Cliques in Association Schemes

Andrew Nagarajah
Supervised by: Prof. Mike Newman

23 March 2021

Abstract

This report is an exploration of some of the topics within *Delsarte theory*, which uses ideas from graph theory, algebra, and optimization to address questions in coding theory in particular, and combinatorics more broadly. The centre of study is the *association scheme*, which provides a setting in which to view various objects, especially *distance-regular graphs*. This perspective enables the computation of various parameters of interest, including the eigenvalues of graphs and upper bounds on codes, cliques, and independent sets.

Contents

1. Introduction

1.1 Coding Theory

I'm also not sure if I should include a section like this, but seeing as coding theory was the original motivation behind Delsarte's LP bound, and it remains (presumably?) a strong motivator for this theory, I figured it might be interesting to mention this as an application.

- Basics of Coding Theory
- Linear Graphs
- Finite Vector Spaces

1.2 Hamming Graphs

1.3 Distance-Regular Graphs

I'm not sure if I should maybe merge this section with P-polynomial section?

- Definition
- Basic parameters
- Examples?

2. Association Schemes

2.1 Association Schemes

- Definition(s)
- Examples?
- Basic parameters

2.1.1 The Bose-Mesner Algebra

2.1.2 Duality

2.2 P-Polynomial Schemes

- Definitions
- "Equivalence" to DRGs

If we want to focus on the LP bound and translation schemes, I'm not sure that this section is necessary, but it is very interesting and provides an important class of examples.

2.2.1 Q-Polynomial Schemes

2.3 Automorphisms

This section may be merged with the following section.

- Action of a regular group of automorphisms
- Cayley graphs
- Eigenspaces from characters

2.4 Translation Schemes

- Equitable partitions of matrices
- Group partitions yielding association schemes
- Dual schemes? (Interesting, but not particularly necessary for the rest of this report)

2.5 The Eigenvalues of the Hamming Scheme

3. Delsarte's Linear Programming Bound

3.1 Linear Programming

- Basics of Linear Programming – done
- Duality – done
- Algorithms?

The terminology and results from this section, except for the adjective *principal* for constraints, follows from [4].

A LINEAR PROGRAMMING PROBLEM (or LINEAR PROGRAM) is an optimization problem in which one seeks to maximise or minimize a linear function of one or more variables, subject to linear constraints. That is, fixing a vector c, one tries to maximize or minimize the linear combinations of the components of c:

$$c_1 x_1 + \cdots + c_n x_n = c^T x$$

for some x. Note that maximizing $c^T x$ is equivalent to minimizing $(-c)^T x$, so that for the theory of linear programming, it suffices to consider maximization problems without loss of generality. As in other optimization problems, the function to be maximized ($c^T x$ in this case) is called the OBJECTIVE (FUNCTION).

In most cases, there will be contraints on the inputs to the objective function, and for the purposes of linear programming these will also have to be linear. That is, there will be a matrix A and vector b such that only inputs x satisfying $Ax \leq b$ will be allowed. (Note that for *vectors* a and b, $a \leq b$ will mean that each component a_i is less than or equal to the corresponding component b_i.) These are called the (PRINCIPAL) CONSTRAINTS, and vectors x which satisfy the constraints will be called FEASIBLE (SOLUTIONS). (Note that in [1], the term *program* is used to refer to a feasible solution.)

If there are no constraints on the problem (and even in some cases where there are) through appropriate choices of feasible solution x, the objective $c^T x$ may be made arbitrarily large, and such problems are called UNBOUNDED. Conversely, if no feasible solutions exist, then the problem is called INFEASIBLE.

Finally, in most applications of linear programming – in particular to the cliques of association schemes – the feasible solutions will be further constrained to those with all non-negative components (i.e. $x \geq 0$). These are called the NON-NEGATIVITY CONSTRAINTS, in constrast with the *principal constraints*. The non-negativity constraints will be required throughout the remainder of this report.

Therefore, for an objective $c^T x$ and constraints $Ax \leq b$, the associated linear program will be written in STANDARD FORM:

$$\max \left\{ c^T x \;\middle|\; Ax \leq b,\ x \geq 0 \right\} .$$

3.1.1 Duality

The most important observation about linear programs (for the purposes of this report, at least) is that they come in dual pairs.

Given a linear program $\mathcal{P}$ written in standard form

$$\max \left\{ c^T x \;\middle|\; Ax \leq b,\ x \geq 0 \right\}$$

its DUAL program is $\mathcal{P}^*$:

$$\min \left\{ b^T y \;\middle|\; A^T y \geq c,\ y \geq 0 \right\} .$$

Re-writing it in standard form,

$$\max \left\{ (-b)^T y \;\middle|\; -A^T y \leq -c,\ y \geq 0 \right\}$$

taking the dual

$$\min \left\{ -c^T x \;\middle|\; -Ax \geq -b,\ x \geq 0 \right\}$$

and re-writing in standard form

$$\max \left\{ c^T x \;\middle|\; Ax \leq b,\ x \geq 0 \right\}$$

the original (called PRIMAL) linear program is recovered.

This demonstrates that $(\mathcal{P}^*)^* = \mathcal{P}$, so that linear programs come in dual pairs.

Theorem 1 (Weak Duality)
If x is a feasible solution to a linear program

$$\max \left\{ c^T x \;\middle|\; Ax \leq b,\ x \geq 0 \right\} ,$$

and y is a feasible solution to its dual program,

$$\min\left\{b^T y \;\middle|\; A^T y \geq c,\; y \geq 0\right\},$$

then $c^T x \leq b^T y$.

Proof. Let u, v, w be vectors with $u \geq 0$, and $v \leq w$. Then for all components i, $u_i \geq 0$ and $v_i \leq w_i$ implies that $u_i v_i \leq u_i w_i$ so that

$$u^T v = \sum_i u_i v_i \leq \sum_i u_i w_i = u^T w \; .$$

In particular, since y is a feasible solution to the dual program, and $x \geq 0$,

$$c \leq A^T y \implies x^T c \leq x^T A^T y = y^T A x \; .$$

(Here one may take the transpose of the whole expression, since the result is a scalar.) Similarly, since x is a feasible solution to the primal program, and $y \geq 0$,

$$b \geq Ax \implies y^T b \geq y^T A x \; .$$

By combining the two inequalities,

$$b^T y = y^T b \geq y^T A x \geq x^T c = c^T x$$

which is the desired result. □

As a result of the weak duality of linear programs, every feasible solution to the dual program provides an upper bound on the maximum of the primal, and every feasible solution to the primal program provides a lower bound on the minimum of the dual.

In fact the extremal values of dual programs (the maximum of the primal, and the minimum of the dual) coincide, although this will not be needed for the purposes of this report. This is referred to as *Strong Duality* of linear programs.

3.2 The LP Bound

Definition 1
The INNER DISTRIBUTION *is TODO. I might also put this in the section on association schemes; I'm not sure if it belongs better there or here.*

Theorem 2 (Delsarte Thm 3.3 [1])

For any inner distribution y,

$$Q^T y \geq 0$$

where Q is the matrix of dual eigenvalues. (Here, $x \geq 0$ means that each component of the vector x is not less than 0.)

Proof. TODO. Note that this will require a number of lemmas which I've omitted here for brevity, but will include in the final product. $\square$

This theorem provides the key inequality that will allow the application of linear programming to cliques in association schemes. However, because the constraint vector in a primal linear program becomes the objective in the dual program, this inequality will require some transformation to make it suitable for use in linear programming.

Let Y be an M-clique with inner distribution y. Then $y_i = 0$ for all $i \notin M$, so $Q^T y \geq 0 \iff Q^T \operatorname{diag}(\chi_M)\, y \geq 0$ since the action of $\operatorname{diag}(\chi_M)$ acting on the left is to zero out the *rows* of y with index not in M. Similarly,

$$Q^T \operatorname{diag}(\chi_M)\, y = Q(0)^T y_0 + Q^T \operatorname{diag}(\chi_{M^*})\, y = \mu + Q^T \operatorname{diag}(\chi_{M^*})\, y$$

since the action of $\operatorname{diag}(\chi_{M^*})$ on the right is to zero out the *columns* of Q^T with index not in M^*, $y_0 = 1$, and

$$Q^T = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \mu_1 & & & \\ \vdots & & * & \\ \mu_d & & & \end{bmatrix} .$$

Finally, since $y \geq 0$, $Q_0^T y \geq 0$ adds no new constraint, so that under the non-negativity constraint $Q^T y \geq 0 \iff \operatorname{diag}(\chi_{N^*})\, Q^T y \geq 0$.

Putting all this together, $Q^T y \geq 0 \iff \operatorname{diag}(\chi_{N^*})\, Q^T \operatorname{diag}(\chi_{M^*})\, y \geq -\mu$ so that Delsarte's LP can be written in standard form:

$$\max\left\{\mathbf{1}^T \operatorname{diag}(\chi_M)\, y \;\middle|\; Q^T \operatorname{diag}(\chi_M)\, y \geq 0,\ y \geq 0,\ y_0 = 1\right\} \tag{3.1}$$
$$= \max\left\{\chi_{M^*}^T y \;\middle|\; -\operatorname{diag}(\chi_{N^*})\, Q^T \operatorname{diag}(\chi_{M^*})\, y \leq \operatorname{diag}(\chi_{N^*})\, \mu,\ y \geq 0\right\} + 1\ . \tag{3.2}$$

Taking the dual yields

$$\min\left\{\mu^T \operatorname{diag}(\chi_{N^*})\, z \;\middle|\; -\operatorname{diag}(\chi_{M^*})\, Q \operatorname{diag}(\chi_{N^*})\, z \geq \chi_{M^*},\ z \geq 0\right\} + 1 \tag{3.3}$$
$$= \min\left\{\mu^T z \;\middle|\; -\operatorname{diag}(\chi_{M^*})\, Q \operatorname{diag}(\chi_{N^*})\, z \geq \chi_{M^*},\ z \geq 0,\ z_0 = 1\right\}\ . \tag{3.4}$$

Therefore, if $z_0 = 1$ is required, recalling that $Q_0 = \mathbf{1}$ and $\operatorname{diag}(\chi_{M^*})\,\mathbf{1} = \chi_{M^*}$, then

$$\begin{aligned}
&\operatorname{diag}(\chi_{M^*})\, Q \operatorname{diag}(\chi_N)\, z \\
=&\operatorname{diag}(\chi_{M^*})\,(Q_0 z_0 + Q \operatorname{diag}(\chi_{N^*})\, z) \\
=&\chi_{M^*} + \operatorname{diag}(\chi_{M^*})\, Q \operatorname{diag}(\chi_{N^*})\, z \\
\leq&0\ .
\end{aligned}$$

This equivalence recover's Delsarte's formulation of the dual linear program:

$$\min\left\{\mu^T z \;\middle|\; \operatorname{diag}(\chi_{M^*})\, Qz \leq 0,\ z \geq 0,\ z_0 = 1\right\}\ . \tag{3.5}$$

3.3 The Ratio Bound

We will use (3.5) frequently.

3.4 The Clique-Coclique Bound

4. Schrijver's SDP Bound

4.1 The Terwilliger Algebra of the Hamming Scheme

4.2 Semi-Definite Programming

5. Computation

I wasn't sure if I ought to mention anything about the code I've written for this project (or even if there's anything worth saying that won't be covered elsewhere in the report).

Also, if there are some specific results that would be interesting to show, but do not fit naturally into other sections of the report, then perhaps they could go here as well.

A. Linear Algebra

A.1 The Spectral Theorem

A.2 Adjacency Matrices

Basic results about the spectra of adjacency matrices, which may be used elsewhere in the report. E.g. the sum of eigenvalues with multiplicity, and consequences.

A.3 Positive Semi-Definite Matrices

Depending on which proof of the clique-coclique bound I use, and how much detail I go into Schrijver's SDP bound, I could make some comments about PSD matrices.

B. Group Theory

B.1 Group Actions

The material of this section comes primarily from [2, Section 1.7, Chapter 4].

Definition 2 (Group Action)
Given a group G and a set X, GROUP ACTION *is a homomorphism $G \to \operatorname{Sym} X$, where $\operatorname{Sym} X$ is the symmetric group on X.*

A group action $\varphi : G \to \operatorname{Sym} X$ induces a product $X \times G \to X$ by mapping $(x, g) \mapsto \varphi(g)(x)$. When the action is clear from context, this will be denoted $x \cdot g$, or simply xg. This is called a RIGHT ACTION, as *g acts on the right of x* (the corresponding notion of a LEFT ACTION can also be defined.)

Conversely, given a product $X \times G \to X$, the same expression defines a map $G \to \operatorname{Sym} X$. If such a product satisfies

$$\begin{gathered} \forall x \in X \; x1_G = x \\ \text{and } \forall x \in X \; \forall g, h \in G \; (xg)h = x(hg) \end{gathered}$$

then the induced map $G \to \operatorname{Sym} X$ will be a homomorphism, so that these definitions are equivalent.

(In [2] this is taken as the definition of a group action, and the homomorphism $G \to \operatorname{Sym} X$ is called its PERMUTATION REPRESENTATION. It will be occasionally convenient to adopt each perspective.)

Definition 3 (Types of Group Actions)
If if a homomorphism $G \to \operatorname{Sym} X$ is injective, then the action is called FAITHFUL. *Note that a group homomorphism is injective if and only if it has a trivial kernel.*

Given a group action $G \to \operatorname{Sym} X$, $g \in G$ is called FIXED POINT-FREE *if $\forall x \in X \; xg \neq x$. The group action itself is called* FIXED POINT-FREE *(or just* FREE*) if all its nontrivial elements are fixed point-free.*

A group action $G \to \operatorname{Sym} X$ is called TRANSITIVE *if $\forall x, y \in X$ there exists some $g \in G$ such that $xg = y$.*

A group action is called REGULAR *if it is simultaneously transitive and free. (This terminology follows [3].)*

Note that if X is a structure with automorphisms (such as a graph or group), G is a subgroup of $\operatorname{Aut} X$, and G acts in the natural way on X (i.e. $xg = g(x)$), then this action is faithful. That is, $\operatorname{Aut} X \leq \operatorname{Sym} X$, so that this action is induced by the identity $G \hookrightarrow \operatorname{Sym} X$, which is clearly injective.

Lemma 1

If an abelian group G acts faithfully and transitively on a set X, then the action is free, and thus also regular. [2, Section 4.1, Exercise 3]

Proof. Let $g \in G$ be nontrivial, and $x \in X$. The goal is to prove that $xg \neq x$.

Since g is not the identity, there exists some $y \in X$ such that $z := yg \neq y$. Furthermore, since G acts transitively on X, there exists some $h \in G$ such that $yh = x \iff y = xh^{-1}$. Then,

$$\begin{aligned} xg &= (yh)g \\ &= y(gh) \\ &= y(hg) && \text{since } G \text{ is abelian} \\ &= (yg)h \\ &= zh\ . \end{aligned}$$

If $zh = x$ then, $z = xh^{-1} = y$, but by definition, $z = yg \neq y$, so $xg = zh \neq x$. □

B.2 Character Theory

Definition, and basic results used.

C. Notation

I've included these sections mostly as an excuse to add the citations to the bibliography, though it may be somewhat useful to have a sort of guide to the similarities and differences in notation in this report, as well as in the references. (At least, I would find such a thing useful.)

C.1 Godsil

See [3] Ch. 10.

C.2 Delsarte

See [1] Ch. 3.

C.3 Schrijver

See [5].

Bibliography

[1] P. Delsarte. “An algebraic approach to the association schemes of coding theory”. PhD thesis. 1973.

[2] David Steven. Dummit. *Abstract algebra*. eng. 3rd ed. New York: Wiley, 2004. ISBN: 0471433349.

[3] C. Godsil. *Algebraic Combinatorics*. Chapman Hall/CRC Mathematics Series. Taylor & Francis, 1993. ISBN: 9780412041310.

[4] Jiří Matoušek. *Understanding and using linear programming*. eng. Universitext. Berlin ; Springer, 2007.

[5] Alexander Schrijver. “New Code Upper Bounds From the Terwilliger Algebra and Semidefinite Programming”. In: *Information Theory, IEEE Transactions on* 51 (Sept. 2005), pp. 2859–2866. DOI: 10.1109/TIT.2005.851748.