

# Semidefinite Optimization

Mastermath  
Spring 2012

Monique Laurent

Centrum Wiskunde & Informatica  
Science Park 123  
1098 XG Amsterdam  
The Netherlands  
`monique@cw.nl`

Frank Vallentin

Delft Institute of Applied Mathematics  
Technical University of Delft  
P.O. Box 5031  
2600 GA Delft  
The Netherlands  
`f.vallentin@tudelft.nl`

March 28, 2012

---

# CONTENTS

<b>1</b>	<b>Background: Convex sets and positive semidefinite matrices</b>	<b>1</b>
1.1	Some fundamental notions . . . . .	2
1.1.1	Euclidean space . . . . .	2
1.1.2	Topology in finite-dimensional metric spaces . . . . .	2
1.1.3	Affine geometry . . . . .	3
1.2	Convex sets . . . . .	4
1.3	Implicit description of convex sets . . . . .	5
1.3.1	Metric projection . . . . .	6
1.3.2	Separating and supporting hyperplanes . . . . .	8
1.4	Explicit description of convex sets . . . . .	11
1.5	Convex cones . . . . .	12
1.6	Examples . . . . .	13
1.6.1	The non-negative orthant and linear programming . . . . .	14
1.6.2	The second-order cone . . . . .	14
1.6.3	The cone of semidefinite matrices . . . . .	14
1.6.4	The copositive cone . . . . .	15
1.7	Positive semidefinite matrices . . . . .	15
1.7.1	Basic facts . . . . .	15
1.7.2	The trace inner product . . . . .	16
1.7.3	Hoffman-Wielandt inequality . . . . .	17
1.7.4	Schur complements . . . . .	18
1.7.5	Block-diagonal matrices . . . . .	19
1.7.6	Kronecker and Hadamard products . . . . .	20
1.8	Historical remarks . . . . .	20
1.9	Further reading . . . . .	22
1.10	Exercises . . . . .	22

<b>2</b>	<b>Semidefinite programs: Basic facts and examples</b>	<b>25</b>
2.1	Primal and dual semidefinite programs . . . . .	26
2.1.1	Primal form . . . . .	26
2.1.2	Dual form . . . . .	27
2.2	Eigenvalue optimization . . . . .	28
2.3	Convex quadratic constraints . . . . .	31
2.4	Robust optimization . . . . .	32
2.5	Examples in combinatorial optimization . . . . .	34
2.5.1	The maximum independent set problem . . . . .	34
2.5.2	The maximum cut problem . . . . .	35
2.6	Examples in geometry . . . . .	36
2.7	Examples in algebra . . . . .	37
2.8	Further reading . . . . .	38
2.9	Exercises . . . . .	39
<b>3</b>	<b>Duality in conic programming</b>	<b>42</b>
3.1	Fundamental properties . . . . .	44
3.1.1	Local minimizers are global minimizers . . . . .	44
3.1.2	Karush-Kuhn-Tucker condition . . . . .	44
3.2	Primal and dual conic programs . . . . .	45
3.2.1	Primal conic programs . . . . .	45
3.2.2	Dual conic programs . . . . .	45
3.2.3	Geometric interpretation of the primal-dual pair . . . . .	46
3.3	Examples . . . . .	47
3.3.1	Linear programming (LP) . . . . .	47
3.3.2	Conic quadratic programming (CQP) . . . . .	48
3.3.3	Semidefinite programming (SDP) . . . . .	48
3.4	Duality theory . . . . .	49
3.5	Some pathological examples . . . . .	52
3.5.1	Dual infimum not attained . . . . .	52
3.5.2	Positive duality gap . . . . .	52
3.6	Strong and weak infeasibility . . . . .	53
3.7	More on the difference between linear and conic programming . . . . .	56
3.8	Further reading . . . . .	56
3.9	Historical remarks . . . . .	57
3.10	Exercises . . . . .	57
<b>4</b>	<b>Interior point methods</b>	<b>60</b>
4.1	Classical barrier methods . . . . .	61
4.1.1	Newton's method . . . . .	61
4.1.2	Barrier method . . . . .	65
4.1.3	Finding a starting point . . . . .	66
4.2	Central path of a semidefinite program . . . . .	67
4.3	Software . . . . .	69
4.4	Historical remarks . . . . .	70
4.5	Further reading . . . . .	71

<b>5</b>	<b>0/1 optimization</b>	<b>74</b>
5.1	Relaxations using quadratic optimization . . . . .	75
5.2	A hierarchy of semidefinite programs . . . . .	78
5.2.1	Harmonic analysis on power sets . . . . .	79
5.2.2	Lasserre's hierarchy . . . . .	82
5.2.3	Example: Independence number . . . . .	86
5.3	Further reading . . . . .	87
5.4	Exercises . . . . .	88
<b>6</b>	<b>Graph coloring and independent sets</b>	<b>90</b>
6.1	Preliminaries on graphs . . . . .	90
6.1.1	Stability and chromatic numbers . . . . .	90
6.1.2	Perfect graphs . . . . .	91
6.2	Linear programming bounds . . . . .	92
6.3	Semidefinite programming bounds . . . . .	94
6.3.1	The theta number . . . . .	94
6.3.2	Computing maximum stable sets in perfect graphs . . . . .	95
6.3.3	Minimum colorings of perfect graphs . . . . .	96
6.4	Other formulations of the theta number . . . . .	97
6.4.1	Dual formulation . . . . .	97
6.4.2	Two more (lifted) formulations . . . . .	98
6.5	Geometric properties of the theta number . . . . .	99
6.5.1	Orthonormal representations . . . . .	99
6.5.2	The theta body $\text{TH}(G)$ . . . . .	100
6.5.3	More on the theta body . . . . .	101
6.6	The theta number for vertex-transitive graphs . . . . .	102
6.7	Bounding the Shannon capacity . . . . .	104
6.8	Further reading . . . . .	106
6.9	Exercises . . . . .	106
<b>7</b>	<b>Approximating MAX CUT and the cut norm</b>	<b>109</b>
7.1	The algorithm of Goemans and Williamson . . . . .	110
7.1.1	Semidefinite relaxation . . . . .	110
7.1.2	Analysis of the algorithm . . . . .	112
7.1.3	Remarks on the algorithm . . . . .	115
7.2	Cut norm and Grothendieck's inequality . . . . .	115
7.2.1	Cut norm of a matrix . . . . .	115
7.2.2	Grothendieck's inequality . . . . .	117
7.2.3	Proof of Grothendieck's inequality . . . . .	118
7.3	Further reading . . . . .	120
7.4	Historical remarks and anecdotes . . . . .	120
7.5	Questions . . . . .	121

<b>8</b>	<b>Generalizations of Grothendieck's inequality and applications</b>	<b>124</b>
8.1	The Grothendieck constant of a graph . . . . .	126
8.1.1	Randomized rounding by truncating . . . . .	127
8.1.2	Quality of expected solution . . . . .	128
8.1.3	A useful lemma . . . . .	128
8.1.4	Estimating $A$ and $B$ in the useful lemma . . . . .	129
8.1.5	Applying the useful lemma . . . . .	130
8.1.6	Connection to the theta number . . . . .	130
8.2	Higher rank Grothendieck inequality . . . . .	131
8.2.1	Randomized rounding by projecting . . . . .	132
8.2.2	Extension of Grothendieck's identity . . . . .	132
8.2.3	Proof of the theorem . . . . .	133
8.3	Further reading . . . . .	133
8.4	Exercises . . . . .	134
<b>9</b>	<b>Optimizing with ellipsoids and determinants</b>	<b>136</b>
9.1	Determinant maximization problems . . . . .	137
9.2	Convex spectral functions . . . . .	138
9.2.1	Minkowski's determinant inequality . . . . .	139
9.2.2	Davis' characterization of convex spectral functions . . . . .	139
9.3	Approximating polytopes by ellipsoids . . . . .	142
9.3.1	Inner approximation . . . . .	142
9.3.2	Outer approximation . . . . .	143
9.4	The Löwner-John ellipsoids . . . . .	144
9.5	Further reading . . . . .	148
9.6	Exercises . . . . .	148

---

---

# CHAPTER 1

---

## BACKGROUND: CONVEX SETS AND POSITIVE SEMIDEFINITE MATRICES

A set  $C$  is called convex if, given any two points  $x$  and  $y$  in  $C$ , the straight line segment connecting  $x$  and  $y$  lies completely inside of  $C$ . For instance, cubes, balls or ellipsoids are convex sets whereas a torus is not. Intuitively, convex sets do not have holes or dips.

Usually, arguments involving convex sets are easy to visualize by two-dimensional drawings. One reason being that the definition of convexity only involves three points which always lie in some two-dimensional plane. On the other hand, convexity is a very powerful concept which appears (sometimes unexpected) in many branches of mathematics and its applications. Here are a few areas where convexity is an important concept: mathematical optimization, high-dimensional geometry, analysis, probability theory, system and control, harmonic analysis, calculus of variations, game theory, computer science, functional analysis, economics, and there are many more.

Our aim is to work with convex sets algorithmically. So we have to discuss ways to represent them in the computer, in particular which data do we want to give to the computer. Roughly speaking, there are two convenient possibilities to represent convex sets: By an implicit description as an intersection of halfspaces or by an explicit description as the convex combination of extreme points. The goal of this chapter is to discuss these two representations. In the context of functional analysis they are connected to two famous theorems, the Hahn-Banach theorem and the Krein-Milman theorem. Since we are only working in finite-dimensional Euclidean spaces (and not in the more general setting of infinite-dimensional topological vector spaces) we can derive the statements

using simple geometric arguments.

Later we develop the theory of convex optimization in the framework of conic programs. For this we need a special class of convex sets, namely convex cones. The for optimization most relevant convex cones are at the moment two involving vectors in  $\mathbb{R}^n$  and two involving symmetric matrices in  $\mathbb{R}^{n \times n}$ , namely the non-negative orthant, the second order cone, the cone of positive semidefinite matrices, and the cone of copositive matrices. Clearly, the cone of positive semidefinite matrices plays the main role here. As background information we collect a number of basic properties of positive semidefinite matrices.

## 1.1 Some fundamental notions

Before we turn to convex sets we recall some fundamental geometric notions. The following is a brief review, without proofs, of some basic definitions and notations appearing frequently in the sequel.

### 1.1.1 Euclidean space

Let  $E$  be an  $n$ -dimensional *Euclidean space* which is an  $n$ -dimensional real vector space having an inner product. We usually use the notation  $x \cdot y$  for the inner product between the vectors  $x$  and  $y$ . This inner product defines a norm on  $E$  by  $\|x\| = \sqrt{x \cdot x}$  and a metric by  $d(x, y) = \|x - y\|$ .

For sake of concreteness we will work with coordinates most of the time: One can always identify  $E$  with  $\mathbb{R}^n$  where the inner product of the column vectors  $x = (x_1, \dots, x_n)^T$  and  $y = (y_1, \dots, y_n)^T$  is the usual one:  $x \cdot y = x^T y = \sum_{i=1}^n x_i y_i$ . This identification involves a linear transformation  $T : E \rightarrow \mathbb{R}^n$  which is an isometry, i.e.  $x \cdot y = Tx \cdot Ty$  holds for all  $x, y \in E$ . Then the norm is the Euclidean norm (or  $\ell_2$ -norm):  $\|x\|_2 = \sqrt{\sum_i x_i^2}$  and  $d(x, y) = \|x - y\|_2$  is the Euclidean distance between two points  $x, y \in \mathbb{R}^n$ .

### 1.1.2 Topology in finite-dimensional metric spaces

The *ball* with center  $x \in \mathbb{R}^n$  and radius  $r$  is

$$B(x, r) = \{y \in \mathbb{R}^n : d(x, y) \leq r\}.$$

Let  $A$  be a subset of  $n$ -dimensional Euclidean space. A point  $x \in A$  is an *interior point* of  $A$  if there is a positive radius  $\varepsilon > 0$  so that  $B(x, \varepsilon) \subseteq A$ . The set of all interior points of  $A$  is denoted by  $\text{int } A$ . We say that a set  $A$  is *open* if all points of  $A$  are interior points, i.e. if  $A = \text{int } A$ . The set  $A$  is *closed* if its complement  $\mathbb{R}^n \setminus A$  is open. The (*topological*) *closure*  $\bar{A}$  of  $A$  is the smallest (inclusion-wise) closed set containing  $A$ . One can show that a set  $A$  in  $\mathbb{R}^n$  is closed if and only if every converging sequence of points in  $A$  has a limit which also lies in  $A$ . A point  $x \in A$  belongs to the *boundary*  $\partial A$  of  $A$  if for every  $\varepsilon > 0$  the ball  $B(x, \varepsilon)$  contains points in  $A$  and in  $\mathbb{R}^n \setminus A$ . The boundary  $\partial A$  is a closed

set and we have  $\overline{A} = A \cup \partial A$ , and  $\partial A = \overline{A} \setminus \text{int } A$ . The set  $A$  is *compact* if every sequence in  $A$  contains a convergent subsequence. The set  $A$  is compact if and only if it is closed and bounded (i.e. it is contained in a ball of sufficiently large, but finite, radius).



Figure 1.1: A compact, non-convex set  $A$ . Which points lie in  $\text{int } A$ ,  $\overline{A}$ ,  $\partial A$ ?

For instance, the boundary of the ball with radius 1 and center 0 is the *unit sphere*

$$\partial B(0, 1) = \{y \in \mathbb{R}^n : d(0, y) = 1\} = \{x \in \mathbb{R}^n : x^\top x = 1\}.$$

Traditionally, it is called the  $(n - 1)$ -dimensional unit sphere, denoted as  $\mathbb{S}^{n-1}$ , where the superscript  $n - 1$  indicates the dimension of the manifold.

### 1.1.3 Affine geometry

A subset  $A \subseteq \mathbb{R}^n$  is called an *affine subspace* of  $\mathbb{R}^n$  if it is a translated linear subspace: One can write  $A$  in the form

$$A = x + L = \{x + y : y \in L\}$$

where  $x \in \mathbb{R}^n$  and where  $L$  is a linear subspace of  $\mathbb{R}^n$ . The *dimension* of  $A$  is defined as  $\dim A = \dim L$ . Affine subspaces are closed under *affine linear combinations*:

$$\forall N \in \mathbb{N} \forall x_1, \dots, x_N \in A \forall \alpha_1, \dots, \alpha_N \in \mathbb{R} : \sum_{i=1}^N \alpha_i = 1 \implies \sum_{i=1}^N \alpha_i x_i \in A.$$

The smallest affine subspace containing a set of given points is its *affine hull*. The affine hull of  $A \subseteq \mathbb{R}^n$  is the set of all possible affine linear combinations

$$\text{aff } A = \left\{ \sum_{i=1}^N \alpha_i x_i : N \in \mathbb{N}, x_1, \dots, x_N \in A, \alpha_1, \dots, \alpha_N \in \mathbb{R}, \sum_{i=1}^N \alpha_i = 1, \right\}.$$



A fact which requires a little proof (exercise). The dimension of an arbitrary set  $A$  is  $\dim A = \dim(\text{aff } A)$ . One-dimensional affine subspaces are *lines* and  $(n-1)$ -dimensional affine subspaces are *hyperplanes*. A hyperplane can be specified as

$$H = \{x \in \mathbb{R}^n : c^T x = \beta\},$$

where  $c \in \mathbb{R}^n \setminus \{0\}$  is the normal of  $H$  (which lies orthogonal to  $H$ ) and where  $\beta \in \mathbb{R}$ . Sometimes we write  $H_{c,\beta}$  for it.

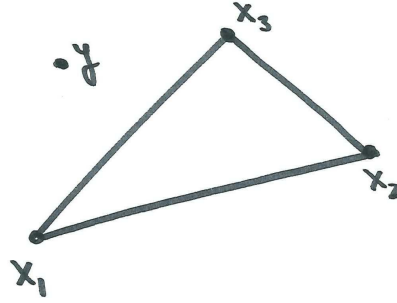


Figure 1.2: Determine (as accurate as possible) the coefficients  $\alpha_1, \alpha_2, \alpha_3$  of the affine combination  $y = \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3$  with  $\alpha_1 + \alpha_2 + \alpha_3 = 1$ .

If the dimension of  $A \subseteq \mathbb{R}^n$  is strictly smaller than  $n$ , then  $A$  does not have an interior,  $\text{int } A = \emptyset$ . In this situation one is frequently interested in the interior points of  $A$  relative to the affine subspace  $\text{aff } A$ . We say that a point  $x \in A$  belongs to the *relative interior* of  $A$  when there is a ball  $B(x, \varepsilon)$  with strictly positive radius  $\varepsilon > 0$  so that  $\text{aff } A \cap B(x, \varepsilon) \subseteq A$ . We denote the set of all relative interior points of  $A$  by  $\text{relint } A$ . Of course, if  $\dim A = n$ , then the interior coincides with the relative interior:  $\text{int } A = \text{relint } A$ .

## 1.2 Convex sets

A subset  $C \subseteq \mathbb{R}^n$  is called a *convex set* if for every pair of points  $x, y \in C$  also the entire line segment between  $x$  and  $y$  is contained in  $C$ . The *line segment* between the points  $x$  and  $y$  is defined as

$$[x, y] = \{(1 - \alpha)x + \alpha y : 0 \leq \alpha \leq 1\}.$$

Convex sets are closed under *convex combinations*:

$$\forall N \in \mathbb{N} \forall x_1, \dots, x_N \in C \forall \alpha_1, \dots, \alpha_N \in \mathbb{R}_{\geq 0} : \sum_{i=1}^N \alpha_i = 1 \implies \sum_{i=1}^N \alpha_i x_i \in C.$$

The convex hull of  $A \subseteq \mathbb{R}^n$  is the smallest convex set containing  $A$ . It is

$$\text{conv } A = \left\{ \sum_{i=1}^N \alpha_i x_i : N \in \mathbb{N}, x_1, \dots, x_N \in A, \alpha_1, \dots, \alpha_N \in \mathbb{R}_{\geq 0}, \sum_{i=1}^N \alpha_i = 1 \right\},$$

which requires an argument. We can give a mechanical interpretation of the convex hull of finitely many point  $\text{conv}\{x_1, \dots, x_N\}$ : The convex hull consists of all centres of gravity of point masses  $\alpha_1, \dots, \alpha_N$  at the positions  $x_1, \dots, x_N$ .

The convex hull of finitely many points is called a *polytope*. Two-dimensional, planar, polytopes are polygons. Other important examples of convex sets are balls, halfspaces, and line segments. Furthermore, arbitrary intersections of convex sets are convex again. The *Minkowski sum* of convex sets  $C, D$  given by

$$C + D = \{x + y : x \in C, y \in D\}$$

is a convex set.



Figure 1.3: Exercise: What is the Minkowski sum of a square and a disk?

Here are two useful properties of convex sets. The first result gives an alternative description of the relative interior of a convex set and the second one permits to embed a convex set with an empty interior into a lower dimensional affine space.

**Lemma 1.2.1.** *Let  $C \subseteq \mathbb{R}^n$  be a convex set. A point  $x \in C$  lies in the relative interior of  $C$  if and only if*

$$\forall y \in C \exists z \in C, \alpha \in (0, 1) : x = \alpha y + (1 - \alpha)z,$$

where  $(0, 1)$  denotes the open interval  $0 < \alpha < 1$ .

**Theorem 1.2.2.** *Let  $C \subseteq \mathbb{R}^n$  be a convex set. If  $\text{int } C = \emptyset$  then the dimension of its affine closure is at most  $n - 1$ .*

### 1.3 Implicit description of convex sets

In this section we show how one can describe a closed convex set implicitly as the intersection of halfspaces (Theorem 1.3.7). For this we show the intuitive fact that through every of its boundary points there is a hyperplane which has the convex set on only one of its sides (Lemma 1.3.5). We also prove an important fact which we will need later: Any two convex sets whose relative interiors do not intersect can be properly separated by a hyperplane (Theorem 1.3.8). After giving the definitions of separating and supporting hyperplanes we look

at the metric projection which is a useful tool to construct these separating hyperplanes.

The *hyperplane* at a point  $x \in \mathbb{R}^n$  with normal vector  $c \in \mathbb{R}^n \setminus \{0\}$  is

$$H = \{y \in \mathbb{R}^n : c^\top y = c^\top x\}.$$

It is an affine subspace of dimension  $n - 1$ . The hyperplane  $H$  divides  $\mathbb{R}^n$  into two closed *halfspaces*

$$H^+ = \{y \in \mathbb{R}^n : c^\top y \geq c^\top x\}, \quad H^- = \{y \in \mathbb{R}^n : c^\top y \leq c^\top x\}.$$

A hyperplane  $H$  is said to *separate* two sets  $A \subseteq \mathbb{R}^n$  and  $B \subseteq \mathbb{R}^n$  if they lie on different sides of the hyperplane, i.e., if  $A \subseteq H^+$  and  $B \subseteq H^-$  or conversely. In other words,  $A$  and  $B$  are separated by a hyperplane if there exists a non-zero vector  $c \in \mathbb{R}^n$  and a scalar  $\beta \in \mathbb{R}$  such that

$$\forall x \in A, y \in B : c^\top x \leq \beta \leq c^\top y.$$

The separation is said to be *strict* if both inequalities are strict, i.e.,

$$\forall x \in A, y \in B : c^\top x < \beta < c^\top y.$$

The separation is said to be *proper* when  $H$  separates  $A$  and  $B$  but does not contain both  $A$  and  $B$ .

A hyperplane  $H$  is said to *support*  $A$  at  $x \in A$  if  $x \in H$  and if  $A$  is contained in one of the two halfspaces  $H^+$  or  $H^-$ , say  $H^-$ . Then  $H$  is a *supporting hyperplane* of  $A$  at  $x$  and  $H^-$  is a supporting halfspace.

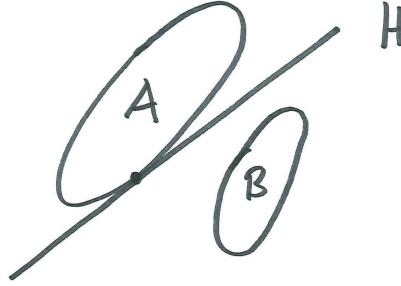


Figure 1.4: The hyperplane  $H$  supports  $A$  and separates  $A$  and  $B$ .

### 1.3.1 Metric projection

Let  $C \subseteq \mathbb{R}^n$  be a non-empty closed convex set. One can project every point  $x \in \mathbb{R}^n$  onto  $C$  by simply taking the point in  $C$  which is closest to it. This fact is very intuitive and in the case when  $C$  is a linear subspace we are talking simply about the orthogonal projection onto  $C$ .

**Lemma 1.3.1.** *Let  $C$  be a non-empty closed convex set in  $\mathbb{R}^n$ . Let  $x \in \mathbb{R}^n \setminus C$  be a point outside of  $C$ . Then there exists a unique point  $\pi_C(x)$  in  $C$  which is closest to  $x$ . Moreover,  $\pi_C(x) \in \partial C$ .*

*Proof.* The argument for *existence* is a compactness argument: As  $C$  is not empty, pick  $z_0 \in C$  and consider the intersection  $C'$  of  $C$  with the ball  $B(z_0, r)$  centered at  $z_0$  and with radius  $r = \|z_0 - x\|$ . Then  $C'$  is closed, convex and bounded. Moreover the minimum of the distance  $\|y - x\|$  for  $y \in C$  is equal to the minimum taken over  $C'$ . As we minimize a continuous function over a compact set, the minimum is attained. Hence there is at least one closest point to  $x$  in  $C$ .

The argument for *uniqueness* requires convexity: Let  $y$  and  $z$  be two distinct points in  $C$ , both having minimum distance to  $x$ . In this case, the midpoint of  $y$  and  $z$ , which lies in  $C$ , would even be closer to  $x$ , because the distance  $d(x, \frac{1}{2}(y + z))$  is the height of the isosceles triangle with vertices  $x, y, z$ .

Hence there is a unique point in  $C$  which is at minimum distance to  $x$ , which we denote by  $\pi_C(x)$ . Clearly,  $\pi_C(x) \in \partial C$ , otherwise one would find another point in  $C$  closer to  $x$  lying in some small ball  $B(\pi_C(x), \varepsilon) \subseteq C$ .  $\square$

Thus, the map  $\pi_C : \mathbb{R}^n \rightarrow C$  defined by the property

$$\forall y \in C : d(y, x) \geq d(\pi_C(x), x)$$

is well-defined. This map is called *metric projection* and sometimes we refer to the vector  $\pi_C(x)$  as the *best approximation* of  $x$  in the set  $C$ .

The metric projection  $\pi_C$  is a contraction:

**Lemma 1.3.2.** *Let  $C$  be a non-empty closed and convex set in  $\mathbb{R}^n$ . Then,*

$$\forall x, y \in \mathbb{R}^n : d(\pi_C(x), \pi_C(y)) \leq d(x, y).$$

*In particular, the metric projection  $\pi_C$  is a Lipschitz continuous map.*

*Proof.* We can assume that  $d(\pi_C(x), \pi_C(y)) \neq 0$ . Consider the line segment  $[\pi_C(x), \pi_C(y)]$  and the two parallel hyperplanes  $H_x$  and  $H_y$  at  $\pi_C(x)$  and at  $\pi_C(y)$  both having normal vector  $\pi_C(x) - \pi_C(y)$ . The points  $x$  and  $\pi_C(y)$  are separated by  $H_x$  because otherwise there would be a point in  $[\pi_C(x), \pi_C(y)] \subseteq C$  which is closer to  $x$  than to  $\pi_C(x)$ , which is impossible. In the same way,  $y$  and  $\pi_C(x)$  are separated by  $H_y$ . Hence,  $x$  and  $y$  are on different sides of the “slab” bounded by the parallel hyperplanes  $H_x$  and by  $H_y$ . So their distance  $d(x, y)$  is at least the width of the slab, which is  $d(\pi_C(x), \pi_C(y))$ .  $\square$

The metric projection can reach every point on the boundary of  $C$ :

**Lemma 1.3.3.** *Let  $C$  be a non-empty closed and convex set in  $\mathbb{R}^n$ . Then, for every boundary point  $y \in \partial C$  there is a point  $x$  lying outside of  $C$  so that  $y = \pi_C(x)$ .*

*Proof.* First note that one can assume that  $C$  is bounded (since otherwise replace  $C$  by its intersection with a ball around  $y$ ). Since  $C$  is bounded it is contained in a ball  $B$  of sufficiently large radius. We will construct the desired

point  $x$  which lies on the boundary  $\partial B$  by a limit argument. For this choose a sequence of points  $y_i \in \mathbb{R}^n \setminus C$  such that  $d(y, y_i) < 1/i$ , and hence  $\lim_{i \rightarrow \infty} y_i = y$ . Because the metric projection is a contraction (Lemma 1.3.2) we have

$$d(y, \pi_C(y_i)) = d(\pi_C(y), \pi_C(y_i)) \leq d(y, y_i) < 1/i.$$

By intersecting the line  $\text{aff}\{y_i, \pi_C(y_i)\}$  with the boundary  $\partial B$  one can determine a point  $x_i \in \partial B$  so that  $\pi_C(x_i) = \pi_C(y_i)$ . Since the boundary  $\partial B$  is compact there is a convergent subsequence  $(x_{i_j})$  having a limit  $x \in \partial B$ . Then, because of the previous considerations and because  $\pi_C$  is continuous

$$\begin{aligned} y &= \pi_C(y) = \pi_C\left(\lim_{j \rightarrow \infty} y_{i_j}\right) = \lim_{j \rightarrow \infty} \pi_C(y_{i_j}) \\ &= \lim_{j \rightarrow \infty} \pi_C(x_{i_j}) = \pi_C\left(\lim_{j \rightarrow \infty} x_{i_j}\right) = \pi_C(x), \end{aligned}$$

which proves the lemma.  $\square$

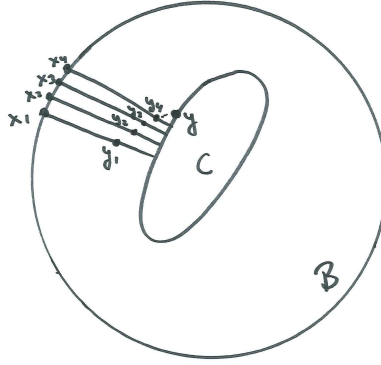


Figure 1.5: The construction which proves Lemma 1.3.3.

### 1.3.2 Separating and supporting hyperplanes

One can use the metric projection to construct separating and supporting hyperplanes:

**Lemma 1.3.4.** *Let  $C$  be a non-empty closed convex set in  $\mathbb{R}^n$ . Let  $x \in \mathbb{R}^n \setminus C$  be a point outside  $C$  and let  $\pi_C(x)$  its closest point in  $C$ . Then the following holds.*

- (i) *The hyperplane through  $x$  with normal  $x - \pi_C(x)$  supports  $C$  at  $\pi_C(x)$  and thus it separates  $\{x\}$  and  $C$ .*
- (ii) *The hyperplane through  $(x + \pi_C(x))/2$  with normal  $x - \pi_C(x)$  strictly separates  $\{x\}$  and  $C$ .*

*Proof.* It suffices to prove (i) and then (ii) follows directly. Consider the hyperplane  $H$  through  $x$  with normal vector  $c = x - \pi_C(x)$ , defined by

$$H = \{y \in \mathbb{R}^n : c^\top y = c^\top \pi_C(x)\}.$$

As  $c^\top x > c^\top \pi_C(x)$ ,  $x$  lies in the open halfspace  $\{y : c^\top y > c^\top \pi_C(x)\}$ . We show that  $C$  lies in the closed halfspace  $\{y : c^\top y \leq c^\top \pi_C(x)\}$ . Suppose for a contradiction that there exists  $y \in C$  such that  $c^\top (y - \pi_C(x)) > 0$ . Then select a scalar  $\lambda \in (0, 1)$  such that  $0 < \lambda < \frac{2c^\top (y - \pi_C(x))}{\|y - \pi_C(x)\|^2} < 1$  and set  $w = \lambda y + (1 - \lambda)\pi_C(x)$  which is a point  $C$ . Now verify that  $\|w - x\| < \|\pi_C(x) - x\| = \|c\|$ , which follows from

$$\|w - x\|^2 = \|\lambda(y - \pi_C(x)) - c\|^2 = \|c\|^2 + \lambda^2\|y - \pi_C(x)\|^2 - 2\lambda c^\top (y - \pi_C(x))$$

and which contradicts the fact that  $\pi_C(x)$  is the closest point in  $C$  to  $x$ .  $\square$

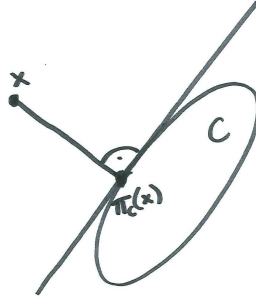


Figure 1.6: A separating hyperplane constructed using  $\pi_C$ .

Combining Lemma 1.3.3 and Lemma 1.3.4 we deduce that one can construct a supporting hyperplane at every boundary point.

**Lemma 1.3.5.** *Let  $C \subseteq \mathbb{R}^n$  be a closed convex set and let  $x \in \partial C$  be a point lying on the boundary of  $C$ . Then there is a hyperplane which supports  $C$  at  $x$ .*

One can generalize Lemma 1.3.4 (i) and remove the assumption that  $C$  is closed.

**Lemma 1.3.6.** *Let  $C \subseteq \mathbb{R}^n$  be a non-empty convex set and let  $x \in \mathbb{R}^n \setminus C$  be a point lying outside  $C$ . Then,  $\{x\}$  and  $C$  can be separated by a hyperplane.*

*Proof.* In view of Lemma 1.3.1 we only have to show the result for non-closed convex sets  $C$ . We are left with two cases: If  $x \notin \overline{C}$ , then a hyperplane separating  $\{x\}$  and the closed and convex set  $\overline{C}$  also separates  $\{x\}$  and  $C$ . If  $x \in \overline{C}$ , then  $x \in \partial \overline{C}$ . By Lemma 1.3.5 there is a hyperplane supporting  $\overline{C}$  at  $x$ . In particular, it separates  $\{x\}$  and  $C$ .  $\square$

As a direct application of the strict separation result in Lemma 1.3.4 (ii), we can formulate the following fundamental structural result for closed convex sets.

**Theorem 1.3.7.** *A non-empty closed convex set is the intersection of its supporting halfspaces.*

This is an implicit description as it gives a method to verify whether a point belongs to the closed convex set in question: One has to check whether the point lies in all these supporting halfspaces. If the closed convex set is given as an intersection of finitely many halfspaces, then it is called a *polyhedron* and the test we just described is a simple algorithmic membership test.

We conclude with the following result which characterizes when two convex sets can be separated properly. When both sets are closed and one of them is bounded, one can show a strict separation. These separation results will be the basis in our discussion of the duality theory of conic programs.

**Theorem 1.3.8.** *Let  $C, D \subseteq \mathbb{R}^n$  be non-empty convex sets.*

- (i)  *$C$  and  $D$  can be properly separated if and only if their relative interiors do not have a point in common:  $\text{relint } C \cap \text{relint } D = \emptyset$ .*
- (ii) *Assume that  $C$  and  $D$  are closed and that at least one of them is bounded. If  $C \cap D = \emptyset$ , then there is a hyperplane strictly separating  $C$  and  $D$ .*

*Proof.* (i) The “only if” part ( $\implies$ ): Let  $H_{c,\beta}$  be a hyperplane properly separating  $C$  and  $D$  with  $C \subseteq H^-$  and  $D \subseteq H^+$ , i.e.,

$$\forall x \in C, y \in D : c^\top x \leq \beta \leq c^\top y.$$

Suppose there is a point  $x_0 \in \text{relint } C \cap \text{relint } D$ . Then  $c^\top x_0 = \beta$ , i.e.,  $x_0 \in H$ . Pick any  $x \in C$ . By Lemma 1.2.1 there exists  $x' \in C$  and  $\alpha \in (0, 1)$  such that  $x_0 = \alpha x + (1 - \alpha)x'$ . Now

$$\beta = c^\top x_0 = \alpha c^\top x + (1 - \alpha)c^\top x' \leq \alpha\beta + (1 - \alpha)\beta = \beta,$$

hence all inequalities have to be tight and so  $c^\top x = \beta$ . Thus  $C$  is contained in the hyperplane  $H$ . Similarly,  $D \subseteq H$ . This contradicts the assumption that the separation is proper.

The “if part” ( $\impliedby$ ): Consider the set

$$E = \text{relint } C - \text{relint } D = \{x - y : x \in \text{relint } C, y \in \text{relint } D\},$$

which is convex. By assumption, the origin 0 does not lie in  $E$ . By Lemma 1.3.6 there is a hyperplane  $H$  separating  $\{0\}$  and  $E$  which goes through the origin. Say  $H = H_{c,0}$  and

$$\forall x \in \text{relint } C, y \in \text{relint } D : c^\top (x - y) \geq 0.$$

Define

$$\beta = \inf\{c^\top x : x \in \text{relint } C\}.$$

Then,

$$C \subseteq \{x \in \mathbb{R}^n : c^\top x \geq \beta\},$$

and we want to show that

$$D \subseteq \{y : c^\top y \leq \beta\}.$$

For suppose not. Then there is a point  $y \in \text{relint } D$  so that  $c^\top y > \beta$ . Moreover, by definition of the infimum there is a point  $x \in \text{relint } C$  so that  $\beta \leq c^\top x < c^\top y$ . But then we find  $c^\top(x - y) < 0$ , a contradiction. Thus,  $C$  and  $D$  are separated by the hyperplane  $H_{c,\beta}$ .

If  $C \cup D$  lies in some lower dimensional affine subspace, then the argument above gives a hyperplane in the affine subspace  $\text{aff}(C \cup D)$  which can be extended to a hyperplane in  $\mathbb{R}^n$  which properly separates  $C$  and  $D$ .

(ii) Assume that  $C$  is bounded and  $C \cap D = \emptyset$ . Consider now the set

$$E = C - D$$

which is closed (check it) and convex. As the origin 0 does not lie in  $E$ , by Lemma 1.3.4 (ii), there is a hyperplane strictly separating  $\{0\}$  and  $E$ : There is a non-zero vector  $c$  and a positive scalar  $\beta$  such that

$$\forall x \in C, y \in D : c^\top(x - y) > \beta > 0.$$

This implies

$$\inf_{x \in C} c^\top x \geq \beta + \sup_{y \in D} c^\top y > \frac{\beta}{2} + \sup_{y \in D} c^\top y > \sup_{y \in D} c^\top y.$$

Hence the hyperplane  $H_{c,\alpha}$  with  $\alpha = \frac{\beta}{2} + \sup_{y \in D} c^\top y$  strictly separates  $C$  and  $D$ .  $\square$

## 1.4 Explicit description of convex sets

Now we turn to an explicit description of convex sets. An explicit description gives an easy way to generate points lying in the convex set.

We say that a point  $x \in C$  is *extreme* if it is not a relative interior point of any line segment in  $C$ . In other words, if  $x$  cannot be written in the form  $x = (1 - \alpha)y + \alpha z$  with  $y, z \in C$  and  $0 < \alpha < 1$ . The set of all extreme points of  $C$  we denote by  $\text{ext } C$ .

**Theorem 1.4.1.** *Let  $C \subseteq \mathbb{R}^n$  be a compact and convex set. Then,*

$$C = \text{conv}(\text{ext } C).$$



*Proof.* We prove the theorem by induction on the dimension  $n$ . If  $n = 0$ , then  $C$  is a point and the result follows.

Let the dimension  $n$  be at least one. If the interior of  $C$  is empty, then  $C$  lies in an affine subspace of dimension at most  $n - 1$  and the theorem follows from the induction hypothesis. Suppose that  $\text{int } C \neq \emptyset$ . We have to show that every  $x \in C$  can be written as the convex hull of extreme points of  $C$ . We distinguish between two cases:

First case: If  $x$  lies on the boundary of  $C$ , then by Lemma 1.3.5 there is a supporting hyperplane  $H$  of  $C$  through  $x$ . Consider the set  $F = H \cap C$ . This is a compact and convex set which lies in an affine subspace of dimension at most  $n - 1$  and hence we have by the induction hypotheses  $x \in \text{conv}(\text{ext } F)$ . Since  $\text{ext } F \subseteq \text{ext } C$ , we are done.

Second case: If  $x$  does not lie on the boundary of  $C$ , then the intersection of a line through  $x$  with  $C$  is a line segment  $[y, z]$  with  $y, z \in \partial C$ . By the previous argument we have  $y, z \in \text{conv}(\text{ext } C)$ . Since  $x$  is a convex combination of  $y$  and  $z$ , the theorem follows.  $\square$

## 1.5 Convex cones

We will develop the theory of convex optimization using the concept of conic programs. Before we can say what a “conic program” is, we have to define convex cones.

**Definition 1.5.1.** A non-empty subset  $K$  of  $\mathbb{R}^n$  is called a convex cone if it is closed under non-negative linear combinations:

$$\forall \alpha, \beta \in \mathbb{R}_{\geq 0} \quad \forall x, y \in K : \alpha x + \beta y \in K.$$

Moreover,  $K$  is pointed if

$$x, -x \in K \implies x = 0.$$

One can easily check that convex cones are indeed convex sets. Furthermore, the direct product

$$K \times K' = \{(x, x') \in \mathbb{R}^{n+n'} : x \in K, x' \in K'\}$$

of two convex cones  $K \subseteq \mathbb{R}^n$  and  $K' \subseteq \mathbb{R}^{n'}$  is a convex cone again.

A pointed convex cone in  $\mathbb{R}^n$  defines a partial order on  $\mathbb{R}^n$  by

$$x \geq y \iff x - y \in K$$

for  $x, y \in \mathbb{R}^n$ . This partial order satisfies the following conditions:

*reflexivity:*

$$\forall x \in \mathbb{R}^n : x \geq x$$

antisymmetry:

$$\forall x, y \in \mathbb{R}^n : x \geq y, y \geq x \implies x = y$$

transitivity:

$$\forall x, y, z \in \mathbb{R}^n : x \geq y, y \geq z \implies x \geq z$$

homogeneity:

$$\forall x, y \in \mathbb{R}^n \forall \alpha \in \mathbb{R}_{\geq 0} : x \geq y \implies \alpha x \geq \alpha y$$

additivity:

$$\forall x, y, x', y' \in \mathbb{R}^n : x \geq y, x' \geq y' \implies x + x' \geq y + y'.$$

In order that a convex cone is useful for practical algorithmic optimization methods we will need two additional properties to eliminate undesired degenerate conditions: A convex cone should be closed and full-dimensional, that is, it has a non-empty interior. Then, we define strict inequalities by:

$$x > y \iff x - y \in \text{int } K.$$

Let  $(x_i)_{i \in \mathbb{N}}$  and  $(y_i)_{i \in \mathbb{N}}$  be sequences of elements in  $\mathbb{R}^n$  which have limits  $x$  and  $y$ , then we can pass to limits in the inequalities:

$$(\exists N \in \mathbb{N} \forall i \geq N : x_i \geq y_i) \iff x \geq y.$$

The separation result from Lemma 1.3.4 specializes to convex cones in the following way.

**Lemma 1.5.2.** *Let  $C \subseteq \mathbb{R}^n$  be a closed convex cone and let  $x \in \mathbb{R}^n \setminus C$  be a point outside of  $C$ . Then there is a linear hyperplane separating  $\{x\}$  and  $C$ . Even stronger, there is a non-zero vector  $c \in \mathbb{R}^n$  such that*

$$\forall y \in C : c^\top y \geq 0 > c^\top x,$$

thus with the strict inequality  $c^\top x < 0$ .

## 1.6 Examples

The convex cone generated by a set of vectors  $A \subseteq \mathbb{R}^n$  is the smallest convex cone containing  $A$ . It is

$$\text{cone } A = \left\{ \sum_{i=1}^N \alpha_i x_i : N \in \mathbb{N}, x_1, \dots, x_N \in A, \alpha_1, \dots, \alpha_N \in \mathbb{R}_{\geq 0} \right\}.$$

Furthermore, every linear subspace of  $E$  is a convex cone, however a somewhat boring one. More interesting are the following examples. We will use them, especially cone of positive semidefinite matrices, very often.

### 1.6.1 The non-negative orthant and linear programming

The convex cone which is connected to linear programming is the non-negative orthant. It lies in the Euclidean space  $\mathbb{R}^n$  with the standard inner product. The *non-negative orthant* is defined as

$$\mathbb{R}_{\geq 0}^n = \{x = (x_1, \dots, x_n)^T \in \mathbb{R}^n : x_1, \dots, x_n \geq 0\}.$$

It is a pointed, closed and full-dimensional cone. A *linear program* is an optimization problem of the following form

$$\begin{aligned} & \text{maximize} && c_1x_1 + \dots + c_nx_n \\ & \text{subject to} && a_{11}x_1 + \dots + a_{1n}x_n \geq b_1 \\ & && a_{21}x_1 + \dots + a_{2n}x_n \geq b_2 \\ & && \vdots \\ & && a_{m1}x_1 + \dots + a_{mn}x_n \geq b_m. \end{aligned}$$

One can express the above linear program more conveniently using the partial order defined by the non-negative orthant  $\mathbb{R}_{\geq 0}^n$ :

$$\begin{aligned} & \text{maximize} && c^T x \\ & \text{subject to} && Ax \geq b, \end{aligned}$$

where  $c = (c_1, \dots, c_n)^T \in \mathbb{R}^n$  is the *objective vector*,  $A = (a_{ij}) \in \mathbb{R}^{m \times n}$  is the matrix of linear *constraints*,  $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$  is the *optimization variable*, and  $b = (b_1, \dots, b_m)^T \in \mathbb{R}^m$  is the *right hand side*. Here, the partial order  $x \geq y$  means inequality coordinate-wise:  $x_i \geq y_i$  for all  $i \in [n]$ .

### 1.6.2 The second-order cone

While the non-negative orthant is a polyhedron, the following cone is not. The *second-order cone* is defined in the Euclidean space  $\mathbb{R}^{n+1} = \mathbb{R}^n \times \mathbb{R}$  with the standard inner product. It is

$$\mathcal{L}^{n+1} = \left\{ (x, t) \in \mathbb{R}^n \times \mathbb{R} : \|x\|_2 = \sqrt{x_1^2 + \dots + x_n^2} \leq t \right\}.$$

Sometimes it is also called *ice cream cone* (make a drawing to convince yourself) or *Lorentz cone*. The second-order cone will turn out to be connected to conic quadratic programming.

### 1.6.3 The cone of semidefinite matrices

The convex cone which will turn out to be connected to semidefinite programming is the cone of positive semidefinite matrices. It lies in the  $n(n+1)/2$ -dimensional Euclidean space of  $n \times n$ -symmetric matrices  $\mathcal{S}^n$  with the trace

inner product. Namely, for two matrices  $X, Y \in \mathbb{R}^{n \times n}$ ,

$$\langle X, Y \rangle = \text{Tr}(X^T Y) = \sum_{i=1}^n \sum_{j=1}^n X_{ij} Y_{ij}, \quad \text{where } \text{Tr} X = \sum_{i=1}^n X_{ii}.$$

Here we identify the Euclidean space  $\mathcal{S}^n$  with  $\mathbb{R}^{n(n+1)/2}$  by the isometry  $T : \mathcal{S}^n \rightarrow \mathbb{R}^{n(n+1)/2}$  defined by

$$T(X) = (X_{11}, \sqrt{2}X_{12}, \sqrt{2}X_{13}, \dots, \sqrt{2}X_{1n}, X_{22}, \sqrt{2}X_{23}, \dots, \sqrt{2}X_{2n}, \dots, X_{nn})$$

where we only consider the upper triangular part (in good old FORTRAN 77 tradition) of the matrix  $X$ .

The cone of *semidefinite matrices* is

$$\mathcal{S}_{\geq 0}^n = \{X \in \mathcal{S}^n : X \text{ is positive semidefinite}\},$$

where a matrix  $X$  is *positive semidefinite* if

$$\forall x \in \mathbb{R}^n : x^T X x \geq 0.$$

More characterizations are given in Section 1.7 below.

### 1.6.4 The copositive cone

The copositive cone is a cone in  $\mathcal{S}^n$  which contains the semidefinite cone. It is the basis of copositive programming and it is defined as the set of all copositive matrices:

$$\mathcal{C}^n = \{X \in \mathcal{S}^n : x^T X x \geq 0 \quad \forall x \in \mathbb{R}_{\geq 0}^n\}.$$

Unlike for the semidefinite cone no easy characterization (for example in terms of eigenvalues) of copositive matrices is known. Even stronger: Unless the complexity classes  $\mathcal{P}$  and  $\mathcal{NP}$  coincide no easy characterization (meaning one which is polynomial-time computable) exists.

## 1.7 Positive semidefinite matrices

### 1.7.1 Basic facts

A matrix  $P$  is *orthogonal* if  $PP^T = I_n$  or, equivalently,  $P^T P = I_n$ , i.e. the rows (resp., the columns) of  $P$  form an orthonormal basis of  $\mathbb{R}^n$ . By  $\mathcal{O}(n)$  we denote the set of  $n \times n$  orthogonal matrices which forms a group under matrix multiplication.

The spectral decomposition theorem is probably the most important theorem about real symmetric matrices.

**Theorem 1.7.1. (Spectral decomposition theorem)** Any real symmetric matrix  $X \in \mathcal{S}^n$  can be decomposed as

$$X = \sum_{i=1}^n \lambda_i u_i u_i^T,$$

where  $\lambda_1, \dots, \lambda_n \in \mathbb{R}$  are the eigenvalues of  $X$  and where  $u_1, \dots, u_n \in \mathbb{R}^n$  are the corresponding eigenvectors which form an orthonormal basis of  $\mathbb{R}^n$ . In matrix terms,  $X = PDP^T$ , where  $D$  is the diagonal matrix with the  $\lambda_i$ 's on the diagonal and  $P$  is the orthogonal matrix with the  $u_i$ 's as its columns.

**Theorem 1.7.2. (Positive semidefinite matrices)** Let  $X \in \mathcal{S}^n$  be a symmetric matrix. The following assertions are equivalent.

- (1)  $X$  is positive semidefinite, written as  $X \geq 0$ , which is defined by the property:  $x^T X x \geq 0$  for all  $x \in \mathbb{R}^n$ .
- (2) The smallest eigenvalue of  $X$  is non-negative, i.e., the spectral decomposition of  $X$  is of the form  $X = \sum_{i=1}^n \lambda_i u_i u_i^T$  with all  $\lambda_i \geq 0$ .
- (3)  $X = LL^T$  for some matrix  $L \in \mathbb{R}^{n \times k}$  (for some  $k \geq 1$ ), called a Cholesky decomposition of  $X$ .
- (4) There exist vectors  $v_1, \dots, v_n \in \mathbb{R}^k$  (for some  $k \geq 1$ ) such that  $X_{ij} = v_i^T v_j$  for all  $i, j \in [n]$ ; the vectors  $v_i$ 's are called a Gram representation of  $X$ .
- (5) All principal minors of  $X$  are non-negative.

The set  $\mathcal{S}_{\geq 0}^n$  of all positive semidefinite matrices is a pointed, closed, convex, full-dimensional cone in  $\mathcal{S}^n$ . Moreover, it is generated by rank one matrices, i.e.

$$\mathcal{S}_{\geq 0}^n = \text{cone}\{xx^T : x \in \mathbb{R}^n\}.$$

Matrices lying in the interior of the cone  $\mathcal{S}_{\geq 0}^n$  are called *positive definite*. The above result extends to positive definite matrices. A matrix  $X$  is positive definite (denoted as  $X > 0$ ) if it satisfies any of the following equivalent properties: (1)  $x^T X x > 0$  for all  $x \in \mathbb{R}^n \setminus \{0\}$ , (2) all eigenvalues are strictly positive, (3) in a Cholesky decomposition the matrix  $L$  is non-singular, (4) any Gram representation has full rank  $n$ , and (5) all the principal minors are positive (in fact already positivity of all the leading principal minors implies positive definiteness; Sylvester's criterion).

### 1.7.2 The trace inner product

The *trace* of an  $n \times n$  matrix  $A$  is defined as  $\text{Tr}(A) = \sum_{i=1}^n A_{ii}$ . The trace is a linear form on  $\mathbb{R}^{n \times n}$  and satisfies the following properties:  $\text{Tr}(A) = \text{Tr}(A^T)$ ,  $\text{Tr}(AB) = \text{Tr}(BA)$ . Moreover, if  $A$  is symmetric then the trace of  $A$  is equal to the sum of the eigenvalues of  $A$ .

One can define an inner product on  $\mathbb{R}^{n \times n}$  by setting

$$\langle A, B \rangle = \text{Tr}(A^T B) = \sum_{i,j=1}^n A_{ij} B_{ij}.$$

This defines the *Frobenius norm* on  $\mathbb{R}^{n \times n}$  by setting  $\|A\| = \sqrt{\langle A, A \rangle}$ . For a vector  $x \in \mathbb{R}^n$  we have  $x^T A x = \langle A, x x^T \rangle$ . For positive semidefinite matrices we have the following result.

**Lemma 1.7.3.** *For a symmetric matrix  $A \in \mathcal{S}^n$ ,*

$$A \geq 0 \iff \forall B \in \mathcal{S}_{\geq 0}^n : \langle A, B \rangle \geq 0.$$

*In other words, the cone  $\mathcal{S}_{\geq 0}^n$  is self-dual:  $(\mathcal{S}_{\geq 0}^n)^* = \mathcal{S}_{\geq 0}^n$ .*

*Proof.* Direct verification using the conditions (1) and (2) in Theorem 1.7.2.  $\square$

### 1.7.3 Hoffman-Wielandt inequality

Here is a nice inequality to know about eigenvalues.

**Theorem 1.7.4. (Hoffman, Wielandt (1953))** *Let  $A, B \in \mathcal{S}^n$  be symmetric matrices with respective eigenvalues  $\alpha_1, \dots, \alpha_n$  and  $\beta_1, \dots, \beta_n$  ordered as follows:  $\alpha_1 \leq \dots \leq \alpha_n$  and  $\beta_1 \geq \dots \geq \beta_n$ . Then,*

$$\sum_{i=1}^n \alpha_i \beta_i = \min\{\text{Tr}(A X B X^T) : X \in \mathcal{O}(n)\}. \quad (1.1)$$

*In particular,*

$$\text{Tr}(AB) \geq \sum_{i=1}^n \alpha_i \beta_i.$$

*Proof.* Write  $A = P D P^T$  and  $B = Q E Q^T$  where  $P, Q \in \mathcal{O}(n)$  and  $D$  (resp.,  $E$ ) is the diagonal matrix with diagonal entries  $\alpha_i$  (resp.  $\beta_i$ ). As  $\text{Tr}(A X B X^T) = \text{Tr}(D Y E Y^T)$  where  $Y = P^T X Q \in \mathcal{O}(n)$ , the optimization problem (1.1) is equivalent to

$$\min\{\text{Tr}(D X E X^T) : X \in \mathcal{O}(n)\}.$$

We want to prove that the minimum is  $\text{Tr}(DE)$ , which is attained  $X = I$ . For this consider the linear program<sup>1</sup>

$$\max_{x, y \in \mathbb{R}^n} \left\{ \sum_{i=1}^n x_i + \sum_{j=1}^n y_j : \alpha_i \beta_j - x_i - y_j \geq 0 \ \forall i, j \in [n] \right\} \quad (1.2)$$

---

<sup>1</sup>We now use only the dual linear program (1.3), but we will use also the primal linear program (1.2) in Chapter 2 for reformulating program (1.1) as a semidefinite program.

and its dual linear program

$$\min_{Z \in \mathbb{R}^{n \times n}} \left\{ \sum_{i,j=1}^n \alpha_i \beta_j Z_{ij} : \sum_{i=1}^n Z_{ij} = 1 \ \forall j \in [n], \sum_{j=1}^n Z_{ij} = 1 \ \forall i \in [n], Z \geq 0 \right\}. \quad (1.3)$$

Note that the feasible region of the linear program (1.3) is the set of all *doubly-stochastic matrices*, i.e., the matrices with non-negative entries where all rows and all columns sum up to one. By Birkhoff's theorem the set of doubly-stochastic matrices is equal to the convex hull of all permutation matrices. In other words, the minimum of (1.3) is equal to the minimum value of  $\sum_{i=1}^n \alpha_i \beta_{\sigma(i)}$  taken over all permutations  $\sigma$  of  $[n]$ . It is an easy exercise to verify that this minimum is attained for the identity permutation. This shows that the optimum value of (1.3) (and thus of (1.2)) is equal to  $\sum_i \alpha_i \beta_i = \text{Tr}(DE)$ .

Now pick any  $X \in \mathcal{O}(n)$ . Observe that the matrix  $Z = ((X_{ij})^2)_{i,j=1}^n$  is doubly-stochastic (by the definition that  $X$  is orthogonal) and that

$$\text{Tr}(DXEX^\top) = \sum_{i,j=1}^n \alpha_i \beta_j (X_{ij})^2,$$

which implies that  $\text{Tr}(DXEX^\top)$  is at least the minimum  $\text{Tr}(DE)$  of program (1.3). This shows that the minimum of (1.1) is at least  $\text{Tr}(DE)$ , which finishes the proof of the theorem.  $\square$

#### 1.7.4 Schur complements

The following notion of *Schur complement* can be very useful for showing positive semidefiniteness.

**Definition 1.7.5. (Schur complement)** Consider a symmetric matrix  $X$  in block form

$$X = \begin{pmatrix} A & B \\ B^\top & C \end{pmatrix}, \quad (1.4)$$

with  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times l}$  and  $C \in \mathbb{R}^{l \times l}$ . Assume that  $A$  is non-singular. Then, the matrix  $C - B^\top A^{-1} B$  is called the Schur complement of  $A$  in  $X$ .

**Lemma 1.7.6.** Let  $X \in \mathcal{S}^n$  be in block form (1.4) where  $A$  is non-singular. Then,

$$X \geq 0 \iff A \geq 0 \text{ and } C - B^\top A^{-1} B \geq 0.$$

*Proof.* The following identity holds:

$$X = P^\top \begin{pmatrix} A & 0 \\ 0 & C - B^\top A^{-1} B \end{pmatrix} P, \quad \text{where } P = \begin{pmatrix} I & A^{-1} B \\ 0 & I \end{pmatrix}.$$

As  $P$  is non-singular, we deduce that  $X \geq 0$  if and only if  $(P^{-1})^\top X P^{-1} \geq 0$  which is thus equivalent to  $A \geq 0$  and  $C - B^\top A^{-1} B \geq 0$ .

### 1.7.5 Block-diagonal matrices

Given matrices  $X_1 \in \mathcal{S}^{n_1}, \dots, X_r \in \mathcal{S}^{n_r}$ ,  $X_1 \oplus \dots \oplus X_r$  denotes the following block-diagonal matrix  $X \in \mathcal{S}^n$ , where  $n = n_1 + \dots + n_r$ ,

$$X = X_1 \oplus \dots \oplus X_r = \begin{pmatrix} X_1 & 0 & \dots & 0 \\ 0 & X_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & X_r \end{pmatrix}. \quad (1.5)$$

Then,  $X$  is positive semidefinite if and only if all the blocks  $X_1, \dots, X_r$  are positive semidefinite.

Given two sets of matrices  $\mathcal{A}$  and  $\mathcal{B}$ ,  $\mathcal{A} \oplus \mathcal{B}$  denotes the set of all matrices  $X \oplus Y$ , where  $X \in \mathcal{A}$  and  $Y \in \mathcal{B}$ . Moreover, for an integer  $m \geq 1$ ,  $m\mathcal{A}$  denotes  $\mathcal{A} \oplus \dots \oplus \mathcal{A}$ , the  $m$ -fold sum.

From an algorithmic point of view it is much more economical to deal with positive semidefinite matrices in block-form like (1.5).

For instance, if we have a set  $\mathcal{A}$  of matrices that pairwise commute, then it is well known that they admit a common set of eigenvectors. In other words, there exists an orthogonal matrix  $P \in \mathcal{O}(n)$  such that the matrices  $P^T X P$  are diagonal for all  $X \in \mathcal{A}$ .

In general one may use the following powerful result about  $C^*$ -algebras which permits to show that certain sets of matrices can be block-diagonalized.

Consider a non-empty set  $\mathcal{A} \subseteq \mathbb{C}^{n \times n}$  of matrices.  $\mathcal{A}$  is said to be a  $C^*$ -algebra if it satisfies the following conditions:

1.  $\mathcal{A}$  is closed under matrix addition and multiplication, and under scalar multiplication.
2. For any matrix  $A \in \mathcal{A}$ , its conjugate transpose  $A^*$  also belongs to  $\mathcal{A}$ .

For instance, the full matrix algebra  $\mathbb{C}^{n \times n}$  is a simple instance of  $C^*$ -algebra, and the algebra  $\bigoplus_{i=1}^r m_i \mathbb{C}^{n_i \times n_i}$  as well, where  $n_i, m_i$  are integers. The following fundamental result shows that up to an orthogonal transformation this is the general form of a  $C^*$ -algebra.

**Theorem 1.7.7. (Wedderburn-Artin theorem)** *Assume  $\mathcal{A}$  is a  $C^*$ -algebra of matrices in  $\mathbb{C}^{n \times n}$  containing the identity matrix. Then there exists a unitary matrix  $P$  (i.e., such that  $PP^* = I_n$ ) and integers  $r, n_1, m_1, \dots, n_r, m_r \geq 1$  such that the set  $P^* \mathcal{A} P = \{P^* X P : X \in \mathcal{A}\}$  is equal to*

$$m_1 \mathbb{C}^{n_1 \times n_1} \oplus \dots \oplus m_r \mathbb{C}^{n_r \times n_r}.$$

See e.g. the thesis of Gijswijt [4] for a detailed exposition and its use for bounding the size of error correcting codes in finite fields.



### 1.7.6 Kronecker and Hadamard products

Given two matrices  $A = (A_{ij}) \in \mathbb{R}^{n \times m}$  and  $B = (B_{hk}) \in \mathbb{R}^{p \times q}$ , their *Kronecker product* is the matrix  $A \otimes B \in \mathbb{R}^{np \times mq}$  with entries

$$A_{ih,jk} = A_{ij}B_{hk} \quad \forall i \in [n], j \in [m], h \in [p], k \in [q].$$

It can also be seen as the  $n \times m$  block matrix whose  $ij$ -th block is the  $p \times q$  matrix  $A_{ij}B$  for all  $i \in [n], j \in [m]$ .

This includes in particular defining the Kronecker product  $u \otimes v \in \mathbb{R}^{np}$  of two vectors  $u \in \mathbb{R}^n$  and  $v \in \mathbb{R}^p$ , with entries  $(u \otimes v)_{ih} = u_i v_h$  for  $i \in [n], h \in [p]$ .

Given two matrices  $A, B \in \mathbb{R}^{n \times m}$ , their *Hadamard product* is the matrix  $A \circ B \in \mathbb{R}^{n \times m}$  with entries

$$(A \circ B)_{ij} = A_{ij}B_{ij} \quad \forall i \in [n], j \in [m].$$

Note that  $A \circ B$  coincides with the principle submatrix of  $A \otimes B$  indexed by the subset of all ‘diagonal’ pairs of indices of the form  $(ii, jj)$  for  $i \in [n], j \in [m]$ .

Here are some (easy to verify) facts about these products, where the matrices and vectors have the appropriate sizes.

1.  $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$ .
2. In particular,  $(A \otimes B)(u \otimes v) = (Au) \otimes (Bv)$ .
3. Assume  $A \in \mathcal{S}^n$  and  $B \in \mathcal{S}^p$  have, respectively, eigenvalues  $\alpha_1, \dots, \alpha_n$  and  $\beta_1, \dots, \beta_p$ . Then  $A \otimes B \in \mathcal{S}^{np}$  has eigenvalues  $\alpha_i \beta_h$  for  $i \in [n], h \in [p]$ . In particular,

$$A, B \geq 0 \implies A \otimes B \geq 0 \quad \text{and} \quad A \circ B \geq 0,$$

$$A \geq 0 \implies A^{\circ k} = ((A_{ij})^k) \geq 0 \quad \forall k \in \mathbb{N}.$$

□

## 1.8 Historical remarks

The history of convexity is astonishing: On the one hand, the notion of convexity is very natural and it can be found even in prehistoric arts. For instance, the Platonic solids are convex polyhedra and carved stone models of some of them were crafted by the late neolithic people of Scotland more than 4,000 years ago. For more information on the history, which unearthed some good hoax, see also John Baez’ discussion of “Who discovered the icosahedron?” <http://math.ucr.edu/home/baez/icosahedron/>.

On the other hand, the first mathematician who realized how important convexity is as a geometric concept was the brilliant Hermann Minkowski (1864–1909) who in a series of very influential papers “Allgemeine Lehrsätze über die konvexen Polyeder” (1897), “Theorie der konvexen Körper, insbesondere

Begründung ihres Oberflächenbegriffs” (published posthumously) initiated the mathematical study of convex sets and their properties. All the results in this chapter on the implicit and the explicit representation of convex sets can be found there (although with different proofs).

Not much can be added to David Hilbert’s (1862–1943) praise in his obituary of his close friend Minkowski:

Dieser Beweis eines tiefliegenden zahlentheoretischen Satzes<sup>2</sup> ohne rechnerische Hilfsmittel wesentlich auf Grund einer geometrisch anschaulichen Betrachtung ist eine Perle Minkowskischer Erfindungskunst. Bei der Verallgemeinerung auf Formen mit  $n$  Variablen führte der Minkowskische Beweis auf eine natürlichere und weit kleinere obere Schranke für jenes Minimum  $M$ , als sie bis dahin Hermite gefunden hatte. Noch wichtiger aber als dies war es, daß der wesentliche Gedanke des Minkowskischen Schlußverfahrens nur die Eigenschaft des Ellipsoids, daß dasselbe eine konvexe Figur ist und einen Mittelpunkt besitzt, benutzte und daher auf beliebige konvexe Figuren mit Mittelpunkt übertragen werden konnte. Dieser Umstand führte Minkowski zum ersten Male zu der Erkenntnis, daß überhaupt der *Begriff des konvexen Körpers* ein fundamentaler Begriff in unserer Wissenschaft ist und zu deren fruchtbarsten Forschungsmitteln gehört.

Ein konvexer (nirgends konkaver) Körper ist nach Minkowski als ein solcher Körper definiert, der die Eigenschaft hat, daß, wenn man zwei seiner Punkte in Auge faßt, auch die ganze geradlinige Strecke zwischen denselben zu dem Körper gehört.<sup>3</sup>

Until the end of the 1940s convex geometry was a small discipline in pure mathematics. This changed dramatically when in 1947 the breakthrough of general linear programming came. Then Dantzig formulated the linear programming problem and designed the simplex algorithm for solving it. Nowadays, convex geometry is an important toolbox for researchers, algorithm designers and practitioners in mathematical optimization.

<sup>2</sup>Hilbert is referring to Minkowski’s lattice point theorem. It states that for any invertible matrix  $A \in \mathbb{R}^{n \times n}$  defining a lattice  $A\mathbb{Z}^n$  and any convex set in  $\mathbb{R}^n$  which is symmetric with respect to the origin and with volume greater than  $2^n \det(A)^2$  contains a non-zero lattice point.

<sup>3</sup>It is not easy to translate Hilbert’s praise into English without losing its poetic tone, but here is an attempt. This proof of a deep theorem in number theory contains little calculation. Using chiefly geometry, it is a gem of Minkowski’s mathematical craft. With a generalization to forms having  $n$  variables Minkowski’s proof lead to an upper bound  $M$  which is more natural and also much smaller than the bound due to Hermite. More important than the result itself was his insight, namely that the only salient features of ellipsoids used in the proof were that ellipsoids are convex and have a center, thereby showing that the proof could be immediately generalized to arbitrary convex bodies having a center. This circumstances led Minkowski for the first time to the insight that the notion of a convex body is a fundamental and very fruitful notion in our scientific investigations ever since.

Minkowski defines a convex (nowhere concave) body as one having the property that, when one looks at two of its points, the straight line segment joining them entirely belongs to the body.

## 1.9 Further reading

Two very good books which emphasize the relation between convex geometry and optimization are by Barvinok [1] and by Gruber [5] (available online). Less optimization but more convex geometry is discussed in the little book of Bonnesen, Fenchel [3] and the encyclopedic book by Schneider [7]. The first one is now mainly interesting for historical reasons. Somewhat exceptional, and fun to read, is Chapter VII in the book of Berger [2] (available online) where he gives a panoramic view on the concept of convexity and its many relations to modern higher geometry.

Let us briefly mention connections to functional analysis. Rudin in his classical book “Functional analysis” discusses Theorem 1.3.8 and Theorem 1.4.1 in an infinite-dimensional setting. Although we will not need these more general theorems, they are nice to know.

The Hahn-Banach *separation theorem* is Theorem 3.4 in Rudin.

**Theorem 1.9.1.** *Suppose  $A$  and  $B$  are disjoint, nonempty, convex sets in a topological vector space  $X$ .*

*(a) If  $A$  is open there exist  $\Lambda \in X^*$  and  $\gamma \in \mathbb{R}$  such that*

$$\Re \Lambda x < \gamma \leq \Re \Lambda y$$

*for every  $x \in A$  and for every  $y \in B$ . (Here,  $\Re z$  is the real part of the complex number  $z$ .)*

*(b) If  $A$  is compact,  $B$  is closed, and  $X$  is locally convex, there exist  $\Lambda \in X^*$ ,  $\gamma_1 \in \mathbb{R}$ ,  $\gamma_2 \in \mathbb{R}$ , such that*

$$\Re \Lambda x < \gamma_1 < \gamma_2 < \Re \Lambda y$$

*for every  $x \in A$  and for every  $y \in B$ .*

The Krein-Milman theorem is Theorem 3.23 in Rudin.

**Theorem 1.9.2.** *Suppose  $X$  is a topological vector space on which  $X^*$  separates points. If  $K$  is a nonempty compact convex set in  $X$ , then  $K$  is the closed convex hull of the set of its extreme points.*

*In symbols,  $K = \overline{\text{conv}(\text{ext}(K))}$ .*

In his blog “What’s new?” Terry Tao [8] gives an insightful discussion of the finite-dimensional Hahn-Banach theorem.

The book “Matrix analysis” by Horn and Johnson [6] contains a wealth of very useful information, more than 70 pages, about positive definite matrices.

## 1.10 Exercises

1.1. Give a proof for the following statement:

Let  $C \subseteq \mathbb{R}^n$  be a convex set. If  $C \neq \emptyset$ , then  $\text{relint } C \neq \emptyset$

1.2. Give a proof for the following statement:

Let  $C \subseteq \mathbb{R}^n$  be a closed convex set and let  $x \in \mathbb{R}^n \setminus C$  a point lying outside of  $C$ . A separating hyperplane  $H$  is defined in Lemma 1.3.4. Consider a point  $y$  on the line  $\text{aff}\{x, \pi_C(x)\}$  which lies on the same side of the separating hyperplane  $H$  as  $x$ . Then,  $\pi_C(x) = \pi_C(y)$ .

1.3. (a) Prove or disprove: Let  $A \subseteq \mathbb{R}^n$  be a subset. Then,

$$\overline{\text{conv } A} = \text{conv } \overline{A}.$$

(b) Construct two convex sets  $C, D \subseteq \mathbb{R}^2$  so that they can be separated by a hyperplane but which cannot be properly separated.

1.4. Show that the  $l_p^n$  unit ball

$$\left\{ (x_1, \dots, x_n)^T \in \mathbb{R}^n : \|x\|_p = \left( \sum_{i=1}^n |x_i|^p \right)^{1/p} \leq 1 \right\}$$

is convex for  $p = 1$ ,  $p = 2$  and  $p = \infty$  ( $\|x\|_\infty = \max_{i=1, \dots, n} |x_i|$ ). Determine the extreme points and determine a supporting hyperplane for every boundary point.

(\*) What happens for the other  $p$ ?

---

## BIBLIOGRAPHY

- [1] A. Barvinok, *A Course in Convexity*, American Mathematical Society, 2002.
- [2] M. Berger, *Geometry revealed, a Jacob's ladder to modern higher geometry*, Springer, 2010.  
<http://www.springerlink.com/content/978-3-540-71132-2>
- [3] T. Bonnesen, W. Fenchel, *Theorie der konvexen Körper*, Springer, 1934.
- [4] D.C. Gijswijt, *Matrix algebras and semidefinite programming techniques for codes*, Ph.D. thesis, University of Amsterdam, 2005.  
<http://arxiv.org/abs/1007.0906>
- [5] P.M. Gruber, *Convex and Discrete Geometry*, Springer, 2007.  
<http://www.springerlink.com/content/978-3-540-71132-2>
- [6] R.A. Horn, C.R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.
- [7] R. Schneider, *Convex bodies: the Brunn-Minkowski theory*, Cambridge University Press, 1993.
- [8] T. Tao, *What's new? The Hahn-Banach theorem, Mengers theorem, and Hellys theorem*, 2007.  
<http://terrytao.wordpress.com/2007/11/30/the-hahn-banach-theorem-mengers-theorem-and-hellys-theorem/>

---

# CHAPTER 2

---

## SEMIDEFINITE PROGRAMS: BASIC FACTS AND EXAMPLES

---

In this chapter we introduce semidefinite programs, give some basic properties, and we present several problems that can be modeled as instances of semidefinite programs, arising from optimization, geometry and algebra.

For convenience we briefly recall some notation that we will use in this chapter. Most of it has already been introduced in Section 1.7.  $\mathcal{S}^n$  denotes the set of symmetric  $n \times n$  matrices. For a matrix  $X \in \mathcal{S}^n$ ,  $X \geq 0$  means that  $X$  is positive semidefinite and  $\mathcal{S}_{\geq 0}^n$  is the cone of positive semidefinite matrices. Analogously,  $X > 0$  means that  $X$  is positive definite and  $\mathcal{S}_{> 0}^n$  is the open cone of positive definite matrices.

Throughout  $I_n$  (or simply  $I$  when the dimension is clear from the context) denotes the  $n \times n$  identity matrix,  $e$  denotes the all-ones vector, i.e.,  $e = (1, \dots, 1)^T \in \mathbb{R}^n$ , and  $J_n = ee^T$  (or simply  $J$ ) denotes the all-ones matrix. The vectors  $e_1, \dots, e_n$  are the standard unit vectors in  $\mathbb{R}^n$ , and the matrices  $E_{ij} = (e_i e_j^T + e_j e_i^T)/2$  form the standard basis of  $\mathcal{S}^n$ .  $\mathcal{O}(n)$  denotes the set of orthogonal matrices, where  $A$  is orthogonal if  $AA^T = I_n$  or, equivalently,  $A^T A = I_n$ .

We consider the *trace inner product*:  $\langle A, B \rangle = \text{Tr}(A^T B) = \sum_{i,j=1}^n A_{ij} B_{ij}$  for two matrices  $A, B \in \mathbb{R}^{n \times n}$ . Here  $\text{Tr}(A) = \langle I_n, A \rangle = \sum_{i=1}^n A_{ii}$  denotes the trace of  $A$ . Recall that  $\text{Tr}(AB) = \text{Tr}(BA)$ ; in particular,  $\langle QAQ^T, QBQ^T \rangle = \langle A, B \rangle$  if  $Q$  is an orthogonal matrix. A well known property of the positive semidefinite cone  $\mathcal{S}_{\geq 0}^n$  is that it is self-dual: for a matrix  $X \in \mathcal{S}^n$ ,  $X \geq 0$  if and only if  $\langle X, Y \rangle \geq 0$  for all  $Y \in \mathcal{S}_{\geq 0}^n$ .

For a matrix  $A \in \mathcal{S}^n$ ,  $\text{diag}(A)$  denotes the vector in  $\mathbb{R}^n$  with entries are the diagonal entries of  $A$  and, for a vector  $a \in \mathbb{R}^n$ ,  $\text{Diag}(a) \in \mathcal{S}^n$  is the diagonal matrix with diagonal entries the entries of  $a$ .

## 2.1 Primal and dual semidefinite programs

### 2.1.1 Primal form

The typical form of a semidefinite program (often abbreviated as SDP) is a maximization problem of the form

$$p^* = \sup_X \{ \langle C, X \rangle : \langle A_j, X \rangle = b_j \ (j \in [m]), \ X \geq 0 \}. \quad (2.1)$$

Here  $A_1, \dots, A_m \in \mathcal{S}^n$  are given  $n \times n$  symmetric matrices and  $b \in \mathbb{R}^m$  is a given vector, they are the *data* of the semidefinite program (2.1). The matrix  $X$  is the *variable*, which is constrained to be positive semidefinite and to lie in the affine subspace

$$\mathcal{W} = \{ X \in \mathcal{S}^n \mid \langle A_j, X \rangle = b_j \ (j \in [m]) \}$$

of  $\mathcal{S}^n$ . The goal is to maximize the linear objective function  $\langle C, X \rangle$  over the *feasible region*

$$\mathcal{F} = \mathcal{S}_{\geq 0}^n \cap \mathcal{W},$$

obtained by intersecting the positive semidefinite cone  $\mathcal{S}_{\geq 0}^n$  with the affine subspace  $\mathcal{W}$ .

A feasible solution  $X \in \mathcal{F}$  is said to be *strictly feasible* if  $X$  is positive definite. The program (2.1) is said to be *strictly feasible* if it admits at least one strictly feasible solution.

One can also handle minimization problems, of the form

$$\inf_X \{ \langle C, X \rangle : \langle A_j, X \rangle = b_j \ (j \in [m]), \ X \geq 0 \}$$

since they can be brought into the above standard maximization form using the fact that  $\inf \langle C, X \rangle = -\sup \langle -C, X \rangle$ .

Note that we write a *supremum* in (2.1) rather than a *maximum*. This is because the optimum value  $p^*$  might not be attained in (2.1). In general,  $p^* \in \mathbb{R} \cup \{\pm\infty\}$ , with  $p^* = -\infty$  if the problem (2.1) is infeasible (i.e.,  $\mathcal{F} = \emptyset$ ) and  $p^* = +\infty$  might occur in which case we say that the problem is unbounded.

We give a small example as an illustration. Consider the problem of minimizing/maximizing  $X_{11}$  over the feasible region

$$\mathcal{F}_a = \left\{ X \in \mathcal{S}^2 : X = \begin{pmatrix} X_{11} & a \\ a & 0 \end{pmatrix} \geq 0 \right\} \text{ where } a \in \mathbb{R} \text{ is a given parameter.}$$

Note that  $\det(X) = -a^2$  for any  $X \in \mathcal{F}_a$ . Hence, if  $a \neq 0$  then  $\mathcal{F}_a = \emptyset$  (the problem is infeasible). Moreover, if  $a = 0$  then the problem is feasible but not strictly feasible. The minimum value of  $X_{11}$  over  $\mathcal{F}_0$  is equal to 0, attained at  $X = 0$ , while the maximum value of  $X_{11}$  over  $\mathcal{F}_0$  is equal to  $\infty$  (the problem is unbounded).

As another example, consider the problem

$$p^* = \inf_{X \in \mathcal{S}^2} \left\{ X_{11} : \begin{pmatrix} X_{11} & 1 \\ 1 & X_{22} \end{pmatrix} \geq 0 \right\}.$$

Then the infimum is  $p^* = 0$  which is reached at the limit when  $X_{11} = 1/X_{22}$  and letting  $X_{22}$  tend to  $\infty$ . So the infimum is not attained.

In the special case when the matrices  $A_j, C$  are diagonal matrices, with diagonals  $a_j, c \in \mathbb{R}^n$ , then the program (2.1) reduces to the linear program (LP):

$$\max \{c^\top x : a_j^\top x = b_j \ (j \in [m]), \ x \geq 0\}.$$

Indeed, let  $x$  denote the vector consisting of the diagonal entries of the matrix  $X$ , so that  $x \geq 0$  if  $X \geq 0$ , and  $\langle C, X \rangle = c^\top x$ ,  $\langle A_j, X \rangle = a_j^\top x$ . Hence semidefinite programming contains linear programming as a special instance.

### 2.1.2 Dual form

The program (2.1) is often referred to as the *primal SDP* in standard form. One can define its *dual SDP*, which takes the form:

$$d^* = \inf_y \left\{ \sum_{j=1}^m b_j y_j = b^\top y : \sum_{j=1}^m y_j A_j - C \geq 0 \right\}. \quad (2.2)$$

Thus the dual program has variables  $y_j$ , one for each linear constraint of the primal program. The positive semidefinite constraint arising in (2.2) is also named a *linear matrix inequality (LMI)*. The following facts relate the primal and dual SDP's. They are simple, but very important.

**Lemma 2.1.1.** *Let  $(X, y)$  be a primal/dual pair of feasible solutions, i.e.,  $X$  is a feasible solution of (2.1) and  $y$  is a feasible solution of (2.2).*

1. **(weak duality)** *We have that  $\langle C, X \rangle \leq b^\top y$  and thus  $p^* \leq d^*$ .*
2. **(complementary slackness)** *Assume that the primal program attains its supremum at  $X$ , that the dual program attains its infimum at  $y$ , and that  $p^* = d^*$ . Then the equalities  $\langle C, X \rangle = b^\top y$  and  $\langle X, \sum_{j=1}^m y_j A_j - C \rangle = 0$  hold.*
3. **(optimality criterion)** *If equality  $\langle C, X \rangle = b^\top y$  holds, then the supremum of (2.1) is attained at  $X$ , the infimum of (2.2) is attained at  $y$  and  $p^* = d^*$ .*

*Proof.* If  $(X, y)$  is a primal/dual pair of feasible solutions, then

$$0 \leq \langle X, \sum_j y_j A_j - C \rangle = \sum_j \langle X, A_j \rangle y_j - \langle X, C \rangle = \sum_j b_j y_j - \langle X, C \rangle = b^\top y - \langle C, X \rangle.$$

The left most inequality follows from the fact that both  $X$  and  $\sum_j y_j A_j - C$  are positive semidefinite and we use the fact that  $\langle A_j, X \rangle = b_j$  to get the second equality. This implies that

$$\langle C, X \rangle \leq p^* \leq d^* \leq b^\top y.$$

The rest of the lemma follows by direct verification.  $\square$



The quantity  $d^* - p^*$  is called the *duality gap*. In general there might be a positive duality gap between the primal and dual SDP's. When there is no duality gap, i.e.,  $p^* = d^*$ , one says that *strong duality* holds, a very desirable situation. This topic and criteria for strong duality will be discussed in detail in the next chapter. For now we only quote the following result on strong duality which will be proved in the next chapter (in the general setting of conic programming).

**Theorem 2.1.2. (Strong duality: no duality gap)** *Consider the pair of primal and dual programs (2.1) and (2.2).*

1. *Assume that the dual program (2.2) is bounded from below ( $d^* > -\infty$ ) and that it is strictly feasible. Then the primal program (2.1) attains its supremum (i.e.,  $p^* = \langle C, X \rangle$  for some  $X \in \mathcal{F}$ ) and there is no duality gap:  $p^* = d^*$ .*
2. *Assume that the primal program (2.1) is bounded from above ( $p^* < \infty$ ) and that it is strictly feasible. Then the dual program (2.2) attains its infimum (i.e.,  $d^* = b^\top y$  for some dual feasible  $y$ ) and there is no duality gap:  $p^* = d^*$ .*

In the rest of this chapter we discuss several examples of semidefinite programs.

## 2.2 Eigenvalue optimization

Given a matrix  $C \in \mathcal{S}^n$ , let  $\lambda_{\min}(C)$  (resp.,  $\lambda_{\max}(C)$ ) denote its smallest (resp., largest) eigenvalue. One can express them (please check it) as follows:

$$\lambda_{\max}(C) = \max_{x \in \mathbb{R}^n \setminus \{0\}} \frac{x^\top C x}{\|x\|^2} = \max_{x \in \mathbb{S}^{n-1}} x^\top C x, \quad (2.3)$$

where  $\mathbb{S}^{n-1} = \{x \in \mathbb{R}^n \mid \|x\| = 1\}$  denotes the unit sphere in  $\mathbb{R}^n$ , and

$$\lambda_{\min}(C) = \min_{x \in \mathbb{R}^n \setminus \{0\}} \frac{x^\top C x}{\|x\|^2} = \min_{x \in \mathbb{S}^{n-1}} x^\top C x. \quad (2.4)$$

(This is known as the Rayleigh principle.) As we now see the largest and smallest eigenvalues can be computed via a semidefinite program. Namely, consider the semidefinite program

$$p^* = \sup \{ \langle C, X \rangle : \text{Tr}(X) = \langle I, X \rangle = 1, X \geq 0 \} \quad (2.5)$$

and its dual program

$$d^* = \inf_{y \in \mathbb{R}} \{ y : yI - C \geq 0 \}. \quad (2.6)$$

In view of (2.3), we have that  $d^* = \lambda_{\max}(C)$ . The feasible region of (2.5) is bounded (all entries of any feasible  $X$  lie in  $[0, 1]$ ) and contains a positive definite matrix (e.g., the matrix  $I_n/n$ ), hence the infimum is attained in (2.6).

Analogously, the program (2.6) is bounded from below (as  $y \geq \lambda_{\max}(C)$  for any feasible  $y$ ) and strictly feasible (pick  $y$  large enough), hence the infimum is attained in (2.6). Moreover there is no duality gap:  $p^* = d^*$ . Here we have applied Theorem 2.1.2. Thus we have shown:

**Lemma 2.2.1.** *The largest and smallest eigenvalues of a symmetric matrix  $C \in \mathcal{S}^n$  can be expressed with the following semidefinite programs:*

$$\lambda_{\max}(C) = \max_{\text{s.t.}} \langle C, X \rangle \quad \text{Tr}(X) = 1, X \geq 0 \quad = \min_{\text{s.t.}} y \quad yI_n - C \geq 0$$

$$\lambda_{\min}(C) = \min_{\text{s.t.}} \langle C, X \rangle \quad \text{Tr}(X) = 1, X \geq 0 \quad = \max_{\text{s.t.}} y \quad C - yI_n \geq 0$$

More generally, also the sum of the  $k$  largest eigenvalues of a symmetric matrix can be computed via a semidefinite program.

**Theorem 2.2.2. (Fan's theorem)** *Let  $C \in \mathcal{S}^n$  be a symmetric matrix with eigenvalues  $\lambda_1 \geq \dots \geq \lambda_n$ . Then the sum of its  $k$  largest eigenvalues is given by any of the following two programs:*

$$\lambda_1 + \dots + \lambda_k = \max_{X \in \mathcal{S}^n} \{ \langle C, X \rangle : \text{Tr}(X) = k, I_n \geq X \geq 0 \}, \quad (2.7)$$

$$\lambda_1 + \dots + \lambda_k = \max_{Y \in \mathbb{R}^{n \times k}} \{ \langle C, YY^T \rangle : Y^T Y = I_k \}. \quad (2.8)$$

There is a simple, elegant proof for this result which relies on a geometric insight about the feasible regions of the two programs (2.7) and (2.8):

$$\mathcal{K}_1 = \{X \in \mathcal{S}^n \mid I \geq X \geq 0, \text{Tr}(X) = k\}, \quad (2.9)$$

$$\mathcal{K}_2 = \{YY^T \mid Y \in \mathbb{R}^{n \times k}, Y^T Y = I_k\}. \quad (2.10)$$

The (non-convex) set  $\mathcal{K}_2$  consists of all projection matrices of rank  $k$  and is clearly contained in the (convex) set  $\mathcal{K}_1$ . As the next lemma shows,  $\mathcal{K}_1$  coincides with the convex hull of  $\mathcal{K}_2$ .

**Lemma 2.2.3.**  *$\mathcal{K}_2$  is the set of extreme points of the convex set  $\mathcal{K}_1$ . Therefore equality  $\mathcal{K}_1 = \text{conv}(\mathcal{K}_2)$  holds.*

*Proof.* The proof uses the following simple observation: For any orthogonal matrix  $P \in \mathcal{O}(n)$ ,  $X \in \mathcal{K}_1$  if and only if  $PXP^T \in \mathcal{K}_1$  and, moreover,  $X$  is an extreme point of  $\mathcal{K}_1$  if and only if  $PXP^T$  is an extreme point of  $\mathcal{K}_1$ . This observation allows us to deal with diagonal matrices and to reduce the lemma to a claim about the extreme points of the following polytope:

$$\mathcal{P} = \{x \in [0, 1]^n : e^T x = k\}. \quad (2.11)$$

Indeed, consider  $X \in \mathcal{S}^n$ , written as  $X = PDP^T$ , where  $P \in \mathcal{O}(n)$ ,  $D$  is the diagonal matrix with the eigenvalues of  $X$  as diagonal entries, and define the

vector  $d = \text{diag}(D) \in \mathbb{R}^n$ . Then,  $X$  belongs to (resp., is an extreme point of)  $\mathcal{K}_1$  if and only if  $D$  belongs to (resp., is an extreme point of)  $\mathcal{K}_1$  or, equivalently,  $d$  belongs to (resp., is an extreme point of)  $\mathcal{P}$ .

Now it suffices to observe that the extreme points of the polytope  $\mathcal{P}$  are the vectors  $d \in \{0, 1\}^n$  with  $e^\top d = k$ . This implies that  $X$  is an extreme point of  $\mathcal{K}_1$  if and only if it has  $k$  non-zero eigenvalues, all equal to 1, which precisely means that  $X \in \mathcal{K}_2$ .  $\square$

We can now conclude the proof of Theorem 2.2.2.

*Proof.* (of Theorem 2.2.2). In program (2.8), we maximize the linear objective function  $\langle C, X \rangle$  over the set  $\mathcal{K}_2$ , while in (2.7) we maximize it over the set  $\mathcal{K}_1$ . In (2.7) we may assume that the maximum is attained at an extreme point of  $\mathcal{K}_1$  which, by Lemma 2.2.3, belongs to  $\mathcal{K}_2$ . Therefore, both programs have the same optimum value, denoted as  $p^*$ . We now show that  $p^* = \lambda_1 + \dots + \lambda_k$ .

Let  $u_1, \dots, u_n$  be an orthonormal set of eigenvectors corresponding to the eigenvalues  $\lambda_1, \dots, \lambda_n$  of  $C$  and let  $Y$  be the  $n \times k$  matrix with columns  $u_1, \dots, u_k$ . Then  $YY^\top \in \mathcal{K}_2$  and  $\langle C, YY^\top \rangle = \sum_{i=1}^k \lambda_i$ , thus showing  $\lambda_1 + \dots + \lambda_k \leq p^*$ .

Denote by  $Q$  the orthogonal matrix with columns  $u_1, \dots, u_n$  and by  $D$  the diagonal matrix with the  $\lambda_i$ 's on the diagonal, so that  $C = QDQ^\top$ . Then  $\langle Q, YY^\top \rangle = \langle D, ZZ^\top \rangle$  after setting  $Z = Y^\top Q$ ; note that  $Z^\top Z = I_k$  and thus  $ZZ^\top \in \mathcal{K}_2$ . Hence  $p^*$  is the maximum value of  $\langle D, M \rangle = \sum_{i=1}^n \lambda_i M_{ii}$  taken over  $M \in \mathcal{K}_2$ . The constraints for  $M \in \mathcal{K}_2$  imply that the vector  $\text{diag}(M)$  belongs to the polytope  $\mathcal{P}$  from (2.11). Therefore the maximum of  $\sum_i \lambda_i M_{ii}$  is at most the maximum of  $\sum_i \lambda_i x_i$  taken over  $x \in \mathcal{P}$ . Now the latter maximum is attained at an extreme point of  $\mathcal{P}$ , from which one derives that it is equal to  $\lambda_1 + \dots + \lambda_k$ . This gives the reverse inequality:  $p^* \leq \lambda_1 + \dots + \lambda_k$ .  $\square$

We mention another result of the same flavor: Given two symmetric matrices  $A, B$ , one can reformulate as a semidefinite program the following optimization problem over orthogonal matrices:

$$\min \{ \text{Tr}(AXBX^\top) : X \in \mathcal{O}(n) \}. \quad (2.12)$$

We already considered this problem in Section 1.7.2 in relation with the Hoffman-Wielandt inequality for eigenvalues (recall Theorem 1.7.4). The semidefinite reformulation uses Kronecker products of matrices (introduced in Section 1.7.6).

**Theorem 2.2.4.** *Let  $A, B \in S^n$ . Then the program (2.12) is equivalent to the semidefinite program*

$$\max \{ \text{Tr}(S) + \text{Tr}(T) : A \otimes B - I_n \otimes T - S \otimes I_n \geq 0 \}. \quad (2.13)$$

Moreover its optimum value is

$$\sum_{i=1}^n \alpha_i \beta_i,$$

where the  $\alpha_i$ 's are the eigenvalues of  $A$  ordered in ascending order:  $\alpha_1 \leq \dots \leq \alpha_n$ , and the  $\beta_i$ 's are the eigenvalues of  $B$  ordered in descending order:  $\beta_1 \geq \dots \geq \beta_n$ .

*Proof.* Let  $D$  be the diagonal matrix whose diagonal entries are the  $\alpha_i$ 's and let  $E$  be the diagonal matrix whose diagonal entries are the  $\beta_i$ 's. As in the proof of Theorem 1.7.4, the program (2.12) is equivalent to

$$\min \{ \text{Tr}(DXEX^\top) : X \in \mathcal{O}(n) \} \quad (2.14)$$

which in turn is equivalent to the linear program (1.2), repeated here for convenience:

$$\max_{s, t \in \mathbb{R}^n} \left\{ \sum_{i=1}^n s_i + \sum_{j=1}^n t_j : \alpha_i \beta_j - s_i - t_j \geq 0 \ \forall i, j \in [n] \right\}. \quad (2.15)$$

We now show that the linear program (2.15) is equivalent to the following semidefinite program

$$\max_{S, T \in \mathcal{S}^n} \{ \text{Tr}(S) + \text{Tr}(T) : E \otimes F - I_n \otimes T - S \otimes I_n \geq 0 \}. \quad (2.16)$$

To see it, let  $S, T$  be feasible for (2.16) and define the vectors  $s = \text{diag}(S)$ ,  $t = \text{diag}(T)$ . Then, as  $E \otimes F$  is a diagonal matrix, the diagonal matrices  $\text{Diag}(s)$  and  $\text{Diag}(t)$  are feasible for (2.16) with the same objective value:  $\text{Tr}(S) + \text{Tr}(T) = \text{Tr}(\text{Diag}(s)) + \text{Tr}(\text{Diag}(t))$ . Now, program (2.16) with the additional condition that  $S, T$  are diagonal matrices can be reformulated as (2.15). Finally, write  $A = PDP^\top$  and  $B = QEQ^\top$  where  $P, Q \in \mathcal{O}(n)$  and observe that

$$(P \otimes Q)(E \otimes F - I_n \otimes T - S \otimes I_n)(P \otimes Q)^\top = A \otimes B - I_n \otimes (QTQ^\top) - (PSP^\top) \otimes I_n.$$

Hence  $S, T$  is feasible for (2.16) if and only if  $S' = PSP^\top$ ,  $T' = QTQ^\top$  is feasible for (2.13), and  $\text{Tr}(S) + \text{Tr}(T) = \text{Tr}(S') + \text{Tr}(T')$ . From this follows the desired equivalence of (2.12) and (2.13). The fact that the optimum value is  $\sum_i \alpha_i \beta_i$  was computed in Theorem 1.7.4.  $\square$

## 2.3 Convex quadratic constraints

Consider a quadratic constraint for a vector  $x \in \mathbb{R}^n$  of the form

$$x^\top A x \leq b^\top x + c, \quad (2.17)$$

where  $A \in \mathcal{S}^n$ ,  $b \in \mathbb{R}^n$  and  $c \in \mathbb{R}$ . In the special case when  $A \geq 0$ , then the feasible region defined by this constraint is convex and it turns out that it can be equivalently defined by a semidefinite constraint.

**Lemma 2.3.1.** *Assume  $A \geq 0$ . Say,  $A = LL^\top$ , where  $L \in \mathbb{R}^{n \times k}$ . Then, for any  $x \in \mathbb{R}^n$ ,*

$$x^\top A x \leq b^\top x + c \iff \begin{pmatrix} I_k & L^\top x \\ x^\top L & b^\top x + c \end{pmatrix} \geq 0.$$

*Proof.* The equivalence follows as a direct application of Lemma 1.7.6: Choose here  $A = I_k$ ,  $B = L^T x \in \mathbb{R}^{k \times 1}$  and  $C = b^T x + c \in \mathbb{R}^{1 \times 1}$ . And take the Schur complement of the submatrix  $I_k$  in the block-matrix on the right hand side.  $\square$

As a direct application, the Euclidean unit ball can be represented by an LMI:

$$\{x \in \mathbb{R}^n : \|x\| \leq 1\} = \left\{x \in \mathbb{R}^n : \begin{pmatrix} 1 & x^T \\ x & I_n \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & I_n \end{pmatrix} + \sum_{i=1}^n x_i \begin{pmatrix} 0 & e_i^T \\ e_i & 0 \end{pmatrix} \geq 0\right\}$$

as well as its homogenization:

$$\mathcal{L}^{n+1} = \{(x, t) \in \mathbb{R}^{n+1} : \|x\| \leq t\} = \left\{x \in \mathbb{R}^n : \begin{pmatrix} t & x^T \\ x & tI_n \end{pmatrix} \geq 0\right\}.$$

So at  $t = t_0$ , we have in the  $x$ -space the ball of radius  $t_0$ . The set  $\mathcal{L}^{n+1}$  is a cone, known as the *second-order cone* (or *Lorentz cone*), to which we will come back in the next chapter.

The fact that one can reformulate linear optimization over the Euclidean ball as a maximization or minimization semidefinite program can be very useful as we will see in the next section.

**Corollary 2.3.2.** *Given  $c \in \mathbb{R}^n$ , the following holds:*

$$\begin{aligned} \min_{\|x\| \leq 1} c^T x &= \min_{x \in \mathbb{R}^n} c^T x \text{ s.t. } \begin{pmatrix} 1 & x^T \\ x & I_n \end{pmatrix} \geq 0 \\ &= \max_{X \in \mathcal{S}^{n+1}} -\text{Tr}(X) \text{ s.t. } X_{0i} = c_i \ (i \in [n]), \ X \geq 0. \end{aligned} \quad (2.18)$$

*Proof.* Apply Lemma 2.3.1 combined with the duality theorem (Theorem 2.1.2).  $\square$

## 2.4 Robust optimization

We indicate here how semidefinite programming comes up when dealing with some robust optimization problems.

Consider the following linear programming problem:

$$\max\{c^T x : a^T x \geq b\},$$

where  $c, a \in \mathbb{R}^n$  and  $b \in \mathbb{R}$  are given data, with just one constraint for simplicity of exposition. In practical applications the data  $a, b$  might be given through experimental results and might not be known exactly with 100% certainty, which is in fact the case in most of the real world applications of linear programming. One may write  $a = a(z)$  and  $b = b(z)$  as functions of an uncertainty parameter  $z$  assumed to lie in a given uncertainty region  $\mathcal{Z} \subseteq \mathbb{R}^k$ . Then one wants to find an optimum solution  $x$  that is *robust* against this uncertainty, i.e., that satisfies

the constraints  $a(z)^\top x \geq b(z)$  for all values of the uncertainty parameter  $z \in \mathcal{Z}$ . That is, solve

$$\max\{c^\top x : a(z)^\top x \geq b(z) \ \forall z \in \mathcal{Z}\}. \quad (2.19)$$

Depending on the set  $\mathcal{Z}$  this problem might have infinitely many constraints. However, for certain choices of the functions  $a(z), b(z)$  and of the uncertainty region  $\mathcal{Z}$ , one can reformulate the problem as a semidefinite programming problem, thus tractable.

Suppose that the uncertainty region  $\mathcal{Z}$  is the unit ball and that  $a(z), b(z)$  are linear functions in the uncertainty parameter  $z = (\zeta_1, \dots, \zeta_k) \in \mathbb{R}^k$ , of the form

$$a(z) = a_0 + \sum_{j=1}^k \zeta_j a_j, \quad b(z) = b_0 + \sum_{j=1}^k \zeta_j b_j \quad (2.20)$$

where  $a_j, b_j \in \mathbb{R}^n$  are known. Then the robust optimization problem (2.19) can be reformulated as a semidefinite programming problem involving the variable  $x \in \mathbb{R}^n$  and a new matrix variable  $Z \in \mathcal{S}^k$ . The proof relies on the result from Corollary 2.3.2, where we made use in a crucial manner of the duality theory for semidefinite programming, for showing the equivalence of both problems in (2.18).

**Theorem 2.4.1.** *Suppose that the functions  $a(z)$  and  $b(z)$  are given by (2.20) and that  $\mathcal{Z} = \{z \in \mathbb{R}^m : \|z\| \leq 1\}$ . Then problem (2.19) is equivalent to the problem:*

$$\begin{aligned} \max_{x \in \mathbb{R}^n, Z \in \mathcal{S}^{k+1}} \quad & c^\top x \quad \text{such that} \quad a_j^\top x - Z_{0j} = b_j \quad (j \in [k]) \\ & a_0^\top x - \text{Tr}(Z) \geq b_0, \quad Z \geq 0. \end{aligned} \quad (2.21)$$

*Proof.* Fix  $x \in \mathbb{R}^n$ , set  $\alpha_j = a_j^\top x - b_j$  for  $j = 0, 1, \dots, k$ , and define the vector  $\alpha = (\alpha_j)_{j=1}^k \in \mathbb{R}^k$  (which depends on  $x$ ). Then the constraints:  $a(z)^\top x \geq b(z) \ \forall z \in \mathcal{Z}$  can be rewritten as

$$\alpha^\top z \geq -\alpha_0 \quad \forall z \in \mathcal{Z}.$$

Therefore, we find the problem of deciding whether  $p^* \geq -\alpha_0$ , where

$$p^* = \min_{\|z\| \leq 1} \alpha^\top z.$$

Now the above problem fits precisely within the setting considered in Corollary 2.3.2. Hence, we can rewrite it using the second formulation in (2.18) – the one in maximization form – as

$$p^* = \max_{Z \in \mathcal{S}^{m+1}} \{-\text{Tr}(Z) : Z_{0j} = \alpha_j \ (j \in [k]), Z \geq 0\}.$$

So, in problem (2.19), we can substitute the condition:  $a(z)^\top x \geq b(z) \ \forall z \in \mathcal{Z}$  by the condition:

$$\exists Z \in \mathcal{S}_{\geq 0}^{m+1} \quad \text{s.t.} \quad -\text{Tr}(Z) \geq -\alpha_0, \quad Z_{0j} = \alpha_j \ (j \in [k]).$$

The crucial fact here is that the quantifier “ $\forall z$ ” has been replaced by the existential quantifier “ $\exists Z$ ”. As problem (2.19) is a maximization problem in  $x$ , it is equivalent to the following maximization problem in the variables  $x$  and  $Z$ :

$$\max_{x \in \mathbb{R}^n, Z \in \mathcal{S}^{m+1}} \{c^\top x : a_0^\top x - \text{Tr}(Z) \geq b_0, a_j^\top x - Z_{0j} = b_j \ (j \in [k])\}$$

(after substituting back in  $\alpha_j$  their expression in terms of  $x$ ).  $\square$

## 2.5 Examples in combinatorial optimization

Semidefinite programs provide a powerful tool for constructing useful convex relaxations for combinatorial optimization problems. We will treat this in detail in a later chapter. For now we illustrate the main idea on the following two examples: finding a maximum independent set and a maximum cut in a graph.

### 2.5.1 The maximum independent set problem

Consider a graph  $G = (V, E)$  with vertex set  $V = [n]$ , the edges are unordered pairs of distinct vertices. A set of nodes (or vertices)  $S \subseteq V$  is said to be *independent* (or *stable*) if it does not contain an edge and the maximum cardinality of an independent set is denoted as  $\alpha(G)$ , known as the *stability number* of  $G$ . The *maximum independent set problem* asks to compute  $\alpha(G)$ . This problem is  $\mathcal{NP}$ -hard.

Here is a simple recipe for constructing a semidefinite programming upper bound for  $\alpha(G)$ . It is based on the following observation: Let  $S$  be an independent set in  $G$  and let  $x \in \{0, 1\}^n$  be its incidence vector, with  $x_i = 1$  if  $i \in S$  and  $x_i = 0$  otherwise. Define the matrix  $X = xx^\top / |S|$ . Then the matrix  $X$  satisfies the following conditions:  $X \geq 0$ ,  $X_{ij} = 0$  for all edges  $\{i, j\} \in E$ ,  $\text{Tr}(X) = 1$ , and  $\langle J, X \rangle = |S|$ . It is therefore natural to consider the following semidefinite program

$$\vartheta(G) = \max_{X \in \mathcal{S}^n} \{\langle J, X \rangle : \text{Tr}(X) = 1, X_{ij} = 0 \ (\{i, j\} \in E), X \geq 0\}, \quad (2.22)$$

whose optimum value  $\vartheta(G)$  is known as the *theta number* of  $G$ . It follows from the above discussion that  $\vartheta(G)$  is an upper bound for the stability number. That is,

$$\alpha(G) \leq \vartheta(G).$$

The dual semidefinite program reads

$$\min_{y \in \mathbb{R}^E, t \in \mathbb{R}} \left\{ t : tI + \sum_{\{i, j\} \in E} y_{ij} E_{ij} - J \geq 0 \right\}, \quad (2.23)$$

and its optimum value is equal to  $\vartheta(G)$  (because (6.13) is strictly feasible and bounded – check it). Here we have used the elementary matrices  $E_{ij}$  introduced in the abstract of the chapter.

We will come back to the theta number in a later chapter. As we will see there, there is an interesting class of graphs for which  $\alpha(G) = \vartheta(G)$ , the so-called *perfect graphs*. For these graphs, the maximum independent set problem can be solved in polynomial time. This result is one of the first breakthrough applications of semidefinite programming obtained in the early eighties.

### 2.5.2 The maximum cut problem

Consider again a graph  $G = (V, E)$  where  $V = [n]$ . Given a subset  $S \subseteq V$ , the *cut*  $\delta_G(S)$  consists of all the edges  $\{i, j\}$  of  $G$  that are cut by the partition  $(S, V \setminus S)$ , i.e., exactly one of the two nodes  $i, j$  belongs to  $S$ . The *maximum cut problem* (or *max-cut*) asks to find a cut of maximum cardinality. This is an  $\mathcal{NP}$ -hard problem.

One can encode the max-cut problem using variables  $x \in \{\pm 1\}^n$ . Assign  $x_i = 1$  to the nodes  $i \in S$  and  $-1$  to the nodes  $i \in V \setminus S$ . Then the cardinality of the cut  $\delta_G(S)$  is equal to  $\sum_{\{i,j\} \in E} (1 - x_i x_j)/2$ . Therefore max-cut can be formulated as

$$\text{max-cut} = \max_{x \in \mathbb{R}^n} \left\{ \sum_{\{i,j\} \in E} (1 - x_i x_j)/2 : x \in \{\pm 1\}^n \right\}. \quad (2.24)$$

Again there is a simple recipe for constructing a semidefinite relaxation for max-cut: Pick a vector  $x \in \{\pm 1\}^n$  (arising in the above formulation of max-cut) and consider the matrix  $X = xx^\top$ . This matrix  $X$  satisfies the following conditions:  $X \geq 0$  and  $X_{ii} = 1$  for all  $i \in [n]$ . Therefore, it is natural to consider the following semidefinite relaxation for max-cut:

$$\text{sdp} = \max_{X \in \mathcal{S}^n} \left\{ \sum_{\{i,j\} \in E} (1 - X_{ij})/2 : X \geq 0, X_{ii} = 1 \ (i \in [n]) \right\}. \quad (2.25)$$

As we will see later this semidefinite program provides a very good approximation for the max-cut problem:  $\text{sdp} \leq 1.13 \cdot \text{max-cut}$ . This is a second breakthrough application of semidefinite programming, obtained in the early nineties.

Let  $L_G \in \mathcal{S}^n$  denote the Laplacian matrix of  $G$ : its  $(i, i)$ th diagonal entry is the degree of node  $i$  in  $G$ , and the  $(i, j)$ th off-diagonal entry is  $-1$  if  $\{i, j\}$  is an edge and 0 otherwise. Note that

$$x^\top L_G x = \sum_{\{i,j\} \in E} (x_i - x_j)^2 \ \forall x \in \mathbb{R}^n, \quad \frac{1}{4} x^\top L_G x = \frac{1}{2} \sum_{\{i,j\} \in E} (1 - x_i x_j) \ \forall x \in \{\pm 1\}^n.$$

The first item shows that  $L_G \geq 0$ , and the second item shows that one can reformulate max-cut using the Laplacian matrix. Analogously one can reformulate



the semidefinite program (2.25) as

$$\text{sdp} = \max \left\{ \frac{1}{4} \langle L_G, X \rangle : X \geq 0, X_{ii} = 1 \ (i \in [n]) \right\}. \quad (2.26)$$

Given a positive semidefinite matrix  $A$ , consider the following quadratic problem

$$\text{opt} = \max \{x^T A x : \|x\|_\infty \leq 1\}. \quad (2.27)$$

where  $\|x\|_\infty = \max_i |x_i|$  is the  $\ell_\infty$ -norm. As we maximize a convex function over the convex set  $[-1, 1]^n$ , the maximum is attained at a vertex, i.e., at a point of  $\{\pm 1\}^n$ . This shows that (2.27) is equivalent to

$$\text{opt} = \max \{x^T A x : x \in \{\pm 1\}^n\}. \quad (2.28)$$

This problem is  $\mathcal{NP}$ -hard – indeed it contains the max-cut problem, obtained when choosing  $A = L_G/4$ .

Note that if we would replace in (2.27) the cube  $[-1, 1]^n$  by the Euclidean unit ball, then we find the problem of computing the largest eigenvalue of  $A$  which, as we saw earlier, can be modeled as a semidefinite program.

Just as for max-cut one can formulate the following semidefinite relaxation of (2.28) (and thus of (2.27)):

$$\text{sdp} = \max \{ \langle A, X \rangle : X \geq 0, X_{ii} = 1 \ \forall i \in [n] \}. \quad (2.29)$$

We will see later that this semidefinite program too gives a good approximation of the quadratic problem (2.27):  $\text{sdp} \leq \frac{\pi}{2} \text{opt}$ .

## 2.6 Examples in geometry

Given vectors  $u_1, \dots, u_n \in \mathbb{R}^k$ , let  $d = (d_{ij})$  denote the vector consisting of their pairwise squared Euclidean distances, i.e.,  $d_{ij} = \|u_i - u_j\|^2$  for all  $i, j \in [n]$ . Thus  $d_{ii} = 0$  for all  $i$ . Now, think of the vectors  $u_i$  as representing the locations of some objects (atoms of a molecule, or sensors in a sensor network). One might be able to determine the pairwise distances  $d_{ij}$  by making some measurements. However, in general, one can determine these distances  $d_{ij}$  only for a subset of pairs, corresponding to the edges of a graph  $G$ . Then the problem arises whether one can reconstruct the locations of the objects (the vectors  $u_i$ ) from these partial measurements (the distances  $d_{ij}$  for the edges  $\{i, j\}$  of  $G$ ).

In mathematical terms, given a graph  $G = (V = [n], E)$  and  $d \in \mathbb{R}_{\geq 0}^E$ , decide whether there exist vectors  $u_1, \dots, u_n \in \mathbb{R}^k$  such that

$$\|u_i - u_j\|^2 = d_{ij} \text{ for all } \{i, j\} \in E.$$

Of course, this problem comes in several flavors. One may search for such vectors  $u_i$  lying in a space of prescribed dimension  $k$ ; then typically  $k = 2, 3$  or

4 would be of interest. This is in fact a hard problem. However, if we relax the bound on the dimension and simply ask for the existence of the  $u_i$ 's in  $\mathbb{R}^k$  for some  $k \geq 1$ , then the problem can be cast as the problem of deciding feasibility of a semidefinite program.

**Lemma 2.6.1.** *Given  $d \in \mathbb{R}_{\geq 0}^E$ , there exist vectors  $u_1, \dots, u_n \in \mathbb{R}^k$  (for some  $k \geq 1$ ) if and only if the following semidefinite program is feasible:*

$$X \geq 0, \quad X_{ii} + X_{jj} - 2X_{ij} = d_{ij} \quad \text{for } \{i, j\} \in E.$$

*Moreover, such vectors exist in the space  $\mathbb{R}^k$  if and only if the above semidefinite program has a feasible solution of rank at most  $k$ .*

*Proof.* Directly, using the fact that  $X \geq 0$  if and only if  $X$  admits a Gram representation  $u_1, \dots, u_n \in \mathbb{R}^k$  (for some  $k \geq 1$ ), i.e.,  $X_{ij} = u_i^\top u_j$  for all  $i, j \in [n]$ . Moreover, the rank of  $X$  is equal to the rank of the system  $\{u_1, \dots, u_n\}$ .  $\square$

Thus arises naturally the problem of finding *low rank* solutions to a semidefinite program. We will come back to this topic in a later chapter.

## 2.7 Examples in algebra

Another, maybe a bit unexpected at first sight, application of semidefinite programming is for testing whether a multivariate polynomial can be written as a sum of squares of polynomials.

First recall a bit of notation.  $\mathbb{R}[x_1, \dots, x_n]$  (or simply  $\mathbb{R}[x]$  for simplicity) denotes the ring of polynomials in  $n$  variables. A polynomial  $p \in \mathbb{R}[x]$  can be written as  $p = \sum_{\alpha} p_{\alpha} x^{\alpha}$ , where  $p_{\alpha} \in \mathbb{R}$  and  $x^{\alpha}$  stands for the monomial  $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ . The sum is finite and the maximum value of  $|\alpha| = \sum_{i=1}^n \alpha_i$  for which  $p_{\alpha} \neq 0$  is the degree of  $p$ . For an integer  $d$ ,  $[x]_d$  denotes the vector consisting of all monomials of degree at most  $d$ , which has  $\binom{n+d}{d}$  entries. Denoting by  $\mathbf{p} = (p_{\alpha})$  the vector of coefficients of  $p$ , we can write

$$p = \sum_{\alpha} p_{\alpha} x^{\alpha} = \mathbf{p}^\top [x]_d. \quad (2.30)$$

**Definition 2.7.1.** *A polynomial  $p$  is said to be a sum of squares (SOS) if  $p$  can be written as a sum of squares of polynomials, i.e.,  $p = \sum_{j=1}^m (q_j)^2$  for some polynomials  $q_j$ .*

It turns out that checking whether  $p$  is SOS can be reformulated via a semidefinite program. Clearly, we may assume that  $p$  has even degree  $2d$  (else  $p$  is not SOS) and the polynomials  $q_j$  arising in a SOS decomposition will have degree at most  $d$ .

Let us now make the following simple manipulation, based on (2.30):

$$\sum_j q_j^2 = \sum_j [x]_d^\top \mathbf{q}_j \mathbf{q}_j^\top [x]_d = [x]_d^\top \left( \sum_j \mathbf{q}_j \mathbf{q}_j^\top \right) [x]_d = [x]_d^\top Q [x]_d,$$

after setting  $Q = \sum_j q_j q_j^\top$ . Having such a decomposition for the matrix  $Q$  amounts to requiring that  $Q$  is positive semidefinite. Therefore, we have just shown that the polynomial  $p$  is SOS if and only if

$$p = [x]_d^\top Q [x]_d \text{ for some matrix } Q \geq 0.$$

Linear conditions on  $Q$  arise by equating the coefficients of the polynomials on both sides in the above identity.

Summarizing, one can test whether  $p$  can be written as a sum of squares by checking the feasibility of a semidefinite program. If  $p$  has degree  $2d$ , this SDP involves a variable matrix  $Q$  of size  $\binom{n+d}{d}$  (the number of monomials of degree at most  $d$ ) and  $\binom{n+2d}{2d}$  (the number of monomials of degree at most  $2d$ ) linear constraints.

One can sometimes restrict to smaller matrices  $Q$ . For instance, if the polynomial  $p$  is homogeneous (i.e., all its terms have degree  $2d$ ), then we may assume without loss of generality that the polynomials  $q_j$  appearing in a SOS decomposition are homogeneous of degree  $d$ . Hence  $Q$  will be indexed by the  $\binom{n+d-1}{d}$  monomials of degree *equal to*  $d$ .

Why bother about sums of squares of polynomials? A good reason is that they can be useful to recognize and certify positive polynomials and to approximate optimization problems dealing with polynomials. Let us just give a glimpse on this.

Suppose that one wants to compute the infimum  $p^{\min}$  of a polynomial  $p$  over the full space  $\mathbb{R}^n$ . In other words, one wants to find the largest scalar  $\lambda$  for which  $p(x) - \lambda \geq 0$  for all  $x \in \mathbb{R}^n$ . This is in general a hard problem. However, if we relax the positivity condition on  $p - \lambda$  and instead require that  $p - \lambda$  is a sum of squares, then it follows from the above considerations that we can compute the maximum  $\lambda$  for which  $p - \lambda$  is SOS using semidefinite programming. This gives a tractable bound  $p^*$  satisfying:  $p^* \leq p^{\min}$ .

In general  $p^*$  might be distinct from  $p^{\min}$ . However in the univariate case ( $n = 1$ ), equality holds:  $p^{\min} = p^*$ . (This will follow from the result in Problem 2.2.) Equality holds also in the quadratic case:  $d = 2$ , and in one exceptional case:  $n = 2$  and  $d = 4$ . This was shown by Hilbert in 1888.

We will return to this topic in a later chapter.

## 2.8 Further reading

A detailed treatment about Fan's theorem (Theorem 2.2.2) can be found in Overton and Womersley [8] and a detailed discussion about Hoffman-Wielandt inequality, Theorem 2.2.4 and applications (e.g. to quadratic assignment) can be found in Anstreicher and Wolkowicz [2].

The recent monograph of Ben-Tal, El Ghaoui and Nemirovski [3] offers a detailed treatment of robust optimization. The result presented in Theorem 2.4.1 is just one of the many instances of problems which admit a robust counterpart which is a tractable optimization problem. Although we formulated it in terms

of semidefinite programming (to fit our discussion), it can in fact be formulated in terms of second-order conic optimization, which admits faster algorithms.

The theta number  $\vartheta(G)$  was introduced in the seminal work of Lovász [3]. A main motivation of Lovász was to give good bounds for the Shannon capacity of a graph, an information theoretic measure of the graph. Lovász succeeded to determine the exact value of the Shannon capacity of  $C_5$ , the circuit on five nodes, by computing  $\vartheta(C_5) = \sqrt{5}$ . This work of Lovász can be considered as the first breakthrough application of semidefinite programming, although the term *semidefinite programming* was coined only later. Chapter 33 of [1] gives a beautiful treatment of this result. The monograph by Grötschel, Lovász and Schrijver [1] treats in detail algorithmic questions related to semidefinite programming and, in particular, to the theta number. Polynomial time solvability based on the ellipsoid method is treated in detail there.

Using semidefinite programming to approximate max-cut was pioneered by the work of Goemans and Williamson [5]. This novel approach and their result had a great impact on the area of combinatorial optimization. It indeed spurred a lot of research activity for getting tight approximations for various problems. This line of research is now also very active in theoretical computer science, where the *unique games conjecture* has been formulated that is directly relevant to the basic semidefinite relaxation (2.25) for max-cut – cf. e.g. the survey by Trevisan [10].

Sums of squares of polynomials are a classical topic in mathematics and they have many applications e.g. to control theory and engineering. In the late 1800s David Hilbert classified the parameters degree/number of variables for which any positive polynomial can be written as a sum of squares of polynomials. He posed the question whether any positive polynomial can be written as a sum of squares of rational functions, known as Hilbert’s 17th problem. This was solved by Artin in 1927, a result which started the field of real algebraic geometry. The survey by Reznick [9] gives a nice overview and historical perspective and the monograph by Delzell and Prestell [4] gives an in-depth treatment of positivity.

## 2.9 Exercises

- 2.1. (a) Formulate the dual SDP of the program (2.7).  
 (b) Give a semidefinite programming formulation for the following problem:

$$\min\{\lambda_1(X) + \dots + \lambda_k(X) : \langle A_j, X \rangle = b_j \ (j \in [m])\},$$

which asks for a matrix  $X \in \mathcal{S}^n$  satisfying a system of linear constraints and for which the sum of the  $k$  largest eigenvalues of  $X$  is minimum.

- 2.2. Let  $p$  be a univariate polynomial.

- (a) Show that  $p$  can be written as a sum of squares if and only if  $p$  is non-negative over  $\mathbb{R}$ , i.e.,  $p(x) \geq 0 \ \forall x \in \mathbb{R}$ .

- (b) Show that if  $p$  is non-negative over  $\mathbb{R}$  then it can be written as sum of two squares.

- 2.3\*\*. (a) Build the dual of the semidefinite programming (2.26) and show that it is equivalent to

$$\frac{n}{4} \min_{u \in \mathbb{R}^n} \{ \lambda_{\max}(\text{Diag}(u) + L_G) : e^T u = 0 \},$$

where  $\text{Diag}(u)$  is the diagonal matrix with diagonal entries  $u_1, \dots, u_n$ .

- (b) Show that the maximum cardinality of a cut is at most

$$\frac{n}{4} \lambda_{\max}(L_G),$$

where  $\lambda_{\max}(L_G)$  is the maximum eigenvalue of the Laplacian matrix of  $G$ .

- (c) Show that the maximum cardinality of a cut in  $G$  is at most

$$\frac{1}{2}|E| - \frac{n}{4} \lambda_{\min}(A_G)$$

where  $A_G$  is the adjacency matrix of  $G$ .

- (d) Show that both bounds in (b) and (c) coincide when  $G$  is a regular graph (i.e., all nodes have the same degree).

- 2.4. Consider the polynomial in two variables  $x$  and  $y$

$$p = x^4 + 2x^3y + 3x^2y^2 + 2xy^3 + 2y^4.$$

- (a) Build a semidefinite program permitting to recognize whether  $p$  can be written as sum of squares.  
 (b) Describe all possible sums of squares decompositions for  $p$ .  
 (c) What can you say about the number of squares needed?

---

## BIBLIOGRAPHY

- [1] M. Aigner and G. Ziegler. *Proofs from THE BOOK*. Springer, 2003.
- [2] K. Anstreicher and H. Wolkowicz. On Lagrangian relaxation of quadratic matrix constraints. *SIAM Journal on Matrix Analysis and its Applications* **22**(1):41–55, 2000.
- [3] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*, Princeton University Press, 2009.
- [4] A. Prestel and C.N. Delzell. *Positive Polynomials - From Hilberts 17th Problem to Real Algebra*. Springer, 2001.
- [5] M.X. Goemans and D. Williamson. Improved approximation algorithms for maximum cuts and satisfiability problems using semidefinite programming. *Journal of the ACM* **42**:1115–1145, 1995.
- [6] M. Grötschel, L. Lovász and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, 1988.
- [7] L. Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory* **IT-25**:1–7, 1979.
- [8] M. Overton and R.S. Womersley. On the sum of the  $k$  largest eigenvalues of a symmetric matrix. *SIAM Journal on Matrix Analysis and its Applications* **13**(1):41–45, 1992.
- [9] B. Reznick. Some concrete aspects of Hilbert’s 17th problem. In *Real Algebraic Geometry and Ordered Structures*. C.N. Delzell and J.J. Madden (eds.), *Contemporary Mathematics* **253**:251–272, 2000.
- [10] L. Trevisan. On Khot’s unique games conjecture. *Bull. Amer. Math. Soc.* **49**:91–111, 2012.

---

---

## CHAPTER 3

---

# DUALITY IN CONIC PROGRAMMING

Traditionally, convex optimization problems are of the form

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{subject to} && f_1(x) \leq 0, \dots, f_N(x) \leq 0, \\ & && a_1^\top x = b_1, \dots, a_M^\top x = b_M, \end{aligned}$$

where the *objective function*  $f_0 : D \rightarrow \mathbb{R}$  and the *inequality constraint functions*  $f_i : D \rightarrow \mathbb{R}$  which are defined on a convex domain  $D \subseteq \mathbb{R}^n$  are *convex*, i.e. their *epigraphs*

$$\text{epi } f_i = \{(x, \alpha) : D \times \mathbb{R} : f_i(x) \leq \alpha\}, \quad i = 0, \dots, N,$$

are convex sets in  $D \times \mathbb{R} \subseteq \mathbb{R}^{n+1}$ . Equivalently, the function  $f_i$  is convex if and only if

$$\forall x, y \in D \quad \forall \alpha \in [0, 1] : f_i((1 - \alpha)x + \alpha y) \leq (1 - \alpha)f_i(x) + \alpha f_i(y).$$

The *equality constraints* are given by vectors  $a_j \in \mathbb{R}^n \setminus \{0\}$  and right hand sides  $b_j \in \mathbb{R}$ . The convex set of *feasible solutions* is the intersection of  $N$  convex sets with  $M$  hyperplanes

$$\bigcap_{i=1}^N \{x \in D : f_i(x) \leq 0\} \cap \bigcap_{j=1}^M \{x \in \mathbb{R}^n : a_j^\top x = b_j\}.$$

The set-up for conic programming is slightly different. We start by considering a fixed convex cone  $K$  lying in the  $n$ -dimensional Euclidean space  $\mathbb{R}^n$ . The

task of conic programming is the following: One wants to maximize (or minimize) a linear function over the feasible region which is given as the intersection of the convex cone  $K$  with an affine subspace:

$$\begin{aligned} & \text{maximize } c^\top x \\ & \text{subject to } x \in K, \\ & \quad a_1^\top x = b_1, \dots, a_m^\top x = b_m. \end{aligned}$$

This differs only slightly from a traditional convex optimization problem: The objective function is linear and feasibility with respect to the inequality constraint functions is replaced by membership in the fixed convex cone  $K$ . In principle, one can transform every convex optimization problem into a conic program. However, the important point in conic programming is that it seems that a vast majority of convex optimization problems which come up in practice can be formulated as conic programs using the three standard cones:

1. the non-negative orthant  $\mathbb{R}_{\geq 0}^n$  – giving linear programming (LP),
2. the second-order cone  $\mathcal{L}^{n+1}$  – giving second-order cone programming (CQP),
3. or the cone of positive semidefinite matrices  $\mathcal{S}_{\geq 0}^n$  – giving semidefinite programming (SDP).

As we will see in the next lecture, these three cones have particular nice analytic properties: They have a self-concordant barrier function which is easy to evaluate. This implies that there are theoretically (polynomial-time) and practically efficient algorithms to solve these standard problems.

In addition to this, the three examples are ordered by their “difficulty”, which can be pictured as

$$\text{LP} \subseteq \text{CQP} \subseteq \text{SDP}.$$

This means that one can formulate every linear program as a conic quadratic program and one can formulate every conic quadratic program as a semidefinite program.

Why do we care about conic programming in general and do not focus on these three most important special cases?

The answer is that conic programming gives a unifying framework to design algorithms, to understand the basic principles of its geometry and duality, and to model optimization problems. Moreover this offers the flexibility of dealing with new cones obtained e.g. by taking direct products of the three standard types of cones.



## 3.1 Fundamental properties

### 3.1.1 Local minimizers are global minimizers

A first fundamental property of convex optimization problems is that every local minimizer is at the same time a global minimizer. A *local minimizer* of the convex optimization problem is a feasible solution  $x \in D$  having the property that there is a positive  $\epsilon$  so that

$$f_0(x) = \inf\{f_0(y) : y \text{ is feasible and } d(x, y) \leq \epsilon\}.$$

Here and throughout we use the notation  $d(x, y)$  to denote the Euclidean distance  $\|x - y\|_2$  between  $x, y \in \mathbb{R}^n$ . To see that local optimality implies global optimality assume that  $x$  is a local but *not* a global minimizer, then there is a feasible solution  $y$  so that  $f_0(y) < f_0(x)$ . Clearly,  $d(x, y) > \epsilon$ . Define  $z \in [x, y]$  by setting

$$z = (1 - \alpha)x + \alpha y, \quad \alpha = \frac{\epsilon}{2d(x, y)},$$

which is a feasible solution because of convexity. Then,  $d(x, z) = \epsilon/2$  and again by convexity

$$f_0(z) \leq (1 - \alpha)f_0(x) + \alpha f_0(y) < f_0(x),$$

which contradicts the fact that  $x$  is a local minimizer.

### 3.1.2 Karush-Kuhn-Tucker condition

A second fundamental property of convex optimization problems is that one has necessary and sufficient conditions for  $x$  being a local (and hence a global) minimizer. Stating and analyzing these kind of conditions is central to the theory of non-linear programming and convex analysis. We just state one fundamental result here without proving it. A proof can be found for instance in the book [2, Chapter 5] by Boyd and Vandenberghe.

We assume that the convex optimization problem satisfies the following condition, known as *Slater's condition*:

*There exists a point  $x \in \text{relint } D$  such that  $f_i(x) < 0$  for all  $i = 1, \dots, N$  and such that  $a_j^\top x = b_j$  for all  $j = 1, \dots, M$ .*

This point is called a *strictly feasible solution* since the inequality constraints hold with strict inequality. Furthermore, we assume that the objective function and that the inequality constraint functions are differentiable. Under these conditions a feasible solution is a global minimizer if and only if the Karush-Kuhn-Tucker (KKT) condition holds: There are  $\lambda_1, \dots, \lambda_N \in \mathbb{R}_{\geq 0}$  and  $\mu_1, \dots, \mu_M \in \mathbb{R}$  so that the following equations are satisfied:

$$\begin{aligned} \lambda_1 f_1(x) &= 0, \dots, \lambda_N f_N(x) = 0, \\ \nabla f_0(x) + \sum_{i=1}^N \lambda_i \nabla f_i(x) + \sum_{j=1}^M \mu_j a_j &= 0. \end{aligned}$$

The KKT-condition is an extension of the method of Lagrange multipliers where one also can consider inequalities instead of only equalities.

## 3.2 Primal and dual conic programs

When defining conic programming we need a “nice” cone  $K$ , satisfying the following properties:  $K$  is closed, convex, pointed, and has a non-empty interior or, equivalently, it is full-dimensional.

### 3.2.1 Primal conic programs

Let  $K \subseteq \mathbb{R}^n$  be a pointed, closed, convex cone with non-empty interior.

**Definition 3.2.1.** Given  $c \in \mathbb{R}^n$ ,  $a_1, \dots, a_m \in \mathbb{R}^n$ , and  $b_1, \dots, b_m \in \mathbb{R}$ , a primal conic program (in standard form) is the following maximization problem:

$$\sup\{c^\top x : x \in K, a_1^\top x = b_1, \dots, a_m^\top x = b_m\},$$

which can also be written in a more compact form as

$$\sup\{c^\top x : x \in K, Ax = b\},$$

where  $A$  is the  $m \times n$  matrix with rows  $a_1^\top, \dots, a_m^\top$  and  $b = (b_1, \dots, b_m)^\top \in \mathbb{R}^m$ .

We say that  $x \in \mathbb{R}^n$  is a *feasible solution (of the primal)* if it lies in the cone  $K$  and if it satisfies the equality constraints. It is a *strictly feasible solution* if it additionally lies in the interior of  $K$ .

Note that we used a supremum here instead of a maximum. The reason is simply that sometimes the supremum is not attained. We shall see examples in Section 3.5.

### 3.2.2 Dual conic programs

The principal problem of duality is to find upper bounds for the primal conic program (a maximization problem), in a systematic, or even mechanical way. This is helpful e.g. in formulating optimality criteria and in the design of efficient algorithms. Duality is a powerful technique, and sometimes translating primal problems into dual problems gives unexpected benefits and insights. To define the dual conic program we need the dual cone  $K^*$ .

**Definition 3.2.2.** Let  $K \subseteq \mathbb{R}^n$  be a cone. The dual cone  $K^*$  of  $K$  is

$$K^* = \{y \in \mathbb{R}^n : y^\top x \geq 0 \text{ for all } x \in K\}.$$

**Lemma 3.2.3.** If  $K$  is a pointed, closed, convex cone with non-empty interior, then the same holds for its dual cone  $K^*$ .

You will prove this in Exercise 3.1. The following property of cones will be useful — you will prove it in Exercise 3.2.

**Lemma 3.2.4.** *Let  $K$  be a closed convex full-dimensional cone. Then we have the equivalence*

$$x \in \text{int } K \iff \forall y \in K^* \setminus \{0\} : y^\top x > 0.$$

**Definition 3.2.5.** *Let*

$$\sup\{c^\top x : x \in K, a_1^\top x = b_1, \dots, a_m^\top x = b_m\} = \sup\{c^\top x : x \in K, Ax = b\}$$

*be a primal conic program. Its dual conic program is the following minimization problem*

$$\inf \left\{ \sum_{j=1}^m y_j b_j : y_1, \dots, y_m \in \mathbb{R}, \sum_{j=1}^m y_j a_j - c \in K^* \right\},$$

*or more compactly,*

$$\inf\{b^\top y : y \in \mathbb{R}^m, A^\top y - c \in K^*\}.$$

We say that  $y \in \mathbb{R}^m$  is a *feasible solution (of the dual)* if  $\sum_{j=1}^m y_j a_j - c \in K^*$ . It is a *strictly feasible solution* if  $\sum_{j=1}^m y_j a_j - c \in \text{int } K^*$ .

### 3.2.3 Geometric interpretation of the primal-dual pair

At first sight, the dual conic program does not look like a conic program, i.e. optimizing a linear function over the intersection of a convex cone by an affine subspace. Although the expression  $z = \sum_{i=1}^m y_i a_i - c$  ranges over the intersection of the convex cone  $K^*$  with an affine subspace, it might be less clear a priori why the objective function  $\sum_{i=1}^m y_i b_i$  has the right form (a linear function in  $z = \sum_{i=1}^m y_i a_i - c$ ).

The following explanation shows how to view the primal and the dual conic program geometrically. This also will bring the dual program into the right form. For this consider the linear subspace

$$L = \{x \in \mathbb{R}^n : a_1^\top x = 0, \dots, a_m^\top x = 0\},$$

and its orthogonal complement

$$L^\perp = \left\{ \sum_{j=1}^m y_j a_j \in \mathbb{R}^n : y_1, \dots, y_m \in \mathbb{R} \right\}.$$

We may assume that there exists a point  $x_0 \in \mathbb{R}^n$  satisfying  $Ax_0 = b$  for, if not, the primal conic program would not have a feasible solution. Note then that

$$b^\top y = x_0^\top A^\top y = x_0^\top \left( \sum_{j=1}^m a_j y_j \right) = x_0^\top \left( \sum_{j=1}^m a_j y_j - c \right) + x_0^\top c.$$

Therefore, the primal conic program can be written as

$$\sup\{c^\top x : x \in K \cap (x_0 + L)\}$$

and the dual conic program as

$$c^\top x_0 + \inf\{x_0^\top z : z \in K^* \cap (-c + L^\perp)\}.$$

Now both the primal and the dual conic programs have the right form and the symmetry between the primal and the dual conic program becomes more clear.

What happens when one builds the dual of the dual? Then one gets a conic program which is equivalent to the primal. This is due to the following lemma.

**Lemma 3.2.6.** *Let  $K \subseteq \mathbb{R}^n$  be a closed convex cone. Then,  $(K^*)^* = K$ .*

*Proof.* The inclusion  $K \subseteq (K^*)^*$  is easy to verify using the definition only. For the reverse inclusion, one needs the separation theorem (Lemma 1.5.2). Let  $x \in \mathbb{R}^n \setminus K$ . Then  $\{x\}$  and  $K$  can be separated by a hyperplane of the form  $H = \{z \in \mathbb{R}^n : c^\top z = 0\}$  for some  $c \in \mathbb{R}^n \setminus \{0\}$ . Say,  $K \subseteq H^+ = \{z : c^\top z \geq 0\}$  and  $c^\top x < 0$ . The inclusion  $K \subseteq H^+$  shows that  $c \in K^*$  and then the inequality  $c^\top x < 0$  shows that  $x \notin (K^*)^*$   $\square$

### 3.3 Examples

Now we specialize the cone  $K$  to the first three examples of Section 1.5. These three examples are useful for a huge spectrum of applications.

#### 3.3.1 Linear programming (LP)

A conic program where  $K$  is the non-negative orthant  $\mathbb{R}_{\geq 0}^n$  is a *linear program*. We write a primal linear program (in standard form) as

$$\sup\{c^\top x : x \geq 0, a_1^\top x = b_1, \dots, a_m^\top x = b_m\} = \sup\{c^\top x : x \geq 0, Ax = b\}.$$

The non-negative orthant is self-dual:  $(\mathbb{R}_{\geq 0}^n)^* = \mathbb{R}_{\geq 0}^n$ . The dual linear program is

$$\inf \left\{ \sum_{j=1}^m b_j y_j : y_1, \dots, y_m \in \mathbb{R}, \sum_{j=1}^m y_j a_j - c \geq 0 \right\} = \inf\{b^\top y : A^\top y - c \geq 0\}.$$

In the case when the problems are not unbounded we could replace the supremum/infimum by maximum/minimum. This is because we are optimizing a linear function over a polyhedron, which is equivalent to optimizing over its set of extreme points, and any polyhedron has finitely many extreme points.

### 3.3.2 Conic quadratic programming (CQP)

A conic program where  $K$  is the second-order cone  $\mathcal{L}^{n+1}$  is a *conic quadratic program*. We write a primal conic quadratic program (in standard form) as

$$\sup\{(c, \gamma)^\top(x, t) : (x, t) \in \mathcal{L}^{n+1}, (a_1, \alpha_1)^\top(x, t) = b_1, \dots, (a_m, \alpha_m)^\top(x, t) = b_m\}.$$

Here  $(x, t)$  stands for the (column) vector in  $\mathbb{R}^{n+1}$  obtained by appending a new entry  $t \in \mathbb{R}$  to  $x \in \mathbb{R}^n$ , we use this notation to emphasize the different nature of the vector's components. Recall the definition of the second-order cone  $\mathcal{L}^{n+1}$ :

$$(x, t) \in \mathcal{L}^{n+1} \text{ if and only if } \|x\|_2 \leq t.$$

The second-order cone is self-dual, too — you will show this in Exercise 3.3

$$(\mathcal{L}^{n+1})^* = \mathcal{L}^{n+1}.$$

The dual conic quadratic program is

$$\inf \left\{ \sum_{j=1}^m y_j b_j : y_1, \dots, y_m \in \mathbb{R}, \sum_{j=1}^m y_j (a_j, \alpha_j) - (c, \gamma) \in \mathcal{L}^{n+1} \right\}.$$

This can be written in a nicer and more intuitive form using the Euclidean norm. Define the matrix  $B \in \mathbb{R}^{n \times m}$  which has  $a_i$  as its  $i$ -th column, and the vectors  $b = (b_j)_{j=1}^m$ ,  $\alpha = (\alpha_j)_{j=1}^m$  and  $y = (y_j)_{j=1}^m$  in  $\mathbb{R}^m$ . Then the dual conic quadratic program can be reformulated as

$$\inf \{ b^\top y : y \in \mathbb{R}^m, \|By - c\|_2 \leq \alpha^\top y - \gamma \}.$$

### 3.3.3 Semidefinite programming (SDP)

A conic program where  $K$  is the cone of semidefinite matrices  $\mathcal{S}_{\geq 0}^n$  is a *semidefinite program*. We write a primal semidefinite program (in standard form) as

$$\sup\{\langle C, X \rangle : X \geq 0, \langle A_1, X \rangle = b_1, \dots, \langle A_m, X \rangle = b_m\}.$$

We have already seen earlier that the cone of semidefinite matrices is self-dual:

$$(\mathcal{S}_{\geq 0}^n)^* = \mathcal{S}_{\geq 0}^n.$$

The dual semidefinite program is

$$\inf \left\{ \sum_{j=1}^m y_j b_j : y_1, \dots, y_m \in \mathbb{R}, \sum_{j=1}^m y_j A_j - C \geq 0 \right\}.$$

Engineers and applied mathematicians like to call an inequality of the form  $\sum_{i=1}^m y_i A_i - C \geq 0$  a *linear matrix inequality (LMI)* between the parameters  $y_1, \dots, y_m$ . It is a convenient way to express a convex constraint posed on the vector  $y = (y_1, \dots, y_m)^\top$ .

### 3.4 Duality theory

Duality is concerned with understanding the relation between the primal conic program and the dual conic program. We denote the supremum of the primal conic program by  $p^*$  and the infimum of the dual conic program by  $d^*$ . What is the relation between  $p^*$  and  $d^*$ ? As we see in the next theorem it turns out that in many cases one has equality  $p^* = d^*$  and that the supremum as well as the infimum are attained. In these cases duality theory can be very useful because sometimes it is easier to work with the dual problem instead of the primal problem.

**Theorem 3.4.1.** *Suppose we are given a pair of primal and dual conic programs. Let  $p^*$  be the supremum of the primal and let  $d^*$  be the infimum of the dual.*

1. **(weak duality)** *Suppose  $x$  is a feasible solution of the primal conic program, and  $y$  is a feasible solution of the dual conic program. Then,*

$$c^\top x \leq b^\top y.$$

*In particular  $p^* \leq d^*$ .*

2. **(complementary slackness)** *Suppose that the primal conic program attains its supremum at  $x$ , and that the dual conic program attains its infimum at  $y$ , and that  $p^* = d^*$ . Then*

$$\left( \sum_{i=1}^m y_i a_i - c \right)^\top x = 0.$$

3. **(optimality criterion)** *Suppose that  $x$  is a feasible solution of the primal conic program, and  $y$  is a feasible solution of the dual conic program, and equality*

$$\left( \sum_{i=1}^m y_i a_i - c \right)^\top x = 0$$

*holds. Then the supremum of the primal conic program is attained at  $x$  and the infimum of the dual conic program is attained at  $y$ .*

4. **(strong duality; no duality gap)** *If the dual conic program is bounded from below and if it is strictly feasible, then the primal conic program attains its supremum and there is no duality gap:  $p^* = d^*$ .*

*If the primal conic program is bounded from above and if it is strictly feasible, then the dual conic programs attains its infimum and there is no duality gap.*

Before the proof one more comment about the usefulness of weak duality: Suppose you want to solve a primal conic program. If the oracle of Delft, gives you  $y$ , then it might be wise to check whether  $\sum_{i=1}^m y_i a_i - c$  lies in  $K^*$ . If so, then this gives immediately an upper bound for  $p^*$ .

The difference  $d^* - p^*$  is also called the *duality gap* between the primal conic program and dual conic program.

One last remark: If the dual conic program is not bounded from below:  $d^* = -\infty$ , then weak duality implies that  $p^* = -\infty$ , i.e., the primal conic program is infeasible.

*Proof.* The proof of **weak duality** is important and simple. It reveals the origin of the definition of the dual conic program: We have

$$\sum_{j=1}^m y_j b_j = \sum_{j=1}^m y_j (a_j^\top x) = \left( \sum_{j=1}^m y_j a_j \right)^\top x \geq c^\top x,$$

where the last inequality is implied by  $\sum_{i=1}^m y_i a_i - c \in K^*$  and  $x \in K$ .

Now **complementary slackness** and the **optimality criterion** immediately follow from this.

**Strong duality** needs considerably more work. It suffices to prove the first statement (since the second one follows using the symmetry between the primal and dual problems). So we assume that  $d^* > -\infty$  and that the dual program has a strict feasible solution. Using these assumptions we will construct a primal feasible solution  $x^*$  with  $c^\top x^* \geq d^*$ . Then, weak duality implies  $p^* = d^*$  and hence  $x^*$  is a maximizer of the primal conic program.

Consider the set

$$M = \left\{ \sum_{j=1}^m y_j a_j - c : y \in \mathbb{R}^m, b^\top y \leq d^* \right\}.$$

If  $b = 0$  then  $d^* = 0$  and setting  $x^* = 0$  proves the result immediately. Hence we may assume that there is an index  $i$  so that  $b_i$  is not zero, and then  $M$  is not empty. We first claim that

$$M \cap \text{int } K^* = \emptyset.$$

For suppose not. Then there exists  $y \in \mathbb{R}^m$  such that  $\sum_{j=1}^m y_j a_j - c \in \text{int } K^*$  and  $y^\top b \leq d^*$ . Assume without loss of generality that  $b_1 < 0$ . Then for a small enough  $\epsilon > 0$  one would have  $(y_1 + \epsilon)a_1 + \sum_{j=2}^m y_j a_j - c \in K^*$  with  $(y_1 + \epsilon)b_1 + \sum_{j=2}^m y_j b_j < y^\top b \leq d^*$ . This contradicts the fact that  $d^*$  is the infimum of the dual conic program.

Since  $M$  and  $K^*$  are both convex sets whose relative interiors do not intersect, we can separate them by an affine hyperplane, according to Theorem 1.3.8. Hence, there is a non-zero vector  $x \in \mathbb{R}^n$  so that

$$\sup\{x^\top z : z \in M\} \leq \inf\{x^\top z : z \in K^*\}. \quad (3.1)$$

We shall use this point  $x$  to construct a maximizer of the primal conic program which we do in three steps.

**First step:**  $x \in K$ .

To see it, it suffices to show that

$$\inf_{z \in K^*} x^\top z \geq 0, \quad (3.2)$$

as this implies that  $x \in (K^*)^* = K$ . We show the inequality by contradiction. Suppose there is a vector  $z \in K^*$  with  $x^\top z < 0$ . Then, for any positive  $\lambda$ , the vector  $\lambda z$  lies in the convex cone  $K^*$ . Making  $\lambda$  extremely large drives  $x^\top \lambda z$  towards  $-\infty$ . But we reach a contradiction since, by (3.1), the infimum of  $x^\top z$  over  $z \in K^*$  is lower bounded since  $M \neq \emptyset$ .

**Second step:** There exists  $\mu > 0$  so that  $a_j^\top x = \mu b_j$  ( $j \in [m]$ ) and  $x^\top c \geq \mu d^*$ .

Since  $0 \in K^*$  we also have that the infimum of (3.2) is at most 0. So we have shown that the infimum of (3.2) is equal to 0. Therefore, by (3.1),  $\sup_{z \in M} x^\top z \leq 0$ . In other words, by the definition of  $M$ , for any  $y \in \mathbb{R}^m$ ,

$$y^\top b \leq d^* \implies x^\top \left( \sum_{j=1}^m y_j a_j - c \right) \leq 0$$

or, equivalently,

$$y^\top b \leq d^* \implies \sum_{j=1}^m y_j (x^\top a_j) \leq x^\top c.$$

This means that the halfspace  $\{y : y^\top b \leq d^*\}$  is contained into the halfspace  $\{y : y^\top (x^\top a_j)_j \leq x^\top c\}$ . Hence their normal vectors  $b$  and  $(x^\top a_j)_j$  point in the same direction. In other words there exists a scalar  $\mu \geq 0$  such that

$$x^\top a_j = \mu b_j \quad (j = 1, \dots, m), \quad \mu d^* \leq x^\top c.$$

It suffices now to verify that  $\mu$  is positive. Indeed suppose that  $\mu = 0$ . Then, on the one hand, we have that  $x^\top c \geq 0$ . On the other hand, using the assumption that the conic dual program is **strictly feasible**, there exists  $\bar{y} \in \mathbb{R}^m$  such that  $\sum_j \bar{y}_j a_j - c \in \text{int } K$ . This implies

$$0 < \left( \sum_{j=1}^m \bar{y}_j a_j - c \right)^\top x = -c^\top x,$$

where strict inequality follows from  $\sum_j \bar{y}_j a_j - c \in \text{int } K$  and  $x \in K \setminus \{0\}$  (use here Lemma 3.2.4). This gives  $c^\top x < 0$ , a contradiction.

**Third step:**  $x^* = x/\mu$  is a maximizer of the primal conic program.

This follows directly from the fact that  $x^*$  is a primal feasible solution (since we saw above that  $x^* \in K$  and  $a_j^\top x^* = b_j$  for  $j \in [m]$ ) with  $c^\top x^* \geq d^*$ .  $\square$



### 3.5 Some pathological examples

If you know linear programming and its duality theory you might wonder why do we always write sup and inf instead of max and min and why do we care about strictly feasibility in Theorem 3.4.1. Why doesn't strong duality always hold? Here are some examples of semidefinite programs showing that we indeed have to be more careful.

#### 3.5.1 Dual infimum not attained

Consider the semidefinite program

$$p^* = \sup \left\{ \left\langle \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}, X \right\rangle : X \geq 0, \left\langle \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, X \right\rangle = 1, \left\langle \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, X \right\rangle = 0 \right\}$$

and its dual

$$d^* = \inf \left\{ y_1 : y_1 \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + y_2 \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} = \begin{pmatrix} y_1 & 1 \\ 1 & y_2 \end{pmatrix} \geq 0 \right\}.$$

In this example,  $p^* = d^* = 0$  and the supremum is attained in the primal, but the infimum is not attained in the dual. Note indeed that the primal is not strictly feasible (since  $X_{22} = 0$  for any feasible solution).

#### 3.5.2 Positive duality gap

There can be a duality gap between the primal and the dual conic programs. Consider the primal semidefinite program with data matrices

$$C = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix},$$

and  $b_1 = 0, b_2 = 1$ . It reads

$$p^* = \sup \{ -X_{11} - X_{22} : X_{11} = 0, 2X_{13} + X_{22} = 1, X \geq 0 \}$$

and its dual reads

$$d^* = \inf \left\{ y_2 : y_1 A_1 + y_2 A_2 - C = \begin{pmatrix} y_1 + 1 & 0 & y_2 \\ 0 & y_2 + 1 & 0 \\ y_2 & 0 & 0 \end{pmatrix} \geq 0 \right\}.$$

Then any primal feasible solution satisfies  $X_{13} = 0, X_{22} = 1$ , so that the primal optimum value is equal to  $p^* = -1$ , attained at the matrix  $X = E_{22}$ . Any dual feasible solution satisfies  $y_2 = 0$ , so that the dual optimum value is equal to  $d^* = 0$ , attained at  $y = 0$ . Hence there is a positive duality gap:  $d^* - p^* = 1$ .

### 3.6 Strong and weak infeasibility

Consider the following two conic programming systems

$$Ax = b, x \in K, \quad (3.3)$$

$$\sum_{j=1}^m y_j a_j = A^T y \in K^*, b^T y < 0. \quad (3.4)$$

Clearly, if (3.3) has a solution then (3.4) has no solution: If  $x$  is feasible for (3.3) and  $y$  is feasible for (3.4) then

$$0 \leq (A^T y)^T x = y^T Ax = y^T b < 0,$$

giving a contradiction. When  $K$  is the non-negative orthant then the converse also holds: If (3.3) has no solution then (3.4) has a solution. This fact follows by applying the separation theorem (Lemma 1.5.2). Indeed, assume that (3.3) has no solution. Then  $b$  does not belong to the cone generated by the columns of  $A$ . By Lemma 1.5.2, there exists a hyperplane, having normal  $y \in \mathbb{R}^m$ , separating  $\{b\}$  and this cone spanned by column vectors. So we have the inequalities  $A^T y \geq 0$  and  $y^T b < 0$ . This shows that  $y$  is feasible for (3.4). We just proved Farkas' lemma for linear programming.

**Theorem 3.6.1. (Farkas' lemma for linear programming)**

Given  $A \in \mathbb{R}^{m \times n}$  and  $b \in \mathbb{R}^m$ , exactly one of the following two alternatives holds:

- (1) Either the linear system  $Ax = b, x \geq 0$  has a solution,
- (2) Or the linear system  $A^T y \geq 0, b^T y < 0$  has a solution.

For general conic programming, it is not true that infeasibility of (3.3) implies feasibility of (3.4). As an illustration, consider the following semidefinite systems:

$$\langle E_{11}, X \rangle = 0, \langle E_{12}, X \rangle = 1, X \geq 0, \quad (3.5)$$

$$y_1 E_{11} + y_2 E_{12} \geq 0, y_2 < 0, \quad (3.6)$$

which are both infeasible.

However, one can formulate the following analogous, although weaker, theorem of alternatives, which needs some strict feasibility condition.

**Theorem 3.6.2.** Let  $K \subseteq \mathbb{R}^n$  be a full dimensional, pointed, closed and convex cone, let  $A \in \mathbb{R}^{m \times n}$  with rows  $a_1^T, \dots, a_m^T$  and let  $b \in \mathbb{R}^m$ . Assume that the system  $Ax = b$  has a solution  $x_0$ . Then exactly one of the following two alternatives holds:

- (1) Either there exists  $x \in \text{int } K$  such that  $Ax = b$ .
- (2) Or there exists  $y \in \mathbb{R}^m$  such that  $\sum_{j=1}^m y_j a_j = A^T y \in K^* \setminus \{0\}, b^T y \leq 0$ .

*Proof.* Again one direction is clear: If  $x \in \text{int } K$  satisfies  $Ax = b$  and  $y$  satisfies  $A^\top y \in K^* \setminus \{0\}$  and  $b^\top y \leq 0$ , then we get  $0 \leq (A^\top y)^\top x = y^\top Ax = y^\top b \leq 0$ , implying  $(A^\top y)^\top x = 0$ . This gives a contradiction since  $x \in \text{int } K$  and  $A^\top y \in K^* \setminus \{0\}$  (recall Lemma 3.2.4).

Assume now that the system in (1) has no solution. By assumption, the affine space  $L = \{x : Ax = b\}$  is not empty, as  $x_0 \in L$ . Define the linear space

$$\mathcal{L} = \{x : Ax = 0\} = \{x : a_1^\top x = 0, \dots, a_m^\top x = 0\}$$

so that  $L = \mathcal{L} + x_0$ . By assumption,  $L \cap \text{int } K = \emptyset$ . By the separation theorem (Theorem 1.3.8), there exists a hyperplane separating  $L$  and  $\text{int } K$ : There exists a non-zero vector  $c \in \mathbb{R}^n$  and a scalar  $\beta$  such that

$$\forall x \in K : c^\top x \geq \beta \quad \text{and} \quad \forall x \in L : c^\top x \leq \beta.$$

Then  $\beta \leq 0$  (as  $0 \in K$ ) and  $c \in K^*$  (as  $c^\top tx \geq \beta$  for all  $x \in K$  and  $t > 0$ , which implies that  $c^\top x \geq 0$ ). Moreover, for any  $x \in \mathcal{L}$  and any scalar  $t \in \mathbb{R}$ , we have that  $c^\top (tx + x_0) \leq \beta$  which implies  $c^\top x = 0$ . Therefore  $c \in \mathcal{L}^\perp$  and thus  $c$  is a linear combination of the  $a_j$ 's, say  $c = \sum_{j=1}^m y_j a_j = A^\top y$  for some  $y = (y_j) \in \mathbb{R}^m$ . So we already have that  $A^\top y \in K^* \setminus \{0\}$ . Finally,  $y^\top b = y^\top Ax_0 = c^\top x_0 \leq \beta \leq 0$  (as  $x_0 \in L$ ).  $\square$

Consider again the above example: the system (3.5) is not strictly feasible, and indeed there is a feasible solution to (3.6) after replacing the condition  $y_2 < 0$  by  $y_2 \leq 0$  and adding the condition  $y_1 E_{11} + y_2 E_{12} \neq 0$ .

We now further investigate the situation when the primal system (3.3) is infeasible. According to the above discussion, there are two possibilities:

1. Either (3.4) is feasible: There exists  $y \in \mathbb{R}^m$  such that  $\sum_{j=1}^m y_j a_j \in K^*$  and  $b^\top y < 0$ . Then we say that the system (3.3) is *strongly infeasible*.
2. Or (3.4) is not feasible.

As we will show below, this second alternative corresponds to the case when the system (3.3) is “weakly infeasible”, which roughly means that it is infeasible but any small perturbation of it becomes feasible. Here is the exact definition.

**Definition 3.6.3.** *The system  $Ax = b$ ,  $x \in K$  is weakly infeasible if it is infeasible and, for any  $\epsilon > 0$ , there exists  $x \in K$  such that  $\|Ax - b\| \leq \epsilon$ .*

For instance, the system (3.5) is weakly infeasible: For any  $\epsilon > 0$  the perturbed system  $\langle E_{11}, X \rangle = \epsilon$ ,  $\langle E_{12}, X \rangle = 1$ ,  $X \geq 0$  is feasible.

**Theorem 3.6.4.** *Consider the two systems (3.3) and (3.4). Assume that the system (3.3) is infeasible. Then exactly one of the following two alternatives holds.*

- (1) *Either (3.3) is strongly infeasible: There exists  $y \in \mathbb{R}^m$  such that  $b^\top y < 0$  and  $\sum_{j=1}^m y_j a_j \in K^*$ .*

(2) Or (3.3) is weakly infeasible: For every  $\epsilon > 0$  there exists  $x \in K$  satisfying  $\|Ax - b\| \leq \epsilon$ .

*Proof.* Assume that (3.3) is not strongly infeasible. Then the two convex sets  $\{y : A^T y \in K^*\}$  and  $\{y : b^T y < 0\}$  are disjoint. By the separation theorem (Theorem 1.3.8) there exists a non-zero vector  $c \in \mathbb{R}^m$  such that

$$\inf\{c^T y : A^T y \in K^*\} \geq \sup\{c^T y : b^T y < 0\}.$$

As  $0 \in K^*$  the infimum is at most 0. Hence,  $b^T y < 0$  implies  $c^T y \leq 0$ . This implies that  $c = \lambda b$  for some positive  $\lambda$  and, up to rescaling, we can assume that  $c = b$ . Therefore,

$$\sum_{j=1}^m a_j y_j \in K^* \implies b^T y \geq 0. \quad (3.7)$$

We show that (3.3) is weakly infeasible. For this consider the following program, where we have two new variables  $z, z' \in \mathbb{R}^m$ :

$$p^* = \inf_{x \in \mathbb{R}^n, z, z' \in \mathbb{R}^m} \{e^T z + e^T z' : Ax + z - z' = b, x \in K, z, z' \in \mathbb{R}_{\geq 0}^m\}, \quad (3.8)$$

where  $e = (1, \dots, 1)^T$  is the all-ones vector. It suffices now to show that the infimum of (3.8) is equal to 0, since this implies directly that (3.3) is weakly infeasible. For this consider the dual program of (3.8), which can be written as (check it)

$$d^* = \sup_{y \in \mathbb{R}^m} \{b^T y : -A^T y \in K^*, -e \leq y \leq e\}. \quad (3.9)$$

Clearly the primal (3.8) is strictly feasible and  $d^* \geq 0$  (since  $y = 0$  is feasible). Moreover,  $d^* \leq 0$  by (3.7). Hence  $d^* = 0$  and thus  $p^* = d^* = 0$  since there is no duality gap (applying Theorem 3.4.1).  $\square$

Of course the analogous result holds for the dual conic program (which follows using symmetry between primal/dual programs).

**Theorem 3.6.5.** Assume that the system

$$\sum_{j=1}^m y_j a_j - c \in K^* \quad (3.10)$$

is infeasible. Then exactly one of the following two alternatives holds.

- (1) Either (3.10) is strongly infeasible: There exists  $x \in K$  such that  $Ax = 0$  and  $c^T x > 0$ .
- (2) Or (3.10) is weakly infeasible: For every  $\epsilon > 0$  there exist  $y \in \mathbb{R}^m$  and  $z \in K^*$  such that  $\|(\sum_{j=1}^m y_j a_j - c) - z\| \leq \epsilon$ .

### 3.7 More on the difference between linear and conic programming

We have already seen above several differences between linear programming and semidefinite programming: there might be a duality gap between the primal and dual programs and the supremum/infimum might not be attained even though they are finite. We point out some more differences regarding rationality and bit size of optimal solutions.

In the classical bit (Turing machine) model of computation an integer number  $p$  is encoded in binary notation, so that its bit size is  $\log p + 1$  (logarithm in base 2). Rational numbers are encoded as two integer numbers and the bit size of a vector or a matrix is the sum of the bit sizes of its entries.

Consider a linear program

$$\max\{c^T x : Ax = b, x \geq 0\} \quad (3.11)$$

where the data  $A, b, c$  is *rational*-valued. From the point of view of computability this is a natural assumption and it would be desirable to have an optimal solution which is also rational-valued. A fundamental result in linear programming asserts that this is indeed the case: If program (3.11) has an optimal solution, then it has a *rational* optimal solution  $x \in \mathbb{Q}^n$ , whose bit size is polynomially bounded in terms of the bit sizes of  $A, b, c$ .

On the other hand it is easy to construct instances of semidefinite programming where the data are rational valued, yet there is no rational optimal solution. For instance, the following program

$$\max \left\{ x : \begin{pmatrix} 1 & x \\ x & 2 \end{pmatrix} \geq 0 \right\}$$

attains its maximum at  $x = \pm\sqrt{2}$ .

Consider now the semidefinite program, with variables  $x_1, \dots, x_n$ ,

$$\inf \left\{ x_n : \begin{pmatrix} 1 & 2 \\ 2 & x_1 \end{pmatrix} \geq 0, \begin{pmatrix} 1 & x_{i-1} \\ x_{i-1} & x_i \end{pmatrix} \geq 0 \text{ for } i = 2, \dots, n \right\}.$$

Then any feasible solution satisfies  $x_n \geq 2^{2^n}$ . Hence the bit-size of an optimal solution is exponential in  $n$ , thus exponential in terms of the bit-size of the data.

### 3.8 Further reading

Conic programs, especially linear programs, conic quadratic programs, and semidefinite programs are the central topic in the text book of Ben-Tal and Nemirovski [3]. There also many interesting engineering applications (synthesis of filters and antennas, truss topology design, robust optimization, optimal control, stability analysis and synthesis, design of chips) are covered. This book

largely overlaps with Nemirovski's lecture notes [5] which are available online. A nutshell version of these lecture notes is Nemirovski's plenary talk "Advances in convex optimization: conic programming" at the International Congress of Mathematicians in Madrid 2006 for which a paper and a video is available online: [6]. It is astonishing how much material Nemirovski covers in only 60 minutes.

A second excellent text book on convex optimization is the book by Boyd and Vandenberghe [2] (available online). Here the treated applications are: approximation and fitting, statistical estimation, and geometric problems. Videos of Boyd's course held at Stanford can also be found there.

The duality theory for linear programming which does not involve duality gaps is explained in every book on linear programming. For example, Schrijver [7, Chapter 7] is a good source.

### 3.9 Historical remarks

The history of conic programming is difficult to trace. Only recently researchers recognized that they give a unifying framework for convex optimization.

In 1956, Duffin in a short paper "Infinite programs" [3] introduced conic programs. His approach even works in infinite dimensions and he focused on these cases. However, the real beginning of conic programming seems to be 1993 when the book "Interior-Point Polynomial Algorithms in Convex Optimization" by Yurii Nesterov and Arkadi Nemirovski was published. There they described for the first time a unified theory of polynomial-time interior point methods for convex optimization problems based on their conic formulations. Concerning the history of conic programs they write:

Duality for convex program involving "non-negativity constraints" defined by a general-type convex cone in a Banach space is a relatively old (and, possibly, slightly forgotten by the mathematical programming community) part of convex analysis (see, e.g. [ET76]). The corresponding general results, as applied to the case of conic problems (i.e., finite-dimensional problems with general-type non-negativity constraints and *affine* functional constraints), form the contents of §3.2. To our knowledge, in convex analysis, there was no special interest to conic problems, and consequently to the remarkable symmetric form of the aforementioned duality in this particular case. The only previous result in spirit of this duality known to us is the dual characterization of the Lovasz capacity number  $\theta(\Gamma)$  of a graph (see [Lo79]).

### 3.10 Exercises

3.1 Let  $K \subseteq \mathbb{R}^n$  be a cone and let  $K^*$  be its dual cone.

(a) Show that  $K^*$  is a closed convex cone.

- (b) If  $K$  is pointed, closed, convex and full-dimensional, show that the same holds for  $K^*$ .

3.2 Let  $K$  be a closed convex full dimensional cone. Show that

$$x \in \text{int } K \iff y^\top x > 0 \quad \forall y \in K^* \setminus \{0\}.$$

- 3.3 (a) For the Lorentz cone, show that  $(\mathcal{L}^{n+1})^* = \mathcal{L}^{n+1}$ .  
 (b) Determine the dual cone of the cone of copositive matrices.
- 3.4 Consider the following location problem: We are given  $N$  locations in the plane  $x_1, \dots, x_N \in \mathbb{R}^2$ . Find a point  $y \in \mathbb{R}^2$  which minimizes the sum of the distances to the  $N$  locations:

$$\min_{y \in \mathbb{R}^2} \sum_{i=1}^N d(x_i, y).$$

- (a) Formulate this problem as a conic program using the cone

$$\mathcal{L}^{2+1} \times \mathcal{L}^{2+1} \times \dots \times \mathcal{L}^{2+1}.$$

- (b) Determine its dual.  
 (c) Is there a duality gap?

---

## BIBLIOGRAPHY

- [1] A. Ben-Tal, A. Nemirovski, *Lectures on Modern Convex Optimization*, SIAM 2001.
- [2] S. Boyd, L. Vandenberghe, *Convex optimization*, Cambridge, 2004.  
<http://www.stanford.edu/~boyd/cvxbook/>
- [3] R.J. Duffin, *Infinite programs*, pages 157–170 in *Linear Equalities and Related Systems* (A.W. Tucker (ed.)), Princeton University Press, 1956.
- [4] M. Grötschel, L. Lovász, A. Schrijver, *Geometric Algorithms in Combinatorial Optimization*, Springer, 1988.
- [5] A. Nemirovski, *Lectures on Modern Convex Optimization*,  
[http://www2.isye.gatech.edu/~nemirovs/Lect\\_ModConvOpt.pdf](http://www2.isye.gatech.edu/~nemirovs/Lect_ModConvOpt.pdf)
- [6] A. Nemirovski, *Advances in convex optimization: Conic programming*, pages 413–444 in: *Proceedings of International Congress of Mathematicians, Madrid, August 22-30, 2006, Volume 1* (M. Sanz-Sol, J. Soria, J.L. Varona, J. Verdera, Eds.), European Mathematical Society Publishing House, 2007.  
Paper: [http://www.icm2006.org/proceedings/Vol\\_I/21.pdf](http://www.icm2006.org/proceedings/Vol_I/21.pdf)  
Video: <http://www.icm2006.org/video/> (Eighth Session)
- [7] A. Schrijver, *Theory of linear and integer programming*, John Wiley & Sons, 1986.



---

---

# CHAPTER 4

---

## INTERIOR POINT METHODS

In this lecture we consider the problem of solving a conic program numerically. First, we recall the situation. Let  $K \subseteq \mathbb{R}^n$  be a pointed, closed, convex cone with non-empty interior. Given are  $c \in \mathbb{R}^n$ ,  $a_1, \dots, a_m \in \mathbb{R}^n$ , and  $b_1, \dots, b_m \in \mathbb{R}$ . The primal conic program in standard form is the following maximization problem:

$$\sup\{c^\top x : x \in K, a_1^\top x = b_1, \dots, a_m^\top x = b_m\}.$$

Since the beginning of the 1990's the theory of efficient interior point methods was developed which basically says that if the cone  $K$  is “nice” (this can be made mathematically precise: a sufficient condition is the existence of a self-concordant barrier function which is computable in polynomial time; for the details we refer to the literature), then there exists a polynomial time algorithm which solves the conic program. Solving in polynomial time means that one can in polynomially many steps approximate an optimal solution within any desired precision where the precision is part of the input.

Here we only sketch the rough idea of interior point methods. The idea is to provide only some background knowledge without giving many details. [This is just enough to implement a program which solves a small conic program with a few variables for instance.] We will ignore many, many technical details: How to guarantee a polynomial time running time? How to implement a method which is numerically stable? Going through all the details (very fascinating applied mathematics!) fills a complete advanced course, namely the LNMB course “Interior point methods”.

For the details we refer to the comprehensive books of Nesterov and Nemirovski [5] and of Boyd and Vandenberghe [2] and to the literature given in Section 4.5.

First we present the classical barrier method. The principal ideas developed there form the backbone of the modern polynomial time interior point methods.

Then, we look at the most important properties of the central path of the primal-dual pair of a semidefinite program.

## 4.1 Classical barrier methods

To explain the basic idea of interior point method we need two ingredients: Newton's method for equality constrained minimization and barrier functions.

### 4.1.1 Newton's method

We start by recalling Newton's method for unconstrained minimization. Newton's method is an iterative method for finding roots of equations in one or more dimensions. It is one of the most important algorithms in numerical analysis and scientific computing. In convex optimization it can be used to find minimizers of convex differentiable functions. The Newton method is also the fundamental algorithm for the design of fast interior point algorithms.

#### Unconstrained minimization

Newton's method is quite general. It is natural to define it in the setting of Banach spaces. Chapter XVIII of the book "Functional analysis in normed spaces" by L.V. Kantorovich and G.P. Akilov is a classical resource for this which also includes the first thorough analysis of the convergence behavior of Newton's method. Nowadays every comprehensive book on numerical analysis contains a chapter stating explicit conditions for the convergence speed of Newton's method.

To keep it as simple and concrete as possible we define it here only for  $\mathbb{R}^n$ . Let  $\Omega$  be an open set of  $\mathbb{R}^n$  and let  $f : \Omega \rightarrow \mathbb{R}$  be a strictly convex, differentiable function. The Taylor approximation of the function  $f$  around the point  $a$  is

$$f(a + x) = \left( f(a) + \nabla f(a)^\top x + \frac{1}{2} x^\top \nabla^2 f(a) x \right) + \text{h.o.t.},$$

where  $\nabla f(a) \in \mathbb{R}^n$  is the *gradient* of  $f$  at  $a$  with entries

$$\nabla f(a) = \left( \frac{\partial}{\partial x_1} f(a), \dots, \frac{\partial}{\partial x_n} f(a) \right)^\top,$$

and where  $\nabla^2 f(a) \in \mathbb{R}^{n \times n}$  is the *Hessian matrix* of  $f$  at  $a$  with entries

$$[\nabla^2 f(a)]_{ij} = \frac{\partial^2}{\partial x_i \partial x_j} f(a),$$

and where h.o.t. stands for "higher order terms". Since the function is strictly convex, the Hessian matrix is positive definite,  $\nabla^2 f(a) \in \mathcal{S}_{>0}^n$ . By  $q : \mathbb{R}^n \rightarrow \mathbb{R}$

we denote the quadratic function which we get by truncating the above Taylor approximation

$$q(x) = f(a) + \nabla f(a)^\top x + \frac{1}{2} x^\top \nabla^2 f(a) x.$$

This is a strictly convex quadratic function and so it has a unique minimizer  $x^* \in \mathbb{R}^n$  which can be determined by setting the gradient of  $q$  to zero:

$$\begin{aligned} 0 &= \nabla q(x^*) \\ &= \left( \frac{\partial}{\partial x_1} q(x^*), \dots, \frac{\partial}{\partial x_n} q(x^*) \right)^\top \\ &= \nabla f(a) + \nabla^2 f(a) x^*. \end{aligned}$$

Hence, we find the unique minimizer  $x^*$  of  $q$  by solving a system of linear equations

$$x^* = -(\nabla^2 f(a))^{-1} \nabla f(a).$$

Now Newton's method is based on approximating the function  $f$  locally at a starting point  $a$  by the quadratic function  $q$ , finding the minimizer (the *Newton direction*)  $x^*$  of the quadratic function, updating the starting point to  $a + x^*$  and repeating this until the desired accuracy is reached:

**repeat**

$$\begin{aligned} x^* &\leftarrow -(\nabla^2 f(a))^{-1} \nabla f(a) \\ a &\leftarrow a + x^* \end{aligned}$$

**until** a stopping criterion is fulfilled.

The following fact about Newton's method are important.

First the good news: If the starting point is close to the minimizer, then the Newton method converges quadratically (for instance the series  $n \mapsto \frac{1}{10^{2^n}}$  converges quadratically to its limit 0), i.e. in every step the number of accurate digits is multiplied by a constant number.

However, if the starting point is not close to the minimizer or if the function is close to being not strictly convex, then Newton's method does not converge well. Consider for example the convex but not strictly convex univariate function  $f(z) = 1/4 z^4 - z$ . Then  $f'(z) = z^3 - 1$  and  $f''(z) = 3z^2$ . So if one starts the Newton iteration at  $a = 0$ , one immediately is in trouble: division by zero. If one starts at  $a = -\sqrt[3]{1/2}$ , then one can perform a Newton step and one is in trouble again, etc. Figure 4.1.1 shows the fractal structure which is behind Newton's method for solving the equation  $f'(z) = z^3 - 1 = 0$  in the complex number plane. One has similar figures for other functions.

This pure Newton method is an idealization and sometimes it cannot be performed at all because it can very well happen, that  $a + x^* \notin \Omega$ . One can circumvent these problems by replacing the Newton step  $a \leftarrow a + x^*$  by a *damped Newton step*  $a \leftarrow a + \theta x^*$  with some step size  $\theta > 0$  which is chosen

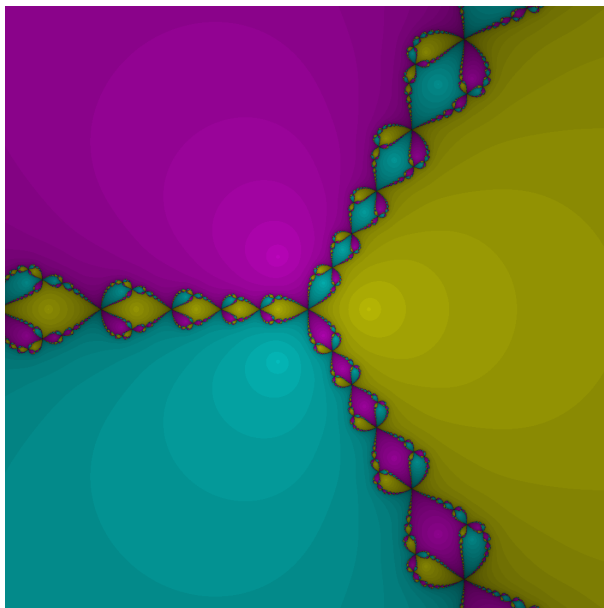


Figure 4.1: Newton fractal of  $z^3 - 1 = 0$ . The three colors indicate the region of attraction for the three roots. The shade of the color indicates the number of steps needed to come close to the corresponding root. (Source: wikipedia).

to ensure e.g. that  $a + \theta x^* \in \Omega$ . Choosing the right  $\theta$  using a line search can be done in many ways. A popular choice is backtracking line search using the Armijo-Goldstein condition.

Let us discuss stopping criteria a bit: One possible stopping criterion is for example if the norm of the gradient is small, i.e. for some predefined positive  $\epsilon$  we do the iteration until

$$\|\nabla f(a)\|^2 \leq \epsilon. \quad (4.1)$$

We now derive a stopping criterion in the case when the function  $f$  is not only strictly convex but also *strongly convex*. This means that there is a positive constant  $m$  so that the smallest eigenvalue of all Hessian matrices of  $f$  is at least  $m$ :

$$\forall a \in \Omega : \lambda_{\min}(\nabla^2 f(a)) \geq m.$$

By the Lagrange form of the Taylor expansion we have

$$\forall a, a+x \in \Omega \exists \xi \in [a, a+x] : f(a+x) = f(a) + \nabla f(a)^\top x + \frac{1}{2} x^\top \nabla^2 f(\xi) x$$

and the strong convexity of  $f$  together with the variational characterization of

the the smallest eigenvalue, which says that

$$\lambda_{\min}(\nabla^2 f(\xi)) = \min_{x \in \mathbb{R}^n \setminus \{0\}} \frac{x^\top \nabla^2 f(\xi) x}{\|x\|^2},$$

gives

$$f(a+x) \geq f(a) + \nabla f(a)^\top x + \frac{1}{2} m \|x\|^2.$$

Consider the function of the right hand side

$$x \mapsto f(a) + \nabla f(a)^\top x + \frac{1}{2} m \|x\|^2.$$

It is a convex quadratic function with gradient

$$x \mapsto \nabla f(a) + mx,$$

hence its minimum is attained at

$$x^* = -\frac{1}{m} \nabla f(a).$$

So we have for the minimum  $\mu^*$  of  $f$

$$\begin{aligned} \mu^* &\geq f(a) + \nabla f(a)^\top \left(-\frac{1}{m} \nabla f(a)\right) + \frac{1}{2} m \left\| \frac{1}{m} \nabla f(a) \right\|^2 \\ &= f(a) - \frac{1}{2m} \|\nabla f(a)\|^2, \end{aligned}$$

which says that whenever the stopping criterion (4.1) is fulfilled we know that  $f(a)$  and  $\mu^*$  are at most  $\epsilon/(2m)$  apart. Of course, the drawback of this consideration is that one has to know or estimate the constant  $m$  in advance which is often not easy. Nevertheless the consideration at least shows that the stopping criterion is sensible.

### Equality-constrained minimization

In the next step we show how to modify Newton's method if we want to find the minimum of a strictly convex, differentiable function  $f : \Omega \rightarrow \mathbb{R}$  in an affine subspace given by the equations

$$a_1^\top x = b_1, a_2^\top x = b_2, \dots, a_m^\top x = b_m,$$

where  $a_1, \dots, a_m \in \mathbb{R}^n$  and  $b_1, \dots, b_m \in \mathbb{R}$ .

We define the Lagrange function

$$L(x, \lambda_1, \dots, \lambda_m) = f(x) + \sum_{i=1}^m \lambda_i a_i^\top x,$$

and the method of *Lagrange multipliers* says that if a point  $y^*$  lies in the affine space

$$a_1^\top y^* = b_1, \dots, a_m^\top y^* = b_m,$$

then it is the unique minimizer of  $f$  if and only if

$$\nabla L(y^*) = 0.$$

To find this point  $y^*$  we approximate the function  $f$  using the Taylor approximation around the point  $a$  by

$$q(x) = f(a) + \nabla f(a)^\top x + \frac{1}{2} x^\top \nabla^2 f(a) x$$

and solve the linear system (in the variables  $x^*$  and  $\lambda_1, \dots, \lambda_m$ )

$$\begin{aligned} a_1^\top (a + x^*) &= b_1, \dots, a_m^\top (a + x^*) = b_m \\ \nabla f(a) + \nabla^2 f(a) x^* + \sum_{i=1}^m \lambda_i a_i &= 0 \end{aligned}$$

to find the Newton direction  $x^*$ . Then we can do the same Newton iterations using damped Newton steps as in the case of unconstrained optimization.

#### 4.1.2 Barrier method

In this section it will be more convenient to consider the following minimization problem instead of the original maximization problem (which is completely equivalent to the maximization problem by switching the sign of the vector  $c$ )

$$\inf \{ c^\top x : x \in K, a_1^\top x = b_1, \dots, a_m^\top x = b_m \}.$$

Using Newton's method for equality constrained minimization we know how to deal with the minimization problem

$$\inf \{ c^\top x : a_1^\top x = b_1, \dots, a_m^\top x = b_m \}.$$

Now we have to answer the question: How do we deal with the constraint  $x \in K$ ? The idea will be to start with a point  $x$  lying in the interior of  $K$  and lying in the affine subspace defined by the  $m$  equations and then apply Newton's method for equality constrained minimization always assuring that the next point will lie in the interior of  $K$ . For this we add to the objective function  $c^\top x$  a barrier function  $\phi(x)$  so that we want to minimize

$$c^\top x + \phi(x) \tag{4.2}$$

instead of  $c^\top x$ . The ideal barrier function would be

$$\phi(x) = \begin{cases} 0, & \text{if } x \in K, \\ \infty, & \text{otherwise.} \end{cases}$$

When we minimize (4.2) we will never consider points not lying in  $K$ . This “almost” works, but Newton’s method is not applicable. The solution is to replace the ideal barrier function by a barrier function  $\phi$  which is a function that is strictly convex and has the property that

$$x \rightarrow \partial K \implies \phi(x) \rightarrow \infty.$$

Then, Newton’s method becomes applicable.

**Example 4.1.1.** *Examples for barrier functions:*

- $K = \mathbb{R}_{\geq 0}^n$ :  $\phi(x) = -\ln(x_1 \cdots x_n)$ .
- $K = \mathcal{L}^{n+1}$ :  $\phi(x, t) = -\ln(t^2 - x_1^2 - \cdots - x_n^2)$ .
- $K = \mathcal{S}_{\geq 0}^n$ :  $\phi(X) = -\ln \det X$ .  
We have  $\nabla \phi(X) = -X^{-1}$  and  $(\nabla^2 \phi(X))H = X^{-1}HX^{-1}$ .

So for a positive parameter  $t > 0$  we can solve

$$\inf \{t(c^\top x) + \phi(x) : a_1^\top x = b_1, \dots, a_m^\top x = b_m\}.$$

using Newton’s method. The optimal solution  $x(t)$  of this minimization problem is called the *central path*. One can show that if  $t$  tends to infinity then  $x(t)$  tends to an optimal solution of the original problem. In the next section we show this in the case of semidefinite programming. Solving  $x(t)$  for large  $t$  is computationally expensive since Newton’s method does not converge fast enough when we start from a point which is not close to the optimal solution. Now the idea of the barrier method is to find  $x(t)$ ’s successively for increasing values of  $t$ . Then one can use the old  $x(t)$  as a starting point for the next Newton method and making use of its quadratic convergence.

In summary the barrier method has the following scheme:

**input:**

objective function  $c$ , constraints  $a_1, \dots, a_m, b_1, \dots, b_m$ ,  
interior point  $x \in \text{int } K$  with  $a_1^\top x = b_1, \dots, a_m^\top x = b_m$ ,  
parameter  $t$ , parameter  $\mu$  (for example  $\mu = 10$ )

**repeat**

compute  $x(t)$  by Newton’s method starting from  $x$   
 $x \leftarrow x(t)$   
 $t \leftarrow \mu t$

**until** a stopping criterion is fulfilled.

### 4.1.3 Finding a starting point

We are still left with the problem of finding a first interior point  $x$  which we need as the input for the previous algorithm. In the case of semidefinite programs the

following approach works: We simply solve another semidefinite program (this is called Phase 1 of the algorithm) for which every symmetric matrix which lies in the affine subspace provides an interior point

$$\inf\{\lambda : X + \lambda I \in \mathcal{S}_{\geq 0}^n, \langle A_1, X \rangle = b_1, \dots, \langle A_m, X \rangle = b_m\},$$

where  $I$  denotes the identity matrix. For this problem it is easy to find a strictly feasible solution: One computes a matrix  $Y$  in the affine subspace  $\langle Y, A_j \rangle = b_j$  and determines its smallest eigenvalue  $\lambda_{\min}(Y)$ . Hence, the matrix  $X = Y + (\epsilon - \lambda_{\min}(Y))I$  is a strictly feasible solution for every  $\epsilon > 0$ . Then we can start to minimize  $\lambda$ . If we find a matrix  $X$  is so that  $\lambda$  is negative, it can be used as a starting point for Phase 2, the original problem. If no such negative  $\lambda$  exists, the original problem is infeasible.

## 4.2 Central path of a semidefinite program

In this section we want to study the central path of a semidefinite program in more detail. These properties give the first ideas for developing a polynomial time interior point algorithm. They also show that the barrier method indeed converges to the right values when  $t$  tends to infinity and they even give the rate of convergence.

In the following we consider a primal-dual pair of a semidefinite program where both the primal and the dual are strictly feasible. Then by strong duality in Theorem 3.4.1 the primal attains the supremum, the dual attains the infimum and there is no duality gap. Let us recall the geometric formulation of the primal-dual pair of a semidefinite program

$$\begin{aligned} & \max\{\langle C, X \rangle : X \geq 0, \langle A_1, X \rangle = b_1, \dots, \langle A_m, X \rangle = b_m\} \\ & = \min \left\{ \sum_{i=1}^m y_i b_i : y_1, \dots, y_m \in \mathbb{R}, \sum_{i=1}^m y_i A_i - C \geq 0 \right\}. \end{aligned}$$

which is (see Section 3.2.3)

$$\begin{aligned} & \max\{\langle C, X \rangle : X \geq 0, X \in X_0 + L\} \\ & = \langle X_0, C \rangle + \min\{\langle X_0, Y \rangle : Y \geq 0, Y \in -C + L^\perp\}, \end{aligned}$$

with linear subspace

$$L = \{X \in \mathcal{S}^n : \langle A_1, X \rangle = 0, \dots, \langle A_m, X \rangle = 0\},$$

and matrix  $X_0 \in \mathcal{S}^n$  with  $\langle A_i, X_0 \rangle = b_i$  for  $i = 1, \dots, m$ .

Let  $t > 0$  be a positive parameter. Consider the strictly convex functions

$$\begin{aligned} P_t(X) &= -t\langle C, X \rangle - \ln \det X, \\ D_t(Y) &= t\langle X_0, Y \rangle - \ln \det Y. \end{aligned}$$



Because the original primal-dual pair is strictly feasible one can show (with some effort) that  $P_t$  attains a unique minimizer  $X(t)$  on the affine subspaces  $X_0 + L$  which is strictly feasible for the primal, and that  $D_t$  attains a unique minimizer  $Y(t)$  on  $-C + L^\perp$  which is strictly feasible for the dual. Hence, this defines the *primal-dual central path*  $(X(t), Y(t))$ .

This primal-dual central path has many nice properties. Some of them are given in the following theorem.

**Theorem 4.2.1.** *For every  $t > 0$  we have the augmented optimality condition*

$$X(t)Y(t) = \frac{1}{t}I. \quad (4.3)$$

*Furthermore, the primal dual central path measures the duality gap between the solutions  $X(t)$  and  $Y(t)$ :*

$$\langle X_0, C \rangle + \langle X_0, Y(t) \rangle - \langle C, X(t) \rangle = \frac{n}{t}.$$

*Proof.* Using Lagrangian multipliers we see (write down the condition explicitly) that a matrix  $X^*$  is the unique minimizer of the strictly convex function  $P_t$  if and only if

$$X^* > 0, \quad X^* \in X_0 + L, \quad \text{and} \quad -tC - (X^*)^{-1} \in L^\perp.$$

In the same way,  $Y^*$  is the unique minimizer of  $D_t$  if and only if

$$Y^* > 0, \quad Y^* \in -C + L^\perp, \quad \text{and} \quad tX_0 - (Y^*)^{-1} \in L.$$

Hence,  $\frac{1}{t}X(t)^{-1}$  is a strictly feasible solution of the dual, and  $\frac{1}{t}Y(t)^{-1}$  is a strictly feasible solution of the primal. The gradient of  $D_t$  at  $\frac{1}{t}X(t)^{-1}$  equals

$$\nabla D_t \left( \frac{1}{t}X(t)^{-1} \right) = tX_0 - \left( \frac{1}{t}X(t)^{-1} \right)^{-1} = tX_0 - tX(t) \in L,$$

Hence, by the characterization of the unique minimizer of  $P_t$  we have  $Y(t) = \frac{1}{t}X(t)^{-1}$ . In the same way one shows symmetrically that  $X(t) = \frac{1}{t}Y(t)^{-1}$ . This implies the first statement.

The second statement follows easily from the first: Let  $y_1, \dots, y_m$  be so that  $\sum_{i=1}^m y_i b_i = \langle X_0, C \rangle + \langle X_0, Y(t) \rangle$ , then

$$\begin{aligned} & \langle X_0, C \rangle + \langle X_0, Y(t) \rangle - \langle C, X(t) \rangle \\ &= \sum_{i=1}^m y_i b_i - \langle C, X(t) \rangle \\ &= \sum_{i=1}^m y_i \langle A_i, X(t) \rangle - \langle C, X(t) \rangle \\ &= \langle Y(t), X(t) \rangle \\ &= \text{Tr} \left( \frac{1}{t}I \right) \\ &= \frac{n}{t}. \end{aligned}$$

□

Compare (4.3) to the optimality condition in Theorem 3.4.1. In particular, it shows that if  $t \rightarrow \infty$ , then  $X(t)$  converges to a primal optimal solution and  $Y(t)$  converges to a dual optimal solution. Another important point for the analysis of interior point algorithms is that the theorem gives the rate of convergence which is proportional to  $\frac{1}{t}$ .

## 4.3 Software

One very good thing about conic programs such as linear programs, convex quadratic programs, and semidefinite programs is that they can be solved efficiently in **theory** and in **practice**. That they can be solved efficiently in theory means that they can be solved in polynomial time to any given precision. That they can be solved efficiently in practice means that there are software packages available which can be used to solve these problems up to some decent sizes.

ILOG CPLEX is known to be a high-performance mathematical programming solver for linear programming, mixed integer programming and quadratic programming. It can be used to solve very large, real-world optimization problems. ILOG CPLEX contains interior point methods for linear programming as well as for convex quadratic programming (but no semidefinite programming). It is free for academic use.

<http://www.ibm.com/software/integration/optimization/cplex-optimizer/>

Semidefinite program solvers are currently slightly less powerful but at least they can solve problems of moderate size involving matrices having size  $1000 \times 1000$ .

One semidefinite programming solver which is easy to use is CVXOPT by Joachim Dahl and Lieven Vandenbergh:

<http://abel.ee.ucla.edu/cvxopt/userguide/index.html>

It is also part of sage. Sage is a free open-source mathematics software system licensed under the GPL. It combines the power of many existing open-source packages into a common python-based interface. In particular it is not difficult to install.

<http://www.sagemath.org/>

Many more software packages for semidefinite programs can be found for example on the NEOS server for optimization:

[http://neos.mcs.anl.gov/neos/solvers/sdp:csdp/SPARSE\\_SDPA.html](http://neos.mcs.anl.gov/neos/solvers/sdp:csdp/SPARSE_SDPA.html)

Here one can also submit the optimization problem online. This has the advantage that one does not have to install the software locally. The input format is explained here:

[http://plato.asu.edu/ftp/sdpa\\_format.txt](http://plato.asu.edu/ftp/sdpa_format.txt)

Essentially one has to specify the matrix sizes and the nonzero entries of the matrices  $C$ ,  $A_i$  and the values of  $b_i$ . One important hint! Note that the role of the primal and dual are “switched” in the documentation.

## 4.4 Historical remarks

Looking at the milestones of the history of mathematical programming shows that interior point methods for conic programs can be seen as the result of the development of efficient, polynomial-time algorithms.

**1947** Dantzig invented the simplex algorithm for linear programming. The simplex algorithm works extremely good in practice, but until today nobody really understands why (although there are meanwhile good theoretical indications). It is fair to say that the simplex algorithm is one of the most important algorithms invented in the last century.

**1972** Klee and Minty found a linear program for which the simplex algorithm is extremely slow (when one uses Dantzig’s most-negative-entry pivoting rule): It uses exponentially many steps.

**1979** Khachian invented the ellipsoid method for linear programming which runs in polynomial time. It is a great theoretical algorithm but until today it did not have any practical impact.

**1984** Karmakar showed that one can use interior-point methods for designing a polynomial-time algorithm for linear programming. Nowadays, interior-point methods can compete with the simplex algorithm.

**1994** Nesterov and Nemirovski generalized Karmarkar’s result to conic programming with the use of self-concordant barrier functions.

**since 1994** Every day conic programming becomes more useful (in theory and practice).

It is fair to say that during the last twenty years there has been a revolution in mathematical optimization based on the development of efficient interior point algorithms for convex optimization problems.

Margaret H. Wright begins her survey “The interior-point revolution in optimization: History, recent developments, and lasting consequences” [10] with:

REVOLUTION:

- (i) a sudden, radical, or complete change;
- (ii) a fundamental change in political organization, especially the overthrow or renunciation of one government or ruler and the substitution of another.

It can be asserted with a straight face that the field of continuous optimization has undergone a revolution since 1984 in the sense of the

first definition and that the second definition applies in a philosophical sense: Because the interior-point presence in optimization today is ubiquitous, it is easy to lose sight of the magnitude and depth of the shifts that have occurred during the past twenty years. Building on the implicit political metaphor of our title, successful revolutions eventually become the status quo.

The interior-point revolution, like many other revolutions, includes old ideas that are rediscovered or seen in a different light, along with genuinely new ideas. The stimulating interplay of old and new continues to lead to increased understanding as well as an ever-larger set of techniques for an ever-larger array of problems, familiar and heretofore unexplored. Because of the vast size of the interior-point literature, it would be impractical to cite even a moderate fraction of the relevant references, but more complete treatments are mentioned throughout. The author regrets the impossibility of citing all important work individually.

## 4.5 Further reading

There are quite some books on interior point methods. The classical barrier method is developed in the book [3] by Fiacco and McCormick. The standard reference is the book [5] by Nesterov and Nemirovski which is not completely easy to read. Boyd and Vandenberghe [2] and Ye [11] as well as [3, Chapter 6] are very helpful. Then, the books [6] by Renegar and [7] by Roos, Terlaky, Vial consider interior point methods for linear programs. There are some surveys available: Nemirovski, Todd [4], Vandenberghe, Boyd [9], Todd [8].

---

## BIBLIOGRAPHY

- [1] A. Ben-Tal, A. Nemirovski, *Lectures on Modern Convex Optimization*, SIAM 2001.
- [2] S. Boyd, L. Vandenberghe, *Convex optimization*, Cambridge, 2004.  
<http://www.stanford.edu/~boyd/cvxbook/>
- [3] A.V. Fiacco, G.P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley & Sons, 1968.
- [4] A.S. Nemirovski, M.J. Todd, *Interior-point methods for optimization*, Acta Numerica (2008), 191-234.  
<http://www2.isye.gatech.edu/~nemirovs/Published.pdf>
- [5] Y. Nesterov, A. Nemirovski, *Interior-point polynomial methods in convex programming*, SIAM, 1994.
- [6] J. Renegar *A mathematical view of interior-point methods in convex optimization*, SIAM, 2001.
- [7] C. Roos, T. Terlaky, J.-Ph. Vial, *Theory and Algorithms for Linear Optimization: An Interior Point Approach*, John Wiley & Sons, 1997.
- [8] M.J. Todd, *Semidefinite optimization*, Acta Numerica **10** (2001), 515–560.  
<http://people.orie.cornell.edu/~miketodd/soa5.pdf>
- [9] L. Vandenberghe, S. Boyd, *Semidefinite programming*, SIAM Review 38 (1996), 49-95.  
<http://stanford.edu/~boyd/papers/sdp.html>

- [10] Margaret H. Wright, *The interior-point revolution in optimization: History, recent developments, and lasting consequence*, Bull. Amer. Math. Soc. 42 (2005), 39–56.  
[http://www.ams.org/journals/bull/2005-42-01/  
S0273-0979-04-01040-7/S0273-0979-04-01040-7.pdf](http://www.ams.org/journals/bull/2005-42-01/S0273-0979-04-01040-7/S0273-0979-04-01040-7.pdf)
- [11] Y. Ye, *Interior Point Algorithms, Theory and Analysis*, John Wiley & Sons, 1997.

---



---

# CHAPTER 5

---

## 0/1 OPTIMIZATION

Linear optimization problems in which the variables only can attain the values 0 or 1 are called 0/1 linear optimization problems. A 0/1 linear optimization problem in standard form is of the form

$$\begin{aligned} \max \quad & c^\top x \\ & x \in \{0, 1\}^n, \\ & Ax \leq b, \end{aligned} \tag{5.1}$$

with  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^n$ .

Many problems in combinatorial optimization can be written as 0/1 linear optimization problems. One example is finding the independence number  $\alpha(G)$  of a graph  $G = (V, E)$ :

$$\begin{aligned} \alpha(G) = \max \quad & \sum_{v \in V} x_v \\ & x \in \{0, 1\}^V, \\ & x_u + x_v \leq 1 \quad \text{for all } \{u, v\} \in E. \end{aligned}$$

Another example is finding the domination number of a graph  $G$ . A *dominating set* of the graph  $G$  is a subset  $U \subseteq V$  of its vertices so that every vertex in  $V$  is connected to at least one vertex in  $U$ . The cardinality of a smallest dominating set is the *domination number*  $\gamma(G)$ :

$$\begin{aligned} \gamma(G) = \min \quad & \sum_{v \in V} x_v \\ & x \in \{0, 1\}^V, \\ & \sum_{u: \{u, v\} \in E} x_u \geq 1 \quad \text{for all } v \in V. \end{aligned}$$

These two problems, like many other problems in combinatorial optimization, are difficult to solve computationally. They are NP-hard. So we do not expect that there is an efficient algorithm solving them which runs in polynomial time.

One possibility to deal with NP-hard optimization problems is to *relax* them. The set of feasible solutions

$$\mathcal{F} = \{x \in \mathbb{R}^n : x \in \{0, 1\}^n, Ax \leq b\}$$

of a 0/1 linear optimization problem is a subset of the vertices of the cube  $[0, 1]^n$ . We denote the convex hull of all feasible solutions, which is a polytope, by  $P = \text{conv } \mathcal{F}$ . So solving (5.1) is equivalent to solving  $\max\{c^\top x : x \in P\}$  because the maximum of the linear function  $c^\top x$  is attained at an extreme point of the polytope  $P$ . By relaxing (5.1) we mean that we replace  $P$  by a larger set  $P'$ , not necessarily a polytope, which contains  $P$  and for which we can solve  $\max\{c^\top x : x \in P'\}$  efficiently. This maximum value provides an upper bound for the original maximization problem (5.1).

In Section 5.1 we explain a simple method how to construct such a set  $P'$  using semidefinite optimization. Here the theta number and the semidefinite relaxation of max cut of a graph will be important examples.

In Section 5.2 we will go much further. We will consider 0/1 polynomial optimization problems which are much more general than 0/1 linear optimization problems. We explain a hierarchy of stronger and stronger relaxations which even converges to the original problem. This method is one of the strongest general purpose techniques to attack difficult combinatorial optimization problems.

## 5.1 Relaxations using quadratic optimization

Another standard class of optimization problems are *quadratic optimization problems*. They are of the form

$$\begin{aligned} \max \quad & x^\top Q_0 x + b_0^\top x + \alpha_0 \\ & x \in \mathbb{R}^n, \\ & x^\top Q_j x + b_j^\top x + \alpha_j = 0 \quad \text{for all } j \in [m], \end{aligned} \tag{5.2}$$

where  $Q_j \in S^n$  are symmetric matrices,  $b_j \in \mathbb{R}^n$  vectors, and  $\alpha_j \in \mathbb{R}$  scalars.

It is easy to see that one can transform 0/1 linear optimization problems into quadratic optimization problems. The constraint

$$x_i^2 - x_i = 0$$

forces feasible solutions to be 0/1-valued. For inequality constraints  $a_j^\top x \leq b_j$  we introduce a slack variable  $s_j$  and the quadratic equality constraint

$$s_j^2 + a_j^\top x - b_j = 0.$$



**Example 5.1.1.** *The independence number can be formulated as*

$$\begin{aligned}\alpha(G) = \max \quad & \sum_{v \in V} x_v \\ & x \in \mathbb{R}^V, s \in \mathbb{R}^E, \\ & x_v^2 - x_v = 0 \quad \text{for all } v \in V, \\ & s_e^2 + x_u + x_v - 1 = 0 \quad \text{for all } e = \{u, v\} \in E.\end{aligned}$$

*Sometimes problems in combinatorial optimization also come naturally as quadratic optimization problems. One example which we already saw is the max-cut problem*

$$\begin{aligned}\text{MAXCUT}(G) = \max \quad & \sum_{\{u,v\} \in E} (1 - x_u x_v)/2 \\ & x \in \mathbb{R}^V, \\ & x_v^2 = 1 \quad \text{for all } v \in V.\end{aligned}$$

*The  $-1/+1$ -constraint  $x_v = \pm 1$  is equivalent to the quadratic constraint  $x_v^2 = 1$ . Also the independence number has a natural quadratic formulation*

$$\begin{aligned}\alpha(G) = \max \quad & \sum_{v \in V} x_v^2 \\ & x \in \mathbb{R}^V, \\ & x_v^2 - x_v = 0 \quad \text{for all } v \in V, \\ & x_u x_v = 0 \quad \text{for all } \{u, v\} \in E.\end{aligned}$$

Now we would like to relax the quadratic optimization problem by a semidefinite program. For this we rewrite quadratic expressions as trace inner products of symmetric matrices with  $n + 1$  rows and columns

$$x^\top Q_j x + b_j^\top x + \alpha_j = \left\langle \begin{pmatrix} \alpha_j & \frac{1}{2} b_j^\top \\ \frac{1}{2} b_j & Q_j \end{pmatrix}, \begin{pmatrix} 1 & x^\top \\ x & x x^\top \end{pmatrix} \right\rangle.$$

Note that the optimization variable has the following special structure: The matrix

$$Y = \begin{pmatrix} 1 & x^\top \\ x & x x^\top \end{pmatrix}$$

is a semidefinite matrix of rank 1. In 0/1 linear programming, the constraint  $x_i^2 - x_i = 0$  translates into  $Y_{i0} = Y_{ii}$ . When we are dealing with  $-1/+1$ -valued variables, the constraint  $x_i^2 = 1$  translates into  $Y_{ii} = 1$ .

One can get a semidefinite relaxation of the quadratic optimization problem

(5.2) if one simply ignores that the optimization variable  $Y$  has rank 1

$$\begin{aligned} \max \quad & \left\langle \begin{pmatrix} \alpha_0 & \frac{1}{2}b_0^T \\ \frac{1}{2}b_0 & Q_0 \end{pmatrix}, Y \right\rangle \\ & Y \in \mathcal{S}_{\geq 0}^{n+1}, \\ & Y_{00} = 1, \\ & \left\langle \begin{pmatrix} \alpha_j & \frac{1}{2}b_j^T \\ \frac{1}{2}b_j & Q_j \end{pmatrix}, Y \right\rangle = 0 \quad \text{for all } j \in [m]. \end{aligned} \quad (5.3)$$

It is clear, as we now longer impose rank-1 constraint of the solution, that the optimal values of this maximization problem (5.3) is an upper bound of the original quadratic optimization problem (5.2). The set of feasible solutions of (5.2)

$$\{x \in \mathbb{R}^n : x^T Q_j x + b_j^T x + \alpha_j = 0, j \in [m]\}$$

is contained in

$$\left\{ (Y_{11}, \dots, Y_{nn})^T \in \mathbb{R}^n : Y \in \mathcal{S}_{\geq 0}^{n+1}, Y_{00} = 1, \left\langle \begin{pmatrix} \alpha_j & \frac{1}{2}b_j^T \\ \frac{1}{2}b_j & Q_j \end{pmatrix}, Y \right\rangle = 0, j \in [m] \right\}.$$

Geometrically this means that we first lift the  $n$ -dimensional situation of the quadratic optimization problem into a space of dimension  $\binom{n+2}{2} - 1$  in which the relaxed semidefinite optimization problem lives. The matrix  $Y$  has  $\binom{n+2}{2} - 1$  variable entries. We associate with variable  $x_i$  the variable  $Y_{i0}$  and with the product  $x_i x_j$  the variable  $Y_{ij}$ . In the higher dimensional space the quadratic equalities translate into linear inequalities. Then we project the (convex) feasible region of semidefinite program back to  $n$  dimensions.

Dualizing (5.3) yields

$$\begin{aligned} \min \quad & y_0 \\ & y_0, \dots, y_m \in \mathbb{R}, \\ & y_0 \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \sum_{j=1}^m y_j \begin{pmatrix} \alpha_j & \frac{1}{2}b_j^T \\ \frac{1}{2}b_j & Q_j \end{pmatrix} - \begin{pmatrix} \alpha_0 & \frac{1}{2}b_0^T \\ \frac{1}{2}b_0 & Q_0 \end{pmatrix} \geq 0. \end{aligned}$$

By weak duality every feasible solution of the dual provides an upper bound of the original quadratic optimization problem. Thus this provides a rather simple way to *prove* upper bounds of the original problem. So we can certify the quality of solutions by this.

**Example 5.1.2.** *Let us apply this relaxation to the quadratic formulations of  $\alpha$  and MAXCUT. For the independence number we get*

$$\begin{aligned} \alpha(G) \leq \max \quad & \sum_{v \in V} Y_{vv} \\ & Y \in \mathcal{S}_{\geq 0}^{V \cup \{0\}}, \\ & Y_{00} = 1, \\ & Y_{0v} = Y_{vv} \quad \text{for all } v \in V, \\ & Y_{uv} = 0 \quad \text{for all } \{u, v\} \in E. \end{aligned}$$

The matrix variable  $Y$  has one row/column more than the number of vertices of the graph. This extra row/column is indexed by the index 0. In Exercise 5.1 you will show that this semidefinite optimization problem is equivalent to the theta number  $\vartheta(G)$ . For MAXCUT one gets that  $\text{MAXCUT}(G)$  is upper bounded by the semidefinite optimization problem

$$\begin{aligned} \max \quad & \left\langle \begin{pmatrix} |E| & 0^\top \\ 0 & -\frac{1}{2} \sum_{\{u,v\} \in E} E_{uv} \end{pmatrix}, Y \right\rangle \\ & Y \in \mathcal{S}_{\geq 0}^{V \cup \{0\}}, \\ & Y_{00} = 1, \\ & Y_{vv} = 1 \text{ for all } v \in V. \end{aligned}$$

(recall  $E_{uv} = \frac{1}{2}(e_u e_v^\top + e_v e_u^\top)$ ). Here it is obvious that this optimization problem is equivalent to the semidefinite relaxation sdp from Chapter 2.

## 5.2 A hierarchy of semidefinite programs

Now we consider 0/1 polynomial optimization problems where the constraints are allowed to be polynomial inequalities rather than linear inequalities

$$\begin{aligned} \max \quad & c^\top x \\ & x \in \{0, 1\}^n, \\ & p_j(x) \leq b_j \text{ for all } j \in [m]. \end{aligned}$$

with polynomials  $p_1, \dots, p_m \in \mathbb{R}[x_1, \dots, x_n]$ . An optimal solution of this 0/1 polynomial optimization problem is attained at a vertex of the polytope

$$P = \text{conv}(\{x \in \mathbb{R}^n : p_1(x) \leq b_1, \dots, p_m(x) \leq b_m\} \cap \{0, 1\}^n)$$

The following standard form of a 0/1 polynomial optimization problems will become handy

$$\begin{aligned} \max \quad & c^\top x \\ & x \in \{0, 1\}^n, \\ & p_j(x) \geq 0, \text{ for all } j \in [m]. \end{aligned} \tag{5.4}$$

The goal of the lecture is to construct a hierarchy of relaxations of  $P$  consisting out of convex bodies which are projections of feasible regions of semidefinite programs (known as *spectrahedra*) and which find the exact polytope  $P$  after at most  $n$  steps:

$$K_1 \supseteq K_2 \supseteq K_3 \supseteq \dots \supseteq K_n = P.$$

One attractive feature of this construction will be that one can optimize a linear function  $c^\top x$  over each step of the hierarchy  $K_t$  in polynomial time once  $t$  is a fixed, which is independent of the input size.

We start by giving the construction of the last step in the hierarchy  $K_n$ . For this we need some facts from combinatorics.

### 5.2.1 Harmonic analysis on power sets

Let  $X$  be a finite set. For many applications this will be simply  $X = [n] = \{1, \dots, n\}$ . By  $2^X$  we denote the set of all subsets of  $X$ , the *power set* of  $X$ . The space

$$L^2(2^X) = \{f : 2^X \rightarrow \mathbb{R}\}$$

of real-valued functions is a Euclidean space with inner product

$$(f, g) = \sum_{A \in 2^X} f(A)g(A), \text{ with } f, g \in L^2(2^X).$$

If  $X = [n]$  then the space  $L^2(2^X)$  is simply the vector space  $\mathbb{R}^{2^n}$  and  $(f, g)$  is the standard inner product between vectors, and  $f(\{i_1, \dots, i_j\})$  is the component of the vector  $f$  with index  $\{i_1, \dots, i_j\}$ .

We like to write  $L^2(2^X)$  nevertheless because then resemblance with concepts in harmonic (Fourier) analysis, like Fourier coefficients, convolutions, functions of positive type, will become more pronounced.

#### Biorthogonal bases

Two explicit bases of  $L^2(2^X)$  will play an important role for us.

The first one is defined as follows: For  $B \in 2^X$  we define  $\chi_B \in L^2(2^X)$  by

$$\chi_B(A) = \begin{cases} 1 & \text{if } A \subseteq B, \\ 0 & \text{otherwise.} \end{cases}$$

The fact that the function  $\chi_B$ , with  $B \in 2^X$ , forms a basis of  $L^2(2^X)$  can be easily seen: If one considers the function  $\chi_B$  as column vectors in  $\mathbb{R}^{2^X}$  where we order the elements of  $2^X$  by increasing cardinality, then the matrix  $(\chi_B)_{B \in 2^X}$  is an upper triangular matrix in which all diagonal elements are not zero. Note that the value  $\chi_B(A)$  coincides with  $\xi(A, B)$  where  $\xi$  is the zeta-function of the Boolean lattice  $2^X$ , see for instance Aigner [1, Chapter IV.1]. From the definition it is immediate, but extremely important, that this basis is *multiplicative*:

$$\chi_B(A \cup A') = \chi_B(A)\chi_B(A').$$

The second basis is the dual basis  $\chi_B^*$  of  $\chi_B$  which is defined by the biorthogonality relation

$$(\chi_B, \chi_{B'}^*) = \begin{cases} 1 & \text{if } B = B', \\ 0 & \text{otherwise.} \end{cases}$$

Although we will not need it here, one can write down the second basis  $\chi_B^*$  explicitly using the Möbius function  $\mu$  of the Boolean lattice  $2^X$ , namely

$$\chi_B^*(A) = \mu(A, B) = \begin{cases} (-1)^{|B|-|A|} & \text{if } A \subseteq B, \\ 0 & \text{otherwise.} \end{cases}$$

**Example 5.2.1.** For the set  $X = \{1, 2, 3\}$  the matrix with column vectors  $\chi_B$  is

$$\begin{array}{c} \emptyset \\ 1 \\ 2 \\ 3 \\ 12 \\ 13 \\ 23 \\ 123 \end{array} \begin{pmatrix} \emptyset & 1 & 2 & 3 & 12 & 13 & 23 & 123 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

and its inverse contains as row vectors  $\chi_B^*$

$$\begin{array}{c} \emptyset \\ 1 \\ 2 \\ 3 \\ 12 \\ 13 \\ 23 \\ 123 \end{array} \begin{pmatrix} \emptyset & 1 & 2 & 3 & 12 & 13 & 23 & 123 \\ 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 0 & 1 & 0 & 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Every function  $f \in L^2(2^X)$  comes in this way with two basis expansions:

$$f(A) = \sum_{B \in 2^X} \hat{f}(B) \chi_B(A) = \sum_{B \in 2^X} \check{f}(B) \chi_B^*(A),$$

where

$$\hat{f}(B) = (f, \chi_B^*), \text{ and } \check{f}(B) = (f, \chi_B).$$

Here again we use the notation  $\hat{f}$  to pronounce the similarity to harmonic analysis. One can think as  $\hat{f}$  as the “Fourier coefficient” of  $f$ . Classically Fourier coefficients of a function  $f \in L^2([0, 2\pi])$  is given by the expansion

$$f(x) = \sum_{n=-\infty}^{\infty} \hat{f}(n) e^{2\pi i n x}$$

In our case the basis functions  $\chi_B$  play the role of the exponential basis functions and the multiplicativity

$$\chi_B(A \cup A') = \chi_B(A) \chi_B(A')$$

corresponds to

$$e^{2\pi i(n+m)x} = e^{2\pi i n x} e^{2\pi i m x}.$$

Formally we are doing harmonic analysis on the semigroup  $(2^X, \cup)$ .

We have *Plancherel's identity*

$$(f, g) = \sum_{B \in 2^X} \hat{f}(B) \check{g}(B) = \sum_{B \in 2^X} \check{f}(B) \hat{g}(B).$$

For a subset  $A \in 2^X$  we define the *shifted function*  $f_A \in L^2(2^X)$  by

$$f_A(A') = f(A \cup A').$$

Then,

$$\hat{f}_A(B) = \hat{f}(B) \chi_B(A)$$

because

$$f_A(A') = f(A \cup A') = \sum_{B \in 2^X} \hat{f}(B) \chi_B(A \cup A') = \sum_{B \in 2^X} \hat{f}(B) \chi_B(A) \chi_B(A').$$

The *convolution* between two functions  $f, g \in L^2(2^X)$  is

$$(f * g)(A) = (f, g_A) = \sum_{B \in 2^X} \check{f}(B) \hat{g}(B) \chi_B(A),$$

and so

$$\widehat{f * g}(B) = \check{f}(B) \hat{g}(B).$$

Note that the convolution is not commutative.

### Functions of positive type

**Definition 5.2.2.** We say that a function  $f \in L^2(2^X)$  is of *positive type* if the symmetric matrix  $M_f \in \mathcal{S}^{2^X}$  defined by

$$M_f(A, B) = f(A \cup B)$$

is *positive semidefinite*.

Because of the special structure of the matrix  $M_f$ , the entry  $M_f(A, B)$  only depends on the union  $A \cup B$ , it is sometimes called the *moment matrix* of  $f$ . The following theorem gives a characterization of functions of positive type.

A side remark: The theorem is in the spirit of Bochner's theorem in harmonic analysis which says that functions of positive type are the Fourier transform of nonnegative measures.

**Theorem 5.2.3.** A function  $f \in L^2(2^X)$  is of *positive type* if and only if it is of the form

$$f(A) = \sum_{B \in 2^X} \hat{f}(B) \chi_B(A), \quad \hat{f}(B) \geq 0.$$

In other words, the cone of all positive semidefinite moment matrices is a polyhedral cone, even simplicial cone, with extreme rays

$$M_B(A, A') = \chi_B(A \cup A'), \quad B \in 2^X.$$

*Proof.* One direction follows immediately from the multiplicativity of  $\chi_B$ . For the other direction suppose that  $f \in L^2(2^X)$  has the expansion

$$f(A) = \sum_{B \in 2^X} \hat{f}(B) \chi_B(A),$$

where  $\hat{f}(B') < 0$  for some  $B' \in 2^X$ . Then,

$$\begin{aligned} \sum_{A, A' \in 2^X} M_f(A \cup A') \chi_{B'}^*(A) \chi_{B'}^*(A') &= \sum_{A, A', B \in 2^X} \hat{f}(B) \chi_B(A \cup A') \chi_{B'}^*(A) \chi_{B'}^*(A') \\ &= \sum_{B \in 2^X} \hat{f}(B) \left( \sum_{A \in 2^X} \chi_B(A) \chi_{B'}^*(A) \right)^2 \\ &= \hat{f}(B') \\ &< 0, \end{aligned}$$

hence  $M_f$  is not positive semidefinite and  $f$  is not of positive type.  $\square$

### 5.2.2 Lasserre's hierarchy

Now we are ready to construct the hierarchy and prove that it converges.

#### Equivalent reformulation

Since we are dealing with 0/1 problems we can assume that the polynomials  $p_1, \dots, p_m$  are square free, i.e. if one considers the polynomial  $p_i$  as the univariate polynomial in the variable  $x_j$  then its degree is at most one.

We identify square free monomials

$$m = x_{i_1} x_{i_2} \cdots x_{i_j} \in \mathbb{R}[x_1, \dots, x_n]$$

with elements in  $L^2(2^X)$  by

$$m = e_{\{i_1, i_2, \dots, i_j\}} = \sum_{A: A \supseteq \{i_1, i_2, \dots, i_j\}} \chi_A^* \in L^2(2^X),$$

where  $e_{\{i_1, i_2, \dots, i_j\}}$  denotes an element of the standard basis in  $L^2(X)$  (so we even work with a third basis of  $L^2(2^X)$  here) and we extend this by linearity to every square free polynomial.

Let  $a \in \{0, 1\}^X$  be a binary vector and let  $A \in 2^X$  be the corresponding subset, i.e. we have  $a_x = 1$  iff  $x \in A$ . Warning: We will switch between 0/1 vectors and subsets without (another) warning.

Then we can evaluate a square free polynomial  $p$  by

$$p(a) = (p, \chi_A) = \check{p}(A),$$

since

$$(m, \chi_A) = \begin{cases} 1 & \text{if } \{i_1, \dots, i_j\} \subseteq A, \\ 0 & \text{otherwise.} \end{cases}$$

**Example 5.2.4.** Given are the set  $X = \{1, 2, 3\}$  and the polynomial  $p(x_1, x_2, x_3) = 2x_1 + 3x_1x_2$ . Then

$$p(1, 1, 0) = (p, \chi_{12}) = (2(\chi_1^* + \chi_{12}^* + \chi_{13}^* + \chi_{123}^*) + 3(\chi_{12}^* + \chi_{123}^*), \chi_{12}) = 5.$$

**Theorem 5.2.5.** Let

$$\mathcal{F} = \{x \in \{0, 1\}^n : p_1(x) \geq 0, \dots, p_m(x) \geq 0\}$$

be the set of feasible solutions of the 0/1 polynomial optimization problem. Then a function  $f \in L^2(2^{[n]})$  is of the form

$$f(A) = \sum_{b \in \mathcal{F}} \hat{f}(B) \chi_B(A), \quad \text{with } \hat{f}(B) \geq 0$$

if and only if

- a)  $f$  is of positive type,
- b)  $p_j * f$  is of positive type for all  $j = 1, \dots, m$ .

The geometric content of this theorem is actually pretty easy: Every feasible solution  $b \in \mathcal{F}$  which is a 0/1-vectors corresponds to a subset  $B \subseteq \{1, \dots, n\}$ . Now we consider the cone  $2^{[n]}$  which is spanned by all feasible solutions, meaning it is spanned by basis vectors  $\chi_B$  where  $B$  corresponds to  $b \in \mathcal{F}$ . This is a polyhedral cone with  $|\mathcal{F}|$  extreme rays  $\mathbb{R}_{\geq 0} \chi_B$ . Now the Theorem says that  $f \in L^2(2^{[n]})$  is an element of this polyhedral cone iff conditions a) and b) are fulfilled. The advantage of these two conditions is that they have an equivalent formulation in terms of semidefinite matrices, namely that the moment matrices  $M_f$  and  $M_{p_j * f}$  are positive semidefinite.

*Proof.* We have

$$(p_j * f)(A) = \sum_{B \in 2^X} \check{p}_j(B) \hat{f}(B) \chi_B(A).$$

Suppose conditions a) and b) are satisfied. By Theorem 5.2.3 the function  $f$  can be represented as

$$f(A) = \sum_{B \in 2^X} \hat{f}(B) \chi_B(A)$$

with  $\hat{f}(B) \geq 0$ . If strict inequality  $\hat{f}(B) > 0$  holds for some  $B$  then for every  $j \in \{1, \dots, m\}$  we have again by Theorem 5.2.3

$$\check{p}_j(B) \hat{f}(B) = p_j(b) \hat{f}(B) \geq 0,$$

because  $p_j * f$  is of positive type. Thus,  $p_j(b) \geq 0$  for all  $j$  and hence  $b \in \mathcal{F}$ . The other implication follows with the same arguments.  $\square$



**Corollary 5.2.6.** *The 0/1 polynomial optimization problem (5.4) is equivalent to the linear optimization problem*

$$\begin{aligned} \max \quad & \sum_{i=1}^n c_i f(\{i\}) \\ & f \in L^2(2^X) \text{ is of positive type,} \\ & f(\emptyset) = 1, \\ & p_j * f \text{ is of positive type for all } j \in [m], \end{aligned}$$

which is equivalent to the semidefinite program

$$\begin{aligned} \max \quad & \sum_{i=1}^n c_i M_f(\{i\}, \{i\}) \\ & M_f \in \mathcal{S}_{\geq 0}^{2^X}, \\ & M_f(\emptyset, \emptyset) = 1, \\ & M_{p_j * f} \in \mathcal{S}_{\geq 0}^{2^X} \text{ for all } j \in [m]. \end{aligned}$$

*Proof.* The fact that these two optimization problems are equivalent follows immediately from the definition of positive type functions.

Since we are maximizing a linear function we can assume that the optimum  $f_0$  of the first problem is attained at the extreme ray of the cone of positive type functions. So it is of the form

$$f_0 = \alpha \chi_B$$

for some  $\alpha \geq 0$  and by the previous theorem for some  $b$  with  $p_j(b) \geq 0$  for all  $j \in [m]$ . Furthermore,

$$1 = f_0(\emptyset) = \alpha \chi_B(\emptyset) = \alpha$$

makes sure that one can recover an optimal 0/1 solution from  $f_0$ . The objective value is

$$\sum_{i=1}^n c_i f(\{i\}) = \sum_{i=1}^n c_i \chi_B(\{i\}).$$

□

### Probabilistic interpretation

The two conditions  $f(\emptyset) = 1$  and  $f$  is of positive type have a simple probabilistic interpretation: The second condition says that  $f$  is the Fourier transform of a positive measure whereas the first condition says that the positive measure is in fact a probability measure

$$1 = f(\emptyset) = \sum_{B \in 2^X} \hat{f}(B) \chi_B(\emptyset) = \sum_{B \in 2^X} \hat{f}(B).$$

Hence,  $f$  determines a probability distribution on the power set  $2^X$ . The set  $B$  is picked with probability  $\hat{f}(B)$ .

## Relaxation

Computationally, the equivalent reformulation in Corollary 5.2.6 is a rather useless statement: We exponentially increased the dimension of the problem and so in a sense we are enumerating all feasible solutions. However, the reformulation can be used to generate systematically valid non-linear inequalities, i.e. projected LMI inequalities, for the 0/1 polytope  $P = \text{conv } \mathcal{F}$ .

Instead of working with full moment matrices lying in  $\mathcal{S}_{\geq 0}^{2^X}$  we only consider in the  $t$ -th step/round of Lasserre's hierarchy truncated moment matrices where the rows/columns are indexed by all subsets with cardinality at most  $t + 1$ ;  
Notation:  $\binom{X}{\leq t+1}$ .

For  $f \in \mathbb{R}^{\binom{X}{\leq 2t+2}}$  define the *truncated moment matrix*  $M_f^{t+1} \in \mathcal{S}^{\binom{X}{\leq t+1}}$  by

$$M_f^{t+1}(A, B) = f(A \cup B)$$

with  $A, B \in \binom{X}{\leq t+1}$ . Let  $p$  be a squarefree polynomial of degree  $d$

$$p = \sum_I p_I m_I$$

with coefficients  $p_I$  and monomials  $m_I$  where  $I$  runs through a subset  $J$  of  $2^X$  where every  $I \in J$  has cardinality at most  $d$ . Then,

$$M_{p*f}^{t+1-\lceil d/2 \rceil}(A, B) = (p * f)(A \cup B) = \sum_I p_I f(I \cup A \cup B)$$

with  $A, B \in \binom{X}{\leq t+1-\lceil d/2 \rceil}$ .

**Definition 5.2.7.** Let  $p_1, \dots, p_m \in \mathbb{R}[x_1, \dots, x_n]$  be polynomials with degrees  $d_1, \dots, d_m$ . Furthermore, let

$$v = \max \{ \lceil d_j/2 \rceil : j \in [m] \}.$$

We define for  $t \geq v - 1$  the  $t$ -th step in Lasserre's hierarchy by

$$\begin{aligned} K_t = \{ (f(\{1\}), \dots, f(\{n\}))^T \in \mathbb{R}^n : f \in \mathbb{R}^{\binom{X}{\leq 2t+2}}, \\ f(\emptyset) = 1, \\ M_f^{t+1} \geq 0, \\ M_{p_j*f}^{t+1-\lceil d_j/2 \rceil} \geq 0, j \in [m] \}. \end{aligned}$$

Optimizing over  $K_t$  can be done in polynomial time once  $t \geq v - 1$  is a fixed constant, i.e. independent of the input size. By the previous considerations we have

$$K_{v-1} \supseteq K_v \supseteq K_{v+1} \supseteq \dots \supseteq K_{n+v-1} = P.$$

### 5.2.3 Example: Independence number

Let us apply Lasserre's hierarchy to the problem of finding the independence number.

For this we consider the polynomials

$$p_{uv}(x) = 1 - x_u - x_v \in \mathbb{R}[x_v : v \in V] \quad \text{for all } \{u, v\} \in E.$$

The 0/1 polytope

$$P = \text{ST}(G) = \text{conv}\{x \in \{0, 1\}^V : p_{uv} \geq 0 \text{ for all } \{u, v\} \in E\}$$

is called the *stable set polytope*<sup>1</sup>.

The following theorem says that one can simplify the conditions of  $K_t$  here. In fact the semidefinite conditions on the matrices  $M_{p_{uv}*f}^t$  can be replaced by the conditions  $f(\{u, v\}) = 0$  in the moment matrix  $M_f^{t+1}$ .

**Theorem 5.2.8.** For  $t \geq 1$  and  $f \in \binom{V}{\leq 2t+2}$  the following three statements are equivalent:

- a)  $M_f^{t+1} \geq 0$  and  $M_{p_{uv}*f}^{t+1} \geq 0$  for all  $\{u, v\} \in E$ ,
- b)  $M_f^{t+1} \geq 0$  and  $f(\{u, v\}) = 0$  for every  $\{u, v\} \in E$ ,
- c)  $M_f^{t+1} \geq 0$  and  $f(U) = 0$  for every subset  $U \in \binom{V}{\leq 2t+2}$  of the vertex set which is not an independent set.

*Proof.* From  $M_f^{t+1} \geq 0$  we can conclude that  $f(U) \geq 0$  for all  $U \in \binom{V}{\leq 2t+2}$  because  $f(U)$  is a diagonal element of the positive semidefinite matrix  $M_f^{t+1}$ .

a) implies b): Consider the  $(\{u\}, \{u\})$ -entry of  $M_{p_{uv}*f}^t$ . It equals (check this identity!)

$$p_{uv} * f(\{u\}) = -f(\{u, v\})$$

Since it is a diagonal entry, it has to be non-negative. So  $f(\{u, v\}) = 0$ .

b) implies c): Suppose  $u, v \in U$  are connected by an edge.

First case:  $|U| \leq t + 1$ , then the  $(\{u, v\}, \{u, v\})$ -entry of  $M_f^{t+1}$  equals 0, and so the  $(\{u, v\}, U)$ -entry, too (why?). Hence also  $f(U) = M_f^{t+1}(U, U) = 0$ .

Second case:  $|U| > t + 1$ . Split  $U$  into  $U = U_1 \cup U_2$  with  $U_1, U_2 \in \binom{V}{\leq t+1}$  and assume  $\{u, v\} \subseteq U_1$ . Then,

$$M_f^{t+1}(U_1, U_1) = 0 \implies M_f^{t+1}(U_1, U_2) = 0 \implies f(U) = M_f^{t+1}(U, U) = 0.$$

c) implies a): We shall prove that the matrix  $M_{p_{uv}*f}^t$  is positive semidefinite. Define  $P_0 = \binom{V \setminus \{u, v\}}{\leq t}$  and  $P_w = \{U \cup \{w\} : U \in P_0\}$  for  $w \in \{u, v\}$ . Then, the

<sup>1</sup>Sometimes independent sets are called stable sets, but who prefers stability over independence?

principal submatrix of  $M_f^{t+1}$  indexed by  $P_0 \cup P_u \cup P_v$  is of the form

$$\begin{matrix} & P_0 & P_u & P_v \\ \begin{matrix} P_0 \\ P_u \\ P_v \end{matrix} & \begin{pmatrix} C & A & B \\ A & A & 0 \\ B & 0 & B \end{pmatrix} \end{matrix}.$$

Since this matrix is positive semidefinite by assumption it follows that the matrix  $C - A - B$  is positive semidefinite because

$$\begin{pmatrix} -x \\ x \\ x \end{pmatrix}^T \begin{pmatrix} C & A & B \\ A & A & 0 \\ B & 0 & B \end{pmatrix} \begin{pmatrix} -x \\ x \\ x \end{pmatrix} = x^T (C - A - B)x.$$

Now consider the matrix  $M_{p_{uv}*f}^t$  where we partition the rows/columns into  $P_0$  and its complement  $P'_0 = (V_{\leq t+1}) \setminus P_0$ . It has the form

$$\begin{matrix} & P_0 & P'_0 \\ \begin{matrix} P_0 \\ P'_0 \end{matrix} & \begin{pmatrix} C - A - B & 0 \\ 0 & 0 \end{pmatrix} \end{matrix}$$

and the result follows.  $\square$

One can show (Exercise 5.1.c)) that Lasserre's hierarchy already converges at step  $\alpha(G) - 1$  to  $\text{ST}(G)$ :

**Theorem 5.2.9.** *For a graph  $G$  with  $\alpha(G) \geq 2$  we have*

$$\text{ST}(G) = K_{\alpha(G)-1}.$$

### 5.3 Further reading

Shor [4] was the first who realized that one can use semidefinite programming to relax quadratic optimization problems. Meanwhile there are many different possibilities available to construct semidefinite programming hierarchies for 0/1 polynomial optimization problems. The one we studied in Section 5.2 is due to Lasserre [2]. Lasserre's hierarchy and other hierarchies are presented and compared by Laurent in [3].

## 5.4 Exercises

- 5.1.\*\* a) Show that the following two semidefinite programs give the same optimal value:

$$\begin{aligned} \max \quad & \sum_{v \in V} Y_{vv} \\ & Y \in \mathcal{S}_{\geq 0}^{V \cup \{0\}}, \\ & Y_{00} = 1, \\ & Y_{0v} = Y_{vv} \quad \text{for all } v \in V, \\ & Y_{uv} = 0 \quad \text{for all } \{u, v\} \in E, \end{aligned}$$

and

$$\begin{aligned} \vartheta(G) = \max \quad & \langle J, X \rangle \\ & X \in \mathcal{S}_{\geq 0}^V, \\ & \langle I, X \rangle = 1, \\ & X_{uv} = 0 \quad \text{for all } \{u, v\} \in E. \end{aligned}$$

- b) Prove that the theta number of the cycle graph with five vertices  $C_5$  is exactly  $\sqrt{5}$ .  
c) Show that  $K_{\alpha(G)-1} = \text{ST}(G)$ .  
d) True or false? For all graphs  $G = (V, E)$  the following equality holds

$$\vartheta(G) = \max \left\{ \sum_{v \in V} x_v : x \in K_1 \right\}.$$

5.2. (Computer exercise)

- a) Do the following two ellipsoids intersect?

$$\begin{aligned} E_1 &= \left\{ x \in \mathbb{R}^3 : x^\top \begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 13/36 & 13/36 \\ 1/9 & 13/36 & 49/36 \end{pmatrix} x \leq 1 \right\}, \\ E_2 &= \left\{ x \in \mathbb{R}^3 : x^\top \begin{pmatrix} \sqrt{2} & \sqrt{2} & \sqrt{2} \\ \sqrt{2} & 2\sqrt{2} & 2\sqrt{2} \\ \sqrt{2} & 2\sqrt{2} & 2\sqrt{2} + 1 \end{pmatrix} x + \begin{pmatrix} -6\sqrt{2} \\ -10\sqrt{2} \\ -10\sqrt{2} \end{pmatrix}^\top x \leq 1 - 13\sqrt{2} \right\} \end{aligned}$$

Solve a semidefinite program to justify your answer.

- b) Let  $x_1, \dots, x_N$  be  $N$  points in  $\mathbb{R}^2$ . Find a point  $x \in \mathbb{R}^2$  which minimizes the maximum Euclidean distance to these points. Compute the point  $x$  for the cities Amsterdam, Athens, Berlin, Copenhagen, Lisbon, Moscow, Prague, Warsaw. For this assume that Europe is part of flatland...

---

## BIBLIOGRAPHY

- [1] M. Aigner, *Combinatorial theory*, Springer, 1979.
- [2] J.B. Lasserre, *An explicit exact SDP relaxation for nonlinear 0-1 programs*, pp. 293–303 in *Integer Programming and Combinatorial Optimization, 8th International IPCO Conference* (K. Aardal, A.M.H. Gerards, ed.), Springer, 2001.
- [3] M. Laurent, *A comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre relaxations for 0-1 programming*, Math. Oper. Res. **28** (2003), 470–496.  
<http://www.cwi.nl/~monique/files/lasserrefinal.ps>
- [4] N.Z. Shor, *Quadratic optimization problems*, Soviet J. Comput. Systems Sci. **25** (1987), 1–11.

---



---

# CHAPTER 6

---

## GRAPH COLORING AND INDEPENDENT SETS

In this chapter we revisit in detail the theta number  $\vartheta(G)$ , which has already been introduced in earlier chapters. In particular, we present several equivalent formulations for  $\vartheta(G)$ , we discuss its geometric properties, and present some applications: for bounding the Shannon capacity of a graph, and for computing in polynomial time maximum stable sets and minimum colorings in perfect graphs.

Here are some additional definitions used in this chapter. Let  $G = (V, E)$  be a graph. Then,  $\overline{E}$  denotes the set of pairs  $\{i, j\}$  of distinct nodes that are not adjacent in  $G$ . The graph  $\overline{G} = (V, \overline{E})$  is called the *complementary graph* of  $G$ .  $G$  is *self-complementary* if  $G$  and  $\overline{G}$  are isomorphic graphs. Given a subset  $S \subseteq V$ ,  $G[S]$  denotes the *subgraph induced by  $S$* : its node set is  $S$  and its edges are all pairs  $\{i, j\} \in E$  with  $i, j \in S$ . The graph  $C_n$  is the circuit (or cycle) of length  $n$ , with node set  $[n]$  and edges the pairs  $\{i, i+1\}$  (for  $i \in [n]$ , indices taken modulo  $n$ ). For a set  $S \subseteq V$ , its *characteristic vector* is the vector  $\chi^S \in \{0, 1\}^V$ , whose  $i$ -th entry is 1 if  $i \in S$  and 0 otherwise. As before,  $e$  denotes the all-ones vector.

### 6.1 Preliminaries on graphs

#### 6.1.1 Stability and chromatic numbers

A subset  $S \subseteq V$  of nodes is said to be *stable* (or *independent*) if no two nodes of  $S$  are adjacent in  $G$ . Then the *stability number* of  $G$  is the parameter  $\alpha(G)$  defined as the maximum cardinality of an independent set in  $G$ .

A subset  $C \subseteq V$  of nodes is called a *clique* if every two distinct nodes in  $C$  are adjacent. The maximum cardinality of a clique in  $G$  is denoted  $\omega(G)$ , the *clique number* of  $G$ . Clearly,

$$\omega(G) = \alpha(\overline{G}).$$

Computing the stability number of a graph is a hard problem: Given a graph  $G$  and an integer  $k$ , deciding whether  $\alpha(G) \geq k$  is an  $\mathcal{NP}$ -complete problem.

Given an integer  $k \geq 1$ , a  $k$ -coloring of  $G$  is an assignment of numbers (view them as *colors*) from  $\{1, \dots, k\}$  to the nodes in such a way that two adjacent nodes receive distinct colors. In other words, this corresponds to a partition of  $V$  into  $k$  stable sets:  $V = S_1 \cup \dots \cup S_k$ , where  $S_i$  is the stable set consisting of all nodes that received the  $i$ -th color. The *coloring* (or *chromatic number*) is the smallest integer  $k$  for which  $G$  admits a  $k$ -coloring, it is denoted as  $\chi(G)$ .

Again it is an  $\mathcal{NP}$ -complete problem to decide whether a graph is  $k$ -colorable. In fact, it is  $\mathcal{NP}$ -complete to decide whether a planar graph is 3-colorable. On the other hand, it is known that every planar graph is 4-colorable – this is the celebrated 4-color theorem. Moreover, observe that one can decide in polynomial time whether a graph is 2-colorable, since one can check in polynomial time whether a graph is bipartite.

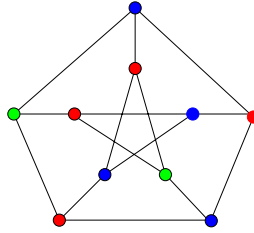


Figure 6.1: The Petersen graph has  $\alpha(G) = 4$ ,  $\omega(G) = 2$  and  $\chi(G) = 3$

Clearly, any two nodes in a clique of  $G$  must receive distinct colors. Therefore, for any graph, the following inequality holds:

$$\omega(G) \leq \chi(G). \quad (6.1)$$

This inequality is strict, for example, when  $G$  is an odd circuit, i.e., a circuit of odd length at least 5, or its complement. Indeed, for an odd circuit  $C_{2n+1}$  ( $n \geq 2$ ),  $\omega(C_{2n+1}) = 2$  while  $\chi(C_{2n+1}) = 3$ . Moreover, for the complement  $G = \overline{C_{2n+1}}$ ,  $\omega(G) = n$  while  $\chi(G) = n + 1$ . For an illustration see the cycle of length 7 and its complement in Figure 6.2.

### 6.1.2 Perfect graphs

It is intriguing to understand for which graphs equality  $\omega(G) = \chi(G)$  holds. Note that any graph  $G$  with  $\omega(G) < \chi(G)$  can be embedded in a larger graph  $\hat{G}$



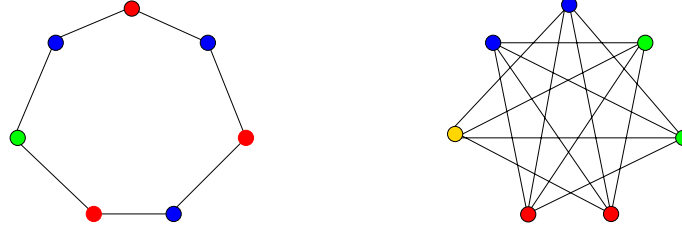


Figure 6.2: For  $C_7$  and its complement  $\overline{C_7}$ :  $\omega(C_7) = 2$ ,  $\chi(C_7) = 3$ ,  $\omega(\overline{C_7}) = \alpha(C_7) = 3$ ,  $\chi(\overline{C_7}) = 4$

with  $\omega(\hat{G}) = \chi(\hat{G})$ , simply by adding to  $G$  a clique of size  $\chi(G)$ . This justifies the following definition, introduced by C. Berge in the early sixties, which makes the problem well posed.

**Definition 6.1.1.** A graph  $G$  is said to be perfect if equality

$$\omega(H) = \chi(H)$$

holds for all induced subgraphs  $H$  of  $G$  (including  $H = G$ ).

For instance, bipartite graphs are perfect. It follows from the definition and the above observation about odd circuits that if  $G$  is a perfect graph then it does not contain an odd circuit of length at least 5 or its complement as an induced subgraph. Berge already conjectured that *all* perfect graphs arise in this way. Resolving this conjecture has haunted generations of graph theorists. It was finally settled in 2004 by Chudnovsky, Robertson, Seymour and Thomas who proved the following result, known as the *strong perfect graph theorem*:

**Theorem 6.1.2. (The strong perfect graph theorem)** A graph  $G$  is perfect if and only if it does not contain an odd circuit of length at least 5 or its complement as an induced subgraph.

This implies the following structural result about perfect graphs, known as the *perfect graph theorem*, already proved by Lovász in 1972.

**Theorem 6.1.3. (The perfect graph theorem)** If  $G$  is a perfect graph, then its complement  $\overline{G}$  too is a perfect graph.

We will mention below some other, more geometric, characterizations of perfect graphs.

## 6.2 Linear programming bounds

Let  $\text{ST}(G)$  denote the polytope in  $\mathbb{R}^V$  defined as the convex hull of the characteristic vectors of the stable sets of  $G$ :

$$\text{ST}(G) = \text{conv}\{\chi^S : S \subseteq V, S \text{ is a stable set in } G\},$$

called the *stable set polytope* of  $G$ . Hence, computing  $\alpha(G)$  is linear optimization over the stable set polytope:

$$\alpha(G) = \max\{e^\top x : x \in \text{ST}(G)\}.$$

We have now defined the stable set polytope by listing explicitly its extreme points. Alternatively, it can also be represented by its hyperplanes representation, i.e., in the form

$$\text{ST}(G) = \{x \in \mathbb{R}^V : Ax \leq b\}$$

for some matrix  $A$  and some vector  $b$ . As computing the stability number is a hard problem one cannot hope to find the full linear inequality description of the stable set polytope (i.e., the explicit  $A$  and  $b$ ). However some partial information is known: several classes of valid inequalities for the stable set polytope are known. For instance, if  $C$  is a clique of  $G$ , then the *clique inequality*

$$x(C) = \sum_{i \in C} x_i \leq 1 \quad (6.2)$$

is valid for  $\text{ST}(G)$ : any stable set can contain at most one vertex from the clique  $C$ . The clique inequalities define the polytope

$$\text{QST}(G) = \{x \in \mathbb{R}^V : x \geq 0, x(C) \leq 1 \ \forall C \text{ clique of } G\} \quad (6.3)$$

and maximizing the linear function  $e^\top x$  over it gives the parameter

$$\alpha^*(G) = \max\{e^\top x : x \in \text{QST}(G)\}, \quad (6.4)$$

known as the *fractional stability number* of  $G$ . Clearly,  $\text{QST}(G)$  is a relaxation of the stable set polytope:

$$\text{ST}(G) \subseteq \text{QST}(G). \quad (6.5)$$

The parameter  $\alpha^*(G)$  is nested between  $\alpha(G)$  and  $\chi(\overline{G})$ , and it can also be interpreted in terms of *fractional colorings* of  $\overline{G}$ .

**Lemma 6.2.1.** *For any graph  $G$ , we have*

$$\alpha(G) \leq \alpha^*(G) \leq \chi(\overline{G}), \quad (6.6)$$

and  $\alpha^*(G)$  is equal to the optimal value of the linear program

$$\min \left\{ \sum_{C \text{ clique of } G} y_C : \sum_{C \text{ clique of } G} y_C \chi^C = e, y_C \geq 0 \ \forall C \text{ clique of } G \right\}. \quad (6.7)$$

*Proof.* The left most inequality in (6.6) follows from the inclusion (6.5) and the right most one from the definitions: If  $x \in \text{QST}(G)$  and  $V = C_1 \cup \dots \cup C_k$  is a partition into  $k$  cliques, then

$$x^\top e = x^\top \left( \sum_{i=1}^k \chi^{C_i} \right) = \sum_{i=1}^k x(C_i) \leq \sum_{i=1}^k 1 = k.$$

We now show that the optimal value of (6.7) is equal to  $\alpha^*(G)$ . For this, first note that in the linear program (6.4) the condition  $x \geq 0$  can be removed without changing the optimal value; that is,

$$\alpha^*(G) = \max\{e^\top x : x(C) \leq 1 \ \forall C \text{ clique of } G\}.$$

Now, applying linear programming duality to this linear program, we obtain the linear program (6.7).  $\square$

For instance, for an odd circuit  $C_{2n+1}$  ( $n \geq 2$ ),  $\alpha^*(C_{2n+1}) = \frac{2n+1}{2}$  (check it) lies strictly between  $\alpha(C_{2n+1}) = n$  and  $\chi(\overline{C_{2n+1}}) = n + 1$ .

Assume that  $G$  is a perfect graph. Then equality holds throughout in relation (6.6). Hence, when  $w \in \mathbb{R}^V$  is the all-ones vector, maximizing the objective function  $w^\top x$  over the stable set polytope  $\text{ST}(G)$  or over its linear relaxation  $\text{QST}(G)$  gives the same optimal values. The same holds if  $w$  is 0/1 valued since this amounts to replacing  $G$  by its subgraph induced by the set of nodes with  $w_v = 1$ , which is again perfect. One can show that the same holds for *any* integer vector  $w \in \mathbb{Z}_{\geq 0}^V$ , which implies that the two polytopes  $\text{ST}(G)$  and  $\text{QST}(G)$  coincide. Moreover, this property characterizes perfect graphs.

**Theorem 6.2.2.** *A graph  $G$  is perfect if and only if  $\text{ST}(G) = \text{QST}(G)$ .*

Although an explicit linear inequality description is known for the stable set polytope of a perfect graph (given by the clique inequalities), it is not clear how to use this information in order to give an efficient algorithm for optimizing over the stable set polytope. As we see later in Section 6.5.2 there is yet another description of  $\text{ST}(G)$  – in terms of semidefinite programming, using the theta body  $\text{TH}(G)$  – that allows to give an efficient algorithm.

## 6.3 Semidefinite programming bounds

### 6.3.1 The theta number

**Definition 6.3.1.** *Given a graph  $G = (V, E)$ , consider the following semidefinite program*

$$\max_{X \in S^n} \{\langle J, X \rangle : \text{Tr}(X) = 1, X_{ij} = 0 \ \forall \{i, j\} \in E, X \geq 0\}. \quad (6.8)$$

*Its optimal value is denoted as  $\vartheta(G)$ , and called the theta number of  $G$ .*

This parameter was introduced by Lovász [3]. He proved the following simple, but crucial result – called the Sandwich Theorem by Knuth [2] – which shows that  $\vartheta(G)$  provides a bound for both the stability number of  $G$  and the chromatic number of the complementary graph  $\overline{G}$ .

**Theorem 6.3.2. (Lovász' sandwich theorem)** *For any graph  $G$ , we have that*

$$\alpha(G) \leq \vartheta(G) \leq \chi(\overline{G}).$$

*Proof.* Given a stable set  $S$  of cardinality  $|S| = \alpha(G)$ , define the matrix

$$X = \frac{1}{|S|} \chi^S (\chi^S)^\top \in \mathcal{S}^n.$$

Then  $X$  is feasible for (6.8) with objective value  $\langle J, X \rangle = |S|$  (check it). This shows the inequality  $\alpha(G) \leq \vartheta(G)$ .

Now, consider a matrix  $X$  feasible for the program (6.8) and a partition of  $V$  into  $k$  cliques:  $V = C_1 \cup \dots \cup C_k$ . Our goal is now to show that  $\langle J, X \rangle \leq k$ , which will imply  $\vartheta(G) \leq \chi(\overline{G})$ . For this, using the relation  $e = \sum_{i=1}^k \chi^{C_i}$ , observe that

$$Y := \sum_{i=1}^k (k\chi^{C_i} - e) (k\chi^{C_i} - e)^\top = k^2 \sum_{i=1}^k \chi^{C_i} (\chi^{C_i})^\top - kJ.$$

Moreover,

$$\left\langle X, \sum_{i=1}^k \chi^{C_i} (\chi^{C_i})^\top \right\rangle = \text{Tr}(X).$$

Indeed the matrix  $\sum_i \chi^{C_i} (\chi^{C_i})^\top$  has all its diagonal entries equal to 1 and it has zero off-diagonal entries outside the edge set of  $G$ , while  $X$  has zero off-diagonal entries on the edge set of  $G$ . As  $X, Y \geq 0$ , we obtain

$$0 \leq \langle X, Y \rangle = k^2 \text{Tr}(X) - k \langle J, X \rangle$$

and thus  $\langle J, X \rangle \leq k \text{Tr}(X) = k$ .  $\square$

An alternative argument for the inequality  $\vartheta(G) \leq \chi(\overline{G})$ , showing an even more transparent link to coverings by cliques, will be given in the paragraph after the proof of Lemma 6.4.2.

### 6.3.2 Computing maximum stable sets in perfect graphs

Assume that  $G$  is a graph satisfying  $\alpha(G) = \chi(\overline{G})$ . Then, as a direct application of Theorem 6.3.2,  $\alpha(G) = \chi(\overline{G}) = \vartheta(G)$  can be computed by solving the semidefinite program (6.8), it suffices to solve this semidefinite program with precision  $\epsilon < 1/2$  as one can then find  $\alpha(G)$  by rounding the optimal value to the nearest integer. In particular, combining with the perfect graph theorem (Theorem 6.1.3):

**Theorem 6.3.3.** *If  $G$  is a perfect graph then  $\alpha(G) = \chi(\overline{G}) = \vartheta(G)$  and  $\omega(G) = \chi(G) = \vartheta(\overline{G})$ .*

Hence one can compute the stability number and the chromatic number in polynomial time for perfect graphs. Moreover, one can also find a maximum stable set and a minimum coloring in polynomial time for perfect graphs. We now indicate how to construct a maximum stable set – we deal with minimum graph colorings in the next section.

Let  $G = (V, E)$  be a perfect graph. Order the nodes of  $G$  as  $v_1, \dots, v_n$ . Then we construct a sequence of induced subgraphs  $G_0, G_1, \dots, G_n$  of  $G$ . Hence each  $G_i$  is perfect, also after removing a node, so that we can compute in polynomial time the stability number of such graphs. The construction goes as follows: Set  $G_0 = G$ . For each  $i = 1, \dots, n$  do the following:

1. Compute  $\alpha(G_{i-1} \setminus v_i)$ .
2. If  $\alpha(G_{i-1} \setminus v_i) = \alpha(G)$ , then set  $G_i = G_{i-1} \setminus v_i$ .
3. Otherwise, set  $G_i = G_{i-1}$ .

By construction,  $\alpha(G_i) = \alpha(G)$  for all  $i$ . In particular,  $\alpha(G_n) = \alpha(G)$ . Moreover, the node set of the final graph  $G_n$  is a stable set and, therefore, it is a maximum stable set of  $G$ . Indeed, if the node set of  $G_n$  is not stable then it contains a node  $v_i$  for which  $\alpha(G_n \setminus v_i) = \alpha(G_n)$ . But then, as  $G_n$  is an induced subgraph of  $G_{i-1}$ , one would have that  $\alpha(G_n \setminus v_i) \leq \alpha(G_{i-1} \setminus v_i)$  and thus  $\alpha(G_{i-1} \setminus v_i) = \alpha(G)$ , so that node  $v_i$  would have been removed at Step 2.

Hence, the above algorithm permits to construct a maximum stable set in a perfect graph  $G$  in polynomial time – namely by solving  $n + 1$  semidefinite programs for computing  $\alpha(G)$  and  $\alpha(G_{i-1} \setminus v_i)$  for  $i = 1, \dots, n$ .

More generally, given integer weights  $w \in \mathbb{Z}_{\geq 0}^V$  on the nodes, one can compute in polynomial time a stable set  $S$  of maximum weight  $w(S)$ . For this, one can apply the algorithm just described for computing a maximum cardinality stable set in the new graph  $G'$  defined in the following way: Replace each node  $i \in V$  by a set  $W_i$  of  $w_i$  nodes pairwise non-adjacent, and make two nodes  $x \in W_i$  and  $y \in W_j$  adjacent if  $i$  and  $j$  are adjacent in  $G$ . One can verify that  $G'$  is perfect and that  $\alpha(G')$  is the maximum weight  $w(S)$  of a stable set  $S$  in  $G$ .

### 6.3.3 Minimum colorings of perfect graphs

We now describe an algorithm for computing a minimum coloring of a perfect graph  $G$  in polynomial time. This will be reduced to several computations of the theta number which we will use for computing the clique number of some induced subgraphs of  $G$ .

Let  $G = (V, E)$  be a perfect graph. Call a clique of  $G$  *maximum* if it has maximum cardinality  $\omega(G)$ .

The crucial observation is that it suffices to find a stable set  $S$  in  $G$  which meets all maximum cliques. Indeed, if such  $S$  is found then one can recursively color  $G \setminus S$  with  $\omega(G) - 1$  colors (in polynomial time), and thus  $G$  with  $\omega(G)$  colors. (Clearly, such a stable set  $S$  exists: any color class  $S$  in a  $\omega(G)$ -coloring must meet all maximum cliques as  $\omega(G \setminus S) = \chi(G \setminus S) = \omega(G) - 1$ .)

The algorithm goes as follows: For  $t \geq 1$ , grow a list  $\mathcal{L}$  of  $t$  maximum cliques  $C_1, \dots, C_t$ . Suppose  $C_1, \dots, C_t$  have been found. Then do the following:

1. We find a stable set  $S$  meeting each of the cliques  $C_1, \dots, C_t$  (see below).

2. Compute  $\omega(G \setminus S)$ .
3. If  $\omega(G \setminus S) < \omega(G)$  then  $S$  meets all maximum cliques and we are done.
4. Otherwise, compute a maximum clique  $C_{t+1}$  in  $G \setminus S$ , which is thus a new maximum clique of  $G$ , and we add it to the list  $\mathcal{L}$ .

The first step can be done as follows: Set  $w = \sum_{i=1}^t \chi^{C_i} \in \mathbb{Z}_{\geq 0}^V$  and compute a stable set  $S$  having maximum possible weight  $w(S)$ , then  $w(S) = t$  and  $S$  meets  $C_1, \dots, C_t$ .

The above algorithm has polynomial running time, since the number of iterations is bounded by  $|V|$ . To see this, define the affine space  $L_t \subseteq \mathbb{R}^V$  defined by the equations  $x(C_1) = 1, \dots, x(C_t) = 1$  corresponding to the cliques in the current list  $\mathcal{L}$ . Then,  $L_t$  contains strictly  $L_{t+1}$ , since  $\chi^S \in L_t \setminus L_{t+1}$  for the set  $S$  constructed in the first step, and thus the dimension decreases at least by 1 at each iteration.

## 6.4 Other formulations of the theta number

### 6.4.1 Dual formulation

We now give several equivalent formulations for the theta number obtained by applying semidefinite programming duality and some further elementary manipulations.

**Lemma 6.4.1.** *The theta number can be expressed by any of the following programs:*

$$\vartheta(G) = \min_{t \in \mathbb{R}, A \in S^n} \{t : tI + A - J \geq 0, A_{ij} = 0 \text{ } (i = j \text{ or } \{i, j\} \in \overline{E})\}, \quad (6.9)$$

$$\vartheta(G) = \min_{t \in \mathbb{R}, B \in S^n} \{t : tI - B \geq 0, B_{ij} = 1 \text{ } (i = j \text{ or } \{i, j\} \in \overline{E})\}, \quad (6.10)$$

$$\vartheta(G) = \min_{t \in \mathbb{R}, C \in S^n} \{t : C - J \geq 0, C_{ii} = t \text{ } (i \in V), C_{ij} = 0 \text{ } (\{i, j\} \in \overline{E})\}, \quad (6.11)$$

$$\vartheta(G) = \min_{B \in S^n} \{\lambda_{\max}(B) : B_{ij} = 1 \text{ } (i = j \text{ or } \{i, j\} \in \overline{E})\}. \quad (6.12)$$

*Proof.* First we build the dual of the semidefinite program (6.8), which reads:

$$\min_{t \in \mathbb{R}, y \in \mathbb{R}^E} \left\{ t : tI + \sum_{\{i, j\} \in E} y_{ij} E_{ij} - J \geq 0 \right\}. \quad (6.13)$$

As both programs (6.8) and (6.13) are strictly feasible, there is no duality gap: the optimal value of (6.13) is equal to  $\vartheta(G)$ , and the optimal values are attained in both programs – here we have applied the duality theorem (Theorem 3.4.1).

Setting  $A = \sum_{\{i, j\} \in E} y_{ij} E_{ij}$ ,  $B = J - A$  and  $C = tI + A$  in (6.13), it follows that the program (6.13) is equivalent to (6.9), (6.10) and (6.11). Finally the formulation (6.12) follows directly from (6.10) after recalling that  $\lambda_{\max}(B)$  is the smallest scalar  $t$  for which  $tI - B \geq 0$ .  $\square$

### 6.4.2 Two more (lifted) formulations

We give here two more formulations for the theta number. They rely on semidefinite programs involving symmetric matrices of order  $1 + n$  which we will index by the set  $\{0\} \cup V$ , where 0 is an additional index that does not belong to  $V$ .

**Lemma 6.4.2.** *The theta number  $\vartheta(G)$  is equal to the optimal value of the following semidefinite program:*

$$\min_{Z \in \mathcal{S}^{n+1}} \{Z_{00} : Z \geq 0, Z_{0i} = Z_{ii} = 1 \ (i \in V), Z_{ij} = 0 \ (\{i, j\} \in \bar{E})\}. \quad (6.14)$$

*Proof.* We show that the two semidefinite programs in (6.9) and (6.14) are equivalent. For this, observe that

$$tI + A - J \geq 0 \iff Z := \begin{pmatrix} t & e^\top \\ e & I + \frac{1}{t}A \end{pmatrix} \geq 0,$$

which follows by taking the Schur complement of the upper left corner  $t$  in the block matrix  $Z$ . Hence, if  $(t, A)$  is feasible for (6.9), then  $Z$  is feasible for (6.14) with same objective value:  $Z_{00} = t$ . The construction can be reversed: if  $Z$  is feasible for (6.14), then one can construct  $(t, A)$  feasible for (6.9) with  $t = Z_{00}$ . Hence both programs are equivalent.  $\square$

From the formulation (6.14), the link to the chromatic number is even more transparent. Indeed, let  $k = \chi(\bar{G})$  and consider a partition  $V = C_1 \cup \dots \cup C_k$  of the node set into  $k$  cliques. For each clique  $C_i$  define the extended characteristic vector  $z_i = (1 \ \chi^{C_i}) \in \mathbb{R}^{1+n}$  obtained by appending an entry 1 to the characteristic vector of  $C_i$ . Define the matrix  $Z = \sum_{i=1}^k z_i z_i^\top \in \mathcal{S}^{1+n}$ . Then one can easily check that the matrix  $Z$  is feasible for the program (6.14) with objective value  $Z_{00} = k$ . Hence this shows again the inequality  $\vartheta(G) \leq \chi(\bar{G})$ .

Applying duality to the semidefinite program (6.14), we obtain<sup>1</sup> the following formulation for  $\vartheta(G)$ .

**Lemma 6.4.3.** *The theta number  $\vartheta(G)$  is equal to the optimal value of the following semidefinite program:*

$$\max_{Y \in \mathcal{S}^{n+1}} \left\{ \sum_{i \in V} Y_{ii} : Y \geq 0, Y_{00} = 1, Y_{0i} = Y_{ii} \ (i \in V), Y_{ij} = 0 \ (\{i, j\} \in E) \right\}. \quad (6.15)$$

*Proof.* One can verify that the dual program of (6.14) reads

$$\max \left\{ - \sum_{i \in V} Y_{ii} + 2Y_{0i} : Y \geq 0, Y_{00} = 1, Y_{ij} = 0 \ (\{i, j\} \in E) \right\} \quad (6.16)$$

<sup>1</sup>Of course there is more than one road leading to Rome: one can also show directly the equivalence of the two programs (6.8) and (6.15).

(check it). As (6.14) is strictly feasible (check it) there is no duality gap, the optimal value of (6.16) is attained and it is equal to  $\vartheta(G)$ . Note that the program (6.15) amounts to adding the constraints  $Y_{ii} = Y_{0i}$  ( $i \in V$ ) to the program (6.16). In order to show that both programs (6.15) and (6.16) are equivalent, it suffices to show that (6.16) admits an optimal solution satisfying these additional constraints.

For this pick an optimal solution  $Y$  to (6.16). In a first step, we show that  $Y_{0i} + Y_{ii} = 0$  for all  $i \in V$ . Indeed, assume that  $Y_{0i} + Y_{ii} \neq 0$  for some  $i \in V$ . Then,  $Y_{ii} \neq 0$ . Let us multiply the  $i$ -th column and the  $i$ -th row of the matrix  $Y$  by the scalar  $-\frac{Y_{0i}}{Y_{ii}}$ . In this way we obtain a new matrix  $Y'$  which is still feasible for (6.16), but with a better objective value: Indeed,  $Y'_{ii} = Y_{ii} \left(-\frac{Y_{0i}}{Y_{ii}}\right)^2 = \frac{Y_{0i}^2}{Y_{ii}}$  and  $Y'_{0i} = -\frac{Y_{0i}^2}{Y_{ii}}$ , so that the  $i$ -th term in the new objective value is

$$-(Y'_{ii} + 2Y'_{0i}) = \frac{Y_{0i}^2}{Y_{ii}} > -(Y_{ii} + 2Y_{0i}).$$

Hence,  $Y_{0i} = -Y_{ii}$  for all  $i \in V$ . Now, we can change the signs on the first row and column of  $Y$  (indexed by the index 0). In this way we obtain a new optimal solution of (6.16) which now satisfies the conditions:  $Y_{ii} = Y_{0i}$  for  $i \in V$ .  $\square$

As explained in Chapter 5 one can define a hierarchy of semidefinite bounds for  $\alpha(G)$ , strengthening the theta number  $\vartheta(G)$ . While  $\vartheta(G)$  is defined using matrices indexed by  $\{0\} \cup V$ , these stronger bounds are obtained by considering matrices indexed by larger index sets, thus lifting the problem in even larger dimensions – see Section 5.2.3 for details.

## 6.5 Geometric properties of the theta number

### 6.5.1 Orthonormal representations

**Definition 6.5.1.** An orthonormal representation of  $G$ , abbreviated as ONR, consists of a set of unit vectors  $\{u_1, \dots, u_n\} \subseteq \mathbb{R}^d$  (for some  $d \geq 1$ ) satisfying

$$u_i^\top u_j = 0 \quad \forall \{i, j\} \in \overline{E}.$$

If  $S$  is a stable set in  $G$  and the  $u_i$ 's form an ONR of  $G$  of dimension  $d$ , then the vectors  $u_i$  labeling the nodes of  $S$  are pairwise orthogonal, which implies that  $d \geq \alpha(G)$ . It turns out that the stronger lower bound:  $d \geq \vartheta(G)$  holds.

**Lemma 6.5.2.** The minimum dimension  $d$  of an orthonormal representation of a graph  $G$  satisfies:  $\vartheta(G) \leq d$ .

*Proof.* Let  $u_1, \dots, u_n \in \mathbb{R}^d$  be an ONR of  $G$ . Define the matrices  $U_0 = I_d$ ,  $U_i = u_i u_i^\top \in \mathcal{S}^d$  for  $i \in [n]$ . Now we define a symmetric matrix  $Z \in \mathcal{S}^{n+1}$  by setting  $Z_{ij} = \langle U_i, U_j \rangle$  for  $i, j \in \{0\} \cup [n]$ . One can verify that  $Z$  is feasible for the program (6.14) defining  $\vartheta(G)$  (check it) with  $Z_{00} = d$ . This gives  $\vartheta(G) \leq d$ .  $\square$



### 6.5.2 The theta body $\text{TH}(G)$

It is convenient to introduce the following set of matrices  $X \in \mathcal{S}^{n+1}$ , where columns and rows are indexed by the set  $\{0\} \cup V$ :

$$\mathcal{M}_G = \{Y \in \mathcal{S}^{n+1} : Y_{00} = 1, Y_{0i} = Y_{ii} \ (i \in V), Y_{ij} = 0 \ (\{i, j\} \in E), Y \geq 0\}, \quad (6.17)$$

which is thus the feasible region of the semidefinite program (6.15). Now let  $\text{TH}(G)$  denote the convex set obtained by projecting the set  $\mathcal{M}_G$  onto the subspace  $\mathbb{R}^V$  of the diagonal entries:

$$\text{TH}(G) = \{x \in \mathbb{R}^V : \exists Y \in \mathcal{M}_G \text{ such that } x_i = Y_{ii} \ \forall i \in V\}, \quad (6.18)$$

called the *theta body* of  $G$ . It turns out that  $\text{TH}(G)$  is nested between  $\text{ST}(G)$  and  $\text{QST}(G)$ .

**Lemma 6.5.3.** *For any graph  $G$ , we have that  $\text{ST}(G) \subseteq \text{TH}(G) \subseteq \text{QST}(G)$ .*

*Proof.* The inclusion  $\text{ST}(G) \subseteq \text{TH}(G)$  follows from the fact that the characteristic vector of any stable set lies in  $\text{TH}(G)$  (check it). We now show the inclusion  $\text{TH}(G) \subseteq \text{QST}(G)$ . For this pick a vector  $x \in \text{TH}(G)$  and a clique  $C$  of  $G$ ; we show that  $x(C) \leq 1$ . Say  $x_i = Y_{ii}$  for all  $i \in V$ , where  $Y \in \mathcal{M}_G$ . Consider the principal submatrix  $Y_C$  of  $Y$  indexed by  $\{0\} \cup C$ , which is of the form

$$Y_C = \begin{pmatrix} 1 & x_C^\top \\ x_C & \text{Diag}(x_C) \end{pmatrix},$$

where we set  $x_C = (x_i)_{i \in C}$ . Now,  $Y_C \geq 0$  implies that  $\text{Diag}(x_C) - x_C x_C^\top \geq 0$  (taking a Schur complement). This in turn implies:  $e^\top (\text{Diag}(x_C) - x_C x_C^\top) e \geq 0$ , which can be rewritten as  $x(C) - (x(C))^2 \geq 0$ , giving  $x(C) \leq 1$ .  $\square$

In view of Lemma 6.4.3, maximizing the all-ones objective function over  $\text{TH}(G)$  gives the theta number:

$$\vartheta(G) = \max_{x \in \mathbb{R}^V} \{e^\top x : x \in \text{TH}(G)\}.$$

As maximizing  $e^\top x$  over  $\text{QST}(G)$  gives the LP bound  $\alpha^*(G)$ , Lemma 6.5.3 implies directly that the SDP bound  $\vartheta(G)$  dominates the LP bound  $\alpha^*(G)$ :

**Corollary 6.5.4.** *For any graph  $G$ , we have that  $\alpha(G) \leq \vartheta(G) \leq \alpha^*(G)$ .*

Combining the inclusion from Lemma 6.5.3 with Theorem 6.2.2, we deduce that  $\text{TH}(G) = \text{ST}(G) = \text{QST}(G)$  for perfect graphs. It turns out that these equalities characterize perfect graphs.

**Theorem 6.5.5.** *For any graph  $G$  the following assertions are equivalent.*

1.  $G$  is perfect.
2.  $\text{TH}(G) = \text{ST}(G)$
3.  $\text{TH}(G) = \text{QST}(G)$ .
4.  $\text{TH}(G)$  is a polytope.

### 6.5.3 More on the theta body

There is a beautiful relationship between the theta bodies of a graph  $G$  and of its complementary graph  $\overline{G}$ :

**Theorem 6.5.6.** *For any graph  $G$ ,*

$$\text{TH}(G) = \{x \in \mathbb{R}_{\geq 0}^V : x^T z \leq 1 \ \forall z \in \text{TH}(\overline{G})\}.$$

In other words, we know an explicit linear inequality description of  $\text{TH}(G)$ ; moreover, the normal vectors to the supporting hyperplanes of  $\text{TH}(G)$  are precisely the elements of  $\text{TH}(\overline{G})$ . One inclusion is easy:

**Lemma 6.5.7.** *If  $x \in \text{TH}(G)$  and  $z \in \text{TH}(\overline{G})$  then  $x^T z \leq 1$ .*

*Proof.* Let  $Y \in \mathcal{M}_G$  and  $Z \in \mathcal{M}_{\overline{G}}$  such that  $x = (Y_{ii})$  and  $z = (Z_{ii})$ . Let  $Z'$  be obtained from  $Z$  by changing signs in its first row and column (indexed by 0). Then  $\langle Y, Z' \rangle \geq 0$  as  $Y, Z' \geq 0$ . Moreover,  $\langle Y, Z' \rangle = 1 - x^T z$  (check it), thus giving  $x^T z \leq 1$ .  $\square$

Next we observe how the elements of  $\text{TH}(G)$  can be expressed in terms of orthonormal representations of  $\overline{G}$ .

**Lemma 6.5.8.** *For  $x \in \mathbb{R}_{\geq 0}^V$ ,  $x \in \text{TH}(G)$  if and only if there exist an orthonormal representation  $v_1, \dots, v_n$  of  $\overline{G}$  and a unit vector  $d$  such that  $x = ((d^T v_i)^2)_{i \in V}$ .*

*Proof.* Let  $d, v_i$  be unit vectors where the  $v_i$ 's form an ONR of  $\overline{G}$ ; we show that  $x = ((d^T v_i)^2) \in \text{TH}(G)$ . For this, let  $Y \in \mathcal{S}^{n+1}$  denote the Gram matrix of the vectors  $d$  and  $(v_i^T d)v_i$  for  $i \in V$ , so that  $x = (Y_{ii})$ . One can verify that  $Y \in \mathcal{M}_G$ , which implies  $x \in \text{TH}(G)$ .

For the reverse inclusion, pick  $Y \in \mathcal{M}_G$  and a Gram representation  $w_0, w_i$  ( $i \in V$ ) of  $Y$ . Set  $d = w_0$  and  $v_i = w_i / \|w_i\|$  for  $i \in V$ . Then the conditions expressing membership of  $Y$  in  $\mathcal{M}_G$  imply that the  $v_i$ 's form an ONR of  $\overline{G}$ ,  $\|d\| = 1$ , and  $Y_{ii} = (d^T v_i)^2$  for all  $i \in V$ .  $\square$

To conclude the proof of Theorem 6.5.6 we use the following result, which characterizes which partially specified matrices can be completed to a positive semidefinite matrix – you will prove it in Exercise 6.1.

**Proposition 6.5.9.** *Let  $H = (W, F)$  be a graph and let  $a_{ij}$  ( $i = j \in W$  or  $\{i, j\} \in F$ ) be given scalars, corresponding to a vector  $a \in \mathbb{R}^{W \cup F}$ . Define the convex set*

$$K_a = \{Y \in \mathcal{S}^W : Y \geq 0, Y_{ij} = a_{ij} \ \forall i = j \in W \text{ and } \{i, j\} \in F\} \quad (6.19)$$

*(consisting of all possible positive semidefinite completions of  $a$ ) and the cone*

$$C_H = \{Z \in \mathcal{S}^W : Z \geq 0, Z_{ij} = 0 \ \forall \{i, j\} \in \overline{F}\} \quad (6.20)$$

*(consisting of all positive semidefinite matrices supported by the graph  $H$ ). Then,  $K_a \neq \emptyset$  if and only if*

$$\sum_{i \in W} a_{ii} Z_{ii} + 2 \sum_{\{i, j\} \in F} a_{ij} Z_{ij} \geq 0 \ \forall Z \in C_H. \quad (6.21)$$

*Proof.* (of Theorem 6.5.6). Let  $x \in \mathbb{R}_{\geq 0}^V$  such that  $x^\top z \leq 1$  for all  $z \in \text{TH}(\overline{G})$ ; we show that  $x \in \text{TH}(G)$ . For this we need to find a matrix  $Y \in \mathcal{M}_G$  such that  $x = (Y_{ii})_{i \in V}$ . In other words, the entries of  $Y$  are specified already at the following positions:  $Y_{00} = 1$ ,  $Y_{0i} = Y_{ii} = x_i$  for  $i \in V$ , and  $Y_{\{i,j\}} = 0$  for all  $\{i,j\} \in E$ , and we need to show that the remaining entries (at the positions of non-edges of  $G$ ) can be chosen in such a way that  $Y \geq 0$ .

To show this we apply Proposition 6.5.9, where the graph  $H$  is  $G$  with an additional node 0 adjacent to all  $i \in V$ . Hence it suffices now to show that  $\langle Y, Z \rangle \geq 0$  for all  $Z \in \mathcal{S}_{\geq 0}^{\{0\} \cup V}$  with  $Z_{ij} = 0$  if  $\{i,j\} \in \overline{E}$ . Pick such  $Z$ , with Gram representation  $w_0, w_1, \dots, w_n$ . Then  $w_i^\top w_j = 0$  if  $\{i,j\} \in \overline{E}$ . We can assume without loss of generality that all  $w_i$  are non-zero (use continuity if some  $w_i$  is zero) and up to scaling that  $w_0$  is a unit vector. Then the vectors  $w_i / \|w_i\|$  ( $i \in V$ ) form an ONR of  $G$ . By Lemma 6.5.8 (applied to  $\overline{G}$ ), the vector  $z \in \mathbb{R}^V$  with  $z_i = (w_0^\top w_i)^2 / \|w_i\|^2$  belongs to  $\text{TH}(\overline{G})$  and thus  $x^\top z \leq 1$  by assumption. Therefore,  $\langle Y, Z \rangle$  is equal to

$$\begin{aligned} 1 + 2 \sum_{i \in V} x_i w_0^\top w_i + \sum_{i \in V} x_i \|w_i\|^2 &\geq \sum_{i \in V} x_i \left( \frac{(w_0^\top w_i)^2}{\|w_i\|^2} + 2w_0^\top w_i + \|w_i\|^2 \right) \\ &= \sum_{i \in V} x_i \left( \frac{w_0^\top w_i}{\|w_i\|} + \|w_i\| \right)^2 \geq 0. \end{aligned}$$

□

## 6.6 The theta number for vertex-transitive graphs

First we mention an inequality relating the theta numbers of a graph and its complement.

**Proposition 6.6.1.** *For any graph  $G = (V, E)$ , we have that  $\vartheta(G)\vartheta(\overline{G}) \geq |V|$ .*

*Proof.* Using the formulation of the theta number from (6.11), we obtain matrices  $C, C' \in \mathcal{S}^n$  such that  $C - J, C' - J \geq 0$ ,  $C_{ii} = \vartheta(G)$ ,  $C'_{ii} = \vartheta(\overline{G})$  for  $i \in V$ ,  $C_{ij} = 0$  for  $\{i,j\} \in \overline{E}$  and  $C'_{ij} = 0$  for  $\{i,j\} \in E$ . Then, we have that  $\langle C, J \rangle, \langle C', J \rangle \geq \langle J, J \rangle = n^2$ , and  $\langle C, C' \rangle = n\vartheta(G)\vartheta(\overline{G})$ . Combining these facts yields the desired inequality. □

We now show that equality  $\vartheta(G)\vartheta(\overline{G}) = |V|$  holds for certain symmetric graphs, namely for vertex-transitive graphs. In order to show this, one exploits in a crucial manner the symmetry of  $G$ , which permits to show that the semidefinite program defining the theta number has an optimal solution with a special (symmetric) structure. We need to introduce some definitions.

Let  $G = (V, E)$  be a graph. A permutation  $\sigma$  of the node set  $V$  is called an *automorphism* of  $G$  if it preserves edges, i.e.,  $\{i,j\} \in E$  implies  $\{\sigma(i), \sigma(j)\} \in E$ . Then the set  $\text{Aut}(G)$  of automorphisms of  $G$  is a group. The graph  $G$  is said to

be *vertex-transitive* if for any two nodes  $i, j \in V$  there exists an automorphism  $\sigma \in \text{Aut}(G)$  mapping  $i$  to  $j$ :  $\sigma(i) = j$ .

The group of permutations of  $V$  acts on symmetric matrices  $X$  indexed by  $V$ . Namely, if  $\sigma$  is a permutation of  $V$  and  $P_\sigma$  is the corresponding permutation matrix (with  $P_\sigma(i, j) = P_{\sigma(i), \sigma(j)}$  for all  $i, j \in V$ ), then one can build the new symmetric matrix

$$\sigma(X) := P_\sigma X P_\sigma^\top = (X_{\sigma(i), \sigma(j)})_{i, j \in V}.$$

If  $\sigma$  is an automorphism of  $G$ , then it preserves the feasible region of the semidefinite program (6.8) defining the theta number  $\vartheta(G)$ . This is an easy, but very useful fact.

**Lemma 6.6.2.** *If  $X$  is feasible for the program (6.8) and  $\sigma$  is an automorphism of  $G$ , then  $\sigma(X)$  is again feasible for (6.8), moreover with the same objective value as  $X$ .*

*Proof.* Directly from the fact that  $\langle J, \sigma(X) \rangle = \langle J, X \rangle$ ,  $\text{Tr}(\sigma(X)) = \text{Tr}(X)$  and  $\sigma(X)_{ij} = X_{\sigma(i)\sigma(j)} = 0$  if  $\{i, j\} \in E$  (since  $\sigma$  is an automorphism of  $G$ ).  $\square$

**Lemma 6.6.3.** *The program (6.8) has an optimal solution  $X^*$  which is invariant under action of the automorphism group of  $G$ , i.e., satisfies  $\sigma(X^*) = X^*$  for all  $\sigma \in \text{Aut}(G)$ .*

*Proof.* Let  $X$  be an optimal solution of (6.8). By Lemma 6.6.2,  $\sigma(X)$  is again an optimal solution for each  $\sigma \in \text{Aut}(G)$ . Define the matrix

$$X^* = \frac{1}{|\text{Aut}(G)|} \sum_{\sigma \in \text{Aut}(G)} \sigma(X),$$

obtained by averaging over all matrices  $\sigma(X)$  for  $\sigma \in \text{Aut}(G)$ . As the set of optimal solutions of (6.8) is convex,  $X^*$  is still an optimal solution of (6.8). Moreover, by construction,  $X^*$  is invariant under action of  $\text{Aut}(G)$ .  $\square$

**Corollary 6.6.4.** *If  $G$  is a vertex-transitive graph then the program (6.8) has an optimal solution  $X^*$  satisfying  $X_{ii}^* = 1/n$  for all  $i \in V$  and  $X^*e = \frac{\vartheta(G)}{n}e$ .*

*Proof.* By Lemma 6.6.3, there is an optimal solution  $X^*$  which is invariant under action of  $\text{Aut}(G)$ . As  $G$  is vertex-transitive, all diagonal entries of  $X^*$  are equal: Indeed, let  $i, j \in V$  and  $\sigma \in \text{Aut}(G)$  such that  $\sigma(i) = j$ . Then,  $X_{jj}^* = X_{\sigma(i)\sigma(i)}^* = X_{ii}^*$ . As  $\text{Tr}(X^*) = 1$  we must have  $X_{ii}^* = 1/n$  for all  $i$ . Analogously, the invariance of  $X^*$  implies that  $\sum_{k \in V} X_{ik}^* = \sum_{k \in V} X_{jk}^*$  for all  $i, j$ , i.e.,  $X^*e = \lambda e$  for some scalar  $\lambda$ . Combining with the condition  $\langle J, X^* \rangle = \vartheta(G)$  we obtain that  $\lambda = \frac{\vartheta(G)}{n}$ .  $\square$

**Proposition 6.6.5.** *If  $G$  is a vertex-transitive graph, then  $\vartheta(G)\vartheta(\overline{G}) = |V|$ .*

*Proof.* By Corollary 6.6.4, there is an optimal solution  $X^*$  of the program (6.8) defining  $\vartheta(G)$  which satisfies  $X_{ii}^* = 1/n$  for  $i \in V$  and  $X^*e = \frac{\vartheta(G)}{n}e$ . Then  $\frac{n^2}{\vartheta(G)}X^* - J \geq 0$  (check it). Hence,  $t = \frac{n}{\vartheta(G)}$  and  $C = \frac{n^2}{\vartheta(G)}X^*$  define a feasible solution of the program (6.11) defining  $\vartheta(\overline{G})$ , which implies  $\vartheta(\overline{G}) \leq n/\vartheta(G)$ . Combining with Proposition 6.6.1 we get the equality  $\vartheta(G)\vartheta(\overline{G}) = |V|$ .  $\square$

For instance, the cycle  $C_n$  is vertex-transitive, so that

$$\vartheta(C_n)\vartheta(\overline{C_n}) = n. \quad (6.22)$$

In particular, as  $C_5$  is isomorphic to  $\overline{C_5}$ , we deduce that

$$\vartheta(C_5) = \sqrt{5}. \quad (6.23)$$

For  $n$  even,  $C_n$  is bipartite (and thus perfect), so that  $\vartheta(C_n) = \alpha(C_n) = \frac{n}{2}$  and  $\vartheta(\overline{C_n}) = \omega(C_n) = 2$ . For  $n$  odd, one can compute  $\vartheta(C_n)$  using the above symmetry reduction:

**Proposition 6.6.6.** *For any odd  $n \geq 3$ ,*

$$\vartheta(C_n) = \frac{n \cos(\pi/n)}{1 + \cos(\pi/n)} \quad \text{and} \quad \vartheta(\overline{C_n}) = \frac{1 + \cos(\pi/n)}{\cos(\pi/n)}.$$

*Proof.* As  $\vartheta(C_n)\vartheta(\overline{C_n}) = n$ , it suffices to compute  $\vartheta(C_n)$ . We use the formulation (6.12). As  $C_n$  is vertex-transitive, there is an optimal solution  $B$  whose entries are all equal to 1, except  $B_{ij} = 1 + x$  for some scalar  $x$  whenever  $|i - j| = 1$  (modulo  $n$ ). In other words,  $B = J + xA_{C_n}$ , where  $A_{C_n}$  is the adjacency matrix of the cycle  $C_n$ . Thus  $\vartheta(C_n)$  is equal to the minimum value of  $\lambda_{\max}(B)$  for all possible  $x$ . The eigenvalues of  $A_{C_n}$  are known: They are  $\omega^k + \omega^{-k}$  (for  $k = 0, 1, \dots, n-1$ ), where  $\omega = e^{\frac{2i\pi}{n}}$  is an  $n$ -th root of unity. Hence the eigenvalues of  $B$  are

$$n + 2x \quad \text{and} \quad x(\omega^k + \omega^{-k}) \quad \text{for } k = 1, \dots, n-1. \quad (6.24)$$

We minimize the maximum of the values in (6.24) when choosing  $x$  such that

$$n + 2x = -2x \cos(\pi/n)$$

(check it). This gives  $\vartheta(C_n) = \lambda_{\max}(B) = -2x \cos(\pi/n) = \frac{n \cos(\pi/n)}{1 + \cos(\pi/n)}$ .  $\square$

## 6.7 Bounding the Shannon capacity

The theta number was introduced by Lovász [3] in connection with the problem of computing the Shannon capacity of a graph, a problem in coding theory considered by Shannon. We need some definitions.

**Definition 6.7.1. (Strong product)** Let  $G = (V, E)$  and  $H = (W, F)$  be two graphs. Their strong product is the graph denoted as  $G \cdot H$  with node set  $V \times W$  and with edges the pairs of distinct nodes  $\{(i, r), (j, s)\} \in V \times W$  with  $(i = j \text{ or } \{i, j\} \in E)$  and  $(r = s \text{ or } \{r, s\} \in F)$ .

If  $S \subseteq V$  is stable in  $G$  and  $T \subseteq W$  is stable in  $H$  then  $S \times T$  is stable in  $G \cdot H$ . Hence,  $\alpha(G \cdot H) \geq \alpha(G)\alpha(H)$ . Let  $G^k$  denote the strong product of  $k$  copies of  $G$ , we have that

$$\alpha(G^k) \geq (\alpha(G))^k.$$

Based on this, one can verify that

$$\Theta(G) := \sup_{k \geq 1} \sqrt[k]{\alpha(G^k)} = \lim_{k \rightarrow \infty} \sqrt[k]{\alpha(G^k)}. \quad (6.25)$$

The parameter  $\Theta(G)$  was introduced by Shannon in 1956, it is called the *Shannon capacity* of the graph  $G$ . The motivation is as follows. Suppose  $V$  is a finite alphabet, where some pairs of letters could be confused when they are transmitted over some transmission channel. These pairs of confusable letters can be seen as the edge set  $E$  of a graph  $G = (V, E)$ . Then the stability number of  $G$  is the largest number of one-letter messages that can be sent without danger of confusion. Words of length  $k$  correspond to  $k$ -tuples in  $V^k$ . Two words  $(i_1, \dots, i_k)$  and  $(j_1, \dots, j_k)$  can be confused if at every position  $h \in [k]$  the two letters  $i_h$  and  $j_h$  are equal or can be confused, which corresponds to having an edge in the strong product  $G^k$ . Hence the largest number of words of length  $k$  that can be sent without danger of confusion is equal to the stability number of  $G^k$  and the Shannon capacity of  $G$  represents the rate of correct transmission of the graph.

For instance, for the 5-cycle  $C_5$ ,  $\alpha(C_5) = 2$ , but  $\alpha((C_5)^2) \geq 5$ . Indeed, if 1, 2, 3, 4, 5 are the nodes of  $C_5$  (in this cyclic order), then the five 2-letter words (1, 1), (2, 3), (3, 5), (4, 2), (5, 4) form a stable set in  $G^2$ . This implies that  $\Theta(C_5) \geq \sqrt{5}$ .

Determining the exact Shannon capacity of a graph is a very difficult problem in general, even for small graphs. For instance, the exact value of the Shannon capacity of  $C_5$  was not known until Lovász [3] showed how to use the theta number in order to upper bound the Shannon capacity: Lovász showed that  $\Theta(G) \leq \vartheta(G)$  and  $\vartheta(C_5) = \sqrt{5}$ , which implies that  $\Theta(C_5) = \sqrt{5}$ . For instance, although the exact value of the theta number of  $C_{2n+1}$  is known (cf. Proposition 6.6.6), the exact value of the Shannon capacity of  $C_{2n+1}$  is not known, already for  $C_7$ .

**Theorem 6.7.2.** For any graph  $G$ , we have that  $\Theta(G) \leq \vartheta(G)$ .

The proof is based on the multiplicative property of the theta number from Lemma 6.7.3 – which you will prove in Exercise 6.2 – combined with the fact that the theta number upper bounds the stability number: For any integer  $k$ ,  $\alpha(G^k) \leq \vartheta(G^k) = (\vartheta(G))^k$  implies  $\sqrt[k]{\alpha(G^k)} \leq \vartheta(G)$  and thus  $\Theta(G) \leq \vartheta(G)$ .

**Lemma 6.7.3.** *The theta number of the strong product of two graphs  $G$  and  $H$  satisfies  $\vartheta(G \cdot H) = \vartheta(G)\vartheta(H)$ .*

As an application one can compute the Shannon capacity of any graph  $G$  which is vertex-transitive and self-complementary (e.g., like  $C_5$ ).

**Theorem 6.7.4.** *If  $G = (V, E)$  is a vertex-transitive graph, then  $\Theta(G \cdot \overline{G}) = |V|$ . If, moreover,  $G$  is self-complementary, then  $\Theta(G) = \sqrt{|V|}$ .*

*Proof.* We have  $\Theta(G \cdot \overline{G}) \geq \alpha(G \cdot \overline{G}) \geq |V|$ , since the set of diagonal pairs  $\{(i, i) : i \in V\}$  is stable in  $G \cdot \overline{G}$ . The reverse inequality follows from Lemma 6.7.3 combined with Proposition 6.6.5:  $\Theta(G \cdot \overline{G}) \leq \vartheta(G \cdot \overline{G}) = \vartheta(G)\vartheta(\overline{G}) = |V|$ . If  $G$  is isomorphic to  $\overline{G}$  then  $\Theta(G \cdot \overline{G}) = \Theta(G^2) = (\Theta(G))^2$  (check the rightmost equality). This gives  $\Theta(G) = \sqrt{|V|}$ .  $\square$

## 6.8 Further reading

In his seminal paper [3], Lovász gives several equivalent formulations for the theta number, and relates it to the Shannon capacity and to some eigenvalue bounds. It is worth noting that Lovász' paper was published in 1979, thus before the discovery of polynomial time algorithms for semidefinite programming. In 1981, together with Grötschel and Schrijver, he derived the polynomial time algorithms for maximum stable sets and graph colorings in perfect graphs, based on the ellipsoid method for solving semidefinite programs. As of today, this is the only known polynomial time algorithm – in particular, no purely combinatorial algorithm is known.

Detailed information about the theta number can also be found in the survey of Knuth [2] and a detailed treatment about the material in this chapter can be found in Chapter 9 of Grötschel, Lovász and Schrijver [1]. In particular, proofs of the geometric characterizations of perfect graphs in Theorems 6.2.2 and 6.5.5 can be found there. Weighted versions of the theta number are considered there, replacing the all-ones objective function  $e^\top x$  by  $w^\top x$  where  $w \in \mathbb{Z}_{\geq 0}^V$ . One can give equivalent characterizations, analogue to those we have given for the all-ones weight function. We have restricted our exposition to the all-ones weight function for the sake of simplicity.

## 6.9 Exercises

6.1 Show the result of Proposition 6.5.9.

6.2\*\* The goal is to show the result of Lemma 6.7.3 about the theta number of the strong product of two graphs  $G = (V, E)$  and  $H = (W, F)$ :

$$\vartheta(G \cdot H) = \vartheta(G)\vartheta(H).$$

(a) Show that  $\vartheta(G \cdot H) \geq \vartheta(G)\vartheta(H)$ .

(b) Show that  $\vartheta(G \cdot H) \leq \vartheta(G)\vartheta(H)$ .

Hint: Use the primal formulation (6.8) for (a), and the dual formulation (6.9) for (b), and think of using Kronecker products of matrices in order to build feasible solutions.

6.3 Let  $G = (V = [n], E)$  be a graph. Consider the graph parameter

$$\vartheta_1(G) = \min_{c, u_i} \max_{i \in V} \frac{1}{(c^\top u_i)^2},$$

where the minimum is taken over all unit vectors  $c$  and all orthonormal representations  $u_1, \dots, u_n$  of  $G$ .

Show that  $\vartheta(G) = \vartheta_1(G)$ .

Hint: For the inequality  $\vartheta(G) \leq \vartheta_1(G)$  think of using the properties of the theta body from Section 6.5.2. For the inequality  $\vartheta_1(G) \leq \vartheta(G)$ , use an optimal solution  $B$  of the dual formulation (6.10) for  $\vartheta(G)$  to build the vectors  $c, u_i$ .



---

## BIBLIOGRAPHY

- [1] M. Grötschel, L. Lovász, A. Schrijver. *Geometric Algorithms in Combinatorial Optimization*, Springer, 1988.  
<http://www.zib.de/groetschel/pubnew/paper/groetschellovaszschrijver1988.pdf>
- [2] D.E. Knuth. The Sandwich Theorem. *The Electronic Journal of Combinatorics* **1**, **A1**, 1994.  
<http://www.combinatorics.org/ojs/index.php/eljc/article/view/v1i1a1>
- [3] L. Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory* **IT-25**:1–7, 1979.

---



---

# CHAPTER 7

---

## APPROXIMATING MAX CUT AND THE CUT NORM

The maximum cut problem (MAX CUT) is the following problem in combinatorial optimization. Let  $G = (V, E)$  be an undirected graph with vertex set  $V$  and edge set  $E \subseteq \binom{V}{2}$ , where edges  $e = \{u, v\} \in E$  are two-element subsets of the vertex set. With every edge  $e = \{u, v\}$  we associate a nonnegative weight  $w_{uv}$ . Since the graph is undirected we assume that the weights are symmetric  $w_{uv} = w_{vu}$ . Furthermore,  $w_{uv} = 0$  whenever  $\{u, v\} \notin E$ . We incorporate all the weights into a symmetric matrix  $W = (w_{uv}) \in \mathcal{S}^V$ . The MAX CUT problem seeks for a partition of the vertex set  $V$  into two parts  $V^-, V^+$ , cutting the graph into two pieces, so that the sum of edges connecting  $V^-$  and  $V^+$ , the *weight of the cut*  $w(V^-, V^+)$ , is maximized:

$$w(V^-, V^+) = \sum_{u \in V^-, v \in V^+} w_{uv}.$$

It is known that the maximum cut problem is an NP-complete problem. So unless the complexity classes P and NP coincide there is no efficient polynomial-time algorithm which solves MAX CUT exactly. In fact, MAX CUT was one of the first problems which were proved to be NP-complete: It is one of Karp's 21 NP-complete problems. Even stronger, Håstad in 2001 showed that it is NP-hard to approximate MAX CUT within a factor of  $\frac{16}{17} = 0.941\dots$ . This is in sharp contrast to the MIN CUT problem, where we want to minimize the weight of a non-trivial cut. The MIN CUT problem (and its dual, the MAX FLOW problem) can be solved using linear programming.

On the positive side, one can compute an  $0.878\dots$ -approximation of MAX CUT in polynomial time, using semidefinite programming. This algorithm, due

to Goemans and Williamson [2], is one of the most influential approximation algorithms which are based on semidefinite programming.

A problem which is related to (in fact a generalization of) MAX CUT is finding the cut norm of a matrix. Let  $A = (A_{ij}) \in \mathbb{R}^{m \times n}$  be a real matrix. The *cut norm* of  $A$  is

$$\|A\|_{\square} = \max_{I \subseteq [m], J \subseteq [n]} \left| \sum_{i \in I} \sum_{j \in J} A_{ij} \right|.$$

Computing the cut norm of a matrix has many applications in combinatorics, especially in graph theory. Examples are finding Szemerédi partitions in graphs, or finding maximum acyclic subgraphs. As the cut norm is a generalization of the MAX CUT problem we only can hope for efficient approximations. Today in this lecture we link the cut norm with Grothendieck's inequality, a famous inequality in functional analysis from an even more famous mathematician. Thereby we will derive another approximation algorithm based on semidefinite programming which is due to Krivine from 1979 (although this connection was only found in 2006).

## 7.1 The algorithm of Goemans and Williamson

### 7.1.1 Semidefinite relaxation

We now want to describe the Goemans-Williamson algorithm. For this we first reformulate MAX CUT as a (non-convex) quadratic optimization problem having quadratic equality constraints. We start by recalling the construction of the semidefinite relaxation of the MAX CUT problem which we already saw in Chapter 2 and Chapter 5.

With every vertex of the graph  $u \in V$  we associate a binary variable  $x_u \in \{-1, +1\}$  which indicates whether  $u$  lies in  $V^-$  or  $V^+$ , i.e.  $u \in V^-$  if  $x_u = -1$  and  $u \in V^+$  if  $x_u = +1$ . We model the binary constraint  $x_u \in \{-1, +1\}$  as a quadratic equality constraint

$$x_u^2 = 1, \quad u \in V.$$

For two vertices  $u, v \in V$  we have

$$1 - x_u x_v \in \{0, 2\}.$$

This value equals to 0 if  $u$  and  $v$  lie on the same side of the cut and the value equals to 2 if  $u$  and  $v$  lie on different sides of the cut. Hence, one can express the weight of a cut, which is defined by the variables  $x_u \in \{-1, +1\}$ , by

$$w(V^-, V^+) = \frac{1}{2} \left( \frac{1}{2} \sum_{u, v \in V} w_{uv} (1 - x_u x_v) \right).$$

Now, the MAX CUT problem can be equivalently formulated as

$$\text{MAXCUT}(W) = \max \left\{ \frac{1}{4} \sum_{u,v \in V} w_{uv}(1 - x_u x_v) : x_u^2 = 1, u \in V \right\}.$$

If we replace in this optimization problem the scalar variables  $x_u \in \{-1, +1\}$  by vector variables  $y_u \in \mathbb{R}^{|V|}$  lying in  $|V|$ -dimensional Euclidean space, and the product  $x_u x_v$  by the inner product  $y_u \cdot y_v = y_u^T y_v$ , then we get the following *vector optimization problem*:

$$\text{SDP}(W) = \max \left\{ \frac{1}{4} \sum_{u,v \in V} w_{uv}(1 - y_u \cdot y_v) : y_u \cdot y_u = 1, u \in V \right\}.$$

Because  $y_u \cdot y_u = 1$ , we see that the vectors  $y_u$  lie on the unit sphere  $S^{|V|-1}$ . Note also, that every feasible solution  $x_u$  of the original problem can be transformed into a feasible solution of the vector optimization problem. We simply set  $y_u = (x_u, 0, \dots, 0)^T$ . This means that the maximum value of the vector optimization problem is at least the value of MAX CUT, thus  $\text{SDP}(W) \geq \text{MAXCUT}(W)$ .

We proceed by showing two things: First, it is not difficult to realize that the vector optimization problem can be reformulated as a semidefinite program. Second, we shall prove that the maximum of the vector optimization problem is not too far from the optimal value of the original MAX CUT problem. We show that the inequality

$$\text{SDP}(W) \geq \text{MAXCUT}(W) \geq 0.878 \dots \cdot \text{SDP}(W)$$

holds for all symmetric weight matrices  $W = (w_{uv})$  with nonnegative entries  $w_{uv}$ .

To show that the vector optimization problem is a semidefinite program we introduce the inner product matrix of the vectors  $y_u$ :

$$Y = (y_u \cdot y_v)_{u,v \in V}.$$

The matrix  $Y$  is a positive semidefinite matrix whose diagonal elements are all equal to one. Furthermore,

$$\frac{1}{4} \sum_{u,v \in V} w_{uv}(1 - y_u \cdot y_v) = \frac{1}{4} \sum_{u,v \in V} w_{uv} - \frac{1}{4} \langle W, Y \rangle.$$

So it suffices to minimize the trace inner product  $\langle W, Y \rangle$  in order to solve the vector optimization problem. Hence, solving the following semidefinite program gives the value of  $\text{SDP}(W)$ :

$$\text{SDP}(W) = \frac{1}{4} \sum_{u,v \in V} w_{uv} - \frac{1}{4} \min \{ \langle W, Y \rangle : Y \geq 0, \langle E_{uu}, Y \rangle = 1, u \in V \},$$

where  $E_{uu}$  denotes the matrix which has a one at position  $(u, u)$  and zeros in all other positions.

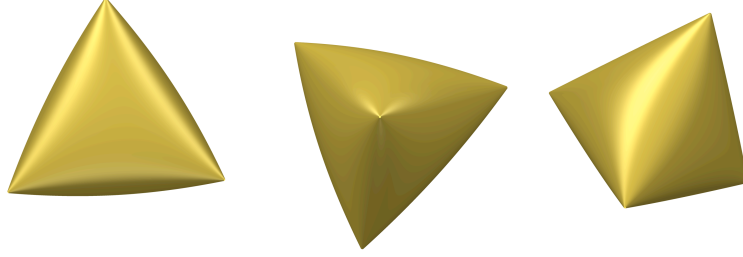


Figure 7.1: Views on spectrahedron behind the semidefinite relaxation.

The figure above illustrates the set of feasible solutions in the case of  $3 \times 3$  matrices. It is an inflated tetrahedron. These figures were generated by the program jSurfer (<http://www.imaginary-exhibition.com/>).

$$Y = \begin{pmatrix} 1 & x & y \\ x & 1 & z \\ y & z & 1 \end{pmatrix} \geq 0 \iff 1 + 2xyz - x^2 - y^2 - z^2 \geq 0, \quad x, y, z \in [-1, 1].$$

### 7.1.2 Analysis of the algorithm

**Theorem 7.1.1.** *For all matrices  $W = (w_{uv})$  with nonnegative weights we have the inequality*

$$\text{SDP}(W) \geq \text{MAXCUT}(W) \geq 0.878 \dots \cdot \text{SDP}(W).$$

*Proof.* The proof is algorithmic and it gives an approximation algorithm which approximates the MAX CUT problem within a ratio of  $0.878 \dots$ . The Goemans-Williamson algorithm has five steps:

1. Solve  $\text{SDP}(W)$  obtaining an optimal solution  $Y$ .
2. Perform a Cholesky decomposition of  $Y$  to find  $y_u \in S^{|V|-1}$ , with  $u \in V$ .
3. Choose a random vector  $r \in S^{|V|-1}$  according to the rotationally invariant probability distribution on the unit sphere.
4. Define a cut by  $x_u = \text{sign}(y_u \cdot r)$ , for  $u \in V$ .
5. Check whether  $\frac{1}{4} \sum_{u,v \in V} w_{uv}(1 - x_u x_v) \geq 0.878 \dots \cdot \text{SDP}(W)$ . If not, go to step 3.

The following lemma, also known as *Grothendieck's identity*, is the key to the proof of the theorem.

**Lemma 7.1.2.** *Let  $x, y$  be unit vectors and let  $r$  be a random unit vector chosen according to the rotationally invariant probability distribution on the unit sphere. Then, the expectation of the random variable  $\text{sign}(x \cdot r) \text{sign}(y \cdot r) \in \{-1, +1\}$  equals*

$$\mathbb{E}[\text{sign}(x \cdot r) \text{sign}(y \cdot r)] = \frac{2}{\pi} \arcsin x \cdot y.$$

*Proof.* By definition, the expectation can be computed as

$$\begin{aligned} \mathbb{E}[\text{sign}(x \cdot r) \text{sign}(y \cdot r)] &= (+1) \cdot \mathbb{P}[\text{sign}(x \cdot r) = \text{sign}(y \cdot r)] \\ &\quad + (-1) \cdot \mathbb{P}[\text{sign}(x \cdot r) \neq \text{sign}(y \cdot r)]. \end{aligned}$$

Note that

$$\mathbb{P}[\text{sign}(x \cdot r) \neq \text{sign}(y \cdot r)] = 1 - \mathbb{P}[\text{sign}(x \cdot r) = \text{sign}(y \cdot r)],$$

so that we only have to compute the probability that the signs of  $x \cdot r$  and  $y \cdot r$  are the same. Since the probability distribution from which we sample the unit vector  $r$  is rotationally invariant we can assume that  $x, y$  and  $r$  lie in a common plane and hence on a unit circle and that  $r$  is chosen according to the uniform distribution on this circle. Then the probability that the signs of  $x \cdot r$  and  $y \cdot r$  are the same only depends on the angle between  $x$  and  $y$ . Using a figure (draw one!) it is easy to see that

$$\mathbb{P}[\text{sign}(x \cdot r) = \text{sign}(y \cdot r)] = 2 \cdot \frac{1}{2\pi} \arccos x \cdot y = \frac{1}{\pi} \arccos x \cdot y.$$

Now,

$$\begin{aligned} \mathbb{E}[\text{sign}(x \cdot r) \text{sign}(y \cdot r)] &= \frac{1}{\pi} \arccos x \cdot y - \left(1 - \frac{1}{\pi} \arccos x \cdot y\right) \\ &= \frac{2}{\pi} \arcsin x \cdot y, \end{aligned}$$

where we used the trigonometric identity

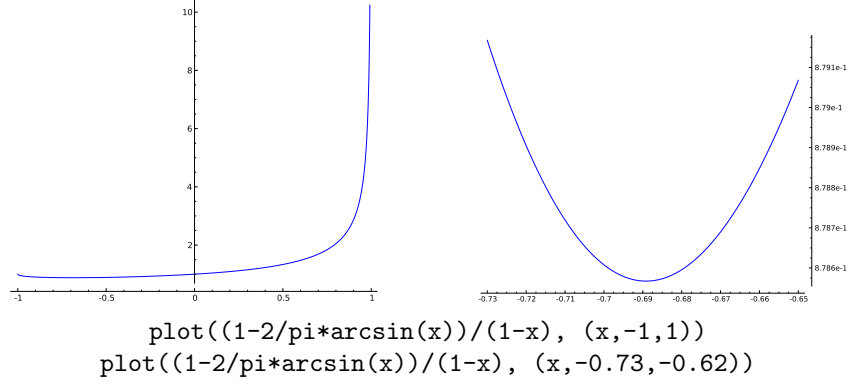
$$\arcsin t + \arccos t = \frac{\pi}{2},$$

to get the desired result.  $\square$

Let us apply Grothendieck's identity: Using elementary univariate calculus one can show that

$$\frac{1 - \mathbb{E}[\text{sign}(x \cdot r) \text{sign}(y \cdot r)]}{1 - x \cdot y} = \frac{1 - \frac{2}{\pi} \arcsin t}{1 - t} \geq 0.878 \dots, \quad (7.1)$$

holds, where  $t = x \cdot y \in [-1, 1]$ . To “see” this one can also plot the function using SAGE:



This can be used to estimate the ratio between  $\text{MAXCUT}(W)$  and  $\text{SDP}(W)$ . Clearly,

$$\text{MAXCUT}(W) \geq \mathbb{E} \left[ \frac{1}{4} \sum_{u,v \in V} w_{uv} (1 - x_u x_v) \right].$$

By linearity of expectation,

$$\mathbb{E} \left[ \frac{1}{4} \sum_{u,v \in V} w_{uv} (1 - x_u x_v) \right] = \frac{1}{4} \sum_{u,v \in V} w_{uv} (1 - \mathbb{E}[x_u x_v]).$$

Since  $w_{uv}$  is nonnegative we can go on by estimating

$$1 - \mathbb{E}[x_u x_v] = 1 - \mathbb{E}[\text{sign}(y_u \cdot r) \text{sign}(y_v \cdot r)]$$

in every summand using (7.1) getting

$$\mathbb{E}[(1 - x_u x_v)] \geq 0.878 \dots (1 - y_u \cdot y_v).$$

Putting it together,

$$\begin{aligned}
\text{MAXCUT}(W) &\geq \mathbb{E} \left[ \frac{1}{4} \sum_{u,v \in V} w_{uv} (1 - x_u x_v) \right] \\
&\geq 0.878 \dots \frac{1}{4} \sum_{u,v \in V} w_{uv} (1 - y_u \cdot y_v) \\
&= 0.878 \dots \cdot \text{SDP}(W),
\end{aligned}$$

which proves the theorem.  $\square$

We finish the explanation of the Goemans-Williamson algorithm by some further remarks. The steps 3. and 4. in the algorithm are called a *randomized rounding procedure* because a solution of a semidefinite program is “rounded” (or better: projected) to a solution of the original combinatorial problem with the help of randomness.

Note also that because the expectation of the constructed solution is at least  $0.878 \dots \text{SDP}(W)$  the algorithm eventually terminates; it will pass step 5 and without getting stuck in an endless loop. One can show that with high probability we do not have to wait long until the condition in step 5 is fulfilled.

### 7.1.3 Remarks on the algorithm

One can modify the Goemans-Williamson algorithm so that it becomes an algorithm which runs deterministically (without the use of randomness) in polynomial time and which gives the same approximation ratio. This was done by Mahajan and Ramesh in 1995.

It remains to give a procedure which samples a random vector from the unit sphere. This can be done if one can sample random numbers from the standard normal (Gaussian) distribution (with mean zero and variance one) which has probability density

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}.$$

Many software packages include a procedure which produces random numbers from the standard normal distribution. SAGE:

```
sage: T = RealDistribution('gaussian', 1)
sage: T.get_random_element()
0.818610064197
```

If we sample  $n$  real numbers  $x_1, \dots, x_n$  independently uniformly at random from the standard normal distribution, then, the vector

$$r = \frac{1}{\sqrt{x_1^2 + \dots + x_n^2}} (x_1, \dots, x_n)^T \in S^{n-1}$$

is distributed according to the rotationally invariant probability measure on the unit sphere.

## 7.2 Cut norm and Grothendieck's inequality

### 7.2.1 Cut norm of a matrix

The cut norm of a matrix  $A = (A_{ij}) \in \mathbb{R}^{m \times n}$  is

$$\|A\|_{\square} = \max_{I \subseteq [m], J \subseteq [n]} \left| \sum_{i \in I} \sum_{j \in J} A_{ij} \right|$$

Related to the cut norm is the following norm which is given as a quadratic optimization problem

$$\|A\|_{\infty \rightarrow 1} = \max \left\{ \sum_{i=1}^m \sum_{j=1}^n A_{ij} x_i y_j : x_i^2 = y_j^2 = 1, i \in [m], j \in [n] \right\}.$$



The notation  $\|A\|_{\infty \rightarrow 1}$  indicates that this is an operator norm of an operator mapping the space  $\ell_\infty^n$  to the space  $\ell_1^m$ . We will not use this fact here.

**Lemma 7.2.1.** *Let  $A \in \mathbb{R}^{m \times n}$  be a matrix.*

a) *We have the relation*

$$4\|A\|_{\square} \geq \|A\|_{\infty \rightarrow 1} \geq \|A\|_{\square}.$$

b) *If the row sum and the column sum of  $A$  are both 0, we have*

$$\|A\|_{\square} = \frac{1}{4}\|A\|_{\infty \rightarrow 1}.$$

c) *There exists a matrix  $B \in \mathbb{R}^{(m+1) \times (n+1)}$  with row sum and column sum equal to zero such that*

$$\|A\|_{\square} = \|B\|_{\square}.$$

*Proof.* a) For  $x_i, y_j \in \{\pm 1\}$  we split

$$\begin{aligned} \sum_{i=1}^m \sum_{j=1}^n A_{ij} x_i y_j &= \sum_{(i,j): x_i=1, y_j=1} A_{ij} + \sum_{(i,j): x_i=-1, y_j=-1} A_{ij} \\ &\quad - \sum_{(i,j): x_i=1, y_j=-1} A_{ij} - \sum_{(i,j): x_i=-1, y_j=1} A_{ij} \end{aligned}$$

and every term is bounded in absolute value from above by  $\|A\|_{\square}$ , hence the first inequality  $4\|A\|_{\square} \geq \|A\|_{\infty \rightarrow 1}$  follows.

For the other inequality let  $I \subseteq [m]$  and  $J \subseteq [n]$  be given so that

$$\|A\|_{\square} = \left| \sum_{i \in I, j \in J} A_{ij} \right|.$$

Define  $x_i = 1$  if  $i \in I$  and  $x_i = -1$  if  $i \notin I$  and similarly  $y_j = \pm 1$ . Then,

$$\begin{aligned} \|A\|_{\square} &= \sum_{i,j} A_{ij} \frac{1+x_i}{2} \frac{1+y_j}{2} \\ &= \frac{1}{4} \left( \sum_{i,j} A_{ij} + \sum_{i,j} A_{ij} x_i + \sum_{i,j} A_{ij} y_j + \sum_{i,j} A_{ij} x_i y_j \right). \end{aligned} \tag{7.2}$$

This proves the second inequality  $\|A\|_{\square} \leq \|A\|_{\infty \rightarrow 1}$  because the absolute value of every of the four summands is at most  $\|A\|_{\infty \rightarrow 1}$ .

b) The second statement of the lemma follows by looking at (7.2) and applying the additional assumption.

c) Simply construct  $B$  by adding a row and a column to  $A$  such that the row sum and the column sum are equal to 0. Checking that  $\|A\|_{\square}$  equals  $\|B\|_{\square}$  is an exercise.  $\square$

The following construction shows that computing the cut norm of a matrix is at least as difficult as computing the MAX CUT of a graph. Let  $G = (V, E)$  be a graph with  $n$  vertices  $v_1, \dots, v_n$  and  $m$  edges  $e_1, \dots, e_m$ , and let  $W = (w_{jk}) \in S^V$  be a weight matrix with nonnegative weights. Now we define a matrix  $A \in \mathbb{R}^{2m \times n}$  whose cut norm coincides with  $\text{MAXCUT}(W)$ . For this orient the edges of the graph in some arbitrary way. If  $e_i$  is arc from  $v_j$  to  $v_k$  then we set

$$A_{2i-1,j} = A_{2i,k} = w_{jk}, \quad A_{2i-1,k} = A_{2i,j} = -w_{jk}.$$

All other entries of  $A$  are zero.

**Lemma 7.2.2.** *Using the construction above we have*

$$\text{MAXCUT}(W) = \frac{1}{4} \cdot \|A\|_{\infty \rightarrow 1} = \|A\|_{\square}.$$

*Proof.* Exercise 7.3 (a). □

## 7.2.2 Grothendieck's inequality

The semidefinite relaxation of  $\|A\|_{\infty \rightarrow 1}$  is

$$\text{SDP}_{\infty \rightarrow 1}(A) = \max \left\{ \sum_{i=1}^m \sum_{j=1}^n A_{ij} u_i v_j : \|u_i\| = \|v_j\| = 1, i \in [m], j \in [n] \right\},$$

where we optimize over  $m + n$  unit vectors  $u_i, v_j \in \mathbb{R}^{m+n}$ . Note that this optimization problem is indeed a semidefinite program (why?).

**Theorem 7.2.3** (Grothendieck's inequality). *There is a constant  $K$  so that for all matrices  $A \in \mathbb{R}^{m \times n}$  the inequality*

$$\|A\|_{\infty \rightarrow 1} \leq \text{SDP}_{\infty \rightarrow 1}(A) \leq K \|A\|_{\infty \rightarrow 1}$$

*holds.*

The smallest constant  $K$  for which the second inequality holds, is called the *Grothendieck constant*  $K_G$ . It is known to lie between  $1.676 \dots$  and  $1.782 \dots$  but its exact value is currently not known. In the following we will prove that

$$K_G \leq \frac{\pi}{2 \ln(1 + \sqrt{2})} = 1.782 \dots$$

The argument will also rely on an approximation algorithm which uses randomized rounding (in a tricky way).

Thereby, and using Lemma 7.2.1, we find an algorithm which approximates the cut norm within a factor of  $(1.782 \dots)^{-1} = 0.561 \dots$

### 7.2.3 Proof of Grothendieck's inequality

1. Solve  $\text{SDP}_{\infty \rightarrow 1}(A)$ . This gives optimal unit vectors

$$u_1, \dots, u_m, v_1, \dots, v_n \in S^{m+n-1}.$$

2. Use these unit vectors to construct new unit vectors

$$u'_1, \dots, u'_m, v'_1, \dots, v'_n \in S^{m+n-1}$$

according to Krivine's trick presented in Lemma 7.2.4 below.

3. Choose a random vector  $r \in S^{m+n-1}$  according to the rotationally invariant probability distribution on the unit sphere.
4. Randomized rounding: Set

$$x_i = \text{sign}(u'_i \cdot r), \quad y_j = \text{sign}(v'_j \cdot r).$$

We analyze the expected quality of the solution  $x_i, y_j$ . By linearity of expectation we have

$$\text{SDP}_{\infty \rightarrow 1}(A) \geq \mathbb{E} \left[ \sum_{i=1}^m \sum_{j=1}^m A_{ij} x_i \cdot y_j \right] = \sum_{i=1}^m \sum_{j=1}^m A_{ij} \mathbb{E} [\text{sign}(u'_i \cdot r) \text{sign}(v'_j \cdot r)].$$

Now by Lemma 7.2.4 the last expectation will turn out to be  $\beta u_i \cdot v_j$ . Then the total sum will be equal  $\beta \text{SDP}_{\infty \rightarrow 1}(A)$  and hence  $K_G \leq \beta^{-1}$ .

Now the following lemma, Krivine's trick, finishes the proof of Theorem 7.2.3.

**Lemma 7.2.4.** *Let  $u_1, \dots, u_m$  and  $v_1, \dots, v_n$  be unit vectors in  $\mathbb{R}^{m+n}$ . Then there exist unit vectors  $u'_1, \dots, u'_m$  and  $v'_1, \dots, v'_n$  in  $\mathbb{R}^{m+n}$  such that*

$$\mathbb{E} [\text{sign}(u'_i \cdot r) \text{sign}(v'_j \cdot r)] = \beta u_i \cdot v_j,$$

holds with

$$\beta = \frac{2}{\pi} \ln(1 + \sqrt{2}) = 0.561 \dots$$

*Proof.* Define the function  $E : [-1, +1] \rightarrow [-1, +1]$  by  $E(t) = \frac{2}{\pi} \arcsin t$ . Then by Grothendieck's identity, Lemma 8.2.2,

$$\mathbb{E} [\text{sign}(u'_i \cdot r) \text{sign}(v'_j \cdot r)] = E(u'_i \cdot v'_j).$$

Now the idea is to invert the function  $E$  so that we have

$$u'_i \cdot v'_j = E^{-1}(\beta u_i \cdot v_j)$$

and use the series expansion

$$E^{-1}(t) = \sum_{r=0}^{\infty} g_{2r+1} t^{2r+1},$$

which is valid for all  $t \in [-1, 1]$  to define  $u'_i$  and  $v'_j$ .

For this define the infinite dimensional Hilbert space

$$H = \bigoplus_{r=0}^{\infty} (\mathbb{R}^{m+n})^{\otimes 2r+1},$$

and the vectors  $u'_i, v'_j \in H$  componentwise by

$$(u'_i)_r = \text{sign}(g_{2r+1}) \sqrt{|g_{2r+1}| \beta^{2r+1}} u_i^{\otimes 2r+1}$$

and

$$(v'_j)_r = \sqrt{|g_{2r+1}| \beta^{2r+1}} v_j^{\otimes 2r+1}.$$

Then

$$u'_i \cdot v'_j = \sum_{r=0}^{\infty} g_{2r+1} \beta^{2r+1} (u_i \cdot v_j)^{2r+1} = E^{-1}(\beta u_i \cdot v_j)$$

and

$$1 = u'_i \cdot u'_i = v'_j \cdot v'_j = \sum_{r=0}^{\infty} |g_{2r+1}| \beta^{2r+1},$$

which defines the value of  $\beta$  uniquely.

It's a fun exercise to work out  $\beta$  explicitly: We have

$$E(t) = \frac{2}{\pi} \arcsin t,$$

and so

$$E^{-1}(t) = \sin\left(\frac{\pi}{2}t\right) = \sum_{r=0}^{\infty} \frac{(-1)^{2r+1}}{(2r+1)!} \left(\frac{\pi}{2}t\right)^{2r+1}.$$

Hence,

$$1 = \sum_{r=0}^{\infty} \left| \frac{(-1)^{2r+1}}{(2r+1)!} \right| \left(\frac{\pi}{2}\beta\right)^{2r+1} = \sinh\left(\frac{\pi}{2}\beta\right),$$

which implies

$$\beta = \frac{2}{\pi} \text{arsinh } 1 = \frac{2}{\pi} \ln(1 + \sqrt{2})$$

because  $\text{arsinh } t = \ln(t + \sqrt{t^2 + 1})$ . □

Last concern: How to find/approximate  $u'_i, v'_j$  in polynomial time? Answer: we approximate the inner product matrix

$$(u'_i \cdot v'_j) = \sum_{r=0}^{\infty} g_{2r+1} \beta^{2r+1} ((u_i \cdot v_j)^{2r+1})$$

by its series expansion which converges fast enough and then we use its Cholesky decomposition.

## 7.3 Further reading

How good is the MAX CUT algorithm? Are there graphs where the value of the semidefinite relaxation and the value of the maximal cut are a factor of  $0.878\dots$  apart or is this value  $0.878\dots$ , which maybe looks strange at first sight, only an artefact of our analysis? It turned out that the value is optimal. In Exercise 7.1 (b) you will see that already for the 5-cycle  $C_5$  the gap is close to  $0.878\dots$ . In 2002 Feige and Schechtmann gave an infinite family of graphs for which the ratio  $\text{MAXCUT} / \text{SDP}$  converges to exactly  $0.878\dots$ . This proof uses a lot of nice mathematics (continuous graphs, Voronoi regions, isoperimetric inequality) and it is explained in detail in the Chapter 8 of the book *Approximation Algorithms and Semidefinite Programming* of Gärtner and Matoušek.

Rather recently in 2007, Khot, Kindler, Mossel, O'Donnell showed that the algorithm is optimal in the following sense: If the unique games conjecture is true, then there is no polynomial time approximation algorithm achieving a better approximation ratio than  $0.878\dots$  unless  $P = NP$ . Currently, the validity and the implications of the unique games conjecture are under heavy investigation. The topic of the unique games conjecture is too hot for this course, although it is very fascinating. The book of Gärtner and Matoušek also contains an introduction to the unique games conjecture.

How good is the upper bound  $K_G$ ? Finding the value of  $K_G$  is an long-standing open problem. The best-known lower bound is  $1.676\dots$  by unpublished work of Davie and Reeds.

It was widely believed that Krivine's analysis gives the right value of  $K_G$ . So it came as a shock (at least to the author of these notes) when Braverman, Makarychev, Makarychev, and Naor proved in 2011 that one can improve it slightly, by a clever modification of Krivine's trick and much more complicated computations.

Alon and Naor [2] discovered the connection between the cut norm and Grothendieck's inequality in 2006. Since then Grothendieck's inequality became a unifying concept in combinatorial optimization; see the survey [3] of Khot and Naor.

## 7.4 Historical remarks and anecdotes

In 2000, Goemans and Williamson won the Fulkerson prize (sponsored jointly by the Mathematical Programming Society and the AMS) which recognizes outstanding papers in the area of discrete mathematics for this result.

About the finding of the approximation ratio  $0.878\dots$  Knuth writes in the article "Mathematical Vanity Plates":

Sometimes people obtain mathematically significant license plates purely by accident, without making a personal selection. A striking example of this phenomenon is the case of Michel Goemans, who received the following innocuous-looking plate from the Massachusetts Registry of Motor Vehicles when he and his wife purchased a Subaru at the beginning of September 1993:



Two weeks later, Michel got together with his former student David Williamson, and they suddenly realized how to solve a problem that they had been working on for some years: to get good approximations for maximum cut and satisfiability problems by exploiting semidefinite programming. Lo and behold, their new method—which led to a famous, award-winning paper [15]—yielded the approximation factor .878! There it was, right on the license, with C, S, and W standing respectively for cut, satisfiability, and Williamson.

Winfried Scharlau tries to answer the question: Who is Alexander Grothendieck?

<http://www.ams.org/notices/200808/tx080800930p.pdf>

## 7.5 Questions

- 7.1\*\* (a) Find an approximation algorithm which approximates MAX CUT without using semidefinite programming: (Hint: What happens if one assigns  $x_u \in \{-1, +1\}$  uniformly at random with probability  $1/2$ ?)
- (b) Let  $W_n$  be the adjacency matrix of the  $n$ -cycle  $C_n$ . Find a closed formula for  $\text{SDP}(W_n)$ . How does this compare to  $\text{MAXCUT}(W_n)$ ?
- (c) Let  $A \in \mathcal{S}_{\geq 0}^n$  be a positive semidefinite matrix. Consider the quadratic optimization problem

$$\text{BQP}(A) = \max \left\{ \sum_{i=1}^n \sum_{j=1}^n A_{ij} x_i x_j : x_i^2 = 1, i \in [n] \right\}$$

and its semidefinite relaxation

$$\text{SDP}_{\text{BQP}}(A) = \max \left\{ \sum_{i=1}^n \sum_{j=1}^n A_{ij} u_i \cdot u_j : \|u_i\|^2 = 1, i \in [n] \right\}$$

Show that

$$\text{SDP}_{\text{BQP}}(A) \geq \text{BQP}(A) \geq \frac{\pi}{2} \text{SDP}_{\text{BQP}}(A)$$

Hint: Let  $X = (X_{ij}) \in \mathcal{S}_{\geq 0}^n$  be a positive semidefinite matrix. Then the matrix

$$\left(\frac{2}{\pi} \arcsin X_{ij}\right)_{1 \leq i, j \leq n} - \frac{2}{\pi} X$$

is positive semidefinite as well (recall: Taylor series of  $\arcsin$  and Hadamard product from Chapter 1).

7.2\*\* Define the Erdős-Rényi random graph model  $G(n, p)$  with  $p \in [0, 1]$  as follows: Choose a graph with  $n$  vertices uniformly at random by connecting two vertices with probability  $p$ . Define accordingly the random weight matrix  $W = (w_{uv}) \in \mathbb{R}^{n \times n}$  by setting  $w_{uv} = 1$  if  $\{u, v\} \in E$  and  $w_{uv} = 0$  if  $\{u, v\} \notin E$ .

Implement the Goemans-Williamson algorithm in SAGE. Use it to estimate numerically for  $n = 100$  and  $p = \frac{m}{10}$  with  $m = 1, 2, \dots, 9$  the average value of  $\text{SDP}(W)$  and the average value of the Goemans-Williamson algorithm.

7.3 (a) Prove Lemma 7.2.2.

(b) Prove

$$\|A\|_{\infty \rightarrow 1} = \max \left\{ \sum_{i=1}^m \sum_{j=1}^n A_{ij} x_i y_j : x_i, y_j \in [-1, 1], i \in [m], j \in [n] \right\}.$$

---

## BIBLIOGRAPHY

- [1] N. Alon, A. Naor, *Approximating the cut-norm via Grothendieck's inequality*, SIAM J. Comp. **35** (2006), 787–803.
- [2] M.X. Goemans, D.P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. ACM **42** (1995), 1115–1145.
- [3] S. Khot, A. Naor, *Grothendieck-type inequalities in combinatorial optimization*, arXiv:1108.2464 [cs.DS]  
<http://arxiv.org/abs/1108.2464>



---

# CHAPTER 8

---

## GENERALIZATIONS OF GROTHENDIECK'S INEQUALITY AND APPLICATIONS

In the second part of the last lecture we considered Grothendieck's inequality: There is a constant  $K$  so that for all matrices  $A \in \mathbb{R}^{m \times n}$  the inequality

$$\|A\|_{\infty \rightarrow 1} \leq \text{SDP}_{\infty \rightarrow 1}(A) \leq K \|A\|_{\infty \rightarrow 1}$$

holds, where:

$$\|A\|_{\infty \rightarrow 1} = \max \left\{ \sum_{i=1}^m \sum_{j=1}^n A_{ij} x_i y_j : x_i^2 = y_j^2 = 1, i \in [m], j \in [n] \right\}.$$

and where the semidefinite relaxation equals

$$\text{SDP}_{\infty \rightarrow 1}(A) = \max \left\{ \sum_{i=1}^m \sum_{j=1}^n A_{ij} u_i \cdot v_j : \|u_i\| = \|v_j\| = 1, i \in [m], j \in [n] \right\}.$$

We saw that  $\|A\|_{\infty \rightarrow 1}$  is closely related to the cut norm which is useful in many graph theoretic applications.

The number  $\|A\|_{\infty \rightarrow 1}$  also has a meaning in theoretical physics. It can be used to find ground states in the Ising model. The Ising model (named after the physicist Ernst Ising), is a mathematical model of ferromagnetism in statistical mechanics. The model consists of discrete variables called spins that can be in

one of two states, namely  $+1$  or  $-1$ , UP or DOWN. The spins are arranged in a graph, and each spin only interacts with its neighbors.

In many cases, the interaction graph is a finite subgraph of the integer lattice  $\mathbb{Z}^n$  where the vertices are the lattice points and where two vertices are connected if their Euclidean distance is one. These graphs are bipartite since they can be partitioned into even and odd vertices, corresponding to the parity of the sum of the coordinates. Let  $G = (V, E)$  be a bipartite interaction graph. The potential function is given by a symmetric matrix  $A = (A_{uv}) \in \mathcal{S}^V$ .  $A_{uv} = 0$  if  $u$  and  $v$  are not adjacent,  $A_{uv}$  is positive if there is ferromagnetic interaction between  $u$  and  $v$ , and  $A_{uv}$  is negative if there is antiferromagnetic interaction. The particles possess a spin  $x \in \{-1, +1\}^V$ . In the absence of an external field, the total energy of the system is given by

$$- \sum_{\{u,v\} \in E} A_{uv} x_u x_v.$$

The ground state of this model is a configuration of spins  $x \in \{-1, +1\}^V$  which minimizes the total energy. So computing the  $x_u \in \{-1, +1\}$  which give  $\|A\|_{\infty \rightarrow 1}$  is equivalent to finding the ground state and computing  $\text{SDP}_{\infty \rightarrow 1}$  amounts to approximate this ground state energy.

In this lecture we consider two generalizations of this bipartite Ising model.

We start by studying the Ising model for *arbitrary interaction graphs* and we find approximations of the ground state energy. The quality of this approximation will clearly depend on properties of the interaction graph. In particular, the theta number will appear here in an unexpected way.

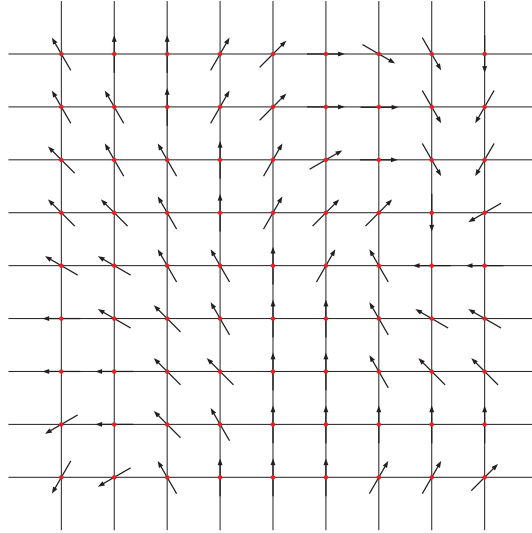


Figure 8.1: Spins in the XY model

Another generalization will be the consideration of *more complicated spins*. Instead of looking only at spins attaining the values  $-1$  and  $+1$  as in the Ising model, the  $r$ -vector model considers spins which are vectors in the unit sphere  $S^{r-1} = \{x \in \mathbb{R}^r : x \cdot x = 1\}$ . The case  $r = 1$  corresponds to the Ising model, the case  $r = 2$  to the XY model, the case  $r = 3$  to the Heisenberg model, and the case  $r = |V|$  to the Berlin-Kac spherical model. We will derive approximations of ground state energies

$$- \max \sum_{\{u,v\} \in E} A_{u,v} x_u \cdot x_v, \quad \text{for } x_u \in S^{r-1} \text{ and } u \in V$$

for fixed  $r$  and for bipartite graphs.

In principle a mixture of both generalizations is possible. We do not give it here as it would require adding even more technical details.

## 8.1 The Grothendieck constant of a graph

The *Grothendieck constant* of an undirected graph  $G = (V, E)$  is the smallest constant<sup>1</sup>  $K(G) = K$  so that for every symmetric matrix  $A \in S^V$  the inequality

$$\begin{aligned} \max \left\{ \sum_{\{u,v\} \in E} A_{uv} f_u \cdot f_v : f_u \in \mathbb{R}^V, u \in V, \|f_u\| = 1 \right\} \\ \leq K \max \left\{ \sum_{\{u,v\} \in E} A_{uv} x_u x_v : x_u = \pm 1, u \in V \right\} \end{aligned}$$

holds true. The left hand side is the semidefinite relaxation of the right hand side. Furthermore, the original Grothendieck constant which we studied in the last lecture is equal to the supremum of  $K(G)$  over all bipartite graphs  $G$ ; see Exercise 8.1 (a).

The following theorem gives a surprising connection between the Grothendieck constant of a graph and the theta number.

**Theorem 8.1.1.** *There is a constant  $C$  so that for any graph  $G$  we have*

$$K(G) \leq C \ln \vartheta(\overline{G}),$$

where  $\vartheta(\overline{G})$  is the theta number of the complementary graph of  $G$ .

The proof of this theorem will again be based on an approximation algorithm which performs randomized rounding of the solution of the semidefinite relaxation.

---

<sup>1</sup>Do not confuse the  $K(G)$  with  $K_G$  of the last lecture.

### 8.1.1 Randomized rounding by truncating

In the algorithm we use the constant  $M = 3\sqrt{1 + \ln \vartheta(\overline{G})}$ . The meaning of it will become clear when we analyze the algorithm.

1. Solve the semidefinite relaxation

$$\Gamma_{\max} = \max \left\{ \sum_{\{u,v\} \in E} A_{uv} f_u \cdot f_v : f_u \in \mathbb{R}^V, u \in V, \|f_u\| = 1 \right\}$$

2. Choose a random vector  $z = (z_u) \in \mathbb{R}^V$  so that every entry  $z_u$  is distributed independently according to the standard normal distribution with mean 0 and variance 1:  $z_u \sim N(0, 1)$ .
3. Round to real numbers  $y_u = z \cdot f_u$  for all  $u \in V$ .
4. Truncate  $y_u$  by setting

$$x_u = \begin{cases} y_u & \text{if } |y_u| \leq M, \\ M & \text{if } y_u > M, \\ -M & \text{if } y_u < -M. \end{cases}$$

We denote by  $\Delta$  the optimal value of the  $\pm 1$ -constrained problem

$$\Delta = \max \left\{ \sum_{\{u,v\} \in E} A_{uv} x_u x_v : x_u = \pm 1, u \in V \right\}.$$

**Important note.** The solution  $x_u$  which the algorithm determines does not satisfy the  $\pm 1$ -constraint. It only lies in the interval  $[-M, M]$  by construction. However, it is easy to show (how? see Exercise 7.3 (b)) that

$$M^2 \Delta = \max \left\{ \sum_{\{u,v\} \in E} A_{uv} x_u x_v : x_u \in [-M, M], u \in V \right\}$$

holds. Similarly,

$$\Gamma_{\max} = \max \left\{ \sum_{\{u,v\} \in E} A_{uv} f_u \cdot f_v : f_u \in \mathbb{R}^V, u \in V, \|f_u\| \leq 1 \right\}$$

In the remainder of this section we shall prove the theorem by giving an explicit bound on the ratio  $\Gamma_{\max}/\Delta$ .

### 8.1.2 Quality of expected solution

The expected quality of the solution  $x_u$  is

$$\begin{aligned}
& \sum_{\{u,v\} \in E} A_{uv} \mathbb{E}[x_u x_v] \\
&= \sum_{\{u,v\} \in E} A_{uv} (\mathbb{E}[y_u y_v] - \mathbb{E}[y_u(y_v - x_v)] \\
&\quad - \mathbb{E}[y_v(y_u - x_u)] + \mathbb{E}[(y_u - x_u)(y_v - x_v)]) \\
&= \Gamma_{\max} - \mathbb{E} \left[ \sum_{\{u,v\} \in E} A_{uv} ((y_u(y_v - x_v) + y_v(y_u - x_u))) \right] \\
&\quad + \mathbb{E} \left[ \sum_{\{u,v\} \in E} A_{uv} (y_u - x_u)(y_v - x_v) \right]
\end{aligned} \tag{8.1}$$

because  $\mathbb{E}[y_u y_v] = f_u \cdot f_v$  (Exercise 8.1 (b)).

### 8.1.3 A useful lemma

To estimate the second and third summand in (8.1) we use the following lemma.

**Lemma 8.1.2.** *Let  $X_u, Y_u$  be random variables with  $u \in V$ . Assume*

$$\mathbb{E}[X_u^2] \leq A \quad \text{and} \quad \mathbb{E}[Y_u^2] \leq B.$$

*Then,*

$$\mathbb{E} \left[ \sum_{\{u,v\} \in E} A_{uv} (X_u Y_v + X_v Y_u) \right] \leq 2\sqrt{AB}(\Gamma_{\max} - \Gamma_{\min}),$$

*where*

$$\Gamma_{\min} = \min \left\{ \sum_{\{u,v\} \in E} A_{uv} f_u \cdot f_v : f_u \in \mathbb{R}^V, u \in V, \|f_u\| \leq 1 \right\}.$$

*Proof.* If  $\mathbb{E}[X_u^2] \leq 1$ , then

$$\mathbb{E} \left[ \sum_{\{u,v\} \in E} A_{uv} X_u X_v \right] \in [\Gamma_{\min}, \Gamma_{\max}]. \tag{8.2}$$

This follows from the fact we can write

$$(\mathbb{E}[X_u X_v])_{u,v \in V} = (f_u \cdot f_v)_{u,v \in V}$$

because the matrix on the left hand side is positive semidefinite (Exercise 8.1 (c)) and thus has a Cholesky factorization.

We introduce new variables  $U_u$  and  $V_u$  to be able to apply (8.2). The new variables are

$$U_u = \frac{1}{2} \left( X_u/\sqrt{A} + Y_u/\sqrt{B} \right), \quad V_u = \frac{1}{2} \left( X_u/\sqrt{A} - Y_u/\sqrt{B} \right).$$

Then  $\mathbb{E}[U_u^2] \leq 1$  and  $\mathbb{E}[V_u^2] \leq 1$  (verify it). So we can apply (8.2)

$$\begin{aligned} & \mathbb{E} \left[ \sum_{\{u,v\} \in E} A_{uv} (X_u Y_v + X_v Y_u) \right] \\ &= 2\sqrt{AB} \left( \mathbb{E} \left[ \sum_{\{u,v\} \in E} A_{uv} U_u U_v \right] - \mathbb{E} \left[ \sum_{\{u,v\} \in E} A_{uv} V_u V_v \right] \right) \\ &\leq 2\sqrt{AB} (\Gamma_{\max} - \Gamma_{\min}). \end{aligned} \quad \square$$

#### 8.1.4 Estimating $A$ and $B$ in the useful lemma

It is clear, that  $\mathbb{E}[y_u^2] = 1$ . We find an upper bound for  $\mathbb{E}[(y_u - x_u)^2]$  in the following lemma.

**Lemma 8.1.3.**

$$\mathbb{E}[(y_u - x_u)^2] = 2 \frac{1}{\sqrt{2\pi}} \int_M^\infty t^2 e^{-t^2/2} dt \leq M e^{-M^2/2}.$$

*Proof.* The equation follows from the definition of  $y_u$  and  $x_u$  and the normal distribution. The inequality is coming from the following simple but noteworthy trick to estimate the integrand by an expression which can be integrated:

$$t^2 e^{-t^2/2} \leq (t^2 + t^{-2}) e^{-t^2/2}.$$

Then,

$$\int (t^2 + t^{-2}) e^{-t^2/2} dt = -\frac{t^2 + 1}{t} e^{-t^2/2} + \text{constant of integration},$$

and the lemma follows by

$$2 \frac{1}{\sqrt{2\pi}} \int_M^\infty t^2 e^{-t^2/2} dt \leq \sqrt{\frac{2}{\pi}} (M + 1/M) e^{-M^2/2} \leq M e^{-M^2/2}$$

because the fact  $M \geq 2$  implies that

$$\sqrt{\frac{2}{\pi}} (M + 1/M) \leq \frac{4}{5} (M + 1/M) \leq M. \quad \square$$

### 8.1.5 Applying the useful lemma

In (8.1) we estimate the second summand by applying the useful lemma with  $X_u = y_u$ ,  $Y_u = y_u - x_u$ . We get

$$-\mathbb{E} \left[ \sum_{\{u,v\} \in E} A_{uv} ((y_u(y_v - x_v) + y_v(y_u - x_u))) \right] \geq -2\sqrt{Me^{-M^2/2}}(\Gamma_{\max} - \Gamma_{\min}).$$

The third summand in (8.1) we estimate by applying the useful lemma with  $X_u = y_u - x_u$ ,  $Y_u = -(y_u - x_u)$ . We get

$$\mathbb{E} \left[ \sum_{\{u,v\} \in E} A_{uv} (y_u - x_u)(y_v - x_v) \right] \geq -Me^{-M^2/2}(\Gamma_{\max} - \Gamma_{\min}).$$

Altogether,

$$\sum_{\{u,v\} \in E} A_{uv} \mathbb{E}[x_u x_v] \geq \Gamma_{\max} - \left( 2\sqrt{Me^{-M^2/2}} + Me^{-M^2/2} \right) (\Gamma_{\max} - \Gamma_{\min}).$$

### 8.1.6 Connection to the theta number

The connection to the theta number comes in the following lemma.

**Lemma 8.1.4.**

$$\frac{\Gamma_{\max} - \Gamma_{\min}}{\Gamma_{\max}} \leq \vartheta(\overline{G}).$$

*Proof.* Exercise 8.2. □

In particular, we have

$$M \geq \tilde{M} = 3\sqrt{1 + \ln((\Gamma_{\max} - \Gamma_{\min})/\Gamma_{\max})}.$$

Furthermore ( $\ln x \leq x - 1$ ),

$$\tilde{M} \leq 3\sqrt{(\Gamma_{\max} - \Gamma_{\min})/\Gamma_{\max}}$$

From this it follows that

$$Me^{-M^2/2} \leq \tilde{M}e^{-\tilde{M}^2/2} \leq \frac{1}{10} \left( \frac{\Gamma_{\max}}{\Gamma_{\max} - \Gamma_{\min}} \right)^2.$$

So,

$$\sum_{\{u,v\} \in E} A_{uv} \mathbb{E}[x_u x_v] \geq \Gamma_{\max} - \frac{2}{\sqrt{10}} \Gamma_{\max} - \frac{1}{10} \frac{\Gamma_{\max}}{\Gamma_{\max} - \Gamma_{\min}} \Gamma_{\max},$$

since  $\Gamma_{\max} - \Gamma_{\min} \geq \Gamma_{\max}$  this leads to

$$\sum_{\{u,v\} \in E} A_{uv} \mathbb{E}[x_u x_v] \geq \frac{1}{4} \Gamma_{\max}.$$

Finally we can put everything together: There is a positive constant  $C$  (which is not difficult to estimate) so that

$$\Delta \geq \frac{1}{M^2} \sum_{\{u,v\} \in E} A_{uv} \mathbb{E}[x_u x_v] \geq \frac{1}{C \ln \vartheta(\overline{G})} \Gamma_{\max},$$

which finishes the proof of Theorem 8.1.1.

## 8.2 Higher rank Grothendieck inequality

Now we model finding ground states in the  $r$ -vector model. Given positive integers  $m, n, r$  and a matrix  $A = (A_{ij}) \in \mathbb{R}^{m \times n}$ , the *Grothendieck problem with rank- $r$ -constraint* is defined as

$$\text{SDP}_r(A) = \max \left\{ \sum_{i=1}^m \sum_{j=1}^n A_{ij} x_i \cdot y_j : x_1, \dots, x_m \in S^{r-1}, y_1, \dots, y_n \in S^{r-1} \right\},$$

where  $S^{r-1} = \{x \in \mathbb{R}^r : x \cdot x = 1\}$  is the unit sphere; the inner product matrix of the vectors  $x_1, \dots, x_m, y_1, \dots, y_n$  has rank  $r$ . When  $r = 1$ , then  $\text{SDP}_1(A) = \|A\|_{\infty \rightarrow 1}$  because  $S^0 = \{-1, +1\}$ .

When  $r$  is a constant that does not depend on the matrix size  $m, n$  there is no polynomial-time algorithm known which solves  $\text{SDP}_r$ . However, it is not known if the problem  $\text{SDP}_r$  is NP-hard when  $r \geq 2$ . On the other hand the *semidefinite relaxation* of  $\text{SDP}_r(A)$  defined by

$$\text{SDP}_{m+n}(A) = \max \left\{ \sum_{i=1}^m \sum_{j=1}^n A_{ij} u_i \cdot v_j : u_1, \dots, u_m, v_1, \dots, v_n \in S^{m+n-1} \right\}$$

can be computed in polynomial time using semidefinite programming.

**Theorem 8.2.1.** *For all matrices  $A \in \mathbb{R}^{m \times n}$  we have*

$$\text{SDP}_r(A) \leq \text{SDP}_{m+n}(A) \leq \frac{1}{2\gamma(r) - 1} \text{SDP}_r(A),$$

where

$$\gamma(r) = \frac{2}{r} \left( \frac{\Gamma((r+1)/2)}{\Gamma(r/2)} \right)^2,$$

and where  $\Gamma$  is the usual Gamma function, which is the extension of the factorial function.

The first three values of  $\frac{1}{2\gamma(r)-1}$  are:

$$\begin{aligned} \frac{1}{2\gamma(1)-1} &= \frac{1}{4/\pi - 1} = 3.65979\dots, \\ \frac{1}{2\gamma(2)-1} &= \frac{1}{\pi/2 - 1} = 1.75193\dots, \\ \frac{1}{2\gamma(3)-1} &= \frac{1}{16/(3\pi) - 1} = 1.43337\dots \end{aligned}$$



For  $r \rightarrow \infty$  the values  $\frac{1}{2\gamma(r)-1}$  converge to 1. In particular, the proof of the theorem gives another proof of the original Grothendieck's inequality albeit with a worse constant  $K_G \leq \frac{1}{4/\pi-1}$ .

### 8.2.1 Randomized rounding by projecting

The approximation algorithm which we use to prove the theorem is the following three-step process.

1. By solving  $\text{SDP}_{m+n}(A)$  we obtain the vectors  $u_1, \dots, u_m, v_1, \dots, v_n \in S^{m+n-1}$ .
2. Choose  $Z = (Z_{ij}) \in \mathbb{R}^{r \times (m+n)}$  so that every matrix entry  $Z_{ij}$  is distributed independently according to the standard normal distribution with mean 0 and variance 1:  $Z_{ij} \sim N(0, 1)$ .
3. Project  $x_i = Zu_i / \|Zu_i\| \in S^{r-1}$  with  $i = 1, \dots, m$ , and  $y_j = Zv_j / \|Zv_j\| \in S^{r-1}$  with  $j = 1, \dots, n$ .

### 8.2.2 Extension of Grothendieck's identity

The quality of the feasible solution  $x_1, \dots, x_m, y_1, \dots, y_n$  for  $\text{SDP}_r$  is measured by the expectation

$$\text{SDP}_r(A) \geq \mathbb{E} \left[ \sum_{i=1}^m \sum_{j=1}^n A_{ij} x_i \cdot y_j \right].$$

**Lemma 8.2.2.** *Let  $u, v$  be unit vectors in  $\mathbb{R}^{m+n}$  and let  $Z \in \mathbb{R}^{r \times (m+n)}$  be a random matrix whose entries are distributed independently according to the standard normal distribution with mean 0 and variance 1. Then,*

$$\begin{aligned} & \mathbb{E} \left[ \frac{Zu}{\|Zu\|} \cdot \frac{Zv}{\|Zv\|} \right] \\ &= \frac{2}{r} \left( \frac{\Gamma((r+1)/2)}{\Gamma(r/2)} \right)^2 \sum_{k=0}^{\infty} \frac{(1 \cdot 3 \cdots (2k-1))^2}{(2 \cdot 4 \cdots 2k)((r+2) \cdot (r+4) \cdots (r+2k))} (u \cdot v)^{2k+1}. \end{aligned}$$

The case  $r = 1$  specializes to Grothendieck's identity from the previous chapter:

$$\begin{aligned} \mathbb{E}[\text{sign}(Zu)\text{sign}(Zv)] &= \frac{2}{\pi} \arcsin(u \cdot v) \\ &= \frac{2}{\pi} \left( u \cdot v + \left( \frac{1}{2} \right) \frac{(u \cdot v)^3}{3} + \left( \frac{1 \cdot 3}{2 \cdot 4} \right) \frac{(u \cdot v)^5}{5} + \cdots \right). \end{aligned}$$

The proof of Lemma 8.2.2 requires quite some integration. The computation starts of by

$$\begin{aligned} & \mathbb{E} \left[ \frac{Zu}{\|Zu\|} \cdot \frac{Zv}{\|Zv\|} \right] \\ &= (2\pi\sqrt{1-t^2})^{-r} \int_{\mathbb{R}^r} \int_{\mathbb{R}^r} \frac{x}{\|x\|} \cdot \frac{y}{\|y\|} \exp \left( -\frac{x \cdot x - 2tx \cdot y + y \cdot y}{2(1-t^2)} \right) dx dy, \end{aligned}$$

where  $t = u \cdot v$ . We will omit the tedious calculation here. For those who cannot resist a definite integral (like G.H. Hardy): it can be found in [4].

The only three facts which will be important is that the power series expansion

$$\mathbb{E} \left[ \frac{Zu}{\|Zu\|} \cdot \frac{Zv}{\|Zv\|} \right] = \sum_{k=0}^{\infty} f_{2k+1}(u \cdot v)^{2k+1}$$

has the following three properties:

1. the leading coefficient  $f_1$  equals  $\gamma(r)$
2. all coefficients  $f_{2k+1}$  are nonnegative
3.  $\sum_{k=0}^{\infty} f_{2k+1} = 1$ .

### 8.2.3 Proof of the theorem

Now we have

$$\begin{aligned} \mathbb{E} \left[ \sum_{i=1}^m \sum_{j=1}^n A_{ij} x_i \cdot y_j \right] &= \sum_{i=1}^m \sum_{j=1}^n A_{ij} \mathbb{E} \left[ \frac{Zu_i}{\|Zu_i\|} \cdot \frac{Zv_j}{\|Zv_j\|} \right] \\ &= f_1 \sum_{i=1}^m \sum_{j=1}^n A_{ij} u_i \cdot v_j + \sum_{i=1}^m \sum_{j=1}^n A_{ij} \sum_{k=1}^{\infty} f_{2k+1}(u_i \cdot v_j)^{2k+1}. \end{aligned}$$

The first summand equals  $f_1 \text{SDP}_{m+n}(A)$ . The second summand is bounded in absolute value by  $(1 - f_1) \text{SDP}_{m+n}(A)$  as you will prove in Exercise 8.1 (d).

Thus for the second sum we have

$$\sum_{i=1}^m \sum_{j=1}^n A_{ij} \sum_{k=1}^{\infty} f_{2k+1}(u_i \cdot v_j)^{2k+1} \geq (f_1 - 1) \text{SDP}_{m+n}(A),$$

which finishes the proof.

## 8.3 Further reading

Section 8.1: The result is from [1] and the presentation of the proof is closely following Chapter 10 of the book *Approximation Algorithms and Semidefinite Programming* of Gärtner and Matoušek, which mostly follows K. Makarychev's thesis.

Section 8.2: The proof is from [3] and it follows the idea of Alon and Naor [2, Section 4] which in turn relies on ideas of Rietz.

More on the definite integral: When working with power series expansions it is sometimes useful to use hypergeometric functions for this. For instance we

have

$$\sum_{k=0}^{\infty} \frac{(1 \cdot 3 \cdots (2k-1))^2}{(2 \cdot 4 \cdots 2k)((r+2) \cdot (r+4) \cdots (r+2k))} (u \cdot v)^{2k+1} \\ = (u \cdot v) {}_2F_1 \left( \begin{matrix} 1/2, 1/2 \\ r/2 + 1 \end{matrix}; (u \cdot v)^2 \right),$$

where  ${}_2F_1$  is a hypergeometric function. Hypergeometric functions are a classical subject in mathematics. In fact, many (all?) functions you know, are hypergeometric functions. However the topic of hypergeometric functions seems somehow to be too classical for many modern universities.

In case you want to know more about them: The book "A=B" by Petkovsek, Wilf and Zeilberger

<http://www.math.upenn.edu/~wilf/AeqB.html>

is a good start.

## 8.4 Exercises

8.1\*\* (a) Why does Theorem 8.1.1 give a proof of the original Grothendieck inequality? Which explicit upper bound for  $K_G$  does it provide? (Determine a concrete number.)

(b) Show that  $\mathbb{E}[y_u y_v] = f_u \cdot f_v$  holds.

(c) Prove that the matrix

$$(\mathbb{E}[X_u X_v])_{u,v \in V}$$

is positive semidefinite.

(d) Show that

$$\left| \sum_{i=1}^m \sum_{j=1}^n A_{ij} \sum_{k=1}^{\infty} f_{2k+1}(u_i \cdot v_j)^{2k+1} \right| \leq (1 - f_1) \text{SDP}_{m+n}(A).$$

8.2 Let  $G = (V, E)$  be a graph. A *vector  $k$ -coloring* of  $G$  is a collection of unit vectors  $f_u \in \mathbb{R}^V$  so that

$$f_u \cdot f_v = -\frac{1}{k-1} \quad \text{if } \{u, v\} \in E.$$

(a) Show that if  $G$  is colorable with  $k$  colors, then it also has a vector  $k$ -coloring.

(b) Find a connection between vector  $k$ -colorings and the theta number.

(c) Prove Lemma 8.1.4.

---

## BIBLIOGRAPHY

- [1] N. Alon, K. Makarychev, Y. Makarychev, A. Naor, *Quadratic forms on graphs*, Invent. Math. **163** (2006), 499–522.
- [2] N. Alon, A. Naor, *Approximating the cut-norm via Grothendieck’s inequality*, SIAM J. Comp. **35** (2006), 787–803.
- [3] J. Briët, F.M. de Oliveira Filho, F. Vallentin, *The Grothendieck problem with rank constraint*, pp. 111–113 in Proceedings of the 19th Symposium on Mathematical Theory of Networks and Systems, 2010
- [4] J. Briët, F.M. de Oliveira Filho, F. Vallentin, *Grothendieck inequalities for semidefinite programs with rank constraint*, arXiv:1011.1754v1 [math.OC]  
<http://arxiv.org/abs/1011.1754>

---

---

## CHAPTER 9

---

# OPTIMIZING WITH ELLIPSOIDS AND DETERMINANTS

### What is an ellipsoid?

There are two convenient ways to represent an ellipsoid.

1. We can define ellipsoids explicitly as the image of the unit ball under an invertible affine transformation

$$\mathcal{E}_{A,c} = \{Ax + c : x \in B\}, \quad \text{where } B = \{x \in \mathbb{R}^n : \|x\| \leq 1\},$$

is the unit ball, where  $A \in \mathbb{R}^{n \times n}$  is an invertible matrix, and where  $c \in \mathbb{R}^n$  is a translation vector.

From linear algebra it is known that every invertible matrix  $A$  has a factorization of the form  $A = XP$  where  $X \in \mathcal{S}_{>0}^n$  is a positive definite matrix and  $P \in \mathcal{O}(n)$  is an orthogonal matrix. So we may assume in the following that the matrix  $A$  which defines the ellipsoid  $\mathcal{E}_{A,c}$  is a positive definite matrix.

In fact one can find this factorization (also called polar factorization) from the singular value decomposition of  $A$

$$A = U^T \Sigma V, \quad U, V \in \mathcal{O}(n), \quad \Sigma = \text{diag}(\sigma_1, \dots, \sigma_n),$$

where  $\sigma_i \geq 0$  are the singular values of  $A$ . Then,

$$A = XP \quad \text{with} \quad X = U^T \Sigma U, \quad P = U^T V.$$

The singular values of  $A$  are at the same time the lengths of the semiaxis of the ellipsoid  $\mathcal{E}_{A,c}$ .

The volume of the ellipsoid equals

$$\text{vol } \mathcal{E}_{A,c} = \det A \text{ vol } B \quad \text{where} \quad \text{vol } B = \frac{\pi^{n/2}}{\Gamma(n/2 + 1)}.$$

2. We can define ellipsoids implicitly by a strictly convex quadratic inequality: Let  $A \in \mathcal{S}_{>0}^n$  be a positive definite matrix, then

$$\begin{aligned} \mathcal{E}_{A,c} &= \{Ax + c : \|x\| \leq 1\} \\ &= \{x + c : \|A^{-1}x\| \leq 1\} \\ &= \{x \in \mathbb{R}^n : (x - c)^\top A^{-1} (x - c) \leq 1\}. \end{aligned}$$

Ellipsoids are important geometric objects partially due to their simple descriptions. They can be used for instance to approximate other more complicated convex sets. In this lecture we will use ellipsoids to approximate polytopes. In particular we will answer the questions:

- Inner approximation: How can we determine an ellipsoid contained in a polytope which has largest volume?
- Outer approximation: How can we determine an ellipsoid containing a polytope which has smallest volume?
- Can we estimate the quality of this inner and of this outer approximation?

## 9.1 Determinant maximization problems

To be able to maximize the volume of ellipsoids we want to maximize the determinant of positive definite matrices. In the next section we will see that the logarithm of the determinant is a concave function so that determinant maximization can be dealt with tools from convex optimization.

In fact one can reformulate determinant maximization problems as semidefinite programs but we will not do this here; dealing directly with determinant maximization problem is generally easier and more efficient.

Here we give the primal-dual pair of a determinant maximization problem together with the corresponding duality theory. All in all it is very similar to semidefinite programming, only the objective function is not linear.

The *primal determinant maximization problem* is defined as

$$\begin{aligned} \sup \quad & n + \langle C, X \rangle + \ln \det X \\ & X \in \mathcal{S}_{>0}^n \\ & \langle A_j, X \rangle = b_j, \quad j = 1, \dots, m, \end{aligned}$$

and its *dual* is

$$\begin{aligned} \inf \quad & b^\top y - \ln \det \left( \sum_{j=1}^m y_j A_j - C \right) \\ & y \in \mathbb{R}^m \\ & \sum_{j=1}^m y_j A_j - C \in \mathcal{S}_{>0}^n \end{aligned}$$

We have statements for weak duality and strong duality, which are very similar to Theorem 3.4.1. A slight difference is complementary slackness and the optimality criterion. The optimality criterion says: Suppose that  $X$  is feasible for the primal,  $y$  is feasible for the dual and the equality

$$X \left( \sum_{j=1}^m y_j A_j - C \right) = I_n$$

holds. Then  $X$  and  $y$  are both optimal (see Exercise 9.1 (a)).

In many cases it is useful to combine determinant maximization problems with linear conic programs. In these cases we want to work with the following primal-dual pair.

Primal:

$$\begin{aligned} \sup \quad & n + \langle C, X \rangle + \ln \det X + c^\top x \\ & X \in \mathcal{S}_{>0}^n \\ & x \in K \\ & \langle A_j, X \rangle + a_j^\top x = b_j, \quad j = 1, \dots, m \end{aligned}$$

Dual:

$$\begin{aligned} \inf \quad & b^\top y - \ln \det \left( \sum_{j=1}^m y_j A_j - C \right) \\ & y \in \mathbb{R}^m \\ & \sum_{j=1}^m y_j A_j - C \in \mathcal{S}_{>0}^n \\ & \sum_{j=1}^m y_j a_j - c \in K^* \end{aligned}$$

## 9.2 Convex spectral functions

In this section we shall prove that the function  $X \mapsto -\ln \det X$  is a (strictly) convex function. For this we will give two proofs. One simple adhoc proof and one which is conceptual. The second proof will characterize all convex functions on symmetric matrices which only depend on the eigenvalues.

### 9.2.1 Minkowski's determinant inequality

**Theorem 9.2.1.** *The function*

$$F : \mathcal{S}_{>0}^n \rightarrow \mathbb{R}, \quad X \mapsto -\ln \det X$$

*is a strictly convex function on the set of positive definite matrices.*

*Proof.* It suffices to show that the function  $X \mapsto -\ln \det X$  is strictly convex on any line segment

$$[X, Y] = \{tX + (1-t)Y : t \in [0, 1], X \neq Y\}$$

in  $\mathcal{S}_{>0}^n$ . Therefore, we compute the second derivative of the one-dimensional function  $f(t) = -\ln \det(tX + (1-t)Y)$  and see that it is always strictly positive: From linear algebra we know that there is a matrix  $T$  with determinant 1 whose inverse simultaneously diagonalizes  $X$  and  $Y$ . Hence,

$$X = T^\top \text{diag}(x_1, \dots, x_n)T \quad \text{and} \quad Y = T^\top \text{diag}(y_1, \dots, y_n)T$$

and

$$\begin{aligned} f(t) &= -\ln(y_1 + t(x_1 - y_1)) - \dots - \ln(y_n + t(x_n - y_n)), \\ \frac{\partial f}{\partial t}(t) &= -\frac{x_1 - y_1}{y_1 + t(x_1 - y_1)} - \dots - \frac{x_n - y_n}{y_n + t(x_n - y_n)}, \\ \frac{\partial^2 f}{\partial t^2}(t) &= \left( \frac{x_1 - y_1}{y_1 + t(x_1 - y_1)} \right)^2 + \dots + \left( \frac{x_n - y_n}{y_n + t(x_n - y_n)} \right)^2 > 0. \quad \square \end{aligned}$$

With the same argument we can derive *Minkowski's determinantal inequality*:

$$(\det(X + Y))^{1/n} \geq (\det X)^{1/n} + (\det Y)^{1/n}$$

which holds for all  $X, Y \in \mathcal{S}_{>0}^n$ .

Geometrically, this means that in the cone of positive semidefinite matrices, the set of matrices having determinant greater or equal than a given constant is a convex set.

### 9.2.2 Davis' characterization of convex spectral functions

The function  $F(X) = -\ln \det X$  is an example of a convex spectral function.

**Definition 9.2.2.** *A convex spectral function is a convex function*

$$F : \mathcal{S}^n \rightarrow \mathbb{R} \cup \{\infty\}$$

*where  $F(X)$  only depends on the spectrum (the collection of the eigenvalues  $\lambda_1(X), \dots, \lambda_n(X)$ ) of the matrix  $X$ . In other words, by the spectral theorem,*

$$F(X) = f(AXA^\top) \quad \text{for all } A \in \mathcal{O}(n).$$

*Hence, there is a function  $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$  which defines  $F$  by the following equation*

$$F(X) = f(\lambda_1(X), \dots, \lambda_n(X)).$$



Note that this implies that the function  $f$  is *symmetric*, i.e. its value stays the same if we permute its  $n$  arguments; it is invariant under permutation of the variables.

In our example

$$f(\lambda_1, \dots, \lambda_n) = \begin{cases} -\ln \prod_{i=1}^n \lambda_i & \text{if all } \lambda_i > 0, \\ \infty & \text{otherwise.} \end{cases}$$

The following theorem is due to Davis (1957). It gives a complete characterization of convex spectral functions.

**Theorem 9.2.3.** *A function  $F : \mathcal{S}^n \rightarrow \mathbb{R} \cup \{\infty\}$  is a convex spectral function if and only if the function  $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$  defined by*

$$F(X) = f(\lambda_1(X), \dots, \lambda_n(X))$$

*is symmetric and convex. In particular,*

$$F(X) = \max_{A \in \mathcal{O}(n)} f((AXA^\top)_{11}, \dots, (AXA^\top)_{nn})$$

*holds.*

*Proof.* One implication follows without any work.

Let  $F$  be a convex spectral function. Then  $f$  is symmetric by definition. It is convex since  $F$  and  $f$  “coincide” on diagonal matrices. Let  $\Lambda = (\lambda_1, \dots, \lambda_n)$ ,  $M = (\mu_1, \dots, \mu_n)$  and  $t \in [0, 1]$  be given. Then

$$\begin{aligned} f(t\Lambda + (1-t)M) &= F(\text{diag}(t\Lambda + (1-t)M)) \\ &= F(t \text{diag}(\Lambda) + (1-t) \text{diag}(M)) \\ &\leq tF(\text{diag}(\Lambda)) + (1-t)F(\text{diag}(M)) \\ &= tf(\Lambda) + (1-t)f(M). \end{aligned}$$

The proof of the other implication is more interesting. It is an application of Birkhoff’s theorem (cf. Chapter 1.7.3), see also the geometric interpretation at the end of this section.

If we show that

$$F(X) = \max_{A \in \mathcal{O}(n)} f((AXA^\top)_{11}, \dots, (AXA^\top)_{nn})$$

holds, then it follows that  $F$  is convex because it is a maximum of a family of convex functions.

For this consider the spectral decomposition of  $X$

$$X = \sum_{j=1}^n \lambda_j u_j u_j^\top.$$

with orthonormal basis  $u_1, \dots, u_n$ . If we assemble these vectors as row vectors in the orthogonal matrix  $A$  we see that

$$f((AXA^\top)_{11}, \dots, (AXA^\top)_{nn}) = f(\lambda_1, \dots, \lambda_n) = F(X)$$

holds. Thus,

$$F(X) \leq \max_{A \in \mathcal{O}(n)} f((AXA^\top)_{11}, \dots, (AXA^\top)_{nn}).$$

The other inequality. For  $A \in \mathcal{O}(n)$  define  $Y = AXA^\top$ . Then,

$$Y_{ii} = e_i^\top Y e_i = e_i^\top \left( \sum_{j=1}^n \lambda_j A u_j (A u_j)^\top \right) e_i = \sum_{j=1}^n \lambda_j ((A u_j)^\top e_i)^2.$$

Here is the trick: The matrix

$$S = (S_{ij})_{1 \leq i, j \leq n} \quad \text{with} \quad S_{ij} = ((A u_j)^\top e_i)^2$$

is doubly stochastic (why?). So by Birkhoff's theorem  $S$  is a convex combination of permutation matrices  $P^\sigma$ .

$$S = \sum_{\sigma \in S_n} \mu_\sigma P^\sigma, \quad \text{where} \quad \mu_\sigma \geq 0, \quad \sum_{\sigma \in S_n} \mu_\sigma = 1.$$

Hence by the convexity and the symmetry of  $f$ , we have

$$\begin{aligned} f(Y_{11}, \dots, Y_{nn}) &= f\left(\sum_{\sigma \in S_n} \mu_\sigma \sum_{j=1}^n P_{1j}^\sigma \lambda_j, \dots, \sum_{\sigma \in S_n} \mu_\sigma \sum_{j=1}^n P_{nj}^\sigma \lambda_j\right) \\ &\leq \sum_{\sigma \in S_n} \mu_\sigma f\left(\sum_{j=1}^n P_{1j}^\sigma \lambda_j, \dots, \sum_{j=1}^n P_{nj}^\sigma \lambda_j\right) \\ &= \sum_{\sigma \in S_n} \mu_\sigma f(\lambda_{\sigma^{-1}(1)}, \dots, \lambda_{\sigma^{-1}(n)}) \\ &= \sum_{\sigma \in S_n} \mu_\sigma f(\lambda_1, \dots, \lambda_n) \\ &= f(\lambda_1, \dots, \lambda_n) \\ &= F(X). \end{aligned}$$

Hence, for all  $A \in \mathcal{O}(n)$

$$f((AXA^\top)_{11}, \dots, (AXA^\top)_{nn}) \leq F(X),$$

and the theorem is proved.  $\square$

We conclude this excursion with a geometric observation which unifies some of our previous considerations. If we project matrices in the Schur-Horn orbitope of  $X$  which is defined by

$$\text{conv}\{AXA^\top : A \in \mathcal{O}(n)\}$$

on the diagonal elements, then we get the permutahedron given by the eigenvalues  $\lambda_1, \dots, \lambda_n$  of  $X$  which is defined as

$$\text{conv}\{(\lambda_{\sigma(1)}, \dots, \lambda_{\sigma(n)}) : \sigma \in S_n\}.$$

From this the Hoffman-Wielandt inequality and the characterization of Davis follow.

Another side remark: Davis' characterization together with Fan's theorem (Theorem 2.2.2) can be used to determine an explicit linear matrix inequality modeling the condition  $F(X) \leq t$  for many functions  $F$ . See Ben-Tal, Nemirovski [3][Proposition 4.2.1] for the complete statement. A similar argument also works for functions depending only on singular values.

### 9.3 Approximating polytopes by ellipsoids

Now we are ready to describe how ellipsoids can be used to approximate polytopes.

Recall that one can represent a polytope in two ways. Either as a convex hull of finitely many points

$$P = \text{conv}\{x_1, \dots, x_N\} \in \mathbb{R}^n,$$

or as a bounded intersection of finitely many halfspaces

$$P = \{x \in \mathbb{R}^n : a_1^\top x \leq b_1, \dots, a_m^\top x \leq b_m\}.$$

The first representation, also called the  $\mathcal{V}$ -representation, is an explicit parameterization whereas the second one, also called the  $\mathcal{H}$ -representation, is implicit. In general it is computationally demanding to transform one representation into the other.

#### 9.3.1 Inner approximation

To formulate the condition that an ellipsoid  $\mathcal{E}_{A,c}$  is contained in a polytope  $P$  we will use the explicit representation of the ellipsoid and the implicit representation of the polytope.

**Proposition 9.3.1.** *The ellipsoid*

$$\mathcal{E}_{A,c} = \{Ax + c : \|x\| \leq 1\}$$

*is contained in the polytope*

$$P = \{x \in \mathbb{R}^n : a_1^\top x \leq b_1, \dots, a_m^\top x \leq b_m\}$$

*if and only if the inequality*

$$\|Aa_i\| \leq b_i - a_i^\top c$$

*holds for all  $i = 1, \dots, m$ .*

*Proof.* We have

$$\begin{aligned}
& a_i^\top (Ax + c) \leq b_i \quad \forall x \in \mathbb{R}^n : \|x\| \leq 1 \\
\iff & \max_{x: \|x\| \leq 1} (Aa_i)^\top x \leq b_i - a_i^\top c \\
\iff & \|Aa_i\| \leq b_i - a_i^\top c,
\end{aligned}$$

because by the Cauchy-Schwarz inequality  $\max_{x: \|x\| \leq 1} (Aa_i)^\top x = \|Aa_i\|$ .  $\square$

The inequality  $\|Aa_i\| \leq b_i - a_i^\top c$  can be directly modeled by a second order cone programming constraint or, using the Schur complement, by a semidefinite constraint.

### 9.3.2 Outer approximation

To formulate the condition that an ellipsoid  $\mathcal{E}_{A,c}$  is containing a polytope  $P$  we will use the implicit representation of the ellipsoid and the explicit representation of the polytope.

**Proposition 9.3.2.** *The ellipsoid*

$$\mathcal{E}_{A^{-1/2},c} = \{x \in \mathbb{R}^n : (x - c)^\top A(x - c) \leq 1\}$$

*contains the polytope*

$$P = \text{conv}\{x_1, \dots, x_N\}$$

*if and only if the matrix*

$$\begin{pmatrix} s & d^\top \\ d & A \end{pmatrix}$$

*is positive semidefinite with  $Ac = d$  and the inequality*

$$x_i^\top Ax_i - 2x_i^\top d + s \leq 1$$

*holds for all  $i = 1, \dots, N$ .*

*Proof.* The point  $x_i$  lies in the ellipsoid  $\mathcal{E}_{A^{-1/2},c}$  if and only if

$$\begin{aligned}
& (x_i - c)^\top A(x_i - c) \leq 1 \\
\iff & x_i^\top Ax_i - 2x_i^\top Ac + c^\top Ac \leq 1 \\
\iff & x_i^\top Ax_i - 2x_i^\top d + d^\top A^{-1}d \leq 1 \\
\iff & x_i^\top Ax_i - 2x_i^\top d + s \leq 1, \quad \text{where } s \geq d^\top A^{-1}d.
\end{aligned}$$

Because the matrix  $A$  is positive definite we can express  $s \geq d^\top A^{-1}d$  using the Schur complement as

$$\begin{pmatrix} s & d^\top \\ d & A \end{pmatrix} \geq 0. \quad \square$$

The constraint

$$x_i^\top A x_i - 2x_i^\top d + s \leq 1$$

can be expressed by the linear matrix inequality

$$\left\langle \begin{pmatrix} 1 & x_i^\top \\ x_i & x_i x_i^\top \end{pmatrix}, \begin{pmatrix} s & d^\top \\ d & A \end{pmatrix} \right\rangle \leq 1.$$

## 9.4 The Löwner-John ellipsoids

Using Proposition 9.3.1 and Proposition 9.3.2 one can find the ellipsoid of largest volume contained in a polytope  $P$  as well as the ellipsoid of smallest volume containing  $P$  by solving determinant maximization problems. In both cases one maximizes the logarithm of the determinant of  $A$ . Because the logarithm of the determinant is a *strictly* concave function both optimization problems have a unique solution. The ellipsoids are called the *Löwner-John ellipsoids*<sup>1</sup> of  $P$ . Notation:  $\mathcal{E}_{in}(P)$  for the ellipsoid giving the optimal inner approximation of  $P$  and  $\mathcal{E}_{out}(P)$  for the ellipsoid giving the optimal outer approximation of  $P$ .

The following theorem can be traced back to John (1948). Historically, it is considered to be one of the first theorems involving an optimality condition for nonlinear optimization.

**Theorem 9.4.1.** (a) Let  $P = \text{conv}\{x_1, \dots, x_N\}$  be a polytope. The Löwner-John ellipsoid  $\mathcal{E}_{out}(P)$  is equal to the unit ball if and only if  $P$  is contained in the unit ball and if there are numbers  $\lambda_1, \dots, \lambda_N \geq 0$  so that

$$\sum_{i=1}^N \lambda_i x_i = 0 \quad \text{and} \quad \sum_{i=1}^N \lambda_i x_i x_i^\top = I_n$$

holds.

(b) Let  $P$  be a polytope. The Löwner-John ellipsoid  $\mathcal{E}_{in}(P)$  is equal to the unit ball if and only if  $P$  is containing the unit ball and if there are unit vectors  $u_1, \dots, u_N$  on the boundary of  $P$  and there are positive numbers  $\lambda_1, \dots, \lambda_N$  so that

$$\sum_{i=1}^N \lambda_i u_i = 0 \quad \text{and} \quad \sum_{i=1}^N \lambda_i u_i u_i^\top = I_n$$

holds.

Before we give the proof we comment on the optimality conditions. The first equality makes sure that not all the vectors  $x_1, \dots, x_N$ , respectively  $u_1, \dots, u_N$ , lie on one side of the sphere. The second equality shows that the vectors behave

---

<sup>1</sup>In the literature the terminology seems to differ from author to author.

similar to an orthonormal basis in the sense that we can compute the inner product of two vectors  $x$  and  $y$  by

$$x^T y = \sum_{i=1}^n \lambda_i (x_i^T x) (x_i^T y).$$

*Proof.* Statement (a) follows from the optimality conditions of the underlying determinant maximization problem. See Exercise 9.1 (b).

Statement (b) follows from (a) by polarity:

Let  $C \subseteq \mathbb{R}^n$  be a convex body. Its *polar body*  $C^*$  is

$$C^* = \{x \in \mathbb{R}^n : x^T y \leq 1 \text{ for all } y \in C\}.$$

The unit ball is self-polar,  $B^* = B$ . Furthermore, for every ellipsoid  $\mathcal{E}$  we have

$$\text{vol } \mathcal{E} \text{ vol } \mathcal{E}^* \geq (\text{vol } B)^2,$$

because direct verification yields

$$(\mathcal{E}_{A,c})^* = \mathcal{E}_{A',c'} \quad \text{with} \quad A' = \left( \frac{A^T A}{(1 + \frac{1}{4} c^T (A^T A)^{-1} c)} \right)^{-1/2}, \quad c' = -\frac{1}{2} (A^T A)^{-1} c,$$

and so

$$\det A \det A' \geq 1.$$

Let  $P$  be a polytope and assume that  $\mathcal{E}_{in}(P) = B$ . We will now prove by contradiction that  $B = \mathcal{E}_{out}(P^*)$ . For suppose not. Then the volume of  $\mathcal{E}_{out}(P^*)$  is strictly smaller than the volume of  $B$  since  $P^* \subseteq B$ . However, by taking the polar again we have

$$\mathcal{E}_{out}(P^*)^* \subseteq P,$$

and  $\text{vol } \mathcal{E}_{out}(P^*)^* > \text{vol}(B)$  a contradiction. So by (a) we have for the vertices  $x_1, \dots, x_M$  of  $P$ , the conditions

$$\sum_{i=1}^M \lambda_i x_i = 0 \quad \text{and} \quad \sum_{i=1}^M \lambda_i x_i x_i^T = I_n,$$

for some non-negative  $\lambda_i$ . We may assume after reordering that only  $\lambda_1, \dots, \lambda_N$  are strictly positive. Then the unit vectors  $u_i = x_i$ , with  $i = 1, \dots, N$ , lie on the boundary of the polytope  $P$  because

$$P = (P^*)^* = \{x \in \mathbb{R}^n : x_i^T x \leq 1, i = 1, \dots, N\}.$$

Now let

$$\mathcal{E}_{in}(P) = \{x \in \mathbb{R}^n : (x - c)^T A^{-2} (x - c) \leq 1\}$$

be the optimal inner approximation of  $P$ . We want to derive from the optimality conditions that  $\det A \leq 1$  holds.

First we realize that the second optimality condition implies that the equation  $\sum_{i=1}^N \lambda_i = n$  holds; simply take the trace.

The points

$$x_i = c + (u_i^\top A^2 u_i)^{-1/2} A^2 u_i$$

lie in  $\mathcal{E}_{in}(P)$  and so  $u_i^\top x_i \leq 1$  holds because this inequality determines a supporting hyperplanes of  $P$ . Then,

$$n \geq \sum_{i=1}^N \lambda_i u_i^\top x_i = \sum_{i=1}^N \lambda_i (u_i^\top A^2 u_i)^{1/2}$$

where we used the first equality when simplifying  $\sum_{i=1}^N \lambda_i u_i^\top c = 0$ . The trace of  $A$  can be estimated by using the second equality

$$\langle A, I_n \rangle = \langle A, \sum_{i=1}^N \lambda_i u_i u_i^\top \rangle = \sum_{i=1}^N \lambda_i u_i^\top A u_i \leq \sum_{i=1}^N \lambda_i (u_i^\top A^2 u_i)^{1/2} \leq n,$$

where we used in the first inequality the spectral factorization of  $A = P^\top D P$ , with orthogonal matrix  $P$  and diagonal matrix  $D$ , together with the Cauchy-Schwarz inequality

$$(u_i^\top P^\top D)(P u_i) \leq (u_i^\top P^\top D^2 P u_i)^{1/2} ((P u_i)^\top P u_i)^{1/2} = (u_i^\top P^\top D^2 P u_i)^{1/2}.$$

Now we finish the proof by realizing that  $(\ln x \leq x - 1)$

$$\ln \det A \leq \text{Tr}(A) - n \leq 0,$$

and so  $\det A \leq 1$ . □

This optimality condition is helpful in surprisingly many situation. For example one can use them to prove an estimate on the quality of the inner and outer approximation.

**Corollary 9.4.2.** *Let  $P \subseteq \mathbb{R}^n$  be an  $n$ -dimensional polytope, then there are invertible affine transformations  $T_{in}$  and  $T_{out}$  so that*

$$B = T_{in} \mathcal{E}_{in}(P) \subseteq T_{in} P \subseteq n T_{in} \mathcal{E}_{in}(P) = n B$$

and

$$\frac{1}{n} B = \frac{1}{n} T_{out} \mathcal{E}_{out}(P) \subseteq T_{out} P \subseteq T_{out} \mathcal{E}_{out}(P) = B$$

holds.

*Proof.* We only prove the first statement, the second follows again by polarity.

It is clear that we can map  $\mathcal{E}_{in}(P)$  to the unit ball by an invertible affine transformation. So we can use the equations

$$\sum_{i=1}^N \lambda_i u_i = 0 \quad \text{and} \quad \sum_{i=1}^N \lambda_i u_i u_i^\top = I_n$$

to show  $T_{in}P \subseteq n\mathcal{E}_{in}(P)$ . By taking the trace on both sides of the second equations we also have

$$\sum_{i=1}^N \lambda_i = n.$$

The supporting hyperplane through the boundary point  $u_i$  of  $P$  is orthogonal the unit vector  $u_i$  (draw a figure). Hence,

$$B \subseteq P \subseteq Q = \{x \in \mathbb{R}^n : u_i^\top x \leq 1, i = 1, \dots, N\}.$$

Let  $x$  be in  $Q$ , then because  $x^\top u_i \in [-\|x\|, 1]$  we have

$$\begin{aligned} 0 &\leq \sum_{i=1}^N \lambda_i (1 - x^\top u_i) (\|x\| + x^\top u_i) \\ &= \|x\| \sum_{i=1}^N \lambda_i + (1 - \|x\|) \sum_{i=1}^N \lambda_i x^\top u_i - \sum_{i=1}^N \lambda_i (x^\top u_i)^2 \\ &= \|x\|n + 0 - \|x\|^2, \end{aligned}$$

and so  $\|x\| \leq n$ . □

If  $P$  is centrally symmetric, i.e.  $P = -P$ , then in the above inequalities  $n$  can be replaced by  $\sqrt{n}$ . See Exercise 9.1 (c).

Another nice mathematical application of the uniqueness Löwner-John ellipsoids is the following.

**Proposition 9.4.3.** *Let  $P$  be a polytope and consider the group  $G$  of all affine transformations which map  $P$  into itself. Then there is an affine transformation  $T$  so that  $TGT^{-1}$  is a subgroup of the orthogonal group.*

*Proof.* Since the volume is invariant under affine transformations with determinant equal to 1 or  $-1$  (only those affine transformations can be in  $G$ ) and since the Löwner-John ellipsoid is the unique maximum volume ellipsoid contained in a polytope we have

$$A\mathcal{E}_{in}(P) = \mathcal{E}_{in}(AP) = \mathcal{E}_{in}(P)$$

for all  $A \in G$ .

Let  $T$  be the affine transformation which maps the Löwner-John ellipsoid  $\mathcal{E}_{in}(P)$  to the unit ball  $B$ . Then for every  $A \in G$

$$TAT^{-1}B = TA\mathcal{E}_{in}(P) = T\mathcal{E}_{in}(P) = B.$$

So  $TAT^{-1}$  leaves the unit ball invariant, hence it is an orthogonal transformation. □



## 9.5 Further reading

Many examples of determinant maximization problems are in Vandenberghe, Boyd and Wu [5]. They treat matrix completion problems, risk-averse linear estimation, experimental design, maximum likelihood estimation of structured covariance matrices, and Gaussian channel capacity. Next to this, they also develop the duality theory and an interior point algorithm for determinant maximization problems.

For more information on convex spectral functions and general eigenvalue optimization problems the survey [4] by Lewis is a good start.

Many more examples of computing ellipsoidal approximations are in the book [3][Chapter 4.9], especially ellipsoidal approximations of unions and intersections of ellipsoids and approximating sums of ellipsoids.

The Löwner-John ellipsoid is an important and useful concept in geometry, optimization, and functional analysis. For instance, Lenstra's polynomial time algorithm for solving integer programs in fixed dimension is based on it (see LNMB course: Integer programming methods).

Another excellent and very elegant source on applications of the Löwner-John ellipsoid in geometry and functional analysis is by Ball [2]. He uses John's optimality criterion to give a reverse isoperimetric inequality (the ratio between surface and volume is maximized by cubes) and to prove Dvoretzky's theorem (high dimensional convex bodies have almost ellipsoidal slices). The proof of the second part of Theorem 9.4.1 (b) is from the beautiful note of Ball [1].

One general strategy when working with convex sets is to find an affine transformation of the convex set so that the unit ball and the convex set are as close as possible. Here the notion of closeness depends of course on the question. In many cases these affine transformations can be found by solving an optimization problem involving positive definite matrices.

## 9.6 Exercises

- 9.1\*\* (a) Prove weak duality of determinant maximization problems: Let  $X$  be a solution of the primal and let  $y$  be a solution of the dual. Then,

$$b^T y - \ln \det \left( \sum_{j=1}^m y_j A_j - C \right) - (n + \langle C, X \rangle + \ln \det X) \geq 0.$$

Hint:  $\ln x \leq x - 1$ .

- (b) Prove Theorem 9.4.1 (a).  
(c) Show the strengthening of Corollary 9.4.2

$$B = T_{in} \mathcal{E}_{in}(P) \subseteq T_{in} P \subseteq \sqrt{n} T_{in} \mathcal{E}_{in}(P) = \sqrt{n} B$$

in the case of centrally symmetric polytopes  $P$ .

(d) Find a polytope  $P$  for which the inclusion

$$B \subseteq P \subseteq nB$$

cannot be improved. Find a centrally symmetric polytope for which

$$B \subseteq P \subseteq \sqrt{n}B$$

cannot be improved.

9.2 (a) Show that the sum of the largest  $k$  eigenvalues of a symmetric matrix is a convex spectral function.

(b) True or false: The second largest eigenvalue of a symmetric matrix is a convex spectral function.

9.3 Compute the gradient of the function

$$F : \mathcal{S}_{>0}^n \rightarrow \mathbb{R}, \quad X \mapsto -\ln \det X.$$

---

## BIBLIOGRAPHY

- [1] K.M. Ball, *Ellipsoids of maximal volume in convex bodies*, Geometriae Dedicata **41** (1992), 241–250.
- [2] K.M. Ball, *An Elementary Introduction to Modern Convex Geometry*, pp. 1–58 in: *Flavors of Geometry* (S. Levy (ed.)), Cambridge University Press, 1997.  
<http://library.msri.org/books/Book31/files/ball.pdf>
- [3] A. Ben-Tal, A. Nemirovski, *Lectures on Modern Convex Optimization*, SIAM 2001.
- [4] A.S. Lewis, *The mathematics of eigenvalue optimization*, Math. Program., Ser. B **97** (2003), 155–176.  
<http://people.orie.cornell.edu/~aslewis/publications/03-mathematics.pdf>
- [5] L. Vandenberghe, S. Boyd, S.-P. Wu, *Determinant Maximization with Linear Matrix Inequality Constraints*, SIAM Journal on Matrix Analysis and Applications **19** (1998), 499–533.  
<http://www.stanford.edu/~boyd/papers/maxdet.html>