

(pseudo)Random Remarks on Randomness

Stephen Glasby

Central Washington University

<http://www.cwu.edu/~glasbys/>

July 2009

- 1 Compact, locally compact, discrete examples
- 2 Complexity analysis: tools
- 3 Constructing random elts: algorithms/Markov chains

1. Compact, locally compact, discrete examples

Dichotomy: Structure \leftrightarrow Randomness

- Prime numbers (Tao and Green)
 - Permutation groups (Erdős and Turán)
 - Linear groups (Diaconis, Kung, Stong, Fullman; Neumann, Praeger, Britnell)
- Varied techniques used: Fourier analysis, Cycle index, Matrix Factorizations, Markov chains, Entropy, Random walks, etc.

Theorem [Gronchov 1944] Let $X_n =$ r.v. number of cycles of a permutation of symmetric group S_n . Then $\lim_{n \rightarrow \infty} \text{Prob}(X_n = x) = N(\mu, \sigma^2)$ where $\mu = \sigma^2 = \log_e n$

1. Compact, locally compact, discrete examples

Dichotomy: Structure \leftrightarrow Randomness

- Prime numbers (Tao and Green)
- Permutation groups (Erdős and Turán)
- Linear groups (Diaconis, Kung, Stong, Fullman; Neumann, Praeger, Britnell)
- Varied techniques used: Fourier analysis, Cycle index, Matrix Factorizations, Markov chains, Entropy, Random walks, etc.

Theorem [Gronchov 1944] Let $X_n =$ r.v. number of cycles of a permutation of symmetric group S_n . Then $\lim_{n \rightarrow \infty} \text{Prob}(X_n = x) = N(\mu, \sigma^2)$ where $\mu = \sigma^2 = \log_e n$

1. Compact, locally compact, discrete examples

S = sample space = group, ring, field, algebra

Compact: Uniform distribution $U(A) = \frac{\mu(A)}{\mu(S)}$ e.g. $SU(n, \mathbb{C})$.

○ Common example S finite: $U(A) = \frac{|A|}{|S|}$.

Locally Compact: $S = \mathbb{R}$

○ Normal distribution $N(\mu, \sigma^2)$ where $(\mu, \sigma^2) = (0, 1)$.

○ Toss coin 20 times to find random real (complex) number

Discrete: $S = \mathbb{Z}$

○ uniform compact support

○ non-uniform and non-compact support

1. Compact, locally compact, discrete examples

S = sample space = group, ring, field, algebra

Compact: Uniform distribution $U(A) = \frac{\mu(A)}{\mu(S)}$ e.g. $SU(n, \mathbb{C})$.

○ Common example S finite: $U(A) = \frac{|A|}{|S|}$.

Locally Compact: $S = \mathbb{R}$

○ Normal distribution $N(\mu, \sigma^2)$ where $(\mu, \sigma^2) = (0, 1)$.

○ Toss coin 20 times to find random real (complex) number

Discrete: $S = \mathbb{Z}$

○ uniform compact support

○ non-uniform and non-compact support

1. Compact, locally compact, discrete examples

S = sample space = group, ring, field, algebra

Compact: Uniform distribution $U(A) = \frac{\mu(A)}{\mu(S)}$ e.g. $SU(n, \mathbb{C})$.

○ Common example S finite: $U(A) = \frac{|A|}{|S|}$.

Locally Compact: $S = \mathbb{R}$

○ Normal distribution $N(\mu, \sigma^2)$ where $(\mu, \sigma^2) = (0, 1)$.

○ Toss coin 20 times to find random real (complex) number

Discrete: $S = \mathbb{Z}$

○ uniform compact support

○ non-uniform and non-compact support

2. Complexity analysis: tools

Analyzing algorithms over infinite fields requires special tools.

Two possible tools:

- Roots of polynomials
- Geometry of hypercubes

Observation. If $S \subseteq F$, $2 \leq |S| < \infty$, and $(c_3, \dots, c_n) \in S^{n-2}$, then $\gcd(f_1, f_2 + c_3 f_3 + \dots + c_n f_n)$ equals $\gcd(f_1, f_2, \dots, f_n)$ with “high probability”

Advantages/Disadvantages

- ⊕ Need *one* gcd computation, not $n - 1$.
- ⊖ Need $|S| \gg \max\{\deg(f_1), \dots, \deg(f_n)\}$ for probability to be high.
- ⊕ Algorithm is *almost* field independent. Fails if $|F|$ is small.

2. Complexity analysis: tools

Analyzing algorithms over infinite fields requires special tools.

Two possible tools:

- Roots of polynomials
- Geometry of hypercubes

Observation. If $S \subseteq F$, $2 \leq |S| < \infty$, and $(c_3, \dots, c_n) \in S^{n-2}$, then $\gcd(f_1, f_2 + c_3 f_3 + \dots + c_n f_n)$ equals $\gcd(f_1, f_2, \dots, f_n)$ with “high probability”

Advantages/Disadvantages

- ⊕ Need *one* gcd computation, not $n - 1$.
- ⊖ Need $|S| \gg \max\{\deg(f_1), \dots, \deg(f_n)\}$ for probability to be high.
- ⊕ Algorithm is *almost* field independent. Fails if $|F|$ is small.

2. Complexity analysis: tools

Theorem [DeMillo & Lipton 1978, Zippel 1979, Schwartz 1980]

Let $0 \neq f(x_1, \dots, x_n) \in R[x_1, \dots, x_n]$, where R is an integral domain. Suppose $S \subseteq R$ and $2 \leq |S| < \infty$. Then

$$\text{Prob}(x \in S^n \text{ satisfies } f(x_1, \dots, x_n) = 0) \leq \frac{\deg(f)}{|S|}$$

- ⊖ Unhelpful if $|S| \leq \deg(f)$ (as probabilities ≤ 1)
- ⊕ Treats “large” fields F (and subrings R) uniformly
- ⊖ What if F is a small finite field?

2. Complexity analysis: tools

Theorem [DeMillo & Lipton 1978, Zippel 1979, Schwartz 1980]

Let $0 \neq f(x_1, \dots, x_n) \in R[x_1, \dots, x_n]$, where R is an integral domain. Suppose $S \subseteq R$ and $2 \leq |S| < \infty$. Then

$$\text{Prob}(x \in S^n \text{ satisfies } f(x_1, \dots, x_n) = 0) \leq \frac{\deg(f)}{|S|}$$

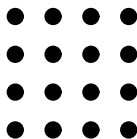
- ⊖ Unhelpful if $|S| \leq \deg(f)$ (as probabilities ≤ 1)
- ⊕ Treats “large” fields F (and subrings R) uniformly
- ⊖ What if F is a small finite field?

2. Complexity analysis: tools

Theorem [SG] Let $S \subseteq F$ satisfy $2 \leq |S| < \infty$. Let $v + U$ be a fixed coset of a proper subspace $U \leq V := F^n$. Then

$$\text{Prob}(x \in S^n \text{ lies in } v + U) \leq |S|^{-\text{codim}(U)} \leq |S|^{-1}.$$

“Proof”



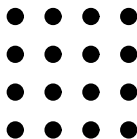
- ⊕ Works for *all* fields (even infinite and small finite)
- ⊕ Does not need polynomial f (certainly do not need $f \neq 0$)
- ⊕ Has many applications (e.g. $\text{Frob}(X)$)
- ⊕ Complexity analysis is easy.

2. Complexity analysis: tools

Theorem [SG] Let $S \subseteq F$ satisfy $2 \leq |S| < \infty$. Let $v + U$ be a fixed coset of a proper subspace $U \leq V := F^n$. Then

$$\text{Prob}(x \in S^n \text{ lies in } v + U) \leq |S|^{-\text{codim}(U)} \leq |S|^{-1}.$$

“Proof”



- ⊕ Works for *all* fields (even infinite and small finite)
- ⊕ Does not need polynomial f (certainly do not need $f \neq 0$)
- ⊕ Has many applications (e.g. $\text{Frob}(X)$)
- ⊕ Complexity analysis is easy.

3. Constructing random elts: algorithms/Markov chains

Three basic methods for constructing random elts:

- (a) Algorithms (e.g. e^X , matrix fact'ns, simple algs)
- (b) Markov chains (e.g. PRA, walks Cayley graphs, ...)
- (c) Physical phenomena (e.g. Radioactivity, Brownian motion, turbulence, etc)

(a1) Exponential map.

G Lie group of $n \times n$ matrices (over \mathbb{R} , \mathbb{C} or \mathbb{H}). Finding $\text{Random}(\text{Lie}(G))$ is easy. Use $\exp: \text{Lie}(G) \rightarrow G$ to construct $\text{Random}(G)$ where $\exp(X) = \sum_{k \geq 0} \frac{X^k}{k!}$. (Converges for $X \in \mathbb{C}^{n \times n}$.)

- \oplus \exp is surjective if G is compact (e.g. $U(n, \mathbb{H})$, $SU(n, \mathbb{C})$, $SO(n, \mathbb{R})$) *c.f.* F. Mezzadri. *Notices AMS* **54** (2009), 592–604.
- \ominus $\exp(\text{Lie}(G))$ need not be dense in G e.g. $G = \text{SL}(2, \mathbb{R})$. Char. poly. of a random elt of G should be a random poly.

3. Constructing random elts: algorithms/Markov chains

Three basic methods for constructing random elts:

- (a) Algorithms (e.g. e^X , matrix fact'ns, simple algs)
- (b) Markov chains (e.g. PRA, walks Cayley graphs, ...)
- (c) Physical phenomena (e.g. Radioactivity, Brownian motion, turbulence, etc)

(a1) Exponential map.

G Lie group of $n \times n$ matrices (over \mathbb{R} , \mathbb{C} or \mathbb{H}). Finding $\text{Random}(\text{Lie}(G))$ is easy. Use $\exp: \text{Lie}(G) \rightarrow G$ to construct $\text{Random}(G)$ where $\exp(X) = \sum_{k \geq 0} \frac{X^k}{k!}$. (Converges for $X \in \mathbb{C}^{n \times n}$.)

- \oplus \exp is surjective if G is compact (e.g. $U(n, \mathbb{H})$, $SU(n, \mathbb{C})$, $SO(n, \mathbb{R})$) *c.f.* F. Mezzadri. *Notices AMS* **54** (2009), 592–604.
- \ominus $\exp(\text{Lie}(G))$ need not be dense in G e.g. $G = \text{SL}(2, \mathbb{R})$. Char. poly. of a random elt of G should be a random poly.

3. Constructing random elts: algorithms/Markov chains

Three basic methods for constructing random elts:

- (a) Algorithms (e.g. e^X , matrix fact'ns, simple algs)
- (b) Markov chains (e.g. PRA, walks Cayley graphs, ...)
- (c) Physical phenomena (e.g. Radioactivity, Brownian motion, turbulence, etc)

(a1) Exponential map.

G Lie group of $n \times n$ matrices (over \mathbb{R} , \mathbb{C} or \mathbb{H}). Finding $\text{Random}(\text{Lie}(G))$ is easy. Use $\exp: \text{Lie}(G) \rightarrow G$ to construct $\text{Random}(G)$ where $\exp(X) = \sum_{k \geq 0} \frac{X^k}{k!}$. (Converges for $X \in \mathbb{C}^{n \times n}$.)

- \oplus \exp is surjective if G is compact (e.g. $U(n, \mathbb{H})$, $SU(n, \mathbb{C})$, $SO(n, \mathbb{R})$) *c.f.* F. Mezzadri. *Notices AMS* **54** (2009), 592–604.
- \ominus $\exp(\text{Lie}(G))$ need not be dense in G e.g. $G = \text{SL}(2, \mathbb{R})$. Char. poly. of a random elt of G should be a random poly.

3. Constructing random elts: algorithms/Markov chains

(a2) Matrix Factorizations.

- $A := \text{Random}(\mathbb{R}^{n \times n})$. Then $A^T A = U D U^T$ where $U \in O(n, \mathbb{R})$, $D = \text{diag}(\lambda_1, \dots, \lambda_n)$ is diagonal and $\lambda_i \geq 0$.

$$\mathbb{R}^{n \times n} \rightarrow O(n, \mathbb{R}): A \mapsto U, \quad \text{then } U = \text{Random}(O(n, \mathbb{R}))$$

- SVD $A := \text{Random}(\mathbb{R}^{n \times n})$. Then $A = U \Sigma V^T$ where $U, V \in O(n, \mathbb{R})$, $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$ is diagonal and $\sigma_i \geq 0$.

$$\mathbb{R}^{n \times n} \rightarrow O(n, \mathbb{R}) \times O(n, \mathbb{R}): A \mapsto (U, V) = \text{Random}(O(n, \mathbb{R})^2)$$

(a3) Simple Algorithms.

$\text{Random}(\text{GL}(n, F)) \oplus$ works for *all* fields

1. $X := \text{Random}(F^{n \times n})$
2. if $\det(X) = 0$ then go to 1.
3. return X

3. Constructing random elts: algorithms/Markov chains

(a2) Matrix Factorizations.

- $A := \text{Random}(\mathbb{R}^{n \times n})$. Then $A^T A = U D U^T$ where $U \in O(n, \mathbb{R})$, $D = \text{diag}(\lambda_1, \dots, \lambda_n)$ is diagonal and $\lambda_i \geq 0$.

$$\mathbb{R}^{n \times n} \rightarrow O(n, \mathbb{R}): A \mapsto U, \quad \text{then } U = \text{Random}(O(n, \mathbb{R}))$$

- SVD $A := \text{Random}(\mathbb{R}^{n \times n})$. Then $A = U \Sigma V^T$ where $U, V \in O(n, \mathbb{R})$, $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$ is diagonal and $\sigma_i \geq 0$.

$$\mathbb{R}^{n \times n} \rightarrow O(n, \mathbb{R}) \times O(n, \mathbb{R}): A \mapsto (U, V) = \text{Random}(O(n, \mathbb{R})^2)$$

(a3) Simple Algorithms.

$\text{Random}(\text{GL}(n, F)) \oplus$ works for *all* fields

1. $X := \text{Random}(F^{n \times n})$
2. if $\det(X) = 0$ then go to 1.
3. return X

3. Constructing random elts: algorithms/Markov chains

Random($SL(n, F)$) \ominus biases n th row when $|F| = \infty$

1. $X := \text{Random}(F^{n \times n})$
2. if $\det(X) = 0$ then go to 1.
3. return $\text{diag}(1, \dots, 1, \det(X)^{-1})X$

Random($Sp(4, F)$) \oplus works for *all* fields

1. $X := \text{Random}(F^{4 \times 4})$
 2. if $\det(X) = 0$ then go to 1.
 3. $A :=$ “canonical coset representative” $XSp(4, F) = ASp(4, F)$.
 4. return $A^{-1}X$
- \oplus O.K. if $|GL(4, F) : Sp(4, F)|$ is “large”

(b) Markov chains and Uncertainty Principle.

Quantum Mechanics trade off position and momentum

Randomness trade off speed alg and uniformness

trade off many small steps vs few large steps

3. Constructing random elts: algorithms/Markov chains

Random($SL(n, F)$) \ominus biases n th row when $|F| = \infty$

1. $X := \text{Random}(F^{n \times n})$
2. if $\det(X) = 0$ then go to 1.
3. return $\text{diag}(1, \dots, 1, \det(X)^{-1})X$

Random($Sp(4, F)$) \oplus works for *all* fields

1. $X := \text{Random}(F^{4 \times 4})$
 2. if $\det(X) = 0$ then go to 1.
 3. $A :=$ “canonical coset representative” $XSp(4, F) = ASp(4, F)$.
 4. return $A^{-1}X$
- \oplus O.K. if $|GL(4, F) : Sp(4, F)|$ is “large”

(b) Markov chains and Uncertainty Principle.

Quantum Mechanics trade off position and momentum

Randomness trade off speed alg and uniformness

trade off many small steps vs few large steps

3. Constructing random elts: algorithms/Markov chains

Random($SL(n, F)$) \ominus biases n th row when $|F| = \infty$

1. $X := \text{Random}(F^{n \times n})$
2. if $\det(X) = 0$ then go to 1.
3. return $\text{diag}(1, \dots, 1, \det(X)^{-1})X$

Random($Sp(4, F)$) \oplus works for *all* fields

1. $X := \text{Random}(F^{4 \times 4})$
 2. if $\det(X) = 0$ then go to 1.
 3. $A :=$ “canonical coset representative” $XSp(4, F) = ASp(4, F)$.
 4. return $A^{-1}X$
- \oplus O.K. if $|GL(4, F) : Sp(4, F)|$ is “large”

(b) Markov chains and Uncertainty Principle.

Quantum Mechanics trade off position and momentum

Randomness trade off speed alg and uniformness

trade off many small steps vs few large steps

3. Constructing random elts: algorithms/Markov chains

Random(\mathbb{Z}) generators $\{-1, 1\}$. X_i r.v. values ± 1 equally likely.
 $X = X_1 + \dots + X_n$ random walk length n .

$$E(X) = 0, V(X) = n, \lim_{n \rightarrow \infty} \frac{E(|X|)}{\sqrt{n}} = \sqrt{\frac{2}{\pi}} \ominus$$

Product Replacement Algorithm [CRL-G & LHS]

Input: G finite group and k fixed generators g_1, \dots, g_k

Output: Random elt of G

Method: Random walk of ordered (generating) k -tuples

First k -tuple is (g_1, \dots, g_k) .

Choose random $i \neq j$. Then k -tuple after $(a_1, \dots, a_k) \in G^k$ has
 $a_i := a_i a_j$ with probability $\frac{1}{2}$, and $a_j := a_j a_i$ with probability $\frac{1}{2}$

After “sufficiently long walk” return a_1

3. Constructing random elts: algorithms/Markov chains

- ⊕ If $k \geq 2K$ where $K := \max$ size of irredundant gen set of G , then $\lim_{t \rightarrow \infty} Q_t = U$ where Q_t = random walk of length t on generating k -tuples
- ⊖ Commonly used when $k \ll 2K$ e.g. $G = S_n^m$ degree mn , $K \geq m(n-1)$.
- ⊕ After $O(n^c)$ steps can get words of length $O(c^n)$.
- ⊖ [Babai & Pak 2000] strong bias to PRA
 - Really want PRA with input ε and choose t such that $\|Q_t - U\|_{tv} < \varepsilon$.

Main contribution

Tools for analyzing randomness for infinite matrix group algorithms

Thank you!

3. Constructing random elts: algorithms/Markov chains

- ⊕ If $k \geq 2K$ where $K := \max$ size of irredundant gen set of G , then $\lim_{t \rightarrow \infty} Q_t = U$ where Q_t = random walk of length t on generating k -tuples
- ⊖ Commonly used when $k \ll 2K$ e.g. $G = S_n^m$ degree mn , $K \geq m(n-1)$.
- ⊕ After $O(n^c)$ steps can get words of length $O(c^n)$.
- ⊖ [Babai & Pak 2000] strong bias to PRA
 - Really want PRA with input ε and choose t such that $\|Q_t - U\|_{tv} < \varepsilon$.

Main contribution

Tools for analyzing randomness for infinite matrix group algorithms

Thank you!

3. Constructing random elts: algorithms/Markov chains

- ⊕ If $k \geq 2K$ where $K := \max$ size of irredundant gen set of G , then $\lim_{t \rightarrow \infty} Q_t = U$ where Q_t = random walk of length t on generating k -tuples
- ⊖ Commonly used when $k \ll 2K$ e.g. $G = S_n^m$ degree mn , $K \geq m(n-1)$.
- ⊕ After $O(n^c)$ steps can get words of length $O(c^n)$.
- ⊖ [Babai & Pak 2000] strong bias to PRA
 - Really want PRA with input ε and choose t such that $\|Q_t - U\|_{tv} < \varepsilon$.

Main contribution

Tools for analyzing randomness for infinite matrix group algorithms

Thank you!