ThomCat: A Bayesian Blackboard model of Hierarchical Temporal Perception

Charles Fox
Robotics Research Group
Engineering Science
University of Oxford

Abstract

We present a Bayesian blackboard system for temporal perception, applied to a minidomain task in musical scene analysis. It is similar to the classic Copycat architecture (Hofstadter 1995) but is derived from rigourous modern Bayesian network theory (Bishop 2006), with heuristics added for speed. It borrows ideas of priming, pruning and attention and fuses them with modern inference algorithms, and provides a general theory of hierarchical constructive perception, illustrated and implemented in a minidomain.

Introduction

Real-time perception of musical structures is an exemplar of hierarchical temporal perception, and is a useful domain in which to demonstrate general theories of scene analysis which emphasize the temporal element. Time is notably absent in many other test domains for scene analysis, especially static visual scenes as opposed to video.

We use perception of semi-improvised musical performance from an 'almost-known' grammar as a specially selected minidomain to illustrate a theory of temporal hierarchical perception. A minidomain is a simplified but non-'toy' problem which is chosen to capture properties of a general task. Here we aim to illustrate ideas about general temporal perception and have chosen a relatively simple but non-trivial task in which to work. *Semi-improvised* music is generated by a performer from a probabilistic context *sensitive* grammar, based on a stochastic context-free grammar (SCFG) such as

$$\begin{aligned}
p(SONG \rightarrow VERSE, CHORUS, VERSE) = 1 \\
p(VERSE \rightarrow A, A, B) = 0.75 \\
p(VERSE \rightarrow B, B, A, A) = 0.25 \\
p(CHORUS \rightarrow K, L, K, L) = 1; p(A \rightarrow c) = 1 \\
p(B \rightarrow c, g) = 1; p(K \rightarrow f, am) = 1; p(L \rightarrow f, g) = 1
\end{aligned}$$

In addition to these context-free term-rewriting probabilities, music superimposes various global probability factors. The most notable is the global *key* which acts to prefer coherent sets of chords and notes throughout the whole musical scene. Similarly, melodic themes are often repeated throughout a performance, with a bias to preferring rewrite rules that have previously been applied. This is typically the case in jazz, rock and many non-Western art musics: a composition is a fluid entity consisting of recognizable chunks and conventions which are combined in real time by the player. At the structural level, performers may decide on the fly to insert extra verses and choruses if the performance is going well and the audience is wanting more; or to cut out sections if it is going badly. At the low level of chords, rhythm players are likely to improvise around the standard chords, with different changes having probabilities. An *almost-known* grammar means that the performance is generated from a grammar which is mostly similar to the grammar known by the perceiver, but may include additional and differing rules and differing probabilities. The perceiver must thus allow for the fallibility of its own knowledge of the grammar and be prepared to learn new rules on the fly.

The context-sensitive nature of the global factors, and the required allowance for one's own failure preclude the use of the fast dynamic programming algorithms used in simplified language processing tasks such as the Inside-Outside algorithm, and more general scene analysis techniques are needed. Indeed this was a reason for selection of the microdomain as representative of general scene analysis.

We consider musical scenes comprised of observations of low-level chords, assuming one chord per bar, and that chords are the terminals of the almost-known grammar, and influenced by global key. This minidomain task is enough to capture and illustrate the general context-sensitive and 'almost-known' requirements of general scene analysis. In particular we do not consider perception of melodies or individual notes in the present work. Our architecture, ThomCat, is largely inspired by the Copycat blackboard system (Mitchell 1993a), which itself was explicitly based upon (and cites) Ising models, Gibbs sampling and simulated annealing (Hofstadter 1987). This statistical mechanics basis is generally under-appreciated due to Copycat's use of nonstandard terminology and heuristics. ThomCat is an explicitly Bayesian version of the Copycat architecture applied to the music minidomain. Creation and destruction of hypotheses by agents is translated into a simple rule which restricts the search space of the Gibbs sampler at each step to the neighbors of active nodes in the previous step. Rivalry is dealt with in the rigorous Bayesian sense of hypothesis comparison, and makes use of a central 'thalamus' lookup table for speed. ThomCat shows how to view and extend Copycat with modern machine learning concepts, and provides a useful method for high-level musical understanding. It may be of interest to music psychologists who could try setting its probabilities to model human perception; and to neuroscientists as a rough plan for some neural functions. While we focus on the domain of musical understanding, the architecture could be carried over to other forms of scene analysis.

Since Kant, perception has been seen as the process of interpreting low level sense data (e.g. pixels in an image or chords in a musical performance) *as* instantiations of known concepts in perceptual space (which may be spatial or temporal or both). The relativistic ideas of (Whorf 1956) emphasized that different subjects may have different sets of concepts through which to view the data, learned from different, unique sets of experiences. (James 1907) emphasized the Pragmatist view that the subject should learn and use the concepts which are the most useful in its typical environments. The use of our minidomain can provide insight into these ideas by showing quantitatively how they may be implemented. In our miniomain, the sense data consist of chord likelihoods in musical bars, and the concepts to be instantiated are the musical structures found in the grammar, along with the concepts of different musical keys.

A key distinction to be made is between perception *of* time and perception *in* time. Time in the former refers to the states of the world over time, while the latter refers to the times at which computations and inferences are made.

We give a model of temporal perception which maintains and beliefs about a *region* of time and updates them as new information arrives and is processed. We note than confusion between perception *of* and *in* has led to many apparent philosophical problems such as those raised by the experiments of (Libet 2004) in which future events appear to be perceived before they occur. Our model of temporal perception shares the *of* and *in* distinction with the Multiple Drafts analysis of (Dennett 1998) but differs in that like Blackboard systems, and Copycat in particular, it constructs a single coherent scene interpretation at each inference step rather than multiple drafts. We restrict on-line inference to a temporal window around the current time, 'freezing' the results outside it. This is a simple version of a 'spotlight of attention' (Crick 1984) and gives insight into how attention could work in general scene analysis.

ThomCat currently runs best using Gibbs sampling similar to Copycat (though extended to use cluster sampling), but we see that once Copycat's architecture is brought into the Bayesian network framework, the path is opened for other modern inference algorithms such as Variational Bayes to run on the same structures. The ability for unseen rules to be perceived in terms of free-floating known components provides a framework to further extend the architecture to learn new rules and probabilities.

Formal task specification

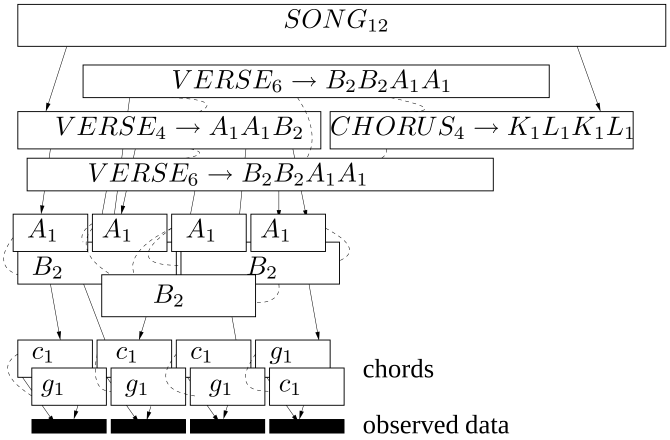
The task is perceive a sequence of incoming chord observations as the maximum a posteriori (MAP) single coherent structure which generated them, or as close an approximation to it as possible. We assume that a musical bar (measure) and chord detector module is available. At each bar t, we receive a set of chord likelihoods $\{(c_i, \lambda_i\}_{i=1}^{N_t}$, where c_i are chord symbols (such as *em, g, c*) and λ_i are likelihoods $\lambda_i = P(D|c_i)$ with D being the raw data received by the chord detector, such as a sound wave or MIDI stream. We allow for major and minor chords from each of the 12 possible root notes.

We are given a prior SCFG generation model, modified to become context sensitive with a set of key coherence factors and to allow for 'almost-known' structures generated by rules which are missing from the grammar. The SCFG is a set of rewrite rules of the form $X_i \rightarrow \{Y_k\}_k, p_i, \pi_i^\emptyset$ where X are symbols. If a symbol ever appears as the left hand side (LHS) X_i of some rule i then it is called a non-terminal. If it does not appear in this way it is called a terminal, and is assumed to be a lowest-level chord symbol such as *em, g, c*. The probability p_i is $P(\{Y_k\}_k|X_i,)$, that is, the probability that the LHS term is rewritten as the RHS sequence rather than as some other RHS sequence, given that the LHS already exists.

The $\pi_i^\emptyset$ values are *free-floating probabilities*, used in understanding 'almost-known' structures which allow the possibility of occasional new rules being used in the performance which are not known to be part of the grammar. We consider the case of new rules whose RHS are comprised of known symbols. To allow the possibility of learning such rules, we must first be able to perceive the sequence of RHS structures as existing in perceptual space, but with no parent structure (e.g. the K in fig. 2). Such percepts are called *free-floating*, and we allow each known symbol X_i to free-float with probability $\pi_i^\emptyset$. This is an important feature in general scene analysis tasks. For example, in visual scenes there is usually no explicit top-level grammatical structure, rather the scene is made of objects at arbitrary positions in space, which in turn are comprised of hierarchies of subobjects. This is a key difference between scene analysis and language processing, the latter almost always assuming the existence of some top level 'sentence' structure which ultimately generates everything in the linguistic 'scene'.

We assume that the whole performance is in a single musical key κ, which affects the chord observations via compatibility factors, $\prod_i \phi(\kappa, c_i)$ so the total probability of a scene interpretation is given by the probability product of all the rewrite and floating rules used in the derivation, multiplied by this factor and normalized. We assume a flat prior over 12 keys (assuming relative major and minors to be equivalent).

We require quasi-realtime inference, meaning that complexity should not increase as the performance progresses. (Though as we present ThomCat as perceptual model rather than a practical application, we do not currently require it to run in real-time on a desktop PC). Scene analysis is generally used to make predictions of the near-future, so to illustrate this we will require a near-optimal prediction of the



$SONG_{12}$
$VERSE_6 \rightarrow B_2B_2A_1A_1$
$VERSE_4 \rightarrow A_1A_1B_2$
$CHORUS_4 \rightarrow K_1L_1K_1L_1$
$VERSE_6 \rightarrow B_2B_2A_1A_1$
A_1
A_1
A_1
A_1
B_2
B_2
B_2
c_1
c_1
c_1
g_1
g_1
g_1
g_1
c_1
chords
observed data

Figure 1: A network instantiated from the test grammar

musical structure of the next bar, and preferably a rough prediction of the structure of the entire future performance.

Architecture: static structures

We view the scene generation process as (causal) Bayesian networks as shown in the example of fig. 1. The existence of each musical structure (called a *hypothesis*), such as a *SONG* causes its components to exist, with a probability given by the grammatical rewrite rule. Each possible hypothesis can also exist as a free-floater, with a probability[1] $\pi^{\emptyset}$. The probability of a hypothesis is given by the noisy-OR function of its parents and its free-floating probability,

$$P(h|pa(h)) = 1 - (1 - \pi^{\emptyset}_{h}) \prod_{x \in pa(h)} I(x)(1 - P(h|x))$$

where x ranges over the parents $pa(h)$ of hypothesis h, and $I(x)$ is an indicator function which is 1 if hypothesis x is true and 0 if it is false.

The lowest level hypotheses are the chords reported by the external bar and chord classifier. To handle these in the same framework as the rest of the grammar, we construct a lowest-level set of nodes called ‘observed bars’ or *ObsBars* which represent the raw data observed by the external classifier. ObsBars then have transition probabilities conditioned on the reported possible chords c_i, $P(obsBar|c_i)$ which are set to the reported likelihoods from the classifier.

The dashed links in the figure connect pairs of mutually exclusive, or *rival* hypotheses. As a single coherent explanation of the scene is required, we cannot allow the existence of pairs of hypotheses which overlap in time and level of explanation. Formally, these links represent pairwise probability factors, $\phi(h_1, h_2) = 1 - I(h_1)I(h_2)$, which when multiplied into the scene joint distribution, set scenes with overlapping rivals to have probability zero, and have no effect otherwise.

The hypotheses are of the form ‘there is a structure of type X which exists in temporal space starting at time t and having length L’. In general, grammars allow the same type X to exist having different lengths, depending on how it becomes expanded later. In our framework this is important because such alternative expansions give rise to rival hypotheses such as the two $VERSE$ structures at the start of the figure, of lengths four and six bars. So hypotheses do not use the raw grammars as seen so far, but instead make use of compiled *lengthened* grammars (*l-grammars*), constructed from SCFGs. l-grammars split up SCFG rules which can exist in different lengths into sets of individual rules which make the lengths explicit, and convert the transition and free-float probabilities appropriately. Details of the l-grammar conversion process using an offline message passing algorithm are given in the appendix. Here we show the result of the l-grammar compilation for the previous example SCFG:

$$S_{16} \rightarrow V_4C_8V_4 : p = 1.0, \pi^{\emptyset} = 0.253$$
$$S_{18} \rightarrow V_4C_8V_6 : p = 0.5, \pi^{\emptyset} = 0.084$$
$$S_{18} \rightarrow V_6C_8V_4 : p = 0.5\pi^{\emptyset} = 0.084$$
$$S_{20} \rightarrow V_6C_8V_6 : p = 1, \pi^{\emptyset} = 0.028$$
$$V_4 \rightarrow A_1A_1B_2 : p = 1, \pi^{\emptyset} = 0.15$$
$$V_6 \rightarrow B_2B_2A_1A_1 : p = 1, \pi^{\emptyset} = 0.05$$
$$C_8 \rightarrow K_2L_2K_2L_2 : p = 1, \pi^{\emptyset} = 0.2$$
$$A_1 \rightarrow c_1 : p = 1, \pi^{\emptyset} = 0.05; B_2 \rightarrow c_1g_1 : p = 1, \pi^{\emptyset} = 0.05$$
$$K_2 \rightarrow f_1am_1 : p = 1, \pi^{\emptyset} = 0.05; L_2 \rightarrow f_1g_1 : p = 1, \pi^{\emptyset} = 0.05$$

The $\pi^{\emptyset}$ in the *unlengthened* grammar are here assigned automatically based on the hierarchical level of the rule. We assume that large-scale structures are more likely than lower-level structures to appear as free-floaters: the performer is very likely to play a top-level $SONG$ and is more likely to insert an extra $VERSE$ into that song than to insert extra A and B riffs into the $VERSE$ and extra bars of chords into those riffs.

In addition to musical structure hypotheses, we also model beliefs about the global key. There are 12 possible keys, each is either true or false, and all are mutual rivals so that only one key is allowed in a single coherent scene. Undirected links are added to the chord-level hypotheses, connecting them to the 12 key nodes and applying the pairwise compatibility potentials. A flat prior is placed over the keys by giving each key an equal $\pi^{\emptyset}$ probability factor.

Discussion of the static structure

The model is a causal Bayesian network in the sense that it respects Pearl’s *do* semantics (Pearl 2000). It is a ‘position-in-time’ model rather than a ‘dynamic state model’, as it conceives of the nodes as existing in a temporal space, and makes inferences about what exists in each part of space. (In contrast, hierarchical dynamic-state models such as (Murphy 2001) would represent the explicit set of structures active at each individual time point. Algorithmically this can

[1] In our microdomain these are probabilities as the possible structure locations and sizes are discretized by the bars. In more general scene analysis this is not the case, and probability *densities* would be used over continuous space and size.

be made to perform the same inference computations as the position-in-time model, but is less satisfactory as a theory of perception: position-in-time models construct an explicit phenomenal space and populate it with objects, which fits well with subjective perceptual experience.) We also note here the presence of much causal independence (Zhang & Poole 1996) in the network which might be exploitable by inference algorithms we choose to run on it.

The multiple hypotheses that may be instantiated from these lengthened rules are similar to those which appear in the context-free Inside-Outside algorithm, which instantiates *every* possible rule at *every* possible start time and *every* possible length. The l-grammar method is generally more efficient as it only ever instantiates hypotheses which can actually ever exist. For example, in our example grammar, there is no way that a $SONG$ of length less than 12 can ever exist, so it is wasteful to consider such hypotheses. This speedup is especially important when we move away from polynomial solvable context free structures to NP-hard context sensitive ones. Ideally, we would instantiate all possible hypotheses allowable by the l-grammar, giving rise to a single, large Bayesian network (not as large as that used by Inside-Outside – but lacking polynomial-time dynamic programming algorithms) and run inference on it. However such a network would still grow as $O(RT^2)$ like Inside-Outside, with R the number of lengthened rules and T the total number of bars, both of which are likely to be large in music and other domains of scenes, rendering even approximate inference impractical (Dagum & Luby 1993). Worse still, to generalize away from the music microdomain to continuous scene analysis domains such as vision, an infinite continuum of hypotheses would be needed which is clearly uncomputable.

Architecture: structural dynamics

Rather than instantiating all possible hypotheses from the outset, we turn to the classical AI architecture of Blackboard Systems for heuristics. Blackboard systems such as Copycat aimed to *construct* a single unitary percept by allowing multiple agents to add, change and remove parts of the percept. Each agent, when called upon, searches the blackboard for any changes it could contribute. If the changes improve the percept, they are made; if they make it worse, they may be made anyway, according to some heuristic, to help escape from local minima. Randomizing the orderings of agents to call upon also adds stochasticity which helps to escape such minima. In Copycat, the acceptance probabilities are made using simulated tempering, based on statistical mechanics. Factors in the scene (analogous to our rewrite probabilities, free-float priors, chord likelihoods, rivalry and key compatibilities factors) define energy contributions, and the objective is to minimize the energy. Changes proposed by agents are accepted with approximate Gibbs probability $P = \frac{1}{Z}\exp(-E/T)$ where T is a variable temperature parameter (Copycat's exact tempering equations are found in the functions `get-temperature-adjusted-probability` and `update-temperature` in source file `formulas.l` in (Mitchell 1993b)). Extending the simulated annealing algorithm to tempering, Copycat defined temperature as a function of the current 'stuckness' of the inference: if there is little change, suggesting a local minimum, then temperature is increased to escape from it.

Classical blackboard systems always maintained a single coherent state, even if that state is unlikely and is part of a path between local minima. Each proposal in sequence was either accepted and updated on the blackboard, or rejected and forgotten. In Copycat, proposals occur rapidly and repeatedly, resulting in a 'flickering halo' (Hofstadter 1995) of accepted proposals flashing briefly into existence around the local minimum. This set of nearby potential but usually-unrealized percepts is analogous to William James' notion of the *fringe* of awareness: the set of percepts which *could* be easily perceived if required – and which may now be quantified using psychophysical priming experiments, which show faster responses to fringe members than to non-members (Wiggs & Martin 1998). In Copycat and other classical blackboards, this fringe is an implicit notion, defined by an observer of the program to be the set of hypotheses which tend to flicker around the current solution.

ThomCat provides a new formulation of the fringe, derived from a heuristic to make its structures tractable. We discussed above how instantiating all possible hypotheses is intractable. But luckily there is some domain knowledge in scene analysis tasks that may inform a heuristic to limit the size of this search space *at each step* of inference. The implicit search space will remain the same, but at any point during inference, only a small subset of it will be instantiated and used to run inference. Recall that the task is to seek MAP solutions, not full joints. Consider a candidate solution set X of hypotheses that together explain the data (i.e. a set of nodes all having value 'true'). Now consider solution set $X' = X \cap x_d$ where x_d is a 'dead' hypothesis. A *dead* hypothesis is one with no 'true' parents or children. Recall that all hypotheses in our models have *small* free-float priors, in this case $\pi_{\emptyset}(x_d)$. Because the hypothesis is floating, setting its value to 'true' will always decrease the total network configuration probability. So as we only seek the MAP solution we know that we can exclude configurations with floating nodes from consideration and computation.

This fact gives a rigorous foundation for a *Bayesian blackboard* approach. We treat the hypothesis nodes as objects on a blackboard, which may be inserted and removed *during* inference. *Cues* in 'true' low-level hypotheses *prime* (i.e. instantiate) higher-level hypotheses and vice-versa, known as bottom-up and top-down priming respectively. This gives rise to a 'spreading activation' of hypotheses, starting at the lowest level of received observations, and spreading up the levels of abstraction, and down again to their top-down predictions. Some of these top-down predictions will eventually reach the lowest level of perception and provide us with the 'what chords comes next' prediction that is our ultimate goal. Note then that hypotheses from the exhaustive set now fall into *three* classes: instantiated-and-'true' (the current percepts, as in classical blackboards); instantiated-but-'false' (the fringe, now explicitly instantiated); and uninstantiated.

Priming may be done by any associative hashing-type function, which rapidly and approximately maps from a

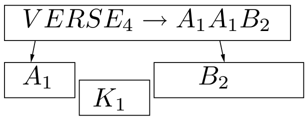


Figure 2: A missing child, and a floating replacement

central object to its fringe. In classic blackboard systems, Hopfield-like association networks and semantic networks were popular; classical neural networks are also useful hash functions for priming. The grammatical nature of the music minidomain allows an especially simple form of priming: when a (lengthened) term is asked to prime, we simply search the grammar for all rules containing it in their LHS or RHS, and instantiate them at the appropriate locations. Pruning (uninstantiation; 'fizzling' in Copycat) is performed in two ways. First, *Conway* pruning (after Conway's Game of Life) removes hypotheses which are 'false' and have no 'true' neighbors. Second, a garbage-collection sweep removes any resulting islands of nodes having no chain of connectivity to observed data.

In music as with most scene domains, it is unlikely that an object (hypothesis) would exist without one of its constituent parts. Initially it may appear that the model lacks a mechanism for penalizing such omissions. In a visual example, a $TABLE$ with a missing LEG is improbable. But if there was data and *instantiated* hypotheses only about three legs, and the scene probability was $\pi^{\emptyset}(TABLE)\prod_{i=1}^{3}P(LEG_i|TABLE)$, then this scene would be assigned a *higher* probability than a similar scene with data about four legs, because the latter scene would include an extra term in the product. To handle such cases, we may wish to include extra probability factors into the joint (as with keys) which specifically penalize the absence of components. The former visual scene would then be assigned probability $\frac{1}{Z}\pi^{\emptyset}(TABLE)\phi_{mc}\prod_{i=1}^{3}P(LEG_i|TABLE)$ where ϕ_{mc} is an explicit 'missing component' penalty. Early implementations of our system included such penalties, but they are in fact superfluous in the music minidomain because we are guaranteed that every bar will contain some chord, and require some explanation. Fig. 2 shows an example: the hypothesis corresponding to rule $VERSE_4 \rightarrow A_1A_1B_2$ is instantiated, but one of the A structures is missing, and a free-floating K takes its place in explaining that bar. However for the K to be 'true' it must have been generated by the free-float prior, which is small compared to the top-down prior on its rival A, so the scene as a whole does receive a penalty. In practice, the A will generally be instantiated anyway if the parent $VERSE$ is 'true' due to the fringe priming method, in which case a penalty will be received because $P(A|VERSE)$ is high so $1 - P(A|VERSE)$ is low.

Inference

We have used hypothesis states 'true' and 'false' informally, as the discussion has been on the network structure and construction. We now consider inference over the structures, which gives meaning to these node states. In the present implementation, hypotheses contain an abstract *inferlet* object, which provides a standard event-based API which may be implemented by various inference algorithms. Events include 'hypothesis just created', 'request made to run a local inference step (fire)', and 'tell me if you are 'true' or 'false''.

The nearest inference method to classical blackboard systems is annealed Gibbs sampling, in which each instantiated node has a single Boolean state, updated according to the Gibbs probabilities given its Markov blanket. In this case, 'true' and 'false' are simply these Boolean Gibbs states at the moment the node is queried. We extend standard Gibbs to sample from *clusters* of coparents rather than single nodes. The rivalry links impose a strong form of causal independence on the network: we know a priori that pairs of rivals will never be on together, so we need not consider these scenarios – they are not part of the 'feasible set'. Care is taken to ensure that all probabilities are soft to ensure detailed balance of the sampler.

Unlike Copycat, ThomCat runs online, updating its percept on the arrival of each new bar (about 2s of music). Inference runs under an annealing cycle: at the initial arrival of the chord likelihood data, the temperature is raised to its maximum. Sampling and cooling run during the bar, and a schedule is chosen so that it reaches near-freezing just before the arrival of the next bar. At this point the Gibbs distribution is almost a Dirac Delta, at a local (and hopefully global) maximum scene probability. An additional 'collapse' step is performed which sets this to an exact Delta, yielding the MAP estimate.

We have implemented two other inference algorithms, Bethe and Variational Bayes (Bishop 2006), using different inferlet classes. These algorithms maintain local distributions $Q_i(h_i)$ over hypotheses h_i rather than simple Boolean states, and are updated according to the Markov blankets $mb(h_i)$. Variational Bayes locally minimizes the relative entropy $KL[Q||P]$ by node updates of the form

$$Q(h_i) \leftarrow \exp\langle \log P(h_i|mb(h_i))\rangle_{Q(mb(h_i))}$$

and the Bethe variational method assumes the Bethe approximation to the free energy leading to updates of the form

$$Q(h_i) \leftarrow \langle P(h_i|mb(h_i))\rangle_{Q(mb(h_i))}.$$

(The Bethe updates are the same as Pearl used in polytree-structured networks, but applied to the loopy case as an approximation.) In noisy-OR networks, the expectations lack analytic solutions and must be computed by brute-force summation over the $2^{|mb(h_i)|}$ configurations of the Boolean Markov blanket nodes (The QMR approximation of (Jaakkola & Jordan 1999) gives a method for faster approximation in particular cases of noisy-OR messages, which could perhaps be extended to ThomCat's networks). The notion of hypotheses being 'true' or 'false' as used in the priming mechanism now becomes vaguer, and we use the heuristic that hypotheses with $Q_i(h_i) > 0.5$ are classed as 'true' for this purpose only. Many typical grammars have troublesome symmetries in their generating probabilities, which can cause variational methods to get stuck between two solutions, or converge very slowly. Empirically it was found that adding a small stochastic component to each node's $P(h_i|pa(h_i))$ at each step is a useful way to break these symmetries and allow convergence to a single solution.

Attention

We require that inference complexity does not increase over time as the scene grows, so we cannot keep growing the network and running inference on all of it as more and more data is received. Rather, it is better to focus *attention* on a fixed-size area of interest. Attention in this case means computing power used to run inference on nodes. It is not usually useful to focus on distant past events, rather we should focus on the region around the current events. However occasionally we may have to backtrack, as both current and past events are influenced by their shared high-level parents. It is also likely that in these 'mind-wandering' moments a higher temperature will be needed to break out of any minima: we must allow inference to become more fluid and risk-taking at these times. This could occasionally result in a minor 'paradigm shift' where the large-scale structure of the perceptual theory is overturned and replaced with something better. Similarly, for some tasks we may need to make time for 'blue-sky' thinking about objects in the distant future: when driving a car or preparing a jazz solo it is useful to have at least a rough map of where the whole journey is going in order to inform percepts and plans about immediate actions. But these distant past and future percepts change rarely – because little recently-received evidence is relevant for changing them – so do not need to be attended to often.

In the music minidomain, the region of interest consists of a few bars either side of the most recently received bar: this is where it is worth applying most effort to get good MAP approximations. We typically take six bars of the recent past, and two bars of the immediate future as the 'window' of attention. Hypotheses having any part inside the window are treated as active, and their inferlets participate in inference. Interactions between attention and priming and pruning need to be handled carefully. Fig. 3 shows an example of the window around the current time. It is important that the high level nodes such as $VERSE$ still receive likelihoods from out-of-attention past events that are 'true', such as the leftmost A structure. Such information acts as part of the prior when interpreting the current window. However, 'false' past hypotheses can be pruned once they fall outside the window, as they contribute no such prior and can never become 'true' again (unless a movable, 'mind-wandering' window was implemented – but in that case they would become re-primed when their region is attended to.) Similarly, new hypotheses are not primed outside the window, including in the future. We do however prime hypotheses in the *near* future inside the window, such as the K, f and am of fig. 3. Inference runs as usual on these hypotheses (though there are no *ObsBars* yet to provide bottom-level likelihoods) and the inferred state gives a prediction of the future, which could be used for example to schedule automated accompaniment actions to play along with the performer. The window moves along by one bar each time an *ObsBar* is received – at the same time as the annealing schedule is reset to its starting maximum temperature.

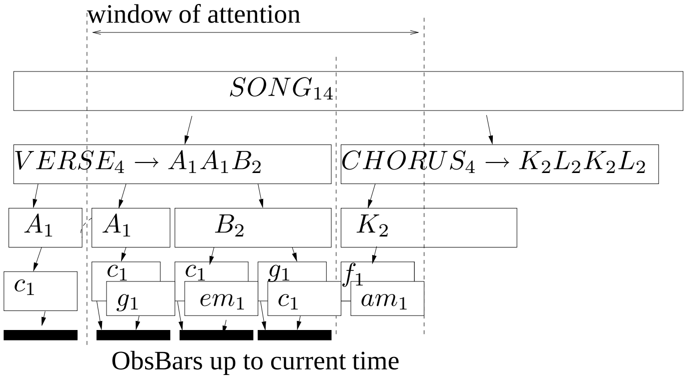


Figure 3: Attention includes the recent past and near future.

Practical issues

After naive implementation of the system with simple data structures and profiling, it became apparent that structural algorithms were consuming more processor time than inference proper, particularly during the single task of searching for pairs of rivals in a global hypothesis list. A large speed gain was made by constructing a central 'thalamus' topographical map of the temporal space, and using it to cache the locations of all hypotheses to allow rapid lookup of rivals. A similar performance gain was made by pre-caching hash maps of all possible primes from each l-grammar rule in a 'hippocampus' to avoid online lookup. The structures are so-called due to their possible suggested similarly to the corresponding biological structures. Once these speedups were implemented, the number of inference rounds per structural update was balanced so as to make the structural updates insignificant with respect to inference proper.

Results

Fig. 4 shows snapshots of the Gibbs state during a Gibbs inference run, using the grammar presented earlier. A deliberate mistake was introduced in the performance: the g chord in bar two has the highest likelihood, where the $SONG$ should generate a c. (a), (b) and (c) show annealing steps on the first five bars. The soft probabilities allow some pairs of rival nodes to be true together at high temperatures, but as the network cools this becomes less probable. Its final state (c) consists of a single free-floating $SONG$, causing a $VERSE$ component which causes riffs and chords. The erroneous g chord is perceived *as* a c due to large top-down pressures. The global KEY is identified as $Cmajor$, further encouraging c chords especially. (d) shows the state later in the performance, when attention has moved along with newly received bars, and now focuses on the new current bar, 17. A new round of annealing has just begun, so the window is at high temperature. Past hypotheses outside the window have been pruned if false, and frozen if true. In the latter case they still contribute likelihood to the large scale $SONG$ structures. The window thus contains a denser concentration of hypotheses undergoing inference, as the full fringe of primed objects is active. By this stage in the performance, the prior information from the past is very strong so correct convergence occurs in just a few annealing steps following each new bar, if the performance coheres with the current percepts. The attention window includes two bars of the near future, which give the current prediction of what happens next. We do not yet have significant comparison results, but early trials suggest Gibbs tends to beat variational methods given similar time and hand-optimized annealing schedules. This is perhaps due to large correlations between node states, not well captured by variational methods.

Discussion

We have presented a model of hierarchical temporal perception in a musical minidomain. The model is intended to be generalizable to other forms of scene analysis – such as vision – and the minidomain was selected to include general features such as global context sensitivity and free-floating structures. The blackboard heuristics of priming and pruning are similar to those of Copycat, as are the Gibbs acceptance probabilities and annealing, but we have placed these ideas on a Bayesian foundation, viewing them as approximation heuristics to exact probabilistic MAP inference. The minidomain uses discretized bars of time which simplifies hypothesis construction, but similar architectures could prime and prune in a continuous space, for example in visual scenes.

The current implementation is slow, requiring more than real-time for accurate perception. This is partly due to lack of software optimization, but Gibbs sampling is inherently slow, with even Copycat's simple *micro*domain tasks taking several minutes to run. Switching from sampling to variational methods removes the need to visit long sequences of states but requires evaluation of exponential configurations of Markov Blankets. It is possible that the inference algorithms could be further tuned to improve accuracy, for example using Metropolis-Hastings steps instead of Gibbs at high temperatures; QMR-style variational approximations; adjusted annealing schedules and tradeoff between attention window size and accuracy. But the problem is inherently NP-hard so will always require some approximation.

With further work, the MAP scenes could be used to generate and schedule *actions*. The task of learning and tuning new rules from scenes containing free-floaters could also be researched. We touched upon the idea of movable 'mind-wandering' attention: this could be explored, for example if the listener realizes it made a mistake in the distant past, it may 'look' back to what happened, and *reconstruct* the past in the light of later data. Reconstruction is literal as we prune away most past hypotheses when attention moves on: so some limited memory of past events (either the low level data or the high-level structures that were constructed) could be used to start the inference, which could then also incorporate messages from later events.

We tentatively suggest that parts of the ThomCat architecture could be implemented neurally as theories of biological computation. The cortical column is hypothesized to function as some form of Bayesian network node, and we suggest an analogous role to ThomCat's structural hypotheses. This shows a clear distinction between 'mean-field' and 'spike coding' neural models analogous to variational and Gibbs inference. Gibbs-like 'spreading activation' has been observed in V1 and Thalamus. We found the use of a central 'thalamus' especially useful to speed up rivalry lookups, and an associative 'hippocampus' which primes fringe concepts ready for inference. Similar 'hippocampal' associations could be useful in storing episodic memories of the type required to reconstruct the past during 'mind-wandering' discussed above.

Like Copycat, ThomCat is intended to capture and refine concepts about human perception, including the nature of priming and pruning, attention, rivalry, and perception of time and in time. It is a relativist, constructivist model, in which different subjects may optimally perceive the same data *as* different scenes depending on their known concepts (in this case, grammatical rules) and priors within those concepts. A strange consequence of this approach to music perception in particular is that composition is a relatively simple task in comparison to listening well: a good listener is one whose prior grammar is accurate for the genre, and who does therefore not expect to hear unmusical scenes. Such a listener should then be able to compose within the genre simply by sampling forwards from the prior distribution – a much simpler task than NP-hard 'backwards' inference! (Fox 2006) used a ThomCat-like grammar to create variations on existing compositions parsed by human experts – this approach could perhaps be fused with ThomCat to automate the perceptual part of the compositional process – leaving the acceptance function still performed by humans or to be further automated.

Python source code for ThomCat is available under the GPL from `5m.org.uk`.

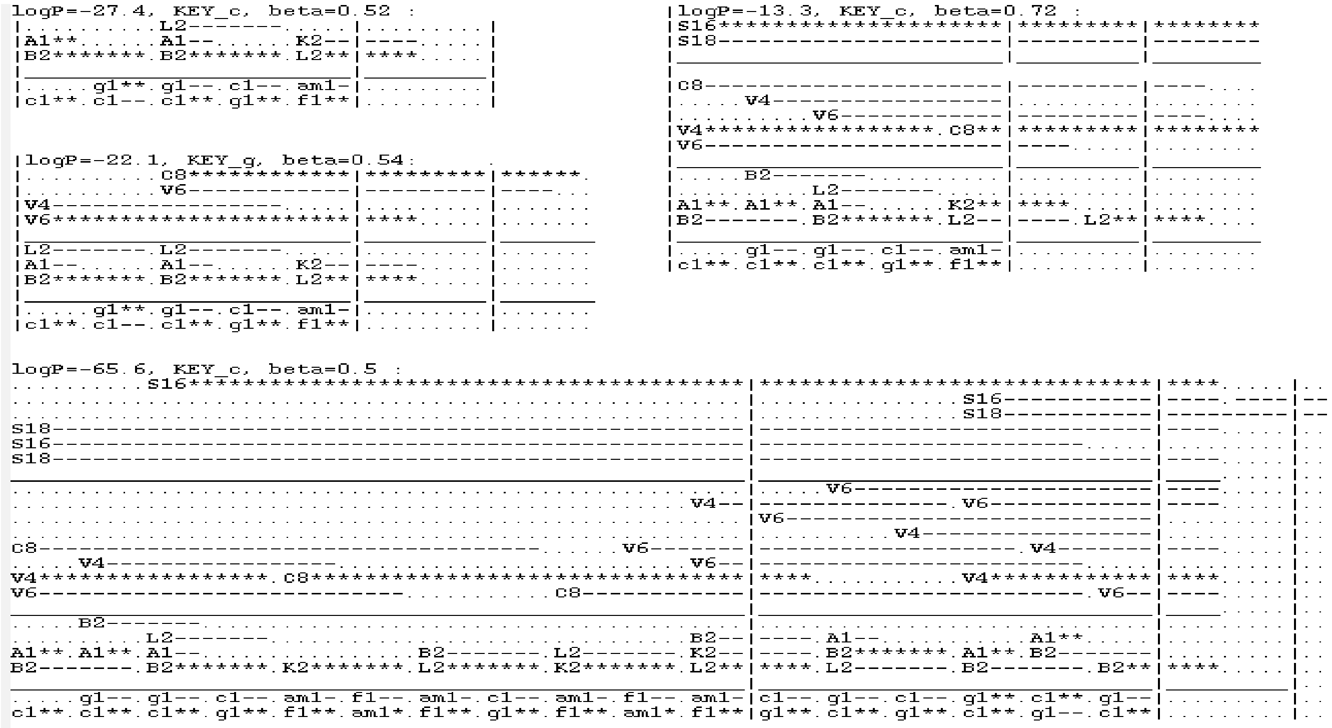
Appendix: lengthened grammar compilation To convert a SCFG to an l-grammar we use a novel message-passing scheme. First, a simple algorithm constructs all possible lengths for each symbol, and creates all possible lengthened rules, with no probability information. We then performing the following for each of these rules.

```
create top level hypotheses node from the rule
while any leave nodes contain nonterminals do
    for each leaf node $l$ do
        $x \leftarrow$ the leftmost nonterminal in $l$
        lookup all rules which rewrite $x$
        for each of these rules $r$ do
            create child of $l$: RHS of $r$ replaces $x$
        end for
    end for
end while
```

We then perform a top-down, dissipative (α) round of message passing followed by an upwards agglomerative (β) round with $\alpha_n = \alpha_{par(n)} T_{G(par(n) \rightarrow G(n))}$ and

$$\beta_n = \left\{ \begin{array}{r} \sum_{m \in ch(n)} \beta_m : \text{for non-leaves} \\ \alpha_n : \text{for leaves} \end{array} \right.$$

where n ranges over nodes and we introduce the notation $G(par(n)) \rightarrow G(n))$ to indicate the rule R in the unlengthened grammar G whose LHS is the symbol in $par(n)$ which was substituted in the transition to the RHS of R. At the end of this process we obtain β values in the first layer of chil-



```
logP=-27.4, KEY_c, beta=0.52 :
|..........L2-------.....|.........|
|A1**......A1--......K2--|----.....|
|B2*******.B2*******.L2**|****.....|
|________________________|_________|
|.....g1**.g1--.c1--.am1-|.........|
|c1**.c1--.c1**.g1**.f1**|.........|
```

```
|logP=-22.1, KEY_g, beta=0.54:           .
|..........C8************|*********|******.
|..........V6------------|---------|----...
|V4-----------------.....|.........|.......
|V6**********************|****.....|.......
|________________________|_________|_______
|L2-------.L2-------.....|.........|.......
|A1--......A1--......K2--|----.....|.......
|B2*******.B2*******.L2**|****.....|.......
|________________________|_________|_______
|.....g1**.g1--.c1--.am1-|.........|.......
|c1**.c1--.c1**.g1**.f1**|.........|.......
```

```
|logP=-13.3, KEY_c, beta=0.72 :
|S16*********************|*********|********
|S18---------------------|---------|--------
|________________________|_________|________
|
|C8----------------------|---------|----....
|.....V4-----------------|.........|........
|..........V6------------|---------|----....
|V4*****************.C8**|*********|********
|V6----------------------|----.....|........
|________________________|_________|________
|.....B2-------..........|.........|........
|..........L2-------.....|.........|........
|A1**.A1**.A1--......K2**|****.....|........
|B2-------.B2*******.L2--|----.L2**|****....
|________________________|_________|________
|.....g1--.g1--.c1--.am1-|.........|........
|c1**.c1**.c1**.g1**.f1**|.........|........
```

```
logP=-65.6, KEY_c, beta=0.5 :
..........S16*****************************************|*****************************|****.....|..
......................................................|...............S16-----------|----.----|--
......................................................|...............S18-----------|---------|--
S18---------------------------------------------------|-----------------------------|----.....|..
S16---------------------------------------------------|------------------------.....|.........|..
S18---------------------------------------------------|-----------------------------|----.....|..
______________________________________________________|_____________________________|____.....|..
......................................................|.....V6----------------------|----.....|..
..................................................V4--|--------------.V6------------|----.....|..
......................................................|V6---------------------------|.........|..
......................................................|..........V4-----------------|.........|..
C8------------------------------------......V6--------|-------------------.V4-------|----.....|..
.....V4-----------------..........................V6--|------------------------.....|.........|..
V4*****************.C8********************************|****...........V4************|****.....|..
V6---------------------------...........C8------------|------------------------.V6--|----.....|..
______________________________________________________|_____________________________|____.....|..
.....B2-------........................................|.............................|.........|..
..........L2-------...............................B2--|----.A1--...........A1**.....|.........|..
A1**.A1**.A1--................B2-------.L2-------.K2--|----.B2*******.A1**.B2-------|.........|..
B2-------.B2*******.K2*******.L2*******.K2*******.L2**|****.L2-------.B2-------.B2**|****.....|..
______________________________________________________|_____________________________|_________|..
.....g1--.g1--.c1--.am1-.f1--.am1-.c1--.am1-.f1--.am1-|c1--.g1--.c1--.g1**.c1**.g1--|.........|..
c1**.c1**.c1**.g1**.f1**.am1*.f1**.g1**.f1**.am1*.f1**|g1**.c1**.g1**.c1**.g1--.c1**|.........|..
```

Figure 4: Snapshots of states during inference. Instantiated hypotheses are shown as hyphens (false states) and asterisks (true states). The vertical lines show the start of the attention window, the present time, and the end of the window, from left to right.

dren which summarize the the probabilities of those parameterizations of the top-level rule, and sum to 1 (so β at the root node is 1). The child β values are the required transition probabilities to insert into the lengthened grammar. Having formed the tree above and passed messages, it is simple to obtain the lengthened $\pi^{\emptyset}$ by $\pi_{\emptyset}(S_{9.5}) = \pi_{\emptyset}(S).\beta(S_{9.5})$. This works because the top-level $\beta(LR)$ for lengthened rules LR compute the prior $P(LR|R)$ where R is the corresponding unlengthened rules.

References

Bishop, C. 2006. *Pattern Recognition and Machine Learning*. Springer.

Crick, F. 1984. Function of the thalamic reticular complex: The searchlight hypothesis. *Proceedings of the National Academy of Sciences* 81:4586–4590.

Dagum, P., and Luby, M. 1993. Approximating probabilistic inference in bayesian belief networks is NP-hard. *Artificial Intelligence* 60(1):141–153.

Dennett, D. C. 1998. *Consciousness Explained*. Penguin.

Fox, C. 2006. Genetic hierarchical music structures. In *FLAIRS06*.

Hofstadter, D. 1987. A non-deterministic approach to analogy, involving the ising model of ferromagnetism. In *The Physics of Cognitive Processes*. ed. E. Caianiello. World Scientific.

Hofstadter, D. 1995. *Fluid Concepts and Creative Analogies*. Basic Books.

Jaakkola, T., and Jordan, M. 1999. Variational methods and the QMR-DT database. *Journal of Artificial Intelligence Research* 10:291–322.

James, W. 1907. *Pragmatism: A New Name for Some Old Ways of Thinking*. Reprinted Dover, 1995.

Libet, B. 2004. *Mind Time: the temporal factor in consciousness*. Harvard Press.

Mitchell, M. 1993a. *Analogy-Making as Perception*. MIT.

Mitchell, M. 1993b. Copycat source code. http://web.cecs.pdx.edu/ mm/ccat-src/.

Murphy, K. 2001. Linear time inference in hierarchical HMMs. In *NIPS*. MIT Press.

Pearl, J. 2000. *Causality*. Cambridge University Press.

Whorf, B. L. 1956. *Language, Thought and Reality*. MIT Press.

Wiggs, C., and Martin, A. 1998. Properties and mechanisms of perceptual priming. *Current Opinion in Neurobiology (CNS)*.

Zhang, N., and Poole, D. 1996. Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*.