Interference Alignment and applications to storage

Instructors: Viveck R. Cadambe, Muriel Médard

viveck@mit.edu
medard@mit.edu
Today

• Revision and continuation (achieving K/2 DoF for the K user interference channel)

• Interference alignment for Distributed Storage
  – A multi-source (non-multicast) network coding application
$K = 3$ user interference channel example with $n = 1$ so that $M = 3$

$T$ is diagonal now.

Solution

$w$ is a random vector.

$V_1 = V_3 = V_{21} = w$

$V_{22} = Tw$
$K = 3$ user interference channel, achieving $3/2$ DoF
$K = 3$ user interference channel, achieving $3/2$ DoF

Goal: To align interference, make $\text{span}(V) \equiv \text{span}(TV)$
$K = 3$ user interference channel, achieving $3/2$ DoF

Goal: To align interference, make $\text{span}(V) \equiv \text{span}(TV)$

Solution 1: $V = \text{eig}(T)$
Works for MIMO
Does not work for SISO because desired signals sit in the span of interference.
$K = 3$ user interference channel, achieving $3/2$ DoF

Goal: To align interference, make $\text{span}(V) \equiv \text{span}(TV)$

Solution 1: $V = \text{eig}(T)$
Works for MIMO
Does not work for SISO because desired signals sit in the span of interference.

Solution 2: Asymptotic Alignment (uses large number of dimensions) (works for SISO)
Over $2n + 1$ symbols, use $V = \{w, Tw, T^2w, \ldots, T^nw\}$
Achieves $3n/2n + 1$ DoF.
(Actually one more vector can be squeezed in to achieve $(3n + 1)/(2n + 1)$ DoF)
Achieving $K/2$ DOF in $K$ user Interference Channel
Achieving $K/2$ DOF in $K$ user Interference Channel

Ignore direct channels.

Enumerate all cross channels $T_1, T_2, \cdots, T_N$

**Critical assumption**

Commutative property: $T_i T_j = T_j T_i$

(e.g. time-varying/frequency selective setting $\rightarrow$ diagonal channels)

All transmitters use the *same* signal space $\mathbf{V}$

All receivers set aside the *same* interference space $\mathcal{I}$

$$\mathbf{V}^{[1]} = \mathbf{V}^{[2]} = \cdots = \mathbf{V}^{[K]} = \mathbf{V}$$

$$\mathcal{I}^{[1]} = \mathcal{I}^{[2]} = \cdots = \mathcal{I}^{[K]} = \mathbf{V}$$

[Cadambe, Jafar, IT08]
Interference Alignment Scheme of [CJ08]

What is the interference space at Receiver 1?
Interference Alignment Scheme of [CJ08]

[Cadambe, Jafar, IT08]

All the interference at all the receivers: $\mathcal{I}$

Goal: Make $\mathbf{V} \equiv \mathcal{I}$
Goal: Simultaneously satisfy “N” Alignment Constraints:
span(\(\mathbf{V}\)) \equiv \text{span}(T_1 \mathbf{V}) \equiv \text{span}(T_2 \mathbf{V}) \equiv \cdots \equiv \text{span}(T_N \mathbf{V})

[Cadambe, Jafar, IT08]
Goal: Simultaneously satisfy “N” Alignment Constraints:
\[ \text{span}(\mathbf{V}) \equiv \text{span}(T_1\mathbf{V}) \equiv \text{span}(T_2\mathbf{V}) \equiv \cdots \equiv \text{span}(T_N\mathbf{V}) \]

Goal: Make \( \mathbf{V} \equiv \mathcal{I} \)

[Camambe, Jafar, IT08]
**Goal**: Simultaneously satisfy "N" Alignment Constraints:
\[
\text{span}(\mathbf{V}) \equiv \text{span}(T_1 \mathbf{V}) \equiv \text{span}(T_2 \mathbf{V}) \equiv \cdots \equiv \text{span}(T_N \mathbf{V})
\]

[ Cadambe, Jafar, IT08 ]

**Initialize**: \( \mathbf{V}_0 = \mathbf{1} \)

\[
\mathbf{V}_1 = \{1, T_1 \mathbf{1}, \ldots, T_N \mathbf{1}\}
\]

\[
\mathbf{V}_2 = \{1, \ldots, T_i \mathbf{1}, \ldots, T_i T_j \mathbf{1}, \ldots, T_i^2 \mathbf{1}\}
\]

\[
\mathbf{V}_n = \{T_1^{\alpha_1} T_2^{\alpha_2} \cdots T_N^{\alpha_N} \mathbf{1}, \alpha_1 + \cdots + \alpha_N \leq n\}
\]

\[
\mathbf{V}_{n+1} = \{T_1^{\alpha_1} T_2^{\alpha_2} \cdots T_N^{\alpha_N} \mathbf{1}, \alpha_1 + \cdots + \alpha_N \leq n + 1\}
\]

\[
|\mathbf{V}_n| = \binom{n + N}{n}
\]

\[
|\mathcal{I}| = \binom{n + N + 1}{n + 1}
\]

\[
\frac{|\mathbf{V}|}{|\mathcal{I}|} = \frac{n + 1}{n + N + 1} \to 1 \text{ as } n \to \infty
\]

**Key**: \( T_i \)'s commute
Today

• Revision and continuation (achieving $K/2$ DoF for the $K$ user interference channel)

• Interference alignment for Distributed Storage
  – A multi-source (non-multicast) network coding application
Distributed Storage System

- Storage Node
- Data Collector
Distributed Storage System

- Storage Node
- Data Collector
Servers (RAID-type systems)
Servers (RAID-type systems)
Distributed Storage: A Canonical Model

$k$ unit-size data sources, $n$ unit-size data storage nodes

$(n, k)$ systematic code

\[ f_{k+1}(A_1, \ldots, A_k) \]  
\[ f_n(A_1, \ldots, A_k) \]
Replication

\[ n = 4 \]
\[ k = 2 \]

\[ A, B \rightarrow 1 \text{ unit} \]

MDS* Codes

\[
\begin{align*}
A \\
B \\
A \\
B \\
A + B \\
A + 2B
\end{align*}
\]

*MDS = Maximum Distance Separable
**Replication**

\[ n = 4 \]
\[ k = 2 \]

\[ A, B \rightarrow 1 \text{ unit} \]

Can tolerate 1 node failure

**MDS* Codes**

- \[ A \]
- \[ A + B \]
- \[ A + 2B \]

*\text{MDS} = \text{Maximum Distance Separable}
Replication

\(n = 4\)

\(k = 2\)

\(A, B \rightarrow 1\) unit

MDS* Codes

\(A\)

\(B\)

\(A\)

\(A + B\)

\(B\)

\(A + 2B\)

Can recover from 1 failed node

*MDS = Maximum Distance Separable
Replication

\( n = 4 \)
\( k = 2 \)

\( A, B \rightarrow 1 \) unit

Can recover from 1 failed node

MDS* Codes

\( A \)

\( B \)

\( A + B \)

\( A + 2B \)

Can recover from 2 failed nodes!

*MDS = Maximum Distance Separable
Distributed Storage: A Canonical Model

$k$ unit-size data sources, $n$ unit-size data storage nodes

$(n, k)$ systematic MDS code

\[
\begin{align*}
\mathbf{A}_1 & \quad \mathbf{A}_2 & \quad \ldots & \quad \mathbf{A}_k & \quad f_{k+1}(\mathbf{A}_1, \ldots, \mathbf{A}_k) & \quad f_n(\mathbf{A}_1, \ldots, \mathbf{A}_k)
\end{align*}
\]
Distributed Storage: The Repair Efficiency Problem

\( k \) unit-size data sources, \( n \) unit-size data storage nodes
\((n, k)\) systematic MDS code

Goal:
Minimum Bandwidth
[Wu-Dimakis 09]
Replication

\[ n = 4 \]
\[ k = 2 \]

\[ A, B \rightarrow 1 \text{ unit} \]

MDS* Codes

\[ A \]

\[ B \]

\[ A + B \]

\[ A + 2B \]

Download 2 units for repair
Erasure Coding for Distributed Storage

• **Redundancy** for a fixed amount storage
  – Maximum Distance Separable Codes (MDS Codes);
    For example, Reed-Solomon Codes

• **Complexity** (Encoding and Update)
  – EVENODD, STAR, RDP, Liberation, X Codes etc.

• **(Recent) Repair Efficiency**
  – Network bandwidth
  – Disk access
Distributed Storage: The Repair Efficiency Problem

$k$ unit-size data sources, $n$ unit-size data storage nodes
$(n, k)$ systematic MDS code

Goal:
Minimum Bandwidth
[Wu-Dimakis 09]

Trivial (suboptimal) solution: $k$ units
Distributed Storage: The Repair Efficiency Problem

$k$ unit-size data sources, $n$ unit-size data storage nodes
$(n, k)$ systematic MDS code

\[ \mathbf{A}_1 \quad \mathbf{A}_2 \quad \ldots \quad \mathbf{A}_k \quad f_{k+1}(\mathbf{A}_1, \ldots, \mathbf{A}_k) \quad \ldots \quad f_n(\mathbf{A}_1, \ldots, \mathbf{A}_k) \]

Goal:
Minimum Bandwidth
[Wu-Dimakis 09]

Trivial (suboptimal) solution: $k$ units

Multi-Source (non-multicast) Network Capacity
Multi-source Network Capacity Problem

\( f_3, f_4 \sim \text{Network code} \)

\[ f_3(A, B) \]

\[ f_4(A, B) \]

Repair Bandwidth

\[ \text{MDS} \rightarrow A, B \]

\[ A \rightarrow A \]
Multi-source Network Capacity Problem

\[ f_3, f_4 \sim \text{Network code} \]

Storage capacity

Repair Bandwidth

A

B

MDS

A, B

A
Multi-source Network Capacity Problem

\[ f_3, f_4 \sim \text{Network code} \]

Storage capacity

Repair Bandwidth = 3 \( \beta \)
Multi-source Network Capacity Problem

\[ f_3, f_4 \sim \text{Network code} \]

Storage capacity

\[ f_3(A, B) \]

Repair Bandwidth = 3 \( \beta \)

MDS

A, B

A
Multi-source Network Capacity Problem

$f_3, f_4 \sim \text{Network code}$

Storage capacity

$f_3(A, B)$

$f_4(A, B)$

Repair Bandwidth $= 3 \, \beta$

Cut-Set Bound
Multi-source Network Capacity Problem

$f_3, f_4 \sim \text{Network code}$

Storage capacity

$f_3(A, B)$

$f_4(A, B)$

Repair Bandwidth $= 3 \beta$

Cut-Set Bound

Destinations on the right side of the cut can decode A,B. Therefore, the flow across the cut must be at least 2 units

$1 + 2\beta \geq 2$

$\Rightarrow$ Repair bandwidth $= 3\beta \geq 1.5$
Distributed Storage: The Repair Efficiency Problem

$k$ unit-size data sources, $n$ unit-size data storage nodes
$(n, k)$ systematic MDS code

Goal:
Minimum Bandwidth
[Wu-Dimakis 09]

Trivial (suboptimal) solution: $k$ units

Optimal to Download
\[ \frac{1}{\text{#Parities}} \] of every (surviving) node*

Multi-Source (non-multicast) Network Capacity

(Excercise) From cut-set bound, Repair Bandwidth $\geq \frac{n-1}{n-k}$, i.e., Download at least $1$/no. of parities from every surviving node.

From Interference Alignment, we can show that cut-set bound is tight

Use Asymptotic Interference Alignment à la [C-J 08]

Interference alignment : a technique useful to hit zeroes for non-desired sources
as per [Koetter-Medard 03] framework
\[ n = 4, k = 2 \] [Wu-Dimakis 09]

Trivial Repair: 4 linear combinations

\[\begin{array}{c}
A_1 \\
A_2
\end{array}\] 

\[\begin{array}{c}
B_1 \\
B_2
\end{array}\] 

\[\begin{array}{c}
A_1 + B_1 \\
A_2 + B_2
\end{array}\] 

\[\begin{array}{c}
2A_1 + B_1 \\
3A_2 + B_2
\end{array}\]
$n = 4, k = 2$, Repair with 3 Linear Combinations

[Wu-Dimakis 09]

Interference Alignment!
$n = 4, k = 2$, Repair with 3 Linear Combinations

\[ A \]
\[ A_2 \]

\[ B \]
\[ B_2 \]

\[ A + B \]
\[ A_2 + B_2 \]

\[ A, B \rightarrow 1 \times 2 \]
\[ T_1, T_2 \rightarrow 2 \times 2 \]
\[ V_1 \rightarrow 2 \times 1 \]

\[ A_1 + A_2 + B_1 + B_2 \]
\[ 2A_1 + 3A_2 + B_1 + B_2 \]

Interference Alignment!

Recovery of $A$
\[ \text{rank}[T_1 V_1 \ V_1] = 2 \]

\[ T_2 V_1 = \lambda_1 V_1 \]
Repair for $n=4$, $k=2$

**Repair of Node 1**

$$T_2 V_1 = \lambda_1 V_1$$

$$\text{rank}[T_1 V_1 \ V_1] = 2$$

**Repair of Node 2**

$$T_1 V_2 = \lambda_2 V_2$$

$$\text{rank}[T_2 V_2 \ V_2] = 2$$
Repair for $n=4$, $k=2$

$\mathbf{T}_2 \mathbf{V}_1 = \lambda_1 \mathbf{V}_1$

$\text{rank} [\mathbf{T}_1 \mathbf{V}_1 \mathbf{V}_1] = 2$

$\mathbf{T}_1 \mathbf{V}_2 = \lambda_2 \mathbf{V}_2$

$\text{rank} [\mathbf{T}_2 \mathbf{V}_2 \mathbf{V}_2] = 2$

$\mathbf{V}_1$ eigen-vector of $\mathbf{T}_2$

$\mathbf{V}_2$ eigen-vector of $\mathbf{T}_1$

Repair Vectors $\mathbf{V}_1, \mathbf{V}_2$, → Beamforming Vectors in Wireless Comm.
Coding matrices $\mathbf{T}_1, \mathbf{T}_2$, → Channel Matrices in Wireless Comm.
Repair Bandwidth-optimal solution for general (n,k)

[Camadbe, Jafar, Maleki 10]
[Suh, Ramachandran 10]

\[ T_i : \Delta \times \Delta \rightarrow \text{Random diagonal} \]
Repair Bandwidth-optimal solution for general \((n,k)\)

\[
\frac{\beta}{\Delta} \approx \frac{1}{n-k} \quad \text{[Cadambe, Jafar, Maleki 10]}
\]

\[
\frac{\text{Repair Bandwidth}}{\Delta} = (n-1)\beta
\]

\(\mathbf{T}_i : \Delta \times \Delta \rightarrow \text{Random diagonal}\)

- \(\Delta\) units
- \(\sum_{i=1}^{\kappa} \mathbf{T}_i' \mathbf{x}_i\)

Each contributes \(\frac{1}{n-k}\) part of \(\mathbf{X}_1\)

All interference aligns perfectly (asymptotically).
Repair Bandwidth-optimal solution for general \((n,k)\)

\[
\frac{\beta}{\Delta} \approx \frac{1}{n - k}
\]

[Cadambe, Jafar, Maleki 10]

\[
\frac{\text{Repair Bandwidth}}{\Delta} = (n - 1)\beta
\]

[Suh, Ramachandran 10]

\[\mathbf{V} : \beta \times \Delta\]

\[\mathbf{T}_i : \Delta \times \Delta \rightarrow \text{Random diagonal}\]

\[\sum_{i=1}^{k} \mathbf{T}_i' \mathbf{x}_i\]

\[\uparrow \Delta \downarrow \]

\[\Delta \text{ units}\]

\[\mathbf{X}_1 \quad \mathbf{X}_2 \quad \mathbf{X}_3 \quad \cdots \quad \mathbf{X}_k\]

\[\uparrow \Delta \downarrow \]

\[\mathbf{V} \quad \mathbf{V} \quad \mathbf{V} \quad \mathbf{V} \quad \mathbf{V}\]

Each contributes \(\frac{1}{n - k}\) part of \(\mathbf{X}_1\)

All interference aligns perfectly (asymptotically).
Repair Bandwidth-optimal solution for general $(n,k)$

\[
\frac{\beta}{\Delta} \approx \frac{1}{n-k} \quad \text{[Cadambe, Jafar, Maleki 10]}
\]

\[
\text{Repair Bandwidth} = \frac{(n-1)\beta}{\Delta} \quad \text{[Suh, Ramachandran 10]}
\]

\[
\mathbf{V} : \beta \times \Delta \quad \mathbf{T}_i : \Delta \times \Delta \quad \text{Random diagonal}
\]

\[
\mathbf{VT}_1 \approx \cdots \approx \mathbf{VT}_N \approx \mathbf{V}
\]

Each contributes $\frac{1}{n-k}$ part of $X_1$

All interference aligns perfectly (asymptotically).
Asymptotic Interference Alignment Scheme

\[
\frac{\beta}{\Delta} \approx \frac{1}{n - k}
\]

\[
V : \beta \times \Delta
\]

\[
T_i : \Delta \times \Delta \rightarrow \text{Random diagonal}
\]

\[
VT_1 \approx \cdots \approx VT_N \approx V
\]

**Goal:** Simultaneously satisfy “N” Alignment Constraints:

\[
\text{span}(V) \equiv \text{span}(T_1V) \equiv \text{span}(T_2V) \equiv \cdots \equiv \text{span}(T_NV)
\]
Asymptotic Interference Alignment Scheme

\[ \frac{\beta}{\Delta} \approx \frac{1}{n-k} \]

\[ \mathbf{V} : \beta \times \Delta \]

\[ \mathbf{T}_i : \Delta \times \Delta \xrightarrow{\text{Random diagonal}} \]

\[ \mathbf{V}\mathbf{T}_1 \approx \cdots \approx \mathbf{V}\mathbf{T}_N \approx \mathbf{V} \]

**Goal:** Simultaneously satisfy “N” Alignment Constraints:
\[ \text{span}(\mathbf{V}) \equiv \text{span}(\mathbf{T}_1\mathbf{V}) \equiv \text{span}(\mathbf{T}_2\mathbf{V}) \equiv \cdots \equiv \text{span}(\mathbf{T}_N\mathbf{V}) \]

Initialize: \( \mathbf{V}_0 = \mathbf{1} \)

\[ \mathbf{V}_1 = \{ \mathbf{1}, \mathbf{T}_1\mathbf{1}, \ldots, \mathbf{T}_N\mathbf{1} \} \]

\[ \mathbf{V}_2 = \{ \mathbf{1}, \ldots, \mathbf{T}_i\mathbf{1}, \ldots, \mathbf{T}_i\mathbf{T}_j\mathbf{1}, \ldots, \mathbf{T}_i^2\mathbf{1} \} \]

\[ \mathbf{V}_m = \{ \mathbf{T}_1^{\alpha_1}\mathbf{T}_2^{\alpha_2} \ldots \mathbf{T}_N^{\alpha_N}\mathbf{1}, \alpha_1 + \alpha_2 + \ldots \alpha_N \leq m \} \]

\[ \mathcal{I} = \mathbf{V}_{m+1} = \{ \mathbf{T}_1^{\alpha_1}\mathbf{T}_2^{\alpha_2} \ldots \mathbf{T}_N^{\alpha_N}\mathbf{1}, \alpha_1 + \alpha_2 + \ldots \alpha_N \leq m + 1 \} \]

**Goal:** Make \( \mathbf{V} \equiv \mathcal{I} \)

\[ m \rightarrow \infty \Rightarrow \frac{|\mathbf{V}|}{\mathcal{I}} \rightarrow 1 \]
References and Extra Reading

Available at http://www.mit.edu/~viveck/code/index.html

- Viveck R. Cadambe, Syed A. Jafar, Cheng Huang, Jin Li, Optimal Repair of MDS Codes in Distributed Storage via Subspace Interference Alignment, Available on arxiv:1106.1250 (Extra reading)
- Viveck R. Cadambe, Syed A. Jafar, Hamed Maleki, Kannan Ramchandran, Changho Suh, Asymptotic Interference Alignment for Optimal Repair of MDS Codes in Distributed Data Storage (Reference)