Computational Model of Jam Session From Statistical Analysis of Music Rendering Features

Takeshi Hori **Kazuyuki Nakamura** **Shigeki Sagayama**
Graduate School of Advanced Mathematical Sciences
Meiji University
{cs51003, knaka, sagayama}@meiji.ac.jp

Abstract

In this papers, we discuss a computational model of a jazz session toward realizing a human-computer-collaborated automatic jazz session system that is statistically trainable using jazz session data. In contrast to previous studies that required human-labeled data of human sensation to analyze the intentions of players, our model is solely based on statistics by assuming that the training data of jazz sessions consist of good combination examples of playing styles by multiple musical instruments to exclude heuristics. For this purpose, the musical performance of an instrument is regarded as a vector trajectory in the feature space along time and is approximated by stochastic state transitions with co-occurrence among other instruments for trainability using sparse data. Therefore, the session model consists of three elements: a stochastic state transition, state co-occurrence between instruments, and correlation between musical performances. A hidden Markov model (HMM) can effectively represent such a session model. This paper focuses on clustering methods for reducing the dimensionality of the feature vector by comparing three methods: *k*-means, the gaussian mixture model (GMM), and non-negative matrix factorization (NMF). The experimental results show that NMF-based clustering yielded the highest prediction accuracies using both a trigram and the HMM.

Introduction

We previously developed an automatic accompaniment system (called Eurydice (Nakamura et al. 2013)) that allows tempo changes and note insertion/deviation/substitution errors in human performance as well as repeats and skips, while other automatic accompaniment systems (Dannenberg 1984) cannot handle such long jumps. As the next step, we are working on an automated jam session system that can follow improvised human performances.

Jam sessions consist of frequently improvised part like jazz, where the players improvise on a score, listen to the performance of others, anticipate their intentions from their previous performances, score information, and interplay. This differentiates the system from a simple extension of an automatic accompaniment system, since a jam session system yields various performances from a human performance and is expected to have a function of composition.

Therefore, jam session systems must estimate the other players' intentions and output a matching performance. Most previous jam session systems observed the human performance, extracted musical features, predicted the next performance using heuristically determined parameters, and generated a matching performance.

In contrast to non-proactive systems (Rowe 1992; Aono, Katayose, and Inokuchi 1994; 1995; Nishijima and Watanabe 1992), where parameters were determined prior to the session, JASPER (Wake et al. 1994) and VirJa Session (Goto et al. 1996) dealt with a piano trio (piano, bass, drums) proactively and interactively in jazz sessions. This system could generate an expressive performance by instantaneously reacting other players' performances by using variable parameters, although this system could not apply a long-range plan of the performance in an actual human performance because these parameters and rules were set statically. Guitarist Simulator (Hamanaka et al. 2004) could learn the reaction model of an actual player, but the correlation between the actual performance and the reaction model was determined through psychological experiments involving a particular participant.

This paper takes a statistical approach to exclude heuristic rules and the human labeling of training data as much as possible.

Session model

Our goal is to build a statistically trainable mathematical model without using subjective rules. In other words, we aim to make it possible to identify the current state of a performance from actual human performance data and scores. For this purpose, we exclude as many heuristic/subjective theories and approaches through psychological quantities as possible, hoping that our approach will lead to a mathematical model of mutual cooperative sessions, not limited to within music, that will be trainable using session data without requiring human labeling.

Fig.1 shows a conceptual block diagram of our ultimate system. In this paper, we chose a jazz piano trio as a configuration example. Input and output data are supplied as MIDI data for ease of obtaining and dealing with the performance

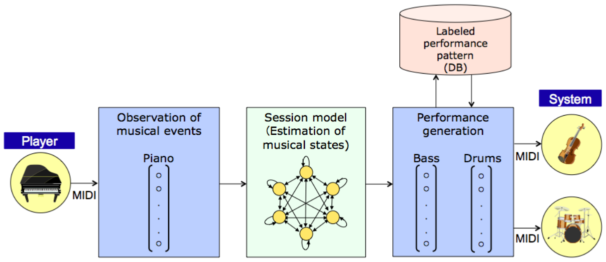
Labeled performance pattern (DB)
Player
MIDI
Observation of musical events
Piano
Session model (Estimation of musical states)
Performance generation
Bass
Drums
System
MIDI
MIDI

Figure 1: Conceptual block diagram

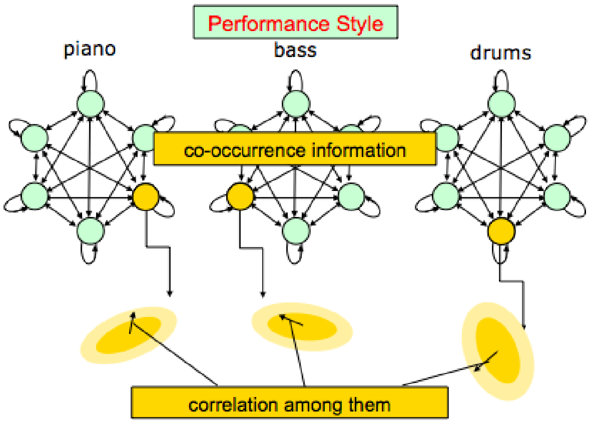

Figure 2: Mathematical model in the case of the piano trio

data.

To represent a musical performance mathematically, we start with a high-dimensional space that includes all possible musical events. Given the ideal space of musical events Ω enclosing all events related to the music, this space is composed of, for instance, information about notes (such as the number of notes, note values, and volumes), gestures, eye contact, and so forth. The music at a given time can be described by a scatter point in the space. Thus, a musical performance is expressed by a trajectory T. Hence, the musical performance is expressed as (Ω, T) in the space of musical events.

In a piano trio, the performances of the participating musical instruments are given as trajectories in this space under certain constraints, in accordance with the conventions of jazz sessions. Firstly, in the case of a piano, the performance is ruled by a time constraint based on the previous performance for musical naturalness. Therefore, the stochastic deviation of the trajectory between times t and $t+1$ is restricted within a certain range. Secondly, since the pianist interacts with other players, the trajectory is also influenced by them. The bassist and drummer are also influenced by the pianist. In other words, each trajectory has co-occurrence and a correlation with the other trajectories. On the other hand, most of the previously proposed session systems attempted to design a model of the instantaneous interaction between a human and a computer, as opposed to our model of participating instruments, each following a musical flow and interacting with other instruments.

There are two major practical problems in dealing with a statistically trainable model for these trajectories: continuity and high dimensionality. In an ideal case where an unlimited amount of data are available, we would be able to statistically estimate the trajectories, co-occurrence, and correlations. However, we have a limited amount of data. Thus, we must reduce the dimensionality of the space of musical events and discretize the space and trajectories.

To reduce the dimensionality of Ω, we extract feature parameters to store information relevant to the jam session. We defined such features as style parameters. To discretize the trajectories, we split the space into subspaces by clustering of the style parameter vectors. If we can assume that the style parameter vectors carry sufficient information on the session states and that the discretized space maintains a sufficient space resolution, it will be possible to characterize trajectories with statistical parameters to automatize the session process.

On this basis, we formulate the model as a hidden Markov model (HMM). In the HMM, the performance of each musical instrument is represented by a time sequence of style vectors and approximated by centroids along multiple hidden-state transitions. The interplay in the session is represented by the co-occurrence between hidden states in distinct HMMs. Moreover, variation beyond the space resolution can be added by including a deviation from the centroids based on the statistical correlation between musical instruments. Fig.2 depicts this mathematical model in the case of a piano trio session.

Therefore, the outline of the statistically trainable session model in Fig.1 consists of three stages: stochastic state transitions, co-occurrence between instruments, and correlation between performances. First, each style parameter vector is extracted from the human performance over time. Second, the time series of the feature vector forms a trajectory and is formulated as stochastic state transitions in an HMM to identify the current state using the Viterbi algorithm. Third, a performance data having the style parameters of the counterparts are retrieved from a database or are automatically generated using the style parameters, and the performances is output.

Style parameters

We wanted to set an effective axis in the session model. We selected parameters closely related to the jazz session on the basis of musical knowledge and call them style parameters. In our present research, we assumed MIDI-format data as the observation input from multiple jam sessions with a constant tempo to enable us to extract feature parameters at every unit time (every bar or every beat in later sections). We defined 68 parameters that are extractable from the music performance at every unit time as follows:

- *Piano-specific features*
  - The number of notes composed of diatonic chords, and the character of notes such as tension notes, avoid

notes, and blue notes.

- The range between the highest and lowest tones.

- *Bass-specific features*
  - The range between the highest and lowest tones.
- *Drums-specific features*
  - Each number of notes of the hi-hat cymbal, snare drum, and crash cymbal.
- *Common features*
  - The number of notes, the number of simultaneous sounds, the average velocities.
  - The ratio of the above features between adjacent time spans.
  - The ratio of sum of the above features throughout the music.
  - The ratio of off-beat notes to all notes in the unit time.

Clustering

To discretize the space with the reduced dimension, we applied a clustering algorithm to describe the trajectory and to train the HMM from musical performance data. Since there are various clustering algorithms, we compared three methods for clustering: (1) k-means clustering, (2) Gaussian mixture model (GMM), and (3) non-negative matrix factorization (NMF). In the training phase, k-means clustering is based on a hard decision while the GMM and NMF are based on a soft decision and are considered to be effective in the case of a limited amount of training data.

k-means clustering

K-means clustering classifies each data point into the cluster whose centroid is closest to the data point. The data are a sequence of vectors of style parameters extracted from the MIDI data. Prior to the k-means process, to reduce the dimensionality of the observation vector, principal component analysis (PCA) was applied with a threshold cumulative contribution ratio of 90%.

Gaussian mixture model (GMM)

The GMM is a model for approximating a distribution by a mixture of normal distributions. The expectation-maximization algorithm (EM algorithm) is generally used for training the model. This algorithm is an iterative method alternately applying an E-step to calculate the expectation of the likelihood and an M-step to update the parameters by maximizing the likelihood. Given m vectors $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m)$, we assign each of them to an appropriate cluster out of k clusters $C = (c_1, c_2, \cdots, c_k)$. θ is a mixture parameter consisting of Gaussian means, variances, and mixture coefficients. Equivalently, instead of maximizing the likelihood function, we maximize the Q-function, which is expressed as follows.

$$Q(\theta,\theta') = \sum_{m=1}^{M}\sum_{k=1}^{K} P(c_k|x_m;\theta') \log P(x_m|c_k;\theta)P(c_k)$$

θ' is initialized by a random value. The posterior probability $P(x_m|c_k;\theta_k)$ in the GMM is assumed to be a normal distribution. From the partial derivative of the Q-function, we derive a set of iterative formulae.

$$\begin{aligned} \mu_k &= \frac{\sum_{m=1}^{M} P(C_k|x_m;\theta')x_m}{\sum_{m=1}^{M} P(C_k|x_m;\theta')}, \\ \sigma_k^2 &= \frac{\sum_{m=1}^{M} P(C_k|x_m;\theta')(x_m-\mu_k)^2}{\sum_{m=1}^{M} P(C_k|x_m;\theta')}, \\ P(c_k) &= \frac{\sum_{m=1}^{M} P(C_k|x_m;\theta')}{M}. \end{aligned}$$

As the log likelihood increases monotonically, θ' is updated until the increment becomes smaller than a preset value used as a convergence criterion.

After training, each vector is assigned to the class with the highest probability. We applied PCA to reduce the dimensionality in the same way as in k-means clustering.

Non-negative matrix factorization (NMF)

NMF is a method for factorizing a non-negative matrix into a pair of non-negative matrices with a lower rank (Lee and Seung 2000) and has also been used as a method for clustering (Kim and Park 2008). Given a non-negative original matrix X, it is approximated by a product of non-negative matrices

$$X \approx HU, \tag{1}$$

where H is the basis matrix and U is the activation matrix. To derive an iterative formulae from (1), we obtain

$$X_{i,j} \approx \sum_k H_{i,k}U_{k,j},$$

where k denotes the index of the basis vector. H and U are initialized with random values. To define the criterion for approximation, a distance measure between X and HU is usually selected from the Euclidean distance, Kullback-Leibler divergence, or Itakura-Saito divergence. In the case of Kullback-Leibler divergence,

$$d_{KL}(x_{ij}, h_i, u_j) = \sum_{i=1}^{I}\sum_{j=1}^{J} x_{ij}\log\frac{x_{ij}}{h_i^T u_j} - x_{ij} + h_i^T u_j.$$

The selected divergence is minimized to find the optimal H and U using an iterative algorithm derived from the above formulae using auxiliary functions. The resulting update rules differ according to the divergence. Using the method of the Lagrange multiplier, we obtain

$$h_{ik} \leftarrow h_{ik}\frac{\sum_j \frac{x_{ij}}{\hat{x}_{ij}}u_{kj}}{\sum_j u_{kj}}, \quad u_{kj} \leftarrow u_{kj}\frac{\sum_i \frac{x_{ij}}{\hat{x}_{ij}}h_{ik}}{\sum_i u_{ik}}.$$

After convergence, each input vector is assigned the class of the basis vector with the largest activation, i.e.,

$$c_k(x_j) = \arg\max_k u_{kj}. \tag{2}$$

To scale the basis vectors, we normalize them as follows:

$$\sum_i h_{ik} = 1.$$

Experimental evaluation

Computing and evaluation methods

Both an N-gram and an HMM can represent the trajectory of a musical performance expressed by stochastic state transitions in the discretized space. To select the most suitable method for clustering, we evaluated the methods (*k*-means clustering, GMM, NMF) from the prediction results of both an N-gram and an HMM when analyzing 13 MIDI data from Yamaha Music Datashop with different time units, bar units and beat units. The number of clusters was varied from 6 to 29.

The accuracy was evaluated through cross validation. In the case of *k*-means clustering, the means of the sample data were used to assign cluster numbers to the test data. Similarly, in the GMM, the average and covariance of the sample data were used. In NMF, the activation matrix U was given by using the original matrix X and the generalized inverse matrices of the basis matrix $H^+$of the sample data.

$$\begin{aligned} U_{sample} &= H^+_{sample} X_{sample} \\ U_{test} &= H^+_{sample} X_{test} \end{aligned}$$

Then cluster numbers were assigned by (2).

Trigram

In an N-gram, given n states $P(s_1, s_2, \cdots, s_n)$, the chain probability is given as follows:

$$P(s_1 s_2 \cdots s_n) = \prod_{i=1}^{n} P(s_i | s_{i-N+1} \cdots s_{i-1}).$$

For a trigram, the number of transitions from $i-2$ to i is expressed by $N(s^i_{i-2})$ and the chain probability is given as

$$P(\omega_i | \omega^{i-1}_{i-2}) = \frac{N(\omega^i_{i-2})}{N(\omega^{i-1}_{i-2})}.$$

Hidden Markov model

The HMM is a tool for representing the probability over sequences of observation. We regard the hidden states as the performance styles and the cluster numbers observed from these states as the musical performances. The state transition probability a_{ij} (the transition probability between i and j) and the observation symbol probability $b_i(o(t))$ (the observation symbol probability of state i at time t) are computed by the Baum–Welch algorithm. From a forward variable $\alpha_t(i)$ describing the probability of state i at time t, a backward variable $\beta_t(i)$ as the probability of state i at time t, and the likelihood $Pr[O|\lambda]$, the E-step is computed as follows:

$$\begin{aligned} \xi_t(i,j) &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{Pr[O|\lambda]}, \\ \gamma_t(i) &= \frac{\alpha_t(i)\beta_t(i)}{Pr[O|\lambda]}. \end{aligned}$$

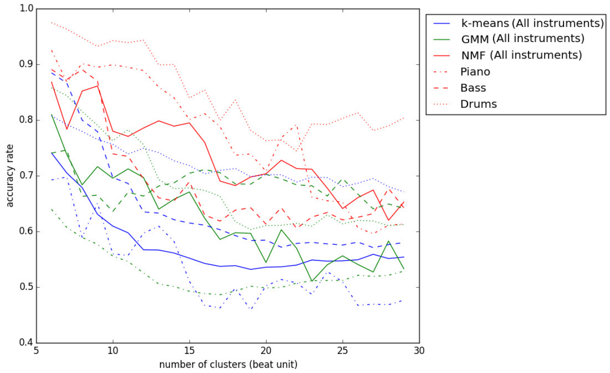

Figure 3: Prediction accuracy for beat-unit time resolution (trigram)

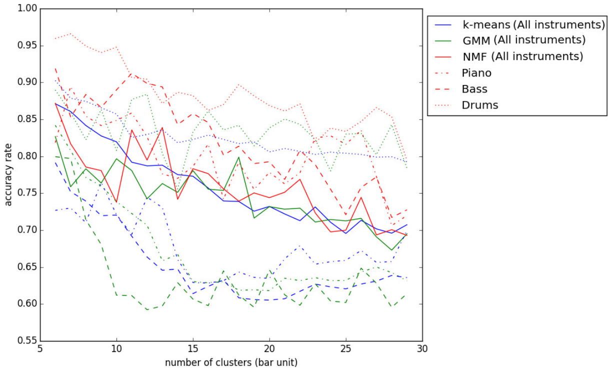

Figure 4: Prediction accuracy for bar-unit time resolution (trigram)

From these formulae, a_{ij} and $b_i(k)$ are updated in the M-step as follows:

$$\begin{aligned} a_{ij} &= \frac{\sum_{t=1}^{T-1} \xi_n(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \\ b_i(k) &= \frac{\sum_{t=1, s.t. o(t)=k}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}. \end{aligned}$$

The prediction is given by the forward algorithm.

Prediction results

The experimental results of the prediction using the trigram are shown in Fig.3-4 and the HMM are shown in Fig.5-6. The time resolution of Fig.3 and Fig.5 is a beat unit and that of Fig.4 and Fig.6 is a bar unit. X axis illustrates the number of clusters and the Y axis illustrates the accuracy rates. The blue line shows the accuracy rates for *k*-means clustering, the green line shows that for the GMM, and the red line shows that for NMF. Different types represent different instruments. For example, the solid line shows the accuracy rate when using all the style parameters and the chain line shows that when only using the style parameters related to the piano. These figures show that NMF achieved the highest

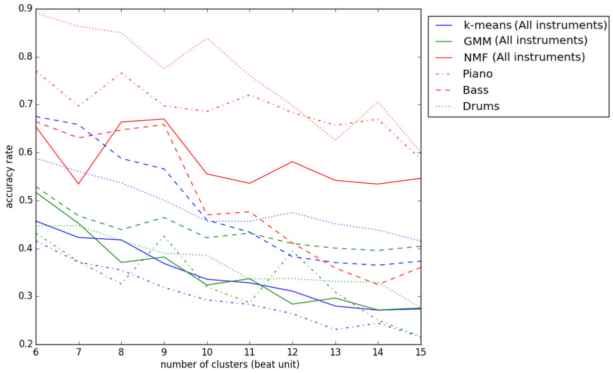

Figure 5: Prediction accuracy for beat-unit time resolution (HMM)

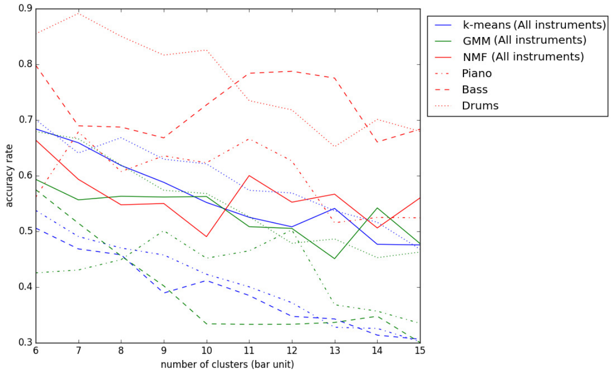

Figure 6: Prediction accuracy for bar-unit time resolution (HMM)

prediction accuracy, in particular, a higher ratio was shown for the beat-unit time resolution. With increasing sparsity of the original matrix, the prediction accuracy tends to rise.

To consider the reason for these results, the vector data of a face photo is approximated by NMF and the basis matrix accurately expresses parts of a face such as the eyes, nose, and mouth, and the face is expressed by this additive combination. Similarly, the style parameters are neither independent nor completely dependent. Fig.7 shows heat maps of basis matrix and Fig.8 shows those of activation matrix. Both the horizontal axis of the basis matrix and the vertical axis of the activation matrix represent the class number. The vertical axis of the basis matrix represents the style parameter and the horizontal axis of the activation matrix shows the performance time. We can see from Fig.7 and Fig.8 that the basis and activation matrices are sparse. Each column of the basis matrix represents a characteristic of the music. Therefore, the additivity of NMF and the character of style parameters might yield high prediction accuracy similarly the case of a face photo.

In addition, the GMM yielded slightly higher prediction accuracy than *k*-means clustering. It is speculated that soft clustering is more effective than hard clustering in this problem. On the other hand, although both the GMM and NMF are soft clustering methods, higher prediction accuracy was obtained by NMF. Further work is needed to clarify the reason for this.

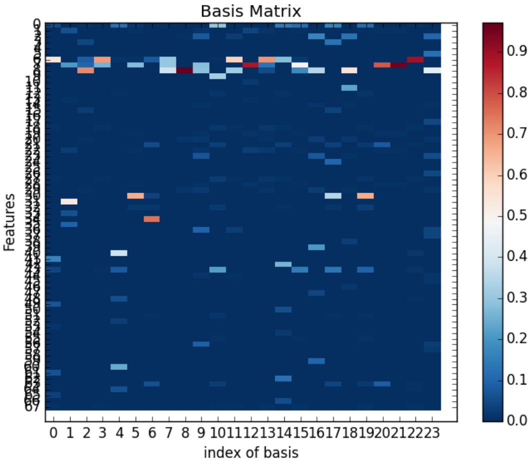

Figure 7: Basis matrix for beat units (number of clusters: 25)

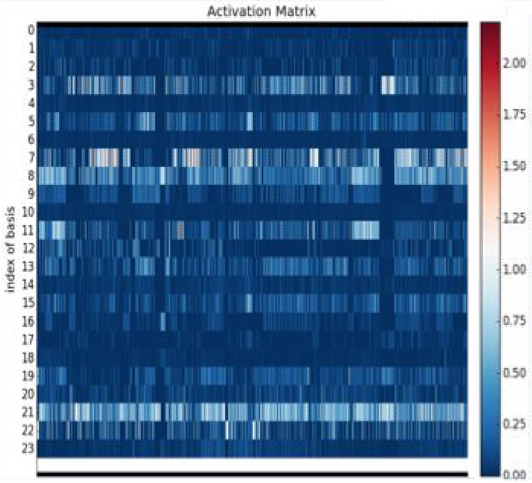

Figure 8: Activation matrix for beat units (number of clusters: 25)

Co-occurrence

Although the prediction accuracies were computed using four pattern (the style parameters of all instruments, only piano, only bass, and only drums), our model includes the co-occurrence between instruments, making it necessary to determine whether there is co-occurrence and whether it is possible to obtain a higher prediction accuracy by including co-occurrence. Table 1-2 show the conditional probabilities $P(Bass_k|Piano_k)$ and $P(Drums_k|Piano_k)$ calculated by using the cluster number in NMF, where the number of cluster is six and k it the cluster number. Table 3-4 show the conditional probabilities calculated by using the hidden states in the HMM, where the number of hidden states is six. k is the state number. These states were assigned by the Viterbi

Table 1: Conditional probabilities (piano-bass / bar-unit time resolution, number of clusters: 6)

	cluster number (bass/bar)					
State number (piano/bar)	0	1	2	3	4	5
0	0	0	0	0	0	1
1	0.023	0.048	0.014	0.114	0.786	0.016
2	0.062	0	0.021	0.229	0.292	0.396
3	0.007	0.013	0.009	0.092	0.866	0.012
4	0.031	0.041	0.004	0.156	0.757	0.011
5	0.102	0.02	0	0.061	0.102	0.714

Table 2: Conditional probabilities (piano-drums / bar-unit time resolution, number of clusters: 6)

	cluster number (drums/bar)					
State number (piano/bar)	0	1	2	3	4	5
0	0	0	1	0	0	0
1	0.134	0.027	0.027	0.002	0.775	0.034
2	0.062	0.375	0.208	0	0.312	0.042
3	0.085	0.066	0.015	0.001	0.762	0.072
4	0.155	0.099	0.025	0.004	0.606	0.112
5	0	0.184	0.02	0	0.224	0.571

Table 3: Conditional probabilities (piano-bass / bar-unit time resolution, six-state HMM)

	Hidden state number (bass/bar)					
State number (piano/bar)	0	1	2	3	4	5
0	0.02	0.367	0	0.143	0.041	0.429
1	0.143	0.014	0	0.146	0	0.708
2	0.021	0.125	0	0.146	0	0.708
3	0.183	0.033	0.05	0.033	0.655	0.05
4	0.146	0.166	0.099	0.13	0.439	0.019
5	0.127	0.137	0.095	0.064	0.565	0.012

Table 4: Conditional probabilities (piano-drums / bar-unit time resolution, six-state HMM)

	Hidden state number (drums/bar)					
State number (piano/bar)	0	1	2	3	4	5
0	0	0.041	0.204	0.245	0	0.51
1	0.033	0.048	0.054	0.057	0.73	0.078
2	0.042	0.604	0	0	0.104	0.25
3	0.067	0.117	0.05	0.033	0.617	0.117
4	0.011	0.157	0.122	0.127	0.296	0.287
5	0.005	0.049	0.073	0.076	0.699	0.098

algorithm. The time resolution is a bar unit. Table 1 and table 3 show that the performance of the bass is effected by the performance of the piano. The conditional probabilities of $P(Drums_k|Piano_k)$ is also stochastic deviation. In other wards, there are stochastic deviations in the performances of all instruments. Hence, it is expected that we can predict the musical performance with more precision by including co-occurrence in the mathematical model of a jazz session.

Conclusions

We first described a statistically trainable session model that is based on the two phases to represent the trajectory of musical performances and three elements to predict them. To represent the trajectory as trainable model from actual musical data, we discretized the trajectory and expressed the space nonlinearly and continuously. Additionally, we selected style parameters that are closely related to the jazz session on the basis of musical knowledge. The stochastic state transitions expressing the discretized trajectory of musical performances were represented by an HMM with co-occurrence between the hidden states of all instruments to model their interaction with the correlation given by the deviation from the centroid of the hidden states.

Secondly, to select the clustering method for quantizing the event space and discretizing the trajectory, we evaluated three clustering methods. (k-means, GMM, and NMF) by computing the prediction accuracy using both a trigram and the HMM. As a result, NMF yielded the highest prediction accuracy for bar, beat time solutions for both the trigram and the HMM. To express and to presume the musical performance effectively, we consider that soft clustering and a sparse basis matrix and activation matrix are effective.

In addition, the conditional probabilities between piano and other instruments show that there are stochastic deviations in the performances of all instruments. Therefore, we can utilize these results to predict the musical performance more precisely.

Finally, we claim that this model has flexibility. For instance, group work with a robot, the reactions of human and many animals, and the dynamics of natural phenomena might be expressed by our model. Moreover, because the model is trainable only using actual observable data and does not include heuristic rules, it can continue to be develop if we could obtain a large amount of learning data.

We plan to build a prototype Jazz editing system that uses NMF and HMM. The system can generate MIDI data of a piano trio from that of piano to edit a bass and drums parts in case data by batch processing on the off-line. We also plan to rebuild the mathematical model as the dynamic Bayesian network (DBN) including the HMM to compute the co-occurrence between all instruments. Additionally, if the co-occurrence between all instruments is given a certain degree of freedom, it might be possible to express the personalities of players because the computer players could decide their musical performances proactively without depending on the performances of the other players. Furthermore, we will use a deep neural network (DNN) for interpolation to deal with the trajectories expressing music continuously rather than discretely.

References

Aono, Y.; Katayose, H.; and Inokuchi, S. 1994. Development of Band-like Musical Assistant System. *IPSJ SIG Technical Reports, 94-MUS-8* 94(103):45–50.

Aono, Y.; Katayose, H.; and Inokuchi, S. 1995. An Improvisational Accompaniment System Obseving Performers Musical Gesture. *Proc. ICMC* 1995:106–107.

Dannenberg, R. B. 1984. An On-line Algorithm for Realtime Accompaniment. *Proc. ICMC* 1984:193–198.

Goto, M.; Hidaka, I.; Matsumoto, H.; Kuroda, Y.; and Muraoka, Y. 1996. A Jazz Session System for Interplay among All Players - VirJa Session (Virtual Jazz Session System). *Proc. ICMC* 1996:346–349.

Hamanaka, M.; Goto, M.; Asoh, H.; and Otsu, N. 2004. Guitarist Simulator: A Jam Session System Statistically Learning Players Reactions. *IPSJ Journal* 45(3):698–709.

Kim, J., and Park, H. 2008. Sparse nonnegative matrix factorization for clustering. *Technical Report GT-CSE-08-01, Georgia Institute of Technology* 2008.

Lee, D. D., and Seung, H. S. 2000. Advances in Neural Information Processing Systems. 13:556–562.

Nakamura, E.; Takeda, R.; Yamamoto, R.; Saito, Y.; Sako, S.; and Sagayama, S. 2013. Score Following Handling Performances with Arbitrary Repeats and Skips and Automatic Accompaniment. *IPSJ Journal* 54(4):1338–1349.

Nishijima, N., and Watanabe, K. 1992. Interactive Music Composer on Neural Networks. *Proc. ICMC* 1992:53–56.

Rowe, R. 1992. Machine Listening and Composing with Cypher. *Computer Music Journal* 16(1):43–63.

Wake, S.; Kato, H.; Saiwaki, N.; and Inokuchi, S. 1994. Cooperative Musical Partner System Using Tension-Parameter: JASPER (Jam Session Partner). *Trans. IPS Japan* 35(7):1469–1481.