methods for pattern discovery in music, with applications in automated stylistic composition) section 9.3 for a description of how to incorporate sectional and nested repetitions into the generation process (accompanying code at www.tomcollinsresearch.net). Also, if you can get hold of it (e.g., British Library), see

Carl Czerny. School of practical composition, volume 3. Robert Cocks & Co, London, UK, 1848. (Year of publication is approximate)

for an interesting historical reference to 'filling' predefined forms such as AABA. For interest's sake I mention it has been shown recently that listeners time-varying emotional repsonses exhibit characteristic behaviour at the beginning of sectional repetitions:

Livingstone, Steven R., Palmer, Caroline, and Schubert, Emery, Emotional response to musical repetition, in Emotion 12(3) (2012), 552-567.

Evaluation

6. The evaluation methodology adheres to Pearce and Wiggins (2004), in that entropy-based measures are employed. I do not think there is any harm in citing Conklin and Witten (1995) at this point as well, as they also use entropy-based measures as an evaluation method.

7. The authors' justification for using entropy-based measures is that it provides an 'application-independent' evaluation. Since the first sentence of your introduction mentions '[w]hether or not computational models can capture elements of style and structure in tonal music', however, why

is a listening study along the lines of Pearce and Wiggins (2001, 2007) not being employed as part of the evaluation? Pearce and Wiggins (2001, Experiment 1) involved asking listeners to distinguish between computer-generated and human-composed melodies, and Pearce and Wiggins (2007) involved listeners rating a mix of such melodies in terms of stylistic success on a 1-7 scale. The Pearce and Wiggins (2001, 2007) listening study methodology could address your introductory question about capturing elements of style and structure, so these need to be mentioned at least as an alternative approach. Recently Collins (2011) combined the methodologies of the aforementioned studies, and also used two groupsconcertgoers and expertsto shed light on the influence of expertise.

8. At the very end of the paper you mention some applications of pitch prediction other than melody generation (which will be what most readers familiar with this literature have in mind). Moving mention of these other applications (melodic phrase segmentation and melody classification, and to aid music transcription systems) to the introduction could make the lack of listening study less conspicuous. It will also mean that the first sentence of Section 5, justifying use of cross-entropy in terms of 'application-independence', refers back to some concrete applications.

9. MIR readers may be uncomfortable with a results section that contains tables of numbers without any mention of standard deviations or significance testing. When you present the mean of the 10 cross-validation folds, why not report the standard deviation as well? I am having trouble telling for instance whether 2.734 for RBM-8 is significantly different from 2.878 for Markov*. The significance of this difference is pivotal for assessing the contribution being made by RBM. Plots with error bars rather than tables would be preferable, facilitating visual comparison and enabling display of metrics for the mean over all datasets, and perhaps one or two of the individual corpora as well.

10. There is a disconnect between the introduction of Table 1 in the text ('average information content values over all datasets are presented in Table 1') and the table caption ('Comparison of prediction cross entropies...'). Please clarify, especially as entropy and cross entropy are introduced in equations (11) and (12), but not information content. Something like this would do: 'IC = log(p), low information content corresponds to a stronger model and vice versa').

11. It is great that the authors try to achieve direct comparison with Pearce and Wiggins (2004), in terms of datasets employed and models included in evaluation. A reference to the model abbreviation in Table 5 of Pearce and Wiggins (2004) that is most directly comparable would be extremely helpful (my guess is Table 5, model C*I, as the value agrees with your Table 1, Markov*): if not, an explanation of why this comparison is not appropriate.

12. Further to previous point, the best-performing model in Pearce and Wiggins (2004) is a combined LTM-STM model, which has a best cross-entropy score of 2.342. This is quite a bit lower than the best RBM cross-entropy, so I am wondering whether the RBM would ever be able to reach this level.

13. Apart from issues raised above, description of the evaluation procedure is generally good, and it is great to see multiple corpora being modelled in parallel.

Music examples

14. In the bad old days of this research area, weak papers finished with a music example or two and said 'this is the kind of music predicted/generated by our system, and it sounds pretty good to us.' Strong papers in this area today offer some kind of quantitative analysis (ANOVA on listening study, stepwise selection on variables calculated from the model output to identify weak features, entropy measures as used by you), *and* some musical examples from the model so that the authors can interpret the quantitative analysis for the reader, shedding light on how the new

modelling attempt leads to an improvement in capturing aspects of style and structure. This paper is somewhere in between, and would be greatly improved by including some discussion of generated music examples.

Extension to polyphonic music

15. Please mention in future work the prospects for applying this method to the prediction of polyphonic music. I am not saying this method in particular has to predict/generate polyphonic output: it is clear to me why models that generate stylistically successful melodies are useful in themselves (a common compositional strategy being to create the melody first, followed by harmonic or contrapuntal development). In future work, however, it would be exciting to see this whole prediction/generation strategy being modelled, not just the first half. (I accept that it may be appropriate to end the folk song modelling at melody-only.)

Suggested cuts in order to address the above comments

While the following sections are interesting and well written, I do not find them to be as important as providing clarifications for the points I made above:

16. Shorten Section 2, giving a two-paragraph summary of multiple viewpoint systems, and referring the reader to Pearce and Wiggins (2004) and Conklin and Witten (1995) for the mathematical details.

17. Shorten the first column of Section 5, removing details about the corpus that you do not require for interpreting results later in the section. Again the details are in Pearce and Wiggins (2004) I think.

18. Paragraph above Figure 2 could go.

Minor corrections

19. Are marked repetitions expanded in your handling of the music data?

20. Last sentence of first paragraph of introduction. You describe Cope (1996) as 'rule-discovery'. 'Recombination of existing music' is more accurate.

21. p. 1 second column second paragraph second line. Use a longer hyphen and without spaces either side. E.g., '—' in latex should do it.

22. p. 2 first column penultimate paragraph fifth line from bottom. 'of two or more of such', the second 'of' is unnecessary.

23. p. 2 first column final paragraph third line from top. 'domains, where', the comma is unnecessary.

24. p. 3 first column top. Something weird going on with latex here, in placing a line of text then the figure. Experiment with `\begin{figure}[t]` to see if this solves the issue.

25. p. 3 first column second line below Figure 1. 'an q r matrix' should read 'a q r matrix'.

26. p. 5 second column second paragraph first line. 'compared te information' should be 'compared the information'.

27 p. 5 second column second paragraph seventh line from bottom. 'on an average' should be 'on average'.

Paper B: Published Version...

A DISTRIBUTED MODEL FOR MULTIPLE-VIEWPOINT MELODIC PREDICTION

Srikanth Cherla[1,2], Tillman Weyde[1,2], Artur d'Avila Garcez[2] and Marcus Pearce[3]
[1]Music Informatics Research Group, Department of Computer Science, City University London
[2] Machine Learning Group, Department of Computer Science, City University London
[3]Centre for Digital Music, Queen Mary University of London
{srikanth.cherla.1, t.e.weyde, a.garcez}@city.ac.uk
marcus.pearce@eecs.qmul.ac.uk

ABSTRACT

The analysis of sequences is important for extracting information from music owing to its fundamentally temporal nature. In this paper, we present a distributed model based on the Restricted Boltzmann Machine (RBM) for melodic sequences. The model is similar to a previous successful neural network model for natural language [2]. It is first trained to predict the next pitch in a given pitch sequence, and then extended to also make use of information in sequences of note-durations in monophonic melodies on the same task. In doing so, we also propose an efficient way of representing this additional information that takes advantage of the RBM's structure. In our evaluation, this RBM-based prediction model performs slightly better than previously evaluated n-gram models in most cases. Results on a corpus of chorale and folk melodies showed that it is able to make use of information present in longer contexts more effectively than n-gram models, while scaling linearly in the number of free parameters required.

1. INTRODUCTION

Sequential structure in music influences our notions of musical style, similarity and the emotions we associate with it. The analysis of sequences in musical scores and equivalent symbolic representations of music is an integral part of Music Information Retrieval, with applications such as music classification [6], computational musicology [26], music creation [19], and music source separation [10]. In the past, this analysis has often been carried out using music generation systems [1, 4, 8, 13, 18].

The present research is based around previous work that adopted ideas proposed in information theory to music [7]. There, *Multiple-viewpoint Systems for Music Prediction* were introduced as a detailed re-interpretation of the key ideas of information theory [22] in music, through an analogy between language and musical style. In that work and what followed [21], Markov models were employed for learning melodic subsequences. While this is a reasonable choice, Markov models are often faced with a problem related to data sparsity known as the *curse of dimensionality*. This refers to the exponential rise in the number of model parameters with the length of the modelled subsequences. Recent research in language modelling has demonstrated that neural networks can be a suitable alternative to more widely used n-gram and variable-order Markov models [2, 5, 17]. There have been some initial results on the success of such models in music [3, 24].

In this paper, we present a model for melody prediction based on one such neural network — the Restricted Boltzmann Machine (RBM) [23]. The choice is motivated by the following. Firstly, the inherent non-linearity of the RBM makes it a suitable candidate for learning complex structures in data, such as those occurring in musical sequences. There exist efficient algorithms for training this model [11, 25]. The RBM, with its straightforward extensibility to deep networks [12], has become a vital building block for creating models that are capable of learning features from the data at multiple levels of abstraction.

We describe here a model for fixed-length subsequences of musical pitch, which compares favourably to n-gram models that were previously evaluated with a prediction task on a corpus of monophonic MIDI melodies [21]. This pitch-only version of the model is then adapted to also make use of note-durations in the melodies, on the same pitch-prediction task. In doing so, we also propose an efficient way to represent this additional information, which takes advantage of the RBM's structure and thus limits model complexity. The structure of the proposed model ensures that it scales only linearly with the length of subsequences to be learned and with the number of symbols in the data. We demonstrate an improvement of results by combining the two models in a manner similar to [7] using the arithmetic mean of their individual probability estimates. An implementation of the model in Python, along with scripts used to generate the results in this paper, are available upon request.

The remainder of this paper is organized as follows. The next section introduces music prediction and multiple viewpoint systems as a framework for music prediction. Section 3 explains the RBM and its discriminative inter-

pretation which make up the basis for the model proposed in this paper. This is followed by a description of the model itself in Section 4. An evaluation of the the model and its comparison with previously evaluated n-gram models is presented in Section 5, followed by discussion on possible directions for future research in Section 6.

2. MUSIC PREDICTION WITH MULTIPLE-VIEWPOINT SYSTEMS

In order to explain music prediction with multiple viewpoints, the analogy to natural language is used here. In statistical language modelling, the goal is to build a model that can estimate the joint probability distribution of subsequences of words occurring in a language L. A statistical language model (SLM) can be represented by the conditional probability of the next word w_T given all the previous ones $[w_1, \ldots, w_{(T-1)}]$ (written here as $w_1^{(T-1)}$), as

$$P(w_1^T) = \prod_{t=1}^{T} P(w_t | w_1^{(t-1)}) \; . \tag{1}$$

The most commonly used SLMs are *n-gram* models, which rely on the simplifying assumption that the probability of a word in a sequence depends only on the immediately preceding $(n-1)$ words [16]. This is known as the Markov assumption, and reduces (1) to

$$P(w_1^T) = \prod_{t=1}^{T} P(w_t | w_{(t-n+1)}^{(t-1)}) \; . \tag{2}$$

Following this approach, musical styles can be interpreted as vast and complex languages [7]. In music prediction, one is interested in learning the joint distribution of *musical event* sequences s_1^T in a *musical language* S. Much in the same way as an SLM, a system for music prediction models the conditional distribution $p(s_t | s_1^{(t-1)})$, or under the Markov assumption $p(s_t | s_{(t-n+1)}^{(t-1)})$. For each prediction, context information is obtained from the events $s_{(t-n+1)}^{(t-1)}$ immediately preceding s_t. Musical events have a rich internal structure and can be expressed in terms of directly observable or derived musical features such as pitch, note duration, inter-onset interval, or a combination of two or more such features. The framework of multiple-viewpoint systems for music prediction [7] was proposed in order to efficiently handle this rich internal structure of music by exploiting information contained in these different musical feature sequences, while at the same time limiting the dimensionality of the models using these features. In the interest of brevity, we limit ourselves to an informal discussion of multiple-viewpoint systems for monophonic music prediction and refer the reader to [7] for the underlying mathematical formulation.

A musical event s refers to the occurrence of a note in a melody. A *viewpoint type* (henceforth written as *type*) τ refers to any of a set of musical features that describe an event. The domain of a *type*, denoted by $|\tau|$ is the set of possible values of that type. A *basic type* is a directly observable or given feature such as *pitch, note duration, key-signature* or *time-signature*. A *derived type* can be derived from any of the basic types or other derived types. A *linked viewpoint type* is created by taking the Cartesian product over two or more types, thus "linking" them.

A *multiple-viewpoint system* (MVS) is a set of models, each of which is trained on subsequences of one *type*, whose individual predictions are combined in some way to influence the prediction of the next event in a given event sequence. Given a context $s_{(t-n+1)}^{(t-1)}$ and an event s_t, each viewpoint τ in an MVS must compute the probability $p_\tau(s_t | s_{(t-n+1)}^{(t-1)})$. While originally n-gram models were proposed to be used with the multiple viewpoints framework, we demonstrate how a distributed model such as the RBM used here can serve as a scalable alternative.

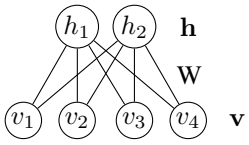

Figure 1. A simple Restricted Boltzmann Machine with four visible, two hidden, and no bias units.

3. RESTRICTED BOLTZMANN MACHINE

The Restricted Boltzmann Machine (RBM) is an undirected graphical model consisting of a set of r visible units $\mathbf{v}$ and a set of q hidden units $\mathbf{h}$. These make up the visible and hidden layers of the RBM respectively. The two layers are fully inter-connected but there exist no connections between any two hidden units, or any two visible units. In its original form, the RBM has binary, logistic units in both layers. Additionally, the units of each layer are connected to a bias unit whose value is always 1.

The edge between the i^{th} visible node and the j^{th} hidden node is associated with a weight w_{ji}. All these weights are together represented in a *weight matrix* $\mathbf{W}$ of size $q \times r$. The weights of connections between visible units and the bias unit are contained in an r-dimensional *visible bias* vector $\mathbf{b}$. Likewise, for the hidden units there is a q-dimensional *hidden bias* vector $\mathbf{c}$. The RBM is fully characterized by the parameters $\mathbf{W}$, $\mathbf{b}$ and $\mathbf{c}$. Figure 1 shows a simple RBM with four visible and two hidden units, without the bias unit to better illustrate its bipartite structure.

The activation probabilities of the units in the hidden layer given the visible layer (and vice versa) are given by the logistic sigmoid function as $p(h_j = 1|\mathbf{v}) = \sigma(c_j + W_{j\cdot}\mathbf{v})$, and $p(v_i = 1|\mathbf{h}) = \sigma(b_i + W'_{i\cdot}\mathbf{h})$ respectively. Due to the RBM's bipartite structure, the activation probabilities of the nodes within one of the layers are independent, if the activation of the other layer is given, i.e.

$$p(\mathbf{h}|\mathbf{v}) = \prod_{j=1}^{q} p(h_j|\mathbf{v}) \tag{3}$$

$$p(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{r} p(v_i|\mathbf{h}) \; . \tag{4}$$

The RBM is a special case of the Boltzmann Machine, which is an energy-based model for representing probability distributions [15]. In such energy-based models, probability is expressed in terms of an energy function. In the case of the RBM, this function is expressed as

$$Energy(\mathbf{v}, \mathbf{h}) = -\mathbf{b}^\top \mathbf{v} - \mathbf{c}^\top \mathbf{h} - \mathbf{h}^\top \mathbf{W} \mathbf{v} \,. \quad (5)$$

Learning in energy-based models can be carried out in a *generative* fashion, by updating the weights and biases in order to minimize the overall energy of the system with respect to the training data. This amounts to maximizing the log-likelihood function of the joint probability distribution $p(\mathbf{v})$, which is given by

$$p(\mathbf{v}) = \frac{e^{-FreeEnergy(\mathbf{v})}}{Z}, \quad (6)$$

with $Z = \sum_{\mathbf{v}} e^{-FreeEnergy(\mathbf{v})}$, where

$$FreeEnergy(\mathbf{v}) = -\log \sum_{\mathbf{h}} e^{-Energy(\mathbf{v},\mathbf{h})} \,. \quad (7)$$

While computing the exact gradient of the log-likelihood function for $p(\mathbf{v})$ is not tractable, an approximation of this gradient called the Contrastive Divergence (CD) gradient has been found to be a successful update rule for training RBMs [11]. With the CD update, the RBM can be trained efficiently.

The RBM described above models the joint probability $p(\mathbf{v})$ of the set of visible units $\mathbf{v}$. However, as described in Section 2, we are interested in a conditional distribution of the form $p(y|\mathbf{x})$. It has been demonstrated in [14] how an RBM can be used for a *discriminative* task such as classification. The posterior class probability distribution of such an RBM has the form

$$p(\mathbf{y} = \mathbf{e}_c|\mathbf{x}) = \sum_{\mathbf{h}} p(\mathbf{y} = \mathbf{e}_c, \mathbf{h}|\mathbf{x}) \quad (8)$$

$$= \frac{e^{-FreeEnergy(\mathbf{x},\mathbf{e_c})}}{\sum_{c'=1\ldots C} e^{-FreeEnergy(\mathbf{x},\mathbf{e}_{c'})}} \quad (9)$$

where $\mathbf{x}$ is the input vector, and $\mathbf{y}$ is a vector that is a *1-of-C* representation of the class (also known as *one-hot* encoding), with C being the number of classes. If $\mathbf{x}$ belongs to a class c, then $\mathbf{y} = \mathbf{e}_c$, where $\mathbf{e}_c$ is a vector with all values set to 0 except at position c. With respect to the RBM, $\mathbf{x}$ and $\mathbf{y}$ together make up the visible layer $\mathbf{v}$.

Assuming a training set $\mathcal{D}_{train} = \{(\mathbf{x}_i, y_i)\}$ where $\mathbf{x}_i$ and $y_i \in \{1, \ldots, C\}$ are the i-th input vector and target class respectively, training the RBM generatively involves minimizing the negative log-likelihood

$$\mathcal{L}_{gen}(\mathcal{D}_{train}) = -\sum_{i=1}^{|\mathcal{D}_{train}|} \log p(\mathbf{x}_i, \mathbf{y}_i) \,. \quad (10)$$

The RBM thus used in a discriminative manner, forms the basis of the prediction model described in the next section.

4. A DISTRIBUTED MODEL FOR USE WITH MULTIPLE VIEWPOINTS

The prediction model we present in this paper models the conditional distribution $p(s_t|s_{(t-n+1)}^{(t-1)})$. It places no restrictions on the *types* associated with events in the context $s_{(t-n+1)}^{(n-1)}$ (*input type*), or the predicted event s_t (*target type*). In the simplest case, both are the same. In the case where they are different, the performance of the model depends on how informative the input types are of the target type. In the present work, we demonstrate this model with two cases where (1) both the input and target viewpoint types are musical pitch, and (2) the input types are pitch and duration, and the target type pitch. The choice of the additional input type in the second case was motivated by simplicity and to lay emphasis on the representation.

For each monophonic melody (in MIDI format) in a given dataset, sequences of the relevant input and target types are first extracted using the MIDI Toolbox [9]. These values are encoded as binary 1-of-$|\tau|$ vectors, where $|\tau|$ is the size of the domain of type τ. In the case where more than one input type exists, their corresponding vectors are simply concatenated. Such an idea is similar to that of the linked viewpoint type proposed in [7]. There are however, two important distinctions between the two. Firstly, the input and target types must be identical in the case of the n-gram models originally proposed for use with multiple-viewpoint systems, whereas this is not a requirement for the RBM model. Secondly, a linked viewpoint between two arbitrary types τ_1 and τ_2 of domain sizes $|\tau_1|$ and $|\tau_2|$ respectively, would have a domain of size $|\tau_1| \times |\tau_2|$ in the case of the n-gram models. Thus, for subsequences of length n, the number of free parameters to be estimated are $(|\tau_1| \times |\tau_2|)^n$ in the worst case. In contrast, the number to be estimated in case of the RBM model, with q hidden units and r visible units, is $(q \times r) + q + r$, where $r = (n-1) \times [(|\tau_1| + 1) + (|\tau_2| + 1)] + |\tau_3|$, and τ_3 the target type. The additional visible unit added to the representation of each of the input types τ_1 and τ_2 in the context is 1 when the corresponding event is absent at the start of a melody. Such a model only scales linearly with the length of the learned subsequences as well as the domain size of each of the involved viewpoint types (assuming q is constant). Its structure is depicted in Figure 2. Here we considered only those cases with a single target type.

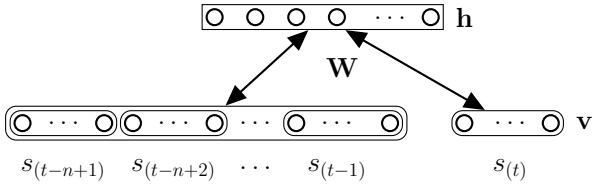

Figure 2. The structure of the prediction model. The set of nodes in the visible layer grouped together on the left make up the context $s_{(t-n+1)}^{(t-1)}$ of the input type(s). The set of nodes $s_{(t)}$ to the far right corresponds to the target type.

To train the model generatively, a subsequence $s^t_{(t-n+1)}$ is clamped to all the nodes in the visible layer. Training is done using the first instantiation of the Contrastive Divergence learning algorithm (CD-1). This simply means that the model parameters are updated after a single step of Gibbs sampling [11]. During prediction, the probability of each of the possible pitches in the prediction space is determined using (9). The distribution generated in this way does not require any kind of smoothing operation for unseen subsequences unlike n-gram models, where in [21] an empirical evaluation of different smoothing techniques was found necessary to establish the most reliable one.

5. EVALUATION

In order to evaluate the proposed prediction model, we make a comparison to a previous study of n-gram models for music prediction in [21]. There, *cross-entropy* was used to measure the information content of the models. This quantity is related to *entropy*, which is defined as

$$H(p) = -\sum_{s \in S} p(s) \log_2 p(s) \; . \tag{11}$$

where $p(s \in S) = p(\chi = s)$ is the probability mass function of a random variable χ distributed over a discrete alphabet $S = \{s_1, \ldots, s_k\}$ such that the individual probabilities are independent and sum to 1. The value of entropy, with reference to a prediction model, is a measure of the uncertainty of its predictions. A higher value reflects greater uncertainty. In practice, one rarely knows the true probability distribution of the stochastic process and uses a model to approximate the probabilities in (11). An estimate of the goodness of this approximation can be measured using cross-entropy (H_c) which represents the divergence between the entropy calculated from the estimated probabilities and the source model. This quantity can be computed over all the subsequences of length n in the test data $\mathcal{D}_{test}$, as

$$H_c(p_{mod}, \mathcal{D}_{test}) = \frac{-\sum_{s_1^n \in \mathcal{D}_{test}} \log_2 p_{mod}(s_n | s_1^{(n-1)})}{|\mathcal{D}_{test}|} \tag{12}$$

where p_{mod} is the probability assigned by the model to the last pitch in the subsequence given its preceding context. Cross-entropy approaches the true entropy as the number of test samples ($|\mathcal{D}_{test}|$) increases.

Evaluation was carried out on a corpus of monophonic MIDI melodies that cover a range of musical styles. The corpus is a collection of 8 datasets containing a total of $54,308$ musical events and was also used to evaluate n-gram models for music prediction in [21]. There, two different models were evaluated both individually and in combination. The first of these was a Long-Term Model (LTM), that was governed by structure and statistics induced from a large corpus of sequences from the same musical style. And the other was a Short-Term Model (STM) which relied on structure and statistics particular to the melody being predicted. The prediction model presented here deals only with long-term effects that are induced from a corpus, and is thus compared with the two best performing LTMs in [21] of unbounded order (labelled there as *C*I*) and order bound 2 respectively. To facilitate a direct comparison between the two approaches, the melodies are not transposed to a default key.

For the RBM model, different hyperparameters were evaluated through a grid search over the learning rate $\lambda = \{0.01, 0.05\}$, the number of hidden units $n_{hid} = \{100, 200, 400\}$, and the weight-cost $w_{cost} = \{0.0001, 0.0005\}$. Each model was trained using mini-batch gradient descent over 500 epochs with a batch size of 100 samples. The momentum μ, was set to 0.5 during the first five epochs and then increased to 0.9 for the rest of the training. Each model was evaluated with 10-fold cross-validation.

We carry out three types of evaluation. The first measures the information content of the pitch-only version of the proposed model using cross-entropy, and compares it to the n-gram models of [21]. It was observed that the RBM model compares favourably with the best of the n-gram models by making better use of information in longer contexts. In the second evaluation, we compare a variant of the model with input types pitch and duration and target type pitch to its pitch-only counterpart. And lastly, we combine these two models using mixture-of-experts and demonstrate how this can further improve the model performance in comparison to the individual models.

The first evaluation is carried out with cross-validation separately for each of the individual datasets. The context length is varied between 1 and 8. It was found that the RBM models with context length greater than 2 perform better than corresponding n-gram models on average. This is illustrated in Figure 3. An RBM model of suitable context length perform marginally better than the best-performing n-gram model — that of unbounded order. The same is the case with the best bounded-order n-gram model (of context length 2) and the RBM model of the same context length. While it was found that the performance of bounded order n-gram models tends to worsen on further increasing the context length, the performance of RBM models continues to improve until a context length of 4. The value of n where the RBM model performs better than the n-gram models of unbounded order is different on different datasets, and typically occurs between $n = 3$ and $n = 7$. The best average model cross-entropy of 2.819 is reached for a context length of 4. For models using longer contexts an increase in training performance was accompanied by a slight worsening of test performance, indicating overfitting. We suspect that the overall performance of the RBM models can be further improved with an optimized grid-search strategy in the hyper-parameter space, but leave this to be explored in the future. The optimal number of hidden units in our search was 100 across all datasets for almost all context lengths, leading to a linear increase in model size with context length.

In the second evaluation, we compared the cross-entropies of the single and multiple input type models (pitch and pitch with duration respectively) using the same target type (pitch), on the Bach chorale subset of the corpus. The re-

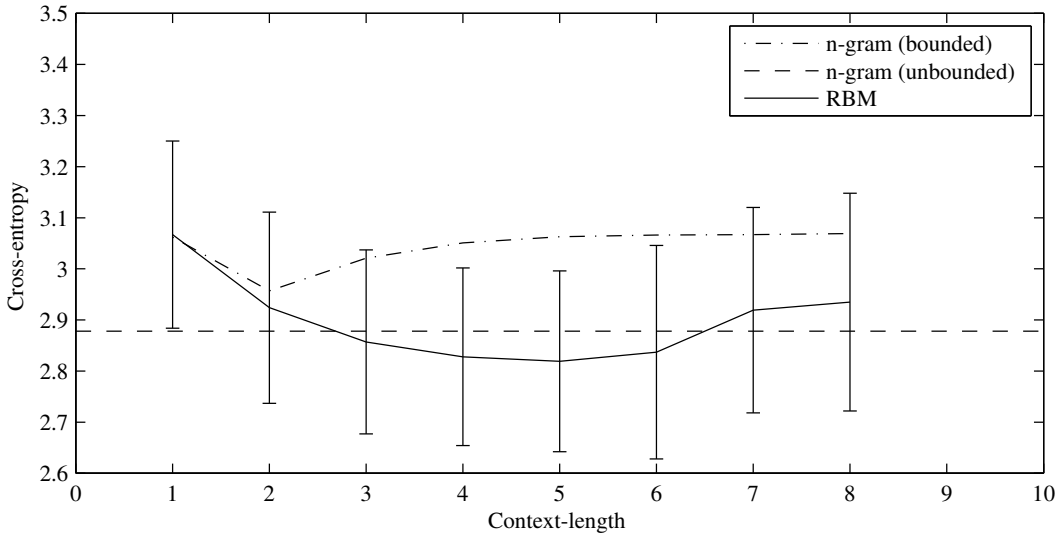

Figure 3. Variation in average cross-entropy of the prediction models with context length l (with standard deviation across folds for the RBM model). The cross-entropy of the RBM models progressively decreases until $l = 4$, while that of the n-gram models evaluated in [21] is minimal at $l = 2$ and increases thereafter. The performance of the n-gram model of unbounded order is indicated by the dashed line.

sults are shown in Table 1. The choice of adding duration was motivated by simplicity but the results show that it was not ideal for improving predictions. This conclusion is also supported by a similar trend observed with n-gram models, where a small deterioration in performance was observed on adding duration. The RBM model shows small performance improvements for some context lengths. This indicates that the representation for multiple input types proposed in Section 4 as an alternative to the linked viewpoints may indeed be effective.

l	1	2	3	4
n-gram (p)	2.737	2.565	2.505	2.473
n-gram (p + d)	2.761	2.562	2.522	2.502
RBM (p)	2.698	2.530	2.490	2.470
RBM(p + d)	2.660	2.512	2.481	2.519
RBM (combined)	2.663	2.486	2.462	2.413

Table 1. Cross-entropies of the single (pitch) and multiple (pitch, duration) input type RBM models and their combination over a range of context lengths l on the Bach chorales dataset. The individual RBM models compare favourably with corresponding n-gram models.

To illustrate the application of the proposed RBM model to multiple viewpoints for music prediction, we combine the pitch-only and the pitch & duration models. We use a simple mixture-of-experts model, i.e., take the arithmetic mean of the distributions each of the two models predicts for pitch. The results of this are listed in the third row of Table 1 and show an improvement over individual models.

6. CONCLUSIONS & FUTURE WORK

We presented a distributed model based on the Restricted Boltzmann Machine for multiple-viewpoint music prediction. It was demonstrated how such a model can be a scalable alternative to n-gram models for simultaneously modelling sequences of multiple musical features. The proposed model was evaluated in comparison with n-gram models and was found to compare favourably with them. It is able to make better use of information in longer event contexts than n-gram models, and also scales linearly with context length.

In the future, we would first like to address some of the issues left open in the present research. These include experiments with more promising viewpoint-type combinations as reported in [7] and [20], the use of alternative data fusion techniques like the weighted mixture- and product-of-experts [20], and further optimization of the existing model parameters. Previous research suggests that combining the LTM and STM improves prediction performance [7, 20] and, in fact, the combined n-gram model reported in [20] (mean cross-entropy: 2.479 for all datasets; 2.342 for the chorale dataset) outperforms the long-term RBMs examined here. Given the improved performance of these long-term RBMs, we expect adding a short-term component will yield the best prediction performance yet observed for this corpus. Extensions of the present model to handle polyphony and higher-level musical structure will also be explored. We would also like to apply the prediction model described here to some of the MIR tasks listed in Section 1. The present model can be potentially extended into a deep network, as demonstrated in [11], which is expected to improve its performance further.

7. ACKNOWLEDGEMENTS

Srikanth Cherla is supported by a Ph.D. studentship from City University London. The authors would like to thank Son Tran for many useful discussions on RBMs, and the reviewers for their valuable feedback on the paper.

8. REFERENCES

[1] Charles Ames. The Markov Process as a Compositional Model: A Survey and Tutorial. *Leonardo*, 22(2):175–187, 1989.

[2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

[3] Greg Bickerman, Sam Bosley, Peter Swire, and Robert Keller. Learning to Create Jazz Melodies using Deep Belief Nets. In *International Conference On Computational Creativity*, 2010.

[4] John Biles. Genjam: A genetic algorithm for generating jazz solos. In *Proceedings of the International Computer Music Conference*, pages 131–131, 1994.

[5] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[6] Darrell Conklin. Multiple viewpoint systems for music classification. *Journal of New Music Research*, 42(1):19–26, 2013.

[7] Darrell Conklin and Ian H Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73, 1995.

[8] David Cope. *Experiments in musical intelligence*, volume 12. AR Editions Madison, WI, 1996.

[9] Tuomas Eerola and Petri Toiviainen. MIR in Matlab: The Midi Toolbox. In *Proceedings of the International Conference on Music Information Retrieval*, pages 22–27. Universitat Pompeu Fabra Barcelona, 2004.

[10] Joachim Ganseman, Paul Scheunders, Gautham J Mysore, and Jonathan S Abel. Evaluation of a Score-informed Source Separation System. In *International Society for Music Information Retrieval Conference (ISMIR)*, pages 219–224, 2010.

[11] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[12] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18:1527–1554, 2006.

[13] Robert M Keller and David R Morrison. A Grammatical Approach to Automatic Improvisation. In *Sound and Music Computing Conference*, pages 11–13, 2007.

[14] Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted Boltzmann machines. In *International Conference on Machine Learning (ICML)*, pages 536–543. ACM Press, 2008.

[15] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting Structured Data*, 2006.

[16] Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[17] Andriy Mnih and Geoffrey E Hinton. A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pages 1081–1088, 2008.

[18] Michael C Mozer. Connectionist music composition based on melodic, stylistic and psychophysical constraints. *Music and connectionism*, pages 195–211, 1991.

[19] Francois Pachet. The continuator: Musical interaction with style. *Journal of New Music Research*, 32(3):333–341, 2003.

[20] Marcus Pearce. *The Construction and Evaluation of Statistical Models of Melodic Structure in Music Perception and Composition*. PhD thesis, 2005.

[21] Marcus Pearce and Geraint Wiggins. Improved methods for statistical modelling of monophonic music. *Journal of New Music Research*, 33(4):367–385, 2004.

[22] Claude E. Shannon. A Mathematical Theory of Communication. *The Bell System Techincal Journal*, 27(July):379–423, 623–656, 1948. Reprinted in *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[23] Paul Smolensky. Parallel distributed processing: explorations in the microstructure of cognition, vol. 1. chapter Information processing in dynamical systems: foundations of harmony theory, pages 194–281. MIT Press, Cambridge, MA, USA, 1986.

[24] Athina Spiliopoulou and Amos Storkey. Comparing probabilistic models for melodic sequences. In *Machine Learning and Knowledge Discovery in Databases*, pages 289–304. 2011.

[25] Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071. ACM, 2008.

[26] Raymond Whorley, Christophe Rhodes, Geraint Wiggins, and Marcus Pearce. Harmonising melodies: Why do we add the bass line first? In *International Conference on Computational Creativity*, pages 79–86, 2013.