Countings in a codebook

Hyun Kwang Kim
(joint work with Phan Thahn Toan)

Department of Mathematics, POSTECH

The 5th Sino-Korea conference on Coding Theory and Its related Topics

Shanghai University

July 3, 2018

Outline

Motivation of our study

- In 1973 Delsarte derived n linear inequalities, which are called linear programming bounds, that should be satisfied by all codes of length n.
- The first inequality, which is called Plotkin's bound, was obtained by Plotkin in 1960. And it can be proved by counting the number of $\binom{0}{1}$ or $\binom{1}{0}$ in a 'what we call' codebook of the code in TWO ways.
- Now natural question arises: Can we prove Delsarte's other inequalities by a counting method?
- We will see in a moment that Plotkin's idea can be FAR GENERALIZED, and can be applied to other problems of coding theory in a 'slightly modified' form.

The codebook of a code

- Let C be a binary (n, M) code with distance distribution $\{A_i\}_{i=0}^{n}$.
- Consider C as a codebook, i.e., C is an $(0, 1)$-matrix of size $M \times n$ in which each codeword $c \in C$ is a row.

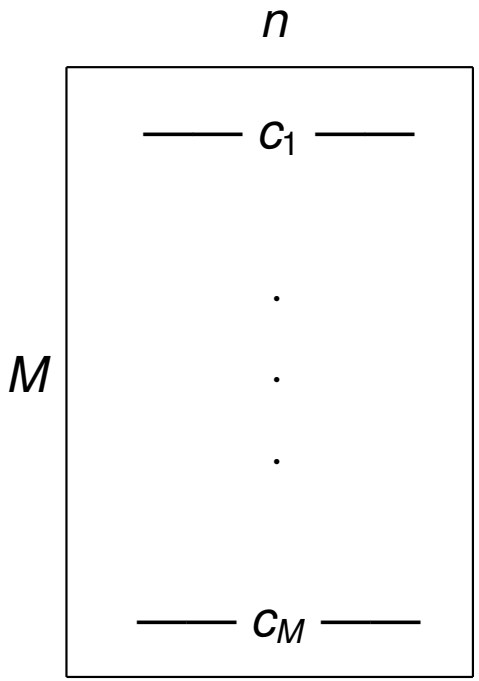

Figure: C as a codebook

Decomposition of Krawtchouk polynomial

Even and odd Krawtchouk polynomial

- We introduce the even (resp. odd) Krawtchouk polynomials by

$$P_k^+(x;n) = \sum_{\substack{j=0 \\ j=even}}^{k} \binom{x}{j}\binom{n-x}{k-j},$$

$$P_k^-(x;n) = \sum_{\substack{j=0 \\ j=odd}}^{k} \binom{x}{j}\binom{n-x}{k-j}.$$

- The ordinary Krawtchouk polynomial which is defined by

$$P_k(x;n) = \sum_{j=0}^{k}(-1)^j\binom{x}{j}\binom{n-x}{k-j}$$

becomes $P_k^+(x;n) - P_k^-(x;n)$ and $P_k^+(x;n) + P_k^-(x;n) = \binom{n}{k}$.

Delsarte's Linear Programming Bounds for binary codes

We state Delsarte's Linear Programming Bounds for binary codes.

Theorem (LP bounds for binary codes)

(a)

$$\sum_{i=1}^{n} P_k^-(i;n)A_i \leq \frac{2M_1}{M}\binom{n}{k}, \tag{1}$$

(b)

$$-\sum_{i=1}^{n} P_k^+(i;n)A_i \leq -\frac{2M_2}{M}\binom{n}{k}, \tag{2}$$

where $M_1 = \begin{cases} \frac{M^2}{4} & \text{if } M \text{ is even,} \\ \frac{M^2-1}{4} & \text{if } M \text{ is odd,} \end{cases}$ and $M_2 = \begin{cases} \frac{M(M-2)}{4} & \text{if } M \text{ is even,} \\ \frac{(M-1)^2}{4} & \text{if } M \text{ is odd.} \end{cases}$

Delsarte's Linear Programming Bounds for binary codes-Cont.

Remark

- Equation (2) $-$ (1) becomes the original Delsarte's linear programming bounds:
  - (2) becomes

$$\frac{2M_2}{M}\binom{n}{k} \leq \sum_{i=1}^{n} P_k^+(i;n)A_i,$$

  - and $-$(1) becomes

$$-\frac{2M_1}{M}\binom{n}{k} \leq -\sum_{i=1}^{n} P_k^-(i;n)A_i.$$

Remark-Cont.

- By adding them, we obtain

$$-\binom{n}{k} \leq \frac{2M_2 - 2M_1}{M}\binom{n}{k} \leq \sum_{i=1}^{n} P_k(i;n)A_i,$$

- and it becomes

$$\sum_{i=0}^{n} P_k(i;n)A_i \geq 0, \ k = 0, 1, \ldots, n,$$

- which is the original Delsarte's linear programming.

- Hence our theorem is "a little" better than the original Delsarte's linear programming.
- Actually it gives that

$$\sum_{i=0}^{n} P_k(i;n)A_i \geq \begin{cases} 0 & \text{if } M \text{ is even,} \\ \frac{1}{M}\binom{n}{k} & \text{if } M \text{ is odd.} \end{cases}$$

Proof of Theorem by codebook counting

- We consider C as a codebook and count the number of $2 \times k$ submatrices of C which has odd number of 1s in TWO ways.
- We begin with row computation:

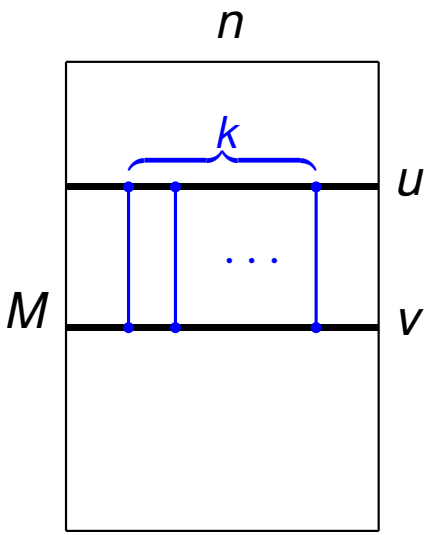

Figure: The contribution from rows u, v

Proof of Theorem-Cont.

- The contribution from two rows u, v becomes

$$\sum_{\substack{j=0 \\ j=odd}}^{k} \binom{d(u,v)}{j} \binom{n-d(u,v)}{k-j}.$$

- Therefore total contribution from the rows becomes

$$\sum_{\substack{u,v \in C \\ u \neq v}} \sum_{\substack{j=0 \\ j=odd}}^{k} \binom{d(u,v)}{j} \binom{n-d(u,v)}{k-j},$$

- and finally, by collecting all pairs with $d(u,v) = i$, it becomes

$$\sum_{i=1}^{n} \sum_{\substack{u,v \in C \\ d(u,v)=i}} P_k^-(i;n) = M \sum_{i=1}^{n} P_k^-(i;n) A_i.$$

Proof of Theorem-Cont.

- We next consider column computation:

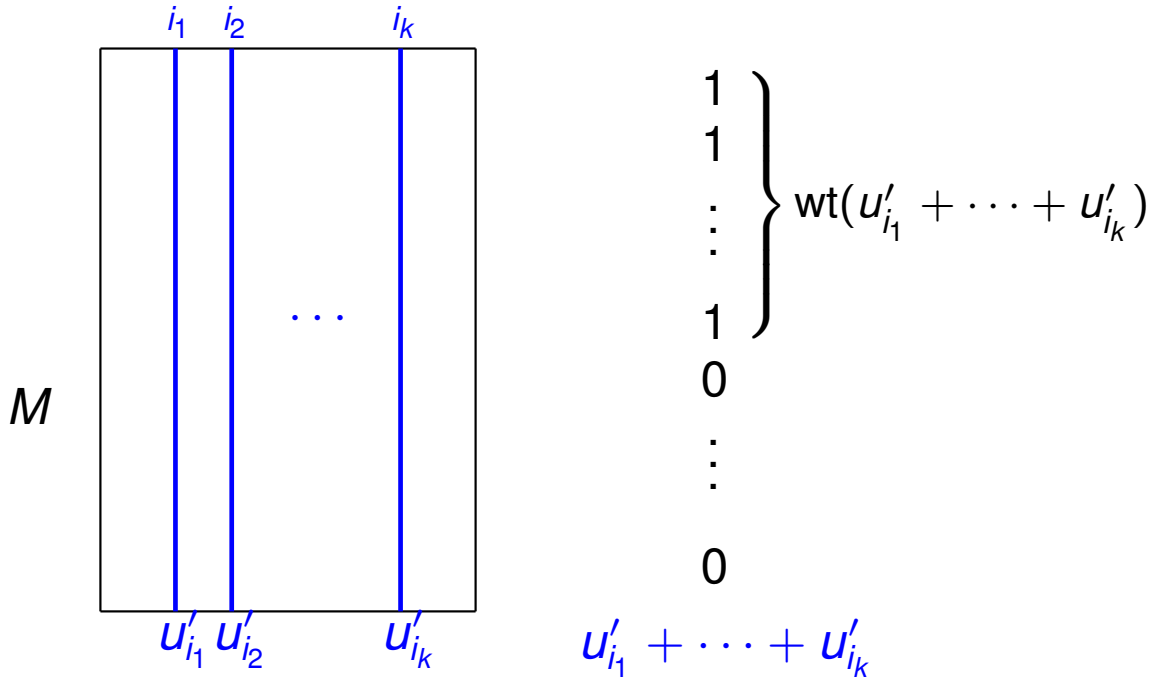

Figure: The contribution from columns $i_1, i_2, \ldots, i_k$

Proof of Theorem-Cont.

- The contribution from columns $i_1, i_2, \ldots, i_k$ becomes

$$2wt(u'_{i_1} + \cdots + u'_{i_k})[M - wt(u'_{i_1} + \cdots + u'_{i_k})].$$

- Therefore the total contribution from columns becomes

$$2 \sum_{i_1 < i_2 < \cdots < i_k} wt(u'_{i_1} + \cdots + u'_{i_k})[M - wt(u'_{i_1} + \cdots + u'_{i_k})] \leq 2M_1 \binom{n}{k}.$$

- Equality holds if and only if $wt(u'_{i_1} + \cdots + u'_{i_k}) = \begin{cases} \frac{M}{2} & \text{if } M \text{ is even,} \\ \frac{(M \pm 1)}{2} & \text{if } M \text{ is odd,} \end{cases}$

  for all $i_1 < i_2 < \cdots < i_k$.

Delsarte's Linear Programming Bounds for q-ary codes

Theorem (Linear Programming Bounds for q-ary codes)

For $k = 1, 2, \cdots, n$, we have

$$\sum_{i=0}^{n} P_k(i; n) A_i \geq 0,$$

where $P_k(i; n)$ is defined by

$$P_k(i; n) = \sum_{j=0}^{k} (-1)^j (q-1)^{i-j} \binom{i}{j} \binom{n-i}{k-j}.$$

Delsarte's Linear Programming Bounds for q-ary codes-Cont.

Remark

- For $k = 0$, $P_0(i; n) = (q-1)^i \geq 0$ for all i. Therefore the inequality also holds for $k = 0$.
- $P_k(i; n)$ is called the q-ary Krawtchouk polynomial. It is known that

$$P_k(i; n) = \sum_{wt(v)=k} \lambda(u \cdot v)$$

where u is a vector of $wt(u) = i$.

- Here $\lambda : \mathbb{F}_q \to \mathbb{S}^1$ is defined as follows: We know $q = p^n$ for some prime p.

$$\lambda(x) = \zeta^{Tr(x)},$$

where ζ is a primitive p-th root of unity, and Tr is the trace map of $GF(q)$ into $GF(p)$.

Notation

Notation

- For $a = (a_1, \cdots, a_j) \in (\mathbb{F}_q^*)^j$, we introduce
- $N(a) = |\{b = (b_1, \cdots, b_j) \in (\mathbb{F}_q^*)^j | b \cdot a \neq 0\}|$,
- $Z(a) = |\{b = (b_1, \cdots, b_j) \in (\mathbb{F}_q^*)^j | b \cdot a = 0\}|$.

Proposition

- $N(a) = \frac{q-1}{q}[(q-1)^j - (-1)^j]$,
- $Z(a) = (q-1)^j - N(a) = \frac{1}{q}(q-1)^j + \frac{q-1}{q}(-1)^j$.
- Notice that these values are independent of the choice of a in $(\mathbb{F}_q^*)^j$.

Target to count

Codebook of a q-ary code

- Let C be a q-ary code of length n with cardinality M.
- Consider C as a codebook, i.e., $C = (c_{mi})$ is an $M \times n$ matrix over $\mathbb{F}_q$.
- Let A be a $2 \times k$ submatrix of C with rows a and b,
- and α an element in $(\mathbb{F}_q^*)^k$.

Our target

- We are going to count the number of pairs (A, α) such that $a \cdot \alpha \neq b \cdot \alpha$ in TWO ways.

Comparison with binary case

Target again

- We are going to count the number of pairs (A, α)
  - where
$$A = \begin{pmatrix} c_{mi_1}, c_{mi_2}, \cdots, c_{mi_k} \\ c_{li_1}, c_{li_2}, \cdots, c_{li_k} \end{pmatrix},$$
    with $m \neq l$ and $i_1 < i_2 < \cdots < i_k$,
  - and $\alpha = (\alpha_1, \cdots, \alpha_k) \in (\mathbb{F}_q^*)^k$
  - such that $\alpha_1 c_{mi_1} + \alpha_2 c_{mi_2} + \cdots \alpha_k c_{mi_k} \neq \alpha_1 c_{li_1} + \alpha_2 c_{li_2} + \cdots \alpha_k c_{li_k}$.

binary case

- If $q = 2$, then α becomes $(1, 1, \cdots, 1)$.
- Therefore, the number of such pairs reduced to the numbers of $2 \times k$ submatrices of C which contain odd number of 1s.

Decomposition of q-ary Krawtchouk polynomial

Even and odd q-ary Krawtchouk polynomial

- We introduce the odd (resp. even) q-ary Krawtchouk polynomials by

$$P_k^-(i;n) = \sum_{j=0}^{k} \frac{[(q-1)^j - (-1)^j]}{2} (q-1)^{k-j} \binom{i}{j} \binom{n-i}{k-j},$$

$$P_k^+(i;n) = (q-1)^k \binom{n}{k} - P_k^-(i;n).$$

- When $q = 2$, it reduced to binary even (resp. odd) Krawtchouk polynomial, namely

$$P_k^-(i;n) = \sum_{\substack{j=0 \\ j=odd}}^{k} \binom{i}{j} \binom{n-i}{k-j},$$

$$P_k^+(i;n) = \sum_{\substack{j=0 \\ j=even}}^{k} \binom{i}{j} \binom{n-i}{k-j}.$$

Decomposition of q-ary Krawtchouk polynomial-Cont.

Proposition

(a) $P_k^+(i;n) + P_k^-(i;n) = \sum_{wt(v)=1} 1 = (q-1)^k \binom{n}{k}$.

(b) $P_k^+(i;n) - P_k^-(i;n) = P_k(n;i)$.

Question

- Can we do a similar job for other 'good' graphs or association schemes?
- We have decomposed 'orthogonal polynomials' for Hamming graphs $H(n,2)$ and $H(n,q)$.

More notation

Notation

- Write $\mathbb{F}_q = \{0 = w_1, w_2, \cdots, w_q\}$.
- For $a = (a_1, \cdots, a_M) \in \mathbb{F}_q^M$, we define

$$\chi_m(a) = |\{j | a_j = w_m\}|.$$

- Thus we trivially have that $\sum_{m=1}^{q} \chi_m(a) = M$.

Definition of $S(k)$

- We are also interested in the sum $S(k)$ defined by $S(k) = \sum_{\alpha \in (\mathbb{F}_q^*)^k} \sum_{\substack{m<l \\ i_1<i_2<\cdots<i_k}} \chi_m(\alpha_1 u'_{i_1} + \cdots + \alpha_k u'_{i_k}) \chi_l(\alpha_1 u'_{i_1} + \cdots + \alpha_k u'_{i_k})$.
- Later we will maximize this term!

Sketch of proof of q-ary LP bound

Sketch

- We count the number $S_1(k)$ of pairs (A, α) such that $a \cdot \alpha \neq b \cdot \alpha$ in TWO ways.
- By row counting, we obtain

$$S_1(k) = \frac{2(q-1)}{q} M \sum_{i=1}^{n} P_k^-(n; i) A_i. \tag{3}$$

- By column counting, we obtain

$$S_1(k) = 2S(k). \tag{4}$$

- It follows from (3), (4) and above proposition that $\sum_{i=1}^{n} P_k^-(i; n) A_i = -(M-1)(q-1)^k nk + \frac{2q}{(q-1)M} S(k)$.
- By maximizing $S(k)$, we deduce that $S(k) \leq (q-1)^k \binom{q}{2} \binom{n}{k} (\frac{M}{q})^2$ and the result follows.

Codebook of a constant weight code

Here we will study the codebook of a constant weight code. Again we begin with binary case.

Codebook of a constant weight code

- Let C be a binary code of length n, cardinality M in which every codeword has constant weight, say w.
- We may consider C as a binary matrix of size $M \times n$ in which every row has w number of 1s.

1-row k-column formula

1-row k-column formula

- We have
$$\sum wt(u'_1 + \cdots + u'_k) = MP_k^-(w; n),$$
where the sum is taken over all distinct k columns $u'_1, \cdots, u'_k$ of C.

- We count the number of $1 \times k$ submatrices of C in two ways.
- Row computation: Each row has w 1s and $n - w$ 0s. Hence the contribution from each row becomes
$$\sum_{\substack{j=0 \\ j=odd}}^{k} \binom{w}{j} \binom{n-w}{k-j} = P_k^-(w; n),$$
and total contribution becomes $MP_k^-(w; n)$.
- Column computation: Take any k columns, say $i_1, \cdots, i_k$ of C. The contribution from these columns becomes $wt(u'_{i_1} + \cdots + u'_{i_k})$. Hence the result.

2-row k-column formula

2-row k-column formula

- We have

$$\sum_{i=\frac{d}{2}}^{n} P_k^-(2i;n)A_{2i} \leq \frac{2}{M}[((\binom{n}{k}) - r_k)q_k(M-q_k) + r_k(q_k+1)(M-q_k-1)],$$

- and we also have

$$-\sum_{i=\frac{d}{2}}^{n} P_k^+(2i;n)A_{2i} \leq \frac{2}{M}[((\binom{n}{k}) - r_k)q_k(M-q_k)]$$
$$+ \frac{2}{M}[r_k(q_k+1)(M-q_k-1)] - (M-1)\binom{n}{k},$$

2-row k-column formula-Cont.

2-row k-column formula

- where q_k and r_k are the quotient and the remainder, respectively, when dividing $MP_k^-(w;n)$ by $\binom{n}{k}$.
- There we can write

$$MP_k^-(w;n) = q_k \binom{n}{k} + r_k$$

with $0 \leq r_k < \binom{n}{k}$.

Proof of 2-row k-column formula

Idea of proof

- We count the number of $2 \times k$ submatrices of C which has odd number of 1s in TWO ways.
- In this process we use the result of 1-row k-column formula intensively.

Sketch of proof

- Row computation: The contribution from rows u, v becomes

$$\sum_{\substack{u,v \in C \\ u \neq v}} \sum_{\substack{j=0 \\ j=odd}}^{k} \binom{d(u,v)}{j} \binom{n-d(u,v)}{k-j}.$$

Proof of 2-row k-column formula-Cont.

Sketch of proof-Cont.

- Therefore the total contribution from the rows becomes

$$\sum_{i=\frac{d}{2}}^{n} \sum_{d(u,v)=2i} P_k^-(2i;n) = M \sum_{i=\frac{d}{2}}^{n} P_k^-(2i;n) A_{2i}.$$

- We next consider column computation:

  - The contribution from columns $i_1, i_2, \ldots, i_k$ becomes

$$2wt(u'_{i_1} + \cdots + u'_{i_k})[M - wt(u'_{i_1} + \cdots + u'_{i_k})].$$

  - Therefore the total contribution from the rows becomes

$$2 \sum_{i_1 < i_2 < \cdots < i_k} wt(u'_{i_1} + \cdots + u'_{i_k})[M - wt(u'_{i_1} + \cdots + u'_{i_k})]$$

Proof of 2-row k-column formula-Cont.

Sketch of proof-Cont.

- We want to maximize the last sum, and consider the distribution of all possible weights
$$wt(u'_{i_1} + \cdots + u'_{i_k})$$
where the k columns $i_1, i_2, \ldots, i_k$ run over all possibilities.
- We conclude that the sum maximized when these weights are 'almost equally' distributed.
- We know from 1-row k-column formula that the sum of possible $\binom{n}{k}$ weights is $MP_k^-(w; n)$.
- Write $MP_k^-(w; n) = q_k\binom{n}{k} + r_k$, with $0 \leq r_k < \binom{n}{k}$. Then the sum become maximum when r_k weights equal to $q_k + 1$, and the remaining $\binom{n}{k} - r_k$ weights equal to q_k.
- We finally conclude that the total contribution from the columns
$$\leq 2[((\binom{n}{k}) - r_k)q_k(M - q_k) + r_k(q_k + 1)(M - q_k - 1)].$$

LP-bound for q-ary constant weight codes

Theorem

- For a q-ary code of length n, cardinality M, constant weight w, and $0 \leq k \leq n$, we have

$$\sum_{i=1}^{n} P_k^-(i;n)A_i \leq \frac{q}{(q-1)M} T(k),$$

- and

$$-\sum_{i=1}^{n} P_k^+(i;n)A_i \leq -(M-1)(q-1)^k \binom{n}{k} + \frac{q}{(q-1)M} T(k).$$

- The value $T(k)$ can be easily computed when q, n, w, M, k are given.

Result

- We apply LP bound for constant weight codes, and improve upper bounds of $A(n, d, w), n \leq 28$ for 22 cases.
- We apply LP bound for q-ary constant weight codes, and proved that $A_3(9, 3, 7) \leq 575$. Previously known best upper bound was $A_3(9, 3, 7) \leq 576$.
- We developed improved semidefinite programming bound for codes, and improve upper bounds of $A(n, d), n \leq 28$ for 2 cases.
- We developed improved semidefinite LP bound for constant weight codes, and improve upper bounds of $A(n, d, w), n \leq 28$ for 23 cases.

Question

- We still don't know what is the counter part of semidefinite programming bound in codebook counting!!

MacWilliams Identity

From now on, we will provide a combinatorial proof of MacWilliams identity from codebook counting. We only consider binary case.

Review of MacWilliams identity

- Let C be a binary $[n, k]$ code with weight distribution $\{A_i\}_{i=0,1,\cdots,n}$. Let $\{B_i\}_{i=0,1,\cdots,n}$ be the weight distribution of its dual code $C^\perp$.
- Let $W_C(x, y)$ (resp. $W_{C^\perp}(x, y)$) be corresponding weight enumerator of C (resp. $C^\perp$).
- MacWilliams identity states that

$$W_C(x, y) = \frac{1}{|C^\perp|} W_{C^\perp}(x + y, x - y).$$

- By definition MacWilliams identity becomes

$$\sum_{i=0}^{n} A_i x^{n-i} y^i = \frac{1}{|C^\perp|} \sum_{i=0}^{n} B_i (x + y)^{n-i} (x - y)^i. \tag{5}$$

Reformulation of MacWilliams Identity

Reformulation of MW

- Putting $x = 1$ in (5) and use the fact that C is an $[n, k]$ code, we obtain

$$\sum_{i=0}^{n} A_i y^i = \frac{1}{2^{n-k}} \sum_{i=0}^{n} B_i (1+y)^{n-i}(1-y)^i.$$

- We set

$$\sum_{i=0}^{n} A_i y^i = \sum_{i=0}^{n} a_i (y-1)^i,$$

- and set

$$\frac{1}{2^{n-k}} \sum_{i=0}^{n} B_i (1+y)^{n-i}(1-y)^i = \sum_{i=0}^{n} b_i (y-1)^i.$$

- Then MacWilliams identity is equivalent to $a_\nu = b_\nu, \nu = 0, 1, \cdots, n$.

Reformulation of MacWilliams Identity-Cont.

Reformulation of MW-Cont.

- By applying the linear operator $\frac{d^\nu}{dy^\nu}|_{y=1}$, we obtain

$$a_\nu = \sum_{i=\nu}^{n} i(i-1)\cdots(i-\nu+1)A_i.$$

- Similarly we obtain

$$b_\nu = 2^{k-\nu}\sum_{i=0}^{\nu}(-1)^i \frac{\nu!}{(\nu-i)!}\frac{(n-i)!}{(n-\nu)!}.$$

- By equating a_ν and b_ν, we finally obtain

$$\sum_{i=0}^{n}\binom{i}{\nu}A_i = 2^{k-\nu}\sum_{i=0}^{n}(-1)^i\binom{n-i}{n-\nu}B_i.$$

MacWilliams Identity from a codebook

- Putting $y = 1$ in (5) and applying a similar method, we finally conclude that
- MacWilliams identity is equivalent to

$$\sum_{i=0}^{n} \binom{i}{\nu} A_i = 2^{k-\nu} \sum_{i=0}^{n} (-1)^i \binom{n-i}{n-\nu} B_i, \nu = 0, 1, \cdots, n,$$

- and to

$$\sum_{i=0}^{n} \binom{n-i}{n-\nu} B_i = 2^{\nu-k} \sum_{i=0}^{n} \binom{n-i}{\nu} A_i, \nu = 0, 1, \cdots, n.$$

- And these identities can be obtained by counting the number of $1 \times \nu$ submatrices $(1, 1, \cdots, 1)$ and $(0, 0, \cdots, 0)$ of the codebook in TWO ways.

References

- Simonis J., de Vroedt C., " A simple proof of the Delsarte inequalities," Designs, Codes and Cryptography, vol. 1(1991), pp. 77 – 82.
- B. G. Kang, H. K. Kim, and P. T. Toan, "Delsarte's linear programming bound for constant-weight codes," IEEE Trans. Inf. Theory, vol. 58(2012), no. 9, pp. 5956 – 5962.
- H. K. Kim and P. T. Toan, "Improved Semidefinite Programming Bound on Sizes of Codes," IEEE Trans. Inf. Theory, vol. 59(2013), no. 11, pp. 7337 – 7345.
- H. K. Kim and P. T. Toan, "New inequalities for q-ary constant-weight codes," Designs, Codes and Cryptography, vol. 73 (2014 November), pp. 369 – 381.

Thank you for your attention!