Lecture 24

Chapter 3: Codes and designs in association schemes

Throughout this chapter, $\mathcal{X} = (X, \{R_i\}_{i=0}^d)$ is a symmetric association scheme with eigenmatrices P and Q. We work over $\mathbb{R}$. Any scalar that we mention is understood to be in $\mathbb{R}$.

17 Linear programming approach to association schemes

In this section we introduce the linear programming approach. We motivate things with a problem.

Problem 17.1. Let Y denote a subset of X such that no two vertices in Y are R_1-related. How large can Y be?

We now attack the above problem. Recall the standard module $V = \mathbb{R}^X$. Define the vector $\psi_Y \in V$ by

$$\psi_Y = \sum_{y \in Y} \hat{y}.$$

For $0 \leq j \leq d$ the scalar $\|E_j \psi_Y\|^2$ is nonnegative. Let us compute this scalar. We have

$$\begin{aligned}
\|E_j \psi_Y\|^2 &= \left\langle \sum_{y \in Y} E_j \hat{y}, \sum_{z \in Y} E_j \hat{z} \right\rangle \\
&= \sum_{y \in Y} \sum_{z \in Y} \langle E_j \hat{y}, E_j \hat{z} \rangle \\
&= \frac{|Y|}{|X|} \sum_{i=0}^{d} a_i Q_j(i),
\end{aligned}$$

where

$$a_i = \frac{|(Y \times Y) \cap R_i|}{|Y|} \qquad (0 \leq i \leq d).$$

Of course $a_i \geq 0$ for $0 \leq i \leq d$. Moreover

$$a_0 = 1, \qquad a_1 = 0, \qquad |Y| = \sum_{i=0}^{d} a_i.$$

We can gain insight about Problem 17.1 by solving the following linear programming problem.

Problem 17.2. Maximize

$$g = \sum_{i=0}^{d} a_i$$

subject to the following constraints:

(i) $a_0 = 1$ and $a_1 = 0$ and $a_i \geq 0$ for $2 \leq i \leq d$;

(ii) $\sum_{i=0}^{d} a_i Q_j(i) \geq 0$ for $0 \leq j \leq d$.

Problems 17.1, 17.2 are related as follows. Let g_0 denote the maximal value of g in Problem 17.2. Then $|Y| \leq g_0$ for all subsets Y from Problem 17.1.

Example 17.3. Assume the association scheme $\mathcal{X}$ is the 3-cube $H(3,2)$. We can see at a glance that for Problem 17.1, the answer is 4. Let us find g_0. We have

$$Q = \begin{pmatrix} 1 & 3 & 3 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -3 & 3 & -1 \end{pmatrix}$$

We maximize $g = \sum_{i=0}^{3} a_i$ subject to

$$\begin{array}{llll} a_0 = 1, & a_1 = 0, & a_2 \geq 0, & a_3 \geq 0, \\ a_2 + 3a_3 \leq 3, & a_2 - 3a_3 \leq 3, & a_3 - a_2 \leq 1. & \end{array}$$

A graph of the inequalities reveals that g is maximized at $(a_2, a_3) = (3, 0)$. Therefore $g_0 = 1 + 0 + 3 + 0 = 4$.

We have some comments about Problem 17.2. Recall that $Q_0(i) = 1$ for $0 \leq i \leq d$. So in part (ii), the case $j = 0$ provides no information, and can be ignored. Since $a_1 = 0$ we can remove a_1 from the entire problem. Concerning part (i), sometimes it is convenient to drop the requirement that $a_0 = 1$. In this case, Problem 17.2 has type (C, M) below.

Fix an integer $d \geq 1$ and define the set $D = \{0, 1, \dots, d\}$. Define a subset $M \subseteq D$ such that $0 \in M$. Define $D^\times = D \backslash \{0\}$ and $M^\times = M \backslash \{0\}$. Pick $C \in \mathrm{Mat}_{d+1}(\mathbb{R})$, with (i,j)-entry denoted $C_j(i)$ for $0 \leq i, j \leq d$. Assume that $C_0(i) = 1$ for $0 \leq i \leq d$.

Problem (C, M): Maximize

$$g = \sum_{i \in M} a_i C_0(i)$$

subject to

$$a_i \geq 0 \quad (i \in M^\times), \qquad \sum_{i \in M} a_i C_j(i) \geq 0 \quad (j \in D^\times). \tag{76}$$

Definition 17.4. A vector $\{a_i\}_{i\in M}$ is called a *program* for (C, M) whenever it satisfies (76) and $a_0 = 1$. A program $\{a_i\}_{i\in M}$ for (C, M) is called *maximal* whenever it gives the maximal value of g. Problem (C, M) is called *feasible* whenever there exists a program for (C, M).

The following problem is related to Problem (C, M).

Problem $(C, M)'$: Minimize

$$\gamma = \sum_{j\in D} \alpha_j C_j(0)$$

subject to

$$\alpha_j \geq 0 \quad (j \in D^\times), \qquad \sum_{j\in D} \alpha_j C_j(i) \leq 0 \quad (i \in M^\times). \tag{77}$$

Definition 17.5. A vector $\{\alpha_j\}_{j\in D}$ is called a *program* for $(C, M)'$ whenever it satisfies (77) and $\alpha_0 = 1$. A program $\{\alpha_j\}_{j\in D}$ for $(C, M)'$ is called *minimal* whenever it gives the minimal value of γ. Problem $(C, M)'$ is called *feasible* whenever there exists a program for $(C, M)'$.

Problem (C, M) and Problem $(C, M)'$ are related as follows.

Lemma 17.6. *Let $\{a_i\}_{i\in M}$ and $\{\alpha_j\}_{j\in D}$ denote programs for (C, M) and $(C, M)'$ respectively. Then $g \leq \gamma$.*

Proof. We show that

$$g \leq \sum_{i\in M}\sum_{j\in D} a_i\alpha_j C_j(i) \leq \gamma.$$

We have

$$\begin{aligned}
g &= \alpha_0 g \\
&= \alpha_0 \sum_{i\in M} a_i C_0(i) \\
&\leq \alpha_0 \sum_{i\in M} a_i C_0(i) + \sum_{j\in D^\times} \alpha_j \left(\sum_{i\in M} a_i C_j(i)\right) \\
&= \sum_{j\in D}\sum_{i\in M} \alpha_j a_i C_j(i) \\
&= \sum_{i\in M}\sum_{j\in D} a_i \alpha_j C_j(i).
\end{aligned}$$

We also have

$$\begin{aligned}
\gamma &= a_0 \gamma \\
&= a_0 \sum_{j\in D} \alpha_j C_j(0) \\
&\geq a_0 \sum_{j\in D} \alpha_j C_j(0) + \sum_{i\in M^\times} a_i \left(\sum_{j\in D} \alpha_j C_j(i)\right) \\
&= \sum_{i\in M}\sum_{j\in D} a_i \alpha_j C_j(i).
\end{aligned}$$

□

We now state the duality theorem for linear programming.

Theorem 17.7. *Assume that Problems (C, M) and $(C, M)'$ are feasible. Then there exists a program $\{a_i\}_{i\in M}$ for (C, M) and a program $\{\alpha_j\}_{j\in D}$ for $(C, M)'$ such that $g = \gamma$. Moreoever $g_0 = g = \gamma = \gamma_0$.*

The proof of Theorem 17.7 can be found in the textbook, pages 110–112.

Next, we consider how to find the programs $\{a_i\}_{i\in M}$ and $\{\alpha_j\}_{j\in D}$ in Theorem 17.7.

Lemma 17.8. *Given a program $\{a_i\}_{i\in M}$ for (C, M) and a program $\{\alpha_j\}_{j\in D}$ for $(C, M)'$. Then $g = \gamma$ if and only if the following* (i), (ii) *hold:*

(i) *for $i \in M^\times$,*

$$a_i \sum_{j\in D} \alpha_j C_j(i) = 0.$$

(ii) *for $j \in D^\times$,*

$$\alpha_j \sum_{i\in M} a_i C_j(i) = 0.$$

Proof. Immediate from the proof of Lemma 17.6. □

Example 17.9. For even $d = 2t$, the *orthogonality graph* Ω_d has the same vertex set as the hypercube $H(d, 2)$; vertices y, z are adjacent in Ω_d whenever $(y, z) \in R_t$ in $H(d, 2)$. A set of vertices Y for Ω_d is called independent whenever no two vertices in Y are adjacent in Ω_d. Our problem is to find the maximal size of an independent set in Ω_d. First assume that t is odd. Recall that $H(d, 2)$ is bipartite, and note that either half of the bipartition is an independent set in Ω_d. This independent set has cardinality 2^{d-1}, which is maximal. Next assume that t is even. In this case, the problem is open. The above linear programming technique gives an upper bound of $g_0 = 2^n/n$ for the size of an independent subset. Thus for $H(4, 2)$ we have $g_0 = 16/4 = 4$. For $H(4, 2)$ the linear programming details are shown below. We have $d = 4$. We have $D = \{0, 1, 2, 3, 4\}$ and $M = \{0, 1, 3, 4\}$. We take $C = Q$ where

$$Q = \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 1 & 2 & 0 & -2 & -1 \\ 1 & 0 & -2 & 0 & 1 \\ 1 & -2 & 0 & 2 & -1 \\ 1 & -4 & 6 & -4 & 1 \end{pmatrix}.$$

Problem (C, M): Maximize

$$g = a_0 + a_1 + a_3 + a_4$$