Can the Computer Learn to Play Music Expressively?

Christopher Raphael*
Department of Mathematics and Statistics,
University of Massachusetts at Amherst,
Amherst, MA 01003-4515,
raphael@math.umass.edu

Abstract

A computer system is described that provides a real-time musical accompaniment for a live soloist in a piece of non-improvised music. A Bayesian belief network is developed that represents the joint distribution on the times at which the solo and accompaniment notes are played as well as many hidden variables. The network models several important sources of information including the information contained in the score and the rhythmic interpretations of the soloist and accompaniment which are learned from examples. The network is used to provide a computationally efficient decision-making engine that utilizes all available information while producing a flexible and musical accompaniment.

1 Introduction

Our ongoing work, "Music Plus One," develops a computer system that plays the role of musical accompanist in a piece of non-improvisatory music for soloist and accompaniment. The system takes as input the acoustic signal generated by the live player and constructs the accompaniment around this signal using musical interpretations for both the solo and accompaniment parts learned from examples. When our efforts succeed, the accompaniment played by our system responds both flexibly and expressively to the soloist's musical interpretation.

We have partitioned the accompaniment problem into two components, "Listen" and "Play." Listen takes as input the acoustic signal of the soloist and, using a hidden Markov model, performs a real-time analysis of the signal. The output of Listen is essentially a running commentary on the acoustic input which identifies note boundaries in the solo part and communicates these events with variable latency. The strengths of our HMM-based framework include automatic trainability, which allows our system automatically adapt to changes in solo instrument and acoustic environment; the computational efficiency that comes with dynamic programming recognition algorithms; and accuracy, due in part to Listen's ability to delay the identification of an event until the local ambiguity is resolved. Our work on the Listen component is documented in [1].

The Play component develops a Bayesian belief network consisting of hundreds of Gaussian random variables including both observable quantities, such as note onset times, and unobservable quantities, such as local tempo. The belief network can be trained during a rehearsal phase to model both the soloist's and accompanist's interpretations of a specific piece of music. This model can then be used in performance to compute *in real time* the optimal course of action given the currently available data. We focus here on the Play component which is the most

*This work is supported by NSF grant IIS-9987898.

challenging part of our system. A more detailed treatment of some aspects of this work is given in [2].

2 Knowledge Sources

As with the human musical accompanist, the music produced by our system must depend on a number of different knowledge sources. From a modeling point of view, the primary task is to develop a model in which these disparate knowledge sources can be expressed in terms of some common denominator. We describe here the three knowledge sources we use.

We work with non-improvisatory music so naturally the musical score, which gives the pitches and relative durations of the various notes, as well as points of synchronization between the soloist and accompaniment, must figure prominently in our model. The score should not be thought of as a rigid grid prescribing the precise times at which musical events will occur; rather, the score gives the basic elastic material which will be stretched in various ways to to produce the actual performance. The score simply does not address most interpretive aspects of performance.

Since our accompanist must follow the soloist, the output of the Listen component, which identifies note boundaries in the solo part, constitutes our second knowledge source. While most musical events, such as changes between neighboring diatonic pitches, can be detected very shortly after the change of note, some events, such as rearticulations and octave slurs, are much less obvious and can only be precisely located with the benefit of longer term hindsight. With this in mind, we feel that any successful accompaniment system cannot synchronize in a purely responsive manner. Rather it must be able to predict the future using the past and base its synchronization on these predictions, as human musicians do.

While the same player's performance of a particular piece will vary from rendition to rendition, many aspects of musical interpretation are clearly established with only a few repeated examples. These examples, both of solo performances and human renditions of the accompaniment part constitute the third knowledge source for our system. The solo data is used primarily to teach the system how to predict the future evolution of the solo part (and to know what can and cannot be predicted reliably). The accompaniment data is used to learn the musicality necessary to bring the accompaniment to life.

We have developed a probabilistic model, a Bayesian belief network, that represents all of these knowledge sources through a jointly Gaussian distribution that contains hundreds of random variables. The observable variables in this model are the estimated soloist note onset times produced by Listen and the directly observable times for the accompaniment notes. Between these observable variables lie several layers of hidden variables that describe unobservable quantities such as local tempo, change in tempo, and rhythmic stress.

3 The Solo Model

We model the time evolution of the solo part as follows. For each of the solo notes, indexed by $n = 0, \ldots, N$, we define a random vector representing the time, t_n, (in seconds) and the "tempo," s_n, (in secs. per beat) for the note. We model this sequence of random vectors through a random difference equation:

$$\begin{pmatrix} t_{n+1} \\ s_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & l_n \\ 0 & 1 \end{pmatrix} \begin{pmatrix} t_n \\ s_n \end{pmatrix} + \begin{pmatrix} \tau_n \\ \sigma_n \end{pmatrix} \quad (1)$$

$n = 0, \ldots, N-1$, where l_n is the musical length of the n^{th} note in beats and the $\{(\tau_n, \sigma_n)^t\}$ and $(t_0, s_0)^t$ are mutually independent Gaussian random vectors.

The distribution of the $\{\sigma_n\}$ will tend concentrate around 0 which expresses the notion that tempo changes are gradual. The means and variances of the $\{\sigma_n\}$ show where the soloist is speeding-up (negative mean), slowing-down (positive mean), and tell us if these tempo changes are nearly deterministic (low variance), or quite variable (high variance). The $\{\tau_n\}$ variables describe stretches (positive mean) or compressions (negative mean) in the music that occur without any actual change in tempo. Thus, the distributions of the $(\tau_n, \sigma_n)^t$ vectors characterize the solo player's rhythmic interpretation. Both overall tendencies (means) and the repeatability of these tendencies (covariances) are expressed by these vectors.

The solo model can be summarized as

$$x_{n+1}^{\text{solo}} = A_n x_n^{\text{solo}} + \xi_n^{\text{solo}} \qquad (2)$$

for $n = 0, \ldots, N-1$ where $x_n^{\text{solo}} = (t_n, s_n)^t$, $\xi_n^{\text{solo}} = (\tau_n, \sigma_n)^t$ and A_n is the 2x2 matrix in Eqn. 1. In Eqn. 2 the $\{\xi_n^{\text{solo}}\}$ and x_0^{solo} are mutually independent Gaussian random vectors.

3.1 Training the Solo Model

The training of the solo distribution revolves around the estimation of the $\xi_n^{\text{solo}} = (\tau_n, \sigma_n)^t$ vectors. Since these vectors cannot be observed directly, we have a missing data problem. Let x_n^{obs} be the n^{th} note estimate produced by Listen which we assume depends only on the "true" note time, t_n. We model

$$x_n^{\text{obs}} = B x_n^{\text{solo}} + \xi_n^{\text{obs}} \qquad (3)$$

where the matrix $B = (1, 0)$ and the $\{\xi_n^{\text{obs}}\}$ are independent 0-mean Gaussian variables with known variances. The $\{x_n^{\text{solo}}\}$, $\{\xi_n^{\text{solo}}\}$ and $\{x_n^{\text{obs}}\}$ variables have a dependency structure expressed in the directed acyclic graph (DAG) of Figure 1 which qualitatively describes Eqns. 2 and 3; this graphical representation of dependency structure provides the key to the training algorithm. Suppose we have several solo performances of a section of music. Having observed the times generated by Listen for each performance, (the darkened circles in the figure), we can use the message passing algorithm to compute posterior distributions on the $\{\xi_n^{\text{solo}}\}$ and x_0^{solo} variables. With these posterior distributions in hand, the EM algorithm [3] provides a simple updating scheme guaranteed to increase the marginal likelihood of the observations at each iteration.

Training the solo evolution model allows our system to predict the future evolution of the solo part and adjust the accompaniment accordingly. It is in this way that we incorporate the soloist's rhythmic interpretation and follow the soloist by anticipating future events. The actual output of our system is, of course, the accompaniment; if the accompaniment is to be played in a musically satisfying way it must do much more than merely synchronize with the soloist. We now describe how we construct the joint probabilistic model on the solo *and* accompaniment parts.

4 Adding the Accompaniment

Our accompaniments are generated through the MIDI (Musical Instrument Digital Interface) protocol, and thus each accompaniment note is described by three parameters: An onset time, a damping time, and an initial velocity (the MIDI term for volume). The damping times can be computed as a function of the onset times in a straight-forward manner: In a *legato* passage each note can be damped when the next note begins; in a *staccato* passage the notes can be damped at prescribed intervals after the note onsets. The MIDI velocities contribute more significantly to the musical quality of the performance so we have elected to learn these from actual MIDI performance data. While interdependencies might well exist between musical timing and dynamics, we have elected to separate our estimation of velocities from the onset times. To this end we learn the velocities by partitioning the accompaniment part into phrases and modeling the velocities on each phrase as a function of a small number of predictor variables such as pitch, score position, etc. These velocities are then used in a deterministic fashion in subsequent performances. The MIDI onset times are, by far, the most important variables since they completely determine the degree of synchronicity between the solo and accompaniment part and largely determine the expressive content of the accompaniment. These are the variables we model jointly with the solo model variables described in the previous section.

We begin by defining a model for the accompaniment part alone that is completely analogous to the solo model. Specifically, we define a process

$$x_{m+1}^{\text{accom}} = C_m x_m^{\text{accom}} + \xi_m^{\text{accom}}$$

for $m = 0, \ldots, M-1$ where the $\{x_m^{\text{accom}}\}$ are (time,tempo) variables for the accompaniment notes, where x_0^{accom} and the $\{\xi_m^{\text{accom}}\}$ are mutually independent Gaussian vectors that express the accompaniment's rhythmic interpretation, and where the $\{C_m\}$ are matrices analogous to the $\{A_n\}$ of Eqn. 2. The means and covariances of the x_0^{accom} and $\{\xi_m^{\text{accom}}\}$ variables are then learned from MIDI performances of the accompaniment using the EM algorithm as with solo model. One might think of the x^{accom} process as representing the "practice room" distribution on the accompaniment part — that is, the way the accompaniment plays when issues of synchronizing with the soloist are not relevant.

We then combine our solo and accompaniment models into a joint model containing the variables of both parts. In doing so, the solo and accompaniment models play asymmetric roles since we model the notion that the accompaniment must follow the soloist. To this end we begin with the solo model exactly as it has been trained from examples as in Eqn. 2. We then define the conditional distribution of the accompaniment part *given* the solo part in a way that integrates the rhythmic interpretation of the accompaniment as represented in the x^{accom} process *and* the desire for synchronicity.

Consider a section of the accompaniment part "sandwiched" between two solo notes as in the upper left panel of Figure 2. For simplicity we assume that m_l and m_r are the indices of the leftmost and rightmost accompaniment notes and that $n(m_l)$ and $n(m_r)$ are the indices of the coincident solo notes of Figure 2. The accompaniment notes $x_{m_l+1}^{\text{accom}}, \ldots, x_{m_r-1}^{\text{accom}}$ have a conditional distribution given $x_{m_l}^{\text{accom}}$ and $x_{m_r}^{\text{accom}}$ that can be represented as follows.

We modify the graph corresponding to the joint distribution on $x_{m_l}^{\text{accom}}, \ldots, x_{m_r}^{\text{accom}}$ by dropping the directions of the edges, adding an edge between $x_{m_l}^{\text{accom}}$ and $x_{m_r}^{\text{accom}}$, and triangulating the graph as in the upper right panel of Figure 2. The joint distribution on $x_{m_l}^{\text{accom}}, \ldots, x_{m_r}^{\text{accom}}$ can be represented on this modified graph by associating each potential in the original graph with a corresponding clique in the modified graph. Then, after a round of message passing, we obtain the equilibrium representation and from this equilibrium we write the joint distribution on $x_{m_l}^{\text{accom}}, \ldots, x_{m_r}^{\text{accom}}$ by

$$\frac{\prod_{C \in \mathcal{C}} \phi_C}{\prod_{S \in \mathcal{S}} \phi_S}$$

where $\mathcal{C}$ and $\mathcal{S}$ are the cliques and separators in the clique tree and $\{\phi_C\}$ and $\{\phi_S\}$ are the clique and separator potentials corresponding to the marginal distributions on the indicated variables. Lauritzen [4] and Lauritzen and Jensen [5] provides two ways of implementing the message passing algorithm in this Gaussian context, although we employ our own method. By the construction of the graph, there will be a clique, C_{root}, containing $E = \{x_{m_l}^{\text{accom}}, x_{m_r}^{\text{accom}}\}$ and hence the joint distribution of the variables of E can be obtained from the equilibrium representation. We denote the Gaussian potential for this marginal by ϕ_E. Then the conditional distribution on $x_{m_l+1}^{\text{accom}}, \ldots, x_{m_r-1}^{\text{accom}}$ given $x_{m_l}^{\text{accom}}$ and $x_{m_r}^{\text{accom}}$ can then be written as

$$\frac{\prod_{C \in \mathcal{C}} \phi_C}{\phi_E \prod_{S \in \mathcal{S}} \phi_S}$$

A causal representation of this conditional distribution can be found by regarding C_{root} as the root of the tree and letting $S(C)$ be the "root side" separator for each clique other than C_{root}; we let $S(C_{\text{root}}) = E$. The desired causal representation is then

$$\prod_{C \in \mathcal{C}} \frac{\phi_C}{\phi_{S(C)}} \tag{4}$$

where each quotient represents the conditional distribution on $C \setminus S(C)$ given $S(C)$.

We then define our conditional distribution of the accompaniment, given the solo part, as follows. Let

$$x_{m_l}^{\text{cond}} = x_{n(m_l)}^{\text{solo}} + \xi_{m_l}^{\text{cond}} \tag{5}$$

$$x^{\text{cond}}_{m_r} = x^{\text{solo}}_{n(m_r)} + \xi^{\text{cond}}_{m_r}$$

where $\xi^{\text{cond}}_{m_l}$ and $\xi^{\text{cond}}_{m_r}$ are 0-mean random vectors with small covariances. Thus we represent the idea that the time and tempo of the accompaniment notes with indices m_l and m_r are small perturbations of the time and tempo for the coincident solo notes. We then define the variables $x^{\text{cond}}_{m_l+1}, \ldots, x^{\text{cond}}_{m_r-1}$ given $x^{\text{cond}}_{m_l}$ and $x^{\text{cond}}_{m_r}$ according to the causal representation of the conditional distribution of $x^{\text{accom}}_{m_l+1}, \ldots, x^{\text{accom}}_{m_r-1}$ given $x^{\text{accom}}_{m_l}$ and $x^{\text{accom}}_{m_r}$ shown in Eqn. 4. A pictorial description of this construction is given in the lower left panel of Figure 2.

Situations arise in which accompaniment notes cannot be sandwiched between a pair of coincident solo notes leading to several other cases that employ the basic idea described above. We will not describe these cases here. Figure 3 shows a DAG describing the dependency structure of a model corresponding to the opening measure of the Sinfonia of J. S. Bach's Cantata 12. The 2nd and 1st layers of the graph are the solo process and the output of Listen as described by Eqns 2 and 3. The 3rd layer denotes "phantom" nodes which arise when accompaniment notes are sandwiched between solo notes yet no coincident solo notes exist. The 4th layer shows the accompaniment notes that are coincident with solo notes as in Eqn. 5 The 5th layer shows the sandwiched accompaniment notes as in Eqn. 4. Finally, for each accompaniment vector (the 4th and 5th layers) we define a variable that deterministically "picks off" the time component of the vector. These variable compose the 6th layer of the graph. Only the top and bottom layers in this graph are directly observable.

5 Real Time Accompaniment

The methodological key to our real-time accompaniment algorithm is the computation of (conditional) marginal distributions facilitated by the message-passing algorithm. At any point during the performance some collection of solo notes and accompaniment notes will have been observed. Conditioned on this information we can compute the distribution on the next unplayed accompaniment note by passing a sequence of messages as in HUGIN's "Collect Evidence." The real-time computational requirement is limited by passing only the messages necessary to compute the marginal distribution on the pending accompaniment note. To this end, every time a model variable is observed all messages moving "away" from that variable are marked as "hot." Every time a message is passed the message is then marked as "cold." When computing the distribution on the pending accompaniment note only the "hot" messages are passed. Usually there are only a few of these.

Once the marginal of the pending accompaniment note is calculated we schedule the note accordingly. Currently we schedule the note to be played at the posterior mean time given all observed information, however other reasonable choices are possible. Note that this posterior distribution depends on all of the sources of information included in our model: The score information, all currently observed solo and accompaniment note times, the predicted evolution of future solo note times learned during the training phase, and the learned rhythmic interpretation of the accompaniment part.

The initial scheduling of each accompaniment note takes place immediately after the previous accompaniment note is played. It is possible that a solo note will be detected before the pending accompaniment is played; in this event the pending accompaniment note is rescheduled based on the new available information. The pending accompaniment note is rescheduled each time an additional solo note is detected until its current schedule time arrives, at which time it is finally played. In this way our accompaniment makes use of all currently available information.

Can the computer learn to play expressively? We presume no more objectivity in answering this question than we would have in judging the merits of our children. However, we believe that the level of musicality attained by our system is truly surprising. We hope that the interested reader will form an independent opinion, even if different from ours, and to this end we have made musical examples available on our web page. In particular, both a "practice room" accompaniment generated from our model and a demonstration of our accompaniment system in action can be heard at http://fafner.math.umass.edu/reverie.

References

[1] Raphael C. (1999), "Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 21, No. 4, pp. 360–370.

[2] Raphael C. , "A Probabilistic Expert System for Automatic Musical Accompaniment," *to appear in: Journal of Computational and Graphical Statistics.*

[3] Lauritzen S. L. (1995), "The EM Algorithm for Graphical Association Models with Missing Data," *Computational Statistics and Data Analysis*, Vol. 19, pp. 191–201.

[4] Lauritzen S. L. (1992), "Propagation of Probabilities, Means, and Variances in Mixed Graphical Association Models," *Journal of the American Statistical Association*, Vol. 87, No. 420, (Theory and Methods), pp. 1098–1108.

[5] Lauritzen S. L. and F. Jensen (1999), ' 'Stable Local Computation with Conditional Gaussian Distributions," *Technical Report R-99-2014*, Department of Mathematic Sciences, Aalborg University.

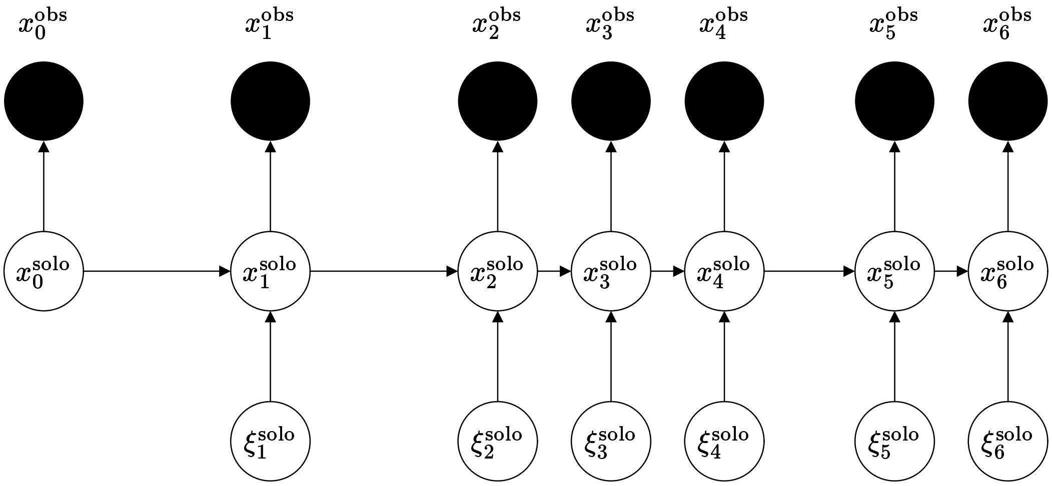

Figure 1: The dependency structure of the $\{x_n^{\rm solo}\}$,$\{\xi_n^{\rm solo}\}$, and $\{x_n^{\rm obs}\}$ variables. The variables with no parents, $x_0^{\rm solo}$ and the $\{\xi_n^{\rm solo}\}$, are assumed to be mutually independent and are trained using the EM algorithm. The horizontal placement of graph vertices in the figure corresponds to their times, in beats, as indicated by the score.

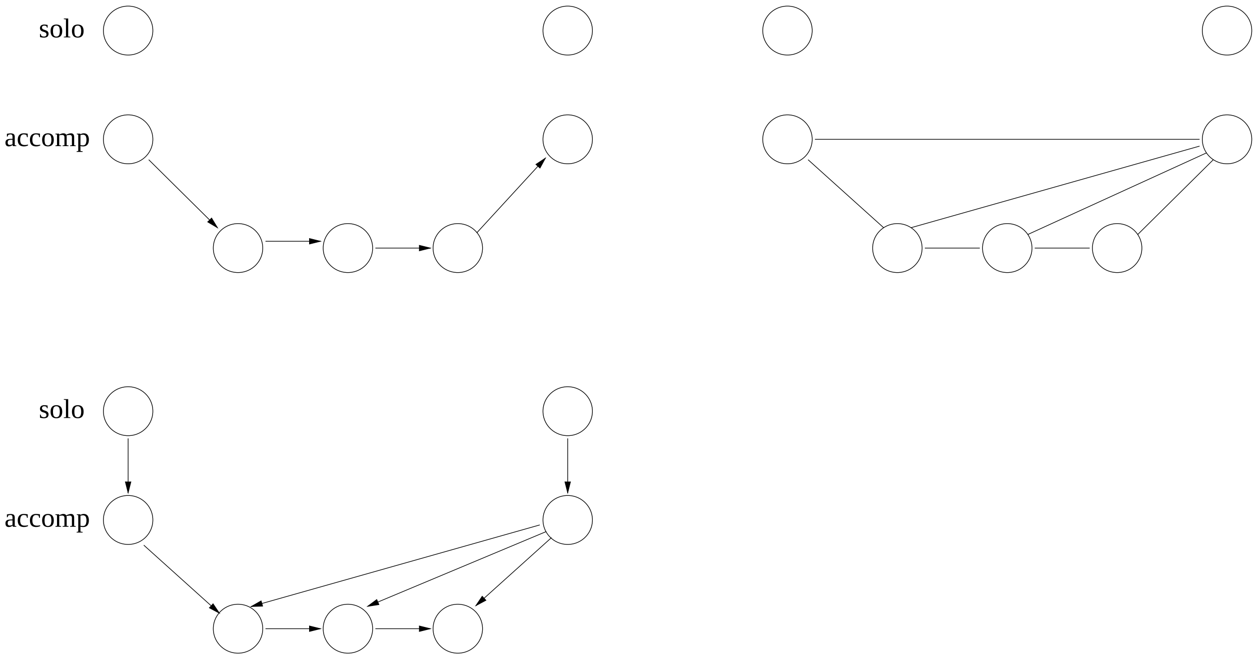

Figure 2: **Upper Left:** A sequence of 5 accompaniment notes, the first and last of which, $x_{m_l}^{\rm accom}$ and $x_{m_r}^{\rm accom}$, coincide with the solo notes $x_{n(m_l)}^{\rm solo}$ and $x_{n(m_r)}^{\rm solo}$. The conditional distribution of each vector given its predecessor is learned during a training phrase. **Upper Right:** An undirected graph of the same variables used for computing the joint distribution on $x_{m_l}^{\rm accom}$ and $x_{m_r}^{\rm accom}$. **Lower Left:** A directed graph showing the dependency structure for the conditional distribution of the $x_{m_l}^{\rm cond}, \ldots, x_{r_l}^{\rm cond}$ given $x_{n(m_l)}^{\rm solo}$ and $x_{n(m_r)}^{\rm solo}$.

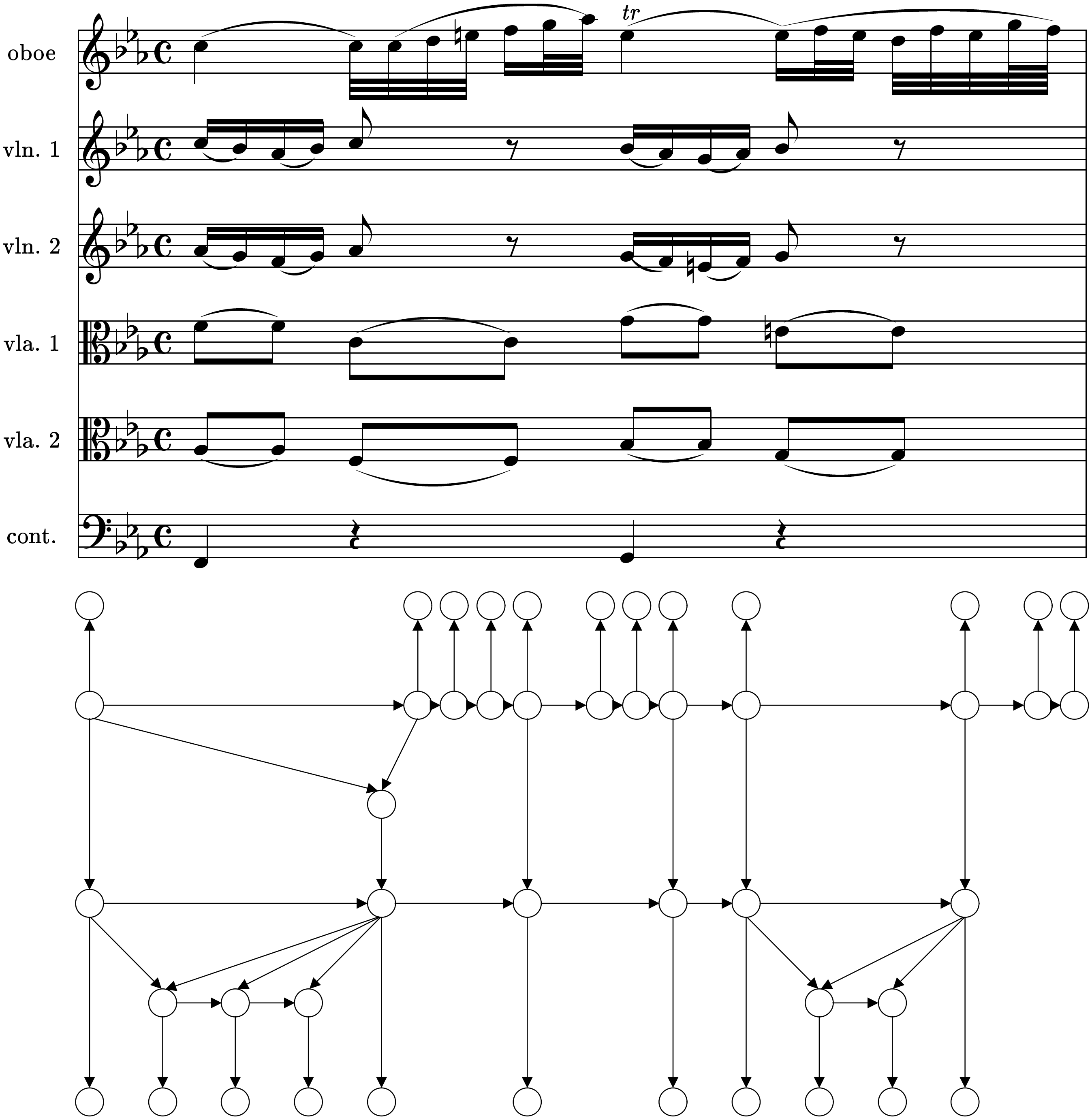

Figure 3: **Top:** The opening measure of the Sinfonia from J.S. Bach's Cantata 12. **Bottom:** The graph corresponding to the first 7/8 of this measures. The nodes in the 1st (top) layer correspond to the estimated solo note times that come from the Listen process $\{x_n^{\text{obs}}\}$; the 2nd layer represents the solo process $\{x_n^{\text{solo}}\}$; the 3rd layer represents the phantom nodes; the 4th layer represents the coincident accompaniment nodes; the 5th layer represents the sandwiched nodes; the 6th layer represents the actual accompaniment observation times.