A Prototype for Exploration of Computational Strangeness in the Context of Rhythm Variation

Peter Knees
Dept. of Computational Perception
Johannes Kepler University Linz, Austria
peter.knees@jku.at

Kristina Andersen
Studio for Electro Instrumental Music (STEIM)
Amsterdam, the Netherlands
kristina@steim.nl

ABSTRACT

In this paper, we follow up on the recently emerged idea of "computational strangeness," which represents algorithmic recommendations as artistic obstructions in creative work. The concept of "strangeness" stems from a process of user engagement in the context of a collaboration between Human Computer Interaction (HCI) and Music Information Retrieval (MIR), in which professional makers of Electronic Dance Music requested tools that abandon the "more-of-the-same" paradigm of similarity-based retrieval in favor of the possibility to get serendipitous, opposing results to stimulate the creative process. We describe a prototype that allows the user to explore the space of dissimilarity for the case of rhythmic patterns by means of a simple dial. This "strangeness dial" allows the gradual adjustment of the degree of desired otherness. We test this initial prototype in a questionnaire-based user study to inform future developments. A central outcome is that dissimilarity-based notions, such as "otherness" and "strangeness", — even more than similarity — are highly subjective concepts that can not be addressed without strategies for personalization.

Categories and Subject Descriptors

H.1.2 [**User/Machine Systems**]: Human Factors; H.3.3 [**Information Search and Retrieval**]: Information filtering

Keywords

computational strangeness; similarity; serendipity; rhythm variation

1. INTRODUCTION AND CONTEXT

Research at the intersection of Human Computer Interaction (HCI) and Music Information Retrieval (MIR) with the goal of improving the tools used by musicians and music producers is the core objective of the EU-funded GiantSteps project [7]. In the context of this project, we engage with music practitioners in order to learn about the processes involved in professional music creation, thereby allowing them a strong peer position in the conceptualization and evaluation of new music interfaces [3]. One of the recent outcomes of this engagement was the concept of *computational strangeness*, i.e., the notion of the computer providing opposition or essentially artistic obstruction to the work process [8, 4].

The ideas of opposing systems, "otherness," and obstruction (subsumed under the term computational strangeness) came up in a number of interviews with musicians, driven by a general question about how sample recommendation could add to their work flow:

> "So if I set it to 100% precise I want it to find exactly what I am searching for and probably I will not find anything, but maybe if I instruct him for 15% and I input a beat or a musical phrase and it searches my samples for that. That could be interesting." USER003

> "What I would probably rather want it to do is make it complex in a way that I appreciate, like I would be more interested in something that made me sound like the opposite of me, but within the boundaries of what I like." USER007

> "I'd like it to do the opposite actually, because the point is to get a possibility, I mean I can already make it sound like me, it's easy." USER001

These quotes suggest that our users would like computers not to provide them with collaborative or content-based recommendations that follow a nearest-neighbor retrieval approach, but instead system qualities that go beyond pure accuracy, such as diversity, novelty, serendipity, or unexpectedness, e.g., [1, 2, 5]. In the context of seeking inspiration for a creative process, "more-of-the-same" recommendations have basically no relevance. Hence, we need to explore the "long tail" of the similarity space, i.e., the space of potentially dissimilar items.

In this paper, we propose an initial prototype to retrieve and explore "strange" results based on similarities in a feature space. For the task of rhythm variation, i.e., controlled retrieval of rhythmic patterns based on a seed pattern (see also [12]), we build a user interface for browsing a collection using a simple (hardware) dial that allows the gradual adjustment of the degree of desired otherness. We test this initial prototype in a questionnaire-based user study to inform future developments.

UMAP 2016 Extended Proceedings – Workshop on Surprise, Opposition, and Obstruction in Adaptive and Personalized Systems (SOAP) July 16, 2016, Halifax, NS, CA
.

2. RELATED CONSIDERATIONS

In the field of Computer Science, the concept of "strangeness" is related to concepts like serendipity and diversity [5] and even more to unexpectedness [1] in consumer-oriented recommender systems in that the item space and its niches need to be explored in order to find desired results. However, while end consumers of, e.g., movies, might be more or less inclined to watch something different and consider it a satisfactory result, music composers and other creatives want to be challenged, possibly by being opposed. This includes the possibility of having a first negative opinion in order to spark a creative streak.

We are not the first to consider opposition in relation to creative electronic music making. In a narrow context of experimental music creation, Collins has addressed the question of opposition in the "Contrary Motion" system [6]. The system opposes a piano player's rhythm in real time by tracking onsets and predicting the metrical structure from a number of hypotheses. Based on the predicted structure of the performance, an oppositional structure is constructed. To this end, periodicity (reciprocal of tempo) and beat position histograms are built from the hypotheses and positions in the most sparse regions of these histograms are chosen as opposing candidate. This should determine a location in the space of actions "where the human is likely not to be" [6]. However, this is a simplistic approach as the author points out himself: "Time series analysis is not used to any great degree at present, and the prior or online construction of a database of musical materials with respect to which 'maximal dissimilarity' can be explored may involve porting work from music information retrieval." Nevertheless, the working hypothesis underlying the system is that being confronted with an oppositional music style can be stimulating for a musician. The evidence collected through our involvement with expert users further supports this hypothesis.

In terms of acoustic retrieval, an early example of a content-based system is *SoundFisher* [13], which lets the user query a database by combining examples with tags and other metadata. In addition to using the underlying similarity measure for retrieval of similar sounds, the user is also given the opportunity to retrieve the least similar sounds. While the effects of this control are not further elaborated on, it shows that the concept of finding "the opposite" is present starting from the beginnings of multimedia retrieval (especially with a focus on creative applications). We assume that the reason for this not being further explored lies in the difficulty of defining "the opposite" in a constructive manner.

While similarity of data items is usually well defined based on a metric (e.g., the nearest neighbours), the concept of dissimilarity is more difficult to grasp as there are many possibilities, particularly in high-dimensional feature spaces where the number of points with comparable distances (however, in different directions, making them mutually dissimilar) can increase drastically. As a consequence, rankings according to distance become somewhat meaningless (or at least non-trivial to interpret) at higher ranks. Furthermore, even in a defined space, "strange", "different", or "the opposite" is typically not easily obtainable, e.g., by simple "inversion" of a feature representation, as this might not be defined or perceived in a musical fashion. We acknowledge this problem and, for our initial prototype, aim at circumventing it by our working definition of strangeness as "expectable otherness" or "consistent dissimilarity" as detailed in the next section. Regardless of the shortcomings of this definition, we utilize the prototype built upon this concept with the intention to gain meaningful insights for future developments.

3. A FUNCTIONAL PROTOTYPE FOR CONTROLLING STRANGENESS

To begin testing the notion of controlled strangeness, we focused on one particular aspect of music, i.e., rhythm, and built a drum pattern browsing tool in Java. Based on a query that can be entered in a binary grid of 16 steps of four instruments (kick, snare, hihat, and open hihat) variations of this pattern are searched for in a database of about 8,000 electronic music drum patterns used in DJ production software. Alternatively, a random pattern from the database can be selected as a starting point for a query. In addition, a hardware dial and/or software slider can be used to control the degree of otherness of the retrieved variations.

We implemented two slightly different methods of "otherness" allowing the user to switch between them. In the prototype they were named "non-similarity" and "strangeness". Investigating a pattern updates the pattern grid and plays it in a loop at a constant speed of 120 BPM. The use of a hardware controller should allow for immediate control of the parameters in real time, i.e., while the controls are used, the played back looped pattern changes accordingly, resulting in a direct feedback and also allowing the user to "play" the browsing interface as an instrument.

A screen-shot of the software interface and a picture of the hardware controller can be seen in Figure 1. The following describes the technical implementation of the interface in more detail.

3.1 Prototype Implementation

Retrieving patterns from the database is based on calculating modified Euclidean distances between the query pattern in the grid (a $16 \times 4 = 64$-dimensional binary vector) and all 64-dimensional patterns in the database, and selecting those patterns with the smallest distance (nearest neighbours). The modification applied to the Euclidean distance gives twice as much emphasis to kick and snare activations as to hihat and open hihat in order to account for perceived dominance. In the interface, the patterns with the smallest distance are displayed as a list of IDs that the user can explore via mouse clicks and through a hardware wheel.

In order to enable the user to not only find similar but also "other" patterns, and to give control over this process, a slider that ranges from "similar" to "non similar" and that is controllable via a hardware dial is introduced. While moving down the list of variations results in exploring the next neighbors based on the distance function, moving this slider towards "non similar" yields bigger "jumps" in the distance space, thus gradually moving away from variations of the query to the most dissimilar. More precisely, the dial allows to shift the search space to farther "non similar" regions in 128 steps, each internally moving the result list 63 positions further down. This plain dissimilarity function satisfies the idea of finding and accessing "the other", however this might lead to results that are more perceived as random for the reasons discussed in the previous section. A simple order based on the similarity value will thus eventually mix examples from different directions making it difficult to grasp the current area in the similarity space. In fact, this way, the

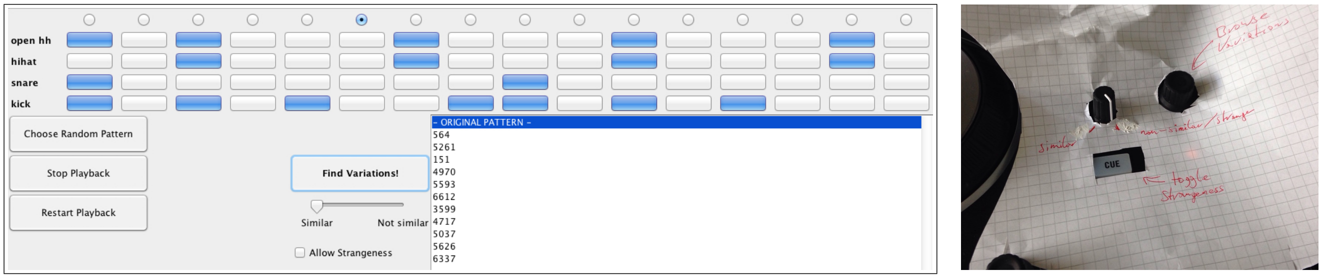

Figure 1: The user interface used in phase 3. Left: the software interface with the drum pattern grid on top and the variation list on the bottom right. Right: a Native Instruments Traktor controller covered with paper and labelled to match the controls in the software interface, i.e., a dial ranging from "similar" to "non similar/strange", a wheel to browse the list of variations on the currently set area of otherness, and a button to toggle between the two types of otherness.

retrieval process is not moved to a different area, but just increases its range, incorporating most diverse results with potentially the only commonality that they have a similar distance to the seed point. We want to address this by providing a more consistent experience when exploring distant regions. We argue that utilizing the same similarity function and space can be beneficial for defining "different" as long as results remain consistent and the expectation of distance is met when browsing farther regions.

For this initial test, this behaviour of "expectable otherness" or "consistent dissimilarity" is our working definition for "strangeness". In order to achieve this, we perform a pre-clustering in the feature space (we use a simple k-means clustering with k set to the number of different settings on the dial, i.e., 128). When retrieving patterns, first, a ranked list of cluster centers based on their distance to the seed pattern is generated. Then, for the final result list, the patterns associated to the closest cluster center are ranked, followed by the ranking of patterns associated to the second-closest cluster center, etc. Thus, the final list is built by first ordering the clusters based on distance as a coarse structure and then the individual examples based on distance within these cluster segments.

We expect this to allow for a more consistent experience when browsing the space and to give "meaning" to the concept of difference when turning the dial to the "strange regions". This second approach to otherness can be enabled by toggling the check box "Allow strangeness", which also changes the labels on the slider to range from "similar" to "strange".

3.2 Experimental Setup

We tested the prototype in a user session with 12 expert users, who each spent up to an hour executing a number of tasks. The test setup consisted of a desktop running the software interface, a Native Instruments Traktor DJ controller covered in paper allowing only two dials and a button to be used (cf. Figure 1 on the right), and a 4 page document with task instructions and a questionnaire section accompanying each task. For the test, in every step, we show at most 11 pattern variations in the list to not distract the user from the given tasks.

The document starts by stating:

> "In front of you is the Pattern Browser which represents a drum step sequencer. You can enable/disable four types of drums along a grid of 16 steps to create a drum pattern (press "Restart Playback" button to play). Alternatively, you can select a random, predefined pattern (Button "Choose Random Pattern"). If you click the button "Find Variations!" a list of similar drum patterns will be displayed."

Then follow task descriptions like:

> "Selecting the variations in the list can also be done using the hardware controller in front of you, using the "Browse Variations" dial. Find variations to a pattern (the prior or a new one) and browse using the dial while having the playback enabled."

Underlying, the document is structured into three (concealed) phases: (1) Exploration of the pattern variation browsing tool, (2) Exploration of the setting range from "similar" to "non-similar," and (3) Exploration of the setting range from "similar" to "strange". All three phases contain the same statements with regard to browsing experience, likelihood of usage in a composition or live performance setting, perceived otherness, and consistency in the variations (see below), which the user can agree or disagree with on a Likert-type scale from 1 (no agreement) to 5 (complete agreement). Furthermore, we give the opportunity to add additional free-form statements as the test progresses and ask for feedback specific to each phase.

3.3 Results

All 12 users managed to execute all tasks in the questionnaire. The interface was overall considered consistent and useful, which can be seen in the responses to the following Likert statements (values in parentheses indicate *mean* ± *std*):

- "I get a consistent sound impression while browsing" [similar variations using the wheel] (4.58±0.90)
- "The list of variations is useful to me if I'm trying to find a new sound." (4.08±0.90)
- "The further I turn the 'strangeness dial' to the right, the more dissimilar the patterns sound from the original." (4.08±1.08)

- "The further I turn the 'similarity dial' to the right, the more dissimilar the patterns sound from the original." (4.00±1.21)

This indicates that the system can be used to find variations and that the underlying distance metric is useful for capturing both similarity and dissimilarity. When we look at the items targeting perceived consistency in both types of otherness and those that explicitly ask for a comparison, results are less conclusive, evidenced by mean values around 3 and high standard deviation values:

- "The list of variations is less consistent towards 'non-similar' than towards 'similar'." (3.25±1.60)
- "The list of variations is equally consistent towards 'non-similar' and towards 'similar'." (2.75±1.60)

and in the statements that compare the two approaches:

- "There are less inconsistencies on the variation lists when using the 'strangeness dial' than when using the 'similarity dial'." (2.92±1.56)
- "'Strangeness' gives me a better browsing experience than 'non-similarity'." (3.33±1.50)
- "'Strangeness' gives me a more consistent and expectable browsing experience than 'non-similarity'." (3.17±1.40)

The high variance in responses is also reflected in the statements given by participants when asked what "strangeness" is or does in their opinion:

> "Non-similarity provides smoother transitions from the original song but doesn't provide a list of consistent songs, when it is on the non-similar side. Strangeness provides a more similar 'strange' list." UI_02

> "I think with strangeness a set of most conceivable types of base beats/rhythms has been determined, and organised as most similar/dissimilar to the original beat. The list of variations is then in tune with the different beat archetypes. With the similar/dissimilar dial, I think the list is composed of selections of different archetype lists" UI_10

> "The strangeness dial seems to filter the list of variations to provide the more dissimilar patterns. It also seems to provide smoother and more reliable transitions between patterns and variations. It seems to retain the predominant beats in the pattern." UI_12

> "I am not sure I can find a difference with 'non-similarity'. I just feel it changes the pattern a bit too strangely/abruptly." UI_04

> "I have no idea! It's just weird for me!" UI_03

> "The strangeness dial inspires to think outside of musical definitions and enhances a more experimental feeling. On the other hand, I believe it is too inconsistent for live usage. I can see how it could challenge composers to think more complex, also when live performing, but to lay down an improvisation based on strangeness can be either super good or super bad." UI_09

This tells us that for some participants the effect of higher consistency among strange results was obvious, while others didn't see any difference or preferred the non-similar approach. For users that noticed the difference, it was an important enhancement in finding patterns and controlling the degree of otherness. For those who did not notice a difference, the presence of the option of "strangeness" was more confusing than inspiring. With regard to usage and trust in musical practice, the feedback was rather positive for composition (non-similarity: 3.75±1.06; strangeness: 3.67±1.07) and heterogeneous for live performance (non-similarity: 3.00 ±1.41; strangeness: 2.67±1.37).

4. DISCUSSION

While the users were overwhelmingly positive about the possibility to control the level of dissimilarity or strangeness, it was clear that not all were able to distinguish between the two types present in the test interface that we defined as initial possible interpretations of the concept. We can attribute some of the less positive experiences to the fact that we were operating on a given database of electronic music patterns. Thus, hand-crafted patterns that were not akin to these styles did not have corresponding entries and as a result the system behavior was unsatisfactory, by returning something unrelated rather than actually rendering the query "strange". It is also worth noting that variation needs to be noticeable at different levels, i.e., we saw that where the dial did not behave as expected (a little move should correspond to a little change), people were dissatisfied with the whole process and could not relate to complete dissimilarity/strangeness (which ideally would be perceived as opposition).

However, the most important — and probably also obvious — finding of the study is that the expectation of a concept like strangeness is highly subjective (something we couldn't account for in this first study), and that, related to this finding, there is different preference on how much and what should be varied.

As we had to restrict the first test prototype to one specific musical property and defined strangeness as "more consistent dissimilarity" according to Euclidean distance, this was unlikely to fulfill the expectations of a strangeness dial for all users. Therefore, despite the confirmation that the possibility to make use of controlled "anti-recommendation" in a search process is a desired feature, we also find that accounting for individual preference in terms of otherness presents the central challenge for such an interface. For some users, the expected effect of applying strangeness is a matter of dimensions, parameters, or musical aspects, while others simply acknowledged that there are many ways of interpreting this concept, as is also evidenced by statements made in the conceptual phase:

> "No, it should be strange in that way, and then continue on in a different direction. That's the thing about strange, that there's so many variations of strange. There's the small, there's the big, there's the left, there's the right, up and down." STR_006

> "Strangeness of genre maybe, how different genre you want. [...] It depends how we chart the parameter of your strangeness, if it's tim-

bre or rhythm or speed or loudness, whatever." STR_001

> "In synth sounds, it's very useful [...] Then the melody can also be still the same, but you can also just change the parameters within the synthesizer." STR_003

These individual notions of the concept of strangeness, lead directly to ideas around personalization and the requirement to build such a concept around user models, potentially including personality traits, e.g., [14, 11], as the degree of opposition and strangeness desired depends on the personality and the working style of the music maker.

5. CONCLUSIONS

We have investigated the acceptance and appropriateness of a prototype embodying a working definition of strangeness, a desired concept for creative work discovered through interactions with music practitioners. While there is a clear request for having such a functionality of "otherness" or even "opposition" to serve as some sort of alter ego to reflect upon and stimulate the creative process, the concrete definition of this functionality and what it entails is highly inconsistent and subjective. This effect appears to be even more severe than in defining "similarity," as there is a priori no common ground among individuals which domains, parameters, and dimensions are affected by this notion.

Apart from the individual differences, in the bigger picture, we can conclude that for a concept like strangeness, the goal is not to optimize for immediate liking or usefulness but primarily in giving results that help in reflecting upon the whole process. Unexpectedness, as defined in [1], plays a role in this process, however the ideas of controlled strangeness and desired opposition also encompass the possibility of including "irrelevant" results, i.e., results whose "utility" value might only become apparent after some undefined time.

"Strange results" — as well as complete opposition, as the extreme form of strangeness — could therefore also help in appreciating other results in a different way, after being exposed to the strange version. Aiming at finding a more adequate interpretation of utility in the context of creative work could therefore bridge the concepts of strangeness and unexpectedness.

It is clear that at this point, results are very preliminary. From the insights gained, we expect that future developments will benefit from framing the questions of otherness and strangeness within a user-centric evaluation framework explicitly designed for explaining user experience, such as [9, 10]. Despite all its shortcomings, the presented prototype is part of an iterative process to engage with practitioners and gain more concrete feedback to narrow the gap between vision and operational models. The presented step provided us with detailed feedback and has encouraged us to proceed further into the exploration of systems for musical obstruction, broadening our focus to include work done in more experimental music creation.

6. ACKNOWLEDGMENTS

This work is supported by the European Union's seventh Framework Programme FP7 / 2007–2013 for research, technological development and demonstration under grant agreement no. 610591 (GiantSteps).

7. REFERENCES

[1] P. Adamopoulos and A. Tuzhilin. On unexpectedness in recommender systems: Or how to better expect the unexpected. *Transactions on Intelligent Systems and Technology*, 5(4):54:1–54:32, Dec. 2014. ACM.

[2] X. Amatriain, P. Castells, A. de Vries, C. Posse, and H. Steck, editors. *Proceedings of the Workshop on Recommendation Utility Evaluation: Beyond RMSE*, Dublin, Ireland, 2012. CEUR-WS.

[3] K. Andersen and F. Grote. GiantSteps: Semi-structured conversations with musicians. In *CHI '15 Extended Abstracts on Human Factors in Computing Systems*, 2015. ACM.

[4] K. Andersen and P. Knees. The Dial: Exploring computational strangeness. In *CHI '16 Extended Abstracts on Human Factors in Computing Systems*, 2016. ACM.

[5] P. Castells, N. Hurley, and S. Vargas. Novelty and diversity in recommender systems. In F. Ricci, L. Rokach, and B. Shapira (eds.), *Recommender Systems Handbook*, 2nd edition, 2015. Springer.

[6] N. Collins. Contrary motion : An oppositional interactive music system. In K. Beilharz, B. Bongers, A. Johnston, and S. Ferguson (eds.), *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2010.

[7] P. Knees, K. Andersen, S. Jordà, M. Hlatky, A. Bucci, W. Gaebele, and R. Kaurson. The GiantSteps project: A second-year intermediate report. In *Proceedings of the 42nd International Computer Music Conference*, 2016.

[8] P. Knees, K. Andersen, and M. Tkalčič. "I'd like it to do the opposite": Music-Making Between Recommendation and Obstruction . In *Proceedings of the 2nd International Workshop on Decision Making and Recommender Systems*, 2015. CEUR-WS.

[9] B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, and C. Newell. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction*, 22(4-5):441–504, Oct. 2012.

[10] P. Pu, L. Chen, and R. Hu. A user-centric evaluation framework for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, 2011. ACM.

[11] M. Tkalčič, M. Kunaver, J. Tasič, and A. Košir. Personality based user similarity measure for a collaborative recommender system. In *Proceedings of the 5th Workshop on Emotion in Human-Computer Interaction-Real World Challenges*, 2009.

[12] R. Vogl and P. Knees. An intelligent musical rhythm variation interface. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, 2016. ACM.

[13] E. Wold, T. Blum, D. Keislar, and J. Wheaten. Content-based classification, search, and retrieval of audio. *MultiMedia*, 3(3):27–36, 1996. IEEE.

[14] W. Wu, L. Chen, and L. He. Using personality to adjust diversity in recommender systems. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, 2013. ACM.