Canonization of Linear Codes

Thomas Feulner

University of Bayreuth

July 12, 2010
Linear Code

A linear code $C$ is a subspace of $\mathbb{F}_q^n$ of dimension $k$.

$n, k, q$ are some fixed parameters.

Generator Matrix

Let $C$ be a linear code. $\Gamma \in \mathbb{F}_q^{k \times n}$ is a generator matrix of $C$, if the rows of $\Gamma$ form a basis of $C$.

Set of Generator Matrices of a code

Let $\Gamma$ be some generator matrix of $C$. The set of all generator matrices of $C$ is the orbit $GL_k(\mathbb{F}_q)\Gamma$. 
Linear Code

A linear code $C$ is a subspace of $\mathbb{F}_q^n$ of dimension $k$.

$n, k, q$ are some fixed parameters.

Generator Matrix

Let $C$ be a linear code. $\Gamma \in \mathbb{F}_q^{k \times n}$ is a generator matrix of $C$, if the rows of $\Gamma$ form a basis of $C$.

Set of Generator Matrices of a code

Let $\Gamma$ be some generator matrix of $C$. The set of all generator matrices of $C$ is the orbit $GL_k(\mathbb{F}_q)\Gamma$. 
A linear code \( C \) is a subspace of \( \mathbb{F}_q^n \) of dimension \( k \).

\( n, k, q \) are some fixed parameters.

Let \( C \) be a linear code. \( \Gamma \in \mathbb{F}_q^{k \times n} \) is a generator matrix of \( C \), if the rows of \( \Gamma \) form a basis of \( C \).

Let \( \Gamma \) be some generator matrix of \( C \). The set of all generator matrices of \( C \) is the orbit \( GL_k(\mathbb{F}_q)\Gamma \).
**Equivalence**

**Definition**

Two linear codes \( C, C' \) are **semilinearly isometric** (or equivalent) \( \iff (\varphi, \alpha, \pi) \Gamma \) is a generator matrix of \( C' \), with

- a column permutation \( \pi \in S_n \)
- an automorphism \( \alpha \) of \( \mathbb{F}_q \) applied to each entry
- a column multiplication vector \( \varphi \in \mathbb{F}_q^n \)
Definition

Two linear codes $C, C'$ are semilinearly isometric (or equivalent) if and only if $(\varphi, \alpha, \pi)\Gamma$ is a generator matrix of $C'$, with

- a column permutation $\pi \in S_n$
- an automorphism $\alpha$ of $\mathbb{F}_q$ applied to each entry
- a column multiplication vector $\varphi \in \mathbb{F}_q^*$
Two linear codes $C, C'$ are semilinearly isometric (or equivalent) if and only if $(\varphi, \alpha, \pi) \Gamma$ is a generator matrix of $C'$, with

- a column permutation $\pi \in S_n$
- an automorphism $\alpha$ of $\mathbb{F}_q$ applied to each entry
- a column multiplication vector $\varphi \in \mathbb{F}_q^*$
Equivalence

Definition

Two linear codes $C, C'$ are **semilinearly isometric** (or equivalent) \( \iff (\varphi, \alpha, \pi)\Gamma \) is a generator matrix of $C'$, with

- a **column permutation** $\pi \in S_n$
- an **automorphism** $\alpha$ of $\mathbb{F}_q$ applied to each entry
- a **column multiplication vector** $\varphi \in \mathbb{F}_q^n$
Goal

Canonization Algorithm Can

Input: A generator matrix $\Gamma$

Output: A generator matrix $\text{Can}(\Gamma)$ which generates an equivalent code such that the result is unique for equivalent generator matrices.

Byproduct: The automorphism group of the code, i.e. the stabilizer subgroup of $\Gamma$. 

Goal

Canonization Algorithm Can

**Input:** A generator matrix $\Gamma$

**Output:** A generator matrix $\text{Can}(\Gamma)$ which generates an equivalent code such that the result is unique for equivalent generator matrices.

**Byproduct:** The automorphism group of the code, i.e. the stabilizer subgroup of $\Gamma$. 

Goal

Canonization Algorithm Can

**Input:** A generator matrix $\Gamma$

**Output:** A generator matrix $\text{Can}(\Gamma)$ which generates an equivalent code such that the result is **unique for equivalent generator matrices**.

**Byproduct:** The automorphism group of the code, i.e. the stabilizer subgroup of $\Gamma$. 
**Goal**

### Canonization Algorithm Can

**Input:** A generator matrix $\Gamma$

**Output:** A generator matrix $\text{Can}(\Gamma)$ which generates an equivalent code such that the result is **unique for equivalent generator matrices**.

**Byproduct:** The automorphism group of the code, i.e. the stabilizer subgroup of $\Gamma$. 
Goal: Canonization

Tool: Group action on generator matrices

\[(\text{GL}_k(\mathbb{F}_q) \times (\mathbb{F}_q^*)^n) \rtimes (\text{Aut}(\mathbb{F}_q) \times S_n)\]
Goal: Canonization

Tool: Group action on generator matrices

\[(\text{GL}_k(\mathbb{F}_q) \times (\mathbb{F}_q^*)^n) \rtimes (\text{Aut}(\mathbb{F}_q) \times S_n)\]

Let $\Gamma, \Gamma' \in \mathbb{F}_q^{k \times n}$ be equivalent generator matrices.
Goal: Canonization

Tool: Group action on generator matrices

\[(GL_k(F_q) \times (F_q^*)^n) \rtimes (\text{Aut}(F_q) \times S_n)\]

Let \(\Gamma, \Gamma' \in F_q^{k \times n}\) be equivalent generator matrices.

\[\text{Can}(\Gamma) = \text{Can}(\Gamma')\]

unique canonical representative

orbit of equivalent generator matrices
The partition and refinement idea

There is a well-known, very fast canonization algorithm for graphs:

```plaintext
nauty (B. McKay)
```

based on

```
Partition & Refinement
```
The Refinement step

Calculate properties of the vertices, invariant under relabeling!

\[
\begin{array}{c}
1 \\
\downarrow \\
0 \\
\downarrow \\
2 \quad 3
\end{array}
\]

Calculate the degree of the vertices

\[
\begin{array}{c|cccc}
  \text{i} & 0 & 1 & 2 & 3 \\
  \text{degree(i)} & 3 & 1 & 2 & 2
\end{array}
\]
The Refinement step

Calculate properties of the vertices, invariant under relabeling!

Calculate the degree of the vertices

<table>
<thead>
<tr>
<th>i</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>degree(i)</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Sort in descending order

<table>
<thead>
<tr>
<th>i</th>
<th>0</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>degree(i)</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
The Refinement step

Calculate properties of the vertices, invariant under relabeling!

Calculate the degree of the vertices

<table>
<thead>
<tr>
<th>i</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>degree(i)</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Sort in descending order

<table>
<thead>
<tr>
<th>i</th>
<th>0</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>degree(i)</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Relabel the vertices

<table>
<thead>
<tr>
<th>i</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>degree(i)</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
The Partition step

Do a backtracking procedure.
The Partition step

Do a backtracking procedure.

Choose a block of vertices which have the same color.
The Partition step

Do a backtracking procedure.

Choose a block of vertices which have the same color.

Investigate all possibilities to color one vertex in this block with a new color and to give it the smallest label.
The Partition step

Do a backtracking procedure.

The comparison of the leaf nodes yields “=”:

- $(1, 3)$ and $(1, 2)(1, 3)$ map the graph to its canonical representative
- $(1, 3)^{-1}(1, 2)(1, 3)$ is the only automorphism
# Comparison: Graphs and linear Codes

<table>
<thead>
<tr>
<th>Group Action</th>
<th>Graphs</th>
<th>linear Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_n \nabla 2^\binom{n}{2}$</td>
<td>$\left( (\text{GL}_k(F_q) \times F_q^n) \rtimes (\text{Aut}(F_q) \times S_n) \right) \nabla F_q^{k \times n}$</td>
<td></td>
</tr>
</tbody>
</table>
### Comparison: Graphs and linear Codes

<table>
<thead>
<tr>
<th>Group Action</th>
<th>Graphs</th>
<th>linear Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_n \parallel 2^{n \choose 2}$</td>
<td>$((GL_k(F_q) \times F_q^n) \rtimes (\text{Aut}(F_q) \times S_n)) \rhd F_q^{k \times n}$ replace by $S_n \parallel \left[ ((GL_k(F_q) \times (F_q^*)^n) \rtimes \text{Aut}(F_q)) \rhd F_q^{k \times n} \right]$</td>
<td></td>
</tr>
</tbody>
</table>
**Comparison: Graphs and linear Codes**

<table>
<thead>
<tr>
<th>Group Action</th>
<th>Graphs</th>
<th>Linear Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_n \not\sim 2^{n\choose 2}$</td>
<td>$S_n \not\sim \left[\left(\text{GL}_k(F_q) \times (F_q^*)^n\right) \times \text{Aut}(F_q)\right] \not\sim F_q^{k \times n}$</td>
<td></td>
</tr>
</tbody>
</table>
### Comparison: Graphs and linear Codes

<table>
<thead>
<tr>
<th>Group Action</th>
<th>Graphs</th>
<th>linear Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$S_n \downarrow 2^{n\choose 2}$</td>
<td>$S_n \downarrow [((\text{GL}_k(F_q) \times (F_q^*)^n) \rtimes \text{Aut}(F_q)) \downarrow F_q^{k \times n}]$</td>
</tr>
<tr>
<td>Re-refinement</td>
<td>$2^{n\choose 2} \rightarrow \chi^n$</td>
<td>$G$-homomorphism for some appropriate $G \leq S_n$</td>
</tr>
</tbody>
</table>

#### Homomorphism of group actions

Let $G$ act on $X, Y$.  
$f : X \rightarrow Y$ is a $G$-homomorphism if

$$f(gx) = gf(x), \ \forall \ x \in X, g \in G$$
## Comparison: Graphs and linear Codes

<table>
<thead>
<tr>
<th></th>
<th>Graphs</th>
<th>linear Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Group Action</strong></td>
<td>$S_n \parallel 2 \binom{n}{2}$</td>
<td>$S_n \parallel \left[ ((\text{GL}_k(F_q) \times (F_q^*)^n) \rtimes \text{Aut}(F_q)) \parallel \mathbb{F}_q^{k \times n} \right]$</td>
</tr>
<tr>
<td><strong>Refinement</strong></td>
<td>$2 \binom{n}{2} \rightarrow X^n$</td>
<td>$((\text{GL}_k(F_q) \times (F_q^*)^n) \rtimes \text{Aut}(F_q)) \parallel \mathbb{F}_q^{k \times n} \rightarrow X^n$</td>
</tr>
<tr>
<td></td>
<td>$G$-homomorphism for some appropriate $G \leq S_n$</td>
<td></td>
</tr>
</tbody>
</table>

### Homomorphism of group actions

Let $G$ act on $X, Y$.

$f : X \rightarrow Y$ is a $G$-homomorphism if

$$f(gx) = gf(x), \ \forall \ x \in X, \ g \in G$$
An example in the binary case

Canonize the matrix

\[
\Gamma = \begin{pmatrix}
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1
\end{pmatrix} \in \mathbb{F}_2^{3 \times 4}
\]
An example in the binary case

Canonize the matrix

\[ \Gamma = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \in \mathbb{F}_2^{3 \times 4} \]

Refinement step

Find a \( S_n \)-homomorphism

\[ f : (GL_3(\mathbb{F}_2) \parallel \mathbb{F}_2^{3 \times 4}) \rightarrow \chi^n \]
An example in the binary case: First Refinement

\[ \Gamma = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \]

Use

\[ f(\text{GL}_3(\mathbb{F}_2) \cdot \Gamma) := \left( \dim(C_0^\Gamma), \ldots, \dim(C_3^\Gamma) \right) \]

\[ C^\Gamma := \text{the code generated by } \Gamma \]

\[ C_i := \text{the puncturing of } C \text{ at postion } i \]
An example in the binary case: First Refinement

\[ \Gamma = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \]

Use

\[ f(\text{GL}_3(\mathbb{F}_2) \cdot \Gamma) := (3, 2, 3, 3) \]
An example in the binary case: First Refinement

\[ \Gamma = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \]

Use

\[ f(\text{GL}_3(\mathbb{F}_2) \cdot \Gamma) := (3, 2, 3, 3) \; \rightsquigarrow \; (2, 3, 3, 3) \]
An example in the binary case: Refinement

\[
\begin{pmatrix}
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 \\
\end{pmatrix}
\rightarrow
\begin{pmatrix}
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
1 & 0 & 1 & 1 \\
\end{pmatrix}
\]

\((0, 1)\)
An example in the binary case: Refinement

Application of the inner group action:

Minimize the fixed columns.

\[
\begin{pmatrix}
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 \\
\end{pmatrix}
\rightarrow
\begin{pmatrix}
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 \\
\end{pmatrix}
\]
An example in the binary case: Refinement

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Application of the inner group action:

- **Minimize** the fixed columns.

Further Application of the inner group action:

- Use just the **stabilizer** of the fixed columns for further minimization.
An example in the binary case: Refinement

\[
\begin{pmatrix}
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 \\
\end{pmatrix}
\rightarrow
\begin{pmatrix}
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 \\
\end{pmatrix}
\rightarrow
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 \\
\end{pmatrix}
\]

Application of the inner group action:

Minimize the fixed columns.

Further Application of the inner group action:

Use just the stabilizer of the fixed columns for further minimization.

Further Refinements:

Restrict to \( G \leq S_n \) stabilizing the colors.
An example in the binary case: Backtracking (Partitioning)

\[
\begin{pmatrix}
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0
\end{pmatrix}
\]
An example in the binary case: Backtracking (Partitioning)

Application of the inner group action:
Minimize the fixed columns

Application of the inner group action:
Prune nodes, whose fixed columns are not minimal.
The whole example

(0, 1)

(1, 2)

(1, 3)

(2, 3)
The whole example

Canonical representative:
A minimal \(\text{(including images of the invariants)}\) leaf node of the pruned tree.
The whole example

Automorphisms:

\[(\text{root} \rightarrow \text{equal leaf node})^{-1} \cdot (\text{root} \rightarrow \text{leaf node})\]
Pruning by Automorphism

Traverse the tree in **depth-first-search**
Pruning by Automorphism

Traverse the tree in **depth-first-search**

![Tree Diagram]

**Application of Automorphisms:**
Prune subtrees which carry no new information.
Canonization of APN-Functions

**CCZ-Equivalence**

CCZ-Equivalence = usual code equivalence

**EA-Equivalence**

Restrict the inner group $\text{GL}_k(F_2)$ to the subgroup

\[
\begin{pmatrix}
1 & 0 & 0 \\
a & A & 0 \\
b & B & C
\end{pmatrix}
\]

**Affine Equivalence**

Restrict the inner group $\text{GL}_k(F_2)$ to the subgroup

\[
\begin{pmatrix}
1 & 0 & 0 \\
a & A & 0 \\
b & 0 & C
\end{pmatrix}
\]
Canonization of APN-Functions

CCZ-Equivalence

CCZ-Equivalence = usual code equivalence

EA-Equivalence

Restrict the inner group $\text{GL}_k(\mathbb{F}_2)$ to the subgroup

$$
\begin{pmatrix}
1 & 0 & 0 \\
\alpha & A & 0 \\
\beta & B & C \\
\end{pmatrix}
$$

Affine Equivalence

Restrict the inner group $\text{GL}_k(\mathbb{F}_2)$ to the subgroup

$$
\begin{pmatrix}
1 & 0 & 0 \\
\alpha & A & 0 \\
\beta & 0 & C \\
\end{pmatrix}
$$
Canonization of APN-Functions

**CCZ-Equivalence**

CCZ-Equivalence = usual code equivalence

**EA-Equivalence**

Restrict the inner group $\text{GL}_k(\mathbb{F}_2)$ to the subgroup

$$
\begin{pmatrix}
1 & 0 & 0 \\
0 & 0 & 0 \\
1 & A & 0 \\
b & B & C
\end{pmatrix}
$$

**Affine Equivalence**

Restrict the inner group $\text{GL}_k(\mathbb{F}_2)$ to the subgroup

$$
\begin{pmatrix}
1 & 0 & 0 \\
0 & 0 & 0 \\
1 & A & 0 \\
b & 0 & C
\end{pmatrix}
$$