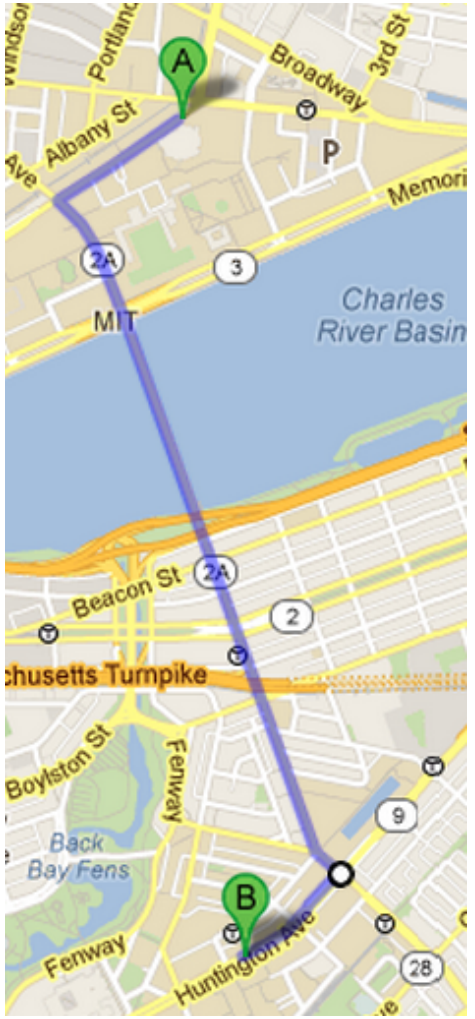
Extractors for Turing-machine sources

Random 2012

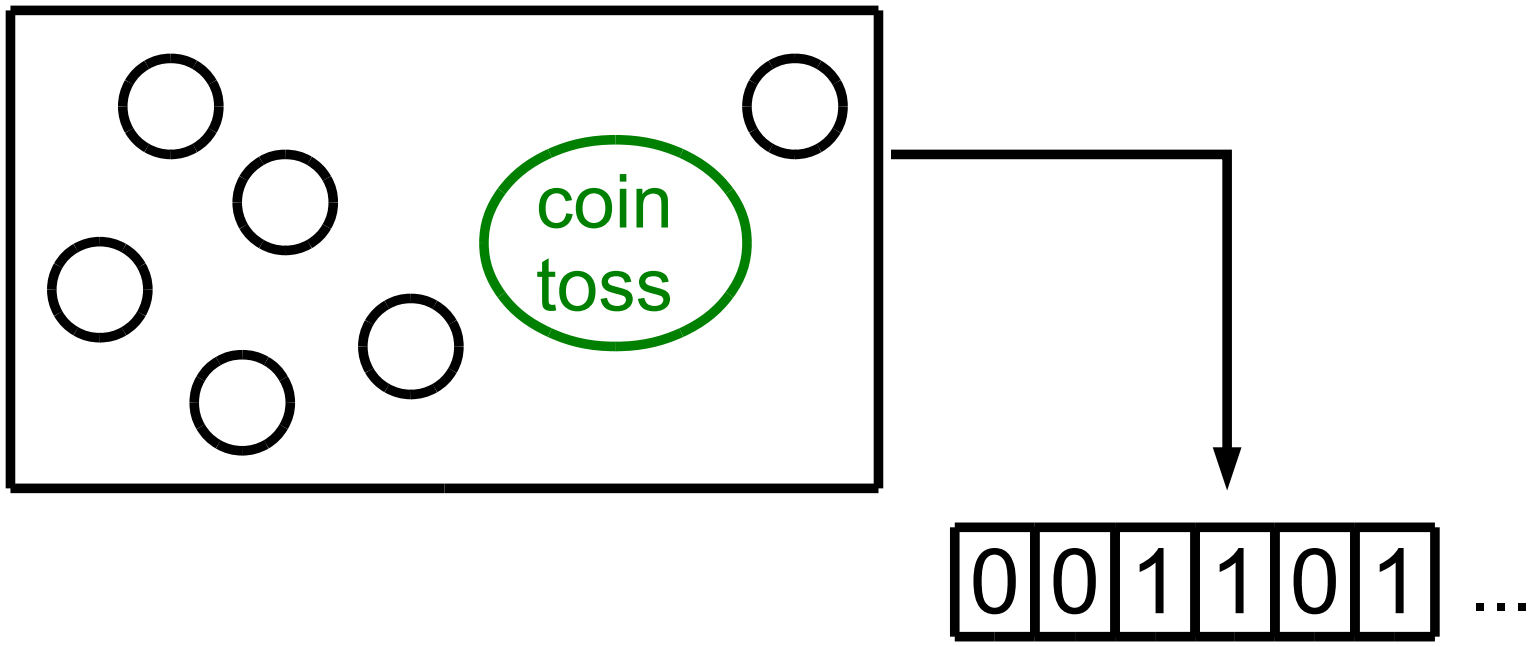
Emanuele Viola

Northeastern University



Turing-machine source

- One-tape machines, initialized to blank (all-zero)
- "Coin-toss" state: writes random bit



- When computation is over, first n bits on tape are sample

This work: extractors

- Theorem: From Turing-machine n-bit source running in time $\leq n^{1.9}$ and with min-entropy $k \geq n^{0.9}$:

  Extract $n^{\Omega(1)}$ bits, $\exp(-n^{\Omega(1)})$ close to uniform

- Matches $\Omega(n^2)$ time lower bound standing since 1960s

This work: sampling lower bound

- Theorem Turing-machine running in time $\leq n^{1.9}$
  cannot sample (X, Y, InnerProduct(X,Y))
  for |X| = |Y| = n

Outline of talk

- Overview of results
- Proof of main theorem
- The complexity of distributions

This work: extractors

- Theorem: From Turing-machine n-bit source running in time $\leq n^{1.9}$ and with min-entropy $k \geq n^{0.9}$:

  Extract $n^{\Omega(1)}$ bits, $\exp(-n^{\Omega(1)})$ close to uniform

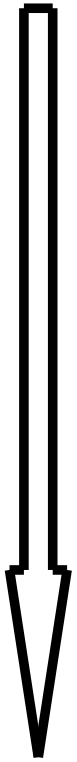
- Proof idea:

  1) Simulate source by one-way, low-memory source

  2) Use extractors [Kamp Rao Vadhan Zuckerman]
  [Chor Goldreich]

Simulate Turing-machine in one-way fashion

Time



0 H	0	0	0	0	0
1	**0 H**	0	0	0	0
1 H	0	0	0	0	0
1	**0 H**	0	0	0	0
1	0	**0 H**	0	0	0
1	0	0	**0 H**	0	0
1	0	0	0	**0 H**	0
1	0	0	**0 H**	0	0
1	0	0	1	**0 H**	0
1	0	0	1	0	**0 H**

= output sample

- Variant of crossing-sequences [Hennie]
- Sample one column at the time, one-way
- Few crossings ⇨ short description (little memory)

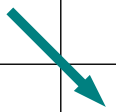
Demo

= output sample

- Sample first column
- 3 crossings, too many, keep sampling

Demo

0 H	0				
1	**0 H**				
1 H	0				
1	**0 H**				
1	0				
1	0				
1	0				
1	0				
1	0				
1	0				



= output sample

- Sample first two columns
- Only one crossing ⇨ all you need to store to continue
- Output first two bits, forget the rest, and continue

Demo

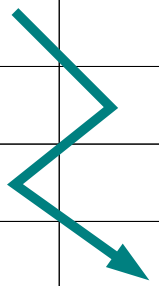
		0			
		0			
		0			
	0 H	0			
		0 H			
		0			
		0			
		0			
		0			
1	0	0			

= output sample

- Sample next column
- Only one crossing ⇨ all you need to store to continue
- Output next bit, forget the rest, and continue

Demo

			0		
			0		
			0		
			0		
		0 H	0		
			0 H		
			0		
			0 H		
			1		
1	0	0	1		

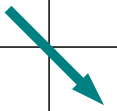


= output sample

- Sample next column
- 3 crossings, too many, keep sampling

Demo

			0	0	
			0	0	
			0	0	
			0	0	
		0 H	0	0	
			0 H	0	
			0	**0 H**	
			0 H	0	
			1	**0 H**	
1	0	0	1	0	



= output sample

- Sample next two columns
- Only one crossing ⇨ all you need to store to continue
- Output next two bits, forget the rest, and continue

Demo

					0
					0
					0
					0
					0
					0
					0
					0
				0 H	0
1	0	0	1	0	**0 H**

= output sample

- Sample last column
- Output last bit

Crossing sequences

- For simulating Turing machine with little memory

Classical E.g., is x palindrome?	**Sampling**
Error	No error
2 blocks generally enough	Useful to have more blocks

Outline of talk

- Overview of results
- Proof of main theorem
- The complexity of distributions

Objects of study

- Classical: Efficient Computation

  $f : \text{INPUT} \rightarrow \text{OUTPUT}$

- Alternative: Efficient Sampling

  $f : \text{RANDOM BITS} \rightarrow \text{OUTPUT DISTRIBUTION}$

Sampling literature

- Generate Random Factored Numbers [Bach '85, Kalai]

- Random Generation of Combinatorial Structures from a Uniform Distribution [Jerrum Valiant Vazirani '86]

- The Quantum Communication Complexity of Sampling [Ambainis Schulman Ta-Shma Vazirani Wigderson '98]

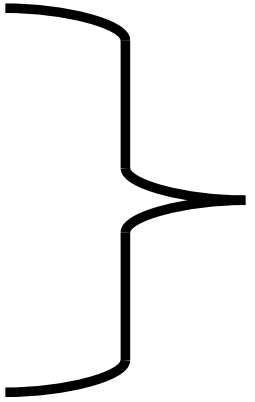
- On the Implementation of Huge Random Objects [Goldreich Goldwasser Nussboim '03]

Recent papers revisit sampling

- First sampling lower bounds for restricted models, e.g. AC^0

- New connections to:
  succinct data structures,
  combinatorics,
  and extractors tigthening [Trevisan Vadhan '00]

- [V. '09]
  [Lovett V.]
  [De Watson] [V.]
  [Beck Impagliazzo Lovett]



Mostly circuit models

- This work: Turing machines