Classification of Linear Codes with Prescribed Minimum Distance and New Upper Bounds

Thomas Feulner

University of Bayreuth

3ICMCTA
September 13, 2011
Introduction

Motivation

- Gaps between lower and upper bounds – http://codetables.de. (Show existence or nonexistence for the upper bound)
- Full classification of linear codes having certain parameters. (There is no self-dual [72, 36, 16]_2-code with automorphism of order 7! – joint work with G. Nebe)

Inspired by

- I. Bouyukliev, E. Jacobsson, Results on Binary Linear Codes With Minimum Distance 8 and 10, arXiv.org, abs/1006.0109, (2010)
Introduction

Motivation

- Gaps between lower and upper bounds – [http://codetables.de](http://codetables.de) (Show existence or nonexistence for the upper bound)

- Full classification of linear codes having certain parameters. (There is no self-dual [72, 36, 16]_2-code with automorphism of order 7! – joint work with G. Nebe)

Inspired by


- I. Bouyukliev, E. Jacobsson, Results on Binary Linear Codes With Minimum Distance 8 and 10, arXiv.org, abs/1006.0109, (2010)
# Introduction

Motivation

Gaps between lower and upper bounds – [http://codetables.de](http://codetables.de). (Show existence or nonexistence for the upper bound)

Full classification of linear codes having certain parameters.

There is no self-dual \([72,36,16]^2\)-code with automorphism of order 7! – joint work with G. Nebe


I. Bouyukliev, E. Jacobsson, Results on Binary Linear Codes With Minimum Distance 8 and 10, arXiv.org, abs/1006.0109, (2010)
Introduction

Motivation

- Gaps between lower and upper bounds – http://codetables.de. (Show existence or nonexistence for the upper bound)
- Full classification of linear codes having certain parameters. (There is no self-dual [72, 36, 16]_2-code with automorphism of order 7! – joint work with G. Nebe)

Inspired by

- I. Bouyukliev, E. Jacobsson, Results on Binary Linear Codes With Minimum Distance 8 and 10, arXiv.org, abs/1006.0109, (2010)
## Introduction

### Motivation
- Gaps between lower and upper bounds – [http://codetables.de](http://codetables.de). (*Show existence or nonexistence for the upper bound*)
- Full classification of linear codes having certain parameters. (*There is no self-dual [72, 36, 16]_2-code with automorphism of order 7! – joint work with G. Nebe*)

### Inspired by
- I. Bouyukliev, E. Jacobsson, Results on Binary Linear Codes With Minimum Distance 8 and 10, arXiv.org, abs/1006.0109, (2010)
Introduction

Motivation

- Gaps between lower and upper bounds – [http://codetables.de](http://codetables.de). (Show existence or nonexistence for the upper bound)
- Full classification of linear codes having certain parameters. (There is no self-dual $[72, 36, 16]_2$-code with automorphism of order 7! – joint work with G. Nebe)

Inspired by

- I. Bouyukliev, E. Jacobsson, Results on Binary Linear Codes With Minimum Distance 8 and 10, arXiv.org, abs/1006.0109, (2010)
Basics

Notation

Let $C$ be an $[n, k, d]_q$-code. If $C^\perp$ has minimum distance $d^\perp$ we also write $[n, k, d]_q^{d^\perp}$.

Semilinear Mappings

A mapping $\sigma : \mathbb{F}_q^n \to \mathbb{F}_q^n$ is called semilinear, if there exists some $\alpha \in \text{Aut}(\mathbb{F}_q)$ with

- $\sigma(u + v) = \sigma(u) + \sigma(v)$
- $\sigma(\kappa u) = \alpha(\kappa)\sigma(u)$
Introduction

Inverting Construction \( Y_1 \)

Results

Basics

Notation

Let \( C \) be an \([n, k, d]_q\)-code. If \( C^\perp \) has minimum distance \( d^\perp \) we also write \([n, k, d]_{d^\perp} \).

Semilinear Mappings

A mapping \( \sigma : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n \) is called semilinear, if there exists some \( \alpha \in \text{Aut}(\mathbb{F}_q) \) with

\[
\begin{align*}
\sigma(u + v) &= \sigma(u) + \sigma(v) \\
\sigma(\kappa u) &= \alpha(\kappa)\sigma(u)
\end{align*}
\]
Basics

Notation

Let $C$ be an $[n, k, d]_q$-code. If $C^\perp$ has minimum distance $d^\perp$ we also write $[n, k, d]_q^{d^\perp}$.

Semilinear Mappings

A mapping $\sigma : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$ is called semilinear, if there exists some $\alpha \in \text{Aut}(\mathbb{F}_q)$ with

- $\sigma(u + v) = \sigma(u) + \sigma(v)$
- $\sigma(\kappa u) = \alpha(\kappa)\sigma(u)$
Basics

Notation
Let $C$ be an $[n, k, d]_q$-code. If $C^\perp$ has minimum distance $d^\perp$ we also write $[n, k, d]_q^{d^\perp}$.

Semilinear Mappings
A mapping $\sigma : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$ is called semilinear, if there exists some $\alpha \in \text{Aut}(\mathbb{F}_q)$ with
- $\sigma(u + v) = \sigma(u) + \sigma(v)$
- $\sigma(\kappa u) = \alpha(\kappa)\sigma(u)$
Basics

**Code Equivalence**

Two \([n, k, d]_q^{d\perp}\)-codes \(C, C'\) are equivalent \(\iff\) exists some semilinear isometry \(\iota\) with \(\iota(C) = C'\).

**Equivalence of matrices**

Similarly we say that generator (parity check) matrices are equivalent if they represent equivalent codes.

**Transversal of equivalence classes**

\(T(n, k, d, d^{\perp}, q)\) denotes a complete set of non-equivalent parity check matrices of all \([n, k, \geq d]_q^{d^{\perp}}\)-codes.
 Basics

**Code Equivalence**

Two \([n, k, d]_{q}^{d\perp}\)-codes \(C, C'\) are equivalent \(\iff\) exists some semilinear isometry \(\iota\) with \(\iota(C) = C'\).

**Equivalence of matrices**

Similarly we say that generator (parity check) matrices are equivalent if they represent equivalent codes.

**Transversal of equivalence classes**

\(T(n, k, d, d\perp, q)\) denotes a complete set of non-equivalent parity check matrices of all \([n, k, \geq d]_{q}^{d\perp}\)-codes.
Basics

Code Equivalence

Two \([n, k, d]_{q}^{d_{\perp}}\)-codes \(C, C'\) are equivalent \iff\ exists some semilinear isometry \(\iota\) with \(\iota(C) = C'\).

Equivalence of matrices

Similarly we say that generator (parity check) matrices are equivalent if they represent equivalent codes.

Transversal of equivalence classes

\(T(n, k, d, d_{\perp}, q)\) denotes a complete set of non-equivalent parity check matrices of all \([n, k, \geq d]_{q}^{d_{\perp}}\)-codes.
A canonical form algorithm

Unique Orbit Representatives

With the help of the algorithm


we can compute unique orbit representatives and hence determine $T(n, k, d, d^\perp, q)$ from a superset $\mathcal{T}(n, k, d, d^\perp, q)$ very efficiently.

Problem

Compute small supersets $\mathcal{T}(n, k, d, d^\perp, q)$ iteratively.
A canonical form algorithm

Unique Orbit Representatives

With the help of the algorithm


we can compute unique orbit representatives and hence determine \( T(n, k, d, d^\perp, q) \) from a superset \( \mathcal{T}(n, k, d, d^\perp, q) \) very efficiently.

Problem

Compute small supersets \( \mathcal{T}(n, k, d, d^\perp, q) \) iteratively.
Construction $Y_1$

Let $C$ be an $[n, k, d]_q^{d^\perp}$-code. Then there exists an $[n - d^\perp, k - d^\perp + 1, \geq d]_q^{d^\perp}$-code.

Proof.

Without loss of generality $C$ has a parity check matrix of the following form

$$
\Delta := \begin{pmatrix}
\Delta' \\
0_{n-d^\perp} \\
\chi \\
c
\end{pmatrix}
$$

with $(0_{n-d^\perp}, c) \in C^\perp$ a codeword of minimum distance $\text{wt}(c) = d^\perp$. The code with parity check matrix $\Delta'$ has got the desired parameters.
Construction $Y_1$

Let $C$ be an $[n, k, d]_q^{d \perp}$-code. Then there exists an $[n - d \perp, k - d \perp + 1, \geq d]_q^{\geq \left\lceil \frac{d \perp}{q} \right\rceil}$-code.

Proof.

Without loss of generality $C$ has a parity check matrix of the following form

$$\Delta := \begin{pmatrix} \Delta' & X \\ 0_{n-d \perp} & c \end{pmatrix}$$

with $(0_{n-d \perp}, c) \in C^\perp$ a codeword of minimum distance $\text{wt}(c) = d \perp$. The code with parity check matrix $\Delta'$ has got the desired parameters.
Inverting Construction $Y_1$

\[
\begin{align*}
n - d^\perp & \geq d \\
\geq \left\lceil \frac{d^\perp}{q} \right\rceil
\end{align*}
\]
Inverting Construction $Y_1$

$$n - d^\perp \geq d \geq \left\lceil \frac{d^\perp}{q} \right\rceil \geq d \geq 1$$
### Inverting Construction $Y_1$

$$n - d_{\perp}$$

<table>
<thead>
<tr>
<th>$\geq d$</th>
<th>$\geq \left\lceil \frac{d_{\perp}}{q} \right\rceil$</th>
<th>$\geq d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\geq 1$</td>
<td>$\geq 2$</td>
<td>$\geq 1$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$n$</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>$k - d_{\perp} - 1$</th>
<th>$d$</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>$k$</th>
<th>$d_{\perp}$</th>
</tr>
</thead>
</table>
\begin{equation}
\begin{align*}
\frac{n}{k} &\geq d \\
\frac{n}{k} - \frac{d^\perp}{q} &\geq d \\
\frac{n}{1} &\geq d \\
\frac{n}{2} &\geq d \\
\frac{n}{3} &\geq d \\
\frac{n}{d^\perp} &\geq d \\
\frac{n}{k - d^\perp - 1} &\geq d \\
\frac{n}{k} &\geq d \\
\frac{n}{k} - \frac{d^\perp}{q} &\geq d \\
\frac{n}{1} &\geq d \\
\frac{n}{2} &\geq d \\
\frac{n}{3} &\geq d \\
\frac{n}{d^\perp} &\geq d \\
\frac{n}{k - d^\perp - 1} &\geq d \\
\frac{n}{k} &\geq d \\
\end{align*}
\end{equation}
Inverting Construction \( Y_1 \)

**Iteration Starting Point**

Let \( S \) be an arbitrary transversal of parity check matrices of all \([n - d^\perp, k - d^\perp + 1, \geq d]_q\)-codes.

**Existence of predecessors**

Each equivalence class of parity check matrices of the \([n, k, \geq d]_q^{d^\perp}\)-codes contains at least one matrix

\[
\tilde{\Delta} = \begin{pmatrix} \Delta' & X \\ 0_{n-d^\perp} & 1_{d^\perp} \end{pmatrix}
\]

with

- \( \Delta' \in S \)
- \( X \in \mathbb{F}_q^{(n-k-1) \times d^\perp} \) with lexicographically ordered columns
Invertin Construction \( Y_1 \)

**Iteration Starting Point**

Let \( S \) be an arbitrary transversal of parity check matrices of all \([n - d^\perp, k - d^\perp + 1, \geq d]_q\) -codes.

**Existence of predecessors**

Each equivalence class of parity check matrices of the \([n, k, \geq d]_q^{d^\perp}\) -codes contains at least one matrix

\[
\tilde{\Delta} = \begin{pmatrix} \Delta' & X \\ 0_{n-d^\perp} & 1_{d^\perp} \end{pmatrix}
\]

with

- \( \Delta' \in S \)
- \( X \in \mathbb{F}_q^{(n-k-1)\times d^\perp} \) with lexicographically ordered columns
Inverting Construction \( Y_1 \)

**Iteration Starting Point**

Let \( S \) be an arbitrary transversal of parity check matrices of all \([n - d^\perp, k - d^\perp + 1, \geq d]_q\)-codes.

**Existence of Predecessors**

Each equivalence class of parity check matrices of the \([n, k, \geq d]_q^{d^\perp}\)-codes contains at least one matrix

\[
\tilde{\Delta} = \begin{pmatrix}
\Delta' \\
0_{n-d^\perp} \\
\end{pmatrix}
\begin{pmatrix}
X \\
1_{d^\perp}
\end{pmatrix}
\]

with

- \( \Delta' \in S \)
- \( X \in \mathbb{F}_q^{(n-k-1) \times d^\perp} \) with lexicographically ordered columns.
**Inverting Construction \( Y_1 \)**

**Iteration Starting Point**

Let \( S \) be an arbitrary transversal of parity check matrices of all \([n - d\perp, k - d\perp + 1, \geq d]_q\)-codes.

**Existence of predecessors**

Each equivalence class of parity check matrices of the \([n, k, \geq d]_q^{d\perp}\)-codes contains at least one matrix

\[
\tilde{\Delta} = \begin{pmatrix}
\Delta' & X \\
0_{n-d\perp} & 1_{d\perp}
\end{pmatrix}
\]

with

- \( \Delta' \in S \)
- \( X \in \mathbb{F}_q^{(n-k-1) \times d\perp} \) with lexicographically ordered columns
A special transversal

Choosing the smallest matrix

\[ \tilde{\Delta} = \begin{pmatrix} \Delta' & X \\ 0_{n-d^\perp} & 1_{d^\perp} \end{pmatrix} \]

in each equivalence class defines a transversal \( T(n, k, d, d^\perp, S, q) \).
Inverting Construction $Y_1$

\[ T(n, k, d, d^\perp, S, q) \]
Inverting Construction $Y_1$

\[ T(n, k, d, d^\perp, S, q) \]

\[ T(n, k, d, 1, S, q) \]
Inverting Construction $Y_1$

\[ n - d^\perp \]

\[ T(n, k, d, 1, S, q) \]

\[ T(n, k, d, 2, S, q) \]

\[ T(n, k, d, d^\perp, S, q) \]
Inverting Construction $Y_1$

\[
T(n, k, d, 1, S, q) \quad T(n, k, d, 2, S, q) \quad T(n, k, d, d_\perp, S, q)
\]
Details

Computation of $T(n, k, d, 1, S, q)$

- Define $\mathcal{T}(n, k, d, 1, S, q) := \left\{ \begin{pmatrix} \Delta' & 0 \\ 0_{n-d\perp} & 1 \end{pmatrix} | \Delta' \in S \right\}$
- Filter $\mathcal{T}(n, k, d, 1, S, q)$ for nonisomorphic copies

Computation of $T(n, k, d, d\perp, S, q), \; d\perp \geq 2$

- Compute $\mathcal{T}(n, k, d, d\perp, S, q)$: For all

\[
\begin{pmatrix}
\Delta' & 0 & x_1 & \cdots & x_{d\perp-2} \\
0_{n-d\perp} & 1 & 1 & \cdots & 1
\end{pmatrix} \in T(n-1, k-1, d, d\perp-1, S, q)
\]

add all possible columns \(x_{d\perp-1} \geq x_{d\perp-2}\) which fulfills the conditions on $d$ and $d\perp$.
- Filter $\mathcal{T}(n, k, d, d\perp, S, q)$ for nonisomorphic copies.
Details

Computation of $T(n, k, d, 1, S, q)$

- Define $\mathcal{T}(n, k, d, 1, S, q) := \left\{ \begin{pmatrix} \Delta' & 0 \\ 0 & 1 \end{pmatrix} | \Delta' \in S \right\}$

- Filter $\mathcal{T}(n, k, d, 1, S, q)$ for nonisomorphic copies

Computation of $T(n, k, d, d^\perp, S, q)$, $d^\perp \geq 2$

- Compute $\mathcal{T}(n, k, d, d^\perp, S, q)$: For all

\[
\begin{pmatrix}
\Delta' & 0 & x_1 & \cdots & x_{d^\perp-2} \\
0_{n-d^\perp} & 1 & 1 & \cdots & 1
\end{pmatrix}
\in T(n-1, k-1, d, d^\perp-1, S, q)
\]

add all possible columns $\left( \begin{array}{c} x_{d^\perp-1} \\ 1 \end{array} \right)$ with $x_{d^\perp-1} \geq x_{d^\perp-2}$ which fulfills the conditions on $d$ and $d^\perp$.

- Filter $\mathcal{T}(n, k, d, d^\perp, S, q)$ for nonisomorphic copies.
Details

**Computation of** $T(n, k, d, 1, S, q)$

- Define $T(n, k, d, 1, S, q) := \left\{ \begin{pmatrix} \Delta' & 0 \\ 0_{n-d} & 1 \end{pmatrix} | \Delta' \in S \right\}$
- Filter $T(n, k, d, 1, S, q)$ for nonisomorphic copies

**Computation of** $T(n, k, d, d^\perp, S, q), \ d^\perp \geq 2$

- Compute $T(n, k, d, d^\perp, S, q)$: For all 
  \[
  \begin{pmatrix}
  \Delta' & 0 & x_1 & \cdots & x_{d^\perp-2} \\
  0_{n-d^\perp} & 1 & 1 & \cdots & 1
  \end{pmatrix}
  \in T(n-1, k-1, d, d^\perp-1, S, q)
  
  add all possible columns ($x_{d^\perp-1}$) with $x_{d^\perp-1} \geq x_{d^\perp-2}$ which fulfills the conditions on $d$ and $d^\perp$.
- Filter $T(n, k, d, d^\perp, S, q)$ for nonisomorphic copies.
Details

**Computation of** $T(n, k, d, 1, S, q)$

- Define $\mathcal{T}(n, k, d, 1, S, q) := \left\{ \begin{pmatrix} \Delta' & 0 \\ 0_{n-d} & 1 \end{pmatrix} | \Delta' \in S \right\}$
- Filter $\mathcal{T}(n, k, d, 1, S, q)$ for nonisomorphic copies

**Computation of** $T(n, k, d, d^\perp, S, q), \ d^\perp \geq 2$

- Compute $\mathcal{T}(n, k, d, d^\perp, S, q)$: For all
  
  $\begin{pmatrix} \Delta' & 0 & x_1 & \cdots & x_{d^\perp-2} \\ 0_{n-d^\perp} & 1 & 1 & \cdots & 1 \end{pmatrix} \in T(n-1, k-1, d, d^\perp-1, S, q)$

  add all possible columns $(\begin{pmatrix} x_{d^\perp-1} \\ 1 \end{pmatrix})$ with $x_{d^\perp-1} \geq x_{d^\perp-2}$ which fulfills the conditions on $d$ and $d^\perp$.
- Filter $\mathcal{T}(n, k, d, d^\perp, S, q)$ for nonisomorphic copies.
Details

**Computation of $T(n, k, d, 1, S, q)$**

- Define $\mathcal{T}(n, k, d, 1, S, q) := \left\{ \begin{pmatrix} \Delta' & 0 \\ 0_{n-d\perp} & 1 \end{pmatrix} \mid \Delta' \in S \right\}$
- Filter $\mathcal{T}(n, k, d, 1, S, q)$ for nonisomorphic copies

**Computation of $T(n, k, d, d\perp, S, q), \ d\perp \geq 2$**

- Compute $\mathcal{T}(n, k, d, d\perp, S, q)$: For all

$$\begin{pmatrix} \Delta' & 0 & x_1 & \cdots & x_{d\perp-2} \\ 0_{n-d\perp} & 1 & 1 & \cdots & 1 \end{pmatrix} \in T(n-1, k-1, d, d\perp-1, S, q)$$

add all possible columns $\begin{pmatrix} x_{d\perp-1} \\ 1 \end{pmatrix}$ with $x_{d\perp-1} \geq x_{d\perp-2}$ which fulfills the conditions on $d$ and $d\perp$.

- Filter $\mathcal{T}(n, k, d, d\perp, S, q)$ for nonisomorphic copies.
Details

Computation of $T(n, k, d, 1, S, q)$

- Define $\mathcal{T}(n, k, d, 1, S, q) := \left\{ \begin{pmatrix} \Delta' & 0 \\ 0_{n-d\perp} & 1 \end{pmatrix} | \Delta' \in S \right\}$
- Filter $\mathcal{T}(n, k, d, 1, S, q)$ for nonisomorphic copies

Computation of $T(n, k, d, d\perp, S, q), \ d\perp \geq 2$

- Compute $\mathcal{T}(n, k, d, d\perp, S, q)$: For all
  \[
  \begin{pmatrix}
  \Delta' & 0 & x_1 & \cdots & x_{d\perp-2} \\
  0_{n-d\perp} & 1 & 1 & \cdots & 1
  \end{pmatrix} \in T(n-1, k-1, d, d\perp-1, S, q)
  \]
  add all possible columns $\begin{pmatrix} x_{d\perp-1} \\ 1 \end{pmatrix}$ with $x_{d\perp-1} \geq x_{d\perp-2}$ which fulfills the conditions on $d$ and $d\perp$.
- Filter $\mathcal{T}(n, k, d, d\perp, S, q)$ for nonisomorphic copies.
An Example: Does a $[21, 14, 6]_4$-code exist?

From [http://codetables.de](http://codetables.de) we determine that $d^\perp \in \{9, 10, 11\}$. The following table gives the number of equivalence classes for $d \geq 6$, distinguished by $d^\perp$:

<table>
<thead>
<tr>
<th>$n$</th>
<th>$n - k = 6$</th>
<th>$n - k = 7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>$1^0 \ldots 5^0 6^1$</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>$1^1$</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>$2^2$</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>$3^7$</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>$4^{13}$</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>$5^9$</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>$6^5$</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>$7^1$</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>$8^1$</td>
</tr>
<tr>
<td>21</td>
<td></td>
<td>$9^0$</td>
</tr>
</tbody>
</table>
An Example: Does a $[21, 14, 6]_4$-code exist?

From http://codetables.de we determine that $d^\perp \in \{9, 10, 11\}$. The following table gives the number of equivalence classes for $d \geq 6$, distinguished by $d^\perp$:

<table>
<thead>
<tr>
<th>n</th>
<th>$n - k = 6$</th>
<th>$n - k = 7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>$1^0 \ldots 4^0 5^1$</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>$1^0 \ldots 5^0 6^1$</td>
<td>$1^1$</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>$1^1$</td>
</tr>
<tr>
<td>14</td>
<td>$2^2$</td>
<td>$3^30$</td>
</tr>
<tr>
<td>15</td>
<td>$3^7$</td>
<td>$4^88$</td>
</tr>
<tr>
<td>16</td>
<td>$4^{13}$</td>
<td>$5^{64}$</td>
</tr>
<tr>
<td>17</td>
<td>$5^9$</td>
<td>$6^{17}$</td>
</tr>
<tr>
<td>18</td>
<td>$6^5$</td>
<td>$7^1$</td>
</tr>
<tr>
<td>19</td>
<td>$7^1$</td>
<td>$8^0$</td>
</tr>
<tr>
<td>20</td>
<td>$8^1$</td>
<td>$9^0$</td>
</tr>
<tr>
<td>21</td>
<td>$9^0$</td>
<td>$10^0$</td>
</tr>
</tbody>
</table>
An Example: Does a $[21, 14, 6]_4$-code exist?

From [http://codetables.de](http://codetables.de) we determine that $d^\perp \in \{9, 10, 11\}$. The following table gives the number of equivalence classes for $d \geq 6$, distinguished by $d^\perp$:

<table>
<thead>
<tr>
<th>n</th>
<th>$n - k = 6$</th>
<th>$n - k = 7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1$^0$…3$^0$ 4$^2$</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1$^0$…4$^0$ 5$^1$</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1$^0$…5$^0$ 6$^1$</td>
<td>1$^2$ 2$^5$1</td>
</tr>
<tr>
<td>13</td>
<td>1$^1$</td>
<td>2$^6$ 3$^{12}$19</td>
</tr>
<tr>
<td>14</td>
<td>2$^2$</td>
<td>3$^{30}$ 4$^{74}$31</td>
</tr>
<tr>
<td>15</td>
<td>3$^7$</td>
<td>4$^{88}$ 5$^{37}$97</td>
</tr>
<tr>
<td>16</td>
<td>4$^{13}$</td>
<td>5$^{64}$ 6$^{26}$1</td>
</tr>
<tr>
<td>17</td>
<td>5$^9$</td>
<td>6$^{17}$ 7$^4$</td>
</tr>
<tr>
<td>18</td>
<td>6$^5$</td>
<td>7$^1$ 8$^0$</td>
</tr>
<tr>
<td>19</td>
<td>7$^1$</td>
<td>8$^0$ 9$^0$</td>
</tr>
<tr>
<td>20</td>
<td>8$^1$</td>
<td>9$^0$ 10$^0$</td>
</tr>
<tr>
<td>21</td>
<td>9$^0$</td>
<td>10$^0$ 11$^0$</td>
</tr>
</tbody>
</table>
Results

Nonexistence

There are no codes with parameters

- $[35, 10, 13]_2$
- $[22, 8, 10]_3, [24, 14, 7]_3, [28, 21, 5]_3$
- $[19, 8, 9]_4, [21, 14, 6]_4, [22, 16, 5]_4, [27, 17, 8]_4, [30, 21, 7]_4, [39, 27, 9]_4$
- $[16, 5, 10]_5, [16, 6, 9]_5, [17, 8, 8]_5$
- $[15, 8, 7]_7, [26, 20, 6]_7$
- $[30, 23, 7]_8, [37, 31, 5]_8$

and 391 derived new upper bounds.

Existence

There is a $[17, 11, 6]_9$-code.
Results

Nonexistence

There are no codes with parameters

- $[35, 10, 13]_2$
- $[22, 8, 10]_3$, $[24, 14, 7]_3$, $[28, 21, 5]_3$
- $[19, 8, 9]_4$, $[21, 14, 6]_4$, $[22, 16, 5]_4$, $[27, 17, 8]_4$, $[30, 21, 7]_4$, $[39, 27, 9]_4$
- $[16, 5, 10]_5$, $[16, 6, 9]_5$, $[17, 8, 8]_5$
- $[15, 8, 7]_7$, $[26, 20, 6]_7$
- $[30, 23, 7]_8$, $[37, 31, 5]_8$

and 391 derived new upper bounds.

Existence

There is a $[17, 11, 6]_9$-code.
Thank you for your attention.