Maximally uniform sequences from stochastic processes

Miller Puckette
University of California, San Diego
msp@ucsd.edu

Abstract

An algorithm is presented that is capable of generating sequences of tokens having specified frequencies of individual elements or subsequences, in a maximally uniform way. As special cases the algorithm can generate Euclidean rhythms and the original ("algae") Lindenmayer system. Secondary sequences constructed from these primary ones show features on a wide range of time scales. A recently completed piece by the composer Kerry Hagan successfully builds these low-level features into a musical idiom.

Keywords

stochastic music

1 Introduction

Composers have used randomness, either real or simulated, for hundreds of years; prototypical aims might be to create open musical forms or to control statistical properties of large collections of musical events. The purpose of this paper is to argue, by example, that probability theory can be used in other, deterministic ways to generate musical structures. The approach taken here will be use a probability space to control statistical behavior in a maximally uniform way, giving results that are much different from the highly non-uniform outputs of simulations or random processes.

The mathematical theory of probability in its modern form was formalized by Kolmogorov, who defined a *probability space* as a triple $(\Omega, \mathcal{F}, P)$ in which Ω, the sample space, is the set of all possible outcomes; $\mathcal{F}$ is a set of allowable subsets of Ω, called *events*, to which we will assign probabilities; and the probability measure P assigns probabilities (numbers between 0 and 1) to sets in $\mathcal{F}$. For example, in the Bernoulli (coin-tossing) space, the sample space consists of all infinite sequences of 0s and 1s. One possible event in $\mathcal{F}$ is "the set of all sequences in which the fifth coin toss comes up heads", and applying P to that set gives its probability, p. P is called a *probability measure* because we can think of the probability of a set as its size or measure.

It is very significant that probability theory eschews any mention of "randomness", which properly belongs to the spooky realm of metaphysics. Philosophers have been arguing about the meaning of randomness for hundreds of years with no resolution in sight; we will not stop even to sketch the difficulties raised by the notion. Nonetheless, there are well understood algorithms that *simulate* randomness, and these are frequently used in musical algorithms of all sorts.

Taking as an example the Bernoulli process, this simulation is done as follows. First we define *random variables* $X_1, X_2, \ldots$, each of which takes the value 0 or 1. (The name "random variable" is a misnomer; there is nothing random about them. A random variable is formally defined as a measurable function whose domain is Ω. For example, the function X_5 has value 1 on all the points of Ω for which the fifth coin toss comes out heads.) These random variables serve us notationally; we can write down the events we're interested in in terms of them. For instance, the probability that the fifth coin comes up heads is $P(\{\omega \in \Omega | X_5(\omega) = 1\})$. In the Bernoulli process in its simplest form, we take the probability of heads to be a number $0 < p < 1$, and assign $P(\{X_n = 1\})$ to equal p for $n = 1, 2, \ldots$.

With the random variables so defined, we simulate randomness by generating a pseudo-random sample point, a sequence of 0s and 1s, as follows. Generate a pseudo-random real number from 0 to 1. If it's below the probability p, set $x_1 = 1$, and otherwise set $x_1 = 0$. Continue this way indefinitely, generating an indefinite sequence $x_1, x_2, \ldots$, which is the desired sample point.

Pseudo-random number generators usually depend on an initial *seed*, and one can generate any desired number of simulations of the Bernoulli (or other) process by supplying different seeds.

2 Maximally uniform Bernoulli sequences

We now use the same probability space, the Bernoulli process, to generate what we will call a *maximally uniform sequence*. The idea is to generate a sequence of 0s and 1s which has the property that, at any point in the sequence, the number of 1s is as nearly as possible exactly p times the total number of items so far.

At each point in the output sequence, we decide which token from the set $\{0, 1\}$ has the greatest *dearth*, i.e., the greatest shortfall in its numbers, Let d_{ij} denote the dearth of the ith token at the jth step. For each step j we compute the two dearth values d_{0j} and d_{1j}:

$$d_{ij} = \sum_{k=1}^{j} P(\{X_k = i\}) - \sum_{k=1}^{j-1} \delta(x_k, i)$$

The first sum counts the mean number of times the token i occurs in the first j steps, and the second counts the actual number of times the token i has occurred before (but not including) the jth step.

At each jth step we then choose x_j to be the state whose computed dearth is the highest. In the case of a tie, we take the lower-numbered state.

Setting $p = 5/12$, for instance, gives a repeating sequence:

```
( 0 1 0 1 0 1 0 0 1 0 1 0 )
```

This is recognizable as the disposition of naturals (0s) and accidentals (1s) in the 12-tone chromatic scale starting at F natural. It is also an example of a Euclidean rhythm[Toussaint and others, 2005]. In general, any Euclidean rhythm can be generated this way.

Formally, the construction may be described (in slightly more general form than before) as follows. Let the random variables X_1, X_2, ..., take on A possible values, numbered between 0 and A-1. Start with a seed vector (an initial dearth vector) $d_{10}, \ldots, d_{A-1,0}$. (The first index is the outcome, and the second one is the step number.) Next, for each step $j >= 0$ in turn, we compute the conditional probability of the outcome i at step j, given all previous outcomes:

$$p_{ij} = P(\{X_k = i | X_0 = x_0, \ldots X_{j-1} = x_{j-1}\})$$

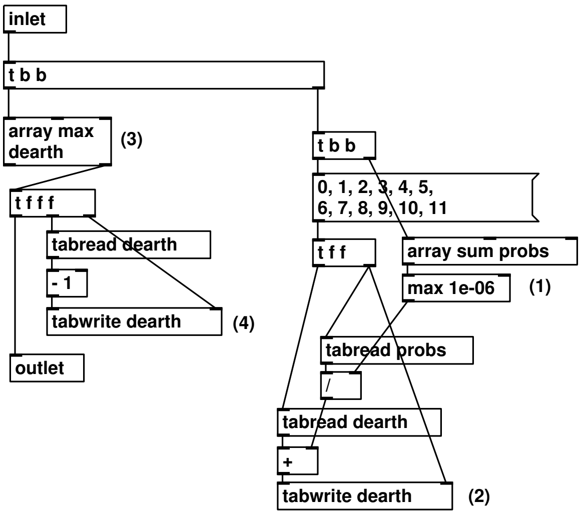


Figure 1: A pure data patch fragment to calculate maximally uniform sequences for independent random variables with probabilities stored in the array *probs*.

(If the random variables are independent and identically distributed, as with the Bernoulli process, we can just take $p_{ij} = p_i$ to be a constant vector with A elements). The outcome for step j is then:

$$x_j = \text{argmax}_i(d_{ij} + p_{ij})$$

In words, we choose the one whose dearth would be highest at step i if it were not chosen. We then update the dearth vector for step $j + 1$:

$$d_{i,j+1} = d_{ij} + p_{ij} - \delta(x_j, i)$$

Here we subtracted one from the dearth for the outcome x_j to mark the fact that we have just output it.

Figure 1 shows an implementation of the algorithm in Pure Data. It requires an array "probs" giving the probabilities of up to 12 possible outcomes, and an array "dearth" in which the dearths are maintained. The patch can be seeded with a starting dearth vector if desired. Sending a bang to the inlet causes the patch to compute each successive outcome. In between calculations the probability table may be changed, allowing the different X_i each to have its own probability distribution if desired and to allow dependencies between the random variables by computing the conditional probabilities given the history of past outcomes.

In the figure, at step (1) we normalize the set of probabilities so that they sum to one; at step (2) we add the normalized probabilities to the dearth array and temporarily store the result in the same array; then (3) we choose the outcome as the argmax of the resulting array, and finally (4) we subtract one from the dearth array at the location corresponding to the chosen outcome.

3 Bernoulli process with irrational probabilities

Non-repeating sequences result if we start with irrational values of p. For example, if $r = (1+\sqrt{2})/2$ is the golden ratio, we set $p = 1/r \approx 0.618$ so that it and $1-p = 1/r^2$ are in the ratio r. The maximally uniform sequence generated is then

```
( 0 1 0 1 0 1 0 0 1 0 1 0 )
```

with the interesting property that all its finite-length subsequences are found in the Lindenmayer algae sequence[Lindenmayer, 1968]:

```
( 0 1 0 0 1 0 1 0 0 1 0 0 1 ... )
```

We can get the original Lindenmayer sequence if we start with a seed dearth vector of

$$(r^{-3}, 0)$$

4 Statistics

Compared to a pseudo-random sequence, a maximally uniform one exhibits entirely different statistical behavior, as exemplified in Figure 2. The maximally uniform process stays within a strip one unit high about a line with slope p. The pseudo-random one diverges gradually away from that line, although slowly enough that the average proportion of 1s tends toward p.

Although the frequencies of the individual outputs of the maximally uniform process hew closely to their probabilities, the frequency of occurrence of successive pairs does not. In particular, if $p < 1/2$ as it is in the above example, there will never be two outcomes of 1 in succession. We can obtain correct frequencies of pairs of outcomes by modeling a first-order Markov process whose outcomes are pairs of successive Bernoulli process outcomes.

5 Distributions of ordered triples

The sequences generated so far do not exhibit behavior on a variety of time scales in a way that might make interesting musical structures. This can be achieved in a simple and surprising way: if a process has more than three output

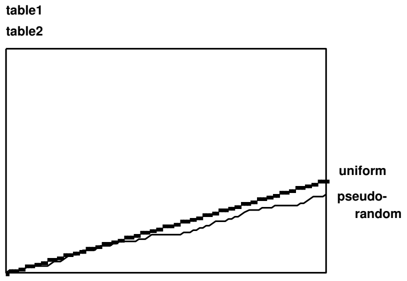


Figure 2: Cumulative sums of outputs from a Bernoulli process with $p = r^2$, comparing a pseudo-random process with a maximally uniform one.

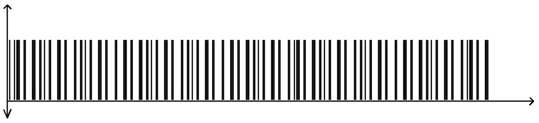


Figure 3: Result of testing for permutation (210) in a golden-section-derived, four-output maximally uniform process.

values, say 0, 1, 2, and 3, then we can ask, at any output stage, in what order the most recent occurrences of 0, 1, and 2 appeared. The six possibilities can be regarded as six outcomes for a secondary, derived process.

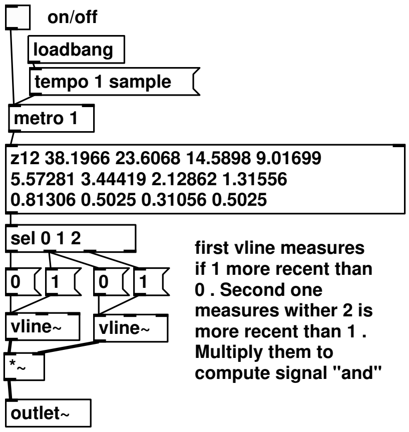
If we did this for a pseudo-random sequence we would merely have a Markov chain. For instance the order (012) would transition, at any step, to (120) or (021) if a 0 or 1 appeared, respectively, and remain the same otherwise.

But applying this same transformation to a maximally uniform process can give much more interesting results. For example, suppose we allow four output states, letting the random variables be independent with probabilities

$$r^{-2}, r^{-3}, r^{-4}, r^{-3}$$

seeding with the vector (0.3, 0, 0, 0), and reporting at each stage a 1 if the first three outputs last appeared in the order (201). The first 300 states are shown in Figure 3.

Although at first glance this seems to be a mere bar code image, on closer examination the golden section appears on many time scales in



first vline measures if 1 more recent than 0 . Second one measures wither 2 is more recent than 1 . Multiply them to compute signal "and"

Figure 4: Result of testing for permutation (210) in a golden-section-derived, four-output maximally uniform process.

the distribution of zeros and ones. This simple example does not make a convincing case for musical utility but at least shows the presence of structure on multiple time scales inherited from the generating probability vector.

The most direct way to use this musically is by granular synthesis, starting identical or slowly changing grains at each audio sample corresponding to an outcome of 1. To do this, we choose a probability distribution (possibly with a dozen or so states), and then a particular permutation desired for the most recent appearance of three of the twelve states, and generate an audio signal that is 1 when the chosen permutation occurs and 0 if a different one does; this gives an audio signal resembling the output shown in figure 3.

Figure 4 shows how this can be done in Pure Data. The metro object outputs a bang for every audio sample (sample-synchronous tempos for metronomes and other such objects require Pd 0.45 or later). The `vline~` objects are able to convert messages to signals with sample accuracy. The “z12” object is an abstraction whose heart is the patch fragment shown in figure 1.

Once this has been computed, we convolve with whatever grain we desire, either using an FIR filter or a filterbank with desired resonant frequencies.

6 Conclusion

As with any such technique, it is up to the composer to make it make music, and up to the listener to judge the result. This technique was the basis for a new, 124-channel piece, *Cubic Zirconia*, presented by Kerry Hagan at the Moss Center for the Arts during this conference. In this piece, fifteen separate processes are run simultaneously, each controlling eight different grain shapes, all with the same generating probabilities (but different seeds) to generate a maximally rich sound field. If this were collapsed to stereo it would be difficult or impossible to hear the structures; the piece relies on the very high channel count to convey an exceedingly complex but highly non-uniform sound field, whose structure at multiple time scales reflects the properties of the chosen probability vector.

So far, the choice of probability vector, seeds, and the selected permutation pattern (i.e., the three particular outcomes and the order in which they must occur to trigger a grain of sound) are all made by a combination of intuition and trial and error. The results can be quite surprising, which is encouraging in a way, but we would like to know more about how the choices affect the heard output.

Nonetheless, even in this early stage of experimentation we can quickly come up with outputs that simultaneously exhibit audible structures both on the micro level (pitches; rhythms) and also over spans of time on the order of 10 minutes. This will be an interesting direction for future exploration.

References

A Lindenmayer. 1968. Mathematical models for cellular interactions in development. *Journal of theoretical biology*, 18(3):280.

Godfried Toussaint et al. 2005. The euclidean algorithm generates traditional musical rhythms. In *Renaissance Banff: Mathematics, Music, Art, Culture*, pages 47–56. Canadian Mathematical Society.