Duet Interaction: Learning Musicianship for Automatic Accompaniment

Guangyu Xia
Carnegie Mellon University
School of Computer Science
5000 Forbes Ave, Pittsburg, PA
gxia@cs.cmu.edu

Roger B. Dannenberg
Carnegie Mellon University
School of Computer Science
5000 Forbes Ave, Pittsburgh, PA
rbd@cs.cmu.edu

ABSTRACT
Computer music systems can interact with humans at different levels, including scores, phrases, notes, beats, and gestures. However, most current systems lack basic musical skills. As a consequence, the results of human-computer interaction are often far less musical than the interaction between human musicians. In this paper, we explore the possibility of *learning* some basic music performance skills from rehearsal data. In particular, we consider the piano duet scenario where two musicians expressively interact with each other. Our work extends previous automatic accompaniment systems. We have built an artificial pianist that can automatically improve its ability to sense and interact with a human pianist, learning from rehearsal experience. We describe different machine learning algorithms to learn aspects of expressive timing and dynamics for duet interaction, explore the properties of the learned models, such as dominant features, limits of validity, and minimal training size, and claim that a more human-like interaction is achieved.

Author Keywords
Interactive, Music Expression, Classical Music, Duet, Automatic Accompaniment.

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-Parameter learning

1. INTRODUCTION
Computer music systems have achieved a wide spectrum of application in music performance, ranging from fixed media to free improvisation. The broad range of practice includes Music Minus One (fixed media), score following & automatic accompaniment, human-computer music performance, and interactive computer music. These systems interact with humans at different levels, such as scores, phrases, notes, beats, and gestures. However, since most current systems lack representations and capabilities of musicianship, the human-computer interaction is often far less musical than the interaction between human musicians. We aim to explore the possibility to empower current computer music systems with musical skills and knowledge. In this paper, we focus on the piano duet scenario in which two pianists expressively interact with each other. The goal is to incorporate musicianship into the existing framework of automatic accompaniment systems, extending the system's ability from passive synchronization to mimicking the behavior of ensemble musicians. Of course, our system does not learn *all* aspects of musicianship. The current system learns aspects of expressive timing and dynamics, and we believe other aspects of musicianship can also be learned in a similar way.

This exploration can be seen as a marriage of two existing fields of research: *expressive performance*, which studies how musicians vary timing and other parameters in music performance, and *automatic accompaniment*, which studies how to create an artificial musician that can follow a score and synchronize its performance with humans.

It is well known that musicians in ensembles interact with each other to achieve a shared musical performance. The art for the musicians is not only to interpret the music on their own, but also to keep in concert with each other by continuously adjusting timing, dynamics, etc. To sense and interact with each other's musical expression is both important and difficult, so musicians spend much time together in rehearsals. It is through rehearsals that musicians become familiar with each other's music interpretation and create their own expressive response. This procedure of learning musical interaction through rehearsal suggests that a computer system could be trained in a similar way by using machine learning algorithms. To be more precise, we look forward to answering the following research question:

How can we build an artificial performer that, with rehearsal experience, automatically improves the ability to sense and interact with a human musician's expression, and what are the fundamental laws that govern the learning processes?

There are many issues to be addressed. First of all, within each performance, how should the artificial performer choose the expressive parameters for each note based on the expression of the human musician? E.g., should the artificial performer completely follow the human musician's tempo, keep steady, or even behave to the contrary? Second, how can we design machine-learning algorithms to distill the models from rehearsal experience? In other words, how can we learn regularity from seemingly irregular data? Third, what are the dominant features that affect the expressive interaction? E.g., is expressive timing affected more by rhythm or by melody? Fourth, what are the limits of validity of the learned models? E.g., which model generalizes across the whole piece of music and which model only works for some specific score locations? Last but not least, how many rehearsals are needed to train the artificial performer; would the number be reasonable in practice?

To answer these questions, we conduct research in four phases. First, features are extracted to represent the music expression of both musicians. Second, function approximations are designed to reveal the relationship between one's music expression and the other's. Third, different properties of the algorithms, such us dominant features, models' limits of validity, and minimum training size are

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. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
NIME'15, May 31-June 3, 2015, Louisiana State University, Baton Rouge, LA.
Copyright remains with the author(s).

explored. Finally, given only one musician's performance, the performance of the artificial performer is synthesized by using learned models, based on which evaluation is done.

The next section presents related work. Section 3 describes the data collection. Section 4 introduces a spectrum of algorithms to learn musicianship from rehearsal data. In Section 5, we present experimental results.

2. RELATED WORK

We review two realms of related work in computer music, which are *Score Following and Automatic Accompaniment* and *Expressive Performance*. The former one focuses on human-computer interactive performance, while the latter one focuses on musical expression. The two realms have been pursued for about 30 years but never really informed each other; this study could be seen as a marriage of the two.

2.1 Score Following and Automatic Accompaniment

In 1984, score following and automatic accompaniment systems were independently introduced by Dannenberg [8] and Vercoe [26]. Given a pre-defined music score, the system is able to follow a musician's monophonic performance in real time and output the accompaniment by strictly following the musician's tempo. Dannenberg's work was soon commercialized by SmartMusic and has been used by thousands of students for music practice. Ever since then, many extensions [2, 6, 7, 9, 15, 16] have been made by Dannenberg and his collaborators. Bloch and Dannenberg developed fast methods for following polyphonic performance input; Grubb and Dannenberg [15] extended this idea further to handle ensemble performance input. Later on, Grubb and Dannenberg [16] developed the first stochastic method for tracking vocal performer. More recently, several advanced probabilistic models have been introduced [6, 7, 19] for more robust score following.

Despite all of these efforts, most attention has been given to the "score following" part of the system, while the musical expression or the "automatic accompaniment" part has been overlooked. As a consequence, while "score following" has already become a more-or-less solved problem, recent systems still compute accompaniment timing by score-performance time mapping and extrapolating to the next note (which was introduced 30 years ago). In other words, computer systems are still "passive" in the sense that they do not have any particular knowledge about performance to actively predict human behavior and make choices on music interpretation. By looking back to Dannenberg's original work, it was clearly stated at the beginning of the "Limitations" section that, "the present set of algorithms make no attempt to adjust tempos in a particularly musical manner… Furthermore, no effort has been made to respond to the soloist in any way other than temporally. For example, a human accompaniment is expected to respond to loudness, articulation, and other nuances in addition to temporal cues."

Raphael's Music Plus One [19] and IRCAM's AnteScofo system [6] consider the accompaniment problem. The former one trains a Bayesian network by rehearsals to achieve more precise synchronization; the latter one uses a synchronization model based on Large's work [18] to achieve more natural tempo adjustment. However, the perspective is still limited to temporal synchronization; the computer's active role in shaping different musical expression is not yet considered.

2.2 Expressive Performance

The discipline of expressive performance studies how to convert static scores into human-like expressive performances by different computational models (See [17] and [29] for comprehensive overviews.) The models fall into three main categories, which are rule-based modeling, case-based modeling, and probabilistic modeling. Generally speaking, probabilistic modeling works better than the others, and there is evidence that even better performance can be achieved by combining different models.

Rule-based systems, appearing in the early 1980s, generate performances based on defined or discovered performance rules [11, 12, 20, 21, 23-28]. Sundberg and his collaborators built the well-known KTH model by an innovative "analysis-by-synthesis" approach in which musicians and researchers worked together in a crowdsourcing way [20, 21]. Others discover rules by collecting measurements from actual performance data. Among them, Todd [23, 24] focused on the relationship between music structure and performance. Widmer developed various data mining methods [27, 28] to discover rules from data automatically. Since the late 1990s, cased-based reasoning systems have appeared, which generate performances by adopting previous performance examples. Two representative ones are the SaxEx system [1] developed by Arcos etc., and DISTALL system [30] developed by Widmer and Tobudic.

More recently, we see probabilistic modeling systems [10, 14]. These systems model the conditional probabilistic distribution of the performance given the score, and then generate new performances by sampling from the learned models. From the machine learning perspective, the underlying graphical models used in these studies serve as good basis for this study. Notice that the foci are quite different. These systems focus on the relationship between score and interpretation, while this paper focuses on the interaction between different interpretations.

3. DATA COLLECTION

Musicians: We invited 10 graduate students from the School of Music in our university to perform duet pieces in 5 pairs.

Music pieces: We selected 3 pieces of music – *Danny Boy*, *Serenade* (by *Schubert*), and *Ashokan Farewell* – based on their suitable length and difficulty for recording. Each pair of musicians performed every piece of music 7 times with instructions to use different interpretations. Therefore, for each piece of music, we have collected $5 \times 7 = 35$ performances. In total, we have collected $35 \times 3 = 105$ performances.

Recording settings: Musicians performed the music by sitting face to face. Pieces were recorded using electronic pianos with MIDI output, therefore all the parameters (dynamics, starting time, ending time, pedal) of every note can be recorded accurately in real time.

Recording procedures: Musicians warm up by practicing the pieces for about 10 minutes together and then start recording. Each recording session records about 15 performances and lasts for about 1 hour.

4. METHODOLOGY

Different function approximations are designed to model the relationship between one pianist's music expression and another's. We start from very low-dimensional representation and local models that only apply to certain notes in music, and gradually step to high-dimensional representation and more general models that can apply to the whole piece of music.

Based on the learned models, an artificial performer will be able to generate (decode) its own music expression by interacting with a human pianist. For piano notes, music

expression is encoded by timing, dynamics, and pedal position. (I.e., once we know these parameters, we can re-synthesize the note.) In this paper, we consider timing features and dynamics features. We focus mainly on timing prediction. Dynamics prediction uses very similar steps, so we only include a brief description of our work on dynamics.

4.1 Expressive Timing

4.1.1 Baseline approach

We use the timing estimation algorithm in [8] as the baseline for comparison to new techniques. As shown in Figure 1, we estimate a linear mapping between real time and reference time (usually score time, in beats) by fitting a straight line to recently performed and recognized note onsets. This mapping can be used to estimate the time of the next note. To make a more fair comparison, we also consider rehearsal performances for the baseline algorithm. Rather than directly taking the score as the reference, we use the "median performance" of the rehearsals.

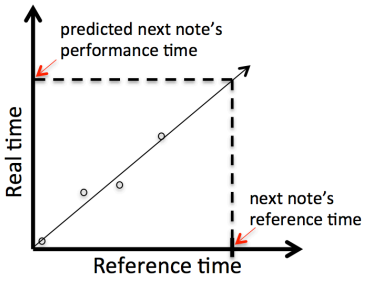

Figure 1. Baseline approach of timing estimation.

Here, we also define *Expressive timing* as the timing difference between actual performance time and the baseline algorithm's predicted performance time by using the score as reference. The expressive timing tells us how much the actual performance timing differs from the score trend.

4.1.2 Note-specific approach

The note-specific approach assumes that expressive timing of the notes is linearly correlated so that we can predict a note's expressive timing based on the expressive timing of previous notes. Intuitively, we assume that when musicians slow down, speed up, or use rubato, there are certain patterns of expressive timing that can be characterized by linear regression. Formally, let $X = [x_1, x_2, \dots, x_N]$ be the expressive timing of the notes played by the 1st pianist; let $Y = [y_1, y_2, \dots, y_M]$ be the expressive timing of the notes played by the 2nd pianist. (N and M are note indices). Then the model is:

$$y_i = \beta_0^{y_i} + \sum_{j=1}^{p} \beta_j^{y_i} x_{over(y_i)-j} \qquad (1)$$

Here, p is the lag parameter and $x_{over(y_i)-j}$ are the p note times in X previous to y_i. Thus, $over(y_i)$ is the smallest index of the element of X whose score time is greater or equal to the score time corresponding to y_i. For example, in Figure 2, let the 1st and 2nd systems be the score for the 1st and 2nd piano, respectively. If the note in the dotted circle corresponds to y_i and the lag parameter p is equal to 3, the notes in the circle would corresponds to $x_{over(y_i)-j}$.

It is important to notice that the note-specific approach trains a different set of parameters for each note, which is reflected by the superscript of β. The advantage of this approach is that each note gets a tailored solution, while the disadvantage is that many training rehearsals are needed.

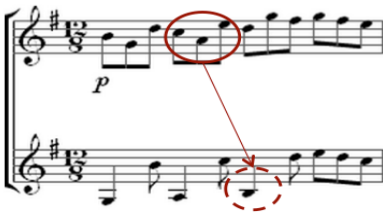

Figure 2. An illustration of the note-specific approach.

4.1.3 Rhythm-specific approach

To improve the generality of the model, the rhythm-specific approach introduces an extra dummy variable to encode the score rhythm context of each note. This is mathematically equivalent to training a different set of parameters for each rhythm context. Intuitively, we assume that notes of the same rhythm context share the same pattern of expressive timing. Formally, let X and Y be the same as in the note-specific approach. The rhythm-specific model is then:

$$y_i = \beta_0^{rhythm(y_i,q)} + \sum_{j=1}^{p} \beta_j^{rhythm(y_i,q)} x_{over(y_i)-j} \qquad (2)$$

where $rhythm(y_i, q)$ is the categorical variable representing the rhythm context of the note y_i within q notes. To be more precise, the rhythm context of y_i is defined as the inter-onset-intervals of the q 1st piano's notes right before y_i. As q increase, the possible values of $rhythm(y_i, q)$ will also increase. For example, in Figure 3, again let the 1st and 2nd systems be the scores for the 1st and 2nd piano, respectively. When q is equal to 3, the two notes in the dotted circles would share the same $rhythm(y_i, q)$. The two notes' rhythm contexts are shown by the circled notes.

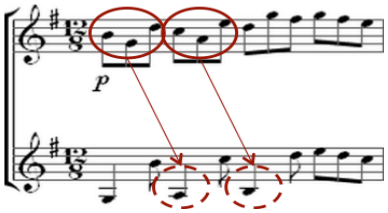

Figure 3. An illustration of the rhythm-specific approach.

It is important to notice that many notes share the same rhythm context within a piece of music and hence share the same set of parameters. As a consequence, the model can gain more information from each rehearsal, and fewer training rehearsals are needed compared to the note-specific approach. However, this improvement doesn't apply to some "odd notes" whose rhythm contexts are unique. For these notes, the rhythm-specific approach reduces to the note-specific approach.

4.1.4 General feature approach

To further improve the model's generality and predict the expressive timing by more than rhythm context, a more general and comprehensive representation is designed. In particular, features are designed from four aspects of expressive interactive performance, as shown in Figure 4.

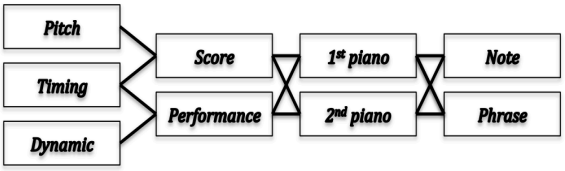

Figure 4. The general feature scheme.

In Figure 4, each column represents an aspect of the feature and each line between the columns represents a possible interaction. (E.g., notice that there's no linkage between the "Pitch" block and the "Performance" block since pitches are defined in the score.) Based on this graph, we extract actual features by going through all the possible paths. E.g., the rhythm context feature in Section 4.1.3 corresponds to the "Timing-Score-1st piano-Note" path and the dependent variable Y corresponds to the "Timing-Performance-2nd piano-Note" path. In total, we exhaust the possible 16 paths and construct a high-dimensional feature space. Formally, let $U = [u_1, u_2, \dots, u_M]$ be the general features (dependent variables excluded) of the notes played by the 2nd pianist; let $Y = [y_1, y_2, \dots, y_M]$ be the same as in Section 4.1.3. The model is:

$$Y = BU \quad (3)$$

where Y is 1-by-M, B is 1-by-P, and U is P-by-M. P is the dimensionality of the feature space. This equation can be solved easily by performing Moore-Penrose pseudo-inverse.

We further consider the group lasso [13] penalty to find the optimal parameter settings by solving the following convex optimization problem:

$$\min_{B \in R^P} \left(\|Y - BU\|_2^2 + \lambda \sum_{l=1}^{L} \sqrt{p_l} \|B_l\|_2 \right) \quad (4)$$

where λ is the penalty parameter, l is the feature group index, p_l is the dimensionality of l^{th} feature group, and B_l is the parameters corresponding to the l^{th} feature group. For our application, a feature group is defined by a specific path in Figure 4. The advantage of group lasso regularization is that it not only reduces the burden for training but also tries to discover the dominant aspect of interactive performance that could be used to predict the expressive timing.

4.1.5 Linear dynamic system approach

Our latest effort on expressive timing feature is to link up the notes and model them by a time-invariant linear dynamic system (LDS). In particular, we assume there exist some low dimensional hidden mental states. The mental states change smoothly over time and control the expressive timing. Intuitively, the LDS approach could be seen as adding another regularization to the expressive timing by adjacent notes' music expression. Formally, we adopt the following graphical representation:

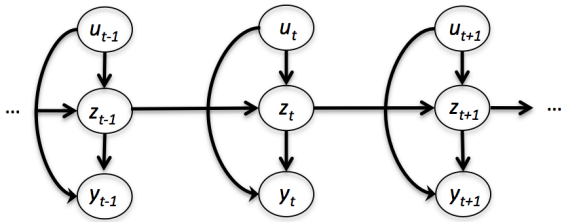

Figure 5. The graphical representation of the LDS.

In Figure 5, u and y are the same representations as in Section 4.1.4, while z represents the hidden state. In LDS, y is referred to as *observation* and u is referred to as *input*. The evolution of this time series can be described by the following equations:

$$z_t = Az_{t-1} + Bu_t + w_t \quad w_t \sim \mathcal{N}(0, Q) \quad (5)$$

$$y_t = Cz_t + Du_t + v_t \quad v_t \sim \mathcal{N}(0, R) \quad (6)$$

To learn the model, we adopted the spectral method [3, 4, 5], also known as subspace identification in control theory. Generally speaking, the spectral method learns the LDS by reduced-ranked partial regressions. Overschee and De Moor [25] give a detailed derivation and proof.

4.2 Expressive Dynamics

To predict expressive dynamics, we follow almost exactly the pipeline of algorithms in Section 4.1 that predict expressive timing. Here we just point out the differences.

For the baseline approach, the trend of dynamics, unlike the trend of timing, is not defined by the score in detail. But we can at least figure out the basis dynamic level by looking at the performance of the 1st piano. By assuming the dynamics of a piece is locally stable, for each note of the 2nd piano, we use its previous 1st piano note's dynamic as our baseline prediction. For the other approaches, the notation x (also X) and y (also Y) now represent dynamics rather than timing. Other notations and all the equations are exactly the same as in Section 4.1.

5. EXPERIMENTS

Remember that we have 35 rehearsals for each piece of music. To compare the results of different methods and to choose the optimal parameters, we use 35-fold cross-validation. The measurement is the average (over different performances) of absolute timing and dynamics differences between the predictions and the ground truths. Therefore, small numbers mean better predictions. The maximum possible training size is 34 (leave-one-out cross-validation). When the training size is less than 34, we randomly sample the training performances from the rehearsals, excluding the test one. We show both detailed results (over score time) and high-level statistics. Because of space limitations, we present only detailed timing results for *Danny Boy*. (The other two pieces and dynamics feature yield similar results).

5.1 Note-specific approach for timing

Figure 6 shows the result of the note-specific approach in which the lag parameter p is 4. The curve with diamond markers represents the baseline method, the curve with square markers represents the note-specific method trained by 8 rehearsals, and the curve with "x" markers represents the note-specific method trained by 34 rehearsals. We can see that the note-specific method works very well when there are a lot of training rehearsals but not so well when the training size is reduced to 8. 34 rehearsals is doable but is considered a large number in practice.

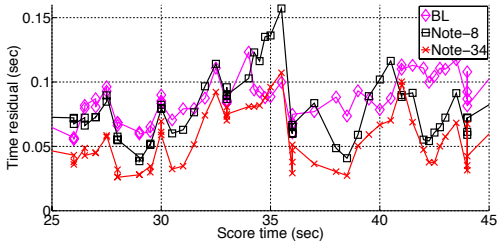

Figure 6. A zoom-in view of the absolute timing residuals of the note-specific approach.

5.2 Rhythm-specific approach for timing

Figure 7 shows the result of the rhythm-specific approach in which the lag parameter p and rhythm context parameter q are both 4. Again, the curve with diamond markers represents the baseline method. The curve with square markers represents the rhythm-specific method trained by 4 rehearsals, and the curve with "x" markers represents the rhythm-specific method trained by 8 rehearsals.

We can see that when there are 8 training rehearsals, the rhythm-specific method improves the performance a lot compared to the note-specific method. However, when we

shrink the training size to 4, the "odd notes" around 41s are not predicted well and in fact are off the scale shown here.

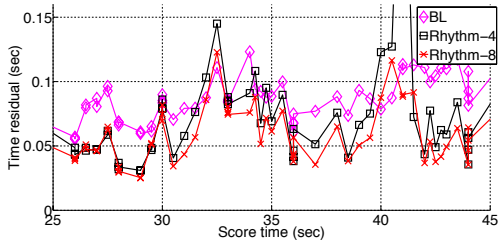

Figure 7. A zoom-in view of the absolute timing residuals of the rhythm-specific approach.

5.3 General feature approach for timing

With the baseline represented in the same way as previous figures, Figure 8 shows the current result of the general feature approach with only 4 training rehearsals, in which the regularization parameter λ is 5 and all the features are extracted from a local context of 6 to 8 beats. The curve with square markers represents the basic regression (without regularization), and the curve with "x" markers represents the regression with group lasso regularization. We can clearly see that with only 4 training rehearsals, basic regression outperforms the baseline most of the time. Group lasso regularization helps a lot and makes it much better than baseline everywhere.

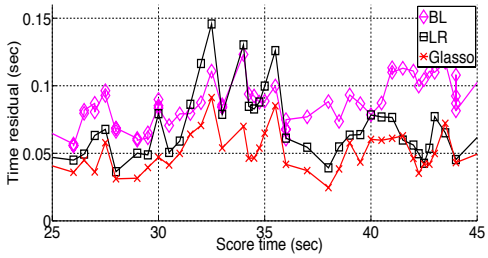

Figure 8. A zoom-in view of the absolute timing residuals of the general feature approach.

An interesting discovery is that group lasso regularization almost only retains rhythm context features and performance timing features, which indicates that expressive timing is mainly affected by rhythm context and not so much affected by the pitch.

5.4 LDS approach for timing

With the baseline and basic regression represented in the same way as the previous figure, Figure 9 shows the current result of the LDS approach with only 4 training rehearsals. As pointed out in Section 4.1.5, the LDS approach can be seen as adding another regularization.

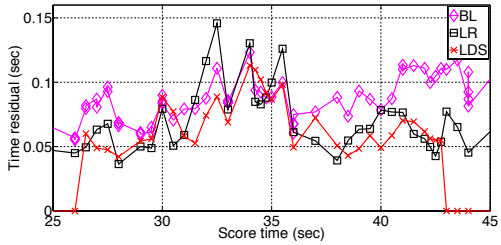

Figure 9. A zoom-in view of the absolute timing residuals of the LDS approach.

We see that this regularization also helps, but not as much as group lasso regularization. The possible reason is that the current feature settings are optimized for the LDS approach. More advanced features can be added for LDS since it can handle higher dimensional features.

5.5 A global view of timing predictions

For each curve in timing experiments, we take its mean over score time to compute a high-level statistic, a single number that describes how much on average our timing prediction differs from ground truth for each note.

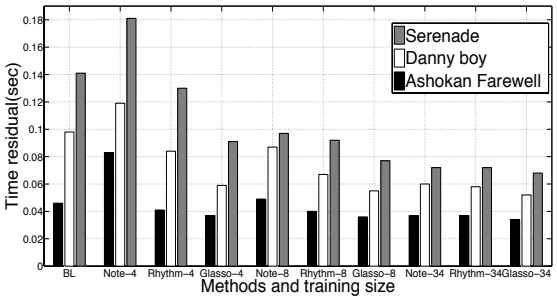

Figure 10. A global view of absolute timing residuals for three pieces of music. (Smaller is better.)

In Figure 10, we show this number for all the three pieces of music. Each color corresponds to a piece of music and each bar corresponds to a specific method trained by a certain number of rehearsals. E.g. the grey bar above Glasso-8 is the result for Serenade computed by the group lasso method with 8 training rehearsals. We see that the best results are generated by the general feature approach with group lasso regularization (regardless of the training size). With only 4 to 8 training examples, we are able to shrink the timing residuals by 10 to 60 milliseconds, especially when the baseline algorithm is not doing a good job.

5.6 A global view of dynamics predictions

Similar to Figure 10, Figure 11 shows how much on average our dynamics predictions differ from ground truth for each note. Again, we see that the best results are generated by the general feature approach with group lasso regularization. With only 4 to 8 training examples, we are able to shrink the dynamics residuals by about 6 in MIDI velocity.

We notice that in both Figure 10 and Figure 11, Note-4, i.e., note-specific approach trained by only 4 rehearsals, is the worst. This is caused by over-fitting since the dimensionality (p+1=5) is larger than the training size.

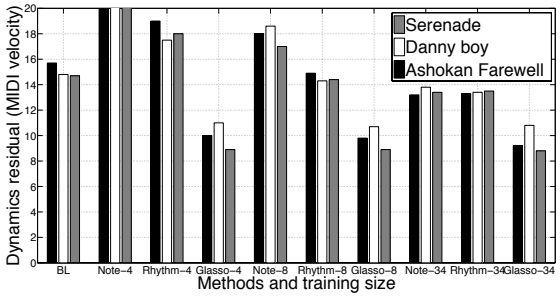

Figure 11. A global view of absolute dynamics residuals for three pieces of music. (Smaller is better.)

6. CONCLUSIONS AND FUTURE WORK

In conclusion, we have designed a spectrum of machine learning algorithms to learn musicianship for duet interaction from rehearsal experience. For both expressive timing and dynamics, we are able to use linear models to make much better prediction compared to the baseline algorithm, which has served for decades as the basis in computer accompaniment systems. With our best algorithms, we see improvement using only 4 to 8 rehearsals. Also, we have two interesting discoveries. First, expressive timing is highly related to the rhythm context, but not affected much by pitch. Second, a 6 to 8 beats local context leads to the best estimations. Shorter context is not informative enough while longer context results in overfitting.

In the future, we will continue the work in the following three directions:

Explore deeper into the LDS approach: The current feature settings are not optimized for the spectral method. We are going to extend the features since spectral methods can handle a much higher feature space. Also, we are working on integrating the group lasso regularization into the LDS approach.

Cross-piece models: So far, the most generalized model only works within a piece. How can we train the model from some pieces and apply it to other pieces? To answer this question, we are going to collect more data and design new features.

Performer-specific models: So far, the models generalize to all performers. We are going to consider the individual performer's character in future work.

7. ACKNOWLEDGEMENTS

Thanks to Yimei Fu, Yiqian Song, Jiuqiang Tang, and Haochuan Liu for their contributions to the recording sessions. Thanks to Mats Küssner for his generous comments.

8. REFERENCES

[1] J. Arcos, R. Mántaras, de López, & X. Serra. SaxEx: A case-based reasoning system for generating expressive performances. Journal of New Music Research, 27, 1998, 194–210.

[2] J. Bloch, & R. Dannenberg. Real-Time Accompaniment of Polyphonic Keyboard Performance. Proceedings of the International Computer Music Conference, 1985, 279-290.

[3] B. Boots. Spectral Approaches to Learning Predictive Representations (No. CMU-ML-12-108). 2012. CARNEGIE-MELLON UNIV, SCHOOL OF COMPUTER SCIENCE.

[4] B. Boots, S. Siddiqi, & G. J. Gordon. Closing the learning-planning loop with predictive state representations. The International Journal of Robotics Research, 30(7), 2011. 954-966.

[5] B. Boots,, & G. Gordon. An Online Spectral Learning Algorithm for Partially Observable Nonlinear Dynamical Systems. In AAAI. 2011.

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

[7] A. Cont. Realtime Audio to Score Alignment for Polyphonic Music Instruments Using Sparse Non-negative constraints and Hierarchical HMMs. IEEE International Conference on Acoustics and Speech Signal Processing (ICASSP), 2006.

[8] R. Dannenberg. An On-Line Algorithm for Real-Time Accompaniment. Proceedings of the International Computer Music Conference, 1984, 193-198.

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

[10] S. Flossmann, M. Grachten, and G. Widmer. Expressive performance rendering with probabilistic models, in Guide to Computing for Expressive Music Performance, A. Kirke and E. Miranda, Eds. Springer, 2013 75–98.

[11] A. Friberg. Generative rules for music performance. Computer Music Journal, 15, 1991, 56–71.

[12] A. Friberg, L. Frydén, L. Bodin, & J. Sundberg. Performance rules for computer-controlled contemporary keyboard music. Computer Music Journal, 15,1991,49–55.

[13] J. Friedman, T. Hastie, & R. Tibshirani. A note on the group lasso and a sparse group lasso. 2010. arXiv preprint arXiv:1001.0736.

[14] G. Grindlay, & D. Helmbold. Modeling, analyzing, and synthesizing expressive piano performance with graphical models. Machine learning, 65(2-3), 2006, 361-387.

[15] L. Grubb, & R. Dannenberg. Automated Accompaniment of Musical Ensembles. Proceedings of the Twelfth National Conference on Artificial Intelligence, AAAI, 1994, 94-99.

[16] L. Grubb, and R. Dannenberg. A Stochastic Method of Tracking a Vocal Performer. Proceedings of the International Computer Music Conference, 1997, 301-308.

[17] A. Kirke, and E. R. Miranda. A Survey of Computer Systems for Expressive Music Performance. *ACM Surveys* 42(1): Article 3. 2009.

[18] E. W. Large, & C. Palmer. Perceiving temporal regularity in music. Cognitive Science, 26, 2002, 1–37.

[19] C. Raphael. Music Plus One and Machine Learning Machine Learning, Proceedings of the Twenty-Seventh International Conference on Machine Learning, ICML. 2010.

[20] J. Sundberg, A. Askenfelt, and L. Fryden. Musical performance. A synthesis-by-rule approach. Computer Music Journal. 7, 1983, 37–43.

[21] J. Sundberg, A. Friberg, & R. Bresin. Attempts to reproduce a pianist's expressive timing with Director Musices performance rules. Journal of New Music Research, 32, 2003, 317–325.

[22] A. Tobudic, & G. Widmer. Relational IBL in music with a new structural similarity measure. In: T. Horváth, & A. Yamamoto (Eds.), Proceedings of the 13th International Conference on Inductive Logic Programming (ILP'03), Szeged, Hungary (pp. 365–382). Berlin: Springer. 2003.

[23] N. Todd. A model of expressive timing in tonal music. Music Perception, 3, 1985, 33–58.

[24] N. Todd. Towards a cognitive theory of expression: The performance and perception of Rubato. Contemporary Music Review, 4, 1989, 405–416.

[25] P. Van Overschee, & B. De Moor. Subspace identification for linear systems: Theory, implementation, and methods. 1996.

[26] B. Vercoe. The Synthetic Performer in the Context of Live Performance. Proceedings of the International Computer Music Conference, 1984, 199-200.

[27] G. Widmer. Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries. Artificial Intelligence, 146, 2003,129–148.

[28] G. Widmer. Machine discoveries: A few simple, robust local expression principles. Journal of New Music Research, 31, 2002, 37–50.

[29] G. Widmer, and W. Goebl. Computational models of expressive music performance: The state of the art, Journal of New Music Research, 2004, 203 -216.

[30] G. Widmer, & A. Tobudic. Playing Mozart by analogy: Learning multi-level timing and dynamics strategies. Journal of New Music Research, 32, 2003, 259–268.