Engineering applications of sum of squares

In this lecture, we consider applications of sum of squares polynomials to different areas in engineering and applied mathematics. The focus of the lecture will be on applications in dynamical systems and control, probability, and statistics. Other applications arise in optimization, packing problems, automated theorem proving, and quantum physics, but are not covered in these notes.

1 Dynamical systems and control

A dynamical system is a system whose state varies over time. Broadly speaking, a *state* is a vector $x(t) \in \mathbb{R}^n$ that describes the system at time t with enough information that one can predict future values of the state if the system is left to its own devices. For example, if we consider a physical system such as a rolling ball, then one could, e.g., consider the position of its center as a 3-dimensional state vector. Or, if the problem at hand is a study of the evolution of the population of wolves and sheep in a certain area, the state vector would simply encompass the current number of wolves and sheep in that area.

As the state vector contains enough information that one can predict its evolution if there is no outside interference, we are able to relate future states back to the current state via so-called *state equations*. Their expression varies depending on whether the system is *discrete time* or *continuous time*. In a discrete-time system, the state $x(k)$ is defined for discrete times $k = 0, 1, 2, \ldots$, and we have

$$x(k+1) = f(x(k)), \tag{1}$$

where f is some function from $\mathbb{R}^n$ to $\mathbb{R}^n$. In a continuous-time system, the state $x(t)$ varies continuously with time $t \geq 0$ and we have

$$\frac{dx(t)}{dt} = f(x(t)), \tag{2}$$

where again f is some function from $\mathbb{R}^n$ to $\mathbb{R}^n$. The goal is generally to understand how the *trajectory* $\{x(k)\}_k$, solution to (1), or $t \mapsto x(t)$, solution to (2), behaves over time. Sometimes such solutions can be computed explicitly and then it is easy to infer their behavior: this is the case for example when f is linear, that is, when $f(x) = Ax$ where $A \in \mathbb{R}^{n \times n}$; see, e.g., [10]. However, when f is more complex, computing closed-form solutions to (1) or (2) can be hard, even impossible, to do. The goal is then to get insights as to different properties of the trajectories without ever having to explicitly compute them. For example, it may be enough to know that the ball we were considering earlier avoids a certain puddle, or that our wolf population always stays within a certain range. This is where sum of squares polynomials come into play—as algebraic certificates of properties of dynamical systems. In Sections 1.1.1 and 1.1.2, for example, we will see how we can certify *stability* and *collision avoidance* of polynomial dynamical systems (i.e., dynamical systems as in (1) and (2) where f is a polynomial) using sum of squares.

We will also review more complex models that better describe the dynamics of our system than what is given in (1) and (2). For example, we have assumed here that our dynamical system is *autonomous*. This means that the function f only depends on $x(t)$ or $x(k)$. But this need not be the case. The function f could also depend on, say, an external input $u(t) \in \mathbb{R}^p$. This is a well-studied class of dynamical systems and the vector $u(t)$ is termed a *control*. We will briefly touch upon an example of such a system in Section 1.1.1. Another alternative to (1) and (2) could be a direct dependency of f on time on top of its dependency on $x(t)$ or $x(k)$. In this case, such a system is called *time-varying*. We will see an example of such a system in Section 1.2. In its most general setting, f can be a function of all three: time, state, and control, but we do not cover problems of this type in their full generality here. If this is of interest to the reader, we recommend reading, e.g., [34].

1.1 Certifying properties of a polynomial dynamical system

Unless otherwise specified, we consider here a continuous-time polynomial dynamical system:

$$\dot{x} = f(x), \tag{3}$$

where $\dot{x}$ is the derivative of $x(t)$ with respect to t and $f : \mathbb{R}^n \to \mathbb{R}^n$ is a vector, every component of which is a polynomial.

1.1.1 Stability

Let $\bar{x}$ be an equilibrium point of (3), that is $f(\bar{x}) = 0$. Note that by virtue of the latter definition, any system that is initialized at its equilibrium point will remain there indefinitely. For convenience, we will assume without loss of generality that the equilibrium point is the origin: this can indeed always be achieved by simply performing a change of variables $y = x - \bar{x}$ in (3). Our goal is to study how the system behaves around its equilibrium point.

Definition 1. *The equilibrium point $\bar{x} = 0$ of (3) is said to be stable if, for every $\epsilon > 0$, there exists $\delta(\epsilon) = \delta > 0$ such that*

$$||x(0)|| < \delta \Rightarrow ||x(t)|| < \epsilon, \ \forall t \geq 0.$$

This notion of stability is known as stability in the sense of Lyapunov, in honor of the Russian mathematician Aleksandr Lyapunov (1857-1918), who died tragically at the age of 61, shooting himself in the head a few hours after the death of his wife.

Intuitively, this notion of stability corresponds to what we would expect it to be: if we can allow for (up to) ϵ-magnitude deviations in our trajectory from the equilibrium point overall, then our system can always withstand some amount of initial perturbation (the magnitude of which is specified by δ). In other words, there always exists a ball around the equilibrium point from which trajectories can start with the guarantee that they will remain close to the equilibrium in the future, where the notion of "close" can be defined as needed.

Definition 2. *The equilibrium point $\bar{x} = 0$ of (3) is said to be locally asymptotically stable if it is stable around 0 and if there exists δ' such that*

$$||x(0)|| < \delta' \Rightarrow \lim_{t \to \infty} x(t) = 0.$$

Definition 3. *The equilibrium point $\bar{x} = 0$ of (3) is said to be globally asymptotically stable (GAS) if it is stable around 0 and if, $\forall x(0) \in \mathbb{R}^n$, $\lim_{t \to \infty} x(t) = 0$.*

We will focus on how one can show *global* asymptotic stability of an equilibrium point in the rest of this section. Analogous results to the ones discussed here exist for both stability and local asymptotic stability and can be found in [34, Chapter 4]. The key element to show global asymptotic stability as we will see next is the existence of a function with certain properties, called a *Lyapunov function*. The idea of searching for such functions to show properties of dynamical systems was first developed by Lyapunov in his thesis [44]. The theorem we give below appears in, e.g., [34].

Theorem 1. *Let $\bar{x} = 0$ be an equilibrium point for (3). If there exists a function $V : \mathbb{R}^n \to \mathbb{R}$ in C^1 such that*

(i) V is radially unbounded, i.e., $||x|| \to \infty \Rightarrow V(x) \to \infty$

(ii) V is positive definite, i.e., $V(x) > 0, \forall x \neq 0$ and $V(0) = 0$

(iii) $\dot{V}(x) := \nabla V(x)^T f(x) < 0$ for all $x \neq 0$ and $\dot{V}(0) = 0$ (here, $\nabla V(x)$ is the gradient of V)

then $\bar{x}$ is globally asymptotically stable.

Such a function is called a Lyapunov function and can be viewed as the generalization of an energy function. Similarly $\dot{V}$ can be viewed as the generalization of a dissipation function. Note that $\dot{V}$ is also the derivative of V with respect to its trajectory as it is equal to $\frac{d}{dt}V(x(t))$ where $x(t)$ is a solution to (3). The proof of the theorem is omitted but can be found in [34, Chapter 4].

As we just saw, the theorem above states a sufficient condition for the equilibrium point to be GAS. Is it the case that whenever the system is GAS, such a Lyapunov function exists? These type of questions give rise to what is known as *converse theorems*. The one given below comes from [36] but this precise formulation appears in [11].

Theorem 2. *Let f be continuous. If $\bar{x} = 0$ is globally asymptotically stable for (3) then there exists a function $V : \mathbb{R}^n \to \mathbb{R}$ in C^∞ satisfying properties (i)-(iii) of Theorem 1.*

Similar theorems to Theorem 2 exist for stability and local asymptotic stability; see [11]. Theorems such as these do not help us however in finding such a function V, as they simply claim its existence within the C^∞ class. To compute Lyapunov functions, we search over the class of polynomial functions. In the context of polynomial dynamical systems, this would seem like an appropriate choice. They are finitely parameterized when their degree is fixed, and so searching for them amounts to searching for a finite number of scalars. Furthermore, polynomials approximate to arbitrary accuracy any continuous function on a compact set. But how restrictive is it in practice to consider polynomial Lyapunov functions? Can we hope for a Theorem such as Theorem 2 with C^∞ replaced by "the set of polynomial functions"? The answer to this is no, as is made clear by the converse theorem that we give now.

Theorem 3. *[6] Consider the polynomial vector field*

$$\begin{aligned} \dot{x} &= -x + xy \\ \dot{y} &= -y. \end{aligned} \tag{4}$$

The origin is a globally asymptotically stable equilibrium point, but the system does not admit a polynomial Lyapunov function.

The proof of this theorem is omitted here but can be found in [6]. The crux of it relies on showing global asymptotic stability via a non polynomial Lyapunov function $V(x, y) = \ln(1 + x^2) + y^2$, and then showing that no polynomial Lyapunov function could exist due to the exponential growth rates of the trajectories (see Figure 1).

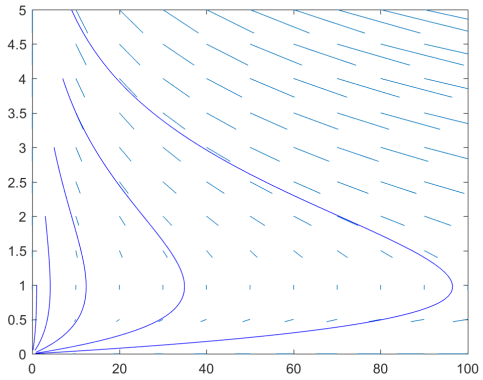


Figure 1: Representation of the polynomial vector field given in (4) with some trajectories

Though this result is negative in nature, it is worth noting that some positive results do exist. In particular, in [55] it is shown that *exponentially stable* polynomial dynamical systems always have polynomial Lyapunov functions on compact sets (we do not define exponential stability here but, at a high level, it is a stronger notion than asymptotic stability as it requires rates of convergence of trajectories to the equilibrium point rather than simply convergence).

Restricting ourselves to polynomial functions does not make the task of searching for Lyapunov functions satisfying (i)-(iii) any easier. Indeed, conditions (ii)-(iii) involve constraining polynomials to be positive over $\mathbb{R}^n$ and we know that simply testing whether a quartic is nonnegative globally is NP-hard [49]. As expected, this is where sum of squares polynomials come into play. A few references on the use of sum of squares optimization in showing asymptotic stability of a polynomial system include [53, 30, 51]. We present a condensed version of these references below.

Definition 4. *A polynomial function V is a sum of squares Lyapunov function for the polynomial system in (3) if*

(i') V is sos

(ii') $-\dot{V}$ is sos.

Note that as $V(0) = 0$ and $\dot{V}(0) = 0$, we must set the constant and linear terms to zero. It is clear that requiring V to be sos and $-\dot{V}$ to be sos implies that they will be nonnegative. This is not however what is required in Theorem 1: there, V and $-\dot{V}$ need to be positive definite. Furthermore, V has to be radially unbounded. How can positive definiteness and radial unboundedness be enforced in practice? One suggestion to enforce positive definiteness of V and $-\dot{V}$ is given in [52, Proposition 5], that we repeat here.

Proposition 1. *Given a polynomial $V(x)$ of degree $2d$, let $\phi_\epsilon(x) = \sum_{i=1}^n \sum_{j=1}^d \epsilon_{ij} x_i^{2j}$ where $\epsilon_{ij} \geq 0$ for all i and j and*

$$\sum_{j=1}^{d} \epsilon_{ij} > \gamma, \text{ for all } i = 1, \ldots, n$$

with γ some fixed positive number. Then, if there exists some $\epsilon = (\epsilon_{ij})_{ij}$ verifying the previous conditions and $V - \phi_\epsilon$ is sos, it follows that V is positive definite.

In the case where V is taken to be a homogeneous polynomial of degree $2d$[1], then one need only keep the monomials of degree $2d$ in $\phi_\epsilon(x)$. In other words, we constrain $V(x) - \sum_{i=1}^n \epsilon_i x_i^{2d}$ to be sos.

For radial unboundedness, it is well known that a polynomial V is radially unbounded if its top homogeneous component, i.e., the homogeneous polynomial formed by the collection of the highest order monomials of V, is positive definite. This can be enforced as described in the paragraph above.

In practice however, as discussed in [2, page 41], these conditions are unwieldy and can usually be done away with. Indeed, finding a feasible polynomial V that satisfies conditions (i')-(ii') is a sum of squares program. When solving programs of this type with interior point methods, the solution returned is at the analytical center of the feasible set, which is generally far from the boundary. Hence, the solution cannot be a nonnegative (but not positive definite) polynomial as these lie on the boundary. This implies that overall, one would obtain polynomials V that satisfy conditions (i)-(iii). This should be checked numerically however. For (i)-(ii), this can be done by checking the eigenvalues of the Gram matrices associated to V and to $-\dot{V}$; for (iii), this can be done by checking the eigenvalues of the Gram matrix associated to the top homogeneous component of V.

We saw that being a sum of squares polynomial is a sufficient, but not necessary, condition for being nonnegative (under certain conditions on the number of variables and degree) . It does not automatically follow however that conditions (i')-(ii') are more conservative than (ii)-(iii) for polynomial V. Indeed, there may be many polynomials satisfying conditions (ii)-(iii), some of which not having a sum of squares certificate, but as long as one of them does, then (i')-(ii') should not be more conservative than (ii)-(iii), with the technical details considered above in mind.

Hence, we now turn our attention to converse questions around the existence of *sum of squares Lyapunov functions* if a polynomial Lyapunov function is known to exist. The first result is a negative one: if a polynomial Lyapunov function of degree $2d$ exists, it does not follow that an sos Lyapunov function of degree $2d$ exists; see an example in [8, Section 3.1]. The related question as to whether an sos Lyapunov function of higher degree exists if a polynomial Lyapunov function exists is unknown for general polynomial dynamical systems. When we restrict ourselves to homogeneous polynomial dynamical systems (i.e., f is homogeneous) however, it is known to hold, see [8].

The specific case of linear systems. In the particular case where the dynamical system is linear, that is

$$\dot{x} = Ax \tag{5}$$

where $A \in \mathbb{R}^{n \times n}$, the previous results simplify considerably. Indeed, $\bar{x} = 0$ is a GAS equilibrium point for (5) if and only if a quadratic Lyapunov function V exists. As V is quadratic, it can be parametrized as $V(x) = x^T P x$ where $P \in \mathbb{R}^{n \times n}$ is positive semidefinite. Enforcing conditions (i)-(iii) then simply amounts to searching for a matrix P such that

$$P \succ 0 \text{ and } A^T P + PA \succ 0.$$

[1] A function f of degree $2d$ is said to be homogeneous if $f(\lambda x) = \lambda^{2d} f(x)$, for any scalar λ.

This is a semidefinite program to solve. If $\bar{x}$ is GAS then such a system will be feasible; see [22, Chapter 5] and [16, Section 2.2] for the discrete-time case.

Example 1. *As an illustrative example of what we have seen so far, we consider a model of a jet engine given in [35] and revisited in [16]. The dynamics of the engine are given by*

$$\begin{aligned} \dot{x} &= -y - \frac{3}{2}x^2 - \frac{1}{2}x^3 \\ \dot{y} &= 3x - y. \end{aligned} \tag{6}$$

We wish to show that the origin is globally asymptotically stable. Using MATLAB and YALMIP[43], we search for a polynomial Lyapunov function V for this system satisfying (i') and (ii'). We start by capping the degree of V at 2, then 4. The solver returns $V = 0$ for degree 2 but a nonzero solution for degree 4. It is easy to check numerically that V is positive definite and radially unbounded, and that $-\dot{V}$ is positive definite too. Hence, the origin is GAS for (6). The vector field as well as trajectories of the system and level sets of V are plotted in Figure 2.

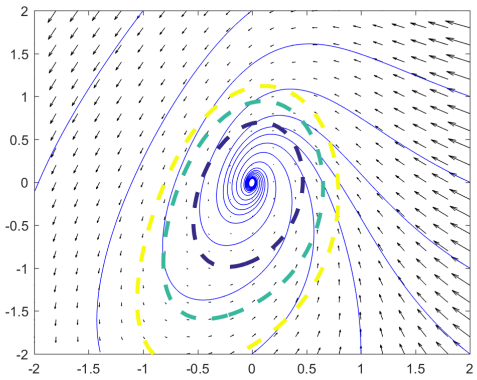


Figure 2: Plot of the vector field given in (6) together with some trajectories (thin lines) and some level sets of the Lyapunov function (dashed thick lines)

Control. So far, we have seen systems of the type $\dot{x} = f(x)$, i.e., autonomous dynamical systems. As mentioned briefly in the introduction, it can be the case that the dynamics depend on the state $x(t)$ but also on an external output $u(x(t))$, called a *control*, i.e.

$$\dot{x} = f(x(t), u(x(t))).$$

One can study many properties of such systems, but if one wants to focus on stability, a natural question to answer is how can one go about designing the controller u in such a way that the size of the region of stability (i.e., the set of initial states from which a trajectory can start and be GAS around its equilibrium) is maximized? We briefly present the results given in [32, 46]. We consider a polynomial control affine system

$$\dot{x} = f(x) + g(x)u(x),$$

where x is the state variable and $u(x)$ is the control. If we can find a Lyapunov function $V(x)$ and a sublevel set $B_\rho = \{x \mid V(x) \leq \rho\}$ of V such that:

$$x \in B_\rho, x \neq 0 \Rightarrow V(x) > 0 \text{ and } \dot{V}(x) < 0, \tag{7}$$

then B_ρ is a subset of the true region of attraction. To do this, we solve

$$\begin{aligned} \max_{\rho, L(x), u(x), V(x)} \quad & \rho \\ \text{s.t. } & V(x) \text{ sos} \\ & -\dot{V}(x) + L(x)(V(x) - \rho) \text{ sos} \\ & L(x) \text{ sos} \\ & V(\sum_j e_j) = 1, \end{aligned}$$

where e_j is the j^{th} standard basis vector for the state space $\mathbb{R}^n$, and $\dot{V}(x) = \frac{\partial V(x)}{\partial x}^T (f(x) + g(x)u(x))$. To see this, note that (7) is implied by the first, second, and third constraint. The last constraint is simply a normalization constraint which prevents ρ from getting arbitrarily big by scaling of the coefficients of V. Solving this problem is not quite an sos program: indeed, the problem is not even convex as we multiply decision variables together (e.g., $L(x)V(x)$). However by alternating optimization over V and ρ, with u and $L(x)$ fixed, and optimization over ρ, u and L with V fixed (using bisection on ρ), we are able to solve this problem using sum of squares optimization. Examples of successful implementations of such techniques can be found in the two papers [32, 46] mentioned above.

1.1.2 Collision avoidance

When Lyapunov theory was first developed, its goal was purely to certify stability of systems. Thus, Lyapunov functions originally referred to those functions whose properties certified stability of equilibrium points (such as the ones defined in Theorem 1). Now, however, the notion of a Lyapunov function has come to englobe any function that is able to certify properties of a system without requiring explicit computation of its trajectories. Following this definition, we will present another category of Lyapunov functions in this paragraph, called *barrier functions*, which prove that systems are *collision-avoidant*.

Throughout, we will consider a polynomial dynamical system as in (3). We define $\mathcal{X}_0$ and $\mathcal{X}_u$ to be two sets in $\mathbb{R}^n$. We assume that the trajectories of our system start in $\mathcal{X}_0$, i.e., $x(0) \in \mathcal{X}_0$. We would like to guarantee that all trajectories of (3) whose initial states $x(0)$ are in $\mathcal{X}_0$ do not enter the unsafe region X_u. Such a system is called collision-avoidant. A sufficient condition for the system to be collision-avoidant is the existence of a barrier certificate, as we describe below.

Theorem 4. *[56] Suppose there exists a barrier certificate, namely a function $B : \mathbb{R}^n \to \mathbb{R}$ in C^1 that satisfies the following conditions:*

(i) $B(x) > 0$ *for all* $x \in \mathcal{X}_u$

(ii) $B(x) \leq 0$ *for all* $x \in \mathcal{X}_0$

(iii) $\dot{B}(x) = \nabla B(x)^T f(x) \leq 0$ *for all* $x \in \mathbb{R}^n$

then there exists no trajectory of (3) that starts from an initial state in $\mathcal{X}_0$ and reaches another state in $\mathcal{X}_u$.

Proof. Assume that a barrier certificate satisfying the conditions above can be found. Let $x(t)$ be a trajectory in $\mathbb{R}^n$ starting at a point $x(0)$ in $\mathcal{X}_0$ and consider the evolution of $B(x(t))$ along this trajectory. By (ii), $B(x(0)) \leq 0$. Furthermore, the derivative of B along the trajectory is nonpositive from (iii). This implies that $B(x(t))$ decreases with t and hence $B(x(t))$ can never become positive. As any $x \in \mathcal{X}_u$ satisfies $B(x) > 0$, it follows that any such trajectory can never reach $\mathcal{X}_u$. □

Just as was done previously, we can search for a barrier certificate within the set of polynomial functions. Under the assumption that the sets $\mathcal{X}_u$ and $\mathcal{X}_0$ are basic semialgebraic sets, i.e., can be written as the intersection of a finite number of polynomial equalities or inequalities, we can rewrite constraints (i)-(iii) using sum of squares polynomials.

Definition 5. *Let $\mathcal{X}_u = \{x \in \mathbb{R}^n \mid g_1(x) \geq 0, \ldots, g_m(x) \geq 0\}$ and $\mathcal{X}_0 = \{x \in \mathbb{R}^n \mid \tilde{g}_1(x) \geq 0, \ldots, \tilde{g}_p(x) \geq 0\}$, where $g_1, \ldots, g_m, \tilde{g}_1, \ldots, \tilde{g}_p$ are polynomials. A sum of squares (sos) barrier function is a multivariate polynomial B such that*

(i) $B(x) = \epsilon + \sigma_0(x) + \sum_{i=1}^m \sigma_i(x) g_i(x)$, *where $\epsilon > 0$ fixed and $\sigma_i, i = 0, \ldots, m$ are sum of squares polynomials*

(ii) $-B(x) = \tau_0(x) + \sum_{i=1}^p \tau_i(x) \tilde{g}_i(x)$, *where $\tau_i, i = 0, \ldots, p$ are sum of squares polynomials*

(iii) $-\dot{B}$ *is sos.*

Note that searching for such a polynomial is a semidefinite program and that if such a polynomial exists, then it follows that it is a barrier certificate, and hence that the system is collision avoidant.

Example 2. *We illustrate the ideas in this paragraph via an example given in [56, 34]. Consider the two dimensional polynomial dynamical system*

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= -x + \frac{1}{3}x^3 - y \end{aligned} \tag{8}$$

and the sets $\mathcal{X}_0 = \{(x,y) \mid (x-1.5)^2 + y^2 \leq 0.25\}$ *and* $\mathcal{X}_u = \{(x,y) \mid (x+1)^2 + (y+1)^2 \leq 0.16\}$. *We wish to show that this system is collision avoidant. With this goal in mind, we search for an sos barrier function as defined in Definition 5 using YALMIP [43] and find one of degree 4. In Figure 3, we have plotted the two sets* $\mathcal{X}_0$ *and* $\mathcal{X}_u$ *as well as some trajectories initialized within* $\mathcal{X}_0$ *and the 0-level set of our barrier function, which truly is a physical barrier in this case.*

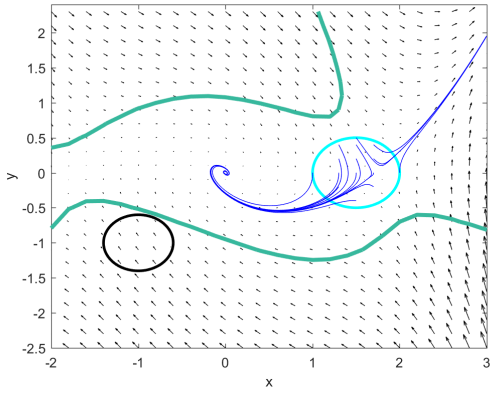


Figure 3: Vector field corresponding to the dynamical system given in (8). The initial set is the light blue circle whereas the unsafe set is the black circle. Some trajectories are plotted in dark blue. The thick green line represents the 0-level set of B. Note that we are guaranteed that no trajectory initialized in the light blue circle will cross the green line.

1.2 Stability of switched linear systems

We now transition from a continuous dynamical system to a discrete dynamical system: in this section, we consider discrete linear systems which are both uncertain and time-varying. More specifically, let Σ be a set of m real $n \times n$ matrices $A_1, \ldots, A_m$ and define the convex hull of Σ to be

$$conv(\Sigma) = \left\{ \sum_{i=1}^{m} \lambda_i A_i \mid \lambda_i \geq 0, i = 1, \ldots, m, \sum_{i=1}^{m} \lambda_i = 1 \right\}.$$

Define the following discrete-time dynamical system

$$x(k+1) = M_k x(k), \text{ where } k = 0, 1, 2 \ldots \text{ is the time index and } M_k \in conv(\Sigma). \tag{9}$$

Note that this dynamical system is linear, but time-varying as the matrix M_k changes with time, and uncertain as we only know that M_k belongs to the convex hull of a set of fixed matrices, without knowing precisely which one it is. As done previously for continuous polynomial dynamical systems, we are interested in knowing whether the equilibrium point $\bar{x} = 0$ is absolutely asymptotically stable (AAS) for (9), i.e., whether $\lim_{k\to\infty} x(k) = 0$ for any $x(0) \in \mathbb{R}^n$ and any sequence of matrices $\{M_k \mid M_k \in conv(\Sigma)\}_k$. As an example of where such a problem and system may arise, consider, e.g., the task of checking whether a drone is stable in a windy environment. By linearizing its dynamics around a desired equilibrium point, the behavior of the drone can be modeled locally by a linear dynamical system. However, as this linear dynamical system is unknown due to parameter uncertainty and modeling error, and time-varying due to the effect of the wind, the drone's behavior is better modeled by a system of the type given in (9).

Define now, for the same family of m matrices Σ, the following dynamical system, called a *switched linear system*:

$$x(k+1) = A_{\sigma(k)}x(k), \tag{10}$$

where $k = 0, 1, 2, \ldots$ is the time index and $\sigma : \mathbb{N} \to \{1, \ldots, m\}$. It so happens that the origin is AAS for (9) if and only if it is asymptotically stable under arbitrary switching (ASUAS) for (10). This means that $\lim_{k\to\infty} x(k) = 0$ for any $x(0) \in \mathbb{R}^n$ and any sequence of matrices $A_{\sigma(1)}, A_{\sigma(2)}, \ldots$ In the following, we will study ASUAS for (10) but of course, all our conclusions will naturally hold for AAS of (9).

First, when $m = 1$, the set Σ is reduced to one matrix A_1 and (10) becomes a discrete-time linear system. It is a well-known fact (see, e.g., [16, Section 2.2]) that a discrete-time linear system is asymptotically stable if and only if the spectral radius of A_1 is strictly less than one. This can be checked in polynomial-time. When $m \geq 2$, an analogous characterization holds but with a generalization of the notion of spectral radius from one matrix to a family of matrices called the *joint spectral radius*.

Definition 6. *[59] Let $\Sigma = \{A_1, \ldots, A_m\}$ be a family of m matrices of size $n \times n$. The joint spectral radius (JSR) of Σ is given by*

$$\rho(\Sigma) = \lim_{k\to\infty} \max_{\sigma\in\{1,\ldots,m\}^k} ||A_{\sigma(1)} \ldots A_{\sigma(k)}||^{1/k}, \tag{11}$$

where $||.||$ is any matrix norm.

Note that when $m = 1$, this definition collapses into

$$\rho(A_1) = \lim_{k\to\infty} ||A_1^k||^{1/k}.$$

The right hand side is the spectral radius of A_1 from Gelfand's formula, hence the joint spectral radius is equal to the spectral radius when $m = 1$. As previously mentioned, ASUAS can be characterized using the JSR, which is what we make explicit now.

Theorem 5. *The origin is ASUAS for the system given in (10) if and only if $\rho(\Sigma) < 1$.*

Unlike the setting of linear systems, where one can decide whether the spectral radius of a matrix is less than one in polynomial time, it is not known whether the problem of testing if $\rho(\Sigma) < 1$ is even decidable. The related question of testing whether $\rho(\Sigma) \leq 1$ is known to be undecidable, already when A contains only 2 matrices [20]. We refer the reader to [21] for more computational complexity results relating to the JSR. With the previous result in mind, it comes as no surprise that, e.g., stability of a switched linear system is not implied by all individual matrices in Σ having spectral radius less than one. This is easy to see on an example: consider the set of matrices Σ given by

$$A_1 = \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix} \text{ and } A_2 = \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix}.$$

Observe that the spectral radii of A_1 and A_2 are zero, which is less than one. However

$$A_1 A_2 = \begin{bmatrix} 4 & 0 \\ 0 & 0 \end{bmatrix}$$

and so $\rho(A)$ is lower bounded by $2 > 1$, and the switched linear system is not stable.

As a consequence, it is of interest to compute upperbounds on the JSR: if these bounds are strictly less than 1, then it will follow that the JSR is as well and the system will be asymptotically stable. A first theorem in this direction, which provides a stepping-stone towards the use of sum of squares polynomials, is given below.

Theorem 6. *[54, Theorem 2.2] Let $p(x)$ be a strictly positive homogeneous polynomial of degree $2d$ that satisfies*

$$p(A_i x) \leq \gamma^{2d} p(x), \forall x \in \mathbb{R}^n, \forall i = 1, \ldots, m.$$

Then, $\rho(A_1, \ldots, A_m) \leq \gamma$.

Proof. If $p(x)$ is strictly positive, then by compactness of the unit ball in $\mathbb{R}^n$ and continuity of p, there exists constants $0 < \alpha \leq \beta$ such that

$$\alpha||x||^{2d} \leq p(x) \leq \beta||x||^{2d} \text{ for all } x \in \mathbb{R}^n.$$

Then

$$\begin{aligned} ||A_{\sigma(k)} \dots A_{\sigma(1)}|| &\leq \max_x \frac{||A_{\sigma(k)} \dots A_{\sigma(1)}x||}{||x||} \\ &\leq \left(\frac{\beta}{\alpha}\right)^{1/2d} \max_x \frac{p(A_{\sigma(k)} \dots A_{\sigma(1)}x)^{1/2d}}{p(x)^{1/2d}} \\ &\leq \left(\frac{\beta}{\alpha}\right)^{1/2d} \gamma^k. \end{aligned}$$

From the definition of the joint spectral radius given in (11), by taking k^{th} roots and the limit $k \to \infty$, we immediately have the upper bound $\rho(A_1, \dots, A_m) \leq \gamma$. □

This theorem clues us into how to use sum of squares polynomials to compute upper bounds on the JSR. We define, as is done in [54], the following quantity:

$$\rho_{SOS,2d} := \begin{bmatrix} \inf_{p \text{ of degree } 2d, \gamma} \gamma \\ \text{s.t. } p \text{ sos} \\ \gamma^{2d}p(x) - p(A_i x) \text{ sos}, i = 1, \dots, m. \end{bmatrix} \tag{12}$$

Note that for fixed d and fixed γ, the computation of $\rho_{SOS,2d}$ is a semidefinite program. Similarly to Section 1.1.1, constraining p and $\gamma^{2d}p(x) - p(A_i x)$ to be sos implies that these polynomials will be nonnegative, and not positive as needed in Theorem 6. In practice, as described in Section 1.1.1, if these semidefinite programs are solved using interior point methods, then positiveness is likely to occur. Consequently, we proceed as above and check a posteriori that positivity is obtained by computing the eigenvalues of the Gram matrices associated to the sos conditions. To obtain the smallest γ such that p sos and $\gamma^{2d}p(x) - p(A_i x)$ sos, we proceed by bisection on γ. Indeed, one cannot optimize outright over γ and p as the decision variables multiply in the second constraint, making it a nonconvex optimization problem. As a consequence, we typically fix d and then solve a sequence of semidefinite programs as we bisect over γ. If the optimal value of γ found for that d is satisfactory for our purposes, we stop there; otherwise, we move on to a higher degree.

The quality of the bound on the JSR obtained using the sum of squares relaxation described in (12) can be quantified via the following theorem; interestingly, it is independent of the number m of matrices.

Theorem 7. *[54, Theorem 3.4] The sos relaxation in (12) satisfies*

$$\binom{n+d-1}{d}^{-1/2d} \rho_{SOS,2d} \leq \rho(A_1, \dots, A_m) \leq \rho_{SOS,2d}.$$

We finish with an illustrative example of the previously-developed techniques.

Example 3. *Consider a modification of Example 5.4. in [5]. We would like to show that the switched linear system defined by the following two matrices*

$$A_1 = \frac{1}{\alpha}\begin{bmatrix} -1 & -1 \\ 4 & 0 \end{bmatrix} \text{ and } A_2 = \frac{1}{\alpha}\begin{bmatrix} 3 & 3 \\ -2 & 1 \end{bmatrix},$$

where $\alpha = 3.92$ is stable under arbitrary switching. We are able to show using YALMIP that for $2d = 6$ and $\gamma = 0.9999$, we recover a feasible polynomial p for the SDP given in (12). (It can be checked that all three polynomials appearing in the sos program are positive.) It follows that $\rho(A_1, A_2) \leq 0.9999 < 1$ and hence the system is ASUAS. We showcase this in Figure 4 where we have plotted the 1-level set of p together with three random trajectories of the switched system initalized at the same point. Note that all three trajectories flow towards the origin and remain within the 1-sublevel set of p.

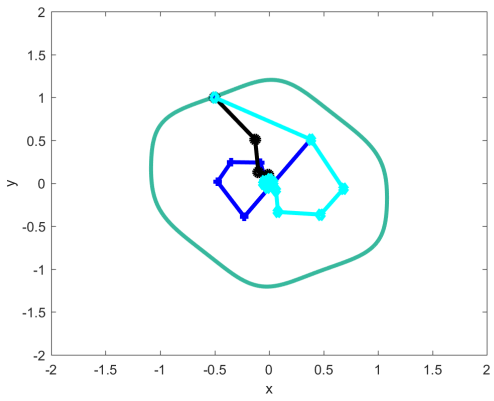


Figure 4: Random trajectories of the switched linear system described in Example 3 together with the 1-sublevel set of the function p obtained (in green)

Using the dual of (12) to generate unstable trajectories As seen above, Theorem 7 provides a lower bound on the JSR. Another way of obtaining a lower bound on the JSR is by simply computing $||A_{\sigma(1)} \dots A_{\sigma(k)}||_2^{1/k}$ for some sequence $\sigma(1), \dots, \sigma(k)$. Can one find ways of generating such sequences so that $||A_{\sigma(1)} \dots A_{\sigma(k)}||_2^{1/k}$ is arbitrarily close to the JSR as $k \to \infty$? In particular, if the JSR is strictly greater than one, is it always possible to generate unstable trajectories? This is what we consider next. To do this, we will present the results given in [41], but specialized to the case of arbitrary switching, as what is considered in the paper is more general and relates to switching governed by automata.

We first explain the process by which such a sequence is generated before presenting the theorem. Let d be an integer and γ be fixed such that $\gamma < \rho_{SOS,2d}$ where $\rho_{SOS,2d}$ is as defined in Problem (12). The key idea here is to use the dual of the feasibility problem given in (12) and the concept of pseudo-expectation which is defined in Section 2.1. As a quick reminder, the dual to the cone of sum of squares polynomials is the set of linear functionals L that map the polynomials of degree less than or equal to $2d$ to the reals, in such a way that $L(s) \geq 0$ for any sum of squares polynomial s of degree less than or equal to $2d$. These functionals are also given the name of *pseudo-expectations* and are denoted by $\tilde{E}$, as they have the property that $\tilde{E}[s(x)] \geq 0$ for all s sos, when it should in fact be the case, if $\tilde{E}$ were truly an expectation, that $\tilde{E}[s(x)] \geq 0$ for all nonnegative s. Hence, for fixed γ, the dual to the feasibility problem of (12) can be written as:

$$\begin{aligned}
&\min_{\tilde{E}_1,\dots,\tilde{E}_m} \quad 0 \\
&\text{s.t.} \sum_{i=1}^{m} \tilde{E}_i[p(A_{\sigma(i)}x)] \geq \gamma^{2d} \sum_{i=1}^{m} \tilde{E}_i[p(x)] \text{ for all } p \text{ sos} \\
&\sum_{i=1}^{m} \tilde{E}_i \left[\sum_{i=1}^{n} x_i^{2d} \right] = 1.
\end{aligned}$$

The algorithm then proceeds as follows. Let $p_0(x)$ be a polynomial in the interior of the sum of squares cone. Pick an integer σ in $\{1, \dots, m\}$ such that $\tilde{E}_\sigma[p_0(x)] > 0$ and set $p_0(x)$ to be $p_1(x)$. This can be done due to the second constraint of the previous optimization problem. Then, at iteration k, do the following: compute $\sigma(k) = \arg\max_{\sigma \in \{1,\dots,m\}} \tilde{E}_\sigma[p_k(A_\sigma x)]$ and replace $p_{k+1}(x)$ by $p_k(A_{\sigma(k)}x)$. The following theorem can then be shown.

Theorem 8. *[41, Theorem 6] For any positive integer d and having solved the dual problem with $\gamma < \rho_{SOS,2d}$, the previously-described algorithm generates a sequence $\sigma(1), \sigma(2), \dots$ such that:*

$$\lim_{k\to\infty} ||A_{\sigma(1)} \dots A_{\sigma(k)}||_2^{1/k} \geq \frac{\gamma}{m^{1/2d}}.$$

As $\rho(A_1, \dots, A_m) \geq \lim_{k\to\infty} ||A_{\sigma(1)} \dots A_{\sigma(k)}||_2^{1/k}$ and $\rho_{SOS,2d} \geq \rho(A_1, \dots, A_m)$, together with the theorem's result, it follows that as $d \to \infty$, $\lim_{k\to\infty} ||A_{\sigma(1)} \dots A_{\sigma(k)}||_2^{1/k}$ gets arbitrarily close to the JSR.

Other areas of application of the JSR. The JSR naturally appears when one wishes to determine whether a switched linear system is asymptotically stable. But this is far from the only application where it is a relevant quantity. In fact, the concept first started gaining notoriety in the context of the study of wavelets [17]. It also appears in economics [19], coding theory [33], combinatorics on words [33], and agent consensus [18], to name a few. We give a brief overview of its role in economics and multi-agent consensus here.

In 1973, Wassily Leontief won a Nobel prize in economics for his work on input-output analysis and how changes in one sector of the economy can impact other sectors. In his model of inputs and outputs, Leontief divides the economy into n sectors and postulates the following relationship between production and demand:

$$x = Ax + d, \tag{13}$$

where d is a vector in $\mathbb{R}^n_+$ where each component corresponds to demand for the sector i, x is also a vector in $\mathbb{R}^n$ where each component describes the production of sector i and A is a nonnegative $n \times n$ matrix, called the consumption matrix, that relates the production of a sector i to the production of other sectors. In other words, if one wants to produce one unit for sector i, then one would need A_{ij} units from sector j. The economy is called *productive* if there exists a nonnegative vector x satisfying (13). For this to occur, the spectral radius of A must be strictly less than one. However, it can be expected that our knowledge of the consumption matrix is uncertain. It may then be the case that instead of exactly knowing the value of A, we simply know that it belongs to the convex hull of matrices $\{A_1, \ldots, A_m\}$. In this case, to determine whether the economy is productive, one needs to consider the joint spectral radius of $\{A_1, \ldots, A_m\}$ instead; see [19] for more details.

The JSR also crops up in the context of multi-agent consensus as we will see now. Our description of the problem comes from [18]. We consider a set $N = \{1, \ldots, n\}$ of agents that try to reach agreement on a common scalar value by exchanging tentative values and combining them. More specifically, each agent i starts with a specific value $x_i(0)$ assigned to him or her. The vector $x(t) = (x_1(t), \ldots, x_n(t))$ with the values held by the agents at time $t = 0, 1, 2, \ldots$ is then updated thus

$$x(t) = A(t)x(t),$$

where $A(t)$ is a stochastic matrix. The goal of [18] is to establish conditions under which $x_i(t)$ converges to a constant c independent of i when $t \to \infty$. It so happens that a measure of the convergence rate of $x(t)$ to the vector of constants $(c, \ldots, c)$ is given by the joint spectral radius of a set of matrices corresponding to a projection of the matrices $A(s), s = 0, 1, \ldots, t$ onto the space orthogonal to the all ones vector; see [18] for more details.

2 Probability and measure theory

2.1 The moment problem

Let $(\mathbb{R}^n, \mathcal{B}, \mu)$ be a measure space where $\mathcal{B}$ is the Borel σ-algebra over $\mathbb{R}^n$ and μ is a measure on $(\mathbb{R}^n, \mathcal{B})$, called a Borel measure. We remind the reader that a *σ-algebra* over $\mathbb{R}^n$ is simply a collection of subsets of $\mathbb{R}^n$ that is closed under complement, as well as countable unions and intersections, and that the Borel σ-algebra is the σ-algebra generated by the open sets of $\mathbb{R}^n$. A *measure* is a function $\mu : \mathcal{B} \to \mathbb{R}^+ \cup \{+\infty\}$ such that $\mu(\emptyset) = 0$ and $\mu(\cup_{i=1}^{+\infty} B_i) = \sum_{i=1}^{+\infty} \mu(B_i)$ for any pairwise disjoint sets B_i in $\mathcal{B}$. An important particular case of a measure is a *probability measure*, which is the set of measures that have the property $\mu(\mathbb{R}^n) = 1$.

Let $\alpha := (\alpha_1, \ldots, \alpha_n)^T$ be a vector of integers of size n and denote by $|\alpha| := \sum_{i=1}^n \alpha_i$. For a vector of variables $x = (x_1, \ldots, x_n)^T$, we can write in shorthand x^α to mean $x_1^{\alpha_1} \ldots x_n^{\alpha_n}$. We are now ready to define the moment of order α of a measure μ on $\mathbb{R}^n$:

$$y_\alpha := \int_{\mathbb{R}^n} x^\alpha d\mu(x).$$

The *moment problem* is then simply an inverse problem: given a sequence $\{y_\alpha\}_{\alpha \in \mathbb{N}^n}$, when is it the case that this sequence is actually a sequence of moments from a measure μ? One can ask a similar question

when the sequence is a *truncated* sequence, i.e., when we have access to a sequence $\{y_\alpha\}_{\alpha\in\mathbb{N}^n,|\alpha|<c}$ where c is a constant: the problem is then called the truncated moment problem. When y *is* a sequence of moments of a measure μ, we say that μ is a representing measure for y.

Though there may seem to be no link a priori between this problem and sum of squares polynomials, they are in fact intimately related via duality. We will focus on the truncated moment problem for simplicity, but very similar results can be found in, e.g., [39] for the moment problem.

Let $\mathbb{N}^n_{2d} := \{\alpha \in \mathbb{N}^n \mid |\alpha| \leq 2d\}$ and define

$$\mathcal{M}_{n,2d} := \{\{y_\alpha\}_{\alpha\in\mathbb{N}^n_{2d}} \mid \exists \text{ a measure } \mu \text{ on } \mathbb{R}^n \text{ such that } y_\alpha = \int_{\mathbb{R}^n} x^\alpha d\mu,\ \forall \alpha \in \mathbb{N}^n_{2d}\}, \tag{14}$$

i.e., $\mathcal{M}_{n,2d}$ is the set of truncated sequences $\{y_\alpha\}$ for which $\{y_\alpha\}$ has a representing measure μ. It is easy to see that $\mathcal{M}_{n,2d}$ is a convex cone and that the truncated moment problem is exactly the problem of understanding which sequences belong to $\mathcal{M}_{n,2d}$.

To give us a better sense of what $\mathcal{M}_{n,2d}$ looks like, we study its dual cone $(\mathcal{M}_{n,2d})^*$. By definition,

$$(\mathcal{M}_{n,2d})^* = \{\{p_\alpha\}_{\alpha\in\mathbb{N}^n_{2d}} \mid \sum_\alpha p_\alpha y_\alpha \geq 0, \forall \{y_\alpha\} \in \mathcal{M}_{n,2d}\}. \tag{15}$$

It so happens that this cone is exactly the cone of nonnegative polynomials of degree less than or equal to $2d$ and in n variables. This is what we show next.

Theorem 9. *Let $P_{n,2d}$ denote the cone of nonnegative polynomials in n variables and of degree less than or equal to $2d$. We have $P_{n,2d} = (\mathcal{M}_{n,2d})^*$.*

Proof. Throughout this proof, we will identify a polynomial p in $P_{n,2d}$ by its coefficients p_α in the standard monomial basis, i.e.,

$$P_{n,2d} \triangleq \{\{p_\alpha\}_{\alpha\in\mathbb{N}^n_{2d}} \mid\ p(x) = \sum_\alpha p_\alpha x^\alpha \geq 0\}.$$

We first show that $P_{n,2d} \subseteq (\mathcal{M}_{n,2d})^*$. Let $\{p_\alpha\}_\alpha \in P_{n,2d}$. For any $\{y_\alpha\}$ in $(\mathcal{M}_{n,2d})^*$, we have:

$$\sum_\alpha p_\alpha y_\alpha = \sum_\alpha p_\alpha \int x^\alpha d\mu = \int \sum_\alpha p_\alpha x^\alpha d\mu = \int p(x) d\mu \geq 0$$

as $p(x)$ is nonnegative and the inclusion follows.

We now show that $P_{n,2d} \supseteq (\mathcal{M}_{n,2d})^*$. Let $p \notin P_{n,2d}$. Then, there exists x_0 such that $p(x_0) < 0$. Let μ be the Dirac measure at point x_0, δ_{x_0} and let $\{y_\alpha\}_{\alpha\in\mathbb{N}^n_{2d}}$ be the sequence of moments associated to μ. We have

$$\sum_\alpha p_\alpha y_\alpha = \int \sum_\alpha p_\alpha x^\alpha d\delta_{x_0}(x) = \int p(x) d\delta_{x_0}(x) = p(x_0) < 0.$$

Hence $\{p_\alpha\} \notin (\mathcal{M}_{n,2d})^*$ and we have shown the converse direction. □

Corollary 1. *It follows that $(P_{n,2d})^* = cl(\mathcal{M}_{n,2d})$ where cl denotes the closure of the set.*

Theorem (9) gives us a strategy for coming up with necessary conditions for membership to $\mathcal{M}_{n,2d}$. Indeed, let $\Sigma_{n,2d}$ denote the cone of sum of squares polynomials of degree $2d$ and in n variables. We have $\Sigma_{n,2d} \subseteq P_{n,2d}$ and so it follows that $(P_{n,2d})^* \subseteq (\Sigma_{n,2d})^*$, and hence:

$$\mathcal{M}_{n,2d} \subseteq (\Sigma_{n,2d})^*.$$

As a consequence, a necessary condition for membership to $\mathcal{M}_{n,2d}$ is membership to $(\Sigma_{n,2d})^*$. It so happens that the latter can be tested using semidefinite programming as we will see now.

Definition 7. *Given an integer d, and a truncated sequence $y = (y_\alpha)_{\alpha\in\mathbb{N}^n_{2d}}$, its moment matrix is the symmetric matrix $M_d(y)$ with rows and columns labeled by $\alpha \in \mathbb{N}^n_d$ and where the $(\alpha,\beta)^{th}$ entry is $y_{\alpha+\beta}$ for $\alpha, \beta \in \mathbb{N}^n_d$.*

Theorem 10. *Let*

$$\mathcal{M}_{\succeq,n,2d} := \{\{y_\alpha\}_{\alpha\in\mathbb{N}^n_{2d}} \mid M_d(y) \succeq 0\}.$$

We have $(\Sigma_{n,2d})^* = \mathcal{M}_{\succeq,n,2d}$.

Proof. We first show that $\Sigma_{n,2d} \subseteq (\mathcal{M}_{\succeq,n,2d})^*$. By taking the dual, it will follow that $\mathcal{M}_{\succeq,n,2d} \subseteq (\Sigma_{n,2d})^*$. Note that by definition of $(\mathcal{M}_{\succeq,n,2d})^*$, we have

$$(\mathcal{M}_{\succeq,n,2d})^* = \{\{p_\alpha\}_{\alpha\in\mathbb{N}^n_{2d}} \mid \sum_\alpha p_\alpha y_\alpha \geq 0,\ \forall\{y_\alpha\} \in \mathcal{M}_{\succeq,n,2d}\} \tag{16}$$

First, let $p \in \Sigma_{n,2d}$: there exists σ such that $p(x) = \sigma(x)^2$. Once again, we identify any polynomial $p \in \Sigma_{n,2d}$ with its coefficients. Let $\vec{\sigma} = (\sigma_\beta)_{\beta\in\mathbb{N}^n_d}$ be the coefficients of σ. It follows that $p_\alpha = \sum_{\{\beta,\gamma \mid \beta+\gamma=\alpha\}} \sigma_\beta\sigma_\gamma$ for any α, and hence for any $\{y_\alpha\} \in \mathcal{M}_{\succeq,n,2d}$,

$$\sum_\alpha p_\alpha y_\alpha = \sum_\alpha y_\alpha \sum_{\beta+\gamma=\alpha} \sigma_\beta\sigma_\gamma = \sum_\beta\sum_\gamma \sigma_\beta\sigma_\gamma y_{\beta+\gamma} = \vec{\sigma}^T M_d(y)\vec{\sigma} \geq 0$$

as $M_d(y) \succeq 0$. So $p \in (\mathcal{M}_{\succeq,n,2d})^*$.

We now show that $(\Sigma_{n,2d})^* \subseteq \mathcal{M}_{\succeq,n,2d}$. Suppose that $\{y_\alpha\} \notin M_{\succeq,n,2d}$, this means that $M_d(y) \not\succeq 0$. This implies that there exists a vector σ_0 such that $u^T M_d(y)u < 0$. Let σ be a polynomial with coefficients u and let $p = \sigma^2$. Clearly, $p \in \Sigma_{n,2d}$. However, by reprising a similar computation as above, $\sum_\alpha p_\alpha y_\alpha < 0$. This means that $\{y_\alpha\} \notin (\Sigma_{n,2d})^*$. □

Note that, given a sequence of numbers y_α, one can construct the matrix $M_d(y)$ and check its positive semidefiniteness. If it is not positive semidefinite, then y_α does not have a representing measure. To construct stronger necessary conditions for membership to $\mathcal{M}_{n,2d}$, one can simply consider well-known hierarchies of inner approximations to $P_{n,2d}$ based on sum of squares and then compute their dual cones; see [16, Section 3.5] for more details around this topic.

Remark 1. *One can rework this section taking into account measures over arbitrary basic semialgebraic sets K. The dual of the set of truncated sequences who have a representing measure μ over K will then simply be the set of polynomials nonnegative over K and so on; see [39] for more information.*

Remark 2. *Recently, Barak et al. introduced the concept of* pseudoexpectation*; see, e.g., [12]. This can be interpreted in the context of what we have seen so far. In our results and the proofs of these results, we identified the cone $\Sigma_{n,2d}$ and the set of coefficients of sos polynomials of degree $2d$ and in n variables. Thus the cone $\Sigma_{n,2d}$ we considered was a cone over $\mathbb{R}^{\mathbb{N}^n_{2d}}$. In reality, $\Sigma_{n,2d}$ is a cone over the space of polynomials of degree less than or equal to $2d$, denoted by $\mathbb{R}_{2d}[x]$. The dual cone $(\Sigma_{n,2d})^*$ is then the set of linear functionals $L : \mathbb{R}_{2d}[x] \to \mathbb{R}$ such that $L(s) \geq 0$ for any $s \in \Sigma_{n,2d}$. Note that there is an isomorphism between this set and $\mathcal{M}_{\succeq,n,2d}$ via the correspondence $L(x^\alpha) = y_\alpha$.*

The pseudoexpectation as defined in [12] is simply another name for these linear functionals, with the added constraint that $L(1) = 1$[2]. We give the formal definition that appears in [12] to contrast: A degree-l pseudoexpectation operator $\tilde{E}$ is a linear operator L that maps polynomials in $\mathbb{R}_l[x]$ into $\mathbb{R}$ and satisfies that $L(1) = 1$ and $L(P^2) \geq 0$ for every polynomial p of degree at most $l/2$.

The intuition behind the name is easy to explain. As $\mathcal{M}_{n,2d}$ is the dual (up to closure) of $P_{n,2d}$, it follows that for a measure μ, we should have

$$E[p(x)] = \int p(x)d\mu = \sum_\alpha p_\alpha \int x^\alpha d\mu \geq 0,$$

for any nonnegative polynomial p. Instead, we have

$$\tilde{E}[p(x)] \geq 0$$

for any sum of squares polynomial p. Though it resembles its counterpart, $\tilde{E}$ is not actually an expectation: it may be the case that $\tilde{E}[p(x)] < 0$ for a nonnegative polynomial p, which would not happen if it were truly an expectation.

[2]This latter constraint is because $E[1] = \int d\mu = 1$ for a probability measure.

The univariate case. The case where $n = 1$ is a special case (along with the cases $2d = 2$ and $(n = 2, 2d = 4)$) in the sense that the set of nonnegative and sum of squares polynomials coincide. In other words, when $n = 1$, $\Sigma_{1,2d} = \mathcal{P}_{1,2d}$. It then follows from from Corollary 1 that

$$(\Sigma_{1,2d})^* = cl(\mathcal{M}_{1,2d}).$$

This gives rise to the following theorem, the formulation of which comes from [16].

Theorem 11. *Let $y = (y_0, y_1, \ldots, y_{2d})$ be a sequence of real numbers such that $y_0 = 1$. If $y \in \mathcal{M}_{1,2d}$, i.e., if there exists a probability measure μ on $\mathbb{R}$ such that y_i is the i^{th} moment of μ, then $y \in (\Sigma_{1,2d})^*$, i.e.,*

$$M_d(y) = \begin{bmatrix} y_0 & y_1 & y_2 & \cdots & y_d \\ y_1 & y_2 & y_3 & \cdots & y_{d+1} \\ y_2 & y_3 & y_4 & \cdots & y_{2d+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_d & y_{d+1} & y_{d+2} & \cdots & y_{2d} \end{bmatrix}$$

is positive semidefinite. Conversely, if $M_d(y) \succ 0$, then y has a representing probability measure μ, i.e., there exists a probability measure μ such that y_i is the i^{th} moment of μ.

Note that positive definiteness of $M_d(y)$ is needed: one can construct sequences y such that $M_d(y) \succeq 0$ but y does not have a representing measure; see [16, Remark 3.147]. Furthermore, the theorem above can be extended to measures over intervals of $\mathbb{R}$ rather than measures over the whole of $\mathbb{R}$; for this, see again [16, Section 3.5.3]. Finally, while this result tells us when a sequence y has a representing measure, it does not explain how one should go about constructing such a measure. Some information as to how to do this in practice can be found in [16, Section 3.5.5].

Example 4. *We check on an easy example that the criterion given in Theorem 11 works. Consider the probability measure μ given by $\mu(dx) = f(x)dx$ where $f(x)$ is the probability distribution function of a standard normal distribution. Let $y = (1, 0, 1, 0, 3)$: y is the vector of moments of μ up to degree $2d = 4$. We construct*

$$M_2(y) = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 3 \end{bmatrix}.$$

As $M_2(y) \succ 0$, we conclude that there does exist a probability measure μ such that y_i is the i^{th} moment of μ, which is as expected.

Dual formulation of the polynomial optimization problem. Using the theory developed above, one can tackle unconstrained polynomial optimization problems of the type:

$$\min_{x \in \mathbb{R}^n} p(x), \tag{17}$$

where p is a polynomial of degree $2d$. Note that the constrained case where $x \in K$, with K basic semialgebraic, can also be considered if we consider measures over K instead; see Remark 1. As noted by Lasserre in [37], one can rewrite (17) as:

$$\min_{\text{prob measures } \mu \text{ over } \mathbb{R}^n} \int p(x) d\mu$$

As $p(x)$ is a polynomial of degree $2d$, we have $p(x) = \sum_{\alpha \in \mathbb{N}^n_{2d}} p_\alpha x^\alpha$. Plugging into the previous expression, we get

$$\min_{\text{prob measures } \mu \text{ over } \mathbb{R}^n} \sum_{\alpha \in \mathbb{N}^n_{2d}} p_\alpha \int x^\alpha d\mu.$$

One can then stop dealing with the probability measure μ itself, but only with the moments $y_\alpha := \int x^\alpha d\mu$, provided that $\{y_\alpha\}$ has a representing probability measure. The problem becomes:

$$\begin{aligned} &\min_{y_\alpha} \sum_\alpha p_\alpha y_\alpha \\ &\text{s.t. } \{y_\alpha\} \in \mathcal{M}_{n,2d}, y_0 = 1. \end{aligned} \tag{18}$$

The dual of this problem is exactly

$$\begin{aligned} &\max_\gamma \gamma \\ &\text{s.t. } p(x) - \gamma \in P_{n,2d}. \end{aligned} \tag{19}$$

Indeed, as $P_{n,2d} = (\mathcal{M}_{n,2d})^*$, we have $p(x) - \gamma \in P_{n,2d} \Leftrightarrow \sum_\alpha p_\alpha y_\alpha - \gamma y_0 \geq 0,\ \forall y_\alpha \in \mathcal{M}_{n,2d}$. This latter inequality is then equivalent to $\sum_\alpha p_\alpha y_\alpha \geq \gamma$ as $y_0 = 1$.

One can then replace $\mathcal{M}_{n,2d}$ by its outer approximation $\mathcal{M}_{\succeq,n,2d}$ in (18): we obtain lower bounds on (18) via semidefinite programming. This is equivalent to replacing in the dual (19) $P_{n,2d}$ by $\Sigma_{n,2d}$, which also gives us lower bounds on (19) using semidefinite programming.

2.2 Bounds on the probability of a random variable and applications to option pricing

Let $(\Omega, \mathcal{F}, p)$ be a measure space, where $\mathcal{F}$ is a σ-algebra over Ω and p is a probability measure. Let X be a random variable, i.e., a measurable mapping from $(\Omega, \mathcal{F}, p) \to (E, \mathcal{E})$, where $E \subset \mathbb{R}$ and $\mathcal{E}$ is a σ-algebra over E. The random variable X induces a new probability measure on $(E, \mathcal{E})$. This measure, denoted by p_X, is such that

$$p_X(S) = p(\{\omega \in \mathbb{R}^n \mid X(w) \in S\})$$

for any set $S \subseteq E$. The notation $p(X \in S)$ is sometimes used as a shorthand for $p_X(S)$. The moment of order k of X, where $k \in \{0, \ldots, K\}$, is defined as the moment of order k of p_X, i.e.,

$$\int_\Omega X^k dp = \int_{\omega \in \Omega} X^k(\omega) dp(\omega) := \int_{x \in E} x^k dp_X(x).$$

By definition of the expectation, the moment of order k of X can be viewed as the expectation of X^k and can consequently be written as $E[X^k]$.

We now describe the problem of interest. Let $\{y_k\}_{k \in \{0,\ldots,2d\}}$ be a sequence of scalars. We consider the set Φ of probability measures

$$\Phi := \{p_X \mid \int_E x^k dp_X(x) = y_k, \forall k \in \{0, \ldots, K\}\}. \tag{20}$$

Note that one can identify Φ with the set of random variables X such that the order-k moment of X coincides with y_k for $k \in \{0, \ldots, K\}$. We will assume throughout that Φ is non-empty (or in other words, $\{y_k\}$ always has at least one representing measure). The problem we are considering is then: given a sequence $\{y_k\}_{k \in \{0,\ldots,K\}}$ as described above, and a set $S \subseteq E$ described by polynomial inequalities, derive a "tight" bound on $p(X \in S) = p_X(S)$, i.e., derive $\sup_{p_X \in \Phi} p_X(S) = \sup_{X \in \Phi} p(X \in S)$.

Using moments of a random variable to upperbound the probability that it belongs to a certain set is a problem that has a rich history within the field of probability theory. Two of the most ubiquitous inequalities in the field, namely that of Markov and that of Chebychev, do exactly this. Indeed, the Markov inequality states that, for any nonnegative random variable X and positive scalar a:

$$p(X \geq a) \leq \frac{E[X]}{a}, \tag{21}$$

and the Chebychev inequality states that for any random variable X:

$$p(|X - E[X]| > t) \leq \frac{var(X)}{t^2}. \tag{22}$$

Note that the upper bound does not depend in any way on the distribution of the random variable X.

How to tackle this problem? By definition, it can be formulated as:

$$\begin{aligned} &\max_{p_X} \int_E \mathbf{1}_S dp_X \\ &\text{s.t.} \int x^k dp_X(x) = y_k, \forall k \in \{0, \dots, K\}, \end{aligned}$$

where $\mathbf{1}_S$ refers to the indicator function of S. The dual to this problem is then exactly

$$\begin{aligned} &\min_{\lambda_k} \sum_{k=0}^{K} \lambda_k y_k \\ &\text{s.t.} \sum_{k=0}^{K} \lambda_k x^k \geq \mathbf{1}_S, \forall x \in E. \end{aligned} \tag{23}$$

Indeed,

$$\int_E \mathbf{1}_S dp_X \leq \int_E \sum_k \lambda_k x^k dp_X = \sum_k \lambda_k \int_E x^k dp_X = \sum_k \lambda_k y_k.$$

Strong duality holds under certain conditions; see, e.g., [14]. If we define λ to be the polynomial $\lambda(x) = \sum_k \lambda_k x^k$, we can rewrite (23) as

$$\begin{aligned} &\min_{\lambda} \sum_k \lambda_k y_k \\ &\text{s.t. } \lambda(x) - 1 \geq 0, \forall x \in S \\ &\quad \lambda(x) \geq 0, \forall x \in E. \end{aligned}$$

As we are enforcing nonnegativity of polynomials over $E \subseteq \mathbb{R}$ or S, we can then simply use sum of squares polynomials to obtain upper bounds on the optimal value of the problem. In the case where E and S are basic semialgebraic sets, this can be done exactly; see [14, 38] for more complex cases such as the multivariate case (i.e., X is a random vector).

Example 5. *We use these methods to see whether the Markov inequality (21) and the Chebychev inequality (22) are tight.*

We start with trying to find an upper bound on $p(X \geq a)$ where $a > 0$ and X nonnegative, using only first moment information. Let X be a nonnegative random variable whose distribution is unknown but its first moment $E[X]$ is known. We have $K = 1$, S is $[a, \infty)$, and E, which is where X takes its values, is $[0, \infty)$. The fact that $K = 1$ implies that $\lambda(x)$ is an affine polynomial, i.e., $\lambda(x) = \lambda_0 + \lambda_1 x$. Hence the problem to solve is the following:

$$\begin{aligned} &\min_{\lambda} \ \lambda_0 + \lambda_1 E[X] \\ &\textit{s.t. } \lambda(x) - 1 \geq 0, \ \forall x \geq a \\ &\quad \lambda(x) \geq 0, \ \forall x \geq 0. \end{aligned}$$

(Note that $y_0 = 1$ as we are considering a probability measure.) One can rewrite the constraints exactly using [16, Section 3.3.1]:

$$\begin{aligned} &\min_{\lambda} \ \lambda_0 + \lambda_1 E[X] \\ &\textit{s.t. } \lambda(x) - 1 = \sigma + \tau \cdot (x - a), \sigma \geq 0, \tau \geq 0 \\ &\quad \lambda(x) = \sigma' + \tau' \cdot x, \sigma' \geq 0, \tau \geq 0, \end{aligned} \tag{24}$$

which is a linear program. It is quite easy to see that

$$\lambda(x) = \frac{1}{a} x$$

is feasible for (24). Indeed, $\frac{1}{a} \geq 0$ and $\lambda(x) - 1 = \frac{1}{a}(x - a)$. The value of the objective is then $\frac{E[X]}{a}$. Hence, $p(X \geq a) \leq E[X]/a$. Is this upperbound tight? It is in the case where $E[X]/a \leq 1$. Indeed, in that case define:

$$X_0 = \begin{cases} a \textit{ with probability } E[X]/a \\ 0 \textit{ with probability } 1 - E[X]/a \end{cases}.$$

We have that $X_0 \in \Phi$ as $E[X_0] = E[X]$. Furthermore, $p(X_0 \geq a) = \frac{E[X]}{a}$. When $E[X]/a \geq 1$, then the bound that is tight is simply 1. This is always an upperbound (take $\lambda_0 = 1$ and $\lambda_1 = 0$) and it is tight in this case as $X_0 = E[X]$ with probability 1 belongs to Φ and achieves the bound $p(X_0 \geq a) = 1$. Hence, a tight upper bound on $p(X \geq a)$ using first order information is given by

$$sup_{\Phi} p(X \geq a) = \begin{cases} E[X]/a & \text{if } E[X]/a \leq 1 \\ 1 & \text{if } E[X]/a > 1 \end{cases}.$$

The first case corresponds to the Markov bound.

Now consider a random variable X whose distribution is unknown but whose first and second order moments, $E[X]$ and $E[X^2]$, are known. We are looking for an upper bound on

$$p(|X - E[X]| \geq t) = p(\{X \geq t + E[X]\} \cup \{X \leq -t + E[X]\}),$$

where $t > 0$, that involves only $E[X]$ and $E[X^2]$. In this case, $K = 2$, $S = (-\infty, -t + E[X]] \cup [t + E[X], +\infty)$, $E = \mathbb{R}$, and we have $\lambda(x) = \lambda_0 x + \lambda_1 x + \lambda_2 x^2$. The problem can then be written as

$$\begin{array}{ll} \min\limits_{\lambda} & \lambda_0 + \lambda_1 E[X] + \lambda_2 E[X^2] \\ s.t. & \lambda(x) - 1 \geq 0, \ \forall x \in (-\infty, -t + E[X]] \cup [t + E[X], +\infty) \\ & \lambda(x) \geq 0, \forall x \in \mathbb{R}. \end{array} \tag{25}$$

Using [16, Theorem 3.72], this is exactly

$$\begin{array}{ll} \min\limits_{\lambda} & \lambda_0 + \lambda_1 E[X] + \lambda_2 E[X^2] \\ s.t. & \lambda(x) - 1 = \sigma(x) + \tau \cdot (x - t - E[X]), \sigma \text{ quadratic and sos, } \tau \geq 0, \\ & \lambda(x) - 1 = \sigma'(x) + \tau' \cdot (E[X] - t - x), \sigma' \text{ quadratic and sos, } \tau' \geq 0, \\ & \lambda \text{ sos}, \end{array} \tag{26}$$

which is a semidefinite program. However, given the simplicity of the case involved, it is easy to get intuition graphically as to what the correct polynomial λ should be from (25). We take

$$\lambda(x) = \left(\frac{x - E[X]}{t} \right)^2.$$

It is immediate that λ is sos and furthermore, we have

$$\lambda(x) - 1 = \left(\frac{x - t - E[X]}{t} \right)^2 + \frac{2}{t}(x - t - E[X])$$

and

$$\lambda(x) - 1 = \left(\frac{E[X] - t - x}{t} \right)^2 + \frac{2}{t}(E[X] - t - x)$$

with $2/t \geq 0$. It follows that λ is a feasible solution to (26) and hence to (25) achieving the bound of

$$\frac{var(X)}{t^2}.$$

So, $var(X)/t^2$ is always a valid upper bound on $p(|X - E[X]| > t)$. Is it tight? Again, the answer is yes, but only when $var(X) \leq t^2$. Indeed, consider

$$X_0 = \begin{cases} E[X] + t & \text{with probability } \frac{var(X)}{2t^2} \\ E[X] - t & \text{with probability } \frac{var(X)}{2t^2} \\ E[X] & \text{with probability } 1 - \frac{var(X)}{t^2} \end{cases}.$$

We have $X_0 \in \Phi$ as $E[X_0] = E[X]$ and $E[X_0^2] = E[X^2]$. Furthermore, $p(|X - E[X]| > t) = var(X)/t^2$. When $var(X) \geq t^2$, a tight upper bound is 1. It is easy to see that 1 is always a valid upper bound by taking $\lambda_0 = 1, \lambda_1 = 0, \lambda_2 = 0$. It is tight in this case as one can choose

$$X_0 = \begin{cases} E[X] + \sqrt{var(X)} & \text{with probability } 1/2 \\ E[X] - \sqrt{var(X)} & \text{with probability } 1/2 \end{cases}.$$

We have $X_0 \in \Phi$ as $E[X_0] = E[X]$ and $E[X_0^2] = E[X^2]$. Furthermore $p(|X_0 - E[X]| \geq t) = 1$. Hence a tight upper bound on $p(|X - E[X]| \geq t$ using first and second order information is given by

$$\sup_{X \in \Phi} p(|X - E[X]| \geq t) = \begin{cases} var(X)/t^2 & \text{if } var(X)/t^2 \leq 1 \\ 1 & \text{if } var(X)/t^2 \geq 1 \end{cases}.$$

The first case is the Chebychev inequality.

Applications to option pricing. Let X be the (random) price of an asset and p_X its probability distribution. Though p_X is unknown, the first and second order moments of X, which we denote by y_1 and y_2, are known. The zero-th order moment of X is trivially $y_0 = 1$. We now consider a European call option on the asset with strike price k. Recall that a European call option is a derivative security which gives the buyer of the call two options on the day it expires: either (s)he buys a fixed amount of the asset at price k, or (s)he does nothing. Hence, the payoff of the buyer of the option will be $\max(0, X - k)$ where X is the price of the asset on the day the call expires: indeed, if the price of the asset is greater than k, then the buyer will use his or her option to get it at the reduced price of k, thus making $X - k$; if the price of the asset is less than k however, then the buyer will chose to not use his or her option, thus making 0. A fair price for this option would be

$$E_{p_X}[\max(0, X - k)],$$

where the expectation is taken with respect to the unknown probability distribution of X. Note that with such a price, the seller does not make a profit on average, but simply breaks even. However, to hedge against uncertainty in the distribution of X, the seller choses to pick

$$\sup_{p_X \in \Phi} E_{p(X)}[\max(0, X - k)]$$

where Φ is as in (20) with $E = [0, +\infty)$ (the price of the asset is always nonnegative) and $K = 2$. One can then use results similar to the previous ones. The problem can be formulated as:

$$\begin{aligned} &\max_{p_X} \int_{\mathbb{R}^+} \max(0, x - k) dp_X(x) \\ &\text{s.t.} \int_{\mathbb{R}^+} x^k dp_X(x) = y_i, i = 0, 1, 2. \end{aligned}$$

Similarly to above, the dual to this problem is then

$$\begin{aligned} &\min_{\lambda_k} \sum_{k=0}^{2} \lambda_k y_k \\ &\text{s.t.} \sum_{k=0}^{2} \lambda_k x^k \geq \max(0, x - k), \forall x \in \mathbb{R}^+. \end{aligned}$$

This is equivalent to

$$\begin{aligned} &\min_{\lambda_k} \sum_{k=0}^{2} \lambda_k y_k \\ &\text{s.t.} \sum_{k=0}^{2} \lambda_k x^k \geq 0, \forall x \in [0, k] \\ &\quad \sum_{k=0}^{2} \lambda_k x^k \geq x - k, \forall x \in [k, +\infty), \end{aligned}$$

which can be solved using semidefinite programming. We refer the interested reader to [13] for other examples of problems of this type. Other areas where optimal bounds on probabilities of events can be useful are decision analysis [61] and queuing theory [67].

3 Statistics and machine learning

3.1 Shape-constrained regression

Regression is one of the most fundamental problems in statistics, with applications in many different areas, including the social and physical sciences. The input to the problem is a series of data points $\{x_i, y_i\}_{i=1,\ldots,m}$ where $x_i \in \mathbb{R}^n$ is a feature vector, and $y_i \in \mathbb{R}$ is the output variable. We will denote by x_i^j the j^{th} component of vector i and we will assume that $x_i \in B$, where B is a (full-dimensional) box in $\mathbb{R}^n$. It is assumed that there is a relationship between x_i and y_i of the form

$$y_i = f(x_i) + \epsilon_i,\ i = 1, \ldots, m$$

where ϵ_i is some random noise with $E[\epsilon_i] = 0$, finite variance, and ϵ_i independent from ϵ_j. The goal of regression is to find a function f within a class of functions $\mathcal{F}$ such that the error between $f(x_i)$ and y_i is minimized. The notion of error that is often used is that of least squares error, which gives us the problem

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{m} (y_i - f(x_i))^2, \tag{27}$$

When $\mathcal{F}$ contains functions that are completely described by a set of parameters $\theta \in \mathbb{R}^p$, the regression is called *parametric* and the optimization can be done over the parameters instead of over $\mathcal{F}$. The case where

$$\mathcal{F} = \{f \mid f(y) = \theta_0 + \theta_1 y_1 + \ldots + \theta_n y_n, \text{ where } \theta_0, \ldots, \theta_n \in \mathbb{R}\},$$

for example, is linear regression and finding f amounts to solving an unconstrained convex quadratic program.

When $\mathcal{F}$ constrains the functions f to have some specific shape (e.g., convex over the box B or monotonous in one variable over B), then we call this problem *shape-constrained* regression. Shape-constrained regression is a very natural problem. In economics for example, if one wants to model a utility function by fitting a regressor to data, then it would make sense to enforce concavity of the regressor. Likewise, we can readily imagine that a number of outputs would depend monotonically on inputs (think, e.g., of the BMI of a person with respect to his or her calorie intake, or the quantity of honey produced in a hive as a function of number of bees). Because of its omnipresence, there have been a number of methods developed to address this problem; see [28, 29, 60, 42, 47]. Here, we consider a method that relies on sum of squares programming, developed in, e.g., [45, 3]. One of its main advantages is that it scales polynomially in the number of features of the problem, which is often a caveat in other methods. We discuss it in more depth below.

Let's consider first the case where we would like to enforce monotonicity of our regressor over B with respect to component j, i.e., we want $y_j \mapsto f(y_1, \ldots, y_{j-1}, y_j, y_{j+1}, \ldots, y_n)$ to be increasing for all $(y_1, \ldots, y_{j-1}, y_{j+1}, \ldots, y_n)$ in the appropriate domain. We will assume here that $f \in C^1$. This is then equivalent to imposing that

$$\frac{\partial f(y)}{\partial y_j} \geq 0,\ \forall y \in B.$$

If $\rho \in \mathbb{R}^n$ is a vector that encodes the monotonicity profile of f with respect to each one of its variables, i.e., $\rho_j = 1$ (resp. 0, -1) if f is increasing (resp. non-monotonic, decreasing) with respect to component j, then the monotonicity-constrained regression problem can be written:

$$\begin{aligned} &\min_{f} \sum_{i=1}^{m} (y_i - f(x_i))^2 \\ &\text{s.t. } \rho_j \frac{\partial f(y)}{\partial y_j} \geq 0,\ \forall y \in B. \end{aligned}$$

To make the problem amenable to computation, we restrict ourselves to searching over the space of polynomial functions, i.e., f is assumed to be a polynomial. The problem remains hard to solve however because of the nonnegativity constraint. Indeed, one can show that even testing whether a polynomial f of degree d has monotonicity profile ρ, over a box B is NP-hard, for d as low as 3 [3]. We consequently replace the nonnegativity constraint by a constraint that involves sum of squares polynomials—see [16, Section 3.4.4] for different ways to do this—and the problem becomes a semidefinite program. The theorem below qualifies the quality of these successive approximations.

Theorem 12. *[3] Let f be a C^1 function with monotonicity profile ρ over B. For any $\epsilon > 0$, there exists an integer d and a polynomial p of degree d such that*

$$\max_{x \in B} |f(x) - p(x)| < \epsilon$$

and such that p has same monotonicity profile ρ over B. Furthermore, this monotonicity profile can be certified using a sum of squares certificate.

Let's consider now the case where we would like to enforce convexity of our regressor f over B. We assume that $f \in C^2$ and that H_f denotes the Hessian of f. This is then equivalent to imposing

$$H_f(y) \succeq 0, \forall y \in B,$$

which is in turn equivalent to

$$z^T H_f(y) z \geq 0, \ \forall z \in \mathbb{R}^n, \ y \in B.$$

Hence the convexity-constrained regression problem can be written

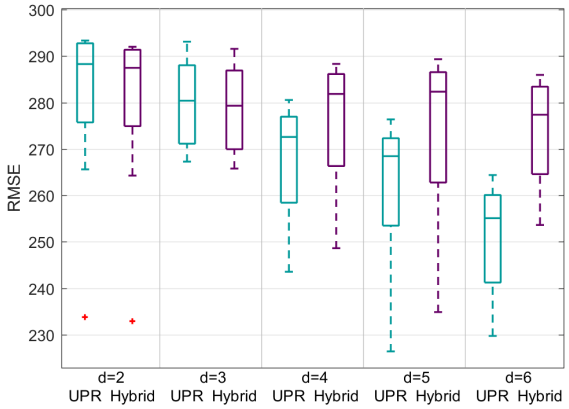
$$\begin{aligned} &\min_f \sum_{i=1}^m (y_i - f(x_i))^2 \\ &\text{s.t. } z^T H_f(y) z \geq 0, \ \forall z \in \mathbb{R}^n, \ \forall y \in B. \end{aligned}$$

We follow the same scheme as previously: we restrict ourselves to polynomial functions, and then replace the nonnegativity constraint of the polynomial (in z and y) $z^T H_f(y) z$ by a constraint that involves sum of squares polynomials. Indeed, as before, the problem of testing whether a polynomial of degree d is convex over a box is NP-hard, even for $d = 3$ [4]. One can qualify the quality of these successive approximations in an identical theorem to Theorem 12.

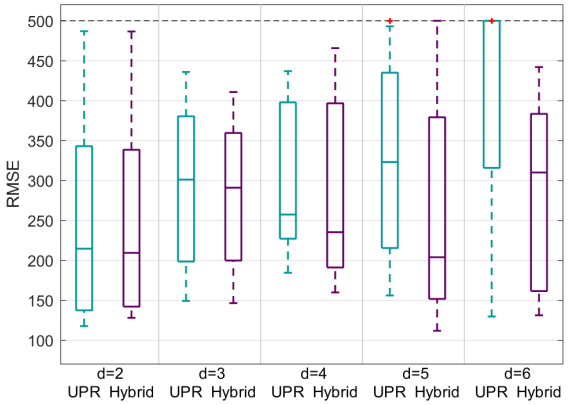
Remark 3. *Let f be a polynomial in n variables. If $y^T H_f(x) y$ is constrained to be a sum of squares (as a polynomial in x and y) then f is said to be* sos-convex*. This is a sufficient condition for (global) convexity as $y^T H_f(x) y$ sos implies that $y^T H_f(x) y \geq 0, \forall x, y \in \mathbb{R}^n$, which implies that $H_f(x) \succeq 0, \forall x \in \mathbb{R}^n$. Optimizing over the set of sos-convex polynomials is a semidefinite program; see [9] for more information on the concept of sos-convexity. This example uses a variant of sos-convexity: we wish to find sufficient conditions for convexity* over a box.

Remark 4. *It goes without saying that both types of constraints (monotonicity and convexity) can be combined if one happens to have the appropriate information.*

Example 6. *We now give an example, taken from [3], relating to the prediction of weekly wages from past data. The data used comes from the 1988 Current Population Survey and is freely available under the name ex1029 in the Sleuth2 R package [58]. It contains* 25361 *observations and* 2 *numerical features: years of experience and years of education. We expect wages to increase with respect to years of education and be concave with respect to years of experience. We run both an unconstrained polynomial regression (denoted by UPR), i.e., $\mathcal{F}$ is the set of polynomials of a certain degree in (27), and a convexity-constrained and monotonocity-constrained regression (denoted by Hybrid and described above) on the data. This is done by computing the Root Mean Squared Error (RMSE) for the data with 10-fold cross validation. The results are given in Figure 5 with varying degrees of the polynomial regressor. Note that for the training data, obviously UPR performs better than Hybrid as it is less constrained and can overfit. The Hybrid method however has a much better generalization error than UPR.*



(a) Values taken by the RMSE on training data



(b) Values taken by the RMSE on testing data

Figure 5: Comparative performance of UPR and Hybring on testing and training sets for 10 fold cross validation.

3.2 Optimal design

We once again consider a regression setting, but this time we are interested in the problem of generating data. Recall that the input to a regression problem are pairs $\{x_i, y_i\}_{i=1,\dots,m}$, where $x_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}$. Statisticians make a difference between the case where the person conducting the study can choose the feature vectors $\{x_i\}_i$, and the case where the feature vectors $\{x_i\}_i$ are imposed. The latter case is called an *observational study*. An illustrative example is that of studying the impact of the amount of cigarettes smoked on the development of lung cancer: our data will contain the amount of cigarettes that each participant chooses to smoke, without our being able to impact this. Indeed, it would be a major ethical breach if we were asking participants to smoke more, e.g., to change our input data.

Of interest in this section is the other case, namely the case where the x_i can be fixed to certain values by the experimenter. This is called an *experimental study*. As an example of such a study, consider the problem of measuring the degree of corrosion of steal under the effects of humidity and temperature. By placing a piece of steal in an environment controlled for humidity and temperature, one is able to obtain the degree of corrosion for any values of humidity and temperature that one wishes to have. This set-up is particularly interesting to statisticians as it enables the experimenter to choose advantageous values of the features. The process of choosing such values is termed *experimental design*. In this section, we will be interested in using sum of squares techniques to understand how to design experiments in an optimal way. We follow the presentation given in [24].

As defined in 2.1, we use the terminology $\mathbb{N}^n_d = \{\alpha \in \mathbb{N}^n \mid \alpha_1 + \ldots + \alpha_n \leq d\}$. For $x = (x_1, \ldots, x_n)$, we also use the shorthand x^α to mean $x_1^{\alpha_1} \ldots x_n^{\alpha_n}$. We will consider a parametric regression setting here, more specifically, a polynomial one, i.e.,

$$y_i = \sum_{\alpha \in \mathbb{N}^n_d} \theta_\alpha x_i^\alpha + \epsilon_i, \;\; i = 1, \ldots, m$$

where $\{\theta_\alpha\}$ are the coefficients of the polynomial, and ϵ_i is random noise with $E[\epsilon_i] = 0$, $var(\epsilon_i) = \sigma^2 < \infty$, and ϵ_i independent from ϵ_j. We assume that $m \geq \binom{n+d}{n}$ and that the x_i can be picked within a compact set $\mathcal{X}$, described by a finite number of polynomial inequalities. Hence, our goal is to come up with points $t_k \in \mathcal{X}$ where $k = 1, \ldots, l$ with $l \leq m$, and a number of times n_k that the values $\{x_i\}$ take value t_k. This information is summarized in a *design matrix*

$$\xi = \begin{bmatrix} t_1 & \ldots & t_l \\ w_1 & \ldots & w_l \end{bmatrix}, \text{ where } w_k = \frac{n_k}{m}, \tag{28}$$

which is what we would like to obtain at the end of the process. In the rest of this paragraph, for convenience, we will denote the standard vector of monomials of degree up to d and in n vari-

ables by $z(x) = (1, x_1, x_2, \ldots, x_n, \ldots, x_n^d)$, and by θ the corresponding vector of coefficients, so that $\sum_{\alpha \in \mathbb{N}_{2d}^n} \theta_\alpha x_i^\alpha = \theta^T z(x)$.

What should be the objective when picking ξ? This depends on what we would like to achieve. In our case, assuming our estimator for θ is the least squares estimator[3]

$$\hat{\theta} = \arg\min_\theta \sum_{k=1}^m (y_k - \theta^T z(t_k))^2,$$

it may be of interest to minimize, in some sense, the variance of $\hat{\theta}$. Indeed, as will be made evident later on, under the assumptions we have on ϵ_i, $\hat{\theta}$ is an unbiased estimator of θ, which means that on average, they are equal. It may then be of interest to ask that $\hat{\theta}$ deviate as little as possible from θ on average: this is exactly equivalent to minimizing the variance of $\hat{\theta}$. Of course, the variance of $\hat{\theta}$ is here a matrix as $\hat{\theta}$ is a vector, so when we claim to minimize the variance of $\hat{\theta}$, we actually mean minimizing its 2-norm, or some other measure. Let us now compute $var(\hat{\theta})$. Some quick algebra gives us that

$$\hat{\theta} = \sum_{k=1}^m y_k \left(\sum_{k=1}^m z(t_k) z(t_k)^T \right)^{-1} z(t_k).$$

Using the fact that $y_k = \theta^T z(x_k) + \epsilon_k$, we get

$$\hat{\theta} = \sum_{k=1}^m \left(\sum_{k=1}^m z(t_k) z(t_k)^T \right)^{-1} (\theta^T z(t_k) + \epsilon_k) z(t_k) = \theta + \left(\sum_{k=1}^m z(t_k) z(t_k)^T \right)^{-1} \sum_{k=1}^m \epsilon_k z(t_k).$$

It then follows that the variance matrix of $\hat{\theta}$ is given by

$$\Sigma(\xi) = \sigma^2 \left(\sum_{k=1}^m z(t_k) z(t_k)^T \right)^{-1}$$

where we have used the facts that $E[\epsilon_k] = 0, \forall k$ and independence of the $\{\epsilon_k\}$. In the more general case where the points t_k are not assumed distinct, the variance matrix is given by

$$\Sigma(\xi) = \sigma^2 \left(\sum_{k=1}^l n_k z(t_k) z(t_k)^T \right)^{-1}.$$

In the rest of this paragraph, we will consider the case where we would like to minimize the 2-norm of the matrix $\Sigma(\xi)$. This is equivalent to minimizing its largest eigenvalue, or if we define the following quantity,

$$F(\xi) := \sum_{k=1}^l w_k z(t_k) z(t_k)^T,$$

it is equivalent to maximizing the minimum eigenvalue of $F(\xi)$. The matrix $F(\xi)$ is a well-known quantity in statistics called the *Fisher information matrix* of the design ξ and maximizing its minimum eigenvalue corresponds to a common notion of optimality in experimental design, that of *E-optimality*. There are many different ways to define optimality, based essentially on minimizing various norms of $\Sigma(\xi)$; we refer the interested reader to [24] for more information on this topic. As mentioned before, the problem of interest here is

$$\max_\xi \lambda_{\min} F(\xi)$$
$$\text{s.t. } \xi \text{ is as in (28).}$$

[3]For clarity of exposition, we suppose for the moment that all t_k are distinct.

This can be rewritten as

$$\begin{aligned} &\max_{n_k,t_k,\gamma} \gamma \\ &\text{s.t. } \sum_{k=1}^{l} \frac{n_k}{m} z_d(t_k) z_d(t_k)^T \succeq \gamma I \\ &\qquad n_k \in \mathbb{N}, \sum_{k=1}^{l} n_k = m. \end{aligned}$$

where I is the identity matrix. We first drop the constraint that $n_k \in \mathbb{N}$, relaxing it to $n_k \geq 0$, and the problem becomes:

$$\begin{aligned} &\max_{w_k,t_k,\gamma} \gamma \\ &\text{s.t. } \sum_{k=1}^{l} w_k z_d(t_k) z_d(t_k)^T \succeq \gamma I \\ &\qquad w_k \geq 0, \sum_{k=1}^{l} w_k = 1. \end{aligned} \tag{29}$$

The matrix $\sum_{k=1}^{l} w_k z_d(t_k) z_d(t_k)^T$ is of size $\mathbb{N}_d^n \times \mathbb{N}_d^n$. We index it by (α, β) where $\alpha, \beta \in \mathbb{N}_d^n$. Note that entry (α, β) of the matrix is exactly

$$\int_{\mathcal{X}} x^\alpha x^\beta d\mu$$

where μ is the Dirac measure given by $\mu(x) = \sum_{k=1}^{l} w_k \delta_{x=t_i}(x)$. In other words, entry (α, β) of the matrix is the $\alpha + \beta$ moment of μ. Define

$$y_\alpha := \int_{\mathcal{X}} x^\alpha d\mu, \alpha \in \mathbb{N}_{2d}^n,$$

for some measure μ over $\mathcal{X}$ and let $M(y)$ be the $\mathbb{N}_d^n \times \mathbb{N}_d^n$ matrix with entry (α, β) given by $y_{\alpha+\beta}$. It follows that (29) can be rewritten as:

$$\begin{aligned} &\max_{w_k,t_k,\gamma,y} \gamma \\ &\text{s.t. } M(y) \succeq \gamma I \\ &\qquad y_0 = 1, y_\alpha = \int_{\mathcal{X}} x^\alpha d\mu \text{ for } \alpha \in \mathbb{N}_{2d}^n, \text{ and } \mu = \sum_{k=1}^{l} w_k \delta_{x=t_i}(x) \end{aligned} \tag{30}$$

Similarly to (14), we now define the following set:

$$\mathcal{M}_{2d}(\mathcal{X}) := \{\{y_\alpha\}_{\alpha \in \mathbb{N}_{2d}^n} \mid \exists \text{ a measure } \mu \text{ on } \mathcal{X} \text{ such that } y_\alpha = \int_{\mathcal{X}} x^\alpha d\mu, \ \forall \alpha \in \mathbb{N}_{2d}^n\}. \tag{31}$$

Note that contrarily to (14), we are considering measures over $\mathcal{X}$ and not over $\mathbb{R}^n$. Hence, the dual of this set is the set of polynomials p with coefficients $\{p_\alpha\}_{\alpha \in \mathbb{N}_{2d}^n}$ such that p is nonnegative *over* $\mathcal{X}$, and not over $\mathbb{R}^n$ as previously; see [40] for a proof. Using this definition, we are able to further relax (30) to a problem that only depends on γ and y:

$$\begin{aligned} &\max_{\gamma,y} \gamma \\ &\text{s.t. } M(y) \succeq \gamma I \\ &\qquad y_0 = 1, y \in \mathcal{M}_{2d}(\mathcal{X}). \end{aligned} \tag{32}$$

Proposition 2. *The dual to (32) is given by*

$$\begin{aligned} &\min_{\lambda \in \mathbb{R}, Q \in \mathbb{N}_d^n \times \mathbb{N}_d^n} \lambda \\ &\text{s.t. } \lambda - z_d(x)^T Q z_d(x) \geq 0, \forall x \in \mathcal{X} \\ &\qquad tr(Q) = 1, Q \succeq 0, \end{aligned} \tag{33}$$

where $z_d(x) = (1, x_1, \ldots, x_n, \ldots, x_n^d)$ is the standard vector of monomials.

Proof. Let λ, Q be feasible for (33) and let γ, y be feasible for (32). As $Q \succeq 0$, there exists a matrix V such that $Q = VV^T$. Furthermore, as $tr(Q) = 1$, then $tr(VV^T) = tr(V^TV) = 1$. Together with the fact that $M(y) \succeq \gamma I$, this implies that $V^TM(y)V \succeq \gamma V^TV$, and in particular,

$$tr(V^TM(y)V) \geq tr(\gamma V^TV) = \gamma tr(V^TV) = \gamma.$$

We have

$$\lambda - \gamma \geq \lambda - tr(V^TM(y)V) = \lambda - tr(M(y)Q) = \lambda - \sum_{\alpha,\beta} M_{\alpha,\beta}(y)Q_{\alpha,\beta}$$

Recall that $M_{\alpha,\beta}(y) = y_{\alpha+\beta}$. As y has a representing measure, it follows that $M_{\alpha,\beta}(y) = \int_{\mathcal{X}} x^{\alpha+\beta}d\mu$. Hence,

$$\lambda - \gamma \geq \lambda - \int_{\mathcal{X}} \sum_{\alpha,\beta} Q_{\alpha,\beta}x^\alpha x^\beta d\mu = \int_{\mathcal{X}} \lambda - z_d^T(x)Qz_d(x)d\mu,$$

where we have used the fact that $y_0 = 1$ in the equality. As $\lambda - z_d^T(x)Qz_d(x) \geq 0$, for all $x \in \mathcal{X}$, we deduce that $\lambda - \gamma \geq 0$. □

Strong duality holds under certain conditions, see [40]. As is, (33) cannot be solved. However, if one replaces the condition that $\lambda - z_d(x)^TQz_d(x)$ be nonnegative over $\mathcal{X}$ by certificates of nonnegativity of the polynomial over $\mathcal{X}$ involving sum of squares polynomials, then the problem becomes a semidefinite program. One can proceed similarly in the primal (32) by relying on outer-approximations to the set $\mathcal{M}_{2d}(\mathcal{X})$.

4 Conclusion: a word on implementation challenges

A topic that is central to applications but that we have barely touched upon so far is how to implement these methods in practice. In particular, what software should we use to solve sum of squares programs? There are two components here to consider: which semidefinite programming solver to use and which parser to use which converts the sum of squares program to a semidefinite program. Indeed, in theory, one could manually convert the sum of squares program at hand into a semidefinite program, but this is generally not considered to be an enjoyable task. It is consequently much more convenient to use a parser whose role is to automate this process. Note however that not all parsers interface with all solvers, and that one need sometimes access parsers or solvers within other software or interfaces (e.g., MATLAB).

We start by reviewing a few semidefinite programming solvers (the list is by no means meant to be exhaustive). Choosing a good-quality solver is a crucial step in coming up with robust solutions to the problem at hand in a reasonable amount of time. MOSEK [1], SDPT3 [63], Sedumi [62], and SDPA [68] are established solvers, with the first being a commercial solver (free with an academic license), and the latter three being free. SDPA interfaces with Python, MOSEK interfaces with C, Java, Python, and MATLAB, and Sedumi and SDPT3 interface with MATLAB. All of these solvers rely on interior point methods, which unfortunately do not always scale very well with the size of the problem. In light of this, new solvers have been developed which rely on augmented Lagrangian methods instead, such as SDPNAL/SDPNAL+ [69], CDCS [70], and SCS [50]. These are all free. Note that the first uses Newton-Conjugate Gradient augmented Lagrangian, whereas the latter two use ADMM. Furthermore SDPNAL/SDPNAL+ and CDCS can be accessed from MATLAB, whereas SCS is written in C and can be used in other C, C++, Python environments, as well as in MATLAB, R and Julia.

In terms of parsers, the most commonly used are perhaps YALMIP [43], Julia [15], SOSTOOLS [57], SPOT [48], Gloptipoly [31], and Macaulay2 [27]. YALMIP is a toolbox for modelling and optimization in MATLAB that has a special sos module. It can be interfaced with many solvers including MOSEK, SDPT3 and Sedumi. Julia is an open-source dynamic programming language for technical computing. Optimization is done via its modeling package JuMP [25] and sum of squares problems can be tackled via one of its packages SumofSquares.jl [23]. SOSTOOLS is a free MATLAB toolbox for sos programs. It can be interfaced with a number of solvers including SeDuMi, SDPT3, CSDP, SDPNAL, SDPNAL+, CDCS and SDPA. SPOT can be viewed as an alternative to SOSTOOLS as it is also a MATLAB toolbox for sos programs, but its focus is towards control theory. GloptiPoly is a free MATLAB toolbox, which solves

what is known as the *generalized moment problem*. This includes the moment problem as described in Section 2.1, and hence can be used to tackle polynomial optimization problems. It can be interfaced with many different solvers including Sedumi, SDPT3 and MOSEK. Finally, Macaulay2 is a free computer algebra system geared towards research in algebraic geometry. However, via a package, it can be used to solve sum of squares programs.

As mentioned above, the direction currently taken in solver development involves replacing interior point methods by methods that can robustly solve very large problems, such as ADMM. This is due to the fact that the size of the semidefinite program generated by a sum of squares program is of order n^d when the polynomials considered in the sos program are of degree $2d$ and in n variables. This limited ability to solve very large sos programs has been one of the main impediments in further disseminating sum of squares techniques. Indeed, possible new applications often feature problems of large scale. This has consequently led to a flurry of research around the question: how can we make solving sos programs more scalable? One such step of course is to construct new solvers for semidefinite programs that rely on more scalable algorithms as we saw above. Another research direction, complementary to this one, is to leverage the structure of the semidefinite program at hand to reduce its size. Structures of interest can include e.g. symmetries in the problem or sparsity; see [26, 64, 65] for some of these directions. A very different research direction involves replacing the semidefinite program at hand by cheaper conic programs with trade-offs in accuracy; see [7, 66] for some examples of this direction. The hope is that by combining these different research directions, one will be able to tackle large-scale sos programs and open up many new areas to the use of sos programming.

References

[1] *MOSEK reference manual*, 2013. Version 7. Latest version available at `http://www.mosek.com/`.

[2] A. A. Ahmadi. Non-monotonic Lyapunov functions for stability of nonlinear and switched systems: theory and computation. Master's thesis, Massachusetts Institute of Technology, June 2008. Available at `http://aaa.lids.mit.edu/publications`.

[3] A. A. Ahmadi, M. Curmei, and G. Hall. Shape-constrained regression and nonnegative polynomials. *In preparation*, 2019.

[4] A. A. Ahmadi and G. Hall. On the complexity of detecting convexity over a box. *arXiv preprint arXiv:1806.06173*, 2018.

[5] A. A. Ahmadi, R. Jungers, P. A. Parrilo, and M. Roozbehani. Joint spectral radius and path-complete graph Lyapunov functions. *SIAM Journal on Optimization and Control*, 2013. To appear.

[6] A. A. Ahmadi, M. Krstic, and P. A. Parrilo. A globally asymptotically stable polynomial vector field with no polynomial Lyapunov function. In *Proceedings of the 50th IEEE Conference on Decision and Control*, 2011.

[7] A. A. Ahmadi and A. Majumdar. DSOS and SDSOS optimization: more tractable alternatives to sum of squares and semidefinite optimization. *arXiv preprint arXiv:1706.02586*, 2017.

[8] A. A. Ahmadi and P. A. Parrilo. Converse results on existence of sum of squares Lyapunov functions. In *Proceedings of the 50th IEEE Conference on Decision and Control*, 2011.

[9] A. A. Ahmadi and P. A. Parrilo. A complete characterization of the gap between convexity and sos-convexity. *SIAM Journal on Optimization*, 23(2):811–833, 2013. Also available at arXiv:1111.4587.

[10] P. J. Antsaklis and A. N. Michel. *Linear Systems*. Birkhäuser, Boston, MA, 2006.

[11] A. Bacciotti and L. Rosier. *Liapunov Functions and Stability in Control Theory*. Springer, 2005.

[12] B. Barak and D. Steurer. Sum-of-squares proofs and the quest toward optimal algorithms. *arXiv preprint arXiv:1404.5236*, 2014.

[13] D. Bertsimas and I. Popescu. On the relation between option and stock prices: a convex optimization approach. *Operations Research*, 50(2):358–374, 2002.

[14] D. Bertsimas and I. Popescu. Optimal inequalities in probability theory: A convex optimization approach. *SIAM Journal on Optimization*, 15(3):780–804, 2005.

[15] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.

[16] G. Blekherman, P. A. Parrilo, and R. Thomas. *Semidefinite optimization and convex algebraic geometry*. SIAM Series on Optimization, 2013.

[17] V. D. Blondel. The birth of the joint spectral radius: An interview with gilbert strang. *Linear Algebra and its Applications*, 428(10):2261–2264, 2008.

[18] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 2996–3000. IEEE, 2005.

[19] V. D. Blondel and Y. Nesterov. Polynomial-time computation of the joint spectral radius for some sets of nonnegative matrices. *SIAM Journal on Matrix Analysis and Applications*, 31(3):865–876, 2009.

[20] V. D. Blondel and J. N. Tsitsiklis. The boundedness of all products of a pair of matrices is undecidable. *Systems and Control Letters*, 41:135–140, 2000.

[21] V. D. Blondel and J. N. Tsitsiklis. A survey of computational complexity results in systems and control. *Automatica*, 36(9):1249–1274, 2000.

[22] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, 1994.

[23] R. D. Brackston, A. Wynn, and M. P. H. Stumpf. Construction of quasi-potentials for stochastic dynamical systems: an optimization approach. *arXiv preprint arXiv:1805.07273*, 2018.

[24] Y. De Castro, F. Gamboa, D. Henrion, R. Hess, and J.-B. Lasserre. Approximate optimal designs for multivariate polynomial regression. *arXiv preprint arXiv:1706.04059*, 2017.

[25] I. Dunning, J. Huchette, and M. Lubin. JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.

[26] K. Gatermann and P. A. Parrilo. Symmetry groups, semidefinite programs, and sums of squares. *Journal of Pure and Applied Algebra*, 192:95–128, 2004.

[27] D. R. Grayson and M. E. Stillman. Macaulay2, a software system for research in algebraic geometry. Available at `http://www.math.uiuc.edu/Macaulay2/`.

[28] M. R. Gupta, A. Cotter, J. Pfeifer, K. Voevodski, K. Canini, A. Mangylov, W. Moczydlowski, and A. Van Esbroeck. Monotonic calibrated interpolated look-up tables. *Journal of Machine Learning Research*, 17(109):1–47, 2016.

[29] L. A. Hannah and D. B. Dunson. Multivariate convex regression with adaptive partitioning. *The Journal of Machine Learning Research*, 14(1):3261–3294, 2013.

[30] D. Henrion and A. Garulli, editors. *Positive polynomials in control*, volume 312 of *Lecture Notes in Control and Information Sciences*. Springer, 2005.

[31] D. Henrion, J.-B. Lasserre, and J. Löfberg. Gloptipoly 3: moments, optimization and semidefinite programming. *Optimization Methods & Software*, 24(4-5):761–779, 2009.

[32] Z. Jarvis-Wloszek, R. Feeley, W. Tan, K. Sun, and A. Packard. Some controls applications of sum of squares programming. In *Proceedings of the 42th IEEE Conference on Decision and Control*, pages 4676–4681, 2003.

[33] R. Jungers. *The joint spectral radius: theory and applications*, volume 385 of *Lecture Notes in Control and Information Sciences*. Springer, 2009.

[34] H. Khalil. *Nonlinear systems*. Prentice Hall, 2002. Third edition.

[35] M. Krstic, I. Kanellakopoulos, P. V. Kokotovic, et al. *Nonlinear and adaptive control design*, volume 222. Wiley New York, 1995.

[36] Y Kurzweil. On the inversion of the second theorem of lyapunov on stability of motion. *Czechoslovak Math. J*, 81(6):217–259, 1956.

[37] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[38] J. B. et al. Lasserre. Bounds on measures satisfying moment conditions. *The Annals of Applied Probability*, 12(3):1114–1137, 2002.

[39] M. Laurent. Sums of squares, moment matrices and optimization over polynomials. In *Emerging applications of algebraic geometry*, pages 157–270. Springer, 2009.

[40] M. Laurent. Sums of squares, moment matrices and optimization over polynomials. In *Emerging applications of algebraic geometry*, pages 157–270. Springer, 2009.

[41] B. Legat, R. M. Jungers, and P. A. Parrilo. Generating unstable trajectories for switched systems via dual sum-of-squares techniques. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, pages 51–60. ACM, 2016.

[42] E. Lim and P. W. Glynn. Consistency of multidimensional convex regression. *Operations Research*, 60(1):196–208, 2012.

[43] J. Löfberg. Pre- and post-processing sum-of-squares programs in practice. *IEEE Transactions on Automatic Control*, 54(5):1007–1011, 2009.

[44] A. M. Lyapunov. *General problem of the stability of motion.* PhD thesis, Kharkov Mathematical Society, 1892. In Russian.

[45] A. Magnani, S. Lall, and S. Boyd. Tractable fitting with convex polynomials via sum-of-squares. *IEEE Conference on Decision and Control and European Control Conference*, 2005.

[46] A. Majumdar, A. A. Ahmadi, and R. Tedrake. Control design along trajectories with sums of squares programming. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2013.

[47] R. Mazumder, A. Choudhury, G. Iyengar, and B. Sen. A computational framework for multivariate convex regression and its variants. *Journal of the American Statistical Association*, 2017.

[48] A. Megretski. SPOT: systems polynomial optimization tools. 2013.

[49] K. G. Murty and S. N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39:117–129, 1987.

[50] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, 2016.

[51] A. Papachristodoulou and S. Prajna. On the construction of Lyapunov functions using the sum of squares decomposition. In *IEEE Conference on Decision and Control*, 2002.

[52] A. Papachristodoulou and S. Prajna. A tutorial on sum of squares techniques for systems analysis. In *American Control Conference, 2005. Proceedings of the 2005*, pages 2686–2700. IEEE, 2005.

[53] P. A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization.* PhD thesis, Citeseer, 2000.

[54] P. A. Parrilo and A. Jadbabaie. Approximation of the joint spectral radius using sum of squares. *Linear Algebra Appl.*, 428(10):2385–2402, 2008.

[55] M. M. Peet. Exponentially stable nonlinear systems have polynomial Lyapunov functions on bounded regions. *IEEE Trans. Automat. Control*, 54(5):979–987, 2009.

[56] S. Prajna and A. Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *Hybrid Systems: Computation and Control*, pages 477–492. Springer, 2004.

[57] S. Prajna, A. Papachristodoulou, and P. A. Parrilo. *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, 2002-05. Available from `http://www.cds.caltech.edu/sostools` and `http://www.mit.edu/~parrilo/sostools`.

[58] F.L. Ramsey and D.W. Schafer. *Sleuth2: Data Sets from Ramsey and Schafer's "Statistical Sleuth (2nd Ed)"*, 2016. R package version 2.0-4.

[59] G. C. Rota and W. G. Strang. A note on the joint spectral radius. *Indag. Math.*, 22:379–381, 1960.

[60] E. Seijo, B. Sen, et al. Nonparametric least squares estimation of a multivariate convex regression function. *The Annals of Statistics*, 39(3):1633–1657, 2011.

[61] J. E. Smith. Generalized chebychev inequalities: theory and applications in decision analysis. *Operations Research*, 43(5):807–825, 1995.

[62] J. Sturm. *SeDuMi version 1.05*, October 2001. Latest version available at `http://sedumi.ie.lehigh.edu/`.

[63] K. C. Toh, R. H. Tütüncü, and M. J. Todd. *SDPT3 - a MATLAB software package for semidefinite-quadratic-linear programming.* Available from `http://www.math.cmu.edu/~reha/sdpt3.html`.

[64] F. Vallentin. Symmetry in semidefinite programs. *Linear Algebra and its Applications*, 430(1):360–369, 2009.

[65] H. Waki, S. Kim, M. Kojima, and M. Muramatsu. Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. *SIAM Journal on Optimization*, 17(1):218–242, 2006.

[66] T. Weisser, J. B. Lasserre, and K.-C. Toh. Sparse-BSOS: a bounded degree SOS hierarchy for large scale polynomial optimization with sparsity. *Mathematical Programming Computation*, 10(1):1–32, 2018.

[67] W. Whitt. On approximations for queues, i: Extremal distributions. *AT&T Bell Laboratories Technical Journal*, 63(1):115–138, 1984.

[68] M. Yamashita, K. Fujisawa, and M. Kojima. Implementation and evaluation of sdpa 6.0 (semidefinite programming algorithm 6.0). *Optimization Methods and Software*, 18(4):491–505, 2003.

[69] L. Yang, D. Sun, and K.-C. Toh. Sdpnal +: a majorized semismooth newton-cg augmented lagrangian method for semidefinite programming with nonnegative constraints. *Mathematical Programming Computation*, 7(3):331–366, 2015.

[70] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Goulart, and A. Wynn. Chordal decomposition in operator-splitting methods for sparse semidefinite programs. *arXiv preprint arXiv:1707.05058*, 2017.