

# Randomness buys depth for approximate counting

Emanuele Viola

Northeastern University

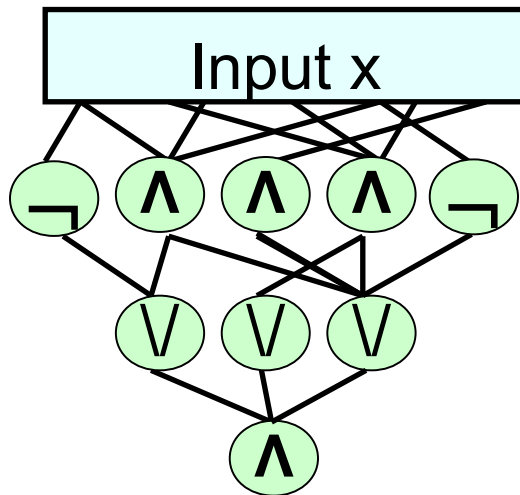
October 2011

# Approximate counting

- $\epsilon$ -approx count (majority) distinguish weights  $n(1/2 \pm \epsilon)$

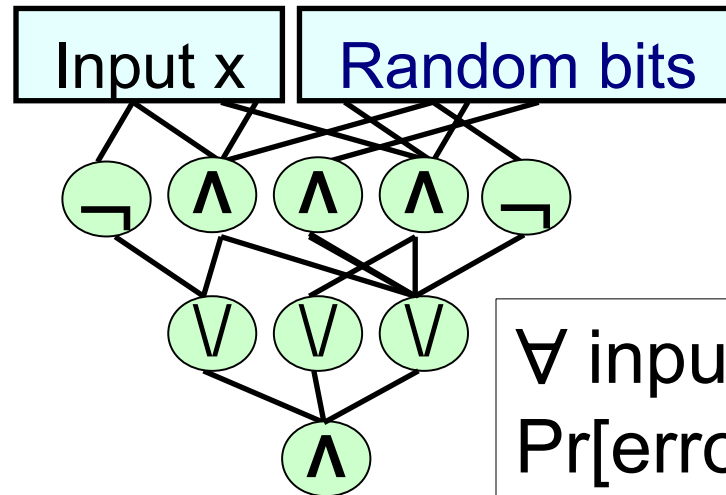
Input:  $x \in \{0,1\}^n$       Output:  $\begin{cases} 1 & \text{if } \sum x_i > n(1/2 + \epsilon) \\ 0 & \text{if } \sum x_i < n(1/2 - \epsilon) \\ 1 \text{ or } 0 & \text{otherwise} \end{cases}$

- Model:  $AC^0$



Depth  $d=3$

- $BP AC^0$



$\forall$  input  $x$ ,  
 $\Pr[\text{error}] < 1/3$

# Approx count in $AC^0$ : surprising and useful

- [Ajtai '83] 0.1-approx count in depth 3, poly(n)-size
- [V] above, explicit
- [Sipser] [Gacs] [Lautemann]  $BPP \subseteq PH$
- [Stockmeyer]  $\#P$  approximated in PH
- [Goldwasser Sipser] approx count in AM
- [Chaudhuri Radhakrishnan]  $LC^0 \neq AC^0$
- .....

yet still gaps in our knowledge!

# Our results

- **$\epsilon$ -approx count** : distinguish weights  $n(1/2 \pm \epsilon)$

- **Theorem**: For every  $d$ :  **$\epsilon$ -approx count**  
in poly-size depth- $d$  BPAC<sup>0</sup>  $\iff \epsilon = \Omega(1/\log^{d-1} n)$  ;  
in poly-size depth- $d$  AC<sup>0</sup>  $\iff \epsilon = \Omega(1/\log^{d-3} n)$  .

Also, BP AC<sup>0</sup> circuits are explicit.

- Previously, depth estimated only within  $> 2$ .  
Could not differentiate between AC<sup>0</sup> and BP AC<sup>0</sup>

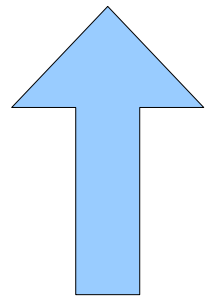
# Our results

- **Corollary:** Randomness buys depth.
- Analogy (all circuits non-uniform)
- BP Size  $n^{O(1)} =$  Size  $n^{O(1)}$  [Adleman]
- BP Size  $n^2 \stackrel{=?}{=} \text{Size } n^2$
- BP  $AC^0$  depth  $d \subseteq AC^0$  depth  $d+2$  [Ajtai Ben-Or]
- BP  $AC^0$  depth  $d \not\subseteq AC^0$  depth  $d+1$  [This work]

# Proof outline

- **Theorem:** For every  $d$ :  $\epsilon$ -approx count  
in poly-size depth- $d$  BPAC<sup>0</sup>  $\iff \epsilon = \Omega(1/\log^{d-1} n)$  ;  
in poly-size depth- $d$  AC<sup>0</sup>  $\iff \epsilon = \Omega(1/\log^{d-3} n)$  .

Also, BP AC<sup>0</sup> circuits are explicit.



BP AC<sup>0</sup> depth  $d \subseteq$  AC<sup>0</sup> depth  $d+2$   
[Ajtai Ben-Or]

- **Not in** depth- $d$  AC<sup>0</sup> for  $\epsilon = o(1/\log^{d-3} n)$
- **In** depth- $d$  BP AC<sup>0</sup> for  $\epsilon = \Omega(1/\log^{d-1} n)$

# Lower bound

- **Lemma:**  $o(1/\log^{d-3} n)$ -approx count **not in** depth- $d$   $AC^0$
- **Proof** by induction on  $d$ :
- **Base case**  $d = 3$ : [V]
- **Induction step:** Switching lemma [Hastad]  
Restriction: leave free  $1/\log n$  fraction variables  
multiply approximation parameter by  $1/\log n$   
Increase depth by 1. ◆

# Outline

- Lower bound
- Upper bound
- New pseudorandom generator

# Upper bound

- **Lemma** ( $\epsilon = 1/\log^{d-1} n$ )-approx count in depth-d BPAC<sup>0</sup>
- [Amano 09, Brody Verbin 10]  
**deterministic** depth-d circuit distinguishing  
i.i.d. bits  $X_1, X_2, \dots, X_n$   $\Pr[X_i=1] = 1/2 \pm \underbrace{1/\log^{d-1} n}_{\epsilon}$
- Right tradeoff, different setting

# Upper bound

- In proof of [Amano 09, Brody Verbin 10] deterministic depth- $(d-1)$  distinguishing i.i.d. bits  $X_1, X_2, \dots, X_n$   $\Pr[X_i=1] = n^{-1} (1 \pm \epsilon \log n)$
- We reduce to above

- Want BP DNF (depth 2)  $D : \forall x$

$$\sum x_i = n(1/2 \pm \epsilon) \Rightarrow \Pr[D(x)=1] = n^{-1}(1 \pm \epsilon \log n)$$

# Upper bound

- Want BP DNF (depth 2)  $D : \forall x$

$$\sum x_i = n(1/2 \pm \epsilon) \Rightarrow \Pr[D(x)=1] = n^{-1}(1 \pm \epsilon \log n)$$

- **Attempt:** AND  $\log(n)$  **randomly**-selected bits  
Probability reduction  $\checkmark$   
 $\log^2 n$  randomness  $\Rightarrow$  not poly-size DNF **X**
- **Better:** AND  $\log(n)$  **pseudorandomly**-selected bits  
 $O(\log n)$  randomness  $\Rightarrow$  poly-size  $\checkmark$   
Probability reduction: **Non-explicit:** chernoff bound.  
**Explicit?**

# Explicit upper bound

- **Need** pseudorandom generator:
  - fools rectangles  $A \times A \times \dots \times A \subseteq [n]^{\log n}$   
( $A$  = input bits set to 1)
  - seed length  $O(\log n)$
  - error  $< 1/n$  (distinguish  $n^{-1}(1 \pm \epsilon \log n)$ )
- Previous generators: seed  $> \log n \log \log n$ 
  - Expander walk:** [Ajtai Komlos Szemerédi]
  - For space:** [Nisan], [N Zuckerman],  
[Impagliazzo N Wigderson]
  - For rectangles:** [Even Goldreich Luby N Velickovic]  
[Armoni Saks W Zhou], [Lu]

# Our pseudorandom generator

- **Theorem:** Pseudorandom generator:
  - fools  $A \times A \times \dots \times A \subseteq [n]^{\log n}$  ( $|A| = n/2$ )
  - seed length  $O(\log n)$
  - error  $< 1/n$
- Two-level expander walk of length  $(\log n)^{1/2}$   
Simple calculations  $\neq$  previous generators  
 $\approx$  approx count in  $AC^0$
- **Challenge:** make error  $1/n^2$

# Conclusion

- **Theorem:** For every  $d$ :  $\epsilon$ -approx count  
in poly-size depth- $d$  BPAC<sup>0</sup>  $\iff \epsilon = \Omega(1/\log^{d-1} n)$  ;  
in poly-size depth- $d$  AC<sup>0</sup>  $\iff \epsilon = \Omega(1/\log^{d-3} n)$  .

Also, BP AC<sup>0</sup> circuits are explicit.

- **Pseudorandom generator:** fool  $A \times \dots \times A \subseteq [n]^{\log n}$   
error  $1/n$ , seed  $O(\log n)$  ( $|A| = n/2$ )
- **Randomness buys depth:**  
BP AC<sup>0</sup> depth  $d \not\subseteq$  AC<sup>0</sup> depth  $d+1$
- **Match** BP AC<sup>0</sup> depth  $d \subseteq$  AC<sup>0</sup> depth  $d+2$  [Ajtai Ben-Or]

- $\Sigma \Pi \sqrt{\cup} \supseteq \subset \subseteq \nabla \wedge$

- $\approx \forall \exists \Omega \Theta \omega \alpha \beta \gamma \delta$

- $\rightarrow \downarrow \Rightarrow \Uparrow \Leftarrow \Leftrightarrow$

- $\neq \approx$

- $\Theta \omega$

- $\in \notin$

- $\pm$

- $\Sigma \Pi \sqrt{\cup} \neq \supseteq \subset \subseteq \in \downarrow \Rightarrow \Uparrow \Leftarrow \Leftrightarrow \nabla \wedge \geq \leq \forall \exists \Omega \alpha \beta \gamma \delta \rightarrow$

- $\neq \approx \tau \Delta \Theta$



Recall: edit style changes ALL settings.

- Click on “line” for just the one you highlight
- To rotate, right-click, position and size
- Format->Style & Formatting allows to set default font