Algebraic and Semi-Algebraic Reasoning For Formal Methods

Lecture 6 - Positivstellensatz and Applications.

Sriram Sankaranarayanan

Proving Entailments

$$(\forall x, y \in \mathbb{R}) \quad x^2 + y^2 \leq 1 \ \wedge \ x + y \leq 0 \ \Rightarrow \ y \leq 1.423$$

Why?

$$(1.423 - y) = \left(\begin{array}{c} 0.765134 \ \ (1 - x^2 - y^2) \ + \\ 0.4 \quad - (x + y) \ + \\ 0.6574 - 0.6x + 0.4y + 0.765134(x^2 + y^2) \end{array} \right)$$

Positivstellensatz

Consider entailment over $\mathbb{R}^n$:

$$p_1 \geq 0 \ \wedge \ \cdots \ \wedge \ p_m \geq 0 \ \models \ p \geq 0$$

- One approach is to try to convert to real Nullstellensatz.

Inequalities to Equalities

Consider polynomials over $p \in \mathbb{R}[x_1, \ldots, x_n]$.

- Let t be a fresh variable.
- $p \geq 0 \;\Leftrightarrow\; p = t^2$
- $p > 0 \;\Leftrightarrow\; t^2 p = 1$
- $p \neq 0 \;\Leftrightarrow\; tp = 1$

Convert entailment back to equalities.

Real Varieties

Given $p_1, \dots, p_m \in \mathbb{R}[x_1, \dots, x_n]$, define the *real variety* as

$$V = \{\vec{x} \in \mathbb{R}^n \mid \bigwedge_{j=1}^{m} p_j = 0\}$$

- We already saw the importance of algebraic closure.
  - Real variety of $1 + x^2 = 0$.

Theorem $V = \emptyset$ if and only if $1 + \sigma \in \langle p_1, \dots, p_m \rangle$, where σ is SOS.

Positivestellensatz

Let $S = \{\vec{x} \in \mathbb{R}^n \mid p_1(\vec{x}) \geq 0 \ \wedge \ \cdots \ \wedge \ p_m(\vec{x}) \geq 0\}$.

- We wish to show that $p \geq 0$ on S for given p.
- **Important:** We will need S to be compact.

Schmugden's Positivstellensatz

Enrich the set of polynomials

$$Q(S) = \{p_1^{e_1} \cdots p_m^{e_m} \mid e_i \in \{0, 1\}\}$$

Note: $|Q(S)| = 2^m$.

Theorem (Schmugden'1991)

- If $p = \sum_{q \in Q(S)} \sigma_q q$ for σ_q SOS then $p_1(\vec{x}) \geq 0 \ \wedge \ \cdots \ \wedge \ p_m(\vec{x}) \geq 0 \models p \geq 0$.
- Conversely, if $p_1(\vec{x}) \geq 0 \ \wedge \ \cdots \ \wedge \ p_m(\vec{x}) \geq 0 \models p > 0$ then $p = \sum_{q \in Q(S)} \sigma_q q$ for σ_q SOS.

Putinar's Positivstellensatz

Let $M = \{\sum_{j=1}^{m} \sigma_j p_j + \sigma_0 \mid \sigma_0, \ldots, \sigma_m \text{ SOS}\}$.

- **Archimedean Property:** There exists a K such that

$$K - (x_1^2 + \cdots + x_n^2) \in M$$

- If time permits, explain connection to Archimedes.

Theorem (Putinar'1993)

- If $p \in M$ then $p_1 \geq 0 \ \wedge \ \cdots \wedge \ p_m \geq 0 \ \models \ p \geq 0$.
- If S compact and M is Archimedean, then $p_1 \geq 0 \ \wedge \ \cdots \ \wedge \ p_m \geq 0 \models p > 0$ then $p \in M$.

Positivstellensatz to Semi-Definite Programming

Problem: prove the following entailment.

$$p_1 \geq 0 \ \wedge \ \cdots \ \wedge \ p_m \geq 0 \ \models \ p \geq 0$$

Strategy: Find, $\sigma_0, \ldots, \sigma_m$ such that

$$p = \sigma_0 + \sum_{j=1}^{m} \sigma_j p_j, \text{ and } \sigma_j \text{ SOS}$$

- Bound the degrees of $\sigma_0, \ldots, \sigma_m \in \mathbb{R}_{2d}[\vec{x}]$.

Reduction to SDP

- Fix a basis of monomials $\mu(\vec{x})$.
- $\sigma_i = \mu^t X_i \mu$
- $p = \sigma_0 + \sum_{j=1}^{m} \sigma_j p_j$
  - Equate monomials on LHS and RHS.
  - $\sum_{j=0}^{m} (P_{i,j}, X_j) = c_i$
- Place $X_1, \ldots, X_n$ in a block diagonal form.

$$X = \begin{bmatrix} X_1 & 0 & \cdots & 0 \\ 0 & X_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & X_n \end{bmatrix}$$

Sum Of Squares: Difficulties

- Positivstellensatz tend to yield SDP instances that often fail *strict feasibility*.
- Find a polynomial $p \in \mathbb{R}[x, y]$ such that
  - $p(\vec{0}) = 0$.
  - p is SOS.
  - $(x^2 + y^2 \leq 1) \vDash p \leq 1$

$$p = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2$$

- $p(0) = 0$ means that $a_0 = 0$.
- p must be SOS.
  - However, this means that $a_1 = 0, a_2 = 0$ also follow immediately.
  - If we do not recognize this, the SDP instance will no longer have strict feasibility.
- Conversion from SOS to SDP requires pre-processing steps.
  - Reference: Löfberg, Pre- and Post-Processing Sum-of-Squares Programs in Practice.

Differential Equation Models

Modeling continuously varying quantities (pressure, temperature, nutrient flow, ..):

Example

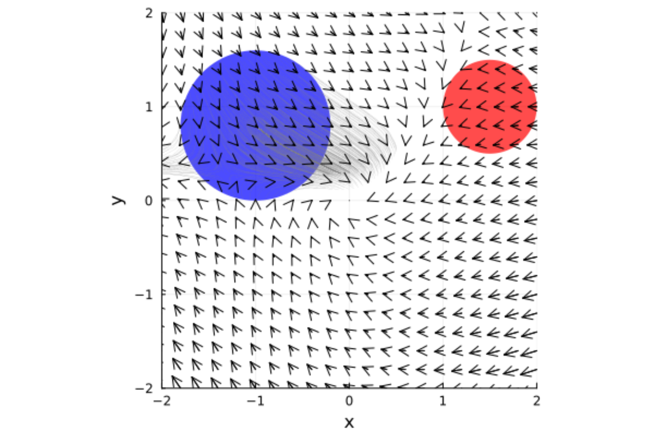
$$\frac{dx}{dt} = -0.2x^2 - 0.2x + 0.3y$$
$$\frac{dy}{dt} = -0.1x - 0.2y + 0.2xy$$

- $\frac{dx_1}{dt} = f_1(x_1, \ldots, x_n), \ldots, \frac{dx_n}{dt} = f_n(x_1, \ldots, x_n)$.
  - $\frac{d\vec{x}}{dt} = f(\vec{x})$

Solution

- Initial condition: $x_1(0), \ldots, x_n(0)$.
- Function: $\tau : \mathbb{R} \to \mathbb{R}^n$.
    - $\tau(0) = (x_1(0), \ldots, x_n(0))$
    - $\forall\ t,\ \frac{d\tau}{dt} = f(\tau(t))$

Proving Safety Properties



Prove If $\vec{x}(0)$ in I, then U is never reached.

Barrier Functions

- Find a Barrier Function $B(x_1, \dots, x_n)$ for $\epsilon > 0$.
- $\vec{x} \in I \ \models \ B \geq \epsilon$
- $\vec{x} \in I \ \models \ B \leq -\epsilon$
- $B(\vec{x}) = 0 \ \models \ \underbrace{(\nabla B) \cdot f \geq \epsilon}_{\text{Lie Derivative}}$

Explanation in lecture.

Barrier Functions Synthesis

Inputs:

- ODE model: $\frac{d\vec{x}}{dt} = f(\vec{x})$
- Initial Set: $g_1(\vec{x}) \geq 0 \ \wedge \ \cdots \ \wedge \ g_k(\vec{x}) \geq 0$
- Unsafe Set: $h_1(\vec{x}) \geq 0 \ \wedge \ \cdots \ \wedge \ h_m(\vec{x}) \geq 0$

Goal: Find a barrier given a *template* (*ansatz*).

$$\sum_{\alpha} c_\alpha \vec{x}^\alpha$$

Ref. Prajna and Jadbabaie, HSCC 2004. Also see monograph: "Positive Polynomials in Control".

Barrier Function Synthesis

Use positivstellensatz to encode conditions on unknown c_α.

- $g_1(\vec{x}) \geq 0 \ \wedge \ \cdots \ \wedge \ g_k(\vec{x}) \geq 0 \ \models \ B(\vec{x}; c) \geq \epsilon$
- $h_1(\vec{x}) \geq 0 \ \wedge \ \cdots \ \wedge \ h_m(\vec{x}) \geq 0 \ \models \ \epsilon - B(\vec{x}; c) \geq 0$

Encoding the barrier condition:

$$B(\vec{x}) = 0 \ \models \ (\nabla B) \cdot f \geq \epsilon$$

Yields a *bilinear* SDP (much harder to solve).

Barrier Function Synthesis (Continued)

- Exponential Barrier Condition

$$\forall\ \vec{x} \in \mathbb{R}^n,\ (\nabla B) \cdot f \geq -\lambda B$$

- Kong et al, Exponential-Condition-Based Barrier Certificate Generation for Safety Verification of Hybrid Systems, CAV 2013.

Barrier Synthesis (Demo)

Over to Jupyter notebook

Polynomial Optimization Problems

- Unconstrained $\min p(x_1, \dots, x_n)$
- Constrained

$$\begin{aligned} \min \quad & p(x_1, \dots, x_n) \\ \text{s.t.} \quad & p_1(x_1, \dots, x_n) \geq 0 \\ & \vdots \\ & p_m(x_1, \dots, x_n) \geq 0 \end{aligned}$$

Finding Bounds on Optima

$$\begin{aligned} \min \quad & p(x_1, \dots, x_n) \\ \text{s.t.} \quad & p_1(x_1, \dots, x_n) \geq 0 \\ & \vdots \\ & p_m(x_1, \dots, x_n) \geq 0 \end{aligned}$$

- Find largest γ such that

$$p_1 \geq 0 \wedge \dots \wedge p_m \geq 0 \models p \geq \gamma$$

- γ will be a lower bound to true optimal value γ^*.

Stability Analysis

Stability is a fundamental property of dynamical systems. - Very important in engineering applications.

Inputs:

- ODE model: $\frac{d\vec{x}}{dt} = f(\vec{x})$
- Equilibrium: $\vec{x}^*$ s.t. $f(\vec{x}^*) = 0$.

Prove: $\vec{x}^*$ is a *globally asymptotically stable* equilibrium.

In lecture, explain stability notions.

Lyapunov Functions

Find $V(x_1, \dots, x_n)$ such that

- $V(x_1, \dots, x_n)$ is positive definite.
- $\nabla V \cdot f(x_1, \dots, x_n)$ is negative definite.

Using SOS to find stability proofs.

Ref. Papachristadoulou and Prajna, CDC 2002.

Over to Demo

Dealing with Floating Point Issues

Numerical instability issues can be serious.

- John Harrison, Verifying Nonlinear Real Formulas Via Sums of Squares, TPHOLS 2007.
- A. Platzer et al., Real-World Verification, CADE 2009.
- Monniaux et al, On the Generation of Positivstellensatz Witnesses in Degenerate Cases.
- Roux, Voronin, Sank., Validating Numerical Semidefinite Programming Solvers for Polynomial Invariants, SAS 2017 and STTT 2019.

Diagonally Dominant SOS

Replace Positivesemidefiniteness by diagonal dominance conditions.

- Ref. Ali Ahmadi et al, DSOS and SDSOS.

Research Horizons

- Dealing with trigonometric functions.
  - In general, this is undecidable.
  - But restricted cases can be decidable.
- Approximating trigonometric functions by polynomials.
  - Somewhat standard approach.
  - Can be very challenging to prove properties.
- Using SOS to prove properties of neural networks.
  - Fazlyab, Pappas and Morari
  - Works by Peter Seiler and Murat Arcak.

Concluding Remarks

Thank you!!