Shane Hoversten
Darrell Conklin, Music Generation from Statistical Models

This paper investigates applying various statistical techniques from computer science toward the goal of music composition. Most of these techniques have been widely deployed in natural language processing problems. All of them, in some general sense, use the idea of "sampling" some corpus, determining the regularities between the elements of that corpus, and then, using those statistics, either analyzing some new sample, or generating a new sample based on (or motivated by) the old. The "elements" to which I allude strike at the heart of this problem and, indeed, all statistical techniques. Figuring out what, exactly, are the elements of a domain will determine in large part the success one might have in modeling it statistically.

Any statistical approach must deal with the "sparse data" problem, meaning, in essence, that any data that does not appear in the corpus will be a blind spot in the analysis. Assorted techniques exist to deal with this problem. The paper mentions smoothing, which attempts to account for unseen data by adjusting the probabilistic estimates of the observed data. The nuances of smoothing, and other fix-ups to the sparse data problem, go far beyond the scope of this (or any other single) paper.

The distance between the application of statistical techniques for language and for music is not large. Language and music both share multi-level structural regularities. The most well-known method for attacking these sorts of problems is with a Hidden Markov Model. Indeed, one could take a quick stab at automatic music composition using an HMM designed for sentence analysis with very little modification to the algorithm. This is both good and bad: good, because the HMM formalism is relatively simple and well-understood. Bad, because its shortcomings are just as apparent in the music as the linguistic domains.

The paper points out that a shortcoming of the HMM is that it is not "creative" in the sense in which Chomsky uses the word. A Markovian process first analyses a corpus of data and determines the relationships between the atoms of that corpus. This means that when used in a generative fashion, an HMM can only emit structure it has already encountered. One can, of course, get around this using various ad-hoc mechanisms, but doing so abandons the precise probabilistic formalism that is one of the advantages of the model.

The other main critique of HMMs is that they cannot capture structure beyond the order of their Markovian assumption. The paper describes this as a failure of context: a 2-element Markov process analyzes musical corpora with only 1 element of context. One could in theory extend the order of the model to take in more context, but this quickly becomes computationally intractable. In practice (at least in NLP domains) small numbers < 5 dominate. In a synthetic application, within these parameters, local structure is generally good, but global coherence is sacrificed. The author mentions repetition as a structural feature lost to naive HMMs; other shortcomings might be thematic coherence and symmetry.

This problem also plagues the NLP community, so it should be no surprise that a number of efforts have been made to address the shortcoming. Hierarchical HMMs take the basic HMM idea and "zoom out" – one can imagine a "level 0" HMM grouping the atoms of the corpus into basic structures, and the "level 1" HMM grouping the "level 0" groups, and so on. This is a nice idea, and has produced some nice results. An even more promising idea is to make the abstraction richer from the start – instead of analyzing statistical regularities among low-level elements (words), tree-induction techniques apply the HMM idea to larger elements: trees. Syntax trees directly address some of the major structural weaknesses described above, and have produced very promising results; in fact, this technique is the current state of the art in certain NLP tasks. It's very new, however, and the author can be forgiven for not including it.

The paper mentions some other statistical techniques: Gibbs sampling, which is an iterative process: start with a target composition, possibly consisting of just a seed element. Then, loop over this composition, and replace one of its elements with some acceptable set of elements. Determining what is "acceptable" can be expensive, and the process strikes me as both less principled than HMMs and just as susceptible to all its contextual and structural failings.

The author proposes, in summary, that the full catalog of pattern matching / machine learning techniques be mined of potential candidates for use in the music domain. In particular, he says that with "music generation rephrased as a classical search and sampling problem, generation algorithms are free to apply deep knowledge and draw from extant pieces to reduce the search space … [and] can use modern pattern discovery methods … to reveal their repetition structure and provide musical cohesion to new productions." The problem with this call to arms is that it's too general; a full-blown and fully general pattern recognizer of the sort the author wants would, if it existed, be a defining moment for most of AI.

Even when the problem domain is restricted, statistical learning techniques applied to natural language have shown that working with these sorts of recursively structured datasets is almost as much voodoo as science, and "good" solutions can require a great deal of hackery and massaging. The best practical solution, for the foreseeable future, is likely to be a hodgepodge of existing technologies, preferably with a human in the loop to fill in the gaps.