Capacity achieving multi-write WOM codes

Amir Shpilka
Technion

Talk outline

- The problem
- A "dream" scheme
- The actual solution

Write-Once Memory (WOM)

- Setting: a memory cell can only be written once. (can only change 0 to 1 and not vice versa)
- Motivation:
  - Punch Cards `70s
  - Laser Disks `90s
  - Flash Memory `00s
- Flash memory: easy to insert electrons to a single cell but removing requires erasing entire block. Erasing shortens life of flash memory.

The problem

- Device WOM coding schemes to allow several writes and handle as many bits as possible.
- Start: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
- Phase I: store some string

1	0	1	0	1	0	1	1	0	0	1	0	0

- Phase II: Can we store more bits in a way that will enable recovery?
- Phase III: and so on...

The problem

- Setting: a memory cell can only be written once. (can only change 0 to 1 and not vice versa)
- Problem: Device WOM coding schemes to allow several writes and handle as many bits as possible.
- (Bad) Solution: partition memory to t blocks. In round i store message on block i.
- Can store a total of n bits
- Can we do better?
  - In 2 writes store a total of more than n bits?
  - What about t writes?
  - Main difficulty: Recovering message in i'th round without knowing the past!

Example

- Rivest-Shamir Scheme: storing 2 bits (4 symbols) in each phase on a 2-write WOM of size 3 (total 4 bits)
- Note: symbol uniquely determined.
- Rate: (2+2)/3.
- Can we do better?

symbol	1st	2nd
1	000	111
2	001	110
3	010	101
4	100	011

Capacity of WOM

- Rivest-Shamir: max rate (capacity) for 2-writes = log(3)
- Generally, for t-writes capacity = log(t+1) [Fu&Vinck]
- R-S encoding schemes highly inefficient (exp(exp(1/ε)) time to be ε-close to capacity)
- Important goal: efficient capacity achieving t-round WOM codes.

(Some) Known Constructions

- [Rivest & Shamir]: Capacity of 2(3)-writes is log(3)≈1.58 (2). Non explicit constructions.
- [...,Cohen Godlewski & Merkx, Wu, Yaakobi Kaiser Siegel Vardy & Wolf]: 2-write WOM codes with rate 1.49
- [Kaiser Yaakobi Siegel Vardy & Wolf]: 3-write WOM code of rate 1.61.
- [S] Capacity achieving 2-write WOM. Block length exp(1/ε), where ε = gap to capacity.
- [Yaakobi-S] 3-write WOM of rate 1.885-ε (block length exp(1/ε))
- [Burshtein & Strugatski] Capacity achieving t-write WOM, based on polar codes. Block length poly(1/ε). Works w.h.p. (i.e. for most messages).

Our results

- Capacity achieving t-write WOM code.
  Block length is exp(1/ε)
  (to achieve rate log(t+1)-ε)

- New technique based on hash functions.

- Method easily applies to every alphabet q

- Comparison to [Burshtein & Strugatski]:
  Ours: Always works. Block length exp(1/ε)
  B-S: Works w.h.p. Block length poly(1/ε)

Talk outline

- The problem
- An ideal solution
- The actual solution

Basic 2-write Scheme

- Intuitively: want to minimize the number of 1's after first write.
- Hence, if we transmit k bits then we will write them on subsets of size ≤ pn where H(p)=k/n.
- In 2nd write we'll need to write the remaining (1-p)n bits and allow decoding. This is the main obstacle.
  Note: we have to recover message without knowing what was written in the 1st round.

2-write Dream Scheme

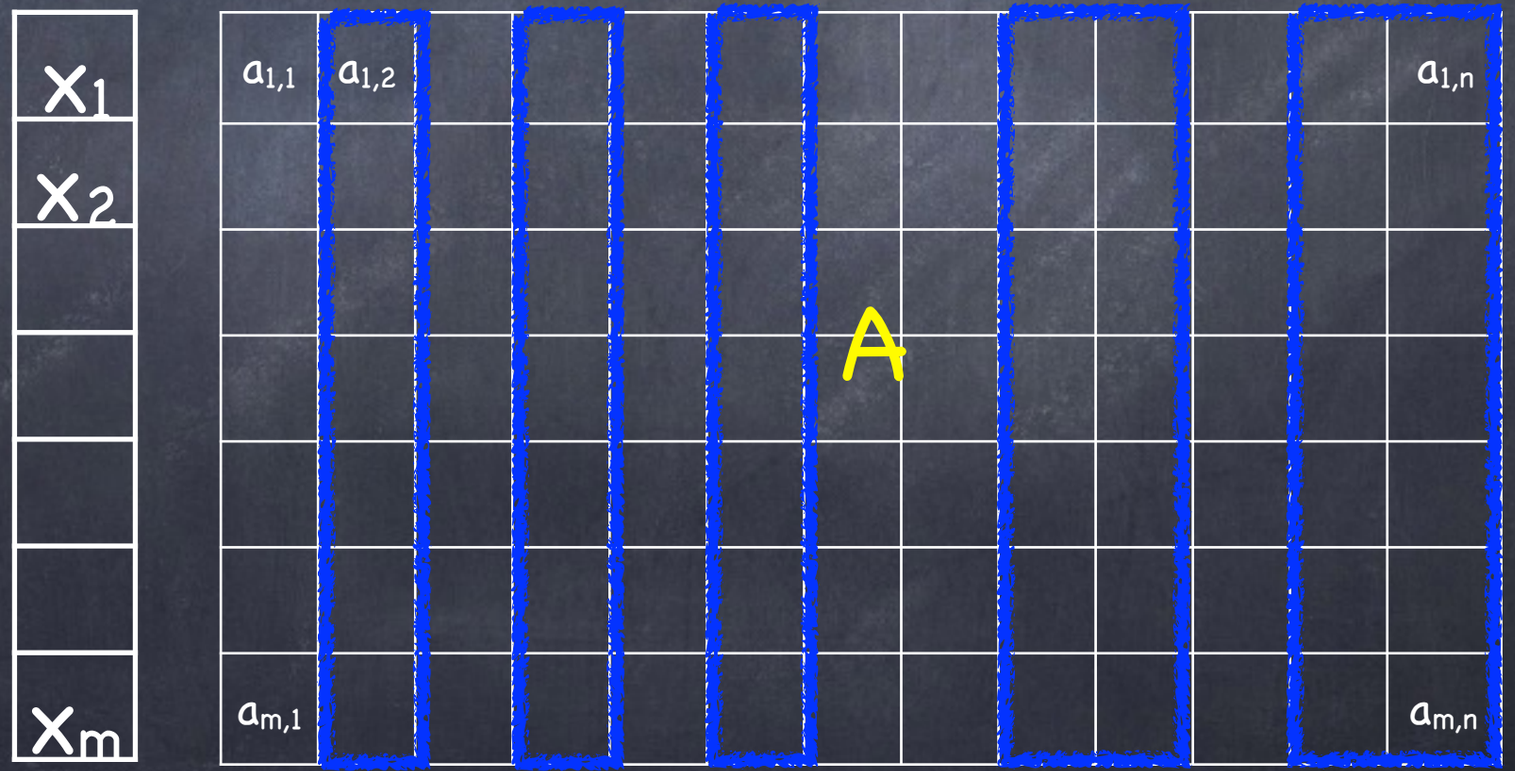
In A any m×m submatrix invertible

First round:

1	0	1	0	1	0	1	1	0	0	1	0	0

Second round input: x

Solve system of equations



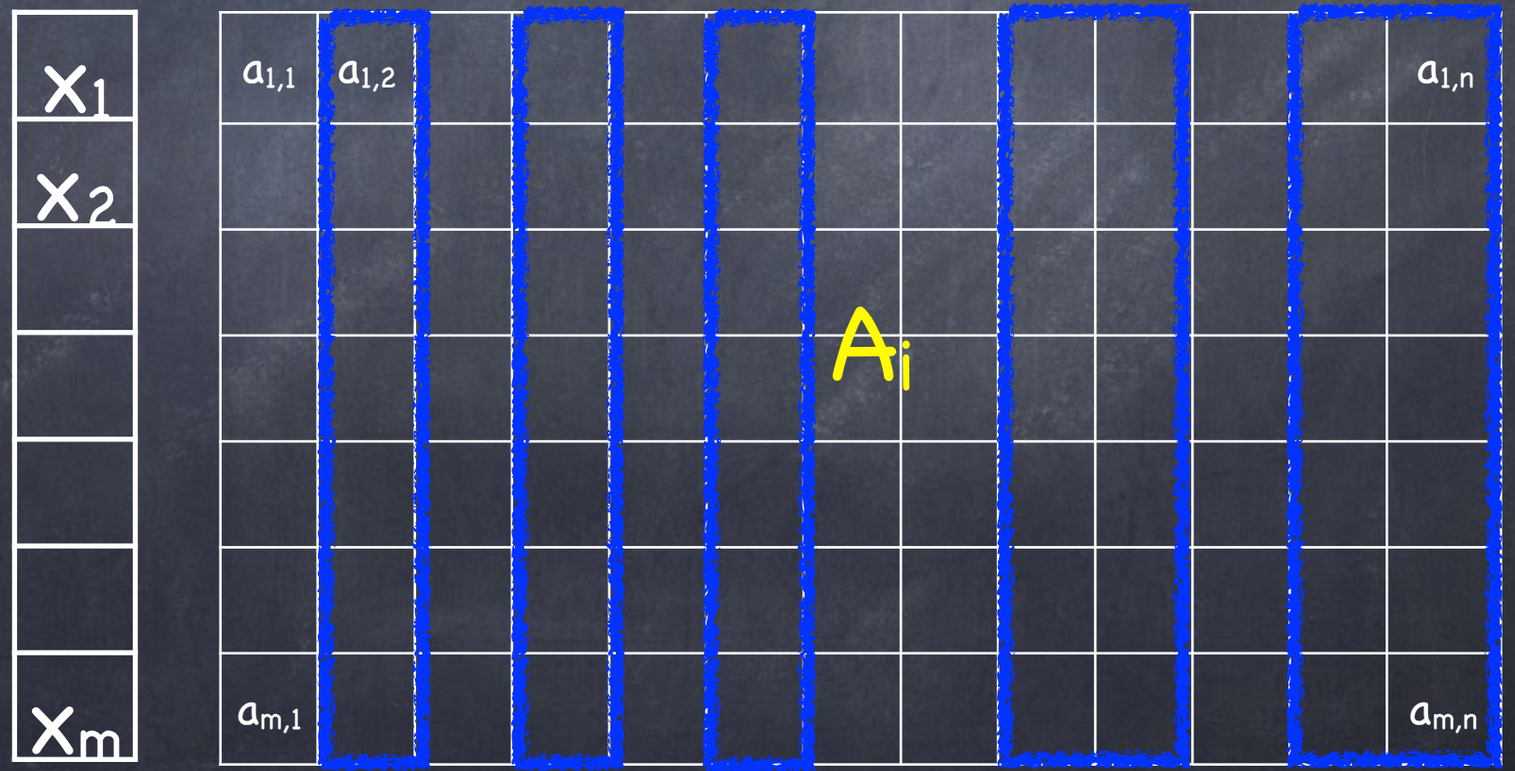
t-write Dream Scheme

Round i: A_i "good" for i'th round

After i-1 writes

i'th write input: x

Find sparse solution to system of equations



Summary of dream scheme

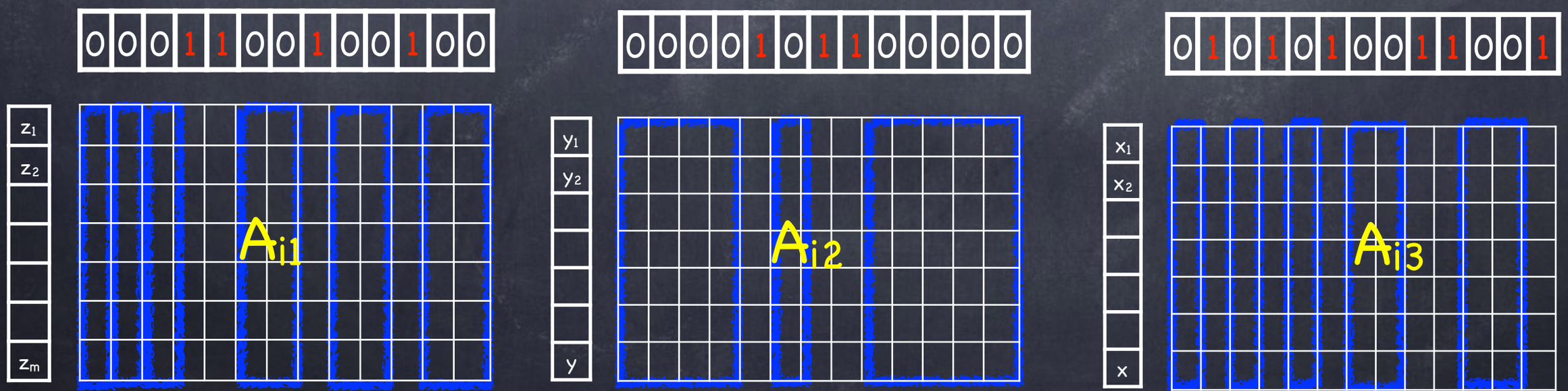
- Dream scheme looks great but:
  - Finding sparse solution is NP hard:
    - Running time exponential in block length
  - No such Boolean A_i:
    - Can pick Random A_i
    - Storing A_i requires large space - hurts rate

Talk outline

- The problem
- An ideal solution
- The actual solution

Fixing the running time

- Assume block length in previous scheme is $n = 1/\varepsilon$
- Finding a sparse solution takes $2^n = \exp(1/\varepsilon)$
- Problem: running time exponential in block length
- (cheap) Solution: concatenate many copies of basic building block



More problems?

- Assume block length in previous scheme is n= 1/ε
- Finding a sparse solution takes 2^n = exp(1/ε)
- Problem: running time exponential in block length
- Solution: concatenate many copies of basic building block!
- Now: N copies of basic construction.
  If N=exp(n) then running time is N·exp(n)=poly(N)
- Problem:
  - Still, good A_i do not exist - can pick Random A_{ij}
  - Storing A_{ij} requires large space - hurts rate
  - Is there a single good A_i?

Enter hash functions

- Problem: Find a m×n matrix A such that for N systems of equations $\{x_i=Ab_i\}$ can find sparse solutions $\{b_i\}$.
- No universal A
- But, if A is a picked from a good hash family then it works whp.
- Hash family: $\mathcal{H}(n,m) = \{h_{a,b}(x) = (ax+b)_{[m]}\}$ (the first m bits of ax+b, where $a,b,x \in F_{2^n}$)
- Main property: Any set S of size 2^{m+d} is mapped onto $\{0,1\}^m$ by random h in $\mathcal{H}(n,m)$ with probability 1- exp(-d).
- Conclusion: can find $h \in \mathcal{H}(n,m)$ that works for all N copies.
- Note: need to write name of h on memory!

Hash functions as extractors

- Hash family: $\mathcal{H}(n,m) = \{h_{a,b}(x) = (ax+b)_{[m]}\}$
- (weak) Pairwise independence: $\forall$ x,y
  $Pr_h[h(x)=h(y)] = (1/2^m)$
- Leftover hash lemma [Impagliazzo]:
  X uniform in set of size $2^m(1/\varepsilon)^2$ and h random in $\mathcal{H}(n,m)$
  then $Ext(x,h) := (h,h(x))$
  is ε-close to uniform distribution on $\{0,1\}^{2n+m}$
  (in statistical (L_1) distance)
- Lesson: if parameters chosen carefully then one h good for all N instances

Recap

- Block length is $n \times N + O(nt)$
- Assume after (i-1)th round memory stores $w_1,\ldots,w_N$, and $x_1,\ldots,x_N \in \{0,1\}^{r_i}$ inputs for i'th write. (r_i chosen to achieve capacity a-la Fu-Vinck)
- Find h_i and $w'_1,\ldots,w'_N$ such that
  - coordinate-wise $w_j \leq w'_j$
  - $h_i(w'_j)=x_j$, for all j.
- Encoding: Write w'_j to memory as well as index of h_i
- Decoding: output $h_i(w'_j)$
- Need block-length $N=\exp(1/\varepsilon)$
- If parameters chosen carefully: encoding and decoding in polynomial time. Rate achieves capacity

Conclusion

- First deterministic capacity achieving t-write WOM with polynomial encoding and decoding
- Can be extended to every alphabet q and any point $(r_1,\ldots,r_t)$ in the capacity region
- Block length exponential in gap to capacity. Improve! (open also for deterministic scheme for 2-write)
- Construction of [S] for 2-writes has block length poly(1/ε) if we allow randomness. Extend to general t!
- Questions may require new techniques (not clear how to find sparse solutions efficiently)

Thank You!