Unsupervised Learning and Interactive Jazz/Blues Improvisation

Belinda Thom
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15207 USA
http://www.cs.cmu.edu/~bthom
bthom@cs.cmu.edu

Abstract

We present a new domain for unsupervised learning: automatically customizing the computer to a *specific* melodic performer by merely listening to them improvise. We also describe BoB, a system that trades customized real-time solos with a specific musician. We develop a probabilistic mixture model, derived from the multinomial distribution, for the clustering and generation of variable sample-sized histograms. With this model, bars of a solo are clustered via the pitch-classes contained therein, adding a new dimension to the problem: the need to learn from sparse histograms. With synthetic data, we quantify the feasibility of handling this issue, and qualitatively demonstrate that our approach discovers powerful musical abstractions when trained on saxophonist Charlie Parker.

Introduction

This research addresses the problem of the computer interacting with a live, improvising musician in real-time. Although a number of interactive improvisational systems have already been built, (Rowe 1993), (Pennycook & Stammen 1993), (Dannenberg & Bates 1995) and (Walker 1997), these works place the burden of implementing "musically-appropriate" behavior upon the musician/composer/programmer. Rather, we have developed a model of improvisation that automatically customizes itself to its user by first listening to them improvise (warmup), and then probabilistically clustering localized segments (**bars**[1]) of their solos in such a way that the warmup data appears maximally likely.

While the long-term goal is to build an agent that is *fun* to trade solos with, the immediate goal is to develop a method that operationalizes (creates a computer algorithm that does) what the musician does. An an improvisor's insight into what makes their behavior musically-appropriate is necessarily non-technical, vague, and abstract (Thom 2000b); in our approach, notions of musical-appropriateness are inferred (learned) by fitting a probabilistic model to the melodic structure found in the user's warmup bars.

Each bar of a warmup is transformed into a **pitch-class** histogram (PCH), which ignores the temporal ordering of a bar's note-sequence, instead focusing on the musician's preference for certain pitch types. We assume that each PCH corresponds to one of the improvisor's C "modes-of-playing," each mode being one of the distinct ways in which the user employs and prefers certain musical **scales**. Under certain assumptions, these histograms can be modeled as a mixture of C multinomials, provided the multinomial components are extended to handle a variable number of sample-sizes.

Existing multinomial mixture model based learners handle at most one count, e.g., *AutoClass* (Hanson, Stutz, & Cheeseman 1991), or one-count-per-bin (Meila & Heckerman 1998). We develop a *variable-sized*-multinomial-mixture model (vMn) in order to handle arbitrary numbers of counts. A new variant of the expectation-maximization algorithm (EM) is derived in order to estimate our model's parameters, using a maximum-a-posteriori (MAP) approach.

This *model-based* approach to clustering is important because it provides essential musical skills: 1) abstract perception (what cluster is a bar in?); 2) abstract generation (sample another bar from some cluster); and 3) an estimation of musical surprise (how likely is a bar?). This approach also gives us a degree of statistical confidence that ad hoc methods — e.g., the melodic clustering methods of (Rolland & Ganascia 1996) or (Hörnel & Ragg 1996), which rely on edit-based distance heuristics — lack (McLachlan & Basford 1988).

While in this paper, learning focuses on a solo's simplest **tonal** features (pitch-class), the same technical learning issues that arise here also apply when learning with a more complete representation. For example, (Thom 1999) also considers intervallic and contour-based musical features; by converting these sequences into per-bar histograms, analogous models are learnable.

By fitting the vMn model to the user's training data, we get abstract, *musician-specific* perception: a bar is perceived via what mode (class, cluster, ...) is most likely to have generated it, which in turn *depends upon the user's warmup data.* It necessarily follows that new PCH samples taken from this vMn model (generation) are musician-specific. As outlined in (Thom 2000b), this model provides the skills needed for real-time musician *and* solo specific response, provided PCH samples can be transformed back into musically-appropriate note-sequences. This transformation is addressed in (Thom 2000c); that appropriate note-

[1] Bold-face terms are musical; see Appendix A.

sequences can be derived from order-ignorant histograms is due to the fact that in the complete representation scheme, histograms embed more temporal knowledge (intervals depend on successive note pairs; contours upon note strings).

In this paper, we describe our solo trading system and its learning scenario. We introduce the vMn model and an EM-based method for fitting its parameters to the training data. We discuss the unique challenge that arises in this domain. Specifically, training sets are small (≈ 120 histograms), and each histogram is relatively sparse (while there are 12 discrete and nominal pitch-class values, the expected number of counts per histogram is ≈ 12.2). This research addresses not only *how* to learn a vMn model, but *what* we can learn from challengingly sparse datasets. The "what" question is investigated by: 1) quantitatively evaluating the performance of synthetic datasets; and 2) qualitatively evaluating the musical-appropriateness of the pitch-class modes that are learned for Bebop saxaphonist Charlie Parker.

In addition to presenting a novel domain and demonstrating that powerful musical abstractions emerge with this vMn approach, our work is significant because it empirically demonstrates that probabilistic clustering of sparse, unlabelled histograms is useful. This success is in part due to the fact that we take advantage of the knowledge that per histogram, *all counts are generated by the same component,* whereas "one count" style approaches treat each as independent of the others. Histograms with larger sample-sizes have more information with which to distinguish themselves — it makes sense to directly incorporate this knowledge into the learning process.

Band-out-of-a-Box (BoB)

We are building Band-OUT-of-a-Box (BoB), an interactive soloist that trades bars of a customized solo with a single musician (Thom 2000b). We now describe how BoB's melodic representation is used to configure itself to the user.

The Domain

BoB is specifically designed for the following scenario. Each time the musician wants to trade solos, they first warmup (for ≈ 10 minutes), improvising freely over a desired song at a fixed tempo. During this time, BoB collects training data by recording their note-stream in real-time. Next, BoB goes offline, creating training set $X =< x_1, \ldots, x_i, \ldots, x_n >$ by: 1) segmenting the note-stream into bars; 2) building one PCH per bar. (Note our use of upper and lower-case letters to distinguish between data points and sets). For PCH x_i, an $m = 12$ dimensional histogram, x_{ij} is the number of times that pitch-class j occurs in bar i. $sz_i = \sum_{j=1}^{m} x_{ij}$ is the total sample-size.

BoB uses a stability/usefulness heuristic to *estimate* how many playing modes, $\hat{C}$, are present in the training data, which in turn, allows a $\hat{C}$-component vMM parameterization, $\hat{\Omega}$, to be estimated so that X appears maximally likely.[2] When BoB goes back online, trading bars of solo with the musician in real-time, $\hat{\Omega}$ is used to abstractly perceive and generate new PCHs with: 1) situation and musician based specificity; and 2) per-bar real-time response.

[2]The hat superscript denotes an estimate.

Representational Issues

Solo segmentation is per bar. While this small, *fixed* time window (≈ 2 [sec]) affords fine-grained responsiveness, it means that PCHs are sparse; it is also the reason that sample-size varies.

Offline learning must occur quickly; small training sets are desirable. Also, musically, local context is crucial (Thom 2000a); one cannot assume that combining multiple warmup sessions necessarily yields better customization.

Unsupervised Learning

We now introduce our probabilistic model, $vMn(\Omega)$, a mixture of C *variable-sized* multinomials. Parametric inference is non-trivial when training data is both observable (histogram bin counts) and unobservable (which component generated what histograms). Learning amounts to segregating the histograms according to those bins that are most heavily used (or vacant).

Example

A subset (21 histograms) of sparse, simulated dataset II (described later) is shown in Figure 1. Each subplot is a different histogram. Bin counts are stacked wire boxes. The gray columns reflect the shape of a histogram's generative probabilities, indicating which bins are highly probable, or *important*.

This dataset, generated by seven components, is displayed so that *each column of subplots has the same generator.* Thus, x_1 is generated by the same component as x_8 and x_{15}. As x_1's first bin has three counts, and this bin is gray, we know these counts make x_1 more likely. "$\times$" marked subplots are less likely, so much so that even when the generative model's parameters are known, if we have to *guess* these histogram generators, we would guess incorrectly.

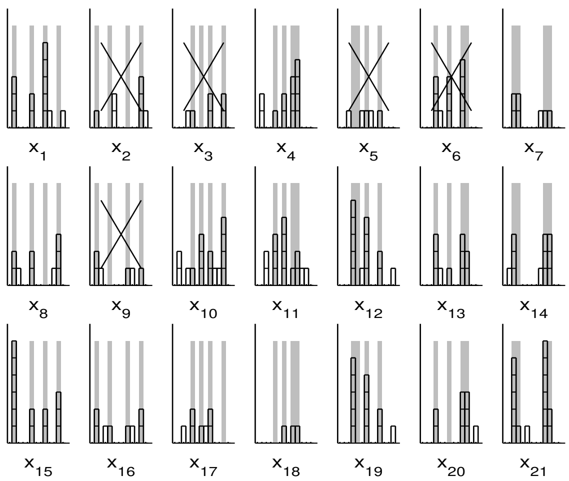

Figure 1: A subset of dataset II

The purpose of this example is to illustrate that learning is *not* trivial. Even knowing the column-arrangement, (which is equivalent to having labels), when the gray columns are hidden, it is not obvious which bins of each component are most important. This figure also illustrates some subtle issues related to sparsity. For example, while smaller than average sample-sizes ($sz_i \leq 8$) coincide with $\frac{4}{5}$ of the $\times$'s (x_2, x_3, x_5, x_9), the smallest sample-size (x_{18}) is not only correctly classified but also has its counts placed on important bins. On the other hand, while x_{10} has the largest sz, $\frac{1}{3}$ of its counts are misleading (reside on non-important bins). Also, in many cases, counts without gray backgrounds do not hurt classification, although they may hurt learning. On example of this is x_8, where non-important counts happen to reside on bins that do not cause other components to appear more likely.

Larger histogram dimensions mean that more bins *could* be important when estimating generators, which means that sample-size may be distributed amongst more bins, less information regarding a particular bin may be available. On the flip side, more dimensions means less chance for competing components to interfere with one another. Regardless, with sparsity, *specific examples are not likely to contain enough information by themselves,* which also *affords more powerful generalization!* A rigorous probabilistic procedure, one that weighs larger sample-sizes more heavily, especially when they boost other data, is needed.

Music Issues

Modelling PCHs with vMn introduces strong assumptions:

1. For each bar, the musician chooses *at random* from a fixed set of probabilities in order to determine what mode (component) to use. Thus, the musician does not switch modes mid-bar, each bar's mode has nothing to do with its neighboring bar's mode, etc.
2. Per bar, mode and the number of notes played are independent. The number of notes in different bars are also independent.
3. For a given a mode, one pitch-class does not depend upon another (feature independence). Melodically, this means that previously improvised notes do not affect future notes (and vice-versa).
4. Pitch-class bins are nominal/discrete. We do not impose any similarity metric upon pitch-class values.
5. Bars that contain more notes contain more information about what mode the musician is realizing (PCHs are *not* normalized).

The first three items deal with independence, reducing the model's complexity, making it less susceptible to overfitting. As (Nigam *et al.* 1998) notes, the focus on decision boundaries often makes a classifier robust to independence violations. Most serious is Item 1; musical bars certainly affect one another — our datasets are not independently sampled. Fortunately, the quality of our music results empirically validates our approach's usefulness.

Item 4 is crucial. While simpler pitch-class similarity schemes have been used to provide learning feedback — e.g., Euclidean, or the $\delta-$based distance metric used to train melody generation in (Feulner & Hörnel 1994) — other context-dependent aspects (for example, scales) are more perceptible musically (Bartlett 1993). Our *nominal* viewpoint allows another mechanism, and, importantly, a customizable one, to determine pitch-class similarity: how likely is it that certain values are preferred in a given mode?

The Multinomial Distribution

The multinomial family, $Mnom(sz, \theta)$, extends the binomial so as to handle more than two discrete, nominal outcomes. Sample-size sz is *constant*; θ is the $r-$dimensional probability vector used to weight outcomes $< 1, \ldots, 2, \ldots, r >$. When sampling from this distribution, $v \sim Mnom(sz, \theta)$, v is an r-dimensional histogram with sz trials distributed amongst r bins. Histogram likelihood is:

$$\Pr(v|sz, \theta) = \prod_{j=1}^{m} \frac{v_j!}{sz!} (\theta_j)^{v_j}.$$

The vMn Distribution

We now define the variable-sized multinomial mixture model, $vMn(\Omega)$, where $\mathrm{ind}(\cdot)$ maps a single-count histogram (vector) into the index of the bin that contains the count (scalar):

$$\begin{aligned} sz &\sim ind(Mnom(1, \eta)) \\ y = ind(z), &\text{ where } z \sim Mnom(1, \pi) \\ x|y &\sim Mnom(sz, \theta_y). \end{aligned}$$

The multinomial components of this model, $\Theta =< \theta_1, \ldots, \theta_c, \ldots, \theta_C >$, control the distribution of counts among bins. Variable sample-size is handled by assuming that a histogram's generator, y, and its sample-size, sz, are independent. Ω refers to parameters Θ, π, and η. $\pi =< \pi_1, \ldots, \pi_c, \ldots, \pi_C >$ controls how often a component is chosen; $\eta =< \eta_{\min(sz)}, \ldots, \eta_{\max(sz)} >$ how often particular sample-sizes occur. The joint likelihood, $< x, y >$, or equivalently $< x, z >$, is:

$$\Pr(x, y|\Omega) = \Pr(sz|\eta)\Pr(y|\pi)\Pr(x|sz, \theta_y).$$

Naive Bayes Classification and vMn

With estimate $\hat{\Omega}$ a Naive-Bayes-Classifier can be built, which maps histograms into one of $\hat{C}$ classes. Bayes-Rule is used to turn the generative model around, so that the *component posteriors,* $\hat{z} =< z_1, \ldots, z_c, \ldots, z_{\hat{C}} >$, can be estimated:

$$z_y = \Pr(y|x, \hat{\Omega}) = \frac{\Pr(y|\hat{\Omega})\Pr(x|y, \hat{\Omega})}{\Pr(x|\hat{\Omega})} = \frac{\hat{\pi}_y \Pr(x|sz, \hat{\theta_y})}{\sum_{c=1}^{C} \hat{\pi}_c \Pr(x|sz, \hat{\theta_c})}.$$

Component estimation (abstract perception) is then:

$$\hat{y} = \mathrm{argmax}_c \left(\Pr(c|x, \hat{\Omega}) \right).$$

$\hat{Y}$ and $\hat{Z}$ are the dataset's estimated generators and posteriors respectively. With synthetic data, we also know Y, the true generative components.

With synthetic data, Ω is known, so we can build an Optimal-Bayes-Classifier, whose component estimation (denoted by superscript "$*$"), is guaranteed to have a minimal expected error rate:

$$err^* = E\left[y \neq y^*\right] \approx \frac{|Y \neq Y^*|}{n} = e^*.$$

The "$\approx$" here indicates our estimation of this expectation. err^* quantifies the *hardness* of parameterization Ω, which depends upon the overlap between component distributions, and is a function of the entire input space. Our approximation, e^*, is based on finite dataset X. When the learning algorithm knows C, hardness can also provide a lower bound with which to compute the additional loss of having to infer Ω:[3]

$$\Delta^{err} = E\left[y \neq \hat{y}\right] - E\left[y \neq y^*\right] \approx \frac{|Y \neq \hat{Y}|}{n} - e^* = \Delta^e.$$

The values of e^* and Δ^e that we report are based on an independent test set.

Unsupervised Learning and vMn

For independent and identically sampled histograms, *average* log-likelihood of *labelled* dataset $< X, Y >$ is:

$$\mathcal{L}_L(X,Y|\hat{\Omega}) = \frac{\log(\Pr(X,Y|\hat{\Omega}))}{n} = \frac{\sum_{i=1}^{n} \log(\Pr(sz_i|\hat{\eta})) + \log(\hat{\pi}_{y_i}) + \log(\Pr(x_i|sz_i,\hat{\theta}_{y_i}))}{n}. \quad (1)$$

For an *unlabelled* dataset, Y and C are hidden. Average log-likelihood is now the joint marginalized with respect to the posterior:

$$\mathcal{L}_U(X|\hat{\Omega}) = \frac{\log(\Pr(X|\hat{\Omega}))}{n} = \frac{\sum_{i=1}^{n} \log(\Pr(sz_i|\hat{\eta})) + \log\left(\sum_{c=1}^{C} z_{ic}\hat{\pi}_c \Pr(x_i|sz_i,\hat{\theta}_c)\right)}{n}. \quad (2)$$

Warmup PCHs are unlabelled; BoB seeks to estimate $\hat{\Omega}$ so that $\mathcal{L}_U$ is maximized. However, $\hat{C}$ must be estimated elsewhere, for to optimize it here is under-constrained.

In vMn, sample-size does *not* affect classification, which is not to say that as sz increases, classification does not become easier. In fact, sz and err^* are inversely related. However, in and of itself, sz provides no information about which component generated a histogram.

With synthetic data, the loss in likelihood associated with having to infer Ω can also be estimated:

$$\Delta^l = l^* - \mathcal{L}_U(X|\hat{\Omega}), \text{ where } l^* = \mathcal{L}_U(X|\Omega)$$

These values are also based on an independent test set.

EM and vMn

Supervised and unsupervised learning both amount to finding an $\hat{\Omega}$ for which their appropriate $\mathcal{L}$ is maximal. It is the log of sums in $\mathcal{L}_U$ that makes its optimization difficult (whereas a closed-form optimum for $\mathcal{L}_L$ exists). $\mathcal{L}_U$'s optimization is non-linear and has multiple roots. We use the EM-algorithm to control the search for local optima (Dempster, Laird, & Rubin 1987). This search is repeated multiple (25) times from different random starting points; the best (most likely) solution is reported.

With each subsequent iteration at time t, EM guarantees that $\mathcal{L}_U(X|\hat{\Omega}^t) \leq \mathcal{L}_U(X|\hat{\Omega}^{t+1})$. In theory, finding a local maximum is equivalent to finding a fixed-point, $\mathcal{L}_U(X|\hat{\Omega}^t) = \mathcal{L}_U(X|\hat{\Omega}^{t+1})$. In practice, a computer's precision requires another form of termination (we stop after 8 digits of improvement and/or 1500 iterations).

EM involves two steps per iteration. The E-step calculates $E[\mathcal{L}_L(X,Z|\hat{\Omega}^t)]$. Indicator vector Z is the only random variable, so this expectation amounts to solving $\hat{Z}^t = E[Z|X,\hat{\Omega}^t]$, whose solution is the posteriors, $\hat{z}_i^t = \Pr(y|x_i,\hat{\Omega}^t)$.

In the M-step, $\hat{Z}^t$ is used in place of Z. Ω is re-estimated according to $\hat{\Omega}^{t+1} = \text{argmax}_v(\mathcal{L}_L(X,\hat{Z}^t|v))$. The solution to this equation is

$$\pi_c^{t+1} = \frac{\frac{1}{C} + \sum_{i=1}^{n} \Pr(z_{ic}|x_i,\hat{\Omega}^t)}{1+n}$$

$$\theta_{cj}^{t+1} = \frac{\frac{1}{m} + \sum_{i=1}^{n} \Pr(z_{ic}|x_i,\hat{\Omega}^t)x_{ij}}{1 + \sum_{i=1}^{n}\sum_{j=1}^{m} \Pr(z_{ic}|x_i,\hat{\Omega}^t)x_{ij}},$$

Both of these estimates are MAP-based: optimal Bayesian parameter estimation, augmented by Laplacean priors on π and θ (Vapnick 1982).

From an improvisational standpoint, the θ priors encode the reasonable musical assumption that no pitch-class is *never* played. The appropriateness of π's priors depends upon the appropriateness of $\hat{C}$; they encode the belief that the musician is never *not* going to use one of their "$\hat{C}$" playing modes.

Estimating C

Estimating C by optimizing $\mathcal{L}_U$ is under-constrained. Rather than adding more constraints, we pick $\hat{C}$ to be the *largest* value that produces a *stable* and *useful* clustering of the training set.

A stable C is defined as one that, when five identical learning experiments are run, produces no disagreements on pair-wise comparisons of solutions. A useful C is defined as one that, on average, is well separated (whose estimated error is below 0.20). Stability quantifies the discrepancies between pairs of solutions' *partitioning* of the training set, shedding light upon how repeatable a learning solution is — how real it is. In BoB, usefulness makes sense: we are most interested in adapting to those playing modes that are markedly different.

Related Work

We know of no other work that empirically validates the feasibility of fully unsupervised vMn learning, or presents learning details for for the fully unsupervised case. While a full-blown vMn style model was used in text-classification (Nigam *et al.* 1998), the focus was on combining labelled and unlabelled data, and sample-size was ignored (histograms were normalized). Fully unsupervised multinomial mixture models usually impose serious restrictions upon the number of samples allowed; in (Hanson, Stutz, & Cheeseman 1991) $sz = 1$. Another common approach, discussed in (Meila & Heckerman 1998), (Hanson, Stutz, & Cheeseman 1991) and (McLachlan & Basford 1988), is to demand that each bin has at most one count. We have developed a new model for this domain because it is important

[3] $\hat{Y}$ must first be permuted to approximate Ω's ordering.

to consider the additional information that larger histograms provide.

Results: Synthetic Data

This domain introduces unique challenges — variable sample-size and sparsity. We now quantify the degree to which learning in such conditions is possible.

In particular, we generate training and test sets with: $C = 7$, $\pi_c = \frac{1}{C}$, $n = 175$ ($\approx$ 25 histograms per cluster), $m = 12$, and $sz \sim Uniform(3, 15)$. On the one hand, we report optimistic results because our learning algorithm knows C. On the other hand, our parameterizations of Ω make learning more difficult than it is likely to be for PCH data. For example, our uniform sz has a larger variance and a smaller mean than Parker's bell-shaped distribution. Also, our components are as "entropic" as possible. Specifically, for each component, four bins are equally *important* (have high probability α of occurring). The other eight bins are equally *non-important* (have low probability β). Each component's important bins are arranged so as to provide maximal interference with the other component's important bins. Thus, discriminating these histograms requires *sets* of features; single features are never sufficient.

We quantified learning performance for three datasets, each harder (having larger e^*) than its predecessors. The values in the right five columns are averaged over 25 experiments. $\Delta^\times$ is the number of misses directly caused by *having* to learn (compare this to $n = 175$ total guesses).

dataset	α	β	l^*	Δ^l	e^*	Δ^e	$\Delta^\times$
I	.248	.001	-9.2	0.18	0.11	0.00	0
II	.21	0.02	-12	0.38	0.22	0.07	12
III	.188	.031	-13	0.40	0.31	0.16	28

While having to learn degrades a classifier's accuracy, and the harder the problem, the worse the degradation, the average cost of having to learn (Δ^e) ranges from negligible to $\frac{1}{2}$ of the optimal (e^*). To add some perspective, consider the worst case, III. We expect to miss 0.47 guesses; an optimal classifier would miss 0.31; a random classifier would miss 0.85. Even with this sparse data (some is shown in Figure 1), learning produces useful results.

Results: Music Data

We now qualitatively argue that our approach produces powerful, customized abstraction when applied to Parker's PCH data. This argument is all the more crucial given that, during solo improvisation, vMn assumptions are likely to be violated.

Our training set was Parker's *Mohawk* improvisations (Goldsen 1978). Figure 2 (left) displays how BoB perceived these solos. Each subsequent row is Parker's next *Mohawk* solo (chorus). The columns are each choruses' bars. The symbol shown in each $< bar, chorus >$ indicates what mode BoB assigned it to. The corresponding mode generator estimates are shown in Figure 2 (right). While $\hat{C} = 4$ and 5 were both stable/useful, solution $\hat{C} = 4$ is presented due to space limitations. Details concerning the **chords**, scales, and melodic styles of the Bebop genre are beyond this paper's scope. The goal here is to merely provide a flavor of the powerful types of musical abstractions that were discussed in (Thom 1999).

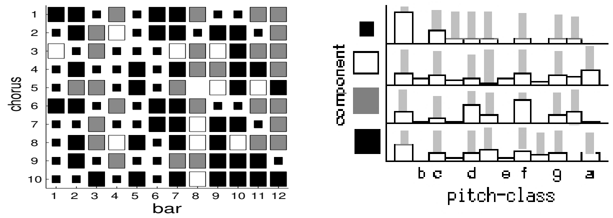

Figure 2: *Mohawk*: Clustering $\hat{Y}$ (left); Parameters $\hat{\theta}_c$ (right)

Several musically appropriate correlations emerge when $\hat{\Omega}$ and $\hat{Y}$ are analyzed. For example, modes ■ and ▪ were most common, accounting for $\frac{80}{119}$ of the bars.[4] The gray backgrounds of these components (right, Figure 2) identify those pitch-classes in the $b^\flat$-Bebop-Major and $e^\flat$-Bebop-Dominant scales respectively. Given the **key** of this song (a 12-bar blues in $b^\flat$), it is appropriate that Parker used these scales, *especially* since they were invented in part to explain his playing style (Baker 1983).

To cite another example, notice the segregation of modes as a function of column (bar), which makes sense given that the chords *change* (and hence the underlying tonal function changes) on bars 1, 5, 7, 9 and 11. A particularly important change occurs in bars 5-6, the only columns in which Parker exclusively uses the Bop-modes. Another case is bars 9-10, the only place were ■ is used more than 50% of the time. Also, bar 12 ends the blues progression. At this point if another chorus is played, it has the musical distinction of being called a "turn-around," whereas if the tune ends, (which is the case in choruses 5 and 10), then a resolution to the tonic ($b^\flat$) is expected. BoB captured Parker's accommodation of these distinctions, using ■ (the $e^\flat$-Major scale) in turn-arounds and restating the tonic via the $b^\flat$-Bop mode.

The component we have not yet discussed, □, exemplifies another powerful aspect of our approach. Notice how all pitch-classes but $e^\flat$ (the bin between d and e) are reasonably probable. While it is musically reasonable to view this component as the $b^\flat$-Major scale (shown in gray), by itself, this viewpoint is limited. For example, with respect to □ we know: 1) how often on average Parker chooses to include non-scalar pitch-classes; 2) that scale-tone $e^\flat$, rather than the non-scalar and musically special "tritone" e, is more surprising; 3) how Parker employs modes in particular contexts (e.g., a choruses' mode-sequence).

In short, a musician's specific improvisational style is *embedded*, or distributed within the learned representation at multiple different and important levels, giving BoB knowledge about what contexts a mode can be used in, what contexts the musician is in, and how to realize a particular mode (i.e., which pitches are most important).

[4]No notes were played in $< 5, 8 >$; this bar was omitted.

Conclusion

We have presented a novel domain for unsupervised learning: automatically customizing the computer to its *specific* melodic performer by listening to them improvise. We also described our system BoB, which performs this task in the context of real-time four bar solo trading. We developed a probabilistic mixture model of variable-sized multinomials and a procedure that uses this model to learn how to perceive/generate variable-sized histograms. Because this model was used to cluster per-bar pitch-class histograms, we added a new dimension to the problem: the need to learn from sparse data, and use synthetic data to show that useful results can be learned in this case. We have also shown that when trained on saxophonist Charlie Parker, powerful musical abstractions emerge on many levels. For example, Parker's playing-modes (scales, or sub/supersets of scales) are distributed among a component parameters, which in turn quantifies the relative importance of various pitch-classes, giving us an idea of *how* to realize a particular mode (something rule/scale-based systems must be told how to do). Parker's contextual employment of mode, which allows one to consider generating *new* like-minded contexts, is also embedded within this learned representation, opening up a whole new range of creative possibilities.

Appendix A: Musical Terms

Term	Definition
bar	the grouping caused by the regular recurrence of accented beats, typically 4 taps of a listener's foot (e.g., ≈ 2.5 seconds)
pitch-class (PC)	a pitch's tone, that piano key on the single octave piano that would be played if octave was ignored: $PC \in \{c, d^\flat, d, ..., b^\flat, b\}$
octave	two pitches with the same PC and 1:2 frequency ratio
scale	a subset of PC's differing in pitch according to a specific scheme
tonal	the principle of **key** in music conveyed through the family relationship of all its tones and chords, (one affect of this is your being able to sing *doh-re-mi-fa-...* on top of a given harmony/melody)
chord	combination of pitches simultaneously performed, producing more or less perfect harmony

References

Baker, D. 1983. *Jazz improvisation : A Comprehensive Method of Study For All Players*. Frangipani Press.

Bartlett, J. C. 1993. Tonal structure of melodies. In Tighe, T. J., and Dowling, W. J., eds., *Psychology & Music: the Understanding of Melody & Rhythm*. Lawrence Erlbaum Associates, Inc.

Dannenberg, R. B., and Bates, J. 1995. A model for interactive art. In *Proceedings of the Fifth Biennial Symposium for Arts & Technology*, 103–111.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1987. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*.

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.

Goldsen, M. H., ed. 1978. *Charlie Parker Omnibook: For C Instruments*. Atlantic Music Corp.

Hanson, R.; Stutz, J.; and Cheeseman, P. 1991. Bayesian classification theory. Technical Report FIA-90-12-7-01, NASA Ames Research Center.

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.

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

Meila, M., and Heckerman, D. 1998. An experimental comparison of several clustering and initialization methods. Technical Report MSR-TR-98-06, Microsoft Research Center.

Nigam, K.; McCallum, A.; Thrun, S.; and Mitchell, T. 1998. Learning to classify text from labeled & unlabeled documents. In *Proceedings of the 1998 AAAI*. AAAI Press.

Pennycook, B., and Stammen, D. 1993. Real-time recognition of melodic fragments using the dynamic timewarp algorithm. In *Proceedings of the 1993 ICMC*. International Computer Music Association.

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

Thom, B. 2000a. Artificial intelligence and real-time interactive improvisation. In *Proceedings from the AAAI-2000 Music and AI Workshop*. AAAI Press.

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

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

Titterington, D. M.; Smith, A. F. M.; and Makov, U. E. 1981. *Statistical Analysis of Finite Mixture Distributions*. Wiley & Sons.

Vapnick, V. 1982. *Estimation of Dependences Based on Emperical Data*. Springer-Verlag.

Walker, W. 1997. A computer participant in musical improvisation. In *CHI 97 Electronic Publications*.