

# RHYTHMIXEARCH: SEARCHING FOR UNKNOWN MUSIC BY MIXING KNOWN MUSIC

**Makoto P. Kato**

Department of Social Informatics, Graduate School of Informatics  
Kyoto University, Kyoto, Japan  
kato@dl.kuis.kyoto-u.ac.jp

## ABSTRACT

We present a novel method for searching for unknown music. *RhythMiXearch* is a music search system we developed that can accept two music inputs and mix those inputs to search for music that could reasonably be a result of the mixture. This approach expands the ability of *Query-by-Example* and allows greater flexibility for users in finding unknown music. Each music piece stored by our system is characterized by text data written by users, i.e., review data. We used *Latent Dirichlet Allocation* (LDA) to capture semantics from the reviews that were then used to characterize the music by *Hevner's* eight impression categories. *RhythMiXearch* mixes two music inputs in accordance with a probabilistic mixture model and finds music that is the most likely product of the mixture. Our experimental results indicate that the proposed method is comparable to human in searching for music by multiple examples.

## 1. INTRODUCTION

Much music content has become available, and music analysis and retrieval systems have recently been rapidly developing. To make finding music easy, many prototype systems for searching for music pieces by using content-based IR techniques have been proposed [17] [6]. They enable users to find music by inputting an audio file as a query, called *Query-by-Example* (QBE), in particular inputting by humming, i.e., *Query-by-Humming* (QBH) [4]. Based on the input audio signals, QBE systems retrieve music by calculating the similarity between the queried music piece and stored music and then return the results in the order of similarity to the query. Searching by example is helpful for obtaining new music similar to music that you have or that you have heard.

However, these content-based IR methods are not able to meet the specific needs of users wanting to find music they have never heard. A common situation is that you want to find a certain piece of music which you imagine

in your mind, but have neither the keywords related to it, music similar to it, nor the ability to sing it. In addition, content-based approaches rank at the top only music similar to what you know well, so you cannot find music very different from yours; the opportunity to discover new music is lost. This is caused by the lack of flexibility in inputting queries. As the amount of digital music content increases, finding the precise music you want requires higher expressiveness of queries.

We present a novel approach to searching for unknown music. *RhythMiXearch* is a music search system we developed that can accept two or more music inputs. By mixing the input music, it searches for music that could reasonably be a result of the mixture. This approach expands the ability of *Query-by-Example* and allows greater flexibility for users in finding unknown music. For example, intuitively, *RhythMiXearch* can introduce music similar to *The Beatles' Let It Be + Coldplay's Viva la Vida* to you.

Stored music in *RhythMiXearch* is characterized on the basis of users' impressions. We retrieved review data on *Amazon.com*, analyzed the text data by using *Latent Dirichlet Allocation* (LDA) [2], and determined the impressions that users received from the music.

There is a strong reason that users' impressions were used as a feature of the music and were extracted from the review data rather than the features of the music itself being used. Consider music that users do not know but want to find. The mood or impression the music will give users is more important than the timbre [1] or rhythm [5] [12] it has. Users are likely not able to imagine the details of the wanted music, such as the timbre and rhythm; they only feel the sense of the music they want, such as the mood and impression. In addition, in our approach to mixing input music, picturing mixtures of timbre and rhythm would be difficult, and for users, the result may not be what is expected or wanted. For detecting the impression given by music pieces, seeing review text written by humans about the music would be more effective than analyzing the music itself. Mood detection by signal analysis has been proposed [15] [14]. However, the final feeling we get from listening to music is a product of knowing the title and artist, listening to the melody, understanding the lyrics, and so on; simply analyzing the timbre and rhythm of a piece is not enough for estimating what listeners will feel. In contrast, reviews are provided by music listeners, so analyzing review text rather than the music itself would be helpful for

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

© 2009 International Society for Music Information Retrieval.

determining the impression given by the music.

Input music is combined based on the features of the music represented by the estimated impression, and our system ranks its stored music pieces by their likelihood of reasonably being a result of the combined input music. Situations in which multiple examples could be used include the following: searching for music that has all the features of multiple music inputs, and searching with multiple inputs of your favorite music. For these situations, we developed a method to combine two music inputs in one query. We named the multiple input query *Query-by-Mixture-of-Examples*.

## 2. RELATED WORK

Characterizing music by using text data has been reported recently. Knees et al. used Web documents to develop a system that searches for music pieces through natural language queries [11] and also presented a method to combine signal-centered features with document-centered ones [9]. They characterized music pieces by using a conventional IR approach, which is the *Vector Space Model* with tf-idf method. In addition to searching for music, artist classification [10] was done by the same text-based approach with the SVM. Pohle et al. [13] describe artists by common topics or aspects extracted from Web documents. A browser application they presented enables users to formulate a query to search for desired artists by simply adjusting slider positions.

Turnbull et al. [16] focused on natural language queries such as “female lead vocals”, called *Query-by-Semantic-Description* (QBSD). In their approach, the *Computer Audition Lab 500-Song* (CAL500) data set was used to learn a word-level distribution over an audio feature space. QBSD can search for music pieces unfamiliar to users, which is the same aim as ours. Terms used as queries to illustrate music, however, are limited with regard to amount and cannot capture subtle nuances to search for wanted music.

Music Mosaics [18] is a concept for creating a new query by concatenating short segments of other music pieces. It applies the signal analysis technique to characterize music and represents pieces of the music by thumbnails. Querying with multiple music pieces in music mosaics is quite similar to our method, but as mentioned above, making a query by assembling pieces of signal information to find unfamiliar music is difficult.

Similar to our approach, MusicSense [3] is a music recommendation system for users reading Web documents such as Weblogs. It adopted a generative model called *Emotional Allocation Modeling* to detect emotions of documents and music with the text. In this model, a collection of terms is considered as generated over a mixture of emotions, like the LDA approach.

## 3. METHODOLOGY

At first, we propose a framework of our approach. Then, we explain song characterization with reviews by using

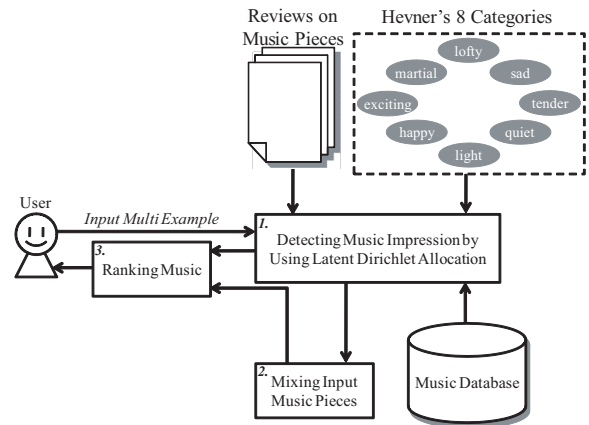


Figure 1. Framework of our approach.

		1		
		spiritual		
		lofty		
	8	awe-inspiring	2	
	robust	dignified	pathetic	
	emphatic	sacred	doleful	
	martial	solemn	sad	
	ponderous	sober	mournful	
	majestic	serious	tragic	
	exalting		melancholy	
7			frustrated	3
exhilarated			depressing	dreamy
soaring			gloomy	yielding
triumphant			heavy	tender
dramatic			dark	sentimental
passionate				longing
sensational				yearning
agitated				pleading
exciting				plaintive
impetuous				
restless				
	6	5	4	
	merry	humorous	lyrical	
	joyous	gay	leisurely	
	happy	playful	satisfying	
	cheerful	whimsical	serene	
	bright	fanciful	tranquil	
		quaint	quiet	
		sprightly	soothing	
		delicate		
		light		
		graceful		

Figure 2. 8 sets of impression words proposed by Hevner. Adjacent sets are similar impressions, and opposite ones are counter-impressions.

LDA, probabilistic mixture model for combining input music pieces, and ranking music pieces by the similarity.

### 3.1 Framework of Our Approach

The framework of our approach is shown in Fig. 1. It consists of three steps: (1) detecting impressions of music pieces by using LDA from music reviews, (2) mixing input music pieces on the basis of the impressions, and (3) ranking stored music pieces by their likelihood of being the result of the mixture.

For extracting impressions from music reviews, we used a generative model named LDA, in which it is assumed that terms in a document are generated by a mixture of topics, i.e., multinomial distributions over topics. The assumption enables us to conjecture the fundamental meanings of documents, and the meanings are represented by the topic distribution for each document.

The sets of impression words for music proposed by Hevner [8] are shown in Fig. 2. The impression words are used to find which impression a review gives in the genera-

tive model, in intuitive terms, by calculating the similarity between reviews and the impression words, where we regard the sets of impression words as documents. We obtain the probability that each distribution over topics for a document would generate a set of impression words if only Hevner's sets of impression words were provided.

Given multiple music inputs, we mix on the basis of the impression probability. Different mixture models are proposed for different situations. Finally, the results from the stored music pieces are returned, ranked by the similarity to the mixture of multiple examples. One easy method is the similarity-based ranking between stored music and the virtual music created as a result of a mixture. We apply this method to our system and introduce a prototype system based on the framework.

### 3.2 Characterizing Songs by Reviews

First, we introduce a method to characterize songs by analyzing text review data with LDA. In the LDA analysis, terms in a document are assumed to be generated from a topic and topics allocated to words are chosen from multinomial distributions for the documents. Each multinomial distribution is selected from the Dirichlet distribution, which is often adopted as a prior distribution for a multinomial distribution.

The LDA generative process consists of choosing parameters for each document  $\mathbf{w}$  as follows.

1. Choose  $\theta \sim \text{Dirichlet}(\alpha)$ .
2. For each  $i^{\text{th}}$  word  $w_i$  in document  $\mathbf{w}$ ,
  - (a) choose a topic  $z_i \sim \text{multinomial}(\theta)$ , and
  - (b) choose a word  $w_i$  from  $p(w_i|z_i, \beta)$ , a multinomial distribution conditioned on the topic  $z_i$ ,

where  $\alpha$  and  $\beta$  are hyper-parameters for a corpus that was assumed to be previously fixed in this paper,  $\theta$  is determined for a document, and  $w_i$  and  $z_i$  for a word.

The probability over the  $i^{\text{th}}$  word for a multinomial distribution  $\theta$  is given by

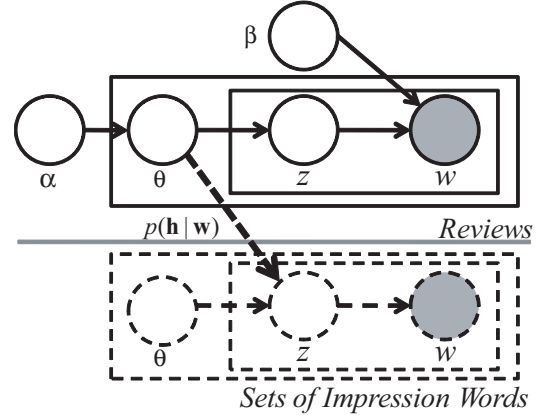
$$p(w_i|\theta, \beta) = \sum_{z_i} p(w_i|z_i, \beta)p(z_i|\theta). \quad (1)$$

The probability  $p(z_i|\theta)$  characterizes a document by the topics, which have lower dimensions  $K$  than the words. Each topic is represented by the word-occurrence  $p(w_i|z_i, \beta)$ .

With multiplication of all the  $N$  words in a document  $\mathbf{w}$  and integration over  $\theta$ , the occurrence distribution of a document  $\mathbf{w}$  is computed as

$$p(\mathbf{w}|\alpha, \beta) = \int p(\theta|\alpha) \left( \prod_{i=1}^N \sum_{z_i} p(w_i|z_i, \beta)p(z_i|\theta) \right) d\theta. \quad (2)$$

Taking the product of all the documents in a corpus, we obtain the occurrence probability of the corpus. We use the Gibbs sampling technique [7] to estimate the parameters for the probability of the corpus and obtain the approximate distribution  $p(w_i|z_i, \beta)$  and the parameter  $\theta$ , which is allocated to each document.



**Figure 3.** Graphical model representation of Latent Dirichlet Allocation and of detecting impressions given by music reviews. The upper outer rectangle represents reviews, and the inner rectangle represents the chosen topics and words in a review. The bottom outer rectangle represents sets of impression words. We estimate impressions of music by calculating the probability  $p(\mathbf{h}|\mathbf{w})$  that a multinomial distribution for a review  $\mathbf{w}$  generates a set of impression words  $\mathbf{h}$ .

After analyzing a corpus, we calculate the probability that a topic distribution for a document would generate a set of impression words. The distribution is denoted by  $p(\mathbf{h}|\mathbf{w})$ , where  $\mathbf{h}$  is a variable for Hevner's sets of impression words  $H$  and is one of the sets. A graphical model representation of LDA and of detecting impressions given by documents is shown in Fig. 3. Through Bayes' theorem,  $p(\mathbf{h}|\mathbf{w})$  is represented by only the product  $p(\mathbf{w}|\mathbf{h})p(\mathbf{h})$ :

$$p(\mathbf{h}|\mathbf{w}) = \frac{p(\mathbf{w}|\mathbf{h})p(\mathbf{h})}{\sum_{\mathbf{h}} p(\mathbf{w}|\mathbf{h})p(\mathbf{h})}, \quad (3)$$

where parameter  $\beta$  is omitted and  $p(\mathbf{h})$  is assumed to be the same for all  $\mathbf{h}$  in  $H$ .

The probability  $p(\mathbf{w}|\mathbf{h})$  is divided by the latent parameters or the topics:

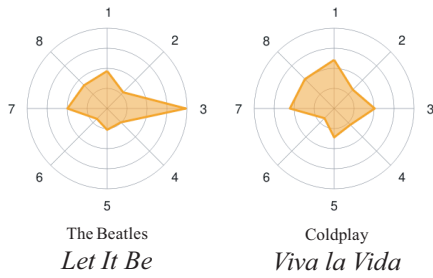
$$p(\mathbf{w}|\mathbf{h}) = \prod_{i=1}^N \sum_{z_i} p(w_i|z_i)p(z_i|\theta_{\mathbf{h}}). \quad (4)$$

$\theta_{\mathbf{h}}$  is a parameter of a multinomial distribution for a set of impression words  $\mathbf{h}$ , which is estimated regarding the set as a document.

Finally, summing up over all documents for a music piece, i.e., reviews, we obtain the probability  $p(\mathbf{h}|m)$  that a music piece  $m$  generates an impression  $\mathbf{h}$ :

$$p(\mathbf{h}|m) = \sum_{\mathbf{w} \in D_m} p(\mathbf{h}|\mathbf{w})p(\mathbf{w}|m), \quad (5)$$

where  $D_m$  is a collection of reviews for music piece  $m$  and we assume the same distribution for  $p(\mathbf{w}|m)$ , i.e.,  $1/|D_m|$ . The probability  $p(\mathbf{h}|m)$  can be explained as an impression represented by a set of words  $\mathbf{h}$  at the probability  $p(\mathbf{h}|m)$  obtained by a user listening to music  $m$ .



**Figure 4.** The left chart represents *Let It Be* by *The Beatles*. The right chart represents *Viva la Vida* by *Coldplay*. The numbers correspond to those in Fig. 2.

Examples are shown in Fig. 4. The reviews for the two pieces of music were downloaded from *Amazon.com*, and the probability  $p(\mathbf{h}|m)$  was visualized by *Google Chart API*<sup>1</sup>.

There are two reasons we put the topic distributions into eight impression categories. First, to measure the similarity between music pieces effectively, we should select the most suitable topic, i.e., give weight to topics that strongly represent the music features and reduce the weight of those that do not relate to the features. This is because all the topics do not necessarily represent features of the music, e.g., a topic may simply indicate that a music piece is expensive. Second, to convey to users why the specific results were returned, the music must be visualized in some way. This is important particularly in a situation when a user wants to find unknown music.

### 3.3 Probabilistic Mixture Model

In the previous subsection, we characterized music pieces by  $p(\mathbf{h}|m)$ , which is the probability that the music  $m$  gives an impression  $\mathbf{h}$  represented by some adjectives. On the basis of this probability, two music pieces input by users are combined and a new probability for the result of the mixture is generated. A basic method is to compute the average of two given distributions  $p(\mathbf{h}|m_x)$  and  $p(\mathbf{h}|m_y)$ , i.e.,  $\{p(\mathbf{h}|m_x) + p(\mathbf{h}|m_y)\} / 2$ . However, this is likely to provide a flattened distribution whose probabilities are similar. An ordinary average operation has a potential problem: a remarkable feature on the distribution may be ignored in the result of the combination. Thus, we propose two mixing operations for two input distributions that can be used in different situations.

#### 3.3.1 Feature-preserved Mixture

To combine two music pieces while preserving their features, we suppose the following probabilistic process.

1. Choose one of two input music pieces at a 1/2 probability.
2. Repeat two impression extractions from the chosen music until the extracted impressions converge.

3. Adopt the concurrent impression as the result of the mixture for the two music pieces.

The process is given by the following equation:

$$p(\mathbf{h}|m_z) = \frac{1}{2} \left\{ \frac{p(\mathbf{h}|m_x)^2}{\sum_{\mathbf{h}} p(\mathbf{h}|m_x)^2} + \frac{p(\mathbf{h}|m_y)^2}{\sum_{\mathbf{h}} p(\mathbf{h}|m_y)^2} \right\}, \quad (6)$$

where  $p(\mathbf{h}|m_x)$  and  $p(\mathbf{h}|m_y)$  are the distributions over the impressions for input music  $m_x$  and  $m_y$ , respectively, and  $p(\mathbf{h}|m_z)$  is that for virtual music  $m_z$  assumed to be the result of the mixture.

The operation to adopt the concurrent impression enhances the outstanding probability in each distribution. This method to combine two music pieces is suitable for a situation where users want music that has the remarkable features of both pieces.

#### 3.3.2 Product Mixture

The second approach to mix two music pieces effectively is to accentuate the features common to both music inputs. This is achieved by the formula

$$p(\mathbf{h}|m_z) = \frac{p(\mathbf{h}|m_x)p(\mathbf{h}|m_y)}{\sum_{\mathbf{h}} p(\mathbf{h}|m_x)p(\mathbf{h}|m_y)}. \quad (7)$$

This operation corresponds to the following process.

1. Repeat extractions of the impression from each music piece until the extracted impressions converge.
2. Adopt the concurrent impression as the result of the mixture for the two music pieces.

This method is suitable for a situation where users want music that has a remarkable feature common to input music  $m_x$  and  $m_y$ . It can be applied for recommending music by using multiple music pieces listened to by users as a query.

### 3.4 Ranking by Similarity between Music Pieces

The virtual music resulting from the combination of two music inputs is characterized by a distribution  $p(\mathbf{h}|m_z)$ , and the music in a system is ranked by closest similarity and returned as a search result. Here, defining the similarity between two music pieces is necessary.

Generally, the *Kullback-Leibler divergence*  $D_{KL}(p||q)$  is used for the similarity of probabilistic distributions  $p$  and  $q$ . This function is not symmetric, thus we take the average of the two versions and define the similarity between two music pieces  $m_x$  and  $m_y$ , letting  $p = p(\mathbf{h}|m_x)$  and  $q = p(\mathbf{h}|m_y)$ :

$$\text{Sim}(m_x, m_y) = \exp \left[ -\frac{1}{2} \{D_{KL}(p||q) + D_{KL}(q||p)\} \right]. \quad (8)$$

Given the distribution  $p(\mathbf{h}|m_z)$  for a virtual music piece, each music piece  $m \in M$  in a system is returned on the basis of the similarity  $\text{Sim}(m_z, m)$ .

<sup>1</sup> <http://code.google.com/intl/en/apis/chart/>

Number of impression neighbors	Average percentage of songs in same genre
1	0.579
5	0.523
10	0.488
20	0.454
50	0.407
100	0.359
All	0.152

**Table 1.** Average percentage of most similar songs in same genre

#### 4. IMPLEMENTATION

We collected music pieces and reviews from Amazon.com with Amazon Web Services<sup>2</sup>, querying by artist names that are listed in CAL500 [16]. We obtained 86,050 pieces, for which 879,666 reviews were written; the average number of reviews per artist was about 10.2. The obtained reviews were analyzed by GibbsLDA++<sup>3</sup>, which is an implementation of Gibbs sampling for LDA. As parameters in LDA, we fixed the number of topics  $K = 100$  and hyper-parameters  $\alpha = 50/K$  and  $\beta = 0.1$ . We then conducted 1000 iterations of Gibbs sampling for the parameter estimation.

#### 5. EVALUATION

##### 5.1 Evaluation of Characterization

Before evaluating our system, the performance of characterization by impressions must be clarified. We evaluated our method in accordance with the objective evaluation by Aucouturier et al. [1]. We calculated the correlation between impression and genre similarity by using the songs in our system. Because Amazon.com has multiple labels on songs, only 356 songs that had only 1 label and more than 20 reviews were used in our evaluation, and the top 11 genres used in our experiment were R&B, country, rap and hip-hop, classic rock, classical, jazz, blues, pop, alternative rock, world music, and soundtracks.

The results can be seen in Table 1 and Fig. 5. In Table 1, the closest  $k$  songs for a song were retrieved, the percentage of the same genre was calculated, and the average was taken for all the songs. There was a low correlation between impression and genres. As indicated in the study on timbre and genre [1], this approach cannot measure the performance correctly because two songs in the same genre do not always give similar impressions. However, comparing the results with those of timbre similarity, we could show the effectiveness of review-centered characterization.

A similarity matrix for each genre is shown in Fig. 5. Each cell represents the average of the similarity between songs in two genres. We could see a difference between songs in the same genre and different genres except in the alternative rock genre.

Rap&Hip-Hop	0.879	0.824	0.841	0.838	0.836	0.531	0.678	0.776	0.811	0.614	0.827
Country	0.824	0.871	0.821	0.806	0.827	0.505	0.719	0.791	0.827	0.555	0.837
Classic Rock	0.841	0.821	0.859	0.837	0.821	0.498	0.687	0.781	0.797	0.699	0.829
World Music	0.838	0.806	0.837	0.852	0.836	0.526	0.679	0.759	0.781	0.691	0.817
R&B	0.836	0.827	0.821	0.836	0.850	0.506	0.691	0.765	0.791	0.619	0.828
Jazz	0.531	0.505	0.498	0.526	0.506	0.766	0.579	0.479	0.541	0.343	0.502
Classical	0.678	0.719	0.687	0.679	0.691	0.579	0.735	0.682	0.734	0.470	0.712
Alternative Rock	0.776	0.791	0.781	0.759	0.765	0.479	0.682	0.750	0.770	0.568	0.788
Soundtracks	0.811	0.827	0.797	0.781	0.791	0.541	0.734	0.770	0.833	0.539	0.806
Blues	0.614	0.555	0.699	0.691	0.619	0.343	0.470	0.568	0.539	0.878	0.613
Pop	0.827	0.837	0.829	0.817	0.828	0.502	0.712	0.788	0.806	0.613	0.841

**Figure 5.** Similarity matrix for 11 genres. Each cell represents the average of similarity between songs in two genres. The black cells represent the maximum similarity in each row, and the gray cells represent the 2nd and 3rd maximum similarity within 10% of the maximum in each row.

##### 5.2 Evaluation of Query-by-Mixture-of-Example

Comparing with results returned by a human, we investigated the performance of our proposed method to search with a query by mixture of example. We asked a student who knows music pieces well to choose reasonable songs as mixture of input queries listed in Table 2. Then, we asked 5 persons to listen input music and output music including both music recommended by human and returned by *RhythMiXearch*, and to evaluate relevance of the outputs in five levels. The result is shown in Fig. 6, where the average scores were taken for each question, and the question numbers correspond to those in Table 2.

In some questions, music recommended by human were considered more relevant than the results returned by *RhythMiXearch*. Our system is inferior to human in performance, however, the result by human should be regarded as the upper bound in the evaluation. In the questions 4 and 5, the results by *RhythMiXearch* obtained higher scores, whereas in the questions 2 and 3, our method failed to return relevant results for mixture of example. The result may show that a human can recommend music only for similar two music like the inputs seen in the question 2 and 3, on the one hand, our system can search for music even for different types of music like the inputs used in the question 4 and 5.

#### 6. CONCLUSION

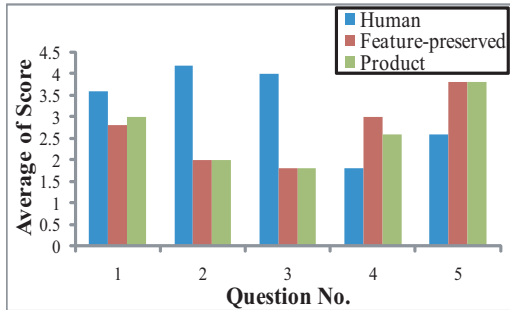
We presented a novel method for searching for unknown music and also presented our developed system *RhythMiXearch*, which can accept two music inputs and mix those inputs to search for music that could reasonably be a result

<sup>2</sup> <http://aws.amazon.com/>

<sup>3</sup> <http://gibbslda.sourceforge.net/>

#	Input A	Input B	Human	Feature-Preserved Mixture	Product Mixture
1	The Beatles, Let It Be	Coldplay, Viva La Vida	Bob Dylan, Blowin' In the Wind	Kiss, Dynasty	The Black Crowes, Lions
2	Michael Jackson, Thriller	Madonna, Like a Virgin	Jamiroquai, Cosmic Girl	Jimi Hendrix, The Jimi Hendrix Experience	*
3	Eminem, The Eminem Show	Britney Spears, Britney	TLC, Silly Ho	Green Day, Nimrod	*
4	Eric Clapton, 461 Ocean Boulevard	John Lennon, Imagine	Eagles, New Kid in Town	Eric Clapton, Me and Mr. Johnson	Cream, Disraeli Gears
5	The Cardigans, First Band on the Moon	Whitney Houston, Whitney	Janis Joplin, Half Moon	Christina Aguilera, Stripped	*

**Table 2.** 5 set of inputs and outputs for evaluation of Query-by-Mixture-of-Example. (\* means the same result as Feature-preserved Mixture.)



**Figure 6.** Average of scores for each question

of the mixture. Our first contribution was to characterize music pieces by reviews with LDA and to evaluate the performance of the representation of the music pieces. The second contribution was to propose a probabilistic mixture model for processing multiple example queries. We believe that *Query-by-Mixture-of-Examples* is an important concept for searching for new music pieces.

## 7. ACKNOWLEDGEMENTS

This research was supported by Exploratory IT Human Resources Project (MITOH Program Youth 2008). I express my sincere gratitude to Professor Michiaki Yasumura, Keio University, for comments on my work. Thanks to GOGA, Inc. for supporting the project. Thanks to members of Tanaka laboratory, especially Yusuke Yamamoto, for helping the user experiments and discussion on my work.

## 8. REFERENCES

- [1] J.J. Aucouturier and F. Pachet. Music similarity measures: What's the use. In *Proc. of the 3rd ISMIR*, pages 157–163, 2002.
- [2] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.
- [3] R. Cai, C. Zhang, C. Wang, L. Zhang, and W.Y. Ma. MusicSense: contextual music recommendation using emotional allocation modeling. In *Proc. of the 15th Multimedia*, pages 553–556, 2007.
- [4] R.B. Dannenberg and N. Hu. Understanding search performance in query-by-humming systems. In *Proc. of the 5th ISMIR*, pages 232–237, 2004.
- [5] J. Foote, M. Cooper, and U. Nam. Audio retrieval by rhythmic similarity. In *Proc. of the 3rd ISMIR*, pages 265–266, 2002.
- [6] M. Goto and K. Hirata. Recent studies on music information processing. *Acoustical Science and Technology*, 25(6):419–425, 2004.
- [7] T.L. Griffiths and M. Steyvers. Finding scientific topics. *Proc. of the National Academy of Sciences*, 101(90001):5228–5235, 2004.
- [8] K. Hevner. Experimental studies of the elements of expression in music. *The American Journal of Psychology*, pages 246–268, 1936.
- [9] P. Knees, T. Pohle, M. Schedl, and G. Widmer. A music search engine built upon audio-based and web-based similarity measures. In *Proc. of the 30th SIGIR*, pages 447–454, 2007.
- [10] Peter Knees, Elias Pampalk, and Gerhard Widmer. Artist classification with web-based data. In *Proc. of the 5th ISMIR*, pages 517–524, 2004.
- [11] Peter Knees, Tim Pohle, Markus Schedl, Dominik Schnitzer, and Klaus Seyerlehner. A document-centered approach to a natural language music search engine. In *Proc. of the 30th ECIR*, pages 627–631, 2008.
- [12] J. Paulus and A. Klapuri. Measuring the similarity of rhythmic patterns. In *Proc. of the 3rd ISMIR*, pages 150–156, 2002.
- [13] T. Pohle, P. Knees, M. Schedl, and G. Widmer. Meaningfully Browsing Music Services. In *Proc. of the 8th ISMIR*, pages 23–30, 2007.
- [14] M. Tolos, R. Tato, and T. Kemp. Mood-based navigation through large collections of musical data. In *Proc. of the IEEE 2nd CCNC*, pages 71–75, 2005.
- [15] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas. Multilabel classification of music into emotions. In *Proc. of the 9th ISMIR*, pages 325–330, 2008.
- [16] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Towards musical query-by-semantic-description using the CAL500 data set. In *Proc. of the 30th SIGIR*, pages 439–446, 2007.
- [17] Rainer Typke, Frans Wiering, and Remco C. Veltkamp. A survey of music information retrieval systems. In *Proc. of the 6th ISMIR*, pages 153–160, 2005.
- [18] G. Tzanetakis, A. Ermolinskyi, and P. Cook. Beyond the query-by-example paradigm: New query interfaces for music information retrieval. In *Proc. of the 2002 ICMC*, pages 177–183, 2002.