

HARMONY-BASED MELODIC SIMILARITY EVALUATION OF MOTIFS

Ruobai Wang, Les Atlas, Cuinn Fey

University of Washington

ABSTRACT

Music similarity evaluation has a broad usage, and a better algorithm for it is very valuable. Melodic similarity is an important aspect of music similarity. This paper proposes a melodic similarity function based on pitch differences of notes. Different from existing methods, our function also considers harmony and consonance, which are important characteristics of music similarity. Our similarity calculation is based upon the concept of a musical motif. A motif is a short melodic idea or pattern that can characterize a song. We use motifs because they are easy to identify, normalize, and compare. We also propose a song recommendation system that uses this similarity evaluation method. We will introduce an LSTM-based note transcriber for the human voice in songs and a motif finder using self-similarity matrices that we designed to support Simsong. Experiments show that each of our proposed modules perform better than well-known methods, and the system works well as a whole.

Index Terms— Melodic similarity, harmony, motif, song recommendation, note transcription

1. INTRODUCTION

The evaluation of music similarity is an important and interesting topic. Calculating the melodic similarity between songs can also show us the ontology of a melody, which means that we can trace back to the original source of a music segment. Musicians are influenced by the music they often listen to, so their compositions may reflect their interests in some of the popular melodies of their formative years. With a program that tracks music similarity we could draw an ancestral tree of a melody pattern showing how it was created, propagated, varied, or finally lost popularity. This module can also be used to give reference comments on whether one song plagiarized an idea from another.

Another example of application of music similarity, is to recommend similar music to listeners based on what they like. Today music is most dominantly provided by music streaming services such as Amazon Music, Spotify, Apple Music, Soundcloud, Bandcamp, and various others. One of the main ways in which these services compete is through music recommendation. Contemporary practices of music recommendation are based on metadata such as genre [1], user classification [2], or melodic similarity. We will propose a project that expands upon the latter.

We can expect some more practical usages of this project. First, it may provide a better recommendation algorithm for music content distributors, and help them make more profit.

Second, music listeners will start to appreciate the patterns of melody, rhythm, or genre they enjoy and begin to seek for new songs containing these patterns, developing their ability to recognize them. This also helps the non-famous musicians spread their songs and increase their influence.

For this project we have also designed an automatic note transcriber, which aims to transcribe the monophonic melody of the human voice from a song. This transcriber shall extract the pitch sequence from a song. Additionally, we have designed a motif finder module, which automatically finds the repeating patterns in a song, and locates the motifs for each pattern.

This paper will contain 7 chapters. In Chapter 2, we will be introducing our melodic similarity function. In Chapters 3 to 4, we will be introducing the other main modules for Simsong project: automatic note transcriber and motif finder. In Chapter 5 we will show the experiment results on different kinds of music. In Chapter 6 we will discuss our note transcriber and some future work for this project. In Chapter 7 we will make a conclusion for this paper.

2. HARMONY-BASED SIMILARITY FUNCTION

2.1. Literature review

We use melodic similarity to measure the similarity between songs. Songs are long, complex, and of varying lengths, so scoring the similarity between two whole songs will not yield any worth-while results. Consequently, we will use short repeating patterns in each song called motifs, and use the similarity between motifs to represent song similarity.

There are many existing methods to calculate melodic similarity. Among them, the edit-distance method and geometric-based method are simple and effective, which will be used in our experiments as baselines. We will also introduce some other similarities.

The edit-distance method [23] draws the pitch contour of each music clip on the time-pitch coordinate, and searches for the optimal time shift and pitch transpose in which the compared song pairs have the most overlap. The similarity

is the proportion of notes that are overlapping. In monophonic cases, the similarity equates to the proportion of time intervals in which the two compared clips have the same pitch. The edit-distance method can be described with a similarity function for each time interval: 1 if two notes have the same pitch, and 0 otherwise.

A variant of edit-distance method is the earth mover's distance (EMD) [24]. It regards each note as an object with weight equal to its time length. The difference between edit distance and EMD is, in the EMD method the weight of a note is concentrated at its starting time point, but in the edit-distance method the weight is evenly distributed over the duration of the note. We can see that, the EMD cannot be generalized to a similarity function for time intervals like the original edit-distance method does.

The geometric-based method [25] is designed for monophonic song clips. It draws the pitch contours of the two compared clips, searches for time shifts and pitch transposes to find the minimum of the area between the two curves, and uses this area as the dissimilarity. To calculate the area, we sum the pitch differences in all time intervals. Similar to the edit-distance method, the geometric-based method can also be described by a similarity function for each time interval, which starts at a high value for notes with the same pitch and linearly decreases when the pitch distance increases.

This paper [26] shows a group which goes further on the geometric-based method. The method fixes the pitch of notes at their start, and then uses interpolation to get the pitch contour. Next, it calculates the first-order derivative of the pitch contour and finds the area between the derivative curves of two song clips. We decided not to use it, because this similarity cannot be calculated note by note.

The time-insensitive version of the edit-distance method [27] does not consider the time length of each note. Time-insensitive method has many variants that compare different kinds of tuples instead of pitches, such as Implication/Realization (I/R) model [28], which compares the trend of pitches instead of the pitch value. Another example is the N-gram method [29], which regards each note and several consecutive notes of that note as a whole. There are two styles of the N-gram method, one requires the two N-grams to be exactly the same, and the other allows permutation within an N-gram. However, we do not use these methods, because they may cause length mismatch when we are comparing fixed-length motifs.

The importance of harmony in melodic similarity has not been thoroughly studied. One group [30] conducted an experiment to evaluate the importance of harmony, and based on the results, they claimed that the distance of pitch contour is primary, and harmony is secondary in melodic similarity. However, their experiment only showed that pitch contour is more important than whether the harmony is "close". In the group's experiments, listeners gave high similarity scores to songs with the same chord sequence and distal pitch contours; Music experts judged songs with same

chord sequences and distal pitch contours as equally similar to songs with distant chord progressions and proximal pitch contours. Therefore, this research actually proved the importance of harmony in melodic similarity.

2.2. Method and Result

Given a function of pitch similarity, we can take the average of the pitch similarities of all time intervals to get the melodic similarity between motifs. Therefore, what we need now is a pitch similarity function. Here we show how we get this pitch similarity function.

We use 40 clips from verse and chorus section of 20 Chinese songs as training data. Each clip has 8 bars, and the tonality is normalized to C major. Both melodies and chords are labelled manually. The time interval for pitch similarity is set to 1/8 bar, which is also the time unit of melody notes. The time unit for chords is half a bar, so there are 16 chords in each clip.

For each pair of clips, we calculate the pitch distance of each time interval, and collect them to form a histogram. Then we calculate the chord similarity of this clip pair by counting the coincident chord notes, which is similar to the polyphonic edit-distance method without fixed time shift and pitch transpose. We set the melodic similarities as follows: for each pair that contains two different motifs, the melodic similarity is set to half of this pair's chord similarity. For each pair that contains identical motifs, the melodic similarity is set to 1.

Then we collect the histograms and melodic similarity values to get a histogram list $B_{N \times H}$ and a similarity value list $c_{N \times 1}$. Here N is the number of possible pairs, and H is the number of bins for the pitch distance histogram.

We want to get the pitch similarity function $x_{H \times 1}$ which can minimize $\|Bx - c\|$. If we take the $L2$ norm, then it becomes the simplest form of quadratic optimization problem, and the solution is $x = (B^T B)^{-1} B^T c$.

However, large pitch distances (more than 1 octave) are very rare, so this problem is not well-conditioned for them. Therefore, to avoid errors, we need a penalty term a for them, where a is larger for pitch distances that occur fewer times. For example, we can let

$$a^2 \sum_{i=1}^N B_{i,j} = N$$

Then what we want to minimize is $\|Bx - c\|_2^2 + \|Ax\|_2^2$, where $A = \text{diag}(a)$, and the result is

$$x = (B^T B + A^2)^{-1} B^T c$$

The optimization result is shown in Figure 2.1. Compared with the functions of edit-distance and geometric-based methods, we can see that it is not monotonic and has a high similarity values for distant but consonant pitches.

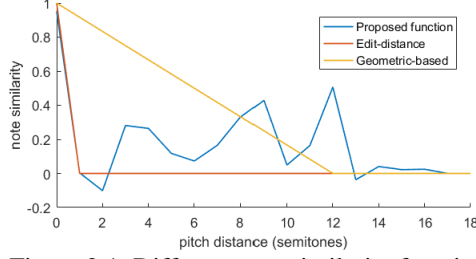


Figure 2.1: Different note similarity functions

3. AUTOMATIC NOTE TRANSCRIBER

3.1. Literature review

There have been many musical pitch tracking systems using probabilistic models on spectrograms [3][4][5][6] or neural networks[7][8]; however, very few researchers have been working on pitch tracking for human singing voices[9], even though the human voice is very important to perceived melody in music. Pitch tracking for speech is well defined[10][11], but distinguishing human voices from musical accompaniment is still an obstacle.

In most cases, musical instruments have lower fundamental frequency than human singing voices, which makes separation possible. However, the frequency range of the human voice and music accompaniments often overlap, so it is best not to use a hard threshold. One idea is to use source separation methods to extract human voices from a song. There are some mature methods [12][13][14][15][16], but they are still not accurate enough for a pitch tracker.

We propose another idea, which is to use a neural network to do both source separation and pitch tracking at the same time. We choose to use an LSTM network [17][18] in our project. LSTM networks takes a vector for each frame as input and outputs another vector for each frame.

3.2. Method

We use MIR-1K database [19] as our training data. MIR-1K is a dataset which contains 1000 clips from several Chinese popsongs. Each clip is about 10 seconds in length. The fundamental frequency ranges from midi numbers 35 (61.7Hz) to 77 (698.5Hz). It uses 40ms frame length, 20ms frame shift, and the existence and value of fundamental frequency is manually labeled for each frame.

For the input features, we are using constant-Q spectrogram [20] for both Fourier spectrogram and the output of YIN method [21]. Constant-Q spectrogram for FFT is simple: according to the spectrum, we calculate the energy in each frequency bin. Each frequency bin is one semitone wide, and the central frequency corresponds to an integer midi number.

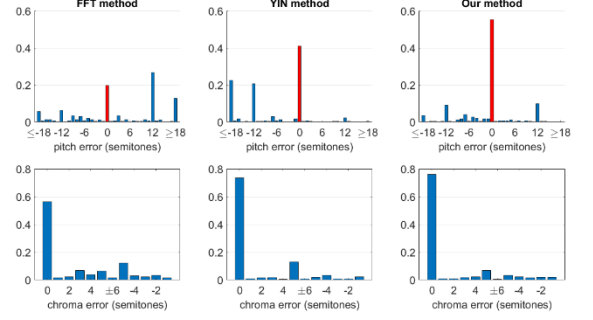


Figure 3.1: Performance of different transcribers

Constant-Q spectrogram for YIN method is calculated as follows: First we get the output of the YIN method, which is a function that indicates how likely it is that each time length can be the fundamental period of the present frame. Then we simply look up the YIN output for the period corresponding to the central frequency of each bin.

FFT spectrum has high energy in frequency bins that are harmonies of f_0 , while the bin of f_0 itself does not necessarily have high energy. YIN spectrum has high energy in the f_0 bin, as well as bins in harmony with f_0 . If both spectra are used as input features, we can expect to eliminate most octave errors in both directions.

The output vector of this LSTM network, is a probability vector that indicates the probability of this frame to have a f_0 corresponding to each midi number or no f_0 at all. In the network training, we use a one-hot vector, because we have the ground truth in the database.

Some songs in the database, from the western hemisphere, use guitar riffs as motifs, which have no human voice. They would be classified as "no fundamental frequency" by this network because it is designed for human voice. However, we need the fundamental frequency of guitar for note transcription. In this case, we can apply the pitch-tracking methods for pure music mentioned above.

As a note transcriber, we want it to identify the length and starting time for each note, instead of just getting the pitch contour for the song framed with a fixed frame shift. This can be done with three steps: first, we calculate the tempo of the song, to know the length of a normalized short time interval, such as one 16th note; second, we get the starting point of these time intervals, where each actual note should fall on one of these points. Third, we get the pitch for each time interval, which indicates the pitch for each 16th note.

Notice that we do not know whether two consecutive time intervals with the same pitch actually belong to the same note. The reason is, we will be using time-sensitive distance measurements in the similarity calculation, and we will separate each note into short time intervals.

To get the pitch contour, we simply average the likelihoods of all frames contained in each time interval. For the estimation of tempo and start points of notes, we use the novelty curve method [22].

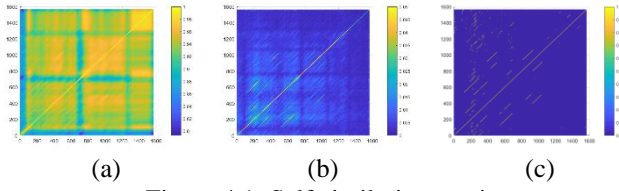


Figure 4.1: Self-similarity matrix

3.3. Evaluation of performance

I tested the note transcriber on the Chinese songs that I manually transcribed. The result is shown in Figure 3.1.

We can see that, our method achieves the highest pitch accuracy of 55.3%, while YIN method has 41.1% and FFT method has 19.7%. Our chroma accuracy is also the highest, which reached 76.0%, while YIN is 73.4% and FFT method is 56.3%.

Also we observed that, FFT method tend to have higher octave errors and -5 chroma error, while YIN method tend to have lower octave error and +5 chroma error. Our method has a good balance between these two methods.

4. AUTOMATIC MOTIF FINDER

There are a lot of papers about music pattern discovery, but we are interested in a self-similarity matrix [31].

To calculate the self-similarity matrix for a song, we first calculate the tempo, and then set the frame length and frame shift to be an 8th note (1/8 bar). Then we calculate its constant-Q spectrogram, and for each frame, we stack the constant-Q spectra for itself, and its 63 consecutive frames, to form the feature vector for self-similarity calculation.

The reason we choose 32 frames (4 bars) as feature length, is that we assume each motif has 4 bars. Though different songs have different length of motifs, a 4-bar motif length is a balance for all music genres.

The self-similarity matrix often has high and flat values in coda passages, as shown in Figure 4.1(a). To avoid this problem, we apply a high-pass filter on the matrix, and the result is shown in Figure 4.1(b). In this figure we see many diagonal lines, which indicate that a certain clip of this song is similar to another clip.

We only keep the points whose similarity value is among the top three in its row or column, and the matrix becomes Figure 4.1(c). There are still some remaining diagonal lines. Each diagonal line that is long enough corresponds to a motif candidate, and its projection onto the x and y axes show the time position of the candidate itself and its most similar counterpart.

After that, we pick one clip from each similar group, and discard the duplicate ones. Each taken clip becomes a raw motif area.

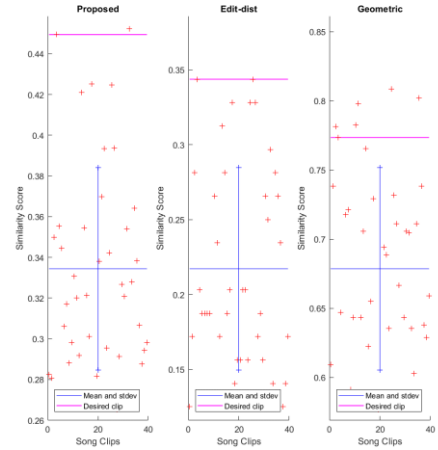


Figure 5.1: Performances of different similarity calculation methods on Eastern music

Then we will find the accurate starting time of each motif. For each raw motif area, we take the frame that has the highest cross-frame similarity as the peak and search for each frame within half bar around the peak.

We assume that a motif should be similar to a few existing motifs, so for each possible motif starting from a certain frame, we calculate its similarity values with all existing motifs in the database for motif locating, and then take the average of the highest 10 values. The motif with the highest average similarity will be selected as the motif of this raw area.

5. EXPERIMENT

5.1. Experiment settings

First let me introduce the terms we use in this experiment. We refer to all possible music in the database as "music in the wild". The input, which we want to find motifs similar to it, is the "reference motif". The motifs in the database that have high similarities with the reference motif, are called "matches". Among the matches, those that are similar with the reference, are called "good matches", and if a motif is already known as similar to the reference, then it is called "the ideal match". The ones not similar with the reference are called "garbage matches".

The experiment will contain two parts, the Eastern part and the Western part. Automatic transcription technique is applied for both parts. In the Eastern part, we have one reference motif, and a database of 40 Eastern music motifs which contains one ideal match. We evaluate the performance of similarity-calculation methods, by normalizing the mean and standard deviation of all similarity values, and see how high the similarity value of the ideal match is.

Method	Similar pairs		Dissimilar		Fisher's score
	Mean	Stdev	Mean	Stdev	
Our proposed	0.74	0.165	0.49	0.042	2.86
Edit-distance	0.68	0.216	0.34	0.065	2.50
Geometric	0.89	0.102	0.79	0.057	0.44
Spectrogram	0.68	0.054	0.65	0.068	0.06

Table 5.1: Performances of different similarity calculation methods on Western music.

In the western part, the database contains 12 motifs from Western music. A musician classified each pair of motif as "highly similar", "marginally similar" and "not similar". We calculate the similarity values of all pairs, and evaluate the performance using the Fisher score [32] of similarity values of "highly similar" and "not similar" groups.

Four different similarity functions are compared. One is our proposed method. Then the edit-distance and geometric-based methods. The last one is the cosine similarity of whitened constant-Q spectrogram.

5.2. Results on Eastern music

Results are shown in Figure 5.1.

Our method achieves a score of 2.21, which means that the similarity value between the reference motif and its ideal match is 2.21 standard deviations above the average. Higher score means better performance. As a comparison, edit distance method, which is a good baseline, scored 1.88. The score for geometric-based method was only 1.30.

5.3. Results on Western music

Results are shown in Table 5.1.

We can see that, our method achieves the best Fisher's score, much higher than the others. Though the performance is low when the motifs are automatically transcribed, our method is still better than other methods, and we argue that the low performance is caused by the transcriber, which is trained on Eastern music and cannot work well for Western ones.

6. FUTURE WORK

Our project is still a starting version, and has room for improvement. Let me talk about some possible research topics.

A more accurate note transcriber is needed. Currently our automatic transcriber has an acceptable performance on Eastern music. However, on Western music, especially the guitar riffs, it has a lot of error. I think the reason is the database we are using. MIR-1K dataset is designed for pitch estimation of human voice in real songs, so it labeled sounds from music instruments as "no fundamental frequency".

Also, MIR-1K only contains monophonic Eastern songs. To improve its performance, we need a dataset that contains songs from different genres and origins. We also need to add a function to transcribe the music when a note is labeled as "no human voice". Finally, the transcriber needs to be modified to support polyphonic songs.

An answer is still needed regarding the relationship between pitch difference and note similarity. We have seen that, a similarity function which assigns higher similarity values to consonant pitch intervals can work better than the similarity functions that are linear to pitch difference. However, there are two problems. First, the similarity values in our experiments are only empirical, and need theoretical explanations. Second, the similarity values in the training data were arbitrary, and we need human tests to create a database with similarity values that are commonly agreed.

Last but not least, the motif discovery is also an important topic. Now we can easily locate the approximate position of motifs in verses and choruses, however the motif discovery in intro or bridge parts often fails. It is also hard to find the accurate time point where a motif starts. We have tried to assign chords and recover the information about bars, and the performance was acceptable on Eastern pop songs. However, the assumption that chords are linked with bars may not hold true on other music genres, so we decided to switch to our current methods. The performance of this method is also not perfect. Since that an accurate automatic motif finder is important for database construction, this is a good direction for future research.

7. CONCLUSION

In this paper, we introduced the three main modules of Simsong project: the LSTM-based note transcriber, the harmony-based melodic similarity function, and the motif finder. The note transcriber is a good improvement to FFT and YIN method, which combines the information from both spectrograms, and avoids massive octave errors. The melodic similarity function has a stably better performance than edit-distance method, the best baseline method in our experiment, and justifies the importance of harmony in similarity evaluations. The motif searcher works well with the other two modules, and they together compose the Simsong system that works as intended.

To improve the performance of Simsong, we expect to see progress in the following aspects in the future: Improvement in accuracy and universality of the note transcriber; Theory-level explanation, and quantification endorsed by human tests, of how harmony affects melodic similarity; Improvement in accuracy of locating motifs and music structures.

8. REFERENCES

- [1] E. Pampalk, S. Dixon, and G. Widmer, "On the evaluation of perceptual similarity measures for music," in of: Proceedings of the sixth international conference on digital audio effects (DAFx-03), 2003, pp. 7-12.
- [2] A. Berenzweig, B. Logan, D. P. Ellis, and B. Whitman, "A large-scale evaluation of acoustic and subjective music-similarity measures," *Computer Music Journal*, vol. 28, pp. 63-76, 2004.
- [3] G. E. Poliner, D. P. Ellis, A. F. Ehmann, E. Gómez, S. Streich, and B. Ong, "Melody transcription from music audio: Approaches and evaluation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 1247-1256, 2007.
- [4] K. Dressler, "Extraction of the melody pitch contour from polyphonic audio," in *Proc. 6th International Conference on Music Information Retrieval*, 2005.
- [5] M. Goto, "A real-time music-scene-description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication*, vol. 43, pp. 311-329, 2004.
- [6] E. Vincent and M. Plumbley, "Predominant-F0 estimation using Bayesian harmonic waveform models," in *2005 Music Information Retrieval Evaluation eXchange (MIREX)*, 2005.
- [7] S. Böck and M. Schedl, "Polyphonic piano note transcription with recurrent neural networks," in *Acoustics, speech and signal processing (ICASSP), 2012 IEEE international conference on*, 2012, pp. 121-124.
- [8] D. Troxel, "Music transcription with a convolutional neural network 2016," *Music Information Retrieval Evaluation eXchange (MIREX)*, 2016.
- [9] E. Pollastri, "A pitch tracking system dedicated to process singing voice for music retrieval," in *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on*, 2002, pp. 341-344.
- [10] S. Wisdom, T. Powers, L. Atlas, and J. Pitton, "Enhancement and recognition of reverberant and noisy speech by extending its coherence," *arXiv preprint arXiv:1509.00533*, 2015.
- [11] H. Kawahara, J. Estill, and O. Fujimura, "Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system STRAIGHT," in *Second International Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications*, 2001.
- [12] Y. Wang and Z. Ou, "Combining HMM-based melody extraction and NMF-based soft masking for separating voice and accompaniment from monaural audio," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, 2011, pp. 1-4.
- [13] Z. Rafii, "Source Separation by Repetition," Northwestern University, 2014.
- [14] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, pp. 2136-2147, 2015.
- [15] J.-L. Durrieu, G. Richard, and B. David, "An iterative approach to monaural musical mixture de-soloing," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, 2009, pp. 105-108.
- [16] Y. Li and D. Wang, "Separation of singing voice from music accompaniment for monaural recordings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 1475-1487, 2007.
- [17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735-1780, 1997.
- [18] A. Mallya, LSTM Forward and Backward Pass. Available: <http://arunmallya.github.io/writeups/nn/lstm/index.html#/>
- [19] C.-L. Hsu and J.-S. R. Jang, MIR-1K Dataset. Available: <https://sites.google.com/site/unvoicedsoundseparation/mir-1k>
- [20] J. C. Brown, "Calculation of a constant Q spectral transform," *The Journal of the Acoustical Society of America*, vol. 89, pp. 425-434, 1991.
- [21] A. De Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, pp. 1917-1930, 2002.
- [22] P. Grosche and M. Müller, "Computing predominant local periodicity information in music recordings," in *Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA'09. IEEE Workshop on*, 2009, pp. 33-36.
- [23] K. Lemström, N. Mikkilä, V. Mäkinen, and E. Ukkonen, "Sweepline and recursive geometric algorithms for melodic similarity," *MIREX 2006*, p. 10, 2006.
- [24] R. Typke, F. Wiering, and R. C. Veltkamp, "Evaluating the earth mover's distance for measuring symbolic melodic similarity," in *MIREX-ISMIR 2005: 6th International Conference on Music Information Retrieval*, 2005.
- [25] G. Aloupis, T. Fevens, S. Langerman, T. Matsui, A. Mesa, Y. Nunez, et al., "Computing a geometric measure of the similarity between two melodies," in *Proceedings of the 15th Canadian Conference on Computational Geometry*, 2003, pp. 81-84.
- [26] J. Urbano, J. Lloréns, J. Morato, and S. Sánchez-Cuadrado, "Mirex 2010 symbolic melodic similarity: Local alignment with geometric representations," *Music Information Retrieval Evaluation eXchange*, 2010.
- [27] A. L. Uitendbogerd, "Variations on local alignment for specific query types," *International Music Information Retrieval Systems Evaluation Laboratory* [10]. URL <http://www.music-ir.org/mirex2006>, 2006.

- [28] M. Grachten, J.-L. Arcos, and R. L. De Mántaras, "Melodic similarity: Looking for a good abstraction level," *representations*, vol. 2, p. 7, 2004.
- [29] I. S. Suyoto and A. L. Uitdenboger, "Simple efficient n-gram indexing for effective melody retrieval," *Proceedings of the Annual Music Information Retrieval Evaluation exchange*, 2005.
- [30] E. Schubert and C. Stevens, "The effect of implied harmony, contour and musical expertise on judgments of similarity of familiar melodies," *Journal of New Music Research*, vol. 35, pp. 161-174, 2006.
- [31] J. Foote, "Visualizing music and audio using self-similarity," in *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, 1999, pp. 77-80.
- [32] Q. Gu, Z. Li, and J. Han, "Generalized fisher score for feature selection," *arXiv preprint arXiv:1202.3725*, 2012.