Improvised Duet Interaction: Learning Improvisation Techniques for Automatic Accompaniment

Gus G. Xia
Dartmouth College
Neukom Institute
Hanover, NH, USA
gxia@dartmouth.edu

Roger B. Dannenberg
Carnegie Mellon University
School of Computer Science
Pittsburgh, PA, USA
rbd@cs.cmu.edu

ABSTRACT

The interaction between music improvisers is studied in the context of piano duets, where one improviser performs a melody expressively with embellishment, and the other plays an accompaniment with great freedom. We created an automated accompaniment player that *learns* to play from example performances. Accompaniments are constructed by selecting and concatenating one-measure score units from actual performances. An important innovation is the ability to learn how the improvised accompaniment should *respond* to the musical expression in the melody performance, using timing and embellishment complexity as features, resulting in a truly interactive performance within a conventional musical framework. We conducted both objective and subjective evaluations, showing that the learned improviser performs more interactive, musical, and human-like accompaniment compared with the less responsive, rule-based baseline algorithm.

Author Keywords

Interactive, Automatic Accompaniment, Duet, Improvisation.

ACM Classification

H.5.1 [Information Interfaces and Presentation] Multimedia Information Systems–Artificial, augmented, and virtual realities, H.5.5 [Information Interfaces and Presentation] Sound and Music Computing. I.2.6 [Artificial Intelligence] Learning

1. INTRODUCTION

Automatic accompaniment systems have been developed for decades to serve as *virtual musicians* capable of performing music interactively with human performers. The first systems invented in 1984 [5][16] used simple models to follow a musician's melody performance and output the accompaniment by strictly following the given score and the musician's tempo. In order to create more interactive virtual performance, many improvements and extensions have been made, including vocal performance tracking [8], embellished melody recognition [6], smooth tempo adjustment [4][12], etc. Recently, studies have achieved more expressive virtual performance with music nuance [17][19] and robot embodiment [18]. However, automatic accompaniment systems generally follow the pitches and rhythms specified in the score, with no improvisation ability.

On the other hand, many systems have been created to improvise in contexts that range from free improvisation [11] to strictly following a set tempo and chord progression [1][13]. Early systems [3][14] incorporated compositional knowledge to created rule-based improvisation, and learning-based improvisation [2, 9, 10, 15] started to appear since 2000. One of the challenges of musical improvisation is to respond to other players while simultaneously adhering to constraints imposed by musical structure. In general, the most responsive computer improvisation systems tend to be free of local constraints such as following in tempo or following chords in a lead sheet. On the other hand, programs that are most aware of tempo, meter, and chord progressions, such as Band-in-a-Box and GenJam, tend to be completely unresponsive to real-time input from other musicians.

This study bridges *automatic accompaniment* and *computer-generated improvisation*. Automatic accompaniment systems illustrate that computers can simultaneously follow strict constraints (playing the notes of a score) while interacting intimately with another player (by synchronizing and, in recent work, even adjusting phrasing and dynamics). This paper considers an extension of this direction where an automatic accompanist not only follows a soloist, but *learns* to improvise an accompaniment, that is, to insert, delete and modify pitches and rhythms in a responsive manner.

We focus on a piano *duet interaction* and consider improvisation in a folk/classical music scenario. The music to be performed consists of a melody and a chord progression (harmony). In this deliberately constrained scenario, the melody is to be expressed clearly, but it may be altered and ornamented. This differs from a traditional jazz improvisation where a soloist constructs a new melody, usually constrained only by given harmonies. In musical terms, we want to model the situation where a notated melody is marked "*ad lib.*" as opposed to a passage of chord symbols marked "*solo.*" A melody that guides the performance enables more straightforward performance pattern learning and also makes the evaluation procedure more repeatable. The second part is simply a chord progression (a lead sheet), which is the typical input for a jazz rhythm section (the players who are not "soloing"). The second player, which we will implement computationally, is free to construct pitches and rhythms according to these chords, supporting the first (human) player who improvises around the melody.

It is important to note that the focus of this study is not the *performance* properties of individual notes (such as timing and dynamics) but the *score* properties of improvised interactive performance. Normally, improvisers play very intuitively, imagining and producing a performance, which might later be transcribed into notation. In our model, we do the opposite, having our system generates a symbolic score where pitch and rhythm are quantized. To gain training examples of improvised scores, we collected an improvised piano duet dataset, which contains multiple improvised performances of each piece. Our general solution is to develop a measure-specific model, which computes the correlation between various aspects of first piano performance and the score of the second piano performance measure-by-measure. Based on the learned model, an artificial performer constructs an improvised part based on a lead sheet, in concert with an embellished human melody

NIME'17, May 15-19, 2017, Aalborg University Copenhagen, Denmark.

performance. Finally, we conduct both objective and subjective evaluations and show that the learned model generates more musical, interactive, and natural improvised accompaniment compared with the baseline estimation.

The next section presents data collection. We present the methodology and experimental results in Sections 3 and 4, respectively. We conclude and discuss limitations and future work in Sections 5 and 6.

2. DATA COLLECTION

To learn improvised duet interaction, we collected a dataset that contains two songs: *Sally Garden* and *Spartacus Love Theme*, each performed 15 times by the same pair of musicians. All performances were recorded using electronic pianos with MIDI output.

The performances were recorded over multiple sessions. For each session, the musicians first warmed up and practiced the pieces together for about 10 minutes before the recording began. (We did not capture any individual or joint practicing procedure, only the final performance results.) Musicians were instructed to perform the pieces with different interpretations (emotions, tempi, etc.). The first piano player would usually choose the interpretation and was allowed (but not required) to communicate the interpretation with the second piano player before the performance.

An overview of the dataset can be seen in Table 1, where each row corresponds to a piece of music. The first column represents piece name. The 2nd to 4th columns represent the number of chords (each chord covers a measure on the lead sheet), average performance length, and the average number of embellished notes in the first piano performance.

Table 1. An overview of the improvised piano duet dataset.

name	#chord	avg. len.	#avg. emb.
Sally Garden	36	1’09’’	27
Spartacus Love Theme	20	53’’	12

3. METHODOLOGY

We present our data preprocessing technique in Section 3.1, where improvised duet performances are transcribed into score representations. Then, we show how to extract performance and score features based on processed data in Section 3.2. In Section 3.3, we present the “measure-specific” model. Based on this learned model, a virtual performer is able to construct an improvised accompaniment, which reacts to an embellished human melody performance, given a lead sheet.

3.1 Data Preprocessing

Improvisation techniques present a particular challenge for data preprocessing: performances no longer strictly follow the defined note sequences, so it is more difficult to align performances with the corresponding scores.

To address this problem, for the first piano part (the melody), we manually aligned the performances with the corresponding scores since we only have 30 performances in total and each of them is very short. For the second piano part, since the purpose is to learn and generate the scores, we want to transcribe the score of each performance before extracting features or learning patterns from it. Specially, since our performances were recorded by electronic pianos with MIDI outputs, we know the ground truth pitches of the score and only need to transcribe the rhythm (i.e., which beat each note aligns to).

The rhythm transcription algorithm contains three steps: *score-time calculation*, *half-beat quantization*, and *quarter-beat refinement*. In the first step, we compute raw score timings of the second piano notes using the local tempi of the aligned first piano part within 2 beats as the guidance. Figure 1 shows an example, where the performance time of the target note is x and its score time is computed as y. In this case, the neighboring context is from 7th to 11th beat, the “+” signs represent the onsets of the first piano notes within 2 beats of the target note, and the dotted line is the tempo map computed via linear regression.

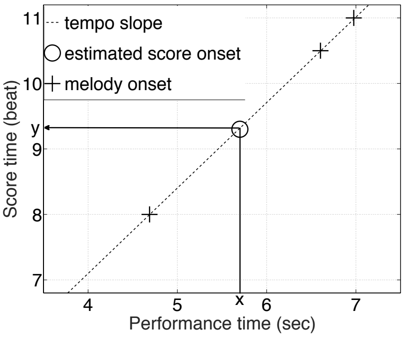


Figure 1. An illustration of rhythm transcription.

In the second step, we quantize the raw score timings computed in the first step by rounding them to the nearest half beats. For example, in Figure 1, y is equal to 9.3 and it will round up to 9.5. In the final step, we re-quantize the notes to ¼ beat if two adjacent notes were quantized to the same half beat in the second step and their raw score time is within the range of ¼ beat ± *error*. In practice, we set the *error* to be 0.07 beat. For the example in Figure 1, if the next note’s raw score time is 9.6, the two notes will be quantized to 9.5 in the second step but re-quantized to 9.25 and 9.5, respectively, in the final step. The rationale of the quantization rules is that for our dataset, most notes align to half-beat and the finest subdivision is ¼ beat

3.2 Feature Representations

Input and output features are designed to serve as an intermediate layer between transcribed data (presented in the last section) and the computational model (to be presented in the next section). The input features represent the score and the 1st piano performance, while the output features represent the transcribed score of the 2nd piano. Note that the *unit* for learning improvisation is a *measure* rather than a note. The reason is that an improvisation choice, especially the choice of improvised rhythm, of a measure is more of an organic whole than independent decisions on each note or beat.

3.2.1 Input Features

The input features reveal various aspects of the duet performance that affect the score of the second piano. Remember that the first piano part follows a pre-defined monophonic melody but allows embellishments. Formally, we use $x = [x_1, x_2, \ldots, x_i, \ldots]$ to denote the input feature sequence with i being the measure index of the improvised accompaniment. To be specific, x_i includes the following components:

Tempo Context: The tempo of the previous measure, which is computed by:

$$\mathrm{TempoContext}_i \overset{\mathrm{def}}{=} \frac{p_{i-1}^{\mathrm{last}} - p_{i-1}^{\mathrm{first}}}{s_{i-1}^{\mathrm{last}} - s_{i-1}^{\mathrm{first}}} \qquad (1)$$

where p_{i-1}^{first}(or s_{i-1}^{first}) and p_{i-1}^{last} (or s_{i-1}^{first}) represent the performance time (or score time) of the first and last note in the previous measure, respectively.

Embellishment Complexity Context: A measurement of how many embellished notes are added to the melody in the previous measure. Formally,

$$\mathrm{EmbComplexityContext}_i \overset{\mathrm{def}}{=} \log\left(\frac{\#P_{i-1} - \#S_{i-1} + 1}{\#S_{i-1} + 1}\right) \qquad (2)$$

where $\#S_{i-1}$ represents the number of notes defined in the score and $\#P_{i-1}$ represents the number of actual performed notes.

Onset Density Context: The onset density of the second piano part in the previous measure, which is defined as the number of score onsets. Note that one chord just count as one onset. Formally:

$$\text{OnsetDensityContext}_i \overset{\text{def}}{=} \#\,\text{Onset}_{i-1} \quad \textbf{(3)}$$

Chord Thickness Context: The chord thickness in the previous measure, which is defined as the average number of notes in each chord. Formally:

$$\text{ChordThicknessContext}_i \overset{\text{def}}{=} \frac{\#\,\text{Note}_{i-1}}{\text{OnsetDensityContext}_i} \quad \textbf{(4)}$$

where $\#\,\text{Note}_{i-1}$ represents the total number of notes in the previous measure.

3.2.2 Output Features

For each measure, we focus on the prediction of its *onset density* and *chord thickness*. Formally, we use $y = [y_1, y_2, \ldots, y_i, \ldots]$ to denote the output feature sequence with i being the measure index. Referring to the notations in Section 3.2.1, y_i includes the following two components:

$$\text{OnsetDensity}_i \overset{\text{def}}{=} \#\,\text{Onset}_i \quad \textbf{(5)}$$

$$\text{ChordThickness}_i \overset{\text{def}}{=} \frac{\#\,\text{Note}_i}{\#\,\text{Onset}_i} \quad \textbf{(6)}$$

To map these two features into an actual score, we use *nearest-neighbor search* treating onset density as the primary criteria and chord thickness as the secondary criteria. Given a predicted feature vector, we first search the training examples (score of the same measure for other performances) and select the example(s) whose onset density is/are closest the predicted onset density. If multiple candidate training examples are selected, we then choose the candidate whose chord thickness is closest to the predicted chord thickness. If there are still multiple candidates left, we randomly choose one from them.

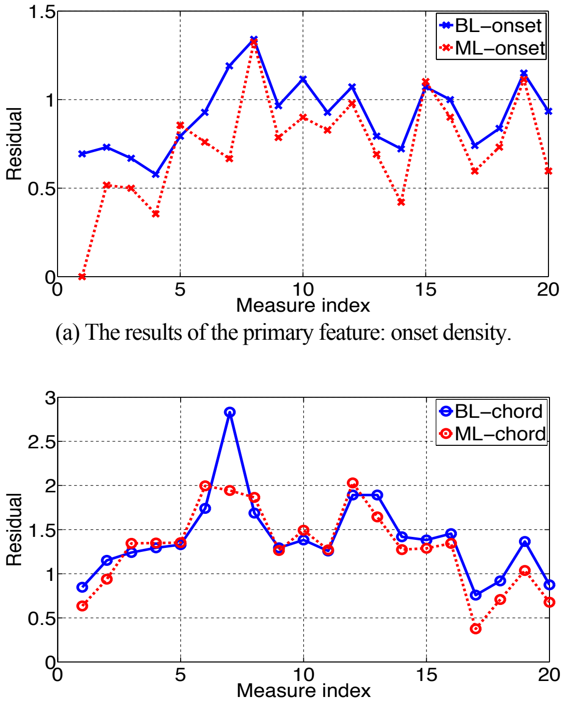
3.3 Model

We developed a measure-specific approach, which trains a different set of parameters for every measure. Intuitively, this approach assumes that the improvisation decision on each measure is linearly correlated to performance tempo, melody embellishments, and the rhythm of the previous measure. Formally, if we use $x = [x_1, x_2, \ldots, x_i, \ldots]$ and $y = [y_1, y_2, \ldots, y_i, \ldots]$ to denote the input and output feature sequences with i being the measure index, the model is:

$$y_i = \beta_0^i + \beta^i x_i \quad (7)$$

For both pieces of music we used in this study, the melody part starts before the accompaniment part as pickup notes in the score. Therefore, when $i = 1$, the input feature x_1 is not empty but only contains the first two components: "tempo context" and "embellished complexity context". (If the accompaniment part comes before the melody part, x_1 would only contain the last two components. In case the two parts start together, we can randomly sample from the training data.)

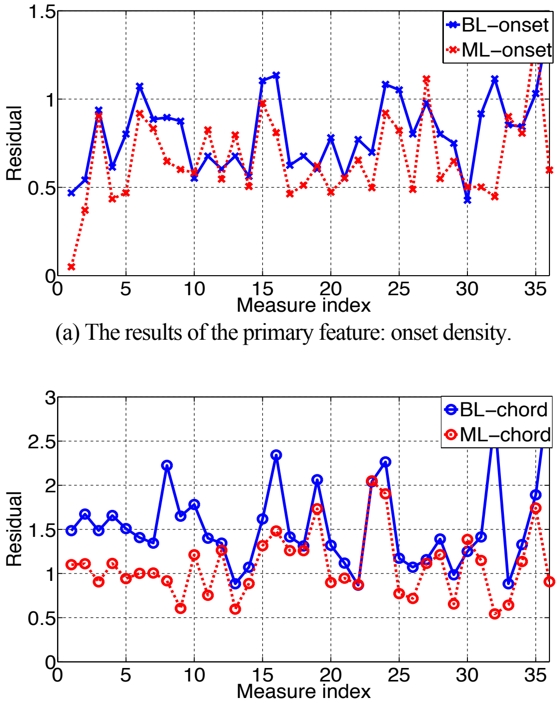
The measure-specific approach is able to model the improvisation techniques even if it does not consider many of the compositional constraints. (For example, what the proper pitches are given a chord, and what the proper choices of rhythm are given the relative position of a measure in the corresponding phrase.) This is because we train a tailored model for each measure and most of these constraints have already been encoded in the training examples. Therefore, when we decode (generate) the performance using nearest-neighbor search on training performances, the final output performance will also meet the music structure constraints.



(a) The results of the primary feature: onset density.

(b) The results of the secondary feature: chord thickness.

Figure 2. The residuals of the piece *Sally Garden.* (Smaller is better.)



(a) The results of the primary feature: onset density.

(b) The results of the secondary feature: chord thickness.

Figure 3. The residuals of the piece *Spartacus Love Theme.* (Smaller is better.)

4. EXPERIMENTS

Our objective evaluation measures the system's ability to predict performance output features of real human performances from input features. We adopted the *mean* of the output features of all training samples as our baseline prediction and compare that to model predictions, using leave-one-out cross validation. For subjective evaluation, we designed a survey and invited people to rate the synthetic performances generated by different models.

4.1 Objective Evaluation

Figure 2 and Figure 3 show results for the two pieces, where we see that for most measures, the measure-specific approach outperforms the baseline. For both figures, the *x* axis represents the measure index and the *y* index represent the mean of the absolute residuals between model prediction and human performance. The subfigure (a) shows the residuals of onset density, while subfigure (b) shows the residuals of chord thickness. The curves with "x" markers show onset density (the primary feature) and the circles mark chord thickness (the secondary feature). The solid curves represent residuals of the baseline approach (sample means) and the dotted curves represent residuals of the measure-specific approach. Therefore, small numbers mean better results.

4.2 Subjective Evaluation

Besides the *objective* evaluation, we invited people to *subjectively* rate our model through a double-blind online survey. (https://www.surveymonkey.com/r/CMUMusic) During the survey, for each performance, subjects first listened to the first piano part (the melody part) alone, and then listened to *three* synthetic duet versions (conditions):

BL: The score of the second piano is generated by the baseline mean estimation.

ML: The score of the second piano is generated by the measure-specific approach.

QT: The score of the second piano is the quantized original (ground truth) human performance.

The three versions share exactly the same first piano part and their differences lie in the second piano part. As our focus is the evaluation of improvisation of pitch and rhythm, the timing and dynamics of all the synthetic versions are generated using the automatic accompaniment approach in [5]. In addition, since the experiment requires careful listening and a long survey could decrease the quality of answers, each subject only listened to 4 of the performances, with 2 performances per piece of music, by random assignment. The order was also randomized both within a performance (for different duet versions) and across different performances.

After listening to each duet version, subjects were asked to rate the second piano part in the duet performance on a 5-point scale from 1 (very low) to 5 (very high) according to three criteria:

Musicality: How musical the performance was.

Interactivity: How close the interaction was between the two piano parts.

Naturalness: How natural (human-like) the performance was.

Since each subject listened to all three versions (conditions) of synthetic duets, we used one-way repeated measures *analysis of variance* (ANOVA) [7] to compute the p-value and mean squared error (MSE). Generally, repeated measurements ANOVA can be seen as an extension of paired t-test in order to compute the difference between more than two conditions. It removes variability due to the individual differences from the within-condition variance and only keeps the variability of how the subject reacts to different conditions (versions of duets).

A total of *n* = 42 subjects (13 female and 29 male) with different music backgrounds have completed the survey. The aggregated result (as in Figure 4) shows that the measure-specific model improves the subjective rating significantly compared with the baseline for all three criteria (with p-values less than 0.05). Here, different colors represent different conditions (versions). The heights of the bars represent the means of the ratings and the error bars represent the MSEs computed via repeated measurements ANOVA.

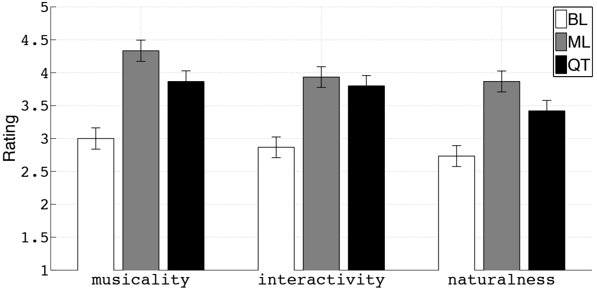


Figure 4. The subjective evaluation results of improvised interactive duet. (Higher is better.)

Surprisingly, our method generates better results than scores transcribed from original human performances (marked "QT"), though the differences are not significant (with the p-values larger than 0.05). Note that this result does not indicate the measure-specific model is better than the *original* human performance because the timing and dynamics parameters are still computed by an automatic accompaniment algorithm for the "QT" version. We also tested whether different pieces or different music backgrounds make a difference but with no significant results.

5. CONCLUSIONS

In conclusion, we created a virtual accompanist with basic improvisation techniques for duet interaction by learning from human duet performances. The experimental results show that the developed measures-specific approach is able to generate more musical, interactive, and natural improvised accompaniment than the baseline mean estimation.

Previous work on machine learning and improvisation has largely focused on modeling style and conventions as if collaboration between performers is the indirect result of playing the same songs in the same styles with no direct interaction. *Our work demonstrates the possibility of learning causal factors that directly influence the mutual interaction of improvisers.* This work and extensions of it might be combined with other computational models of jazz improvisation, including models that make different assumptions about the problem (such as allowing "free" melodic improvisation) or have stronger generative rules for constructing "rhythm section" parts. This could lead to much richer and more realistic models of improvisation in which mutual influences of performers are appreciated by listeners as a key aspect of the performance.

6. LIMITATIONS AND FUTURE WORK

As mentioned above, the current method needs 15 rehearsals to learn the performance of each measure, which is a large number in practice. To shrink the training set size, we plan to consider the following factors in improvised duet interactions: 1) general improvisation rules that apply to different measures or even different pieces of music, 2) complex music structures, and 3) performer preferences and styles. Also, the current subjective evaluation is conducted on audience only; we are going to invite multiple performers as our subjects.

7. ACKNOWLEDGMENTS

We would like to thank to Laxman Dhulipa and Andy Wang for their contribution to the piano duet dataset. We would also like to thank to Spencer Topel and Michael Casey for their help and suggestions.

8. REFERENCES

[1] J. Biles. Genjam: A genetic algorithm for generating jazz solos. In *Proceedings of the 1994 International Computer Music Conference*, 1994, 131-137.

[2] M. Bretan, G. Weinberg, and L. Heck: A unit selection methodology for music generation using deep neural networks. *arXiv preprint arXiv:1612.03789*, 2016.

[3] J. Chadabe. Interactive Composing: An Overview. *Computer Music Journal*, 1984, 22-27,.

[4] A. Cont., ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters. In *Proceedings of International Computer Music Conference*, 2008.

[5] R. Dannenberg. An online algorithm for real-time accompaniment. In *Proceedings of the International Computer Music Conference*, 1984, 193-198.

[6] R. Dannenberg and H. Mukaino. New techniques for enhanced quality of computer accompaniment. In *Proceedings of the International Computer Music Conference*, 1988, 243–249.

[7] R. Ellen and E. Girden. *ANOVA: Repeated Measures*. No. 84. Sage, 1992.

[8] L. Grubb and R. Dannenberg. A stochastic method of tracking a vocal performer. In *Proceedings of the International Computer Music Conference*, 1997, 301-308.

[9] G. Hoffman and G. Weinberg. Interactive improvisation with a robotic marimba Player. *Autonomous Robots* 31, 2-3, 2011, 133-153.

[10] M. Kaliakatsos-Papakostas, F. Andreas, and N. Michael Intelligent real-time music accompaniment for constraint-free improvisation. In *Proceedings of the 24th International Conference on Tools with Artificial Intelligence*, 2012.

[11] G. Lewis, "Too Many Notes: Computers, Complexity and Culture in Voyager." *Leonardo Music Journal,* 10, 2000, 33-39.

[12] D. Liang, G. Xia, and R. Dannenberg, A framework for coordination and synchronization of media. In *Proceedings of the New Interfaces for Musical Expression*, 2011.

[13] PG Music. *Band-in-a-Box, RealBand, and more.* 2017. http://www.pgmusic.com/, (accessed 2017).

[14] R. Rowe. *Interactive Music Systems: Machine Listening & Composing*. MIT Press, 1993.

[15] B. Thom. Unsupervised learning and interactive jazz/blues improvisation. *In Proceedings of the Twelfth National Conference on Artificial Intelligence*, 2000.

[16] B. Vercoe, .The synthetic performer in the context of live performance. In *Proceedings of the International Computer Music Conference*, 1984, 199-200.

[17] G. Xia and R. Dannenberg. Duet interaction: learning musicianship for automatic accompaniment. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2015.

[18] G. Xia, et al. Expressive humanoid robot for automatic accompaniment. In *Proceedings of the Sound and Music Computing Conference*, 2016.

[19] G. Xia, Y. Wang, R. Dannenberg, and G. Gordon. Spectral learning for expressive interactive ensemble music performance. In *Proceedings of the 16th International Society for Music Information Retrieval Conference*, 2015.