Finding *My* Beat: Personalised Rhythmic Filtering for Mobile Music Interaction

Daniel Boland
School of Computing Science
University of Glasgow, United Kingdom
daniel@dcs.gla.ac.uk

Roderick Murray-Smith
School of Computing Science
University of Glasgow, United Kingdom
rod@dcs.gla.ac.uk

ABSTRACT
A novel interaction style is presented, allowing in-pocket music selection by tapping a song's rhythm on a device's touchscreen or body. We introduce the use of rhythmic queries for music retrieval, employing a trained generative model to improve query recognition. We identify rhythm as a fundamental feature of music which can be reproduced easily by listeners, making it an effective and simple interaction technique for retrieving music. We observe that users vary in which instruments they entrain with and our work is the first to model such variability. An experiment was performed, showing that after training the generative model, retrieval performance improved two-fold. All rhythmic queries returned a highly ranked result with the trained generative model, compared with 47% using existing methods. We conclude that generative models of subjective user queries can yield significant performance gains for music retrieval and enable novel interaction techniques such as rhythmic filtering.

Author Keywords
Rhythm; Music; Tapping; Machine Learning;

ACM Classification Keywords
H.5.2. Information Interfaces and Presentation: Haptic I/O

General Terms
Human Factors; Design; Performance

INTRODUCTION
Interaction with music was previously as simple as turning the dial of a radio or selecting the desired music CD. With music libraries now digital, mobile and of increasingly large scales, listeners are often confronted with hierarchical menus or must type in a query to retrieve their desired music. In mobile music-listening contexts users instead often simply give up control and randomly shuffle their music e.g. using an iPod shuffle [18]. We identify a need for casual mobile interaction with music, with users able to assert control over their music listening experience when they need to without having to divert their full attention to their device.

MobileHCI '13, August 27 – 30, 2013, Munich, Germany.

ACM 978-1-4503-2273-7/13/08$15.00.

Figure 1. Users are able to select music without taking their device out of their pocket by simply tapping a rhythm or tempo on the device – allowing for casual eyes-free music interaction.

We present a novel interaction style using rhythmic input for sorting music, supported by a Bayesian approach to modeling the user's music-listening intent. An initial study is described with findings about the subjective way in which users annotate musical rhythm. We describe methods for interpreting rhythmic queries and identifying relevant musical works, and also detail a generative model for rhythmic queries which can be trained to specific users. Experimental results are given demonstrating the benefit of the use of a trained generative model over existing onset detection approaches. Feedback from participants was generally positive and included a suggested use case for in-pocket interaction.

Our approach involves a Bayesian combination of evidence about the user's intended song from the rhythmic pattern and tempo of their query. The system orders the music collection according to the inferred belief about the user's intent, presenting the intended song and other songs with similar tempo or rhythmic properties. The approach outlined sets a foundation for incorporating additional sources of evidence.

As a demonstrator for the techniques discussed in this paper, we present a mobile phone interface for searching music by tapping the music's rhythm or general tempo onto the device, detected by accelerometer, as depicted in figure 1. This system allows for a casual style of music interaction, allowing the user to assert control over their mobile music player without needing to remove it from their pocket.

MOTIVATION

Tapping the rhythm of a song onto a mobile device does not require the full attention of the user, indeed it does not even require the user to look at the screen or remove the device from their pocket. The case for such casual music interaction is outlined, along with a consideration of musical rhythm as a universal music feature.

Casual Music Interaction

In mobile music listening contexts, users are often unable to micro-manage their music listening experience. As in [17], we consider the interaction from a control-theory perspective, with casual interactions not requiring the user to engage in a tight control loop. When listening to music whilst walking for example, users would have to stop to enter a tightly-coupled interaction with a touchscreen device to select the next track. Instead, mobile music listening devices offer a random shuffle feature, allowing users to give up control and enjoy a serendipitous listening experience [12]. By allowing users to casually provide evidence about their music listening intent, they can be empowered to influence their music listening experience without suffering undue distraction.

Rhythm as Input Modality

Rhythm is a fundamental feature of music – more important for comprehending musical sequences than the absolute positioning of events in the time domain [24] and one which listeners can easily reproduce [4]. Perhaps surprisingly, rhythm is a greater factor for people when they assess the similarity of musical patterns than pitch [15]. Exploiting this predisposition to tapping rhythm as a form of music retrieval would allow for music selection on a device with very limited sensing ability – a microphone, button or single capacitive sensor would suffice.

Recent work has enabled capacitive sensors to detect touch input through fabric, supporting gestural input including drawn letters [20]. Whilst such an input modality could support explicitly 'typing' a music query, it would require users to engage in a more tightly-coupled interaction loop than with rhythmic querying i.e. having to think of an exact track and spelling it rather than casually tapping a beat. Tapping input can now also be detected via headphones [13] making it an ideal input for mobile music-listening contexts. Minimising the technological footprint of an interaction in this way not only lowers cost but also frees designers from the encumberance of integrating displays, keyboards etc.

Cultural Aspects

Our goal is not only to allow users to sort their music by tapping a rhythm. We seek to show that modelling the variance in how users represent music can improve a system's ability to understand users' queries. How users query for music is subjective and culturally dependent [11] with no one interaction style suitable for all. We envision a style of music interaction where users can combine a variety of querying styles to refine their search through a music collection. Of particular interest is that while common music filtering techniques such as genre classification are culturally specific, the use and cognition of rhythm is universal across cultures [4]. This positions our work as a cross-cultural style of music interaction – a theme we explore in our evaluation.

BACKGROUND & RELATED WORK

We consider the existing efforts to implement a system of retrieving music by tapping a song's rhythm and also recent developments in the detection of the music event onsets which underpin any such system.

Query by Tapping

The retrieval of a musical work by tapping its rhythm is a problem which has received some consideration in the Music Information Retrieval community and is termed 'Query by Tapping' (QBT). The term was introduced in [7] which demonstrated that rhythm alone can be used to retrieve musical works, with their system yielding a top 10 ranking for the desired result 51% of the time. Their work is limited however in considering only monophonic rhythms i.e. the rhythm from only one instrument, as opposed to being polyphonic and comprising of multiple instruments. Their music corpus consists of MIDI representations of tunes such as "You are my sunshine" which is hardly analogous to real world retrieval of popular music.

Rhythmic interaction has been recognised in HCI [10, 25] with [5] introducing rhythmic queries as a replacement for hot-keys. In [2] tempo is used as a rhythmic input for exploring a music collection – indicating that users enjoyed such a method of interaction. The consideration of human factors is also an emerging trend in Music Information Retrieval [22]. Our work draws upon both these themes, being the first QBT system to adapt to users. A number of key techniques for QBT are introduced in [6] which describes rhythm as a sequence of time intervals between notes – termed inter-onset intervals (IOIs). They identify the need for such intervals to be defined relative to each other to avoid the user having to exactly recreate the music's tempo. A major limitation of prior QBT systems is that they do not support music data which is polyphonic (comprises of multiple voices/instruments). Such systems use one rhythmic sequence as being the de facto rhythm for a musical work, requiring that all users tap a musical rhythm in the same way.

In previous implementations of QBT, each IOI is defined relative to the preceding one [6]. This sequential dependency compounds user errors in reproducing a rhythm, as an erroneous IOI value will also distort the following one. The approach to rhythmic interaction in [5] however used k-means clustering to classify taps and IOIs into three classes based on duration. The clustering based approach avoids the sequential error however loses a great deal of detail in the rhythmic query and so we explore a hybrid approach in this note.

Onset Detection

In order to compare user queries against a music library, we must compute the intervals (IOIs) between the rhythmic events within the music. Onset detection is the task of finding such events and has been studied at length within the field of Music Information Retrieval.

Figure 2. Music playlist initially sorted alphabetically (left) and after a query for an upbeat hard rock song "Any Way You Want it" (right)

An evaluation of onset detection algorithms in [1] showed the most precise onset detection method reviewed was their variant of the 'spectral flux' technique introduced by [14] which measures how quickly the power spectrum of a signal is changing. They also discuss the benefits of adaptive whitening introduced in [23] which adaptively normalises frequency bands' magnitudes to improve onset detection in music with highly varying dynamics, such as the rock music used in this work. We use these onset detection techniques to update the existing work on query by tapping to a state of the art implementation. This is then used as a baseline to which we compare our use of user query models with polyphonic data.

INTERACTION TECHNIQUE

It is common when interacting with music systems to be presented with a list of music, perhaps as a playlist which is played sequentially. We propose that rhythmic queries can be used to infer a belief over such a music space about which songs a user wishes to listen to. In this work we develop a complete interaction which allows a user to 'shake up' a list of music by tapping a rhythm on their device, with the music then being sorted by rhythmic and tactus similarity. The user can then play through a playlist arranged by these features or proceed to select their intended song. Such an interaction highlights the flexibility of rhythmic queries, allowing users to find songs of a given tempo or with certain rhythmic properties or to simply select a specific song. We implement a demonstrator system and evaluate it with Singaporean users, demonstrating its viability and cross-cultural application, as well as addressing some issues inherent to Query by Tapping.

Exposing System Belief & Uncertainty

Displaying a ranked list of songs would lose some of the information about the user's interest which the system has inferred. For example several songs may have a very similar level of belief held about them and this would not be communicated by simply displaying a sorted list. By exposing the uncertainty in the interaction, we expect users will be better able to understand the state of the system and produce the most discriminative queries.

Figure 3. Participants entered rhythmic queries via the touchscreen of a Nokia N9 mobile phone.

To better expose the beliefs held by the system, we scale the size of each list entry by this belief. If one song alone is a particularly strong candidate then it will be much larger than the other entries. Similarly, where there is uncertainty across a number of songs, these will be a similar size – making the user aware of the uncertainty within the interaction. This approach incorporates the uncertainty in the output as well as the input of the system and can be applied generally, for example distorting a music map based on the beliefs or scaling words in a word cloud of the music.

INITIAL STUDY

We conducted an initial study to explore the feasibility of music filtering using rhythmic queries. 10 European participants were invited to produce rhythmic queries of songs which they selected from a corpus of 1000 songs. The corpus was collected from the participants' own MP3 music collections and was the same for all participants, with IOIs obtained using the state-of-the-art onset detection techniques discussed previously. The rhythmic queries were entered by the participant tapping on the touchscreen of a Nokia N9 which had been configured to log the time intervals between taps. A query comparison was also performed using the techniques described later in this paper. For this initial study, the phone screen was blank and users were instructed to select a song and then "tap the rhythm of the song on the touchscreen, in order to select that piece of music."

Observations

As an initial sanity check, queries produced by multiple users for the same song were compared against each other. Surprisingly, little similarity was identified for a large number of the songs. In discussions with users, it became apparent that a variety of strategies were employed when annotating the rhythm of a piece of music. In particular, users identified particular instruments which they would entrain with – annotating those instruments' onset events when available. Also, not all users were as verbose when producing the queries, with some users using fewer taps than others to represent the same rhythm in a piece of music. A depiction of how two users sample from the available instrument onsets is given in figure 4.

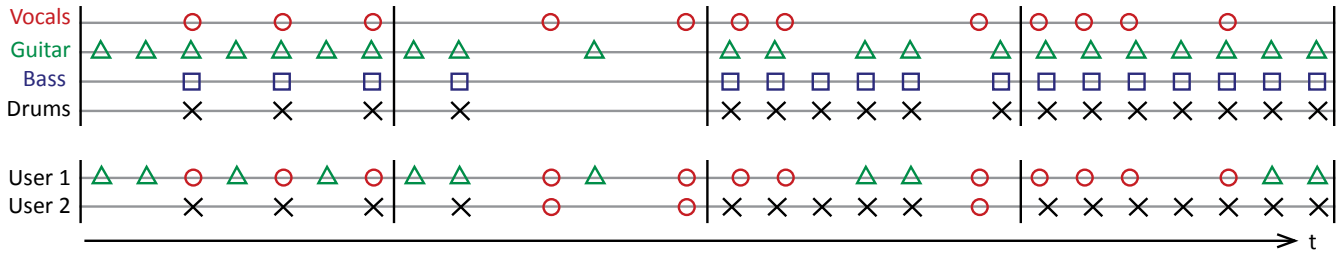

Figure 4. Users construct queries by sampling from preferred instruments. User 1 prefers Vocals and Guitar whereas User 2 prefers Drums and Bass.

The queries were compared against the entire corpus and a belief about the user's intent for each song was inferred. It was expected that rhythmic queries would at the very least improve the belief about the target song. An additional metric is to rank the songs in terms of this belief, with the intended song ideally ranking first. In this case only 68% of the queries collected led to an increase in belief about the target song, indicating that the corpus entries for the remaining 32% did not match the users' queries. This reinforces the observation that users do not annotate rhythm in the same way as each other, or indeed as the onset detection algorithm.

Discussion

The observations made in this initial study indicate that music retrieval using rhythmic queries is much more complex than originally estimated. In particular, there is a need for learning user-specific habits in producing rhythmic queries. The conversations with users provided some insight into how their tapping strategy may be modelled, with their affinities for the various instruments and their verbosity identified as important features. Instrument affinity can be captured as a list of the available instruments, ordered in terms of priority. Users switched instruments when their preferred instrument became available. User verbosity is more difficult to model, in the next section we turn to music literature to identify 'referent level' as a means of capturing the user's degree of verbosity.

The work in this paper explores the use of these features to construct a generative model of a user's queries, addressing the variance between user query production and allowing input queries to be predicted and matched. An alternative approach however would be to provide instruction or feedback to the user about how to tap rhythmic queries in a way which the system understands.

INTERPRETING RHYTHMIC QUERIES

The task of interpreting a rhythmic query has been broken up here into two steps – identifying a reference beat (tactus) for the query and then defining the rhythm relative to this beat. We must take this approach of defining rhythmic events relative to each other as users cannot accurately reproduce the absolute timing of music. We introduce a third step which allows the use of the tactus as additional noisier evidence.

Tactus

The way in which people process rhythm has been proposed by [4] to be universal across cultures, in that we hear an IOI as being relative to a previous IOI according to simple ratios. Complex rhythms are thus distorted into distinct categories of IOIs, each defined relative to each other. Crucially, the absolute timing values and thus the tempo are not of interest, only the pattern of relative IOIs encode the rhythm. In order to represent the relative IOIs that make up this pattern, we need to first identify the lowest common denominator of the intervals. Such a common unit is termed the 'tactus' and is estimated here by taking the autocorrelation of the histogram of IOI categories, giving a unit in which they can be defined.

For controlled non-musical rhythms, [5] used k-means clustering to identify long and short interval clusters. As we expect users to generate IOIs by sampling from distributions around an unknown number of IOI categories, we identify the categories by fitting a Gaussian Mixture Model selected using the Bayes Information Criterion. An example of such clustering can be seen in figure 5. Whilst the autocorrelation could be performed on the histogram of raw IOI values, we have found using the clustered values to be more robust.

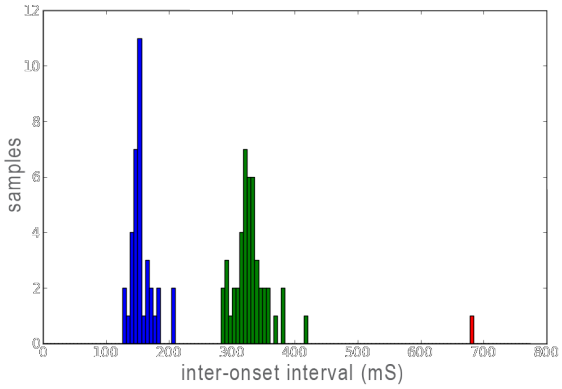

Figure 5. Histogram of IOIs in a rhythmic query, showing the clustering around categories of IOI values. Note that the mean of each category is double the previous, e.g. 170, 340, 680mS giving a tactus of 170mS.

Rhythmic String Matching

As rhythm is comprehended categorically, each interval is classified to an IOI category i.e. multiple of the tactus. These categories are assigned labels 'A', 'B' etc. We thus encode the rhythmic query as a string of category labels. For example, an interval double the length of the tactus would be classified as 'B'. An example query is depicted in figure 6, showing the mapping from interval to string character.

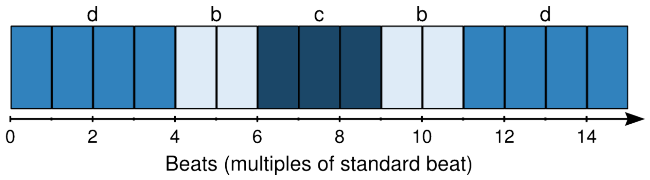

Figure 6. A rhythmic query depicted with alternately coloured intervals. The mapping between interval duration and string characters is shown.

The problem of matching rhythm can now be generalised to string matching, for which many efficient algorithms exist. As in [6], the Smith-Waterman local alignment algorithm is used as this is able to match a query against any part of a song. Similarly, the algorithm was adapted to scale the penalty with the mismatch error. An advantage of our approach over that in [6] is that no thresholds are required with the mismatch error being proportional to the difference between normalised IOIs:

$$E_{\text{IOI}} \propto (\text{IOI}_{\text{query}} - \text{IOI}_{\text{song}})$$

A further feature of the Smith-Waterman algorithm is that it was developed to allow for gaps in sequences, in order to match spliced DNA sequences [21]. This feature is also useful in QBT, as sections of a song in which the generative model fails to correspond to the user's query will simply be considered as a gap and given a limited penalty. The cost function of the string matching algorithm can assign a different score S for matched, missing or incorrect IOIs. A parameter G weights the penalty for a gap, such that a gap is equivalent to G IOI mismatches. In this work we take $G = 2$, assuming that if a query has two consecutive mismatched IOIs then the query is no longer conforming to the model at that point. The penalty scores are calculated as follows:

$$\begin{aligned} S_{\text{match}} &= 1 \\ S_{\text{mismatch}} &= -abs(IOI_A - IOI_B) \\ S_{\text{gap}} &= -G \times S_{\text{match}} \end{aligned}$$

The algorithm constructs an $n_{\text{query}} \times m_{\text{song}}$ matrix H (as in figure 7) where n is query length and m is the target sequence length. If the strings were identical then the diagonal of the matrix would identify each matching character pair, thus diagonal movements incur no penalty. In the example shown, one sequence has an 'A' removed (the downward step) to give a better match and thus a penalty is deducted from the score.

Penalties are assigned when the other movements are required in order to create a match, with a back-tracking process used at the end to find the (sub)path with the least penalty. This process allows for the best matching subsequences to be identified – in this work, a query matched against a larger song.

Tactus as a Feature

Previous work on QBT defines the rhythm irrespective of tempo (or tactus), as is done here. It has been shown however that tempo can be a useful feature in browsing a music collection [2]. We propose that tactus (being related to tempo) should be used as an additional feature to weight the ranking of rhythmic queries. The weighting given to this feature

$$H = \begin{pmatrix} & C & C & B & D & C & B & C \\ B & 0 & 0 & 10 & 0 & 0 & 10 & 0 \\ C & 10 & 10 & 0 & 0 & 10 & 0 & 20 \\ C & 10 & 20 & 0 & 0 & 10 & 0 & 10 \\ B & 0 & 0 & 30 & 10 & 0 & 20 & 0 \\ D & 0 & 0 & 10 & 40 & 20 & 0 & 10 \\ A & 0 & 0 & 0 & 20 & 20 & 10 & 0 \\ C & 10 & 10 & 0 & 0 & 30 & 10 & 20 \\ B & 0 & 0 & 20 & 0 & 10 & 40 & 20 \\ C & 10 & 10 & 0 & 10 & 10 & 20 & 50 \end{pmatrix}$$

Figure 7. The Smith-Waterman algorithm compares a query against a target sequence, matching 'CCBD-CBC'.

could additionally be adapted to each user though that is not explored in this work. We defined the tactus error function logarithmically such that halving a duration was equivalent to doubling it:

$$E_{\text{SB}} = \left(\log_2 \left(\frac{\text{SB}_{\text{Query}}}{\text{SB}_{\text{Song}}} \right) \right)^2$$

The tactus error is used as a prior over the music space when performing the rhythmic string comparison, biasing the results to those with similar tactus values. This helps discern amongst songs which are temporally very different but which share a similar rhythmic pattern. Where users only wish to listen to a particular style of music or cannot recall the rhythm of a song, they can simply tap a query at a desired tempo. If the rhythmic events are equally spaced (as in a metronome) then only the tactus is used to discriminate amongst the songs.

It is worthwhile to note that tactus is not necessarily the inverse of tempo. Tempo is often calculated as 'beats per minute', with an average value acquired across all the rhythmic events. It follows from this that the measured tempo would be highly dependent upon the section of song used to produce a query. Tactus however is the base unit which all the rhythmic events are multiples of and should be more stable throughout a piece of music. This distinction is only of interest from a technical perspective and generally, tactus is inversely proportional to tempo. In this paper and in discussions with users, we use the term tempo for the sake of convenience.

GENERATIVE MODEL

Rhythmic queries for a given song can vary greatly between subjects though are typically consistent within subjects. In order to build a database against which rhythmic queries can be matched, a generative model is required which can account for this variability. The use of the generative model encodes the knowledge about user behaviour obtained from the initial study. In essence, the model is designed to answer the question "*What would the user do?*" to achieve an outcome (selecting a target song). Training the model to users can be done by setting a fixed outcome and asking users to provide the input they would provide to achieve that outcome. The inference of the user's intended music is conditioned entirely upon the model and so should inherently improve as the generative model is improved or trained.

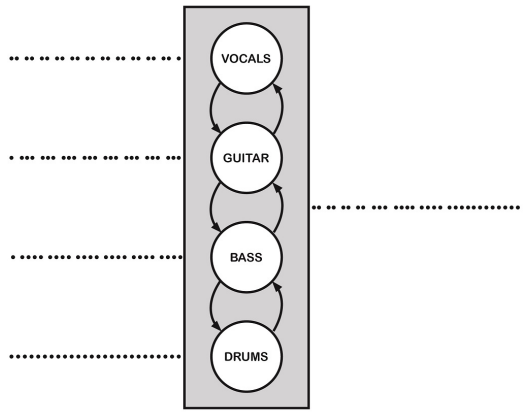

Figure 8. The generative model samples onset events from multiple instrument streams, producing a single output sequence.

Instrument Affinity

Music which is polyphonic will have a sequence of notes for each instrument and thus a sequence of IOIs for each instrument. As observed in the initial study, users typically switch between instruments as they feature in the music. This switching behaviour follows the user's preference of instruments to tap to. We have termed this set of preferences for instruments the affinity vector A_{ff} which ranks the available instruments in terms of the user's affinity. The generative model uses A_{ff} to switch to a preferred instrument's IOI sequence as those instruments feature. The behaviour of this model can be considered as a finite state transducer with a state for each instrument, sampling from the instrument sequence corresponding to the current state, as in figure 8.

Users are able anticipate notes and will not switch to another instrument state if their preferred instrument sequence will shortly resume, this look-ahead behaviour is also implemented in the generative model. The model samples notes up to $500mS$ in advance and stores them in a look-ahead buffer, only when the buffer is empty does the state change to the next available in the affinity vector. Whenever a note event occurs for an instrument with a greater affinity, the model immediately changes state. A simplifying assumption is made that the model is entirely deterministic, a probabilistic approach to switching may better model user behaviour but would add a great deal of complexity.

Referent Period

As music is highly structured, rhythm can be thought of as a hierarchy, where a note on one level could be split into multiple notes on a lower level. Individuals have a referent period i.e. a tempo at which information processing is natural to them and are likely to synchronize at a level in the hierarchy closest to their referent period. It has been shown that musical training and acculturation result in higher referent levels [3]. Modeling such variables allows for optimized rhythm matching. In order to model the differing referent periods of users, a music corpus must contain onset data for several levels of rhythmic complexity. The appropriate level is selected when training the generative model, though varies between songs.

INFERRING USER INTENT

The task of ranking songs based on some rhythmic evidence can be seen as an inference task and not only as a traditional retrieval task. Previous work in information retrieval has introduced the use of query models to encode knowledge about how a user produces a query [9]. Our work is similar in the use of a generative model as a query likelihood model. When producing a rhythmic query, the user uses their internal query model $\vec{M}_u$. They then produce a query using this model, which is matched against the music corpus. The problem can thus be expressed using Bayes' theorem:

$$p\left(d_j \,|\, q, \vec{M}_u\right) = \frac{p\left(q \,|\, d_j, \vec{M}_u\right) p\left(d_j \,|\, \vec{M}_u\right)}{p\left(q \,|\, \vec{M}_u\right)}$$

That is, we can infer a belief about the intended song conditioned upon the query q by computing the likelihood of the query being produced for each song d_j in the music space. The prior $p\left(d_j \,|\, \vec{M}_u\right)$ should be non-informative, currently there is no evidence that music listening intent is directly conditioned upon the user's query model for tapping to music. In order to perform the above inference we must train the generative model of user queries $\vec{M}_u$. As described earlier, for training we take a fixed outcome (i.e. selecting a target song d_t) and ask users to provide a suitable query q so as to achieve that outcome. We then infer a belief about which generative model parameters were used to construct the query:

$$p\left(\vec{M}_u \,|\, q, d_t\right) = \frac{p\left(q \,|\, d_t, \vec{M}_u\right) p\left(\vec{M}_u \,|\, d_t\right)}{p\left(q \,|\, d_t\right)}$$

In this work the prior $p\left(\vec{M}_u \,|\, d_t\right)$ is uninformative however it is probable that the user's approach to tapping music is conditioned upon the particular song to some extent. In wider use where a large corpus of queries has been collected, we could compute a prior belief about the tapping model used for a given song. This should improve the inference of the user's general tapping model. For the work here we train the model for a given song and a given participant to account for this however also look at training across songs for a subset of participants.

Query Likelihood Function

In order to infer a belief about whether a user is interested in a given song, we must compute the likelihood $p\left(q \,|\, d_j, \vec{M}_u\right)$ of their query conditioned upon their wanting that song and the user's query model. We use the string matching function to compare user queries with those in the database and assign beliefs to songs accordingly. The more edits that are required to match the query to the stored song sequence, the lower the estimated likelihood of that query for that song. As we are only interested in ranking the songs, we do not need to compute the marginals.

EVALUATION

To evaluate the performance difference due to the various query likelihood models discussed in this paper, a within-subjects experiment was performed. The use of the query likelihood models was controlled as a factor with three levels: *Baseline* (onset detection), *Untrained GM* (polyphonic data with generative model) and *Trained GM* (polyphonic data with trained generative model). Given the parameters of the generative model, the target space against which queries are matched is greater than in the baseline case. For the baseline, there is one possible sequence for each of the 300 songs. The model has four instrument sequences for each song, sampled using the generative model with 96 possible parameter permutations, yielding a target song space of $28,800$ sequences.

Experimental Setup

A corpus of MIDI and MP3 music data was acquired from popular rhythm games, featuring note onset times (from which we compute IOIs) for each instrument in 300 rock and pop songs. Whilst the size of this corpus was limited by our source of data, it does reflect real-world usage – [8] gives it as the median music file collection size in Germany. Participants selected at least two songs from the corpus and listened to them to ensure familiarity. They were then asked to produce at least three rhythmic queries for each song by tapping a section of the song's rhythm on the touchscreen of a Nokia N9 mobile phone. No feedback was provided to the participant after each query. The queries were used to train the generative model using leave-one-out cross-validation. Participants were provided with a set of headphones to control background noise as a factor,

Quantitative data was captured in the form of rank results, with songs ranked according to the inferred belief. Qualitative data was captured during a discussion with participants where they were asked to comment on the style of interaction presented and whether they found it enjoyable and/or useful. Eight unpaid British participants volunteered, four female, four male, ages 18 – 72 (mean: 30). Half of the participants were university students and one a retiree. Participants were instructed to "tap the rhythm of the song on the touchscreen, in order to select that piece of music." No limit was made on the length of the queries. The participants were not musicians, otherwise musical background was not controlled.

Rhythmic queries were captured using a Nokia N9, running software developed in QML and C++ using the Qt framework. Our variant of the Smith-Waterman algorithm was implemented in C. We chose to infer a belief over the model parameters rather than use the Maxmimum Likelihood Estimate as our goal was for the target song to always be in the on-screen (top 20) rankings, rather than optimising for the highest possible rankings at the cost of some queries failing.

Results

The two measures of interest are how rapidly a user can filter their music collection and the probability distribution of achieving a rank position in the results. Query performance typically improves with query length as seen in figure 9.

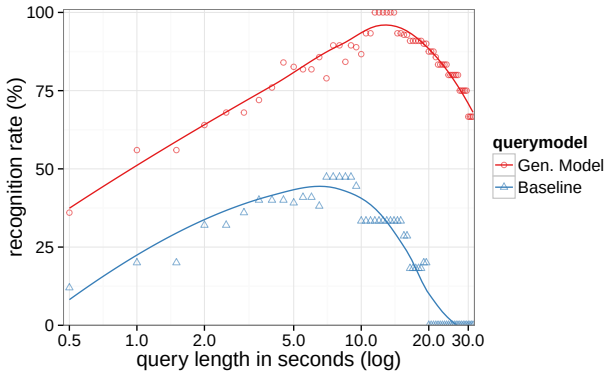

Figure 9. Percentage of queries yielding a highly ranked result (in the top 20 i.e. 6.7%) plotted against query length in seconds.

Higher rankings are achieved for all query lengths when using the trained generative model. Queries with lengths of approaching 10 seconds always yielded an on-screen result (in the 8.5% of the corpus). Up to this point the recognition rate improves with query length, as the additional information is incorporated. A key feature of the results is that queries over 10 seconds lead to a rapid fall-off in performance.

A general linear model repeated measures ANOVA showed that mean rank score differed statistically with the training of the generative model ($F(2, 48) = 9, 31$, $P < 0.001$). A pairwise comparison was performed, showing a statistically significant mean improvement of the trained generative model over the other two query likelihood models ($P < .001$). Notably, the improvement of the untrained generative model over the the baseline monophonic model was not statistically significant ($P = .279$). A comparison of performance using the three query models can be seen in figure 10, showing distributions of result rankings of the queries' target songs. Four users provided queries for additional songs. In these cases the model could be trained on the queries for the other songs however performance fell, yielding a top 20 result only 70% of the time.

Participants said they enjoyed using QBT as an interaction style, often choosing to continue the interaction beyond the requirements of the experiment. The experiment was viewed as a game, with half of the participants requesting further at-

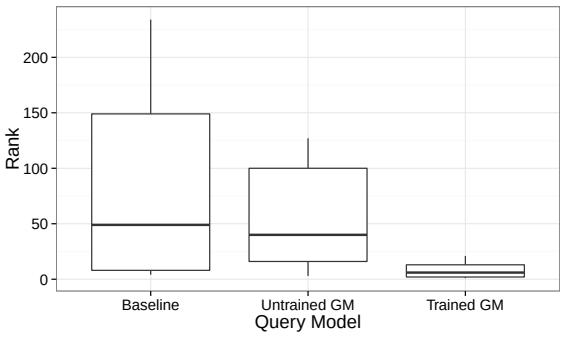

Figure 10. Box plots of retrieval ranks using the query models.

tempts to improve their results. One participant identified in-pocket music selection as an interesting use case. Another expressed concern about the scalability of using rhythmic queries for especially large music libraries. All the participants immediately grasped the concept of query by tapping and were able to readily produce rhythmic queries. Notably, one participant claimed to only tap to one instrument.

Focus Group

An exploration of the application concept was conducted using a prototype demonstrator with a small focus group of five Singaporean participants – four female, one male. The particpants all used mobile media players and were aged 26 – 59 (mean: 40). They also spoke and listened to music in Chinese and English, giving a view of how the system performs cross-culturally. The demonstrator was presented to them and they were able to interact freely with it, an informal conversation followed to capture their impressions. Each participant was asked to compare the interaction with their usual way of listening to music in a mobile context, to consider the 'in-pocket' use case, whether they would feel comfortable using the interaction style in public and whether the interaction style was suitable to the music they listened to.

The discussions with participants identified that they all often listened to music using the 'shuffle' feature of their phones or music players, occasionally using the menu interface to select particular music. The use of rhythm to shuffle music was well received as a superior option to random shuffle, with P3 noting that random shuffle can lead to inappropriate music selections at night and that shuffling by tempo could avoid this. Participants were very positive about the in-pocket use case, with P1 saying that it "means I can select music when it's raining" and P2 saying that "it can be hard to take my phone out to change song because I have small pockets." P3 and P4 noted that fear of theft sometimes prevented them from removing their phone to select music. Social acceptability of the interaction is important as the use-case for in-pocket music selection includes being in mobile public contexts. None of the users indicated an issue with tapping rhythms on the phone on the bus (P5 likened it to drumming on your lap).

Participants also offered up some concerns about the interaction technique. P2 doubted their ability to produce rhythmic queries for all their music despite doing well with the demo, citing their lack of musical knowledge. They appreciated the inclusion of the tempo-only sorting ability to mitigate this. P5 pointed out that the 'intensity' of their taps is an overlooked feature of rhythm and expected it would be interpreted.

DISCUSSION

The results show that accounting for subjectivity in users' rhythmic queries greatly improves retrieval performance despite increasing the target space. Whilst validity was shown for average music collections, scalability is a concern, with larger music libraries requiring additional sources of evidence for effective retrieval. A further concern is the drop-off in performance for large rhythmic queries, possibly due to the user having a limited section of music in mind when they begin producing a rhythmic query.

Improving the System

Our results suggest that performance for ours and existing techniques may be greater if query length is limited. Further improvements for the generative model should also allow for participants who tap one instrument exclusively. Training across songs for a subset of users showed a fall in performance for some songs, indicating that tapping style varies with song as well as with user. As we discuss previously, the prior belief of tapping style for particular songs could be learned after acquiring a large amount of queries across users.

Users

As expected, participants stated they used shuffle frequently – the ubiquity of shuffle is noted in [18] and attributed to the popularity of the iPod shuffle MP3 player. Similarly, users felt comfortable with the interaction technique – previous work has shown that users feel comfortable producing rhythmic and tapping gestures in public [19]. Of great interest is that all of the participants felt the interaction suited both their English and Chinese music, we noted before that existing techniques such as browsing by genre are culturally specific and so this is a key advantage of this interaction style.

Limitations

While we have demonstrated the benefits of the techniques presented, much work remains in studying rhythmic querying as an interaction technique. The evaluations here aim to validate the use of the generative model. A wider study could provide further insight into rhythmic querying behaviour – already we have identified additional features to incorporate such as tap intensity, single-instrument annotation and song-specific tapping strategies. A key limitation of our work is that the training and validation queries are acquired in the same session - a longitudinal study may show that users' rhythmic querying behaviour changes over time. Overall, we show that QBT can be greatly improved with the use of a trained generative model and is an interaction technique worthy of much further exploration.

Incorporating Subjectivity

It could be argued that one would of course expect better results from the use of ground-truth polyphonic music data than from the use of detected onset events. The untrained generative model provides an example of this, showing an apparent though non-significant improvement over the baseline (onset detection). After training the generative model, the improvement in retrieval performance is dramatic. It is not surprising that incorporating knowledge about the user improves the performance of the system. The use of a generative model does not in itself yield much of an improvement however it provides a mechanism by which one can incorporate the prior knowledge about the user. It is only when we address the issue of subjectivity in how users produce their queries that significant performance increases are seen. We addressed subjectivity through the use of a simple model based on initial discussions with users, it is likely that far more powerful models could be constructed. That this interaction style is only made usable by addressing the user's subjectivity in expressing their rhythmic query is a key outcome of this work.

Scalability

The techniques employed aimed to ensure a top 20 (onscreen) result (as shown in the quantitative results) to avoid the issue of failed queries. That users were unsure of their ability to achieve this level of performance indicates that further work could be done to improve users' confidence during the interaction, for example with real-time feedback. It is to be expected that as the size of the music collection is increased, retrieval performance using rhythmic queries will fall. We have shown this style of interaction to be valid for an average music collection of 300 songs. Performance is far greater as collection size is reduced, with queries yielding first ranked results 65% of the time when the collection is halved to 150 songs. Our Bayesian approach allows for this issue of scalability to be addressed through the introduction of additional sources of evidence. For example a different interaction style would use sung queries – providing pitch, rhythm and tactus as evidence. Such an interaction would also benefit from the user query modelling approach introduced here. Other evidence sources could include the dynamics of the tap events, for example a 'strumming' action could denote guitar events.

Rather than implement a simple retrieval of a musical work by tapping its rhythm, we were able to re-order an entire collection of music according to the rhythm and tempo evidence. This means that for this or larger collections, the drop-off in retrieval performance is acceptable as the user is still able to assert control over their music collection, ordering it according to the evidence they provide. The queried song could even just be a landmark track, which the user selects to indicate the type of music they want. At the very least, the user can shuffle their music by tempo and rhythmic similarity. The intended song need only be ranked in the top 20 results displayed onscreen, with the user then able to select the song easily.

Query Performance

A surprising result is the sharp drop-off in retrieval performance with query length. One might expect that as more evidence is introduced, retrieval performance would increase, indeed such a relationship is seen in queries up to $10s$ in length. It is unlikely that users recall the entirety of a song in advance of producing a rhythmic query – instead they would select a memorable or distinctive passage of the music. The drop-off in performance could reflect that users have continued beyond the salient part of the music they had in mind. This issue could be addressed by limiting the length of rhythmic queries or by providing real-time visual or haptic feedback to the user so that they entrain with the song as it begins playing. This result has wider implications for rhythmic interaction (for example in the use of rhythmic 'hot-keys' in [5]) in that it indicates an upper length for rhythmic patterns. More generally, this result could suggest that any music content based querying technique such as humming or singing may also suffer from falling performance for queries over ten seconds. It is worth noting that whilst mean rank result improves with the use of the generative model and with training, the most significant change is in the 'long tail' of poor results. In the initial study we saw that around two thirds of elicited queries had some match to the music data. The use of a generative model allows for the retrieval of the remaining third of queries which suffer from subjectivity. Our work not only improves mean query performance but also makes for a more consistent user experience, cutting off the long tail of poor results caused by subjectivity in query production.

Combining Evidence

An advantage of taking this Bayesian approach to sorting the music space is that we can combine evidence in the form of a prior over the music space. In this paper we use this to incorporate the tactus as an additional source of evidence, having previously separated the rhythmic pattern from the tactus. There are many other sources of evidence about listening intent which could be taken as a prior belief over the music space, for example the user's listening history for the music tracks. A further benefit is in being able to introduce a hyperprior i.e. a belief about the distribution of some hyperparameter of the prior. In our case this allows us to avoid over-fitting our belief about the user's query model based on a limited number of training cases. We infer a belief about which query model is used with a uniform hyperprior, to ensure that the inferred belief is not too concentrated upon one model. This reduces the ranking of good queries however has the effect of improving the recall of queries where the learned model is not the best match. A trade-off must be struck with this uncertainty acting as something of a twiddle-factor, though in real-world usage it would not be required due to the availability of more training data.

FURTHER CHALLENGES

This work has implications for future research in interaction with music, demonstrating a need for considering user variance. We aim to demonstrate the utility of generative models of user queries for inferring user intent in a range of music querying interaction styles. Having identified the benefits of improving consensus with the user with a trained model, there is an opportunity for further study in using feedback to train users towards a consensus. The rhythmic matching algorithm could be improved further through the use of music theory, for example where a whole note is replaced by four quarter notes, the penalty should be very little. Applying such techniques in similar efforts in matching pitch in melodies led to the Mongeau-Sankoff algorithm [16] and eventually to services such as Shazam and SoundHound. Such work combined with improvements in onset detection could allow robust commercial applications of rhythmic music retrieval. Given that users can recall a great number of musical works and the results shown here, future research could build upon this work to use musical rhythm for interaction tasks other than just the retrieval of music e.g. tapping a rhythm on a phone in-pocket to dial a corresponding contact when using a bluetooth headset.

CONCLUSIONS

The interaction technique presented in this work enables users to enjoy a casual style of music retrieval – empowering them to interact with their music in new contexts such as in-pocket music selection. By shuffling the music playlist by the rhythmic query, users can provide uncertain queries about a type of song or query for a particular song without having to recall its title etc.

We show that existing state-of-the-art techniques for rhythmic querying perform poorly for real music with multiple instruments. We improve upon previous efforts by using trained user query models to sample from polyphonic music data. These user models allowed for a dramatic improvement in retrieval performance, with the intended song always appearing in the top ranked results. This work highlighted the issues caused by subjective music queries and developed a personalisable music retrieval system. We have developed a novel technique that is not only effective in the sorting or retrieval of music but also introduces an enjoyable game-like element to music retrieval. Users enjoyed using the system in trials, often asking to continue use beyond the experimental requirements in order to attempt to improve their ranking. In particular we highlight that users were able to generate rhythmic queries from their subjective interpretation and memory of music rather than using a memorised rhythmic pattern. Removing the need for memorisation in this way has applications beyond music retrieval, for example the work on rhythmic hot-keys could benefit from the presented approach.

The techniques developed here achieve our goal of casual in-pocket music control with users able to see the benefit of the interaction style and identify use cases relevant to them. Personalised rhythmic querying is an effective and enjoyable interaction technique. Using a generative model of subjective queries has taken rhythmic music retrieval from a concept with potential to a usable interaction style. By understanding how users query for music and encoding this knowledge as a generative model, we present an interaction technique which ensures the right song is only ever a few taps away.

ACKNOWLEDGMENTS

We are grateful for support from Bang & Olufsen and the Danish Strategic Council of Research.

REFERENCES

1. Böck, S., Krebs, F., and Schedl, M. Evaluating the Online Capabilities of Onset Detection Methods. In *Proc. ISMIR 2012* (2012), 49–54.
2. Crossan, A., and Murray-Smith, R. Rhythmic Interaction for Song Filtering on a Mobile Device. *Haptics and Audio Interface Design* (2006), 45–55.
3. Drake, C., and Ben El Heni, J. Synchronizing with Music: Intercultural Differences. *Annals of the New York Academy of Sciences 999* (2003), 429–437.
4. Drake, C., and Bertrand, D. The Quest for Universals in Temporal Processing in Music. *Annals of the New York Academy of Science 930* (2001), 17–27.
5. Ghomi, E., Faure, G., Huot, S., and Chapuis, O. Using rhythmic patterns as an input method. *Proc. CHI* (2012), 1253–1262.
6. Hanna, P. Query by tapping system based on alignment algorithm. In *Proc. ICASSP* (2009), 1881–1884.
7. Jang, J., Lee, H., and Yeh, C.-H. Query by Tapping: A New Paradigm for Content-based Music Retrieval from Acoustic Input. *Proc. PCM* (2001).
8. Karaganis, J., and Renkema, L. *Copy Culture in the US and Germany*. American Assembly, 2013.
9. Lafferty, J., and Zhai, C. Document language models, query models, and risk minimization for information retrieval. In *Proc. SIGIR 2001*, ACM (2001), 111–119.
10. Lantz, V., and Murray-Smith, R. Rhythmic interaction with a mobile device. In *Proc. NordiCHI*, ACM (2004), 97–100.
11. Lee, J., Downie, J., and Cunningham, S. Challenges in cross-cultural/multilingual music information seeking. In *Proc. of MIR*, Citeseer (2005), 1–7.
12. Leong, T., Vetere, F., and Howard, S. The serendipity shuffle. In *Proc. OZCHI* (2005), 25–28.
13. Manabe, H., and Fukumoto, M. Headphone taps: a simple technique to add input function to regular headphones. In *Proc. MobileHCI 2012*, ACM (2012), 177–179.
14. Masri, P. *Computer modelling of sound for transformation and synthesis of musical signals.* PhD thesis, University of Bristol, 1996.
15. Monahan, C. B., and Carterette, E. C. Pitch and duration as determinants of musical space. *Music Perception 3*, 1-32 (1985).
16. Mongeau, M., and Sankoff, D. Comparison of musical sequences. *Computers and the Humanities* (1990).
17. Pohl, H., and Murray-Smith, R. Focused and casual interactions: Allowing users to vary their level of engagement. In *Proc. CHI* (2013).
18. Quiñones, M. Listening in Shuffle Mode. *Lied und populäre Kultur/Song and Popular Culture*, 2007 (2007), 11–22.
19. Rico, J., and Brewster, S. Usable gestures for mobile interfaces: evaluating social acceptability. In *Proc. CHI* (2010).
20. Saponas, T. S., Harrison, C., and Benko, H. Pockettouch: through-fabric capacitive touch input. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, ACM (2011), 303–308.
21. Smith, T. F., and Waterman, M. S. Identification of common molecular subsequences. *Molecular Biology 147*, 1 (1981), 195–197.
22. Stober, S. *Adaptive Methods for User-Centered Organization of Music Collections*. PhD thesis, Otto-von-Guericke-University, Magdeburg, 2011.
23. Stowell, D., Plumbley, M., and Mary, Q. Adaptive whitening for improved real-time audio onset detection. *Proc. ICMC 2007* (2007).
24. Trehub, S. E. Human processing predispositions and musical universals. In *The Origins of Music*, N. L. Wallin, B. Merker, and S. Brown, Eds. MIT Press, 2000, ch. 23, 427–448.
25. Wobbrock, J. O. Tapsongs: tapping rhythm-based passwords on a single binary sensor. In *Proc. UIST* (2009), 93–96.