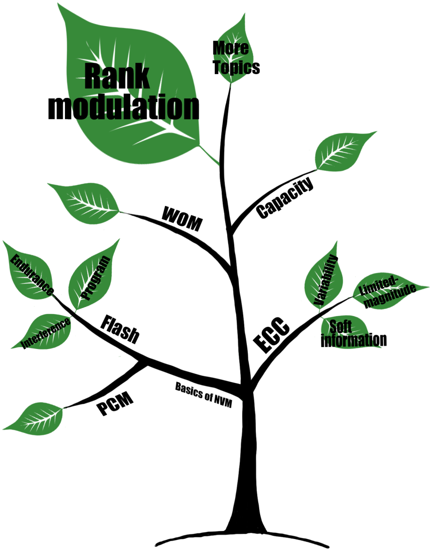
ISIT Tutorial – Part II

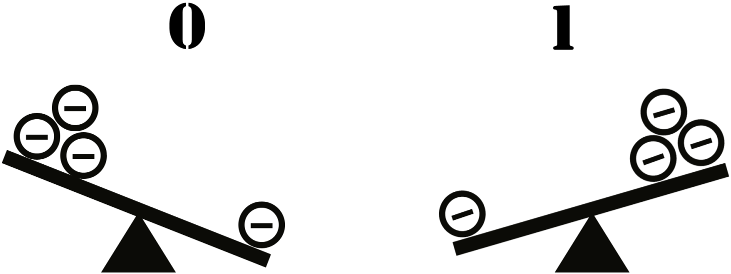
Anxiao (Andrew) Jiang
(joint presentation with Lara Dolecek)

Department of Computer Science and Engineering
Texas A&M University

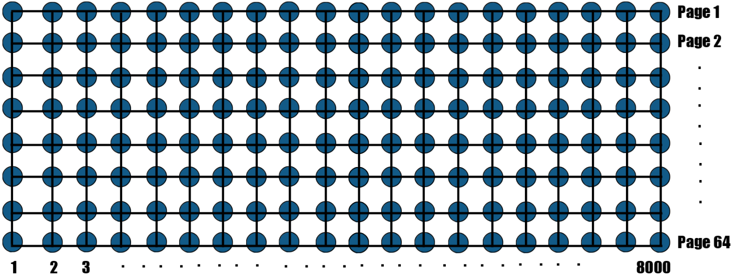
June 29, 2014


Rank modulation
More Topics
Capacity
WOM
Endurance
Program
Interference
Flash
Variability
Limited-magnitude
Soft information
ECC
PCM
Basics of NVM

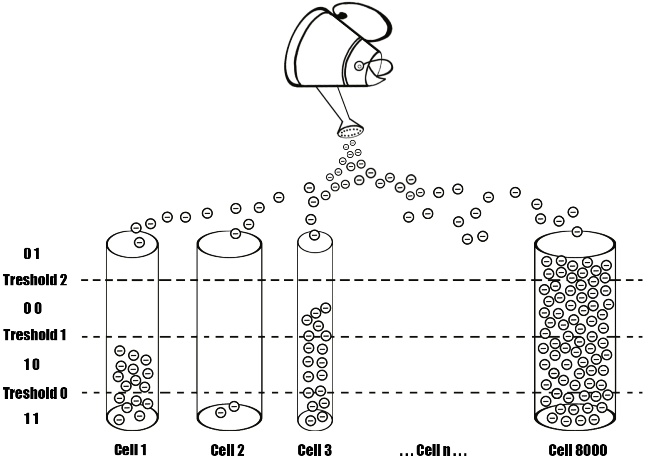
Rank Modulation



Parallel cell programming for MLC



Challenges of parallel cell programming for MLC



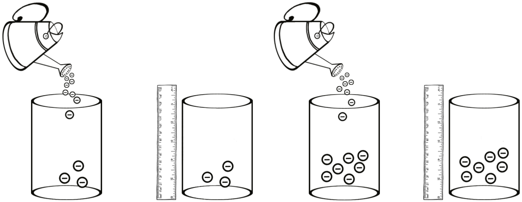
Muti-level cell (MLC): Parallel programming, common thresholds, heterogeneous cells, random process of charge injection, over-injection of charge, disturbs and inter-cell interference, block erasure, difficulty in adjusting threshold voltages, very careful repeated charge injection and measuring.

Challenges of parallel cell programming for MLC

Dilemma among:

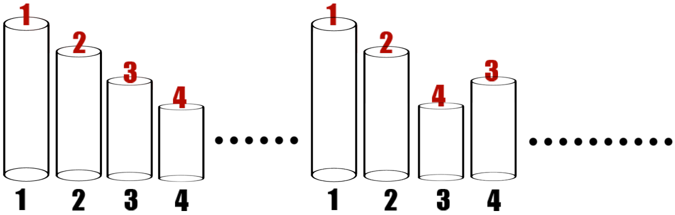
- Capacity
- Speed
- Reliability and endurance

Due to: Inflexibility in adjusting cell levels.



Definition (Rank Modulation)

Use the relative order of cell levels to represent data.

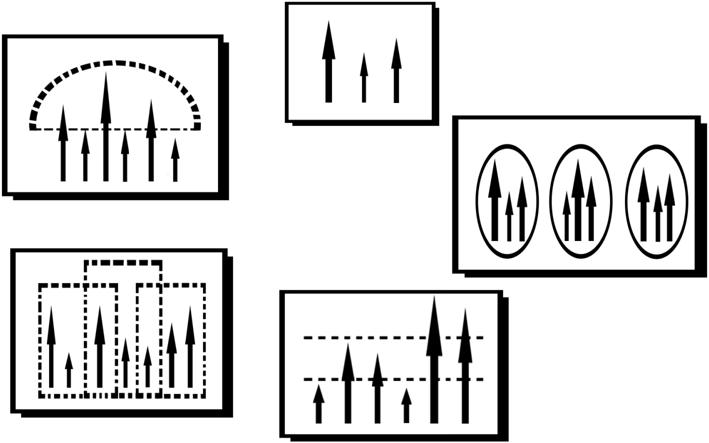


A. Jiang, R. Mateescu, M. Schwartz and J. Bruck, "Rank modulation for flash memories," in ISIT 2008.

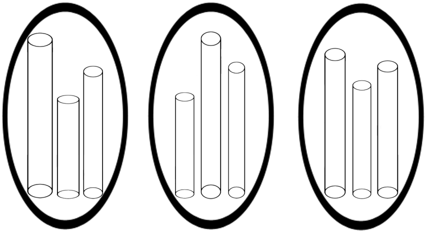
Some advantages of rank modulation:

1. Flexibility in adjusting relative cells levels, even though we can only increase cell levels;
2. Tolerance for charge leakage / cell level drifting;
3. Enable memory scrubbing without block erasure.

2. Extended models of rank modulation



- Extension: Rank modulation with multiple permutations

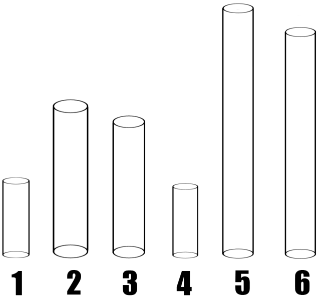


Some advantages: (1) Enable the building of long codes; (2) Cells in different permutations can have very close cell levels.

F. Zhang, H. Pfister and A. Jiang, "LDPC codes for rank modulation in flash memories," in ISIT 2010.

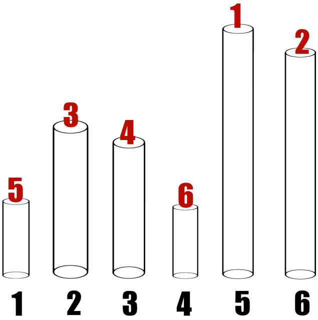
- Extension: Rank modulation with multi-set permutation

Example: A group of $n = 6$ cells

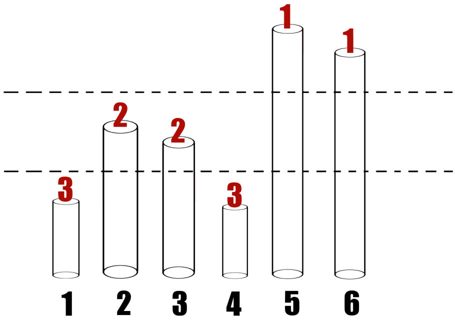


Some advantages: Similar to multiple permutations, but more suitable if cells can be programmed accurately.

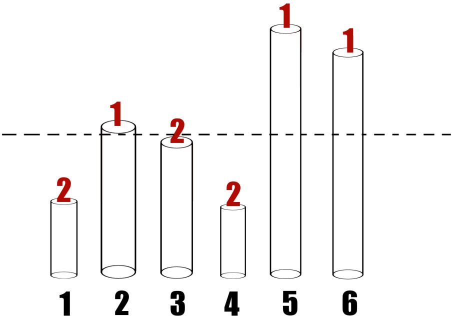
Example: Every rank has one cell



Example: Every rank has two cells



Example: Every rank has three cells



- Extension: Bounded rank modulation

  Z. Wang, A. Jiang and J. Bruck, "On the capacity of bounded rank modulation for flash memories," in ISIT 2009.

- Extension: Local rank modulation

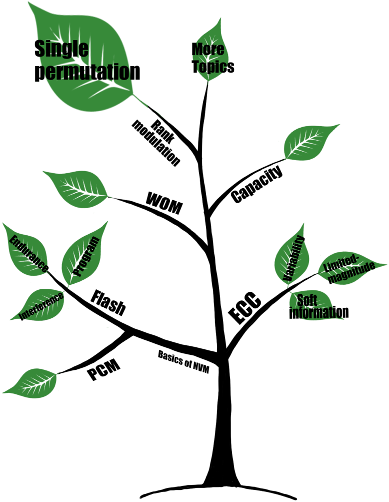
  M. Schwartz, "Constant-weight Gray codes for local rank modulation," in ISIT 2010.

- Extension: Partial rank modulation:

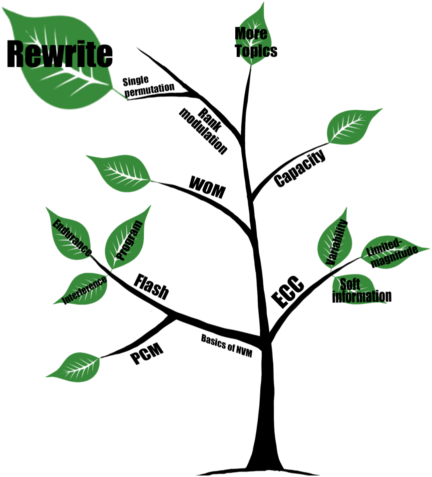
  Z. Wang and J. Bruck, "Partial rank modulation for flash memories," in ISIT 2010.

Some advantages: Faster read, and/or enabling long codewords.

Coding with single permutation

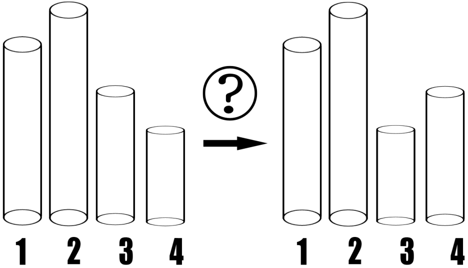


Single permutation: **rewrite**

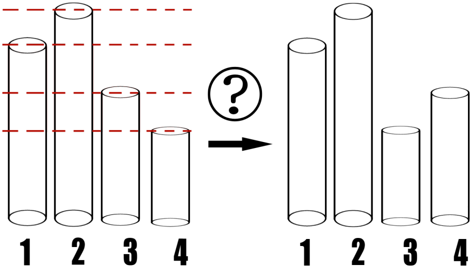


Definition (Rewrite)

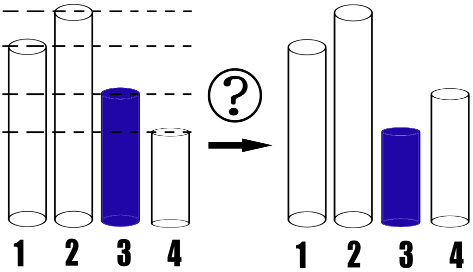
Change data by changing the permutation – by moving cell levels up.



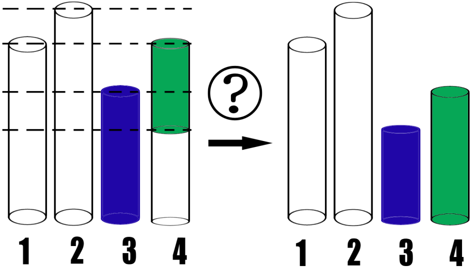
Virtual levels to help us estimate rewriting cost (increase in cell levels).



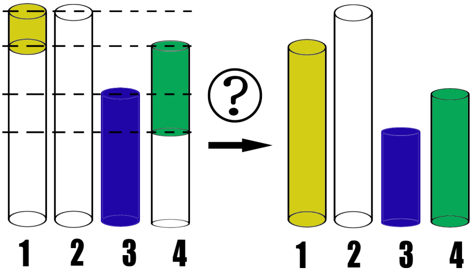
Get the permutation right from low to high.



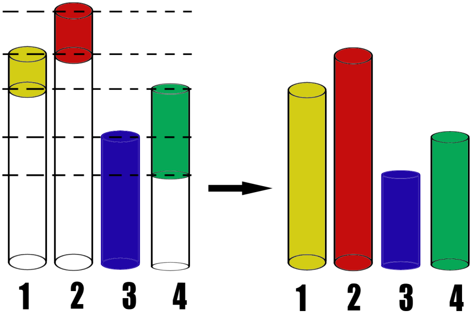
Get the permutation right from low to high.



Get the permutation right from low to high.



Rewriting cost: **1**.



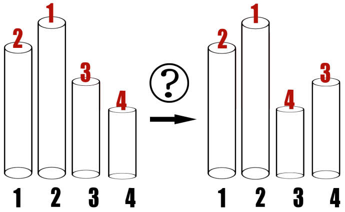
Can we know rewriting cost directly from the two permutations, **without** running the previous rewriting algorithm?

Can we know rewriting cost directly from the two permutations, **without** running the previous rewriting algorithm?**Yes**.

Can we know rewriting cost directly from the two permutations, **without** running the previous rewriting algorithm?**Yes**.

Theorem (Rewriting cost)

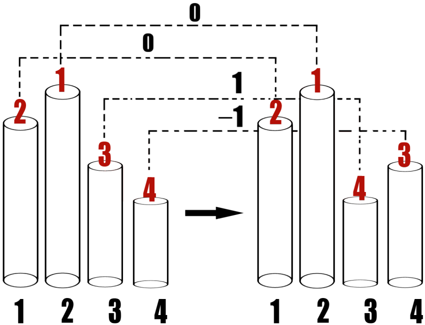
The rewriting cost equals the maximal increase in cells' ranks.



E. En Gad, A. Jiang and J. Bruck, Compressed encoding for rank modulation, ISIT 2011.

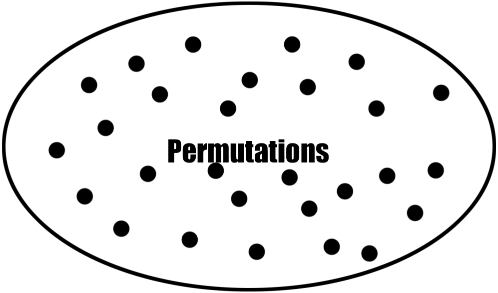
Theorem (Rewriting cost)

The rewriting cost equals the maximal increase in cells' ranks.

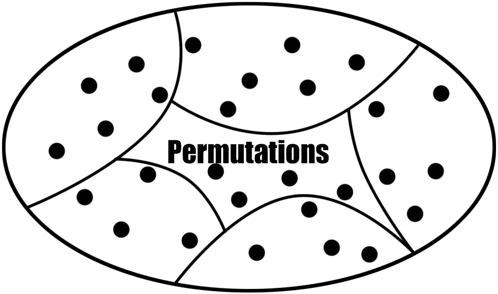


Rewriting cost: **1**.

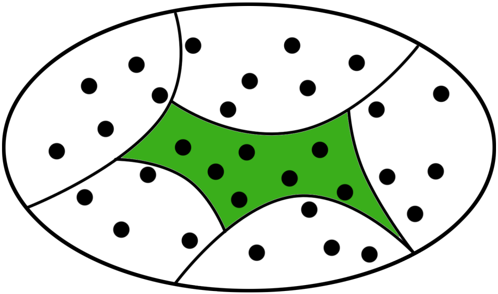
Consider: Store data of k values in n cells.



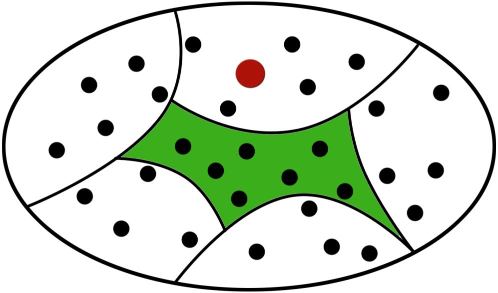
Every subset of permutations represents one value of the data.



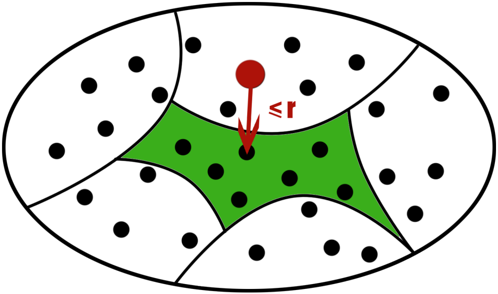
Consider one such subset, which represents one particular data value.



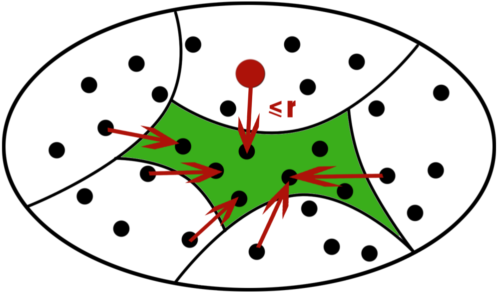
Say the red dot is the current state of the n cells. We want to change the data to the value represented by the green subset $\cdots$



Bound the rewriting cost by r.



The green subset needs to be a *dominating set* of incoming covering radius r.

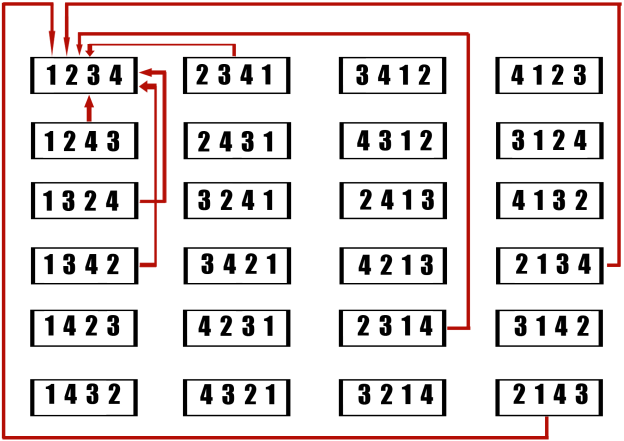


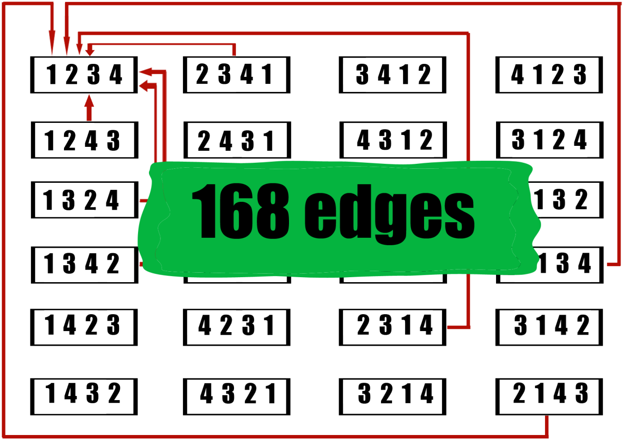
We show an optimal code as an example.
Parameters: $n = 4$ cells, $k = 6$ data values, rewriting cost $r = 1$.

1 2 3 4	2 3 4 1	3 4 1 2	4 1 2 3
1 2 4 3	2 4 3 1	4 3 1 2	3 1 2 4
1 3 2 4	3 2 4 1	2 4 1 3	4 1 3 2
1 3 4 2	3 4 2 1	4 2 1 3	2 1 3 4
1 4 2 3	4 2 3 1	2 3 1 4	3 1 4 2
1 4 3 2	4 3 2 1	3 2 1 4	2 1 4 3

E. En Gad, A. Jiang and J. Bruck, "Compressed encoding for rank modulation," in ISIT 2011.

Every permutation has $r!(r+1)^{n-r}$ permutations within radius (rewriting cost) r, including itself.
So a permutation has $2^{n-1} - 1 = 7$ neighboring nodes within radius $r = 1$.




1 2 3 4
2 3 4 1
3 4 1 2
4 1 2 3
1 2 4 3
2 4 3 1
4 3 1 2
3 1 2 4
1 3 2 4
1 3 2
1 3 4 2
1 3 4
1 4 2 3
4 2 3 1
2 3 1 4
3 1 4 2
1 4 3 2
4 3 2 1
3 2 1 4
2 1 4 3
168 edges

1 2 3 4	2 3 4 1	3 4 1 2	4 1 2 3	Subgroup<(1,2,3,4)>
1 2 4 3	2 4 3 1	4 3 1 2	3 1 2 4	Coset
1 3 2 4	3 2 4 1	2 4 1 3	4 1 3 2	Coset
1 3 4 2	3 4 2 1	4 2 1 3	2 1 3 4	Coset
1 4 2 3	4 2 3 1	2 3 1 4	3 1 4 2	Coset
1 4 3 2	4 3 2 1	3 2 1 4	2 1 4 3	Coset

Every row (subgroup) is a dominating set of radius 1.

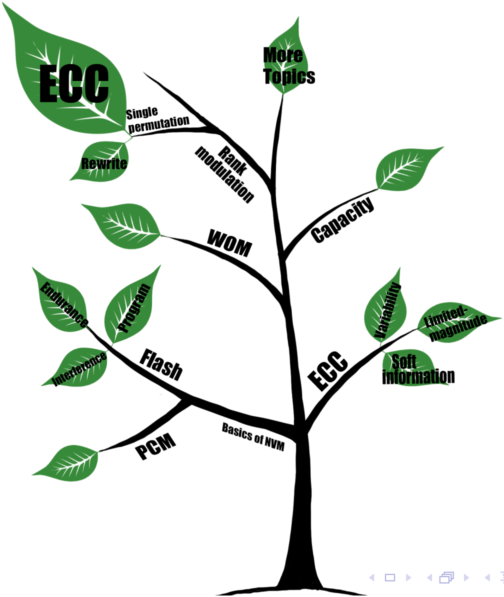
1 2 3 4	2 3 4 1	3 4 1 2	4 1 2 3	Subgroup<(1,2,3,4)>
1 2 4 3	2 4 3 1	4 3 1 2	3 1 2 4	Coset
1 3 2 4	3 2 4 1	2 4 1 3	4 1 3 2	Coset
1 3 4 2	3 4 2 1	4 2 1 3	2 1 3 4	Coset
1 4 2 3	4 2 3 1	2 3 1 4	3 1 4 2	Coset
1 4 3 2	4 3 2 1	3 2 1 4	2 1 4 3	Coset

Every row (subgroup) is a dominating set of radius 1.

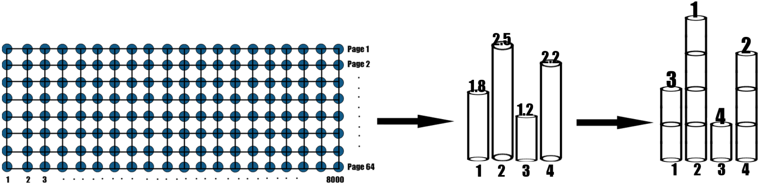
1 2 3 4	2 3 4 1	3 4 1 2	4 1 2 3	Subgroup<(1,2,3,4)>
1 2 4 3	2 4 3 1	4 3 1 2	3 1 2 4	Coset
1 3 2 4	3 2 4 1	2 4 1 3	4 1 3 2	Coset
1 3 4 2	3 4 2 1	4 2 1 3	2 1 3 4	Coset
1 4 2 3	4 2 3 1	2 3 1 4	3 1 4 2	Coset
1 4 3 2	4 3 2 1	3 2 1 4	2 1 4 3	Coset

So we can map the 6 cosets to 6 data values. The code has a bounded rewriting cost of 1.

Single permutation: **error correction**

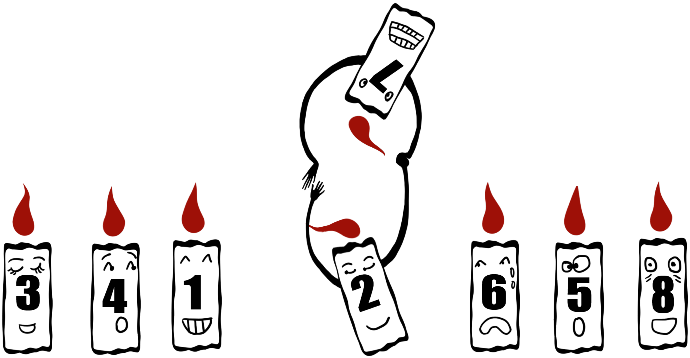


1. Model errors: Noise modeling, and error quantization.



2. Design ECC.

Kendall-τ distance



Definition (Kendall-*tau* distance)

The number of adjacent transpositions to change one permutation into another. (The distance is symmetric.)

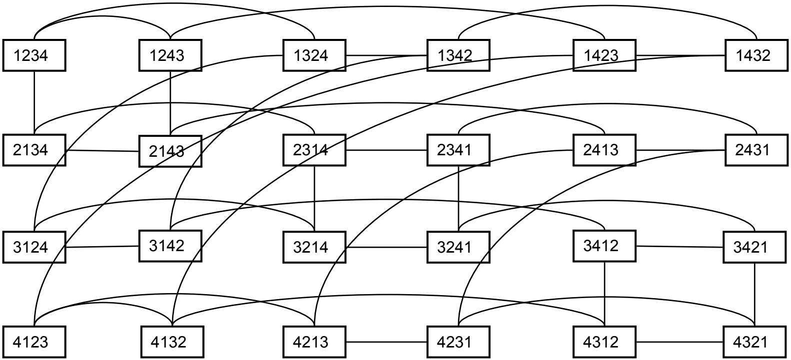
Example

For permutations $\alpha = [2, 1, 3, 4]$ and $\beta = [2, 3, 4, 1]$, the Kendall-τ distance $d_\tau(\alpha, \beta) = 2$ because
$[2, 1, 3, 4] \rightarrow [2, \mathbf{3}, \mathbf{1}, 4]] \rightarrow [2, 3, \mathbf{4}, \mathbf{1}]$.

We can define an adjacency graph for permutations based on Kendall-τ distance.

Example

Permutations S_n with $n = 4$.

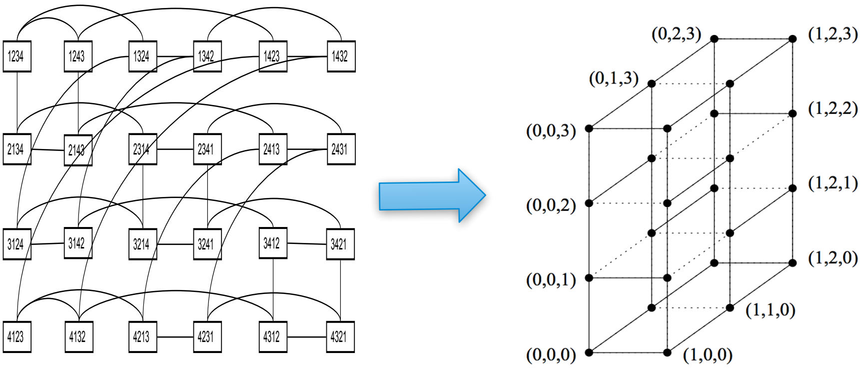


An technique for ECC construction: Embedding

Other techniques: Interleaving (product of sub-codes), modular (for limited-magnitude errors), etc.

Theorem

The adjacency graph for permutations is a subgraph of an $(n-1)$-dimensional array, whose size is $2 \times 3 \times \cdots \times n$.



A. Jiang, M. Schwartz and J. Bruck, Error-correcting codes for rank modulation, in ISIT 2009.

Construction (One-Error-Correcting Rank Modulation Code)

Let $C_1, C_2 \subseteq S_n$ denote two rank modulation codes constructed as follows. Let $A \in S_n$ be a general permutation whose inversion vector is $(x_1, x_2, \cdots, x_{n-1})$. Then A is a codeword in C_1 iff the following equation is satisfied:

$$\sum_{i=1}^{n-1} i x_i \equiv 0 \pmod{2n-1}$$

A is a codeword in C_2 iff the following equation is satisfied:

$$\sum_{i=1}^{n-2} i x_i + (n-1) \cdot (-x_{n-1}) \equiv 0 \pmod{2n-1}$$

Between C_1 and C_2, choose the code with more codewords as the final output.

For the above code, it can be proved that:

- The code can correct one Kendall error.
- The size of the code is at least $\frac{(n-1)!}{2}$.
- The size of the code is at least half of optimal.

Codes correcting more Kendall errors are constructed based on embedding.

First, consider codes of the following form:

- Let $m \geq n-1$ and let $h_1, \cdots, h_{n-1}$ be a set of integers, where $0 < h_i < m$ for $i = 1, \cdots, n-1$. Define the code as follows:

$$\mathcal{C} = \{(x_1, x_2, \cdots, x_{n-1}) \mid \sum_{i=1}^{n-1} h_i x_i \equiv 0 \mod m\}$$

[1] A. Barg and A. Mazumdar, "Codes in Permutations and Error Correction for Rank Modulation," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, pp. 854–858, June 2010.

Fact: The above code can correct t Kendall errors if all the syndromes caused by up to t errors are all distinct.

How to find such integers $h_1, \cdots, h_{n-1}$?

Theorem (Bose-Chowla)

Let q be a power of a prime, and let $m = \frac{q^{t+1}-1}{q-1}$. Then there exist $q+1$ integers $j_0 = 0, j_1, \cdots, j_q$ in $\mathbb{Z}_m$ such that the sums

$$j_{i_1} + j_{i_2} + \cdots + j_{i_t} \qquad (0 \le i_1 \le i_2 \le \cdots \le i_t \le q)$$

are all different modulo m.

The Bose-Chowla theorem is useful when all the errors in the embedded $(n-1)$-dimensional L_1 space are positive errors.

To also handle negative errors, we can "enlarge" the coefficients:

Theorem

For $1 \leq i \leq q+1$ *let*

$$h_i = \begin{cases} j_{i-1} + \frac{t-1}{2} m & \text{for odd } t \\ j_{i-1} + \frac{t}{2} m & \text{for even } t \end{cases}$$

where the numbers j_i *are given by the Bose-Chowla theorem. Let* $m_t = t(t+1)m$ *if* t *is odd and* $m_t = t(t+2)m$ *if* t *is even. For all* $e \in \mathbb{Z}^{q+1}$ *such that* $||e|| \leq t$ *the sums (i.e., syndromes)* $\sum_{i=1}^{q+1} e_i h_i$ *are all distinct and nonzero modulo* m_t*.*

More ideas (example): Map each dimension of the $(n-1)$-dimensional space to bits using Gray code. Then binary ECC can be turned into ECC for permutations.

Theorem

There is a code of length $n = q - 1$ and size at least $q^{\log_p(n-2t-1)}$. It corrects all patterns of up to t Kendall errors in the rank modulation scheme under a decoding algorithm of complexity polynomial in n.

Theorem

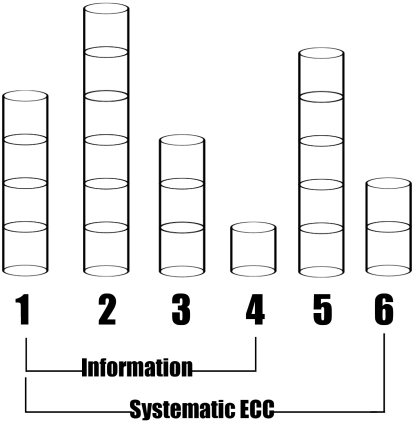
Let $\mathcal{A}$ be a binary code of length

$$m = (n+1)\lfloor \log n \rfloor - 2^{\lfloor \log n \rfloor + 1} + 2,$$

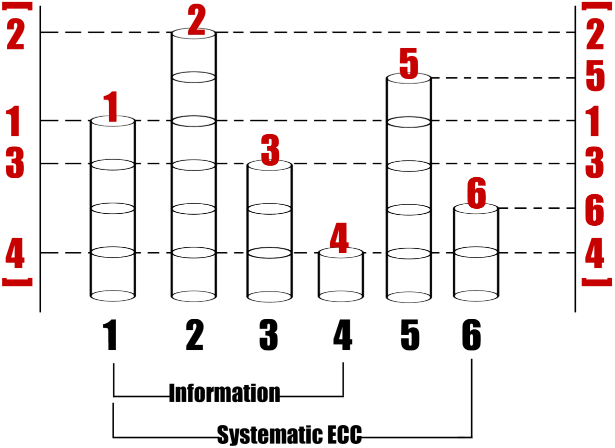
cardinality M and Hamming distance d. Then there is a rank modulation code on n elements of cardinality M with distance at least d in the Kendall space.

[1] A. Mazumdar, A. Barg and G. Zemor, "Constructions of Rank Modulation Codes," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, 2011.

Systematic ECC for Rank Modulation



Systematic ECC for Rank Modulation



Theorem (systematic one-error-correcting code)

For $k \geq 3$, there is a $(k+2, k)$ systematic code for correcting one Kendall-τ error. (And the code has an optimal size unless a perfect $(k+1, k)$ code exists.)

Theorem (systematic multiple-error-correcting code)

For any $2 \leq k < n$, there exists an (n, k) systematic code of minimum distance $n - k$.

Theorem (capacity of systematic ECC)

When $n \to \infty$, systematic ECCs achieve the same capacity as general ECCs.

H. Zhou, A. Jiang and J. Bruck, "Systematic error-correcting codes for rank modulation," in ISIT 2012.

Single permutation: capacity with Kendall-τ distance

Let the number of cells $n \to \infty$. Consider capacity.

Theorem (Capacity of Rank Modulation ECC with $n \to \infty$)

Let $A(n,d)$ be the maximum number of permutations in S_n with minimum Kendall-tau distance d. We call

$$C(d) = \lim_{n\to\infty} \frac{\ln A(n,d)}{\ln n!}$$

the capacity of rank modulation ECC of Kendall-tau distance d. Then,

$$C(d) = \begin{cases} 1 & \text{if } d = O(n) \\ 1-\epsilon & \text{if } d = \Theta(n^{1+\epsilon}),\ 0<\epsilon<1 \\ 0 & \text{if } d = \Theta(n^2) \end{cases}$$

[1] A. Barg and A. Mazumdar, "Codes in Permutations and Error Correction for Rank Modulation," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, pp. 854–858, June 2010.

Now, consider finite n.

Theorem (Ball-packing Bound)

For an ECC of M codewords for n cells and minimum Kendall-τ distance d,

$$M \leq \frac{n!}{|\mathcal{B}_{\lfloor (d-1)/2 \rfloor}|}.$$

The ball size can be computed:

$$|\mathcal{B}_r| = \sum_{i=0}^{r} |\mathcal{S}_i|$$

$$|\mathcal{S}_r| = I_n(r) = \binom{n+r-1}{r} + \sum_{j\geqslant 1}(-1)^j \times \left(\binom{n+r-u_j-1}{r-u_j} + \binom{n+r-u_j-j-1}{r-u_j-j} \right)$$

A. Jiang, M. Schwartz and J. Bruck, "Correcting charge-constrained errors in the rank modulation scheme," in *IEEE Trans. Information Theory*, May 2010.

Theorem (Gilbert-Varshamov-like bound)

Let n, M and d be positive integers such that

$$M \leq \frac{n!}{|\mathcal{B}_{d-1}|},$$

then there exists an (n, M, d) *ECC.*

Theorem (Singleton-like bound)

Let C be an (n, M, d) *ECC. Then,*

1. *Let t be the largest integer such that* $M > \frac{n!}{(n-t)!}$. *If* $0 \leq t \leq n-2$, *then* $d \leq \binom{n-t}{2}$.
2. *If* $M = \frac{n!}{(n-t)!}$ *for some integer* $2 \leq t \leq n-2$, *then* $d \leq \binom{n-t}{2} + 1$.

Let $P(n, d)$ be the largest value of M such that there exists an (n, M, d) ECC.
The following are recursive bounds.

Theorem (Monotonicity)

$$P(n+1, d) \geq P(n, d),$$
$$P(n, d) \geq P(n, d+1).$$

Theorem (Code shortening)

$$P(n+1, d) \leq (n+1) \cdot P(n, d).$$

Theorem (Code puncturing)

$$P(n+1, d+n) \leq \lceil \frac{n+1}{d+n} \rceil \cdot P(n, d).$$

Theorem (Code lengthening)

$$P(n+1, d) \geq \lceil \frac{n+1}{d} \rceil \cdot P(n, d).$$

Theorem (Code extending)

$$P(n+1, 2\delta) \geq \lceil \frac{n}{2\delta} \rceil \cdot P(n, 2\delta - 1).$$

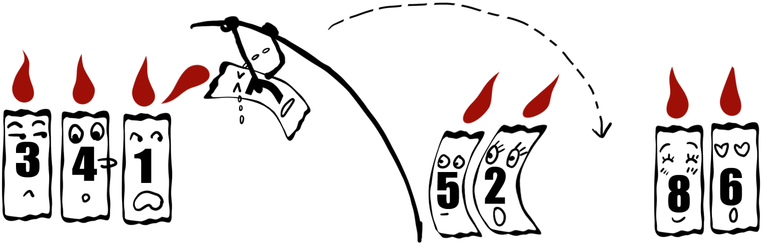
Furthermore, if there exists an $(n, 2\delta - 1)$ code of size $P(n, 2\delta - 1)$ with M_e even codewords (i.e., permutations that can be described as a product of an even number of transpositions) and M_o odd codewords, then

$$P(n+1, 2\delta) \geq \lceil \frac{n+1}{2\delta} \rceil M_e + \lceil \frac{n}{2\delta} \rceil M_o.$$

Theorem

$$P(n, 2\delta) \geq \frac{1}{2} P(n, 2\delta - 1).$$

Translocation distance (Ulam metric)



Definition (Translocation distance)

Given two permutations $\pi, \sigma \in \mathbf{S}_n$, their translocation distance $d_o(\pi, \sigma)$ is defined as the minimum number of translocations it takes to change π to σ (or from σ to π).

Definition (Ulam distance)

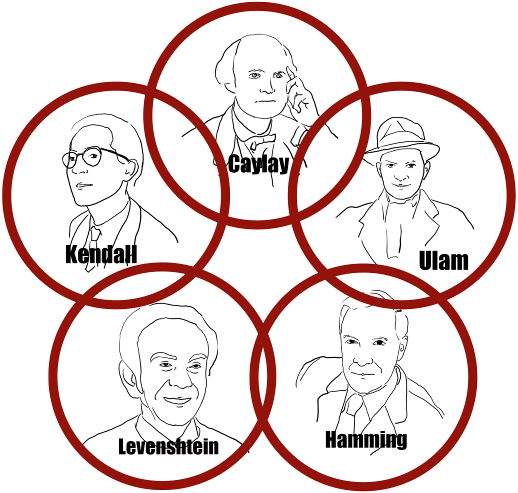
Given two permutations $\pi, \sigma \in \mathbf{S}_n$, let $L(\pi, \sigma)$ denote the length of their long common subsequence. Then, the Ulam distance is defined as $n - L(\pi, \sigma)$.

Theorem (Translocation distance equals Ulam distance)

$$d_o(\pi, \sigma) = n - L(\pi, \sigma).$$

Farzad Farnoud (Hassanzadeh), Vitaly Skachek, and Olgica Milenkovic, "Error-correction in flash memories via codes in the Ulam metric," in *IEEE Trans. Information Theory*, May 2013.

Different metrics are related:



Theorem (Levenshtein and Ulam)

Levenshtein's insertion/deletion distance is twice the Ulam distance.

Theorem (Ulam and Kendall)

Ulam distance is no greater than Kendall-τ distance, and is at least the Kendall-τ distance divided by $n - 1$ (where n is the length of the permutation).

Theorem (Ulam and Hamming)

Ulam distance is no greater than Hamming distance, and is at least the Hamming distance divided by n.

Theorem (Ulam and Caylay)

Ulam distance is at most twice the Caylay distance, and is at least the Caylay distance divided by $n - 1$.

Theorem (Caylay and Hamming)

Caylay distance is at most Hamming distance, and is at least half of Hamming distance.

Let $A_o(n,d)$ be the maximum size of a code of length n and minimum Ulam distance d.

Theorem (Maximum code size)

$$\frac{(n-d+1)!}{\binom{n}{d-1}} \leq A_o(n,d) \leq (n-d+1)!$$

Let $\mathcal{C}_o(d)$ denote the asymptotic capacity of codes with minimum Ulam distance $d = d(n)$, namely,

$$\mathcal{C}_o(d) = \lim_{n\to\infty} \frac{\ln A_o(n,d)}{\ln n!}.$$

Theorem (Capacity)

$$\mathcal{C}_o(d) = 1 - \lim_{n\to\infty} \frac{d(n)}{n}.$$

In comparison,

Theorem (Capacity of Hamming distance)

$$\mathcal{C}_H = 1 - \lim_{n\to\infty} \frac{d(n)}{n}.$$

Theorem (Capacity of Caylay distance)

$$1 - 2\lim_{n\to\infty} \frac{d(n)}{n} \le \mathcal{C}_T \le 1 - \lim_{n\to\infty} \frac{d(n)}{n}.$$

Theorem (Capacity of Kendall-τ distance)

$$C_K = 1 - \epsilon, \quad \textit{for } d = \Theta(n^{1+\epsilon}).$$

Ideas on code construction: Interleaving, and metric embedding

Theorem (Single right-translocation error-correcting code)

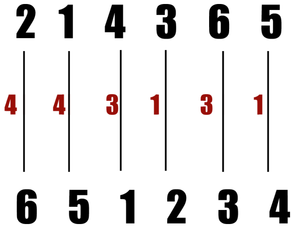
There is a single right-translocation error-correcting code of length n whose cardinality is $\frac{1}{4}(\lceil \frac{n}{2} \rceil !)^2$.

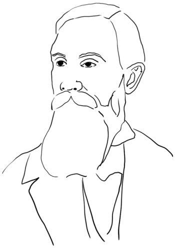
Theorem (Single translocation error-correcting code)

When n is a multiple of 3, there is a single translocation error-correcting code of length n whose cardinality is $\frac{1}{8}(\frac{n}{3}!)^3$.

Multiple translocation ECC: There exists a family of codes with Ulam distance $2t+1$, length $n = s(2t+1)$ for some integer $s \geq 4t+1$, and cardinality $M = (A_H(s, 4t+1))^{2t+1}$.

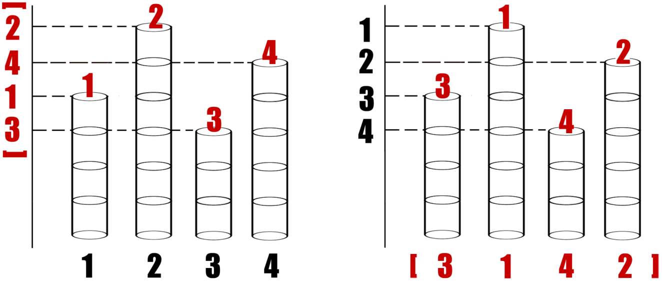
Infinity Norm





Chebyshev

Note: Here the permutation is the *ranks of n cells*, not the *cell indexes of n ranks.* (It is the right case below.)



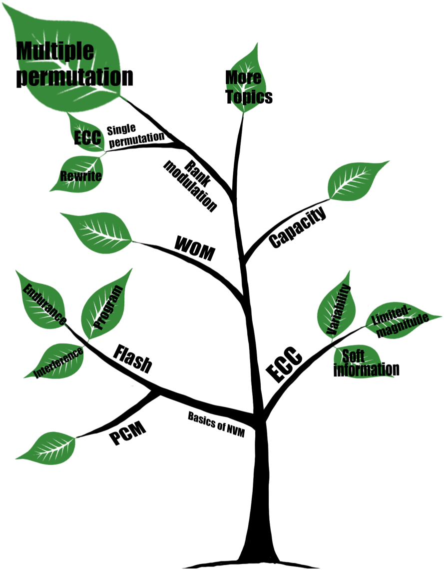
Theorem

For a code of length n and minimum infinity-norm distance d, its cardinality is at most

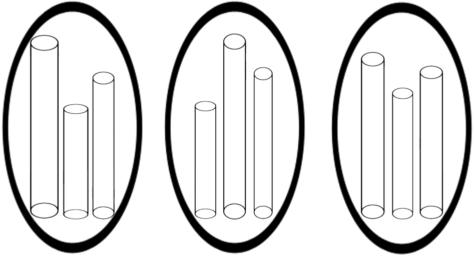
$$\frac{n!(\lfloor \frac{d+r}{2} \rfloor - r)!(\lceil \frac{d+r}{2} \rceil - r)!}{(d!)^{\lfloor \frac{n}{d} \rfloor - 1}(d-r)!\lfloor \frac{d+r}{2} \rfloor!\lceil \frac{d+r}{2} \rceil!}$$

where $r = n \mod d$.

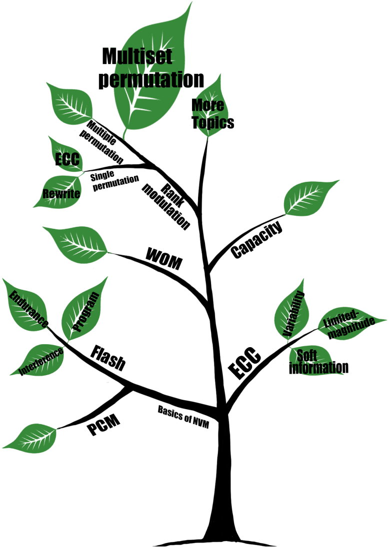
Itzhak Tamo and Moshe Schwartz, "Correcting limited-magnitude errors in the rank-modulation scheme," in *IEEE Trans. Information Theory*, June 2010.


Multiple permutation
More Topics
ECC
Single permutation
Rewrite
Rank modulation
Capacity
WOM
Endurance
Program
Variability
Limited-magnitude
Interference
Flash
ECC
Soft information
PCM
Basics of NVM

Coding with multiple permutations

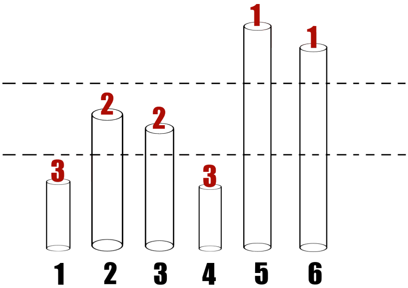


Fan Zhang, Henry Pfister, and A. Jiang, “LDPC codes for rank modulation in flash memories,” in ISIT 2010.
Reolyn Heymann, Jos H. Weber, Theo G. Swart, and Hendrik C. Ferreira, “Concatenated permutation block codes based on set partitioning for substitution and deletion error-control,” in ITW 2013.



Multiset permutation
More Topics
Multiple permutation
ECC
Single permutation
Rewrite
Rank modulation
Capacity
WOM
Endurance
Program
Variability
Limited-magnitude
Flash
Interference
ECC
Soft information
PCM
Basics of NVM

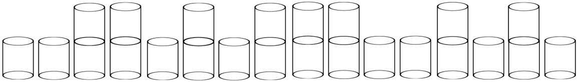
Coding with multiset permutation



For cells of the same rank,

1. Their analog levels' relative order does not matter.
2. In practice, we would like their analog levels to be close to each other, for better fault tolerance.

Rank modulation with multiset permutation: an intermediate form between single-permutation rank modulation and SLC/MLC.



[1] Hongchao Zhou, A. Jiang and J. Bruck, "Error-correcting schemes with dynamic thresholds in nonvolatile memories," in ISIT 2011.
[2] Eyal En Gad, A. Jiang and J. Bruck, "Trade-offs between instantaneous and total capacity in multi-cell flash memories," in ISIT 2012.
[3] Qing Li, "Compressed rank modulation," in Allerton 2012.

Multiset permutation: **error correction**

The study of ECC for rank modulation with different metrics can be extended from single permutation to multiset permutation.

[1] Frederic Sala, Ryan Gabrys, and Lara Dolecek, "Dynamic threshold schemes for multi-level non-volatile memories," in *IEEE Trans. Communications*, July 2013. (Decomposability distance, Kendall-τ distance)

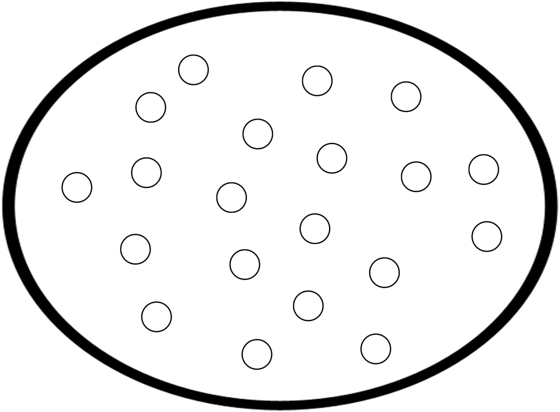
[2] Sarit Buzaglo, Eitan Yaakobi, Jehoshua Bruck, and Tuvi Etzion, "Error-correcting codes for multipermutations," in ISIT 2013. (Kendall-τ distance)

[3] Farzad Farnoud and Olgica Milenkovic, "Multipermutation codes in the Ulam Metric for nonvolatile memories," in JSAC special issue on Communication Methodologies for the Next-generation Storage Systems, May 2014. (Ulam distance, Hamming distance)

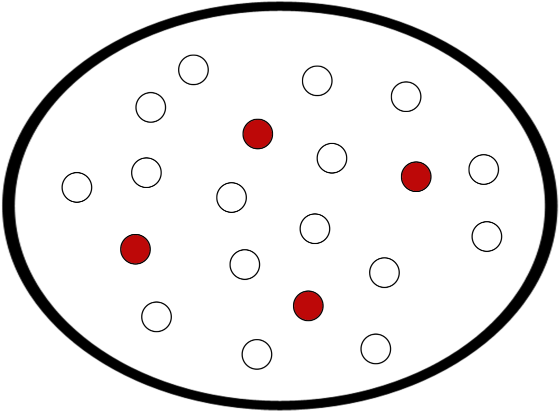
High-rate rewriting code based on **polar codes**.

Eyal En Gad, Eitan Yaakobi, Anxiao (Andrew) Jiang and Jehoshua Bruck, “Rank-modulation rewriting codes for flash memories,” in ISIT 2013.

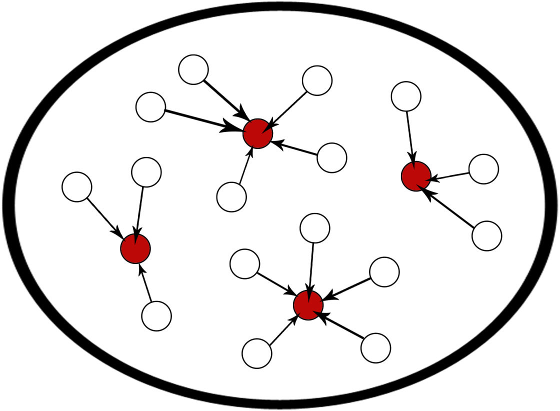
States of cells (for both SLC/MLC and rank modulation):



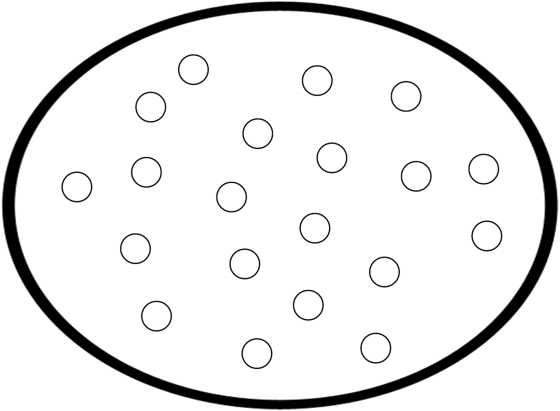
ECC codewords:



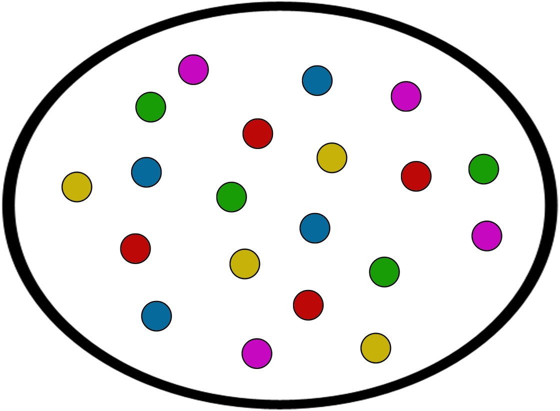
ECC decoding:



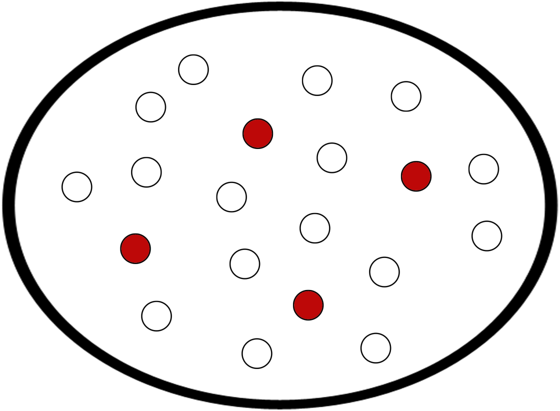
Now consider rewriting.
States of cells:



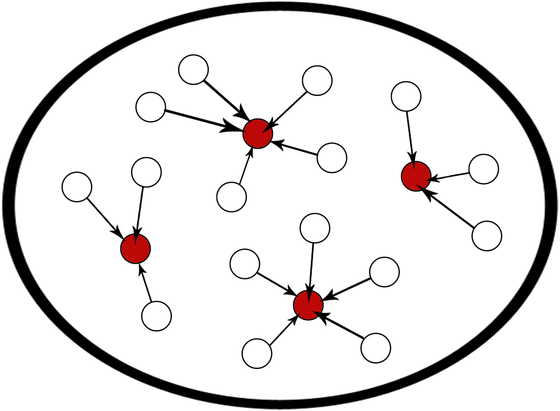
Each color represents a data value:



Say “red” is the data value we want to (re)write:



Rewriting:

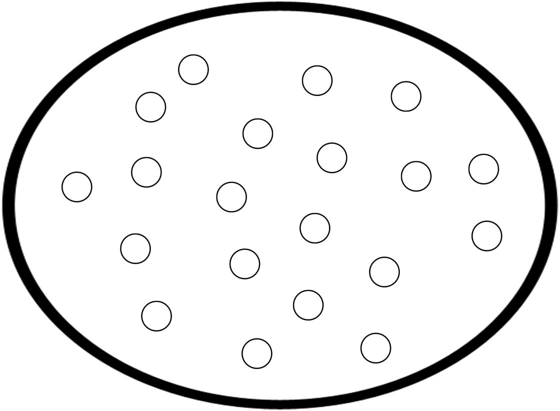


Duality between error correction and rewriting

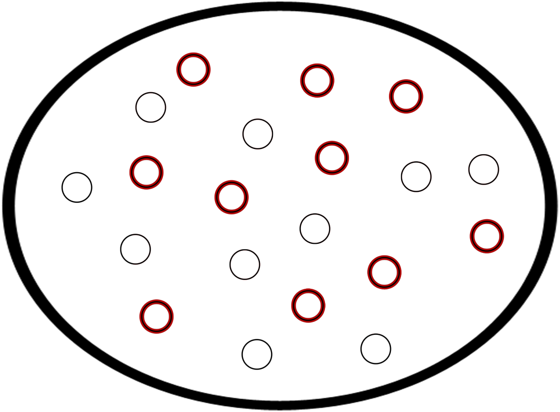
- Decoding of ECC $\Leftrightarrow$ Encoding of rewriting code.
- Encoding of ECC $\Leftrightarrow$ Decoding of rewriting code.
- Decoding of ECC is hard $\Leftrightarrow$ Encoding of rewriting is hard.
- Random coding can achieve capacity for ECC $\Leftrightarrow$ Random coding can achieve capacity for rewriting.
- Encoding of rewriting should be EXACT (i.e., no error) $\Rightarrow$ block decoding of ECC (not allowing any bit error after decoding)
- Additional constraint for WOM/Flash: Cell levels can only go up for rewriting.
- Additional constraint for rewriting code in general: There are multiple rewrites, so the code has multiple stages.

Now consider: Combining error correction and rewriting.

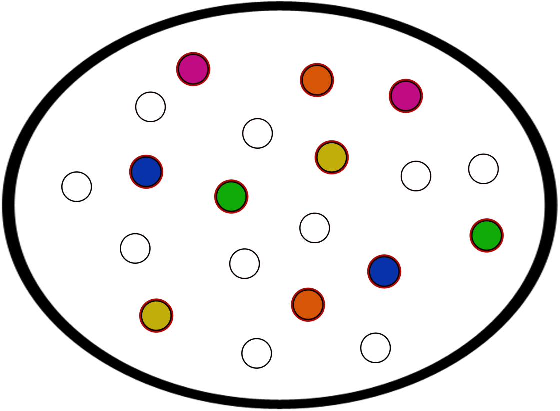
States of cells:



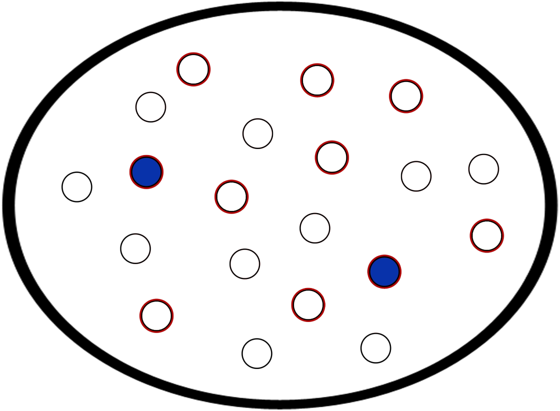
ECC codewords:



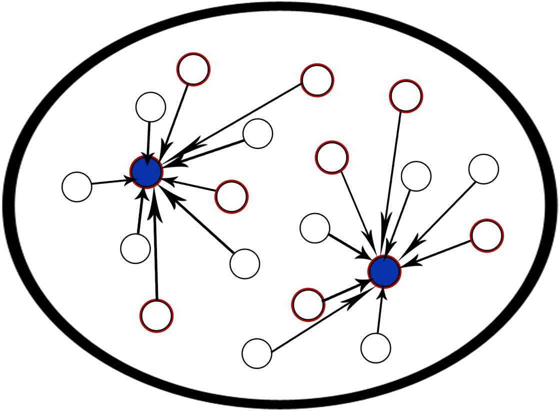
Each color represents a data value:



Say “blue” is the data value we want to (re)write:



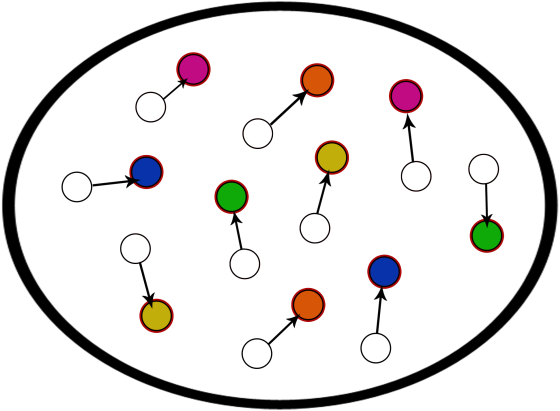
Rewriting (for error-correcting rewriting code):



For error-correcting rewriting code:

- Encoding (rewriting) $\Leftrightarrow$ ECC decoding.

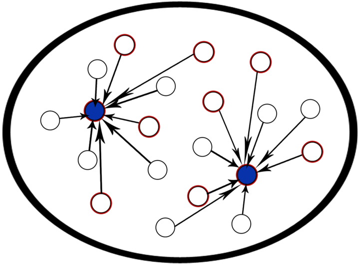
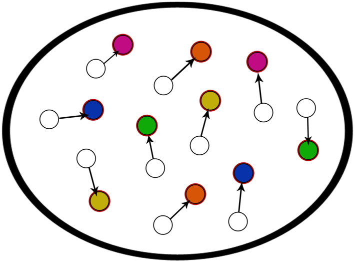
ECC decoding (for error-correcting rewriting code):



For error-correcting rewriting code:

- Encoding (rewriting) ⇔ ECC decoding.
- Decoding ⇔ ECC decoding.
- Both encoding and decoding are hard

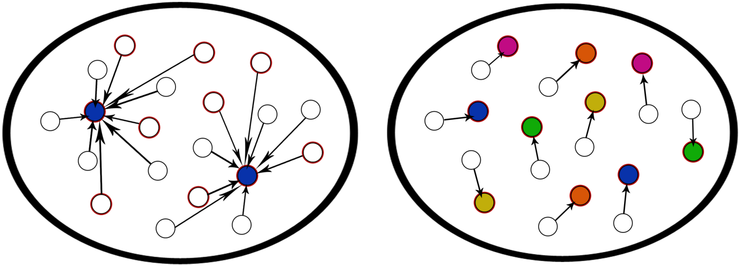
Error correction and rewriting:

Is there an ECC with these properties:

- Decoding is efficient (for random channels, asymptotically).
- Focus on block decoding (that is, decoding the whole codeword correctly).

Error correction and rewriting:



Is there an ECC with these properties:

- Decoding is efficient (for random channels, asymptotically).
- Focus on block decoding (that is, decoding the whole codeword correctly).

POLAR CODE.

Polar code:

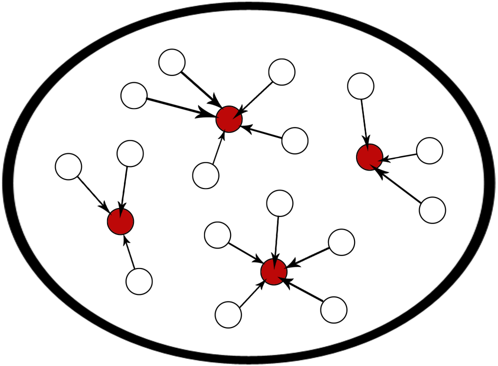
- Decoding (and encoding) complexity:

$$O(N \log N).$$

- Additional feature (important for constructing rewriting codes): The separation between *frozen bits* and *non-frozen bits*.

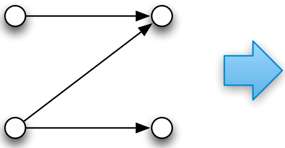
Erdal Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," in ISIT 2008.

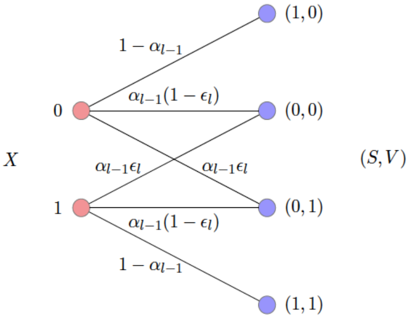
Duality between channel coding (ECC) and source coding (data compression).



S. B. Korada and R. Urbanke, "Polar codes are optimal for lossy source coding," in *IEEE Trans. Information Theory*, April 2010.

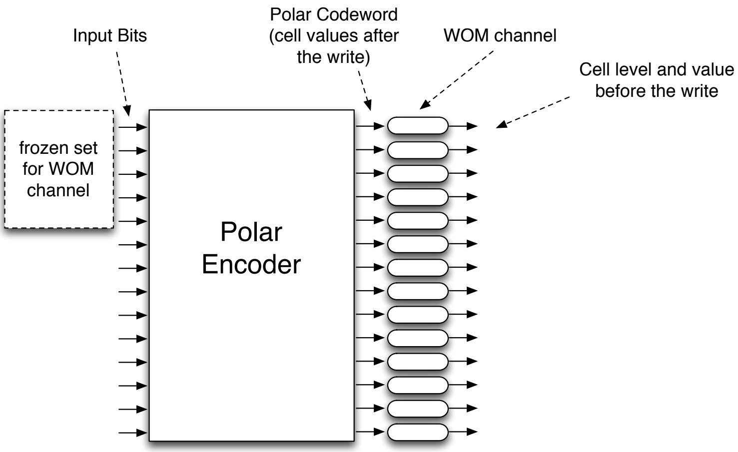
Polar code is used to construct capacity-achieving WOM codes.

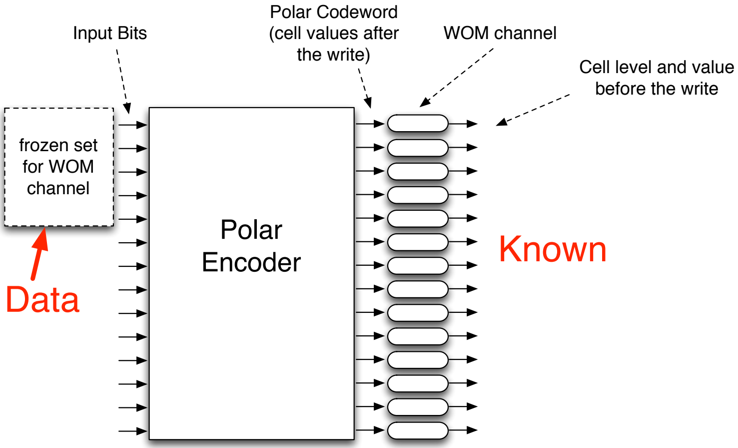


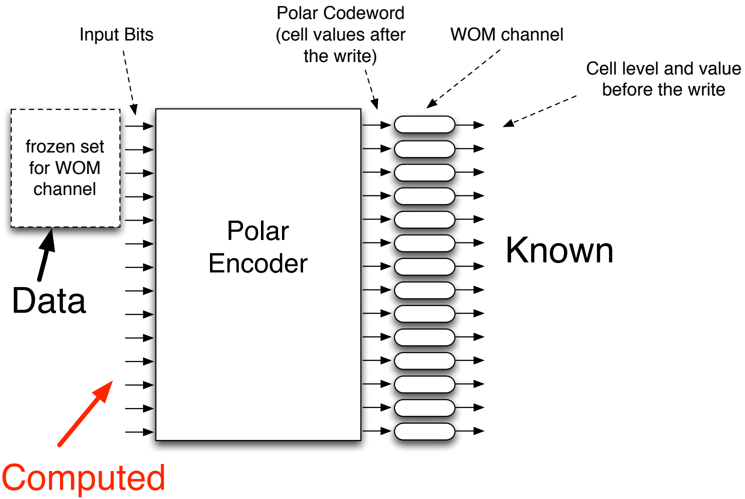


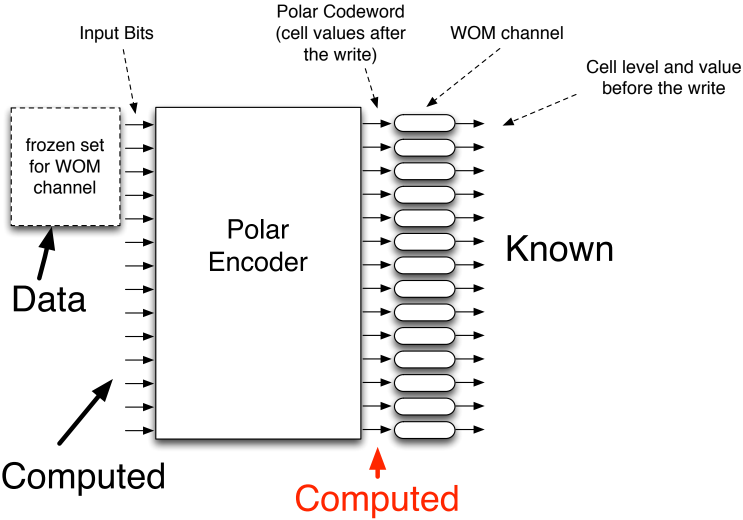
David Burshtein and Alona Strugatski, "Polar write once memory codes," in ISIT 2012.

Polar WOM code's process of a rewrite: **Encode**

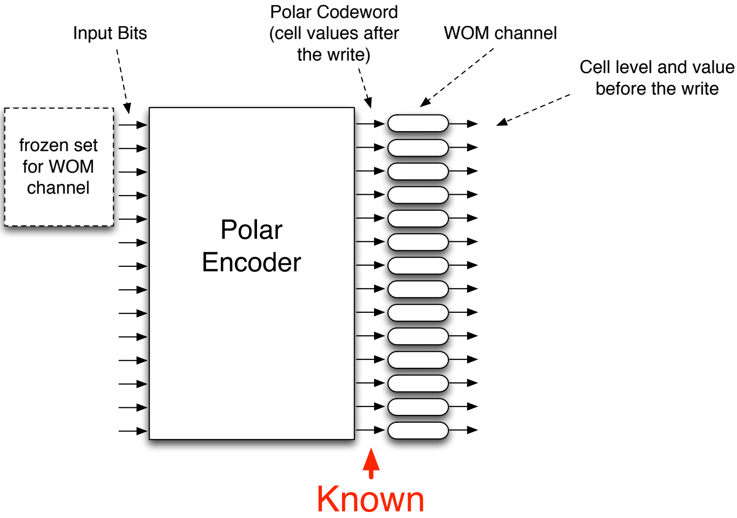


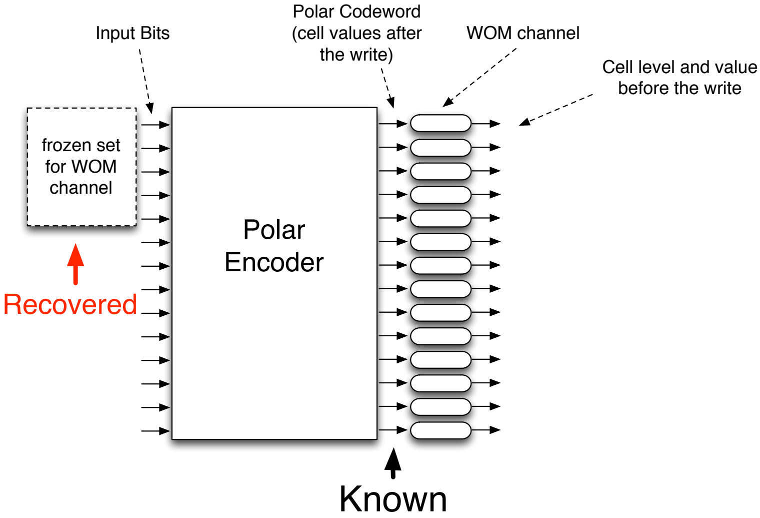

Input Bits
Polar Codeword (cell values after the write)
WOM channel
Cell level and value before the write
frozen set for WOM channel
Polar Encoder
Known
Data


Input Bits
Polar Codeword (cell values after the write)
WOM channel
Cell level and value before the write
frozen set for WOM channel
Polar Encoder
Known
Data
Computed


Input Bits
Polar Codeword (cell values after the write)
WOM channel
Cell level and value before the write
frozen set for WOM channel
Polar Encoder
Known
Data
Computed
Computed

Polar WOM code's process of a rewrite: **Decode**



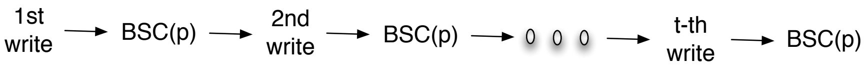

Input Bits
Polar Codeword
(cell values after
the write)
WOM channel
Cell level and value
before the write
frozen set
for WOM
channel
Polar
Encoder
Recovered
Known

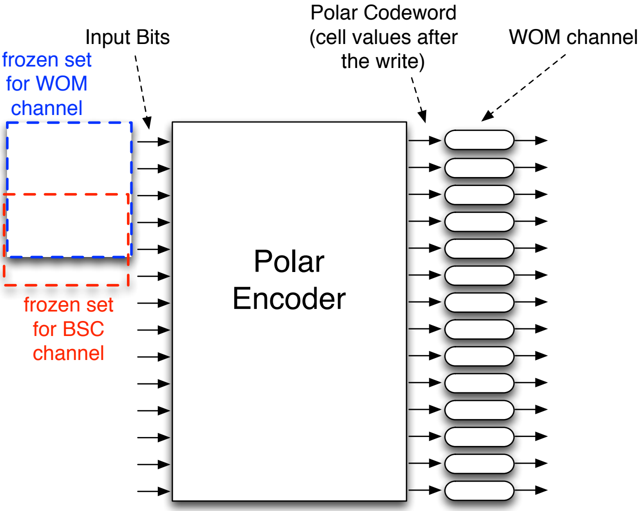
For **Rewriting** to be used in flash memories, it is **CRITICAL** to combine it with **Error-Correcting Codes**.

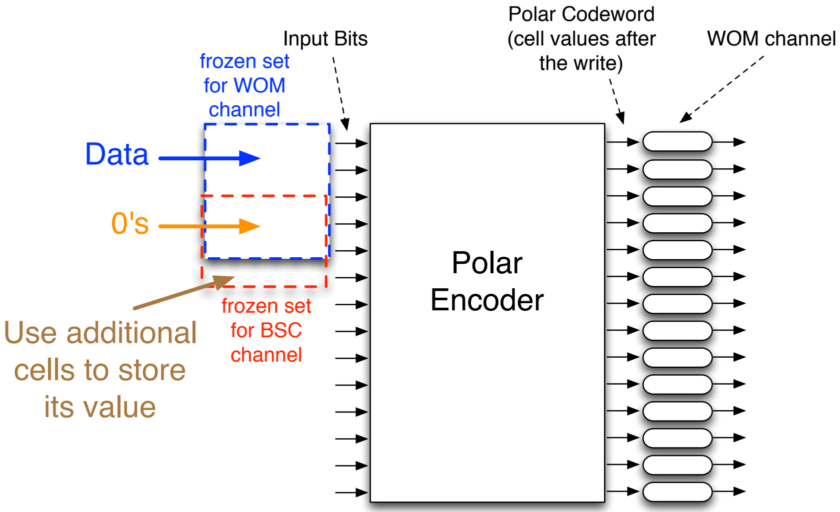
A joint coding scheme for rewriting and error correction, which can correct a substantial number of errors and supports any number of rewrites.

A. Jiang, Yue Li, Eyal En Gad, Michael Langberg, and Jehoshua Bruck, "Joint rewriting and error correction in write-once memories," in ISIT 2013.

Model of rewriting and noise:




Input Bits
frozen set for WOM channel
Polar Codeword (cell values after the write)
WOM channel
Polar Encoder
frozen set for BSC channel


Input Bits
Polar Codeword (cell values after the write)
WOM channel
frozen set for WOM channel
Data
0's
Use additional cells to store its value
frozen set for BSC channel
Polar Encoder

Lower bound to achievable sum-rate (for WOM):

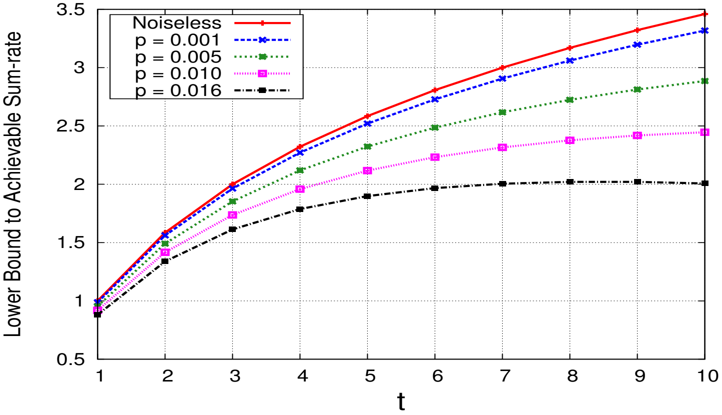


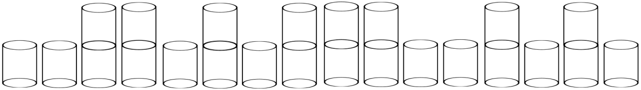
Fig. 6. Lower bound to achievable sum-rates for different error probability p.

Also thanks to polar code:

- Rewriting code for **rank modulation with multiset permutation**.

E. En Gad, E. Yaakobi, A.Jiang and J. Bruck, "Rank-modulation rewriting codes for flash memories," in ISIT 2013.

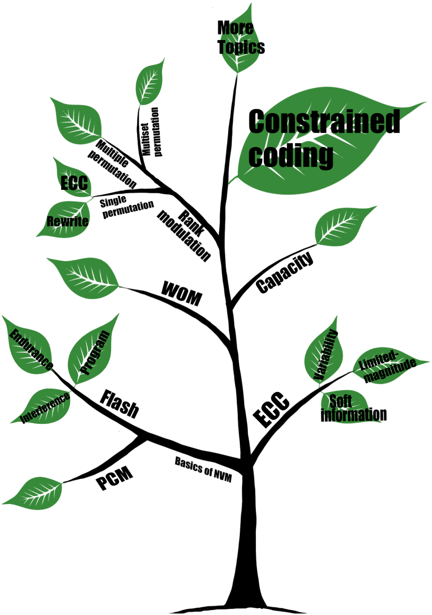
Can we design error-correcting rewriting codes for rank modulation with multiset permutation?



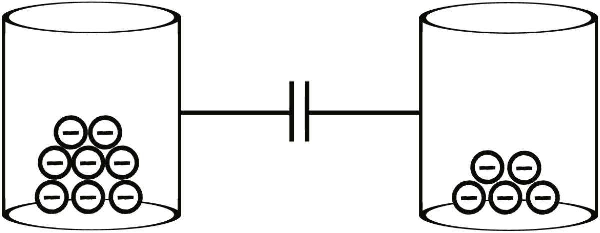
Further reading

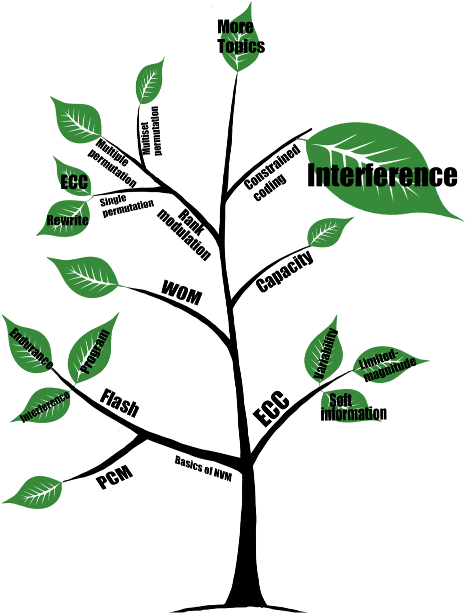
- D. Slepian, "Permutation modulation," in *Proc. IEEE*, Mar. 1965.
- I. F. Blake, G. Cohen, and M. Deza, "Coding with permutations," *Inf. Control*, 1979.
- A. J. Han Vinck and H. C. Ferreira, "Permutation trellis-codes," in ISIT 2001.
- C. J. Colbourn, T. Klove, and A. C. H. Ling, "Permutation arrays for powerline communication and mutually orthogonal Latin squares," in *IEEE Trans. Information Theory*, June 2004.
- I. Tamo and M. Schwartz, "On the labeling problem of permutation group codes under the infinity metric," in *IEEE Trans. Information Theory*, Oct. 2012.
- M. Schwartz and I. Tamo, "Optimal permutation anticodes with the infinity norm via permanents of $(0, 1)$-matrices," in *Journal of Combinatorial Theory, 2011.*

- T. Klove, T. Lin, S. Tsai and W. Tzeng, "Permutation arrays under the Chebyshev distance," in *IEEE Trans. Information Theory*, June 2010.
- Y. Yehezkeally and M. Schwartz, "Snake-in-the-box codes for rank modulation," in *IEEE Trans. Information Theory*, Aug. 2012.
- T. Wadayama and M. Hagiwara, "LP-decodable permutation codes based on linearly constrained permutation matrices," in *IEEE Trans. Information Theory*, Aug. 2012.
- M. Schwartz, "Quasi-cross lattice tilings with applications to flash memory," in *IEEE Trans. Information Theory*, Apr. 2012.
- M. Qin, A. Jiang and P. H. Siegel, "Parallel programming of rank modulation," in ISIT 2013.


More Topics
Constrained coding
Multiset permutation
Multiple permutation
ECC
Rewrite
Single permutation
Rank modulation
Capacity
WOM
Endurance
Program
Interference
Flash
Variability
Limited-magnitude
Soft information
ECC
PCM
Basics of NVM

Constrained Coding

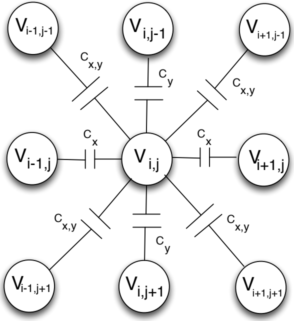



More Topics
Multiset permutation
Multiple permutation
ECC
Single permutation
Rewrite
Rank modulation
Constrained coding
Interference
Capacity
WOM
Endurance
Program
Interference
Flash
Variability
Limited-magnitude
Soft information
ECC
PCM
Basics of NVM

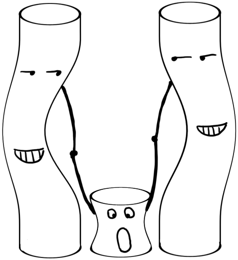
Constrained coding for inter-cell interference

Inter-cell interference in flash memory:
The V_{th} shift of middle cell caused by shifting of neighboring cells is

$$\begin{aligned}\Delta V_{i,j} = \; & C_x(\Delta V_{i-1,j} + \Delta V_{i+1,j}) + C_y(\Delta V_{i,j-1} + \Delta V_{i,j+1}) \\ & + C_{x,y}(\Delta V_{i-1,j-1} + \Delta V_{i+1,j-1} + \Delta V_{i-1,j+1} + \Delta V_{i+1,j+1})\end{aligned}$$



One constraint to set for q-level cells: The difference between adjacent levels cannot be too large.
A concrete example: Avoid $(q-1)0(q-1)$ pattern for adjacent cell levels.



Minghai Qin, Eitan Yaakobi, and Paul Siegel, "Constrained codes that mitigate intercell interference in read/write cycles for flash memories," in JSAC Special Issue, May 2014.

Constrained rank modulation: The difference between adjacent cell ranks cannot be too large.

A more specific constraint:

Definition (Single neighbor k-constraint)

The difference between two adjacent cells' ranks cannot be more than k.

Frederic Sala and Lara Dolecek, "Constrained rank modulation schemes," in ITW 2013.

Definition (path-scheme)

A graph with n nodes labelled by $1, 2, \cdots, n$. Two nodes i and j have an edge iff $|i - j| \leq k$.

k-constrained permutation $\Leftrightarrow$ Hamiltonian path in path-scheme

Let $|A_{n,k}|$ denote the total number of k-constrained permutations of length n.

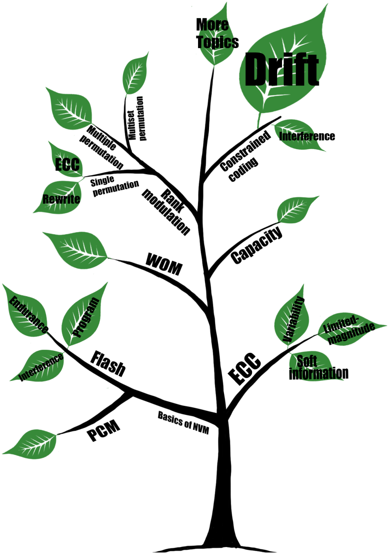
Define capacity as $\mathcal{C}(k) = \lim_{n\to\infty} \frac{\ln |A_{n,k}|}{\ln n!}$.

Theorem (Bounds for code size)

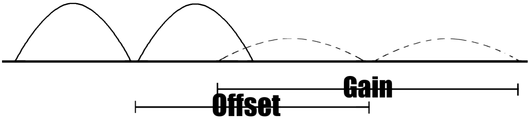
$$(k+1)!\lfloor\frac{k}{2}\rfloor^{n-k-1} \leq |A_{n,k}| \leq 2(2k)^n - (k!)^{\lfloor\frac{n}{k}\rfloor}$$

Theorem (Capacity)

If $k = \Theta(n^{\epsilon})$, where $0 \leq \epsilon \leq 1$, the capacity of k-constrained rank modulation codes is $\mathcal{C}(k) = \epsilon$.


More Topics
Drift
Interference
Constrained coding
Multiset permutation
Multiple permutation
ECC
Single permutation
Rewrite
Rank modulation
Capacity
WOM
Endurance
Program
Interference
Flash
Variability
Limited-magnitude
Soft information
ECC
PCM
Basics of NVM

Flash memory and PCM commonly have cell level drifting.



Definition (Model for Cell Level Drifting)

Let $\mathbf{x} = (x_1, x_2, \cdots, x_n) \in \{0, 1, \cdots, q\}^n$ be the original codeword. Let $\mathbf{v} = (v_1, v_2, \cdots, v_n)$ be additive noise. Let $\mathbf{r} = (r_1, r_2, \cdots, r_n)$ be received codeword. Then,

$$\mathbf{r} = a(\mathbf{x} + \mathbf{v}) + b$$

where $a > 0$ is the scaling factor (gain), and b is the offset.

In practice, a and b are often *unknown* and *time-variant*.

Kees A. Schouhamer Immink, "Coding schemes for multi-level channels with unknown gain and/or offset," in ISIT 2013.

Definition (Pearson Distance)

$$\rho_{\mathbf{r},\hat{\mathbf{x}}} = \frac{\sum_{i=1}^{n}(r_i - \bar{r})(\hat{x}_i - \bar{\hat{x}})}{\sigma_r \sigma_{\hat{x}}}$$

Minimize Pearson distance to find codeword. The estimation is robust to gain and offset.

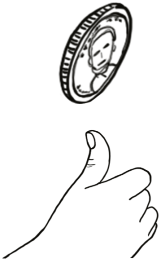
Use T-constrained codes for the Pearson-distance based decoding.

Kees A. Schouhamer Immink and Jos H. Weber, "Minimum Pearson distance detection for multi-level channels with gain and/or offset mismatch," draft 2014.

Further reading

- A. Berman and Y. Birk, "Error correction scheme for constrained inter-cell coupling in flash memory," NVMW 2011.
- G. Dong, S. Li and T. Zhang, "Using data post-compensation and pre-distortion to tolerate cell-to-cell interference in MLC NAND flash memory," in *IEEE Trans. Circuits and Systems I*, 2010.
- E. Ordentlich, G. Ribeiro, R. M. Roth, G. Seroussi, and P. O. Vontobel, "Coding for limiting current in memristor crossbar memories," NVMW 2011.
- Y. Cassuto, S. Kvatinsky and E. Yaakobi, "Sneak-path constraints in memristor crossbar arrays," in ISIT 2013.

Little summary and more topics

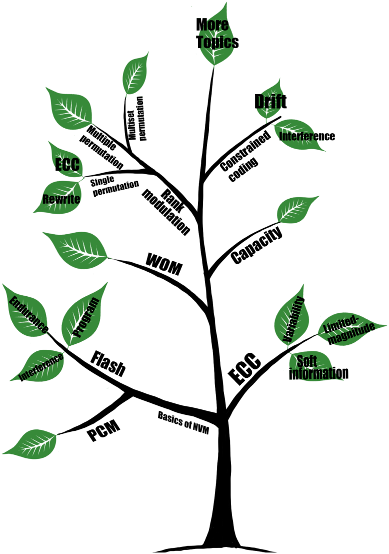


A little summary:

Nonvolatile memories are good,
nonvolatile memories are strange.
We need the right metrics,
to characterize noise and constraints.
The metrics are often related,
so are coding techniques.
If all goes well hand in hand,
We get reliability and speed.

Some further topics:

- Coding combined with NVM file systems
- Cryptography and security for NVM
- Compression in NVM
- Information-theoretic memory cell design and architecture design
- Coding for networked NVMs and NVM arrays


More Topics
Drift
Interference
Constrained coding
Multiset permutation
Multiple permutation
ECC
Rewrite
Single permutation
Rank modulation
Capacity
WOM
Endurance
Program
Interference
Flash
Variability
Limited-magnitude
Soft information
ECC
PCM
Basics of NVM

Acknowledgments

- ISIT tutorial co-chairs Narayana (Prasad) Santhanam and Li Ping
- Colleagues with contributions to the area

Acknowledgments

Jehoshua Bruck

Acknowledgments

Jehoshua Bruck

Wenyi Zhu (Artist)

Acknowledgments

Jehoshua Bruck

Wenyi Zhu
(Artist)

Yao Li

Sean Huang

Ryan Gabrys

Nicolas Bitouze

Acknowledgments

Resources

- IEEE JSAC Special Issue on Data Storage, May 2014.
- Annual Flash Memory Summit, San Jose, August 2014.
  - Student Travel Grants Available for Student Presenters and Workshop Assistants!
- Annual Non-Volatile Memories Workshop, UCSD, March 2014.

Resources

- IEEE JSAC Special Issue on Data Storage, May 2014.
- Annual Flash Memory Summit, San Jose, August 2014.
  - Student Travel Grants Available for Student Presenters and Workshop Assistants!
- Annual Non-Volatile Memories Workshop, UCSD, March 2014.