Artificial Intelligence and Real-Time Interactive Improvisation

Belinda Thom
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
http://www.cs.cmu.edu/~bthom
bthom@cs.cmu.edu

Introduction

My long-term goal is to interactively improvise with the computer, using it as a tool when I practice at home alone in order to help me capture and experiment with my own spontaneous musical ideas. Towards this end, I am building Band-OUT-of-a-Box (BoB), a software system that interacts with a live, improvising (monophonic) musician in the jazz/blues setting. My goal is for BoB to provide personalized improvisational companionship to a specific musician, trading "musically-appropriate" short improvised solos on top of a fixed harmonic background. Spontaneity and personalization make it *crucial* that the system automatically configures itself, *learning* its aesthetic musical sense from its user's improvisational example. As such, my focus is on developing techniques that enable BoB to perceive and generate solos in a musically-appropriate, musician-specific manner. Here I will focus on the perceptual part, which provides the abstraction needed to guide the musically-intentful generation discussed in (Thom 2000b).

In addition to providing a novel and interesting testbed for synthesizing machine learning (ML) and computer-music techniques, I claim that a subtle change in emphasis occurs when focusing on personalized improvisational companionship. This shift is important because it urges us to rethink some of the basic ways we have traditionally applied ML techniques to "music-understanding" (MU) tasks, namely the analysis and algorithmic composition of melody. My intent here is to convince you that this claim is reasonable by illustrating the elusive nature of this domain, which motivates the need for a new ML/MU-based approach. My experience developing techniques for implementing musically-appropriate perception in BoB are then described, followed by some concrete examples that demonstrate how this new approach perceives Bebop saxophonist Charlie Parker.

Personalized Improvisational Companionship

With personalized improvisational companionship, success depends upon the system's ability to get to know you and your *current* musical mood, using this knowledge to listen, interact, and adapt to you in the musical conversation that is evolving. As noted by drummer Keith Copeland, in the ideal interaction:

> I can answer him halfway through his phrase [...] He'll bring you to the point where you can actually sing what he's going to play next, and then, instead of playing that, he'll play something against it which complements what you're singing in your head. (Berliner 1994)

In pursuing this goal on the computer, the observations of practicing improvisors compel my thinking and beg to be addressed. Below is a brief sketch; the reader is left to draw their own conclusions.

1. As guitarist and experimental improvisor Derek Bailey notes: "There is no general widely held theory of improvisation and I would have thought it self-evident that improvisation has no existence outside of its practice" (Bailey 1992). Thus, musical specificity, in contrast to musical generality, is imperative.
2. Notes jazz saxophonist Steve Lacy: "The difference between composition and improvisation is that in composition you have all the time you want to decide what to say in fifteen seconds, while in improvisation you have fifteen seconds." This statement, while obvious, takes on a new meaning when rephrased — since improvisation is not composed, what does it mean to *author* it? Because many interactive computer music systems ultimately expect the composer/musician/programmer to configure them with an appropriate musical aesthetic (e.g., (Rowe 1993) (Dannenberg 1993)), those approaches are less useful here.
3. It is glaringly obvious that one cannot expect the improvisor to *operationalize* (i.e., write a computer algorithm that does) what they do. For example, consider Jazzist Ronnie Scott's frustratingly vague cognitive construction of his musical personality: "I would like ideally to express my, I don't know, personality or whatever, musically, to

the limits of my ability. [... to ...] play in such a way that it would be recognizable as me, and it would express something to people about the way I feel about things" (Bailey 1992). Bailey cuts to the chase: "... the improvisors I spoke to, in almost all cases, did not find any sort of technical description adequate. It wasn't that they weren't interested in technical matters. They just did not find them suitable for illuminating improvisation. They finally chose to describe it in so-called 'abstract' terms. And it became clear that, whatever its deficiencies, this is the best method available" (Bailey 1992).

In spite of the computational difficulties these observations raise, an assumption commonly made in the the arts-and-entertainment community, as exemplified by work in believable agents (Mateas 1999), gives one hope (Thom 2000a). We can assume the musician — provided a modicum of musical competence and adaptation — is willing to "suspend their disbelief" that the computer is simply "a stupid computer." This observation certainly applies when I use the *non-interactive* commercial accompaniment program, *Band-in-a-Box* (PGMusic 1999).

Musically Appropriate Perception

In the context of BoB, I define musically-appropriate perception to be:

> the transformation of a given bar of a solo (lick, phrase, motive, ...)[1] into one of a finite number of classes (components, clusters, etc.).

In other words, perception *is* clustering, where the goal is for each class to reflect one of the musician's different playing-modes (bag-of-licks, moods, aesthetic affects, etc.).

In addition to classifying solos, I require that musically-appropriate perception:

1. **Is based on a probabilistic model.** As McLaughlin warns, ad hoc clustering functions,[2] lacking a statistical basis, are not a priori believable (McLachlan & Basford 1988). Probabilistic models also provide important musical skills: perception and per-class generation of novel samples, and a metric for musical surprise (how unlikely is a sample?).
2. **Is circular.** Define the musical meaning (class) of a solo in terms of what the *musician* does (rather than in terms of what some theory purports).
3. **Accommodates musical common-sense.** Provide access to details regarding things that humans perceive as inappropriate when violated (melodies should contain a local tonal sense, not jump around too much, etc.).
4. **Defines similarity in terms of average local trends** (versus individual notes). Avoid building in predefined biases concerning what pitch-classes (intervals, etc.) are more "alike" — let the solos themselves determine such things.

[1] The syntax "x (y,z,...)" indicates that where "x" is used, one could equally well use "y" or "z".

[2] For example, methods using "edit-based" similarity heuristics, e.g.,(Rolland & Ganascia 1996) (Hörnel & Ragg 1996).

Because we do not know which solos belong to what classes (or even how many modes-of-playing the musician has), to learn this classification function *from the musician's improvised examples* is an *unsupervised learning* problem. The goal is not to "best predict some target function" but to "best explain the musical features that are contained in the examples," letting the solos determine what is "similar." Because this learning scenario is musician-specific, circularity is given. What remains is for the classifier to internalize common-sense musical features, which is representation dependent. With an adequate representation, I believe that the perception that emerges from this definition will be within the musician's own context and, with respect to them, musically-appropriate.

Machine Learning and Music Understanding

Because we transcribe and analyze music in symbolic terms, it seems natural to use knowledge-based methods to solve "music-understanding" (MU) based tasks (namely, melodic analysis and algorithmic composition). However, *creating* melody, as simple an act as improvised humming, is fundamental — as much about the "ingenious" use of exceptions as rules (Loy 1991) (Papadopoulos & Wiggins 1999). How does one engineer this creativity in situations where the immediate, unknown environment determines what is appropriate? ML-techniques *infer* musical knowledge about the world from a set of examples (training set), offering an alternative to human-engineered approaches. However, machine learning is not a panacea: the performance obtained depends entirely upon the representation and the learning algorithm.

Typical ML/MU-based systems rely on one or more of the following scenarios:

1. Supervised learning is used, the concept (class) to be learned explicitly provided in the training set (e.g., (Dannenberg, Thom, & Watson 1997) (Biles 1994)). Often supervised learning applies because musical sequences are recast as *prediction problems.* For example: predict the next note (or few notes) given some context (history) (e.g., (H. Hild & Menzel 1991) (Feulner & Hörnel 1994) (Thom 1995) (Widmer 1996) (Reis 1999)).
2. Base learning on a large musical corpus (sets of jazz tunes, J.S. Bach chorales, etc.); of the citations in Item 1, only (Dannenberg, Thom, & Watson 1997) and (Biles 1994) avoid this.
3. Explicitly build higher-level features into the representation. For instance, (Feulner & Hörnel 1994), (Hörnel & Menzel 1998), and (Rolland & Ganascia 1996) describe features related to scalar/harmonic functionality.

Unfortunately, I do not believe these approaches map to understanding a particular improvisor in a particular situation at a particular moment.

Consider Item 1. Suppose we tried to learn a mapping from bar-of-notes into pitch-class-of-the-next-note-to-be-played.
During training, what cost[3] should we assign to predictions that do not match their target values in the training set? Typically, zero-one cost (sum-of-squared-errors) is used, which treats all pitch-class values except the target value as equally wrong. However, improvisors will sensibly tell you that the appropriateness of a note is not all-or-nothing and depends on many things: the notes preceding/following it; the harmonic and melodic context; the evolving trends in musical surprise; etc. As discussed in (Hörnel & Menzel 1998), we can address this problem with a probabilistic model: penalize according to how confidently the classifier misclassifies an example (cross-entropy-error (Bishop 1995)). Although mistakes are not all-or-nothing, we are still modelling the *isolated* next note as entirely dependent upon the previous bar, while melodic improvisors tend to think in terms of *localized chunks* (licks, phrases, motives, ...).

Ultimately, this line of thinking leads me to question the suitability of using strings-of-notes-based prediction to simulate the creative behavior required in personalized improvisational companionship. With interactive improvisation, the goal is to "listen to me, but not to play my stuff back at me"[4] — to do something a little bit different, which can range from transforming a situation into something unexpected to bringing it back to the norm. Novelty, outliers, and average behaviors are equally important, yet prediction-based paradigms explicitly attempt to memorize their training sequences, and failing that, estimate sequences that best describe the data's average sequential behavior. While all learning schemes attempt to balance generalization and memorization needs, I fear that "predicting-the-next-thing" contains a more myopic, anti-creative bias than a less rigid approach might.

Now consider Item 2. The main reason to train with a larger corpus is to minimize the chance of over-fitting, which results when there is not enough data upon which to base inference.[5] While this approach has been useful in learning to predict if something is in the style of so-and-so (e.g., J.S. Bach (Hörnel & Menzel 1998)), it is inappropriate here. What "corpus" should we use? Seasoned improvisors report that musicians play very differently at different times — What did they eat for dinner? Who are they listening to these days? etc. The only thing that makes sense to train upon is the user's *recent* improvisations, obtained via a preliminary "warmup" session (as was done in (Dannenberg, Thom, & Watson 1997)). While this training set captures the musician's current state-of-mind, its size is limited. Nevertheless, when context is properly localized, improvement in harmonic understanding has been reported (e.g., training on a single song (Thom 1995) (Widmer 1996)); it is reasonable to expect similar results with melodic improvisation.

[3]Learning algorithms minimize cost; this directly effects inference.

[4]Jazzist John McNeil (Berliner 1994).

[5]Rather than generalizing, algorithms with too little data tend to memorize their training sets.

Finally, consider Item 3. A system's ability to generalize is often improved by including pertinent higher-level features, because the system no longer needs to learn them. When these features rely on an "expert" human, as in the case of determining what chords were played during a particular jazz solo, I question their usefulness in this domain. Consider (Hörnel *et al.* 1999), notable in its attempt to standardize the evaluation of two different improvisational learning systems. While the improvised skills that were learned are impressive, each system knew what chords were transcribed, a non-trivial task given that: chord substitutions abound; roots are often missing; color-tones are common; etc. Add to this the fact that transcriptions are often inaccurate, and, as one seasoned improvisor said with respect to the chords in Parker's *Omnibook*, "we always used to change those."

Band-OUT-of-a-Box: a New Approach

A common thread in these ML critiques concerns what it means to *generalize* when *creating* an improvised solo. For example, in BoB, an ideal solo response generator should be able to:

- Identify some trend in several "related" musical fragments (approach memorization: use few datapoints).
- Extend this "trend" in some novel way (generalizing, but not really, because the data may not support such a leap).

To my knowledge, current ML/MU approaches do not provide this type of flexibility or functionality. To address this shortcoming, I propose to learn perception via an *explicit* probabilistic model, one that infers its *own* notions of similarity by converting trajectories of individual notes into explicit (but trivially calculable) higher-level features that summarize various viewpoints of a melody's local surface structure.

As detailed in (Thom 1999), BoB's novel perception scheme proceeds as follows. For each *bar*:

1. **Convert the bar of a solo into a tree.** The tree's internal structure encodes rhythm; an in-order walk of the leaves encodes the pitch-sequence. While this tree is suitable for encoding rhythm — it makes sense to create a novel, yet similar, rhythm by growing and/or collapsing various parts of a tree — the encoding of pitch is nothing more than a glorified string-of-notes, which motivates the need for the next item below.
2. **Calculate summary views of the pitch-sequence.** Namely, three histograms are calculated, recording the usage of pitch-classes, intervals, and changes in directional motion.
3. **Determine to which class each histogram is most likely to belong and then assign it to the most likely one.** There is a separate clustering scheme for pitch-class, intervals, and direction; a bar's overall perception is thus the *combination* of all three clusterings.

Examples From Cluster I

Examples From Cluster II

Examples From Cluster III

Table 1: Examples from Charlie Parker's *Mohawk*

Because BoB's probabilistic clustering model is learned, this last step provides musician-specific abstraction (generalization). Specifically, BoB's training procedure involves:

1. Calculating the histograms of the musician's "warmup" session (training set).
2. For each histogram type, using a probabilistic mixture-of-multinomials model (Thom 2000c) to cluster them *according to their tendencies to prefer certain histogram-bins.*

In this scheme, learning amounts to:

- Inferring the parameters of C different discrete probability distributions (C playing-modes).
- Classifying the histograms according to how likely they were to have been generated by a particular mode's distribution.

The parametric inference bullet is most important because it enables BoB to quantify musical surprise (likelihood estimation), and generate novel per-cluster samples (abstract-driven generation).

In this learning scheme, complexity is minimized because many parameters are assumed to be independent, e.g., pitch-classes and intervals are not correlated; histogram bin-values are treated nominally; etc. Furthermore, because learning is based upon an explicit probabilistic model — versus a black-box style approach (e.g. neural nets) — it is reasonable to assume that training will be more efficient (both data and computation-wise), and that the model will be easier to understand.[6] This model also facilitates the application of musically reasonable priors (Thom 2000c), a technique which combats over-fitting. Finally, by using these multiple views of melodic structure, we embed a level of temporal knowledge into the representation, which is needed in order to generate musically sensible *temporal* pitch-sequences from these histogram-based clusters (Thom 2000b).

This method of perception, when applied to Charlie Parker's *Mohawk* improvisations, produces impressive musician-specific abstractions (Thom 1999) (Thom 2000c). For example, the pitch-class clusterings imply musically reasonable scale usage. Four different examples from three different inferred clusters are shown in Table 1 in order to demonstrate the types of generalizations that emerge. Analyzing Cluster I's parameters reveals a preference for pitch-classes in the B$^\flat$-Major-Bop scale, intervals in the m2 to m3 range (with an occasional P4, P5, m6, or M6), and runs of upwards/downwards motion (upwards being more common). While the most likely pitches are B$^\flat$ and E$^\flat$, the smaller G$^\flat$ probability (the scale's m6) has high discriminating power. Similarly, Cluster II corresponds to the E$^\flat$-Bop7 scale, the same set of intervals, and runs such that downwards motion is more common; the most likely pitches are B$^\flat$ and C; the smaller A$^\flat$ is distinctive. Cluster III corresponds to the E$^\flat$-Major scale, prefers the unison-interval[7] and intervals in the m3 to P4 range (the lack of m2 is distinctive). In Cluster III, the amount of upwards/downwards motion ensures that long runs are uncommon; the most likely pitches are D and F.

Listening to these examples is most impressive. While in some ways they are quite different (especially with respect

[6] And easier to extend with online adaptation.

[7] Trees are constructed so that syncopated rhythms contain unison-intervals.

to "edit-based" heuristics), in other ways they just seem to fit together (which is not surprising given BoB's multiple-viewed representation scheme). When experimenting with concatenating bars from the different clusters together, I found it encouraging how easily they tend to combine — one could almost use them as distinct outcomes in Mozart's dice-rolling compositional games.

It is important to understand the role that histogram-sparsity plays in BoB's generalizations. For example, part of what allows these examples to look "quite different," yet "belong to the same cluster," is that histogram dimensions are relatively large when compared to their sample-size (e.g., 12 pitch-class values versus ≈ 12.2 notes-per-bar!). In short, we cannot expect a particular histogram to contain enough information by itself to unequivocally determine its generative parameters; that per-class examples will prefer different important (i.e., highly probable) bin values, and hence look fairly different, is the result. To state this another way, consider the possibility that part of what makes a musician's response appear "creative" is the fact that, when they are using one of their modes, they only reveal certain parts of its essence in a localized context.

I have already outlined some of the ways in which BoB's approach deviates from other ML/MU approaches. However, it is also important to mention the common ground that is shared. For example, as discussed in (Hörnel *et al.* 1999) (Hörnel & Menzel 1998) and (Höthker 1999), lack of "motivation" in computer-generated music has led to a focus on building hierarchical music models, so that more abstract levels can guide global structure while lower levels can generate specific note-strings. Although I only have space to briefly mention this aspect of BoB's design, its per-bar perception does provide the functionality needed for such a hierarchy by providing crucial musical skills: 1) abstract-driven per-bar solo generation; 2) the ability to reason at multiple time scales, 3) the ability to predict abstract temporal trends; etc.

Conclusion

In this paper I have introduced the domain of personalized improvisational companionship, identifying ways in which this task urges us to seek out new machine learning music-understanding paradigms. Broadly, I have focused on: 1) using unsupervised learning methods that are based upon probabilistic models so that the data can affect what is similar with respect to its discrete music features; 2) combining multiple viewpoints of melodic surface, looking at local averages rather than strings-of-notes; and 3) inferring musically useful abstractions from smaller (more localized, situation-specific) training sets. I have also presented Band-OUT-of-a-Box (BoB), a system in which these ideas are being implemented, along with concrete examples that demonstrate the power and subtlety of this approach in perceiving Charlie Parker's *Mohawk* solos.

References

Bailey, D. 1992. *Improvisation, Its Nature & Practice in Music*. Da Capo Press.

Berliner, P. F. 1994. *Thinking in Jazz, The Infinite Art of Improvisation*. University of Chicago Press.

Biles, J. 1994. Genjam: A genetic algorithm for generating jazz solos. In *Proceedings of the 1997 ICMC*. International Computer Music Association.

Bishop, C. M. 1995. *Neural Networks for Pattern Recognition*. Clarendon Press.

Dannenberg, R. B.; Thom, B.; and Watson, D. 1997. A machine learning approach to musical style recognition. In *Proceedings of the 1997 ICMC*. International Computer Music Association.

Dannenberg, R. B. 1993. Software design for interactive multimedia performance. *Interface - Journal of New Music Research* 22(3):213–218.

Feulner, J., and Hörnel, D. 1994. Melonet: Neural networks that learn harmony-based melodic variations. In *Proceedings of the 1994 ICMC*. International Computer Music Association.

H. Hild, J. F., and Menzel, W. 1991. Harmonet: A neural net for harmonizing chorales in the style of j.s. bach. In R. P. Lippmann, J. E. M., and Touretzky, D. S., eds., *Advances in Neural Information Processing 4*, 267–274. Morgan Kaufman.

Hörnel, D., and Menzel, W. 1998. Learning musical structure & style with neural networks. *Computer Music Journal* 22(4):44–62.

Hörnel, D., and Ragg, T. 1996. Learning musical structure and style by recognition, prediction and evoloution. In *Proceedings of the 1996 ICMC*. International Computer Music Association.

Hörnel, D.; Langnickel, J.; Sandberger, B.; and Sieling, B. 1999. Statistical vs. connectionist models of bebop improvisation. In *Proceedings of the 1999 ICMC*. International Computer Music Association.

Höthker, K. 1999. Modelling motivic processes of melodies with markov chains. In *Proceedings of the 1999 ICMC*. International Computer Music Association.

Loy, G. 1991. Connectionism and musiconomy. In Todd, P. M., and Loy, D. G., eds., *Music and Connectionism*. MIT Press. 23–36.

Mateas, M. 1999. An oz-centric review of interactive drama & believable agents. In Wooldridge, M. J., and Veloso, M. M., eds., *Lecture Notes in Artificial Intelligence*, volume 1600. Springer-Verlag. 297–329.

McLachlan, G. J., and Basford, K. E. 1988. *Mixture Models: Inference & Applications to Clustering*. Marcel Dekker.

Papadopoulos, G., and Wiggins, G. 1999. Ai methods for algorithmic composition: A survey, a critical view and future prospects. In *Proceedings from the AISB'99 Symposium on Musical Creativity*.

PGMusic. 1999. Band-in-a-box. http://www.pgmusic.com.

Reis, B. Y. 1999. Simulating music learning: On-line perceputally guided pattern induction of contex models for multiple-horizon prediction of melodies. In *Proceedings from the AISB'99 Symposium on Musical Creativity*.

Rolland, P., and Ganascia, J. 1996. Automated motive-oriented analysis of musical corpuses: a jazz case study. In *Proceedings of the 1996 ICMC*. International Computer Music Association.

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

Thom, B. 1995. Predicting chords in jazz: the good, the bad & the ugly. In *Proceedings from the IJCAI 95 Music and AI Workshop*.

Thom, B. 1999. Learning melodic models for interactive melodic improvisation. In *Proceedings of the 1999 ICMC*. International Computer Music Association.

Thom, B. 2000a. Bob: an interactive improvisational music companion. In *Proceedings of the Fourth International Conference on Autonomous Agents*.

Thom, B. 2000b. Generating musician-specific melodic improvisational response in real-time. Submitted to the International Computer Music Conference.

Thom, B. 2000c. Unsupervised learning and interactive jazz/blues improvisation. In *Proceedings of the AAAI-2000*. AAAI Press.

Widmer, G. 1996. Recognition & exploitation of contextual clues via incremental meta-learning. In *Proceedings of the 1996 ICML*.