Math 268: Seminar in Combinatorics - Logic and Randomness

1. Motivation and introduction

ML randomness	KL randomness
c.e. strategy (bet amount is approximable from below)	computable strategy
monotonic	nonmonotonic

Theorem (Muchnik, Semenov, Uspensky)**.** *Any ML-random set is KL-random.*

Question. Is every KL-random ML-random. Or, is there a KL-random that is not ML-random?

2. Kastermans, Lempp

"Comparing Notions of Randomness", Theoretical Computer Science, 411:3 (2010), 602-616.

In this paper, Kastermans and Lempp study two versions of nonmonotonic randomness: injective randomness and permutation randomness. The picture is

$$\text{ML-random} \subseteq \text{KL-random} \subseteq \text{injective random} \subseteq \text{permutation random}.$$

They prove:

Theorem. *ML-random $\neq$ permutation random and ML-random $\neq$ injective random*

Of course, this means that to solve the ML-KL question (negatively), it suffices to prove KL-random = injective random.

3. Definitions - ML randomness

Typically, we will define randomness via martingales. But, tests will be useful as well.

Definition. A **martingale** is a function $d : \{0,1\}^* \to \mathbb{R}^{\geq}$ satisfying

$$d(\sigma) = \frac{d(\sigma 0) + d(\sigma 1)}{2}.$$

A martingale **succeeds** on $X \in \{0,1\}^\infty$ if $\limsup_n d(X \upharpoonright n) = \infty$.

Definition. An infinite sequence $X \in \{0,1\}^\infty$ is **ML random** if no c.e. martingale succeeds on it.

Definition. A **Martin-Löf test** (ML test) is a uniformly c.e. sequence of open sets $\{V_k\}$ such that $\forall k \mu(V_k) \leq 2^{-k}$ (where μ is Lebesgue measure). An infinite string $X \in \{0,1\}^\omega$ is disqualified by the ML test if

$$X \in \bigcap_k V_k.$$

Fact. *An infinite string $X \in \{0,1\}^\omega$ is ML random if and only if it passes all ML tests.*

4. Definitions - KL randomness, part 1

To define nonmonotonic betting strategies, one has to encode both the bit values discovered by the strategy and their location in the infinite sequence being played. One way to do so follows MMNRS and uses finite assignments and observation functions.

Definition. A **finite assignment** is a sequence $w = \{(n_i, a_i)\}_{i=0}^{k} \in (\mathbb{N} \times \{0,1\})^*$. We interpret a finite assignment as listing the place and value of bits scanned so far. That is, for a given string X, a finite assignment would say that

$$X(n_i) = a_i$$

for each $i = 0, \ldots, k$. We denote by FA the set of all finite assignments; that is, $FA = (\mathbb{N} \times \{0,1\})^*$.

A **scan rule** is a partial function $s : FA \to \mathbb{N}$ such that

$$\forall w \in FA \, (s(w) \notin \mathrm{dom}(w)) \, .$$

In other words, any partial function that does not repeatedly look at any bits in the string.

A **nonmonotonic stake function** is a partial function $q : FA \to [0,2]$.

A **nonmonotonic betting strategy** is a pair $b = (s, q)$ where s is a scan rule and q is a nonmonotonic stake function. Alternatively, a nonmonotonic betting strategy is $b : \{0,1\}^* \to \mathbb{N} \times [0,2]$ and we interpret

$$b(\sigma) = (\text{next bit to scan}, \text{amount to bet})$$

where σ is the sequence of bit-values in the locations we chose to scan (and the "next bit to scan" always satisfies the "no repeated scan" rule). Note that given such a function, the associated scan and stake rules can be recovered inductively.

The **capital function**, $d_b : FA \to [0,2]$ is defined inductively by $d_b(\lambda) = 1$ and for $w \in FA$ and $i \in \{0,1\}$

$$d_b(w \,\hat{}\, (s(w), i)) = \begin{cases} d_b(w)q(w) & \text{if } i = 0 \\ d_b(w)(2 - q(w)) & \text{if } i = 1 \end{cases}$$

Definition. Fix a nonmonotonic betting strategy $b = (s, q)$ and an infinite string X.

The **observation function** is a partial function $o_b^X : \mathbb{N} \to FA$ defined by

$$o_b^X(0) = \lambda \qquad \text{and} \qquad o_b^X(k+1) = o_b^X(k) \,\hat{}\, \big(s(o_b^X(k)), X(s(o_b^X(k)))\big)$$

if $o_b^X(k), s(o_b^X(k))$ are both defined (and it is undefined otherwise).

For any $X \in \{0,1\}^\omega$ and for all $k \in \mathbb{N}$

$$d_b^X(k) = d_b(o_b^X(k)).$$

The strategy b **succeeds on** X if $\limsup_k d_B^X(k) = \infty$.

Fact. *A function $d : FA \to \mathbb{R}^{\geq}$ is a* **nonmonotonic martingale** *if for any $w \in FA$, there is an $n \in \mathbb{N}$*

$$d(w) = \frac{d(w \,\hat{}\, (n,0)) + d(w \,\hat{}\, (n,1))}{2}$$

The capital functions of nonmonotonic betting strategies are nonmonotonic martingales.

Warning: Contents not proofread.

Proof. For $b = (s, q)$ let $w \in FA$ and put $n = s(w)$. Then

$$\frac{d_b(w\,\hat{}\,(n,0)) + d_b(w\,\hat{}\,(n,1))}{2} = \frac{d_b(w)c_b(w\,\hat{}\,(n,0)) + d_b(w)c_b(w\,\hat{}\,(n,1))}{2}$$
$$= \frac{d_b(w)q(w)) + d_b(w)(2 - q(w))}{2} = d_b(w).$$

If d_b is partial, it satisfies the definition of partial martingales. □

Definition. For $b = (q, s)$ a nonmonotonic betting strategy and $w \in FA$, we define what it means for a finite assignment $v \in FA$ extending w to be **consistent with** b by induction on $|v| - |w|$:

- if $|v| - |w| = 1$ then $v = w\,\hat{}\,(n, i)$ for some $n \in \mathbb{N}, i \in \{0, 1\}$. In this case: v is consistent with b if and only if $n = s(w)$.
- if $|v| - |w| = k + 1$ and $v = w\,\hat{}\,u\,\hat{}\,(n, i)$ is such that $w\,\hat{}\,u$ is consistent with b then v is consistent with b if and only if $n = s(w\,\hat{}\,u)$.

Lemma (Kolmogorov's Inequality extended to nonmonotonic strategies)**.** *If $b = (q, s)$ is a nonmonotonic betting strategy, $w \in FA$, S a prefix-free set of extensions of w consistent with b then*

$$\sum_{v \in S} 2^{-|v|} d_b(v) \leq 2^{-|w|} d_b(w).$$

Fact (Merkle 2003; DH p. 310; MMNRS p. 7)**.** *A sequence is KL random if and only if no partial computable betting strategy succeeds on it.*

Proof. Given a partial computable nonmonotonic betting strategy that succeeds on some string X, build a total computable nonmonotonic betting stratgey that also succeeds on X. Let $s_0, s_1, \ldots$ be the sequence of locations of X scanned by the partial computable strategy d such that

$$\limsup_k d(X(s_k)) = \infty.$$

Partition $\{s_i\}$ into odd and even locations; it must be the case that $\limsup = \infty$ for at least one of the corresponding subsequences. Define a total computable nonmonotonic betting strategy that emulates d on the subsequence with $\limsup = \infty$ and uses the other subsequence to "wait" for convergence of each of the computations. □

5. Definitions - KL randomness, part 2

An alternative approach is suggested in Miller-Nies and Kastermans-Lempp: nonmonotonic strategies playing against some infinite sequence can be thought of as monotonic strategies playing against a different sequence.

Definition (Miller-Nies p. 10)**.** An assignment is a sequence $\sigma \in (\mathbb{N} \times \{0, 1\})^{\leq \omega}$. The bit sequence associated with an assignment σ is defined as $\alpha(\sigma) = \pi_2(\sigma) \in \{0, 1\}^{\leq \omega}$. A **scan rule** is a partial computable function $s : (\mathbb{N} \times \{0, 1\})^* \to \mathbb{N}$ such that $s(w) \notin \text{dom}\,(w)$. The assignment given by s for an infinite string $Z \in \{0, 1\}^\infty$ is denoted σ_s^Z.

Definition. An infinite sequence $X \in \{0, 1\}^\infty$ is KL-random if for each scan rule s, if the associated assignment σ_s^X is infinite then the corresponding bit sequence, $\alpha(\sigma_s^X) \in \{0, 1\}^\infty$ is partial computably random.

6. Definitions - Weak nonmonotonic randomness

A potentially weaker notion of nonmonotonic randomness occurs if we force the scan function to only be a function of the number of bets made so far, but not of which bits were bet on or what the bit values are.

Definition. A scan rule is **injective** if there is some injective function $h : \omega \to \omega$ such that for any $w \in (\omega \times 2)^*$, $s(w) = h(|w|)$.

Note: On Downey-Hirschfeldt p. 319, this is called an **oblivious** scan rule. That is, $s : \mathbb{N}^* \to \mathbb{N}$ and $s(w) \notin w$.

Definition (Miller, Nies)**.** A real X is **permutation random** if no partial computable betting strategy (q, s) where $s(w) = h(|w|)$ and h is a partial computable permutation of $\mathbb{N}$ succeeds on X. That is, if no strategy which checks all bets of X succeeds on X.

Equivalently, X is permutation random if $X \circ h$ is a partial computable random for each computable permutation h.

Definition (Miller, Nies)**.** A real X is **injective random** if no partial computable betting strategy (q, s) where $s(w) = h(|w|)$ and h is a partial computable injection succeeds on X.

Equivalently, X is injective random if $X \circ h$ is a partial computable random for each computable injection h.

Fact. *The notions of injective randomness and permutation randomness do not change if we require h to be total.*

Proof. In order to be successful, the strategy $b = (q, s, h)$ must have $\limsup_n d_b^X(n) = \infty$. Since $d_b^X(n) = f(q, s) = f(q, h)$, it must be defined for all n in order to be unbounded. This does not constrain q to be total: there are some FAs that have nothing to do with X. But, since h depends only on length (number of bets), in order for h to be defined for all the bets against X, it must be defined for all bets against any infinite sequence. In particular, this means that the scan functions that may witness non-randomness are total. □

Fact (MMNRS Remark 9, Buhrman et al Theorem 3.3)**.** *Requiring the stake function to be total does make a difference: total permutation random is equivalent to computable random.*

This will be a key ingredient of the proof of the separation of ML-randomness and permutation randomness.

Question. What is total injective random equivalent to?

Fact. *KL-random $\subseteq$ injective random $\subseteq$ permutation random.*

Proof. The proofs are straightforward applications of the definitions. Suppose $b = (q, s, h)$ with h total computable injective succeeds on X. The fact that h is injective guarantees that for each $w \in FA$, $s(w) \notin \text{dom}\ (w)$. Therefore, $b = (q, s)$ is a nonmonotonic betting strategy that succeeds on X and X is not KL-random.

Similarly, suppose there is $b = (q, s, h)$ which succeeds on X where h is a partial computable permutation of ω, then this b also witnesses that X is not injective random. □

7. Total permutation random

Lemma (11 in Kastermans Lempp)**.** *For a total nonmonotonic betting strategy $b = (s, q)$ with $s = h(n)$ and h is a permutation on $\mathbb{N}$, there is a computable monotonic martingale which succeeds on every infinite sequence on which b succeeds.*

Proof. There are two main conceptual pieces to the proof:

(A) For each n, there is a finite time at which (no matter what string is being played against), all bits that might be necessary in deciding how to bet on the n th one are available. The monotonic betting strategy will take an average over all possible values of these bits.
(B) The martingale achieved by taking this average succeeds on any reals that the original one did via a "slowly-but-surely" winning version.

To address part (B), we use the following lemma.

Lemma (9 in Kastermans Lempp)**.** *If $b = (s, q)$ is a partial computable nonmonotonic betting strategy then there is a partial computable nonmonotonic betting strategy $\bar{b} = (\bar{s}, \bar{q})$ that succeeds on the same infinite sequences as b and which satisfies, for all $\sigma, \tau \in \{0,1\}^*$,*

$$d_{\bar{b}}(\sigma\tau) > d_{\bar{b}}(\sigma) - 2, \qquad \text{and} \qquad d_{\bar{b}}(\sigma) < 2(|\sigma| + 1). \tag{1}$$

That is, the strategy never loses too much of its money, and also earns money slowly.

This is one version of the savings trick: whenever the strategy has capital greater than 2, move 2 into savings and only bet with remaining capital.

Using Lemma 9, we therefore assume that the capital of nonmonotonic betting strategies satisfies (1). Define the function $\hat{d}_b(\sigma) : \{0,1\}^* \to \mathbb{R}^{\geq}$ (which we will show is a monotonic martingale) as

$$\hat{d}_b(\sigma) = \sum_{\tau \in \{0,1\}^{l_\sigma}, \sigma \sqsubseteq \tau} d_b^\tau(n_\sigma) 2^{-(l_\sigma - |\sigma|)},$$

where $\sigma \in \{0,1\}^*$, $n_\sigma, l_\sigma \in \mathbb{N}$ such that for each $0 \leq j < |\sigma|$, there is $0 \leq k < n_\sigma$ such that $h(k) = j$; for each $0 \leq i < n_\sigma$, $h(i) \leq l_\sigma$; and $l_\sigma > |\sigma|$. The first condition guarantees that at most n_σ many bets are needed to get all the bits of σ. The second condition guarantees that in the course of these n_σ many bets, we look at bits of the string up to length l_σ at most.

Claim: the value of $\hat{d}_b(\sigma)$ is invariant under different choices of n_σ, l_σ, so long as they satisfy the above-mentioned requirements.

Recall that $d_b^\tau(n)$ is the capital of the nonmonotonic strategy b playing against (an extension of) τ for n steps. In order for it to be defined, τ must be long enough to include all bits scanned in the course of this play. Let $n \geq n_\sigma, l \geq l_\sigma$. Then, for each $\tau \in \{0,1\}^l$, $d_b^\tau(n)$ is defined because condition (2) guarantees that the first n bets of the strategy scan at most

the bits up to position l, and τ has length l so can settle all these bets.

$$\begin{aligned}\sum_{\tau\in\{0,1\}^l,\sigma\sqsubseteq\tau} d_b^\tau(n)2^{-(l-|\sigma|)} &= \cdots\langle\text{use martingale property for each bet of } b \text{ beyond } n_\sigma\text{th one}\rangle\cdots \\ &= \sum_{\tau\in\{0,1\}^l,\sigma\sqsubseteq\tau} d_b^\tau(n_\sigma)2^{-(l-|\sigma|)} = \sum_{\tau\in\{0,1\}^{l_\sigma},\sigma\sqsubseteq\tau}\ \sum_{\alpha\in\{0,1\}^{l-l_\sigma}} d_b^{\tau\alpha}(n_\sigma)2^{-(l-|\sigma|)} \\ &= \sum_{\tau\in\{0,1\}^{l_\sigma},\sigma\sqsubseteq\tau} d_b^\tau(n_\sigma)\sum_{\alpha\in\{0,1\}^{l-l_\sigma}} 2^{-(l-|\sigma|)} = \sum_{\tau\in\{0,1\}^{l_\sigma},\sigma\sqsubseteq\tau} d_b^\tau(n_\sigma)2^{l-l_\sigma}2^{-(l-|\sigma|)} \\ &= \hat{d}_b(\sigma).\end{aligned}$$

Claim: $\hat{d}_b$ satisfies the martingale property.

Let $n = \max\{n_\sigma, n_{\sigma 0}, n_{\sigma 1}\}$, $l = \max\{l_\sigma, l_{\sigma 0}, l_{\sigma 1}\}$. Then,

$$\begin{aligned}\hat{d}_b(\sigma 0) + \hat{d}_b(\sigma 1) &= \sum_{\tau\in\{0,1\}^l,\sigma 0\sqsubseteq\tau} d_b^\tau(n)2^{-(l-|\sigma|-1)} + \sum_{\tau\in\{0,1\}^l,\sigma 1\sqsubseteq\tau} d_b^\tau(n)2^{-(l-|\sigma|-1)} \\ &= \sum_{\tau\in\{0,1\}^l,\sigma\sqsubseteq\tau} d_b^\tau(n)2^{-(l-|\sigma|-1)} = 2\sum_{\tau\in\{0,1\}^l,\sigma\sqsubseteq\tau} d_b^\tau(n)2^{-l-|\sigma|} = 2\hat{d}_b(\sigma).\end{aligned}$$

Claim: $\hat{d}_b$ succeeds on every infinite sequence that b succeeds on.

Suppose b succeeds on some $X \in \{0,1\}^\infty$. To show that $\limsup_k \hat{d}_b^X(k) = \infty$, let C be arbitrary and we will show there is some k such that $\hat{d}_b(X \upharpoonright k) \geq C$. By assumption that b succeeds on X, there is k such that $d_b^X(k) \geq C + 2$. Then, for n, l sufficiently large,

$$\hat{d}_b(X \upharpoonright k) = \sum_{\tau\in\{0,1\}^l, X\upharpoonright k\sqsubseteq\tau} d_b^\tau(n)2^{-l+k} \geq \sum_{\tau\in\{0,1\}^l, X\upharpoonright k\sqsubseteq\tau} (C+2-2)2^{-l+k} = C\sum_{\tau\in\{0,1\}^l, X\upharpoonright k\sqsubseteq\tau} 2^{-l+k} = C,$$

where the inequality follows by property (1) of "slowly-but-surely" winning martingales.

Claim: $\hat{d}_b$ is computable.

Since the sum defining $\hat{d}_b$ is computable and its terms are given by the total computable function $d_b^\tau(n)$, it's sufficient to argue that the maps $\sigma \mapsto n_\sigma$ and $\sigma \mapsto l_\sigma$ are computable. Since h is a computable permutation, we can enumerate its range $\{h(0), h(1), \ldots\}$. At some finite point, this list will contain the set $\{0, 1, \ldots, |\sigma| - 1\}$. This finite point is $n_\sigma - 1$. That is, for each $0 \leq i < |\sigma|$, there is some $0 \leq j < n_\sigma$ such that $h(j) = i$. We can also extract l_σ from this list: $l_\sigma = \max\{h(j) : 0 \leq j < n_\sigma\} + 2$. (The extra 2 guarantees that $l_\sigma > |\sigma|$ since $h(i) = |\sigma| - 1$ for some i so $h_i + 2 = |\sigma| + 1$.)

□

8. Avoiding monotonic betting strategies

Lemma. *For any two martingales d, d', and any nonnegative real r, the weighted sum $d + rd'$ is a martingale.*

Fact (Section 2.2 in Kastermans, Lempp)**.** *If $\{d_i\}$ is a countable enumeration of monotonic martingales all with initial capital less than or equal to 1, there is an infinite sequence $A \in \{0,1\}^\infty$ on which none of the d_is succeeds.*

Proof. Define A inductively. Let $A(0)$ be such that $d_0(A(0)) < 2$. (If the initial capital of d_0 is strictly less than 1 then $A(0)$ can be picked to be 0 by default; if the initial capital of d_0 is exactly 1, then pick the bit that makes d_0 lose its first bet.) Suppose we have defined $(A \upharpoonright (n+1)) = \sigma$ and auxiliary positive real numbers $s_1, \ldots, s_n$ such that

$$\hat{d}_n(\sigma) = d_0(\sigma) + s_1 d_1(\sigma) + \cdots + s_n d_n(\sigma) < 2.$$

Put

$$s_{n+1} = \frac{2 - \hat{d}_n(\sigma)}{2 d_{n+1}(\sigma)},$$

which is positive by the induction hypothesis. Then consider the martingale $(\hat{d}_n + s_{n+1} d_{n+1})$ playing against $(A \upharpoonright (n+1)) = \sigma$:

$$(\hat{d}_n + s_{n+1} d_{n+1})(\sigma) = \hat{d}_n(\sigma) + \frac{2 - \hat{d}_n(\sigma)}{2 d_{n+1}(\sigma)} d_{n+1}(\sigma) = 1 + \frac{1}{2}\hat{d}_n(\sigma).$$

The induction hypothesis then gives that $(\hat{d}_n + s_{n+1} d_{n+1})(\sigma) < 2$. We choose the next bit of A so that this martingale loses the bet on it and hence its capital remains below 2:

$$(\hat{d}_n + s_{n+1} d_{n+1})(\sigma A(n+1)) < 2.$$

Claim that for each i, for all $n \geq i$, $d_i(A \upharpoonright n) < \frac{2}{s_i}$. This claim suffices because then

$$\limsup_n d_i(A \upharpoonright n) \leq \frac{2}{s_i}$$

for each i and hence no martingale in the list succeeds on A.

Proof of claim: by the construction, for each n,

$$d_0(A \upharpoonright (n+1)) + s_1 d_1(A \upharpoonright (n+1)) + \cdots + s_n d_n(A \upharpoonright (n+1)) < 2.$$

Since each of these terms is nonnegative, whenever $i \leq n$ we have

$$s_i d_i(A \upharpoonright (n+1)) < 2, \qquad \text{thus,} \qquad d_i(A \upharpoonright (n+1)) < \frac{2}{s_i}.$$

□

9. ML random $\neq$ permutation random

The idea is to diagonalize against all partial computable nonmonotonic betting strategies whose scan functions are given by permutations on $\mathbb{N}$. More specifically, the proof constructs an infinite sequence $A \in \{0,1\}^\infty$ such that

(1) A is not ML random: there is a monotonic betting strategy whose stake function is computably approximable below and which succeeds on A; and
(2) A is permutation random: no partial computable nonmonotonic betting strategy with scan functions given by a permutation on $\mathbb{N}$ succeeds on A.