Lattice-Based WOM Codebooks that Allow Two Writes

Brian M. Kurkoski
Japan Advanced Institute of Science and Technology

October 29, 2012
2012 International Symposium on Information Theory and its Applications
Honolulu, Hawaii, USA
Flash Memories

Flash memories store charge on transistors called “cells”
Increasing the number of bits/cell increases data storage density:

- **SLC**
  - 1 bit/cell
  - \( q = 2 \)

- **MLC**
  - 2 bit/cell
  - \( q = 4 \)

- **TLC**
  - 3 bit/cell
  - \( q = 8 \)

Flash Memories Wear Out

- To re-write a memory, must first erase it
- Each time the flash memory is erased, the error rate increases

![Graph showing bit error rate vs. program/erase cycles](image)

Grupp, et al.
WOM Codes for Non-Binary Flash

Codes for “Write Once Memories”

- Pioneered by Rivest and Shamir [1982]
- Memory can change from 0→1 state but not 1→0
  - non-binary case: 0 → 1 → 2 → ... → q – 1
- Remarkable! Possible to re-use a “write once” memory!
- Application: Increase flash write endurance

WOM Codes Rates

- WOM codes allowing \( t = 2 \) writes, but \( q \)-levels
- \( n \) is the number of cells
- Code rate for write \( i \) is \( R_i \), normalized rate \( \tilde{R}_i \)
- Questions. How to:
  - maximize \( R_1 + R_2 \)?
  - maximize \( R_1 + R_2 \) subject to \( R_1 = R_2 \)?

Conventional Rate Definition:

\[
R = \frac{\log_2 \# \text{ of messages}}{n} \text{ bits/cell}
\]

Normalized rate:

\[
\tilde{R} = \frac{1}{\log_2 q} \frac{\log_2 \# \text{ of messages}}{n}
\]
Capacity of Non-Binary WOM Codes

Fu and Han Vink [1999] gave capacity of a $t$-write code into $q$-ary cells.

For $t=2$:

$$R_1 + R_2 \leq \log_2 \left( \frac{q + 1}{q - 1} \right)$$

But, capacity-achieving rates are not equal in general, $R_1 \neq R_2$

Equal rates $R_1 = R_2$ is of practical concern.

Gabrys and Dolecek [2011] found an upper bound on equal-rate capacity:

$$2R_1 = 2R_2 \leq \frac{2}{3} \log \left( \frac{q(q + 1)(2q + 1)}{6} \right)$$
The normalized sum rate is a measure of **efficiency**

- Efficiency increases as \( q \) increases
In This Talk: WOM Properties of Lattices

Lattices have an inherent error-correction property
What about the WOM properties of lattices?

Outline of this talk:

- Sphere packings and lattices
- Lattice Codes = intersection of a **shaping region** and a lattice
- WOM properties of lattices
  - Using **continuous approximation**, code rate is from the **volume of the shaping region**
  - The ideal shaping region is **hyperbolic**
  - Give an expression for normalized sum rate under equal rate assumption,
    \[
    \tilde{R} \geq 1 - \frac{1}{e \ln 2} \frac{1}{\log_2(q - 1)}
    \]
    close to Gabrys-Dolecek upper bound
A **Sphere Packing** is an arrangement of non-overlapping spheres in space.
A Lattice Is A Linear Sphere Packing

A lattice is a linear subgroup of $\mathbb{R}^n$

$G$: $n$-by-$n$ generator matrix

$x = G \cdot b$

$b = (b_1, b_2, \ldots, b_n)^t$: $n$-by-1 vector of integers

$x = (x_1, x_2, \ldots, x_n)^t$: $n$-by-1 vector, lattice point

Lattices:
- Have a rich theory
- Can correct errors, achieve AWGN capacity

What about the WOM properties of lattices?

$Lattice$ (linear)

$$G = \begin{bmatrix} 5 & 0 \\ 2.15 & 4.3 \end{bmatrix}$$

Hexagonal Lattice
16 codewords, $d_{\text{min}} = 4.29$
Lattice Code Construction

- Lattice $\Lambda$ is infinite code over reals
- “minimum distance”

\[
\text{q-1}
\]

Shaping region $\mathcal{R}$ finite

Codebook $\mathcal{C} = \Lambda \cap \mathcal{R}$ is finite

WOM Lattice Code Construction

Construct a code using two regions $\mathcal{R}_1$ and $\mathcal{R}_2$

- Codebook for region 1, $\mathcal{C}_1 = \mathcal{R}_1 \cap \mathcal{C}$
- Codebook for region 2, $\mathcal{C}_2 = \mathcal{R}_2 \cap \mathcal{C}$
- Separated by boundary $B$
Lattice Code Construction

Lattice $\Lambda$ is infinite code over reals “minimum distance”

$\cap$

$q - 1$

$0$

$\mathcal{R}$ finite

Boundary $B$

Construct a code using two regions $\mathcal{R}_1$ and $\mathcal{R}_2$

- Codebook for region 1, $\mathcal{C}_1 = \mathcal{R}_1 \cap \mathcal{C}$
- Codebook for region 2, $\mathcal{C}_2 = \mathcal{R}_2 \cap \mathcal{C}$
- Separated by boundary $B$
n-Dimensional Lattice in n Flash Cells

$n = 2$

$n = 3$

$n = 4, 5, 6, ...$

Warning
image not available

2 flash cells

3 flash cells

4, 5, 6, ... flash cells
Code Rates using Continuous Approximation

number of points in $\mathcal{R}$ \cdot Volume of Voronoi region $\approx$ Volume of $\mathcal{R}$

$|\mathcal{C}| \cdot V(\Lambda) \approx V$

region $\mathcal{R}_i$, with volume $V_i$

Normalized Rate

$$\tilde{R}_i = \frac{\log_2 |\mathcal{C}_i|}{\log_2 |\mathcal{C}|}$$

$$\tilde{R}_i \approx 1 - \frac{\log_2 V_i}{\log_2 V(\Lambda)}$$

Code rates $R_i$ expressed as volume $V_i$
Approximation Improves as $q \to \text{large}$

Continuous approximation was used by Forney for AWGN channels
- well-known shaping gain of 1.53 dB
Cell values increase
= rectangular “accessible region”

Recall that cell values can only increase
A path from:
▷ initial state
▷ terminal state
Cell values increase
= rectangular region “accessible points”

Recall that cell values can only increase.
A path from:
- initial state
- terminal state

Consider a **current state**
The “accessible points” are in a rectangular region.
Maximizing the Rate: B is a Hyperbola

$V_2(x)$: volume of space from $x$

**Hypothesis** For any $x \in B$, selecting $V_2(x)$ equal to a constant $V_2$ will maximize the rate.

For any point on $B$, the volume $V_2$ should be constant:

$$V_2 = (1 - x_1)(1 - x_2)$$

and in $n$ dimensions:

$$V_2 = \prod_{i=1}^{n}(1 - x_i)$$

So, $B$ is a hyperbola. We have a hyperbolic shaping region.
Maximizing the Rate: 
B is a Hyperbola

$V_2(x)$: volume of space from $x$

**Hypothesis** For any $x \in B$, selecting $V_2(x)$ equal to a constant $V_2$ will maximize the rate.

For any point on $B$, the volume $V_2$ should be constant:

$$V_2 = (1 - x_1)(1 - x_2)$$

and in $n$ dimensions:

$$V_2 = \prod_{i=1}^{n}(1 - x_i)$$

So, $B$ is a hyperbola. We have a hyperbolic shaping region.
We can calculate the volume (and thus the rate).

For $n = 2$:

$$V_1 = 1 - (1 - \alpha) + (1 - \alpha) \log(1 - \alpha)$$

For arbitrary $n$:

$$V_1 = 1 - e^{-z} \sum_{m=0}^{n-1} \frac{z^m}{m!},$$

where $z = -\log(1 - \alpha)$

Assume equal rates for first and second writes:

$$V_1 = V_2$$

$$1 - e^{-z} \sum_{m=0}^{n} \frac{z^m}{m!} = e^z - 1$$

The solution $z^*$ can only be found numerically. But can form an upper bound:

$$\tilde{R} \geq 1 - \frac{1}{e \ln 2} \frac{1}{\log_2(q - 1)}$$

for the cubic lattice
Hyperbolic shaping approaches capacity as $n \to \infty$

**Conditions:**
- $t=2$ writes
- Capacity: non-equal rates (equal rates cannot achieve capacity)
- Hyperbolic lower bound: equal rates

lower bound approaches capacity as $q \to \infty$

If lower bound is tight, then gap to capacity for equal rates is small
What about encoding?

Bhatia, Iyengar and Siegel considered $n = 2$ [ITW 2012]

$\bullet$ $n = 2$ Easy to label all points $\rightarrow$ encode at the promised rate

Problem:

$\bullet$ for $n > 2$ some points are not “consistent” — there may be a rate penalty

$\bullet$ Hyperbolic shaping bound may not be tight (still unknown!)

Example: 2-write 8-level WOM Code

image thanks: Paul Siegel
Bhatia et al., Constructions with $n=2$
A Large Gap Remains!

![Graph showing capacity and bounds for $n=2$]
Summary: WOM Properties of Lattices

WOM-properties of lattices

- Used a “continuous approximation”
  - Convert a discrete problem to a continuous problem
- Shaping region is a hyperbola in $n$ dimensions
- Compute a lower bound on the code rate
- Much work to do on achievability of coding schemes