information into account and the third performs joint key and chord estimation. All three systems have weights that are derived from conditional probabilities and only consider pairwise time dependencies. Added to the fact that all formulas in the MLNs are conjunctions of positive literals, this means that the resulting MNs instantiated from the MLNs can be equally well represented as DBNs (**?**).
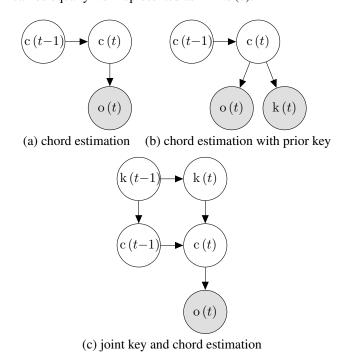

(a) chord estimation   (b) chord estimation with prior key

(c) joint key and chord estimation

Figure 2: Equivalent DBN representations of the systems presented in (**?**), where $o(t)$, $c(t)$ and $k(t)$ stand for respectively the chroma observation, chord and key at time t. Shaded nodes represent observed variables and white nodes are hidden variables.

The diagrams for those three systems are displayed in Figure 2. The formulations as DBNs will actually be more compact and elegant because continuous observations densities can be used and the observation probabilities no longer need to be unrolled in time. Moreover, the DBN can be reused multiple times by changing the observations presented to it, whereas the MLNs need to be redefined for every song because of the conflation between model and evidence. Another advantage of this formulation is that optimised inference algorithms can be used. For instance, we can now see that it is not surprising that Papadopoulos and Tzanetakis found the difference between the chord estimation MLN and a chord HMM to be statistically not significant (**?**), as they describe exactly the same system. The only difference is that the MLN description needs to resort to a more generally applicable inference algorithm because of the time-unrolled observations and suboptimal network layout, whereas the HMM can use the Viterbi algorithm for decoding (resulting in a speedup of orders of magnitude).

The second group of MLN examples is arguably more interesting, precisely because they cannot be formulated as
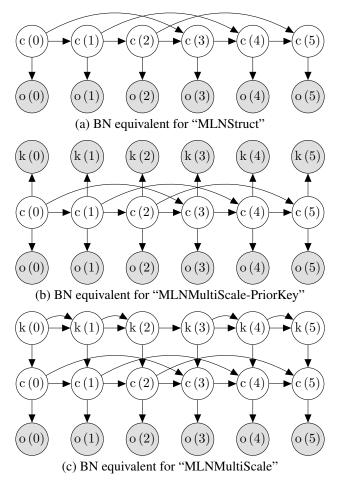

(a) BN equivalent for "MLNStruct"

(b) BN equivalent for "MLNMultiScale-PriorKey"

(c) BN equivalent for "MLNMultiScale"

Figure 3: Equivalent BN representations of the more complex systems presented in (**?**), where $o(t)$, $c(t)$ and $k(t)$ stand for respectively the chroma observation, chord and key at time t. Shaded nodes represent observed variables and white nodes are hidden variables. It is assumed that nodes $[0:2]$ and $[3:5]$ belong to the same measure and that the time segment pairs $(0, 3), (1, 4)$ and $(2, 5)$ are given as highly similar.

DBNs (but still as BNs). It comprises three of the systems presented in (**?**), one of which was previously presented in (**?**). The motivation behind them is the fact that one drawback of DBNs is their inability to take longer term dependencies between chords directly into account. We assume that we dispose of a number of pairs that indicate high similarity between its (non-adjacent) constituents. These similarity pairs can come from a prior structural analysis under the assumption that a repetition of a chorus, for instance, will contain the same chords as the previous chorus. Papadopoulos and Tzanetakis use manual structural annotations to derive these similarity pairs, such that they are binary (either similar or not). The repetitions are then exploited to improve the chord estimation. In this sense, this approach is comparable to other systems that take into account repetition, such

as (**?**; **?**). The difference is that here the knowledge about the repetitions is integrated probabilistically into the inference, whereas those previous approaches do a deterministic early fusion of the repeated observations. In essence, the three systems are constructed by unrolling the previous three DBN systems through time such that the similarity knowledge can be integrated by drawing additional edges between the similar time segments. Example graphs with six time segments are shown in Figure 3. The most complex example not only takes long term chord similarity based on structure into account, but also medium term key similarity by connecting keys in the same measure under the assumption that key changes within a measure are highly unlikely.

Inference in the resulting BNs will not be faster than in the optimal, propositionally described MNs (which have the same graphs, but with undirected edges instead), but the BNs have the advantage of being easier to interpret. The weights proposed in (**?**) for the similarity-based edges are fixed without justification or interpretation, presumably found through a parameter sweep. Because of the global normalisation in MN, this makes it hard to interpret the numbers. Even worse, the interpretation of the other weights (derived from conditional probabilities) will be affected by the addition of the supplemental structural edges, which are by definition unpredictable and song-dependent, and depend on the given evidence (**?**). The whole graph then quickly becomes opaque. In a BN, the structural edges will get a conditional probability assigned, which has to be normalised locally to preserve the total probability in each node. The optimal parameter still can – and likely has to – be found through a parameter sweep, but the result will be interpretable and will not affect the other parameters in unforeseeable ways. A BN will also make it easier, for example, to consolidate the key transition probabilities and the probabilistic tying of keys within the same measure in the last graph of Figure 3 into a single edge (currently they are represented by separate edges connecting the same nodes).

6 Conclusion and future work

In this paper, we studied MLNs for music analysis by analysing a couple of previously proposed example systems for chord estimation and reformulating them as (dynamic) Bayesian networks. The theoretical advantages of MLN are an easier, more powerful formulation of graphical models and an improvement in learning/inference algorithms through exploiting regularity in the networks. In practice, however, we saw that the MLN systems formulated in (**?**; **?**; **?**) are not optimally exploiting the capabilities of the MLN framework. This is partly because these formulations introduce unnecessary variables to create FOL where propositional logic would suffice, and partly because the lack of support for continuous observation distributions and sequences forces the use of convoluted workarounds. We furthermore showed that it is not necessary to resort to Markov networks to express the underlying ideas, but that Bayesian networks can use the same principles, with improved interpretability. We argue that because music signals are variable through time and causal, a directional left-to-right relationship arises naturally when modelling music,therefore the as-

sociated graphs will be chainlike. These conclusions are not specific to music, but apply to other fields as well, especially when sequential data with continuous observations is involved.

This is not to say that MLNs are not useful for solving a number of problems, even in the musical domain. Candidate problems for which MLNs would provide a good fit have only discrete variables and would ideally model static processes. Any data that is topographically organised would be suitable, as such a layout increases the chance of cyclical relationships and regularity that cannot be properly modelled by (dynamic) Bayesian networks. In the musical domain, one application that fulfils these requirements would be similarity discovery or clustering between artists, as long as no time-varying features are used.

Another case where MLNs can be helpful is when a large body of knowledge about the problem at hand already exists in first-order logic form. In contrast, the formulas in (**?**; **?**; **?**) are relatively simple, just describing relationships between node pairs. In this scenario, the related paradigm of Bayesian logic networks (BLNs) (**?**) might be an alternative if no bidirectional connections are needed. An implementation is available in the same ProbCog toolkit, which has the advantage that arithmetic are supported such that it is possible to impose $t_2 = t_1 + 1$ through a logical rule. Continuous observation distributions require the same workaround as with MLNs though.

Another alternative for MLNs we could explore in the future is ProbLog (**?**). The advantage of this approach is that it is built around a complete Prolog engine, which makes it easy to extend with custom functionality, such as arbitrary continuous distributions. In our first preliminary experiments, however, we already encountered problems to make it scale to the typical network sizes of our problems, despite recent advances in this aspect (**?**).