LYRIC-BASED RHYTHM SUGGESTION

Eric Nichols

Indiana University
Center for Research on Concepts and Cognition

ABSTRACT

Which comes first—the lyrics or the music? Here we consider the lyrics-first approach to songwriting and seek to augment the process by developing a creativity-support tool which uses lyrics as a creative constraint for rhythm generation. We describe a novel algorithm for suggesting possible melodic rhythms to complement the accent structure of a given set of lyrics. The algorithm is composed of three main components: a scoring function used to judge the relative success of candidate rhythms, a database of English pronunciation to determine syllable stress levels, and traditional search techniques to find high-scoring rhythms in a large space of candidate rhythms. Preliminary results are encouraging: given existing song lyrics, the "correct" human-composed rhythm is generally found high in the list of suggestions.

1. INTRODUCTION

"Music is heightened speech"—this may be an old cliché (Bernstein 1976), but when writing music with lyrics it can serve as compositional advice. Books on songwriting and music composition often suggest writing melodies by considering the natural accents of spoken text (Peterik et. al. 2002, Jarret and Day 2008). We provide a formal, algorithmic approach to generating rhythms for a given lyric according to this suggestion. Essentially, we consider all possible rhythms for the text and assign high scores to rhythms where the natural syllable stresses match up well with the metric accents implied by the time signature. Other heuristics are used to refine the results.

Our task is made somewhat easier not only by some simplifying assumptions about the possible rhythms allowed (e.g. melisma is not allowed), but also because our goal is not to generate "the correct rhythm". Instead, we seek to inspire a human composer by providing a set of possible rhythms. In the future, this rhythm generator could be one of a collection of creativity-enhancing tools in composer's toolbox.

2. RELATED WORK

While automatic generation of rhythms to fit a given lyric appears to be novel, rhythm generation without lyric constraints is common—for instance, unless rhythmic information is provided by the programmer, any algorithmic composition system must generate rhythms. Scoring generated rhythms is crucial to our work and that of some other systems. For example, Temperley (2007) gives a Bayesian approach for computing probabilities of possible metric interpretations of rhythms specified in terms of onset times; this approach, however, does not use lyrics and solves the rather different problem of scoring different interpretations, rather than scoring different composed rhythms.

Temperley (2001) describes a preference-rule approach for determining metric interpretations of music; again, this is a different problem than ours, but it does include one preference rule applicable to music with lyrics. Metric Preference Rule 8, the Linguistic Stress Rule, states: "Prefer to align strong beats with stressed syllables of text." A version of this rule features prominently in our algorithm.

3. ALGORITHM

The conceptual algorithm to generate suggested rhythms is quite simple:

1. Input the lyrics, desired total duration, and time signature
2. Define the space of all possible rhythms matching the given number of syllables and total duration
3. Define a scoring function for rhythms given the lyrics
4. Search this space to find relatively high-scoring rhythms
5. Display a ranked list of the highest-scoring rhythms to the user

For example, given the lyric "Some Enchanted Evening", a duration of four quarter notes, and a 4/4 time signature, the algorithm would consider a space of all rhythms with six syllables, and find high-scoring rhythms as in Figure 1.

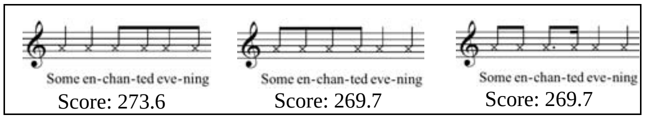
Some en-chan-ted eve-ning
Score: 273.6
Some en-chan-ted eve-ning
Score: 269.7
Some en-chan-ted eve-ning
Score: 269.7

Figure 1. Some suggested rhythms with scores for "Some Enchanted Evening"; 273.6 was the maximum score possible for this lyric. Each of these possibilities was ranked in the top 15 out of a possible 4282 rhythms.

3.1. Rhythm Space

As used here, a *rhythm* is a particular sequence of musical durations superimposed on a *metric grid*. A metric grid defines the relative strengths of major beats within each measure. Two possible metric grids are:

Strongest–weak–Strong–weak (4/4 time)
Strong–weak–weak (3/4 time)

The leftmost rhythm of Figure 1 is defined by the duration sequence {quarter, quarter, eighth, eighth, eighth, eighth} along with the 4/4 metric grid above. We assume use of the 4/4 time signature and this grid for the remainder of this paper. In order to simplify the search problem, we restrict the space of possible rhythms by only considering the following eight possible durations for each note: whole, dotted-half, half, dotted-quarter, quarter, dotted-eighth, eighth, sixteenth. Additional durations (such as triplets) can be considered at the expense of computation time. We further assume that each syllable of the given lyric will be associated with a single note; i.e., melisma is not allowed. Considering an indefinite number of notes in the generated rhythms would both require additional scoring heuristics and cause an explosion in the size of search space.

Rests are not allowed in the middle of the generated rhythm, because they can be approximated by extending the duration of the previous syllable by the length of the desired rest. However, to allow the rhythm to begin on an upbeat, we optionally generate a single rest (of one of the allowed durations, excluding a whole rest) at the beginning of the rhythm.

The rhythm space searched by the algorithm is the set of all possible duration sequences conforming to the restrictions described above, where the total duration of the sequence equals the user-specified total duration. The size of this space is bounded above by D^{S+1}, where D is the number of legal durations and S is the number of syllables. In the Figure 1, this upper bound is 8^7=2,097,152. However, the restriction to a four quarter-note duration reduces the search space to 4282 possible rhythms.

3.2. Scoring Function

We define a scoring function f to assign a score (s) to a given (rhythm, lyric) pair:

$$f(r, x) \to s \ \in \mathbb{R} \tag{1}$$

where r is the rhythm and x is the lyric. f is defined using several simple heuristics based on four features, f_1, f_2, f_3, and f_4. Two features are considered to be positive and contribute to a higher score:

- f_1: Accented syllables on strong beats
- f_2: Rare words beginning on strong beats

The other two features are negative and reduce the score:

- f_3: Long durations beginning on 8th- or 16th-note offbeats
- f_4: Accented syllables of short relative duration

f is simply a linear combination of these features:

$$f = f_1 + f_2 - f_3 - f_4 \tag{2}$$

3.2.1. f_1: Accented syllables on strong beats

This is a particular implementation of the Linguistic Stress Rule mentioned above (Temperley 2001). The stress on each syllable is assigned an integer value of 0, 1, or 2, corresponding to unstressed, secondary stress, and primary stress, respectively. Stress data (for American English words) is given by the Carnegie Mellon Pronouncing Dictionary (1998), with stress levels 1 and 2 swapped for consistency (in the original CMU database, "2" corresponds to secondary stress).

The definition of f_1 rewards stressed syllables on strong beats:

$$f_1 = stressLevel * beatScore \tag{3}$$

The *stressLevel* is 0, 1, or 2 as above, and *beatScore* is set to 40, 20, and 10, for syllables starting on a downbeat, strong beat (half-note level) or weak beat (quarter-note level), respectively. Syllables starting on an offbeat (such as the second eighth note of a measure) have a *beatScore* of 0.

3.2.2. f2: Rare words beginning on strong beats

We introduce a new heuristic based on our observation that common words such as "the" or "of" tend to fall on weaker syllables. We implement the converse: rare words should be rewarded when their first syllable appears on strong beats.

$$f_2 = wordRarity * beatScore \tag{4}$$

where *beatScore* is defined as above. *wordRarity* is determined using a frequency count of words in the British National Corpus (BNC) (Kilgraff 1998). The frequency counts range from 1 to 6,187,927 (for "the"). *wordRarity* is computed by the transform:

$$wordRarity = 2\left(1 - \frac{Log_{10}count}{7}\right) \tag{5}$$

so that it falls in the range (0,2], much like *stressLevel*. *wordRarity* is set to 2 for words not found in the BNC.

In a statistical survey of stress patterns in English text, Tempeley (2009) handles the problem of determining the stress level of *function words* such as "the" or "of" by reducing the stress level specified in the CMU database for a set of predetermined function words. It would be possible to use this approach here instead of *wordRarity*; however, using this additional continuous feature is useful here in generating additional differentiation between similar rhythms; i.e., less rhythms will be assigned the exact same score, reducing ties (and thus arbitrary sort order) in the sorted list of rhythms.

3.2.3. f_3: Long durations beginning on offbeats

Components f_1 and f_2 by themselves can assign high scores to some uncharacteristic rhythmic patterns such as {sixteenth, dotted eighth}. One approach might be to penalize uncommon rhythms based on a statistical analysis of existing music. Instead, we adopt a simple heuristic that penalizes long duration notes that begin on offbeats.

$$f_3 = penalty * (stressLevel + wordRarity) \quad (6)$$

$penalty$ is defined as follows:

- $penalty$=10 if onset is on a eighth-note offbeat and duration > eighth note
- $penalty$=40 if onset is on the first sixteenth-note offbeat in a beat and duration > eighth, or if onset is on the final sixteenth-note offbeat in a beat and duration > sixteenth
- $penalty$=0 otherwise

3.2.4. f_4: Accented syllables of short relative duration

If the primary accent of a word has a duration shorter than an adjacent syllable, a small penalty is applied:

$$f_4 = 1 \quad (7)$$

3.3. Search

When the number of syllables in the text is small (less than 7), brute-force evaluation of every possible rhythm is feasible. We implemented the algorithm for this case by enumerating all possible rhythms, scoring each rhythm using *f*, and sorting the results in order of decreasing score. We provide an interface, used in the results discussed below, for users to browse the sorted list of rhythms.

For larger numbers of syllables, we compute an approximate set of the best rhythms using the greedy method of *beam search* (Raphael and Nichols 2008). While not guaranteed to provide the optimal solutions, the local nature of the scoring function suggests that beam search will provide a good approximation. We perform beam search by growing a tree of rhythms, starting at the first syllable of the lyric and proceeding to the right. Tree branches are generated for each choice of duration for the following note—here, our tree has a branching factor of 8. Scores are computed at each node for the partial rhythms under construction. When the number of leaf nodes becomes larger than a predetermined size, low-scoring nodes are pruned. The best-scoring nodes constitute the "beam", which is continually grown and pruned until the tree has the desired height.

Note that the results for large numbers of syllables will likely not be as good as results for short phrases because the scoring function does not account for larger-scale form. We return to this topic in the future work section below.

4. RESULTS

We performed an initial evaluation of the algorithm using a collection of short phrases taken primarily from two musical theatre songbooks—a Sondheim collection, excerpts from *King and I*, and few other miscellaneous songs. Future evaluation will use a more comprehensive corpus; the aim here is to illustrate the merits and weak points of this algorithm. Input phrases were restricted to those in 4/4 time which could be represented in the rhythmic space defined above (i.e. no triplets), and consisting of no more than seven syllables.

4.1. Method

For each phrase, we compute the sorted list of suggested rhythms as described above, and record the rank of the original "target" rhythm (the human-composed rhythm notated in the score) as well as the number of possible rhythms. We consider the target rhythm to begin on a downbeat—pickup notes are represented using a full measure with a rest at the beginning. Using this data we compute the "percentile" score of the target rhythm. If the target rhythm has a high percentile, it suggests that the scoring function is behaving in a manner consistent with human rhythm composition, and hence we can expect some reasonable suggestions from the algorithm.

Table 1 gives the results for 30 phrases. The average percentile rank for target rhythms was 92.4, indicating that only 7.6% of possible rhythms scored higher than the target. This indicates that the scoring function has some success in ranking candidate rhythms. The raw rank scores are also important; the target rhythm for "Once Upon a Time" is ranked as #3, indicating that it would be very easy to find by browsing through the start of the ranked list. The target rhythm for "There is Nothing Like a Dame", on the other hand, is ranked 32,237, so it would not be found by a simple browsing of the suggestions. If we assume that a user of the system could browse through the top 30 suggestions, then the target rhythm appears in this list for 9 of our 30 sample phrases (23.1%).

Note that the number of possible rhythms depends on the number of syllables and the specified target duration of the entire phrase. For example, "Green Finch and Linnet Bird" has a duration of one 4/4 measure, yielding a space of 4282 rhythms. "I Feel You, Johanna", on the other hand, spans 4.5 measures, resulting in 6984 possibilities.

4.2. Discussion

What do these numbers mean? Remember that the goal is simply to suggest good rhythms—finding the target rhythm in the top 30 suggestions 23% of the time does not mean that the algorithm fails the other 77% of the time.

Lyric	Dur.	Rank	Rhythms	%
Green Finch and Linnet Bird	16	14	4282	99.7
Anyone can Whistle	16	164	4282	96.2
I Feel You, Johanna	72	731	6984	89.5
Something familiar	32	26	8572	99.7
The sun comes up	16	99	590	83.2
Nothing's gonna harm you	32	2605	49198	94.7
I Remember Sky	32	1450	8572	83.1
By the sea, Mister Todd	28	1527	35584	95.7
Here's to the ladies who lunch	24	1724	74369	97.7
Bit by Bit	16	22	136	83.8
Putting it together	24	2601	22333	88.4
Once upon a time(1)	12	3	576	99.5
Once upon a time(2)	20	764	3576	78.6
With so Little to be Sure	48	19354	487684	96.0
If there's anything at all	48	18481	487684	96.2
Some Enchanted Evening	16	14	4282	99.7
My Funny Valentine	32	12	49198	99.9
The Story of My Life	32	9165	49198	81.4
When I Think of Tom	32	444	8572	94.8
There is Nothing Like a Dame	32	32237	230196	86.0
It's a very ancient	32	3973	49198	91.9
We Kiss in a Shadow	32	2273	49198	95.4
We hide from the moon	32	1020	8572	88.1
He will not always say	32	1204	49198	97.6
Alone and awake	40	857	8131	89.5
I Have Dreamed	20	24	161	85.1
We've just been introduced	44	1586	64150	97.5
Shall We Dance?	24	19	171	88.9
The Face I See	40	52	787	93.4
Dreams, foolish dreams	28	10	1214	99.2

Table 1. Total phrase duration (in 16th notes), rank, # of possible rhythms, and percentile of target rhythm for each input phrase. The number of possible rhythms varies based on the number of syllables and the target duration.

Good suggestions other than (or very similar to) the target can still occur at the top of the list. The average percentile score of 92% suggests that the lyrics impose severe constraints on likely-composed rhythms, which are partially captured in our heuristics.

To improve performance, the scoring function could be improved or additional constraints added. For the latter case, consider the two instances of “Once Upon a Time”. These are the same motive, repeated in canon in the score. However, the first instance begins on the downbeat and lasts exactly three beats, while the second instance begins on beat three—giving it a longer target duration (we include two beats of rest). Constraining the motive to three beats causes the target rhythm to appear very close to the top of the list. Relaxing the constraint and allowing a longer total duration increases the size of rhythm space significantly and reduces the percentile score. In our rhythm suggestion application, it is reasonable to allow the user to direct the search by imposing additional constraints.

5. FUTURE WORK

An obvious way to improve this algorithm is to train the parameters of the scoring function using a machine learning technique along with a corpus of data providing (lyric, rhythm) pairs. For example, the Linear Dynamic Programming method (Raphael and Nichols 2008) should be applicable here to learn the penalty values in Eqns. 6 and 7 as well as to assign relative weights to all the terms in Eqn. 2. Alternatively, certain parameters could be exposed to a user of the system to provide more explicit control over the style of the results. This would facilitate including additional heuristics relating to features such as syncopation, which our current rules penalize implicitly.

Also of critical interest is extending the algorithm to work with longer spans of text. The local heuristics described above fail to account for larger-scale structure. Consider the two adjacent phrases “We Kiss in a Shadow / We hide from the moon”. The target rhythm is nearly identical for these parallel phrases; a more sophisticated algorithm might prefer a similar rhythm for each, greatly reducing the search space. Repetition of rhythmic motives helps unify a longer rhythmic pattern, but without a heuristic encouraging such repetition, high-scoring rhythmic sequences may lack larger-scale structure.

6. ACKNOWLEDGEMENTS

This work was supported by the Center for Research on Concepts and Cognition at Indiana University and helpful discussions with Ian Knopke and Christopher Raphael.

7. REFERENCES

[1] Bernstein, L. *The Unanswered Question: Six Talks at Harvard.* Cambridge, MA: Harvard University Press, 1976.

[2] Jarret, S. and Day, H. *Music Composition for Dummies.* Hoboken, NJ: Wiley 2008.

[3] Kilgraff, A. “BNC database and word frequency lists”. 1998. URL: http://www.kilgarriff.co.uk/bnc-readme.html

[4] Peterik, J., Austin, D., and Bickford, M. *Songwriting for Dummies*. Hoboken, NJ: Wiley 2002.

[5] Raphael, C. and Nichols, E. "Training Music Sequence Recognizers with Linear Dynamic Programming," in *MML 2008 International Workshop on Machine Learning and Music*, Helsinki, Finland, 2008, pp 19-20.

[6] Temperley, D. *The Cognition of Basic Musical Structures*. Cambridge, MA: The MIT Press, 2001.

[7] Temperley, D. "Distributional Stress Regularity: A Corpus Study." *Journal of Psycholinguistic Research* 38, 75-92.

[8] Temperley, D. *Music and Probability*. Cambridge, MA: The MIT Press, 2007.