Computational Creativity via Human-Level Concept Learning

Paul Bodily, Benjamin Bay, and Dan Ventura
Computer Science Department
Brigham Young University
Provo, UT 84602 USA
paulmbodily@gmail.com, benjamin.bay@gmail.com, ventura@cs.byu.edu

Abstract

A common framework is helpful for effective evaluation, collaboration, and incremental development of creative systems. The Hierarchical Bayesian Program Learning (HBPL) framework was recently shown to be highly effective at learning human-level concepts, achieving new standards of performance in one-shot classification, parsing, and generation of hand-written characters. We argue that the HBPL framework is well-suited for modeling creative artefacts in general, one reason being that it allows explicit modeling of intention, structure, and substructure. Furthermore, the major challenge to the HBPL framework, namely how joint distributions should be factored, focuses system designers' attention on the philosophical debates that occur among artists themselves, suggesting that the HBPL framework might also serve as a more precise scaffolding for such debates. We demonstrate the framework's efficacy using lyrical composition as a specific example. In addition to being able to generate novel artefacts, we illustrate how HBPL models can be used to incorporate creative knowledge in broader applications including recommendation systems.

Introduction

People possess the ability to learn and combine concepts they already know to understand and even create new concepts. As an example, many pedagogical models (e.g., (Englemann and Bruner 1974)) teach children to read by systematically mastering and combining simple concepts: symbols represent sounds; symbols are read left to right; sounds are combined to form words; periods delimit phrases; sentences wrap to subsequent lines, etc. This process of hierarchical learning is at the heart of a branch of machine learning called *human-level concept learning*. Human-level concept-learning is characterized by three fundamental ideas (Lake, Salakhutdinov, and Tenenbaum 2015):

- *Compositionality* - observations are constructed through a combination of parts
- *Causality* - capturing abstract representations of the causal process that produces an artefact
- *Learning-to-learn* - parameters, constraints, parts, etc. are learned from training with related concepts and then applied to learning novel concepts

Hierarchical Bayesian program learning (HBPL) describes a framework that models human-level concept learning. This framework has recently been shown to be extremely effective (better even than deep-learning algorithms) in one-shot classification, parsing, and generation of hand-written characters (Lake, Salakhutdinov, and Tenenbaum 2015). The HBPL model for hand-written characters works by factoring a joint probability distribution over characters ψ into a product of conditional distributions,

$$P(\psi) = P(\kappa)\prod_{i=1}^{\kappa} P(n_i|\kappa)P(S_i|i,n_i)P(R_i|S_1,...,S_{i-1}), \quad (1)$$

where each conditional distribution is a model of a *sub-concept*: $P(\kappa)$ models the number of strokes per character; $P(n_i|\kappa)$ models the number of substrokes for the ith stroke for a character with κ strokes; $P(S_i|i,n_i)$ models the ith stroke with n_i substrokes; and $P(R_i|S_1,...,S_{i-1})$ models the relation of the ith stroke to the previous strokes. Some of these models are further decomposed. This process of decomposition allows the system to empirically learn sub-concepts in order to learn and generate new character types.

In this paper we investigate concept learning as a tool for building computationally creative systems. In particular, we find that the HBPL model provides a powerful framework for producing novel, typical artefacts that include elements of surprise by virtue of its wide range of expression.

As a proof of concept, we demonstrate the application of the HBPL model to the problem of lyrical pop music composition; however, the principles are readily applicable in other domains. Lyrical pop music is an ideal subject insofar as it naturally decomposes into multiple subconcepts, each of which can be further factored. The system we describe also demonstrates how existing models can be incorporated in defining subconcept distributions, using the specific example of Pachet et al.'s constrained Markov model (2011).

Modeling with HBPL

The most significant challenge to the HBPL model is deciding how and how far to factor the joint distribution. Bayes' theorem suggests that the factoring is irrelevant: any factoring should reproduce the joint when terms are multiplied:

$$P(A,B) = P(A|B)P(B) = P(B|A)P(A).$$

However, in practice we are only ever able to approximate distributions. Furthermore we at times make unproven in-

dependence assumptions to increase the power of our models (as discussed below). The factorization therefore leaves some "fingerprints" on the artefacts it produces according to the extent that each of the factors is accurately modeled.

Given that the space of possible artefacts is essentially infinite for many domains, it can be challenging to accurately train models for each subconcept given the relatively few artefacts that have actually been created. But often an approximation is sufficient to get a reasonable, working model. That we must use approximate distributions encourages the use of a modular framework for a few reasons. First, a modular framework affords the metacreator the opportunity to improve upon or substitute alternative approximative distributions for subcomponents. Second, multiple approximations can be combined to create improved approximations.

Depending on the complexity of the artefact class, the decision of how to factor the joint distribution can have significant impact on the power of the model. Some factorings generate subconcept models that may be easier to approximate. Some factorings may lend themselves to more reasonable independence assumptions. Choosing a good factorization often requires a deep understanding of the artefact domain.

For relatively simple artefacts, the decision of how to factor the joint is more straightforward. For example, consider just a few of the independence assumptions that Lake et al.'s model makes about hand-written characters (2015):

1. The number of substrokes per stroke, though dependent on the number of strokes, is independent from the number of substrokes in previous strokes and from the stroke-order position of the current stroke.
2. A substroke identity (i.e., shape) depends on the stroke-order position and the number of substrokes in the current stroke, but not directly on the total number of strokes in the character nor on the substroke identities of any but the directly previous substroke.
3. How strokes connect to previous strokes is independent of the number of strokes, substrokes, or substroke identities.

Initially these all seem like very reasonable simplifying assumptions, especially when considering how well the model performs. However if hand-written characters were more widely considered and utilized as an art-form, there may be some disagreement about how accurate these assumptions really are. Furthermore, the greater disagreements would likely come from what this choice of factoring says about the intuition behind how a character is generated: first randomly select a number of strokes κ; then select a number of substrokes n for each of those strokes based on κ; select the substroke shapes based on n and κ; and finally select the relationship between strokes. For most non-artistic character implementers, there is nothing wrong with this intuition. However, a calligrapher might feel that generating a new character really starts with choosing a substroke shape or a relationship between strokes. Note that the HBPL model could easily be adapted to model either of these alternative intuitions; but more importantly it highlights the debate of whether or not it is important *what* the model is doing as long as it appropriately classifies and generates character types.

In contrast, consider some potential independence assumptions and intuition represented in a model of lyrical compositions:

1. The structure, harmony, melody and lyrics are all independent of the inspiring source, given the intention.
2. The pitches of the melody are dependent on the harmony.
3. The number of syllables in the lyrics are dependent on the number of notes in the melody.
4. The lyrics are independent of the harmony, given the melody.

There are likely to be disagreements over some aspects of this factorization, reflecting philosophical biases of individual artists. Similar debates would arise, for example, in asking song-writers, "which do you write first: the lyrics or the melody?" Or asking story-writers, "which comes first: the characters or the story?" The fact remains that the same artefacts are produceable by multiple factorizations and the majority of those who appreciate the creativity of a song or a story do so without any knowledge of which factorization created it. These debates about how the model should be factored are the very same debates in which artists themselves engaged. By requiring the metacreator to precisely define how the joint should be factored, the HBPL model focuses attention on these debates and represents a computational framework in which differing perspectives can be readily compared and evaluated. For a discussion of different philosophies of lyrical composition and how they are represented as factorizations of the joint distribution over lyrical compositions see Bodily and Ventura (2017).

Composition

Analogous to equation 1, we define the conditional distribution on compositions γ, given an inspiration ι, as follows,

$$P(\gamma|\iota) = P(\nu|\iota)P(\tau|\nu)P(\eta|\nu,\tau)P(\mu|\nu,\tau,\eta)P(\lambda|\nu,\tau,\mu),$$

with the following definitions:
$P(\nu|\iota) =$ distribution over intentions ν given ι,
$P(\tau|\nu) =$ distribution over structure τ given ν,
$P(\eta|\nu,\tau) =$ distribution over harmony η given ν and τ,
$P(\mu|\nu,\tau,\eta) =$ distribution over melody μ given ν, τ, and η, and
$P(\lambda|\nu,\tau,\mu) =$ distribution over lyrics λ given ν, τ, and μ.

Although this factorization is dependent on the domain of lyrical composition, there are strong cross-domain parallels for many of the factors, which we will examine. This factorization of the distribution over compositions makes several independence assumptions which are discussed by Bodily and Ventura (2017). Given our factorization decisions, we generally find that the learned distributions broadly agree with musical intuition about how each of the subconcepts is defined as discussed in figure captions.

Intention, $P(\nu|\iota)$ *Intention* can be defined as the objectives which influence the creation of an artefact and can address several different facets (Bay, Bodily, and Ventura 2017):

- *Thematic intention* - the semantic purpose of the artefact (e.g., subject, emotion)
- *Cultural intention* - the sociocultural context for the artefact (e.g., society, language, era, genre)
- *Structural intention* - the target organization or arrangement of an artefact (e.g., technique, rhyme scheme, meter)

Whereas intention ν represents *what/how* we want to communicate, the *inspiration* ι represents the inspiring source for ν or *why* we want to communicate ν. Although many creative systems model intention (e.g., via a fixed intention, a user-defined intention, or randomly selecting an intention), a major advantage to the HBPL model is that we can explicitly condition the intention for an artefact on an inspiration. We discuss inspiration more below.

In our working lyrical composition example, we use a randomly selected thematic intention. Though several of the remaining subconcept models are conditioned on ν, it is only explicitly discussed in relation to $P(\lambda|\nu, \tau, \mu)$. We include it elsewhere as a reminder that intention can and should influence creativity wherever possible. We will assume that conditioning on ν is elsewhere accomplished by conditioning training on data representative of ν and leave a deeper exploration of its implementation for future work.

Structure, $P(\tau|\nu)$ In many domains of creativity structure can be thought of hierarchically. For example in a computer game the global structure may describe aspects of the flow between levels, but the levels themselves also have significant substructural elements that are intuitively independent from the global structure. We can thus factor our model of structure τ as

$$P(\tau|\nu) = P(\zeta|\nu)P(\sigma|\nu, \zeta)$$

where

$P(\zeta|\nu) =$ distribution over global structure ζ given ν and

$P(\sigma|\nu, \zeta) =$ distribution over segment structure σ given ν and ζ.

Global structure defines the boundary and relationships between subparts of an artefact. Examples might include the abstract sequence of plot line elements in story writing (e.g., "hero cycle" vs "tragedy") or the proportions of different abstract food groups in recipe generation (e.g., "chili" vs "sandwich") (e.g., (Morris et al. 2012)). In lyrical pop music, these subparts are readily apparent in the sequence of verses (V) and choruses (C) (which define large-scale repetitions in one or more musical viewpoints) and intros (I), outros (O), and bridges (B) (generally not wholly repeated). We refer to a subpart in our model as a *segment* and its value (e.g., "verse") as its *segment type*. A global structure for lyrical composition is a sequence of segment types $\zeta = (\zeta_1, ..., \zeta_n)$ with arbitrary length, where $\zeta_i \in \{I, V, C, B, O\}$. We define $|\zeta|$ as the number of segment types in ζ.

There are several ways to approximate $P(\zeta|\nu)$. One severely limited approximation is a *fixed* structure (e.g., I,V,C,V,C,B,C,O). Despite the range of possible compositions that are uncomputable by this approximation, this limitation would likely be overlooked if enough variation exists in other subcomponent models.

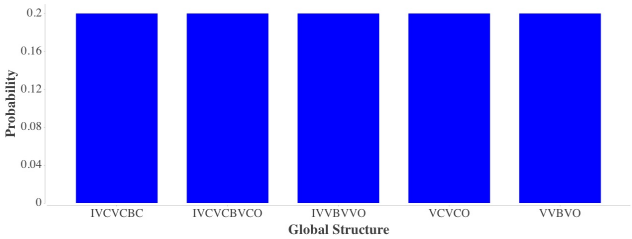

Figure 1: A visual representation of a possible probability distribution over global song structures composed of verses (V), choruses (C), intros (I), outros (O), and bridges (B).

A second approximation is a *distributional model* which learns a multinomial distribution of possible structures from a corpus of composition artefacts (e.g., see Figure 1). The disadvantage to the distributional model is that it can only produce structures seen in training.

A third, more powerful approximation uses a *constrained Markov model*. This model factors $P(\zeta|\nu)$ into a distribution over the number of segments in a song, $P(|\zeta|)$, and a single-order Markov model for sequences of segment types:

$$P(\zeta|\nu) = P(|\zeta|)P(\zeta_1)\prod_{i=2}^{|\zeta|} P(\zeta_i|\zeta_{i-1})$$

Note that an unconstrained, unsmoothed Markov model for $P(\zeta_i|\zeta_{i-1})$ provides no guarantee that a sequence of length $|\zeta|$ can or will be generated, nor that the sequence will end naturally (e.g., with an outro). With Pachet et al.'s *constrained* Markov model we can constrain the length and the way the sequence ends. This modifies the way $P(\zeta|\nu)$ is factored by conditioning ζ_i on both i and ζ_{i-1}:

$$P(\zeta|\nu) = P(|\zeta|)P(\zeta_1)\prod_{i=2}^{|\zeta|} P(\zeta_i|i, \zeta_{i-1})$$

When generating, a length is sampled from $P(|\zeta|)$ and a constrained Markov model for the sampled length is constructed from the unconstrained model $P(\zeta_i|\zeta_{i-1})$ with the added constraint that the song must end on an "end" token. This model is capable of creating sensible structures of reasonable length that were not seen in the training data. Empirical distributions for approximating $P(|\zeta|)$ and $P(\zeta_i|\zeta_{i-1})$ are shown in Figures 2 and 3 respectively.

A fourth possible solution for generating global structure would be to use a generative grammar, learned or manually constructed, similar to what was done by (Steedman 1984).

In addition to global structure, we also model *segment structure*, $P(\sigma|\nu, \zeta)$. Though this segment structure could be included as part of global structure, modeling this substructure independently leverages principles of abstraction and polymorphism in order to facilitate novel combinations of substructures. For example in story-generation the global structure might dictate something about the abstract content of each paragraph (e.g., protagonist faces a trial, protagonist

Figure 2: A visual representation of a possible probability distribution over the number of segments per song. Red corresponds to high probability, blue to low.

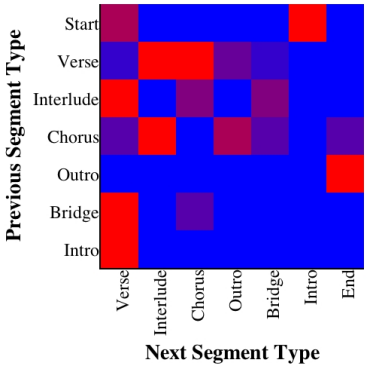

Figure 3: A visual representation of a possible single-order Markov transition matrix for segment types. Red corresponds to high probability, blue to low. The results largely agree with intuition. For example, songs generally start with an intro and occasionally with a verse; songs generally end with an outro and occasionally a chorus; and segments of the same type do not generally follow one another.

learns lesson, etc.), whereas the segment structure might define the narrative style for the paragraph (e.g., dramatic visualization, retrospection, dialogue, etc.) or add definition to the abstract content (e.g., the trial is a storm, the trial is losing a loved one, etc.). Modeling these structures independently enables the model to combine narrative styles with plot elements in ways that were not seen during training.

A segment in a composition (e.g., a verse) exhibits structure in the number of measures, the number of syllables or notes per segment, which lyrics rhyme or repeat, and patterns in harmony, pitch, or rhythm. We define a segment structure for lyrical composition as a sequence of pairs $\sigma = ((l_1, C_1), ..., (l_{|\zeta|}, C_{|\zeta|}))$, where l_i is the measure length of the ith segment (corresponding to ζ_i) and $C_i = \{c_{i1}, ..., c_{in}\}$ is a set of constraints which apply to the ith segment.

Constraints define restrictions on different musical viewpoints in order to create rhyme and repetitive motifs. A constraint, c_{ij}, is defined for a particular viewpoint $v \in \{Harmony, Pitch, Rhythm, Lyric\}$; with a condition $d \in \{Equals, Matches, RhymesWith, HasExpectation\}$; with a Boolean value t that defines whether the condition d needs to be satisfied or unsatisfied in order to satisfy the constraint c_{ij}; and with $m \in [0, l_i)$ and $b \in [0.0, bpm_m)$ representing the measure and beat offset within the segment to which the constraint applies (bpm_m is the beats per measure of m). Each condition d has different sub-variables

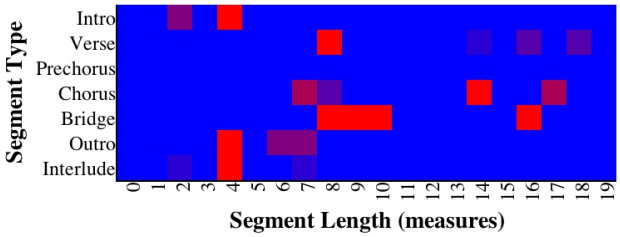

Figure 4: A visual representation of an empirically derived probability distribution over song segment lengths, conditioned on segment type. Red corresponds to high probability, blue to low. The results largely agree with intuition: intros, outros, and interludes tend to be shorter; verses, bridges and choruses tend to be longer.

and dimensionality:

- $Equals$ conditions - $c_{ij} = (v, d = Equals, t, m, b, S)$, where to satisfy d, the v token at or near measure m, beat b must equal a v token in the set of tokens S if t is $true$ and must not equal any v token in S if t is $false$.
- $Matches$ conditions - $c_{ij} = (v, d = Matches, t, m, b, m_2, b_2)$, where to satisfy d the v token at or near measure m, beat b and at or near measure m_2, beat b_2 within the segment must be equal if t is $true$ and not equal if t is $false$.
- $RhymesWith$ conditions - $c_{ij} = (v = Lyric, d = RhymesWith, t, m, b, m_2, b_2)$, where to satisfy d the $Lyric$ tokens at or near measure m, beat b and at or near measure m_2, beat b_2 within the segment must rhyme if t is $true$ and not rhyme if t is $false$.
- $HasExpectation$ conditions - $c_{ij} = (v, d = HasExpectation, t, m, b, s)$, where to satisfy d the v token at or near measure m, beat b must have an expectation value above a threshold s if t is $true$ and not have an expectation value above s if t is $false$. This constraint can be used to create a structure of expectation (as discussed by Meyer (2008)) in order to model patterns of surprise and tension.

Note that the attribute t could allow the system to learn how to intelligently break rules. For example, the system could intelligently learn when *not* to rhyme when perhaps a rhyme would normally be expected.

We define the distribution over segment structures σ as

$$P(\sigma|\nu, \zeta) = \prod_{i=1}^{|\zeta|} P(C_i|l_i)P(l_i|\zeta_i).$$

To approximate $P(l_i|\zeta_i)$ we can learn a probability distribution over segment lengths conditioned on segment type (see Figure 4). Under the assumption that the constraint set for a segment is independent of the segment type given its length, we can approximate $P(C_i|l_i)$ using a probability distribution over sets of constraints conditioned on segment length (e.g., see Figure 5).

Much of the work that has been done with finite-length Markov processes with constraints has required the user to

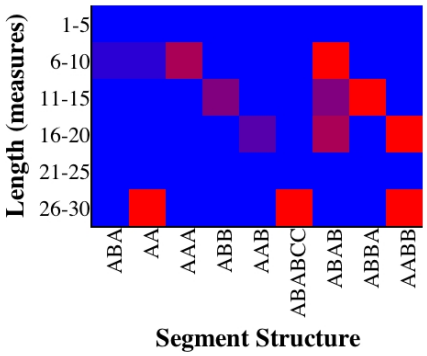

Figure 5: A visual representation of an empirically derived probability distribution over song segment rhyme structures conditioned on segment length. Red corresponds to high probability, blue to low.

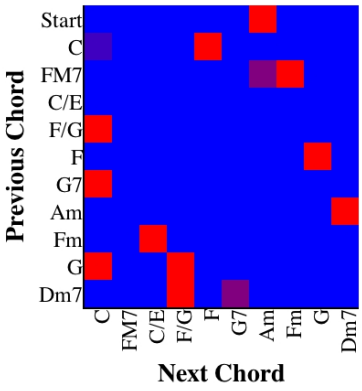

Figure 6: A subsection of a visual representation of an empirically derived single-order Markov transition matrix for harmonic chord sequences for chorus segments. Red corresponds to high probability, blue to low. As expected for songs normalized to the key of C major, there is high probability that the song starts on a C major chord.

specify the desired constraints in the composition process (e.g., (Pachet and Roy 2014; Barbieri et al. 2012)). This step of learning a model of constraints gives the system increased autonomy to choose its *own* constraints and then generate artefacts to meet those constraints.

With regard to modeling distributions for implicit features of an artifact (e.g., rhyme constraints), empirically-derived distributions can incur significant AI challenges. Artefacts used for training often fail to label global and even segment structure, and therefore these implicit features must be manually labeled or somehow inferred. Though our current system learns structure from a small manually-annotated dataset, our goal in future work is to use sequence alignment over multiple viewpoints to infer global structure, finding regions of a composition where harmony, melody, and lyrics all match (i.e., chorus) or where only harmony and melody match (i.e., verse). Sequence alignment is also a promising approach to finding segment structure (e.g., Hirjee and Brown (2010) use alignment to detect rhyme scheme).

Having modeled the abstract structural representation, the system proceeds to model the *operational* representation of the artefact (e.g., paint strokes, narrative text, recipe ingredients, etc.). Whether modeled jointly or factored, the operational variables describing the artefact composition are conditioned on the constraints imposed by the intention and global/segment structure. Adapting Pachet and Roy's definition of a jazz leadsheet (2014), we define the operational representation of a lyrical composition as parallel sequences of chords η, notes μ, and lyrics λ each with the same total duration. η, μ, and λ are defined in the following sections.

Harmony, $P(\eta|\nu,\tau)$ We define a harmony as a sequence of positioned chords $\eta = (C_1, ..., C_n)$ of arbitrary length. Each positioned chord $C_i = (I_i, d_i)$ has an identity $I_i = (r_i, q_i, s_i)$, with root pitch $r_i \in [0, 11]$, chord quality q_i (e.g., major, minor, dominant, etc.[1]), and bass pitch $s_i \in [0, 11]$; and a duration $d \in \mathbb{R}_{>0}$. We normalize all root and bass pitches based on the labeled key signature of the training instance at the harmony position.

We can factor $P(\eta|\nu,\tau)$ into independent sequential models regulating chord duration and chord identity:

$$P(\eta|\nu,\tau) = P(I_1|\tau)P(d_1|\tau)\prod_{i=2}^{n} P(I_i|I_{i-1},\tau)P(d_i|d_1,...,d_{i-1},\tau).$$

In this formulation, the length of the sequence n is dynamically determined such that $\Sigma_{i=0}^{n} d_i$ equals the segment duration.

Deciding how to implement $P(I_i|I_{i-1},\tau)$ and $P(d_i|d_1,...,d_{i-1},\tau)$ is non-trivial. A few possibilities for probabilistic sequence models include:

1. a *fixed generator* generates a fixed token, essentially ignoring conditioned variables
2. a *probability distribution* over tokens, conditioned on segment type and/or beat position, but not previous token
3. a *Markov model* that generates a new sequence for each segment, independent of segment type
4. a *set of Markov models* - one model per segment type
5. a *hidden Markov model* - hidden states representing the segment type

Each model has limitations that must be considered in the context for which it is intended. Of these, our implementation uses model 4 for $P(I_i|I_{i-1},\tau)$ (see Figure 6) and model 2 for $P(d_i|d_1,...,d_{i-1},\tau)$ (for a discussion of the relative musical merits of these models see Bodily and Ventura (2017)).

The decision to assume that duration and chord are independent, though potentially erroneous, is deliberate. This is based on the reasoning that the strength of a probabilistic model depends on the number of instances used to train the model. Each time a distribution adds a conditional variable,

[1] possible values for q_i are defined according to the MusicXML 2.0 specification for chord qualities

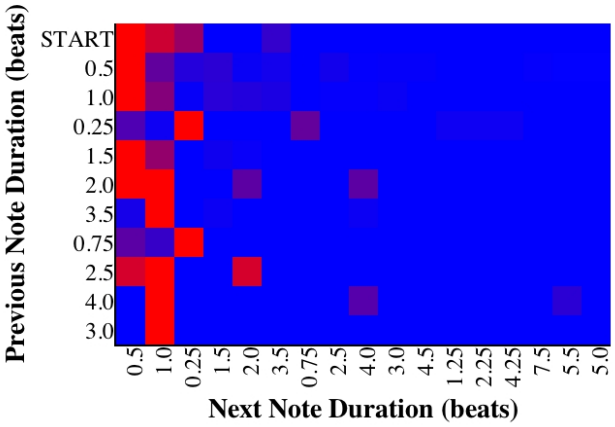

Figure 7: A visual representation of an empirically derived single-order Markov model for melodic rhythm durations for verse segments in 4/4. Red corresponds to high probability, blue to low.

the power of the model is reduced. We feel that the duration and chord are sufficiently independent that the model strength recovered by assuming independence outweighs the cost of ignoring any dependence between them.

Melody, $P(\mu|\nu,\tau,\eta)$ A melody is a sequence of positioned notes $\mu = (N_1, ..., N_n)$ of arbitrary length. Each note $N_i = (p_i, d_i)$ has a pitch $p_i \in [-1, 127]$ (corresponding to a MIDI note value, -1 representing a rest) and a duration $d_i \in \mathbb{R}_{>0}$. We factor $P(\mu|\nu,\tau,\eta)$ into independent sequential models regulating note pitch and duration:

$$P(\mu|\nu,\tau,\eta) = P(p_1|\eta)P(d_1|\tau)\prod_{i=2}^{n} P(p_i|p_{i-1},\eta)P(d_i|d_{i-1},\tau).$$

The length of the sequence n is dynamically determined such that $\Sigma_{i=0}^{n} d_i$ does not exceed the segment duration.

Of these models only pitch is conditioned on η. To model $P(p_i|p_{i-1},\eta)$ our implementation uses a single-order Markov chain of scale steps where the scale is defined by the contextual harmony of η. To model $P(d_i|d_{i-1},\tau)$ we use a segment-specific Markov chain of note durations (see Figure 7). Any of the probabilistic sequence models considered for harmony could also be considered here.

Lyrics, $P(\lambda|\nu,\tau,\mu)$ Several models of natural language generation (NLG) and in particular NLG in poetry and music have been published (Paris, Swartout, and Mann 2013). As these models continue to improve, so will their application in lyrical composition. This demonstrates the robustness of the HBPL framework: as improved submodels are conceived and implemented, the joint model is also improved.

We define lyrics as a sequence of stressed syllables $\lambda = (S_1, ..., S_n)$ where $|\lambda| \leq |\mu|$. A stressed syllable $S_i = (t_i, p_i, \epsilon_i)$ has a text representation t_i, a pronunciation p_i (e.g., sequence of ARPAbet phonemes), and a stress $\epsilon_i \in [0, 2]$. Each syllable $S_i \in \lambda$ corresponds to one and only one note $N_j \in \mu$.

We factor $P(\lambda|\nu,\tau,\mu)$ to construct λ as a sequence of lyric phrases $(\phi_1, ..., \phi_n)$ where the number of phrases n and the length l_{ϕ_i} (in syllables) of each phrase are computed as a function of the notes in μ and the rhyme constraints in τ (i.e., we assume rhyme constraints denote phrase endings):

$$P(\lambda|\nu,\tau,\mu) = \prod_{i=1}^{n} P(\phi_i|l_{\phi_i},\nu,\tau)P(l_{\phi_i}|\mu,\tau).$$

We empirically derive $P(l_{\phi_i}|\mu,\tau)$. For $P(\phi_i|l_{\phi_i},\nu,\tau)$ we create a probability distribution of lyric templates conditioned on l_{ϕ_i} which we use to sample templates. These templates, the $RhymesWith$ constraints of τ, and ν are given as input to an independent module that generates novel, intentioned lyrics (see Bay, Bodily, and Ventura (2017)). The module uses existing lyric segments as syntactic templates for the creation of novel lyric segments. It intelligently selects and replaces words based on 1) semantic similarity, 2) part-of-speech tag, 3) the cultural and thematic intention of ν, and 4) the rhyme constraints imposed by τ.

The advantage of using a template-based approach to lyrics generation is that it maintains syntactic coherence. The primary shortcomings are that resulting lyrics provide limited syntactic novelty from the training data and make no inherent effort at providing global semantic cohesion.

A Note on Constrained Markov Models Pachet et al.'s constrained Markov model requires that the length of the sequence be defined *a priori* (2011). One short-coming in our current implementation is that because we have included duration as part of the definition for both harmony and melody (rather than having each chord or note representative of a fixed duration as demonstrated by Pachet and Roy (2014)) the length of a harmony or melody sequence depends on the durations of each sampled chord or note. While this violates the Markov property and prevents us from being able to effectively use constrained Markov models, we favor the current implementation for reasons related to data sparsity issues and the complexity of implementing higher-order constrained (hidden) Markov model. We hope in the future to overcome both of these hurdles and to shift to "Markov-friendly" definitions for melody and harmony in order to more fully incorporate the constraints defined in τ using constrained Markov or constrained hidden Markov models.

Results and Discussion

We present results of implementing the HBPL framework in the context of a discussion of some of the model's implications. We trained submodels on a small manually-annotated subset of the Wikifonia leadsheet dataset.

Using the Joint as a Submodel

Because of the hierarchical nature of HBPL, a joint model of an artefact class (e.g., the model of $P(\gamma|\iota)$ just described) can serve as a submodel for other models. For example, we define the joint probability distribution on inspirations ι, compositions γ, and renderings ρ^m as follows,

$$P(\iota,\gamma,\rho^1,...,\rho^m) = P(\iota)P(\gamma|\iota)\prod_{m=1}^{M} P(\rho^m|\iota,\gamma).$$

Figure 8: Three measures of a sample composition generated using the HBPL framework. The full composition and others can be found online at `popstar.cs.byu.edu`.

In essence we decompose a model of music *creation* to individually model the inspiration for the artefact, the symbolic (abstract) representation of the artefact, and the concrete rendering of the artefact.

Inspiration, $P(\iota)$ *Inspiration* (i.e., the method for deriving intention) may be more closely related to an artist's or system's "creative spark". For example, observers often perceive greater creativity in artefacts which in some way relate to them or to their culture (Colton 2008). In the joint probability distribution on inspirations ι, compositions γ, and renderings ρ^m, we define $P(\iota)$ not as the distribution over intentions, but as the distribution over inspiring sources for the intention. In other words, not "what was the artefact intended to communicate?", but "what was the inspiring *source* for what the artefact intended to communicate?"

In general this demonstrates an unanticipated benefit of factorization: we can condition on any variable that could be argued to influence the artefact's creation. Many creative systems implicitly define inspiration based on the corpora that the data trains on. With the concept learning framework, we can model this attribute explicitly.

This represents an aspect not present in the model originally presented by Lake et al. (2015): not only are we modeling *what* artefacts can be generated, but also *why* they are generated. One possible way to model inspiration is to use an observer's environment or culture as an inspiring source. Research in electroencephalogram-based affective computing (i.e., reading brain waves) suggests that computers may soon be able to perceive an observer's emotional state beyond those of their human counterparts (Volioti et al. 2016). Alternatively, inspiration could be modeled using sentiment analysis in a variety of online domains. We plan to explore models of inspiration further in future research.

Rendering, $P(\rho^m|\iota,\gamma)$ The example model $P(\gamma|\iota)$ described above defines symbolic lyrical compositions (i.e., a leadsheet). However, evaluating an abstract artefact generally requires a concrete rendering of the artefact, whose distribution we model as $P(\rho^m|\iota,\gamma)$. As a proof of concept, we implemented and trained the described HBPL model on a small corpus of hand-annotated lyrical pop composition data. To concretely render compositions created using this model, we generated both printed sheet music (e.g., Figure 8) and an MP3 audio recording[2]. Our MP3 audio file features computer-sung lyrics accompanied by synthesized piano and bass comping chords[3].

Implications for Recommendation Systems Lake et al. present the model of $P(\psi)$ given in equation 1 as a submodel of the factoring of the joint probability distribution on character types ψ, tokens ω^m, and binary images I^m (2015):

$$P(\psi,\theta^1,...,\theta^M,I^1,...,I^M) = P(\psi)\prod_{m=1}^{M} P(I^m|\theta^m)P(\theta^m|\psi).$$

This means that given an image, the system can discover the motor program (i.e., abstract character type) that most likely generated it. This allows the system to one-shot classify and generate pairs of images that represent the same character type (specific examples of which were not seen in training).

By analogy, a model for $P(\gamma)$ (similar to $P(\gamma|\iota)$ just described) could be inserted into a joint probability on composition types γ, arrangements α^m, and audio recordings ρ^m,

$$P(\gamma,\alpha^1,...,\alpha^M,\rho^1,...,\rho^M) = P(\gamma)\prod_{m=1}^{M} P(\rho^m|\alpha^m)P(\alpha^m|\gamma).$$

The implications of this model are more broadly significant: the HBPL framework is capable of inferring abstract representations of concrete artefacts, representations which more directly define meaning, composition, and causality. This is significant for two reasons. First, in some realms of creativity, simply deriving the abstract representation of an artefact is valuable (e.g., automatically transcribing sheet music from audio). Second, having an abstract representation allows concrete artefacts to be compared according to symbolic, conceptual criteria (e.g., recommendation systems based on meaning, or in the case of music, harmony, melodic pitch or rhythm, etc.). Though work has been done to approximate $P(\alpha^m|\gamma)$ (Benetos et al. 2013), effective comparison of artefacts hinges on the other terms in the factorization, $P(\gamma)$ and $P(\rho^m|\alpha^m)$, which are lacking.

Fitness and Self-Evaluation

The HBPL framework is designed to restrict the generation process *in situ* to produce only meaningful artefacts (as compared to a generate-and-test procedure). As discussed by Ventura (2016), this "baked-in" self-evaluation mechanism has the added benefit of being able to explain to some extent both the novelty, value, and motivation behind generated artefacts. Given its ability to compute probabilities, the HBPL framework could thus also be potentially leveraged as a fitness function for other types of generative models.

[2] audio recordings can be found at `popstar.cs.byu.edu`

[3] generated using Harmony Assistant (v9.7.0f) and Virtual Singer (v3.2)

Big (Need for) Data

Any empirically-driven model requires training on a dataset representative of the artefact domain. Even if we had digital access to all of the compositions ever written, it would represent an infinitesimal portion of the songs that *could* be written. This is a challenge in many machine learning domains. Unique to the pop music domain, however, is that data is highly proprietary. What *is* available is extremely limited and of relatively poor quality. Compared to natural language, artefacts in music generally require relatively complex representations and relatively few possess the domain knowledge required to generate or transcribe the needed data. Among those who *do* understand and use it, music formatting can vary wildly and inexactly—creating additional challenges for a by-the-bit computer parser. Computers will only learn to speak music as quickly as we either formalize and ubiquitize the language of music *or* endow computers with AI tools to fill in the gaps on their own.

The particular challenge of accessing high-quality symbolic *pop* music datasets is significant. There is a dearth of well-annotated resources for those interested in studying any or all of the aspects of pop music composition. There is, however, much we can do to improve the situation. First, we need to make resources that *are* available more accessible (guitar tabs, lyrics sites, beatles). Second, we need to establish a better case for how society and industries stand to benefit from computational pop music research in order to generate a productive dialogue for the support and collaboration of those in possession of large pop music datasets (sheet music sites, spotify, etc., asking for APIs, etc). Note that this is different than asking them to simply give us their proprietary data. Third, we can do more to recognize contributions of novel datasets.

Conclusion

HBPL is a powerful framework for accomplishing tasks in computational creativity. Using principles of compositionality, causality, and learning-to-learn, such models are able to effectively learn and generate examples of complex creative concepts. Its probabilistic framework lends itself well to modeling important aspects of creativity such as inspiration and intention. The HBPL framework by nature compels researchers in domain-specific subareas of computational creativity to engage in the debates that the artists themselves are having, namely "how should an artefact be created?" and "does it matter?" To the extent that these challenges are effectively addressed on the scale of defining and training subconcept models, the HBPL model represents a useful framework for designing and assessing creative systems.

References

Barbieri, G.; Pachet, F.; Roy, P.; and Esposti, M. D. 2012. Markov constraints for generating lyrics with style. In *Proceedings of the Twentieth European Conference on Artificial Intelligence*, 115–120. IOS Press.

Bay, B.; Bodily, P.; and Ventura, D. 2017. Deterministic text transformation via constrained vector-word representation. To appear in Proceedings of the Eighth International Conference on Computational Creativity.

Benetos, E.; Dixon, S.; Giannoulis, D.; Kirchhoff, H.; and Klapuri, A. 2013. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems* 41(3):407–434.

Bodily, P., and Ventura, D. 2017. Musical metacreation using the hierarchical Bayesian program learning framework. Submitted to MUME2017.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *AAAI Spring Symposium: Creative Intelligent Systems*, volume 8.

Englemann, S., and Bruner, E. 1974. *DISTAR: Reading Level I*. Chicago: Science Research Associates.

Hirjee, H., and Brown, D. 2010. Using automated rhyme detection to characterize rhyming style in rap music. *Empirical Musicology Review* 5(4).

Lake, B. M.; Salakhutdinov, R.; and Tenenbaum, J. B. 2015. Human-level concept learning through probabilistic program induction. *Science* 350(6266):1332–1338.

Meyer, L. B. 2008. *Emotion and Meaning in Music*. University of Chicago Press.

Morris, R. G.; Burton, S. H.; Bodily, P. M.; and Ventura, D. 2012. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *Proceedings of the Third International Conference on Computational Creativity*, 119–125.

Pachet, F., and Roy, P. 2014. Imitative leadsheet generation with user constraints. In *Proceedings of the Twenty-first European Conference on Artificial Intelligence*, 1077–1078. IOS Press.

Pachet, F.; Roy, P.; Barbieri, G.; and Paris, S. C. 2011. Finite-length Markov processes with constraints. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 635–642.

Paris, C.; Swartout, W. R.; and Mann, W. C. 2013. *Natural language generation in artificial intelligence and computational linguistics*, volume 119. Springer Science & Business Media.

Steedman, M. J. 1984. A generative grammar for jazz chord sequences. *Music Perception: An Interdisciplinary Journal* 2(1):52–77.

Ventura, D. 2016. Mere generation: Essential barometer or dated concept? In *Proceedings of the Seventh International Conference on Computational Creativity*, 17–24.

Volioti, C.; Hadjidimitriou, S.; Manitsaris, S.; Hadjileontiadis, L.; Charisis, V.; and Manitsaris, A. 2016. On mapping emotional states and implicit gestures to sonification output from the 'intangible musical instrument'. In *Proceedings of the Third International Symposium on Movement and Computing*, 30. ACM.