International Journal of Pure and Applied Mathematics

Volume 16 No. 2 2004, 171-177

MODIFYING A PARITY-CHECK MATRIX TO A SEPARABLE MATRIX

J.Y. Guo[1 §], F.K. Hwang[2]

[1,2]Department of Applied Mathematics
National Chiao Tung University
1001 Ta Hsueh Road, Hsinchu, 300, TAIWAN, R.O.C.
[1]e-mail: davidguo@math.nctu.edu.tw
[2]e-mail: fhwang@math.nctu.edu.tw

Abstract: The $\overline{d}$-separable matrix M and the transpose of the parity check matrix H of an e-error-correcting code satisfy similar requirement, but one is based on Boolean sum, while the other on modulo-2 sum. Consequently, H cannot be used directly as M with $d = e$. Kautz and Singleton [2] gave a method to modify H with $e = 2$. They suggested that the method can be extended to $e = 3$. In this paper, we give such a method for the $\overline{3}$-separable matrix. We also discuss some result for $e = 4$.

AMS Subject Classification: 15A30
Key Words: parity-check, separable, group testing, nonadaptive

1. Introduction

A nonadaptive group testing algorithm, or a pooling design in DNA terminology, can be represented by its incidence matrix $M = (m_{ij})_{t\times n}$ where rows are indexed by tests, columns by test-objects, and $m_{ij} = 1$ if object j is in test i and $m_{ij} = 0$ if not. Among the n objects, some are positive and the other are negative. A test on a subset S of objects induces a positive outcome if S contains a positive object, and a negative outcome if otherwise. Let $U = (u_1, u_2, \ldots, u_t)$ denote the outcome vector, where $u_i = 1$ if test i has a

Received: July 1, 2004

§Correspondence author

$$\begin{pmatrix}
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0
\end{pmatrix}$$

Figure 1: A $\overline{2}$-separable matrix

positive outcome and $u_i = 0$ otherwise. Then U is merely the Boolean sum of the columns representing the positive objects, or simply, the positive columns.

Suppose there are n objects containing exactly d positive ones. Then M can identify the d positive objects if and only if the Boolean sums of all sets of d columns are distinct. Such a matrix is called a d-separable matrix. This can be extended to the case that there are at most d positive objects. Then the requirement on M is that the Boolean sums of all sets of up-to-d columns are distinct. Such a matrix is called $\overline{d}$-separable. Figure 1 gives an example of a $\overline{2}$-separable matrix in which the Boolean sums of all sets S of columns, $1 \leq |S| \leq 2$, are distinct (a few such sums are indicated). While d-separable and $\overline{d}$-separable matrices have been extensively studied (see [1] for a general reference), their existence is still rare in general.

Kautz and Singleton [2] observed that the parity check matrix of an e-error-correcting code has the property that the modulo-2 sums of all sets of up-to-e columns are distinct, the exact requirement for an $\overline{e}$-separable matrix except for the difference between the Boolean sum and the modulo-2 sum. Kautz and Singleton gave a method for $e = 2$ and suggested that $e = 3$ is also doable. In this paper we give the $e = 3$ method. We also give a construction of a 4-separable matrix based on H with $e = 4$.

2. Some General Remarks

Let $H = (h_{ij})$ be the transpose of the parity check matrix of an e-error-correcting code, and $D = \{j \mid \text{object } j \text{ is positive}\}$, $|D| \leq d$, be an implicit set that we want to find. Let R_i be the i-th row of H, and $P_i = \{h_{ij} \mid j \in D\}$ (a multiset of 0s and 1s). $|A|$ denotes the number of the elements of the set A. By the definition of P_i, $|P_i| = |D|$ for all i. $P_i^0(P_i^1)$ is the 0-subset (1-subset)

of P_i. P_j "divides" P_i^0 if P_j contains both 1 and 0 in the bits (columns) of P_i^0. P_i^* is the normal "set" of P_i. V is the outcome vector of H. Naturally, v_i is the ith entry of V. See the following example:

$$H := \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & \cdots \\ 0 & 0 & 1 & 1 & 0 & \cdots \\ 1 & 1 & 1 & 0 & 1 & \cdots \\ \vdots & & & & & \ddots \end{pmatrix}.$$

Suppose $e = 3$ and $D = \{1, 2, 3\}$. Then $P_2 = \{0, 0, 1\}$, $P_2^* = \{0, 1\}$, $P_2^0 = \{0, 0\}$, $P_2^1 = \{1\}$, $P_3 = \{1, 1, 1\}$, $P_3^* = \{1\}$, $P_3^0 = \emptyset$, and $P_3^1 = \{1, 1, 1\}$. $|P_2| = |P_3| = 3$, $|P_2^*| = |P_2^0| = 2$, $|P_3^*| = 1$, and $|P_3^0| = 0$. Moreover, P_1 divides P_2^0, P_3 does not divide P_2^0. No row can divide P_2^1, because $|P_2^1| = 1$.

e-error-correcting code can correct up-to-e errors of the transmitted code word. Let E be the set of error bits in the code word. Then we can use the same E to denote the set of columns in H which we will refer to as error columns. When we use M for group testing, the outcome vector U is the Boolean sum of the error columns (interpreted as positive columns in group testing). Therefore U cannot determine E since the theory of error correcting code is based on V (the modulo-2 sum). We need to reconstruct V from U which can be done only by expanding H to a matrix M with more rows. Then V for H can be reconstructed from U for M.

For $e = 2$, we modify each entry of H by

$$0 \to \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \qquad 1 \to \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

to obtain M. See Kautz and Singleton[2].

For $e = 3$, then this transformation is not enough. For example, $(0, 0, 1)$ and $(0, 1, 1)$ have different parities, but after the transformation

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \text{ and } \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

have the same union $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Kautz and Singleton suggested to convert every pair of rows of H into four rows through the transformation

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \to \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix} \to \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 1 \\ 0 \end{pmatrix} \to \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 1 \\ 1 \end{pmatrix} \to \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

Let P_{ij} be the 4 rows of M that are substituted from the i-th and j-th rows of H. For the former example,

$$P_{12} = \{\begin{pmatrix}0\\0\\1\\0\end{pmatrix}, \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, \begin{pmatrix}0\\0\\0\\1\end{pmatrix}\}.$$

For the sake of convenience, we write $P_{12} = \{0010, 1000, 0001\}$. Then $P_{13} = \{0001, 0100, 0001\}$, and $P^*_{13} = \{0001, 0100\}$. Let U be the outcome vector of M, u_{ij} be the 4 entries of U with respect to P_{ij}. Here, $u_{12} = 1011$, $u_{23} = 0101$.

We quote a property of a parity check matrix.

Lemma 1. *Any two columns of a parity check matrix are distinct.*

3. $\overline{3}$-Separable

Let the function $w(x)$ denote the number of 1's in the set x. In this section, we assume $e = 3$, H is the parity check matrix of 3-error-correcting code, and M is the matrix transformed from H .

Lemma 2. *For any i, j, $w(u_{ij}) \neq 4$.*

Proof. Because of $e = 3$, $|D| \leq 3$. Therefore, For any i, j, $|P_{ij}| = |D| \leq 3$ implies $w(u_{ij}) \leq 3$. □

Lemma 3. *If $w(u_{ij}) = 3$, then we can determine v_i and v_j.*

Proof. $w(u_{ij}) = 3$ means $|P_{ij}| = 3$ and all elements in P_{ij} are unique. Therefore, we can determine P_{ij} by the positions of 1's in u_{ij}. Furthermore, we know $w(P_i)$ and $w(P_j)$. Then v_i is equal to 0 or 1 according to $w(P_i)$ being even or odd. Similar to v_j. □

Lemma 4. *If $|P^*_i| = 2$, then either there exists k such that $w(u_{ik}) = 3$ or $|P_i| = 2$.*

Proof. If $|P_i| \neq 2$, then $|P_i|$ must equal to 3. Without loss of generality, suppose $P_i = \{0, 0, 1\}$ and $D = \{1, 2, 3\}$,

$$R_i \begin{pmatrix} \vdots & & \cdots \\ 0 \quad 0 \quad 1 & \cdots \\ \vdots & & \ddots \end{pmatrix}.$$

If there is no k such that P_k divides P_i^0, then the first two columns of H are the same contrary to Lemma 1. Therefore, there is a k such that $|P_{ik}| = 3$, i.e. $w(u_{ij}) = 3$. □

Lemma 5. *If $w(u_{ij}) = 2$, then we can determine v_i and v_j.*

Proof. If $w(u_{ij}) = 2$, then at least one of $|P_i^*|$ and $|P_j^*|$ is 2. Without loss of generality, suppose $|P_i^*| = 2$. By Lemma 4, if $|P_i| = 2$, then there are only two positives and we can decide v_i and v_j. Else there exists a k such that $w(u_{ik}) = 3$. By Lemma 3, we can decide v_i and v_k. For v_j, if $|P_j^*| = 2$, then, similarly, we can decide v_j. If $|P_j^*| = 1$, then $P_j = \{1, 1, 1\}$ or $P_j = \{0, 0, 0\}$. And we can decide $v_j = 1$ or $v_j = 0$. □

Lemma 6. *If $w(u_{ij}) = 1$, then we can determine v_i and v_j.*

Proof. If $w(u_{ij}) = 1$, then $|P_i^*| = |P_j^*| = 1$. For v_i, suppose there exists a k such that $w(u_{ik}) = 2$. By Lemma 5, we can decide v_i, v_k. Suppose no such k exists, then there is only one positive column. We can easily decide v_i is 1 or 0. For v_j, the argument is similar to v_i. □

Theorem 7. *M is $\overline{3}$-separable.*

Proof. By Lemma 1 to Lemma 5, the outcome vector V of M_p can be inferred from the outcome vector of M_s, thus identify the up-to-3 columns whose modulo-2 sum would have generated V. □

The former arguments sound complicated for decoding because most are involved with determining the number of positives. Actually, as soon as we detect how many positives, 1, 2, or 3, then we can decode very easily.

4. 4-Separable

In this section, we assume there exist exactly 4 positives in the test.

Lemma 8. *If $P_i = \{0, 0, 1, 1\}$ for certain i, then M is 4-separable.*

Proof. We can reconstruct V only by checking u_{ik} for all k. If $w(u_{ik}) = 3$, then $v_k = 1$, else $v_k = 0$ no matter $w(u_{ik}) = 2$ or 4. By the outcome vector V of H, we can find out all 4 positives. □

In the process of decoding, if $|P_i^*| = 1$, so that $P_i = \{0,0,0,0\}$ or $\{1,1,1,1\}$, then it is easy to find out by some $w(u_{ij}) \leq 2$. Furthermore, no matter $P_i = \{0,0,0,0\}$ or $P_i = \{1,1,1,1\}$, $v_i = 0$. Therefore we omit such rows in H. Let P_j' be the ordered set of P_j. Moreover, if $u_{ij} = 1001$ or 0110, then $P_i' = P_j'$ or $P_i' = \overline{P_j'}$. Then we can delete R_j and keep R_i from H, because v_j must equal to v_i. Naturally, if we get rid of R_i from H, we must delete R_{ik} from M for all k.

From now on, these trivial rows are all deleted when we mention M and H.

Lemma 9. *If there is a $P_i = \{0,0,1,1\}$ for some i, then we can identify it.*

Proof. If $P_i = \{0,0,1,1\}$, then we can both find at least one P_j dividing P_i^0 and at least one $P_{j'}$ dividing P_i^1. Note that j' may equal to j. If $P_i \neq \{0,0,1,1\}$, i.e. $P_i = \{0,1,1,1\}$ or $\{0,0,0,1\}$, then no row can divide either P_i^0 or P_i^1 since there is only one 1 or 0 in P_i. Therefore, If there is a $P_i = \{0,0,1,1\}$, we will find it. □

Theorem 10. *M is 4-separable.*

Proof. Once we delete those trivial rows, all of the rest P_i are $\{0,1,1,1\}$, $\{0,0,1,1\}$, and $\{0,0,0,1\}$. If there is $\{0,0,1,1\}$, by Lemma 9, we can find it. Then by Lemma 8, V can be determined. If there is no $P_i = \{0,0,1,1\}$, then $v_i = 1$ for all i. □

If the number of positives is less than 4, then we reduce the solution set to two solutions obtained by assuming 4 positives and 3 positives, respectively. Note that if the number is less than 3, it is easy to find the positive columns.

Theorem 11. *Let D_1, D_2, D_3, and D_4 be the solutions of assuming 4 positives, and C_1, C_2, and C_3 be the solutions of assuming 3 positives. Then $\{C_i\}$ is contained in $\{D_i\}$.*

Proof. Suppose not, let D_1 be not in $\{C_i\}$. Then $(\bigcup C_i) \cup D_1 = (\bigcup D_i)$, contradicting 4-separability. Therefore we can be sure that $\{C_1, C_2, C_3\} = \{D_2, D_3, D_4\}$ are positive regardless of $d = 4$ or 3. □

Acknowledgements

This research is partially supported by a Republic of China National Science Grant NSC 92-2115-M-009-014.

References

[1] D.Z. Du, F.K. Hwang, *Combinatorial Group Testing and its Applications*, 115-116.

[2] W.H. Kautz, R.C. Singleton, Nonrandom binary superimposed codes, *IEEE Transactions on Information Theory*, **10** (1964) 363-377.