SDP Relaxations for MAXCUT

from Random Hyperplanes to Sum-of-Squares Certificates

CATS @ UMD

March 3, 2017

Overview

MAXCUT, Hardness and UGC

The MAXCUT Problem

Problem (MAXCUT)

For an n-vertex graph G, find a bipartition $(S, \bar{S})$ *that maximizes* $\left|\left\{\{u, v\} \in E(G) \mid u \in S, v \in \bar{S}\right\}\right|$.

The MAXCUT Problem

Problem (MAXCUT)

For an n-vertex graph G, find a bipartition $(S, \bar{S})$ that maximizes $\left|\left\{\{u, v\} \in E(G) \mid u \in S, v \in \bar{S}\right\}\right|$.

NP-Complete by a straightforward reduction from MAX-2-SAT:

$$\{u, v\} \in E(G) \mid u \in S, v \in \bar{S} \longmapsto (x_u \wedge \neg x_v) \vee (\neg x_u \wedge x_v).$$

The MAXCUT Problem

Problem (MAXCUT)

For an n-vertex graph G, find a bipartition $(S, \bar{S})$ *that maximizes* $\left|\left\{\{u, v\} \in E(G) \mid u \in S, v \in \bar{S}\right\}\right|$.

NP-Complete by a straightforward reduction from MAX-2-SAT:

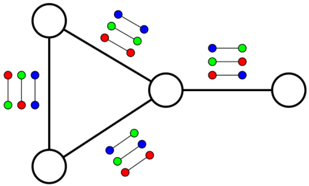
$$\{u, v\} \in E(G) \mid u \in S, v \in \bar{S} \longmapsto (x_u \wedge \neg x_v) \vee (\neg x_u \wedge x_v).$$

Theorem (Håstad 2001, Trevisan et al. 2000)

It is NP-hard to approximate MAXCUT better than $\frac{16}{17} \approx 0.941$.

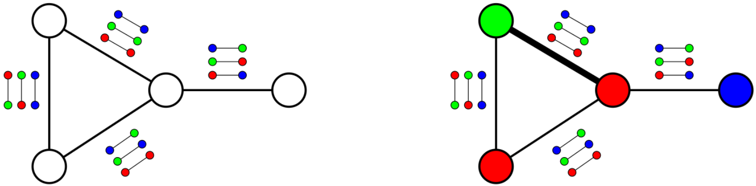
Unique Games and Label Covers

In pictures:



Unique Games and Label Covers

In pictures:



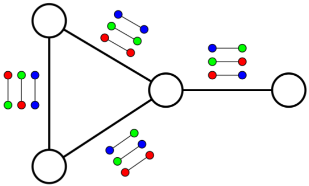
Problem ((c, s)-Gap Label Cover with Unique Constraints)

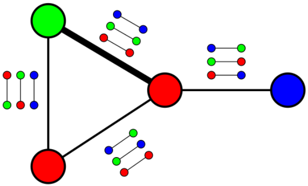
The promise problem (L_{yes}, L_{no})*:*

- $L_{yes} = \{G \mid \textit{some assignment satisfies} \geq c\textit{-fraction of the constraints}\}$.
- $L_{no} = \{G \mid \textit{every assignment satisfies} \leq s\textit{-fraction of the constraints}\}$.

Unique Games and Label Covers

In pictures:





Problem ((c, s)-Gap Label Cover with Unique Constraints)

The promise problem (L_{yes}, L_{no}):

- $L_{yes} = \{G \mid$ *some assignment satisfies* $\geq$ *c-fraction of the constraints*$\}$.
- $L_{no} = \{G \mid$ *every assignment satisfies* $\leq$ *s-fraction of the constraints*$\}$.

Conjecture (Unique Games Conjecture (UCG))

For every sufficiently small pair of constants $\epsilon, \delta > 0$, $\exists k$ *s.t. the* $(1-\delta, \epsilon)$*-gap label-cover problem with k colors is NP-hard.*

LP Shortcomings and SDP

LP Rounding for MAXCUT?

A standard approach to design an approximation algorithm is to formulate an Integer Linear Program (ILP), then solve a *relaxed* LP yielding a (super)optimal solution in *real* variables, then *round* the solution back to integers trying to preserve optimality.

LP Rounding for MAXCUT?

A standard approach to design an approximation algorithm is to formulate an Integer Linear Program (ILP), then solve a *relaxed* LP yielding a (super)optimal solution in *real* variables, then *round* the solution back to integers trying to preserve optimality.

$$
\begin{aligned}
(LP1) \quad & \max \sum_{\{u,v\} \in E(G)} z_{uv} \\
\text{s.t.} \quad & z_{uv} \leq x_u + x_v \\
& z_{uv} \leq (1 - x_u) + (1 - x_v) \\
& z_{uv}, x_u \in \{0, 1\}
\end{aligned}
$$

LP Rounding for MAXCUT?

A standard approach to design an approximation algorithm is to formulate an Integer Linear Program (ILP), then solve a *relaxed* LP yielding a (super)optimal solution in *real* variables, then *round* the solution back to integers trying to preserve optimality.

$$
\begin{aligned}
(LP1) \quad & \max \sum_{\{u,v\} \in E(G)} z_{uv} \\
\text{s.t.} \quad & z_{uv} \leq x_u + x_v \\
& z_{uv} \leq (1 - x_u) + (1 - x_v) \\
& z_{uv}, x_u \in \{0, 1\}
\end{aligned}
$$

Claim

Allowing $x_u \in [0,1]$, we can set $x_u = \frac{1}{2}\ \forall u$ and get $z_{uv} = 1\ \forall \{u, v\}$. Hence, the optimal value of LP1 is $|E|$.

LP Rounding for MAXCUT?

A standard approach to design an approximation algorithm is to formulate an Integer Linear Program (ILP), then solve a *relaxed* LP yielding a (super)optimal solution in *real* variables, then *round* the solution back to integers trying to preserve optimality.

$$
\begin{aligned}
(LP1) \quad & \max \sum_{\{u,v\} \in E(G)} z_{uv} \\
\text{s.t.} \quad & z_{uv} \leq x_u + x_v \\
& z_{uv} \leq (1 - x_u) + (1 - x_v) \\
& z_{uv}, x_u \in \{0, 1\}
\end{aligned}
$$

Claim

Allowing $x_u \in [0,1]$, we can set $x_u = \frac{1}{2}\ \forall u$ and get $z_{uv} = 1\ \forall \{u,v\}$. Hence, the optimal value of LP1 is $|E|$.

Bad: For K_n MAXCUT $\sim |E|/2$.

0.5-approximation by Random Guessing

Each vertex $u \in E(G)$ is added independently to S with probability $\frac{1}{2}$.

0.5-approximation by Random Guessing

Each vertex $u \in E(G)$ is added independently to S with probability $\frac{1}{2}$.

$$
\begin{aligned}
Pr[\{u, v\} \text{ is cut}] &= Pr[(u \in S) \wedge (v \in \bar{S})] + Pr[(u \in \bar{S}) \wedge (v \in S)] \\
&= Pr[u \in S] \cdot Pr[v \in \bar{S}] + Pr[u \in \bar{S}] \cdot Pr[v \in S] \\
&= \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times \frac{1}{2} \\
&= \frac{1}{2}.
\end{aligned}
$$

0.5-approximation by Random Guessing

Each vertex $u \in E(G)$ is added independently to S with probability $\frac{1}{2}$.

$$\begin{aligned}
Pr[\{u, v\} \text{ is cut}] &= Pr[(u \in S) \wedge (v \in \bar{S})] + Pr[(u \in \bar{S}) \wedge (v \in S)] \\
&= Pr[u \in S] \cdot Pr[v \in \bar{S}] + Pr[u \in \bar{S}] \cdot Pr[v \in S] \\
&= \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times \frac{1}{2} \\
&= \frac{1}{2}.
\end{aligned}$$

Remark: Edge-by-edge analysis.

0.5-approximation by Random Guessing

Each vertex $u \in E(G)$ is added independently to S with probability $\frac{1}{2}$.

$$\begin{aligned} Pr[\{u, v\} \text{ is cut}] &= Pr[(u \in S) \wedge (v \in \bar{S})] + Pr[(u \in \bar{S}) \wedge (v \in S)] \\ &= Pr[u \in S] \cdot Pr[v \in \bar{S}] + Pr[u \in \bar{S}] \cdot Pr[v \in S] \\ &= \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times \frac{1}{2} \\ &= \frac{1}{2}. \end{aligned}$$

Remark: Edge-by-edge analysis.
Remark(2): Sub-exponential lower bounds on the size of LP relaxations beating $\frac{1}{2} \sim$ [P. Kothari et al., CATS@UMD February 17, 2017].

High Level Goals for SDP

- Generalize LP.

High Level Goals for SDP

- Generalize LP.
- Simulate nonlinear programming, *in some linear way*.

High Level Goals for SDP

- Generalize LP.
- Simulate nonlinear programming, *in some linear way*.
- Provide a systematic way to expand the search space by introducing more and more variables to obtain tighter approximations to the problem at hand.

Features and Analogies to LP?

- Variables: real-valued $\implies$ *vector-valued*.

Features and Analogies to LP?

- Variables: real-valued $\implies$ *vector-valued*.
- Linear constraints: on variables $\implies$ on *inner products*.

Features and Analogies to LP?

- Variables: real-valued $\Longrightarrow$ *vector-valued*.
- Linear constraints: on variables $\Longrightarrow$ on *inner products*.
- Recover vectors defining a solution by decomposing resulting matrix.

Features and Analogies to LP?

- Variables: real-valued $\Longrightarrow$ *vector-valued*.
- Linear constraints: on variables $\Longrightarrow$ on *inner products*.
- Recover vectors defining a solution by decomposing resulting matrix.
- Convex, well-behaved, solvable in polynomial time.

Features and Analogies to LP?

- Variables: real-valued $\implies$ *vector-valued*.
- Linear constraints: on variables $\implies$ on *inner products*.
- Recover vectors defining a solution by decomposing resulting matrix.
- Convex, well-behaved, solvable in polynomial time.
- Analysis involves statements about high dimensional geometric graphs (e.g., size of indep. set and expansion properties).

Positive Semidefinite Matrices

Definition

A symmetric matrix $X \in \mathbb{R}^{n \times n}$ is positive semidefinite (PSD), or $X \succeq 0$, if any of the following equivalent conditions holds:

1. $a^T X a \geq 0, \forall a \in \mathbb{R}^n$.
2. X admits a Cholseky decomposition $X = LL^T$.
3. All eigenvalues of X are non-negative.

Positive Semidefinite Matrices

Definition

A symmetric matrix $X \in \mathbb{R}^{n \times n}$ is positive semidefinite (PSD), or $X \succeq 0$, if any of the following equivalent conditions holds:

1. $a^T X a \geq 0,\ \forall a \in \mathbb{R}^n$.
2. X admits a Cholseky decomposition $X = LL^T$.
3. All eigenvalues of X are non-negative.

Remark: Condition(1) encodes an (uncountably) infinite set of linear constraints!

Positive Semidefinite Matrices

Definition

A symmetric matrix $X \in \mathbb{R}^{n\times n}$ is positive semidefinite (PSD), or $X \succeq 0$, if any of the following equivalent conditions holds:

1. $a^T X a \geq 0, \ \forall a \in \mathbb{R}^n$.
2. X admits a Cholseky decomposition $X = LL^T$.
3. All eigenvalues of X are non-negative.

Remark: Condition(1) encodes an (uncountably) infinite set of linear constraints!

Remark(2): Condition(2) is equivalent to $X_{ij} = \langle v_i, v_j \rangle$ for vectors $\{v_1, \ldots, v_n\}$ corresponding to the columns of L.

Standard Form (Primal)

$$
\begin{aligned}
(P) \quad & \min \langle C, X \rangle := \sum_{ij} C_{ij} X_{ij} \\
\text{s.t.} \quad & \langle A_i, X \rangle = b_i \quad \forall i \in \{1, \ldots, m\}, \\
& X \succeq 0
\end{aligned}
$$

Standard Form (Primal)

$$
\begin{aligned}
(P) \quad & \min \langle C, X \rangle := \sum_{ij} C_{ij} X_{ij} \\
\text{s.t.} \quad & \langle A_i, X \rangle = b_i \quad \forall i \in \{1, \dots, m\}, \\
& X \succeq 0
\end{aligned}
$$

Remark: LP is a special case of SDP for which X is a diagonal matrix.

Dual Form

$$
\begin{aligned}
(D) \quad & \max b^T y \\
\text{s.t.} \quad & \sum y_i A_i + S = C, \\
& S \succeq 0
\end{aligned}
$$

Dual Form

$$
\begin{aligned}
(D) \quad & \max b^T y \\
\text{s.t.} \quad & \sum y_i A_i + S = C, \\
& S \succeq 0
\end{aligned}
$$

Remark: Weak duality holds.

Powerful Technical Lemma

Lemma ($\sim$ Condition(4) for PSD matrices)

For a symmetric matrix $X \in \mathbb{R}^{n \times n}$, $X \succeq 0$ iff $\langle A, X \rangle \geq 0$, $\forall A \succeq 0$.

Powerful Technical Lemma

Lemma ($\sim$ Condition(4) for PSD matrices)

For a symmetric matrix $X \in \mathbb{R}^{n \times n}$, $X \succeq 0$ iff $\langle A, X \rangle \geq 0$, $\forall A \succeq 0$.

Fact(1): $\langle A, X \rangle = Trace(A^T X)$.

Powerful Technical Lemma

Lemma ($\sim$ Condition(4) for PSD matrices)

For a symmetric matrix $X \in \mathbb{R}^{n \times n}$, $X \succeq 0$ *iff* $\langle A, X \rangle \geq 0$, $\forall A \succeq 0$.

Fact(1): $\langle A, X \rangle = Trace(A^T X)$.
Fact(2): $Trace(AB) = Trace(BA)$ (cyclic permutations).

Powerful Technical Lemma

Lemma ($\sim$ Condition(4) for PSD matrices)

For a symmetric matrix $X \in \mathbb{R}^{n \times n}$, $X \succeq 0$ *iff* $\langle A, X \rangle \geq 0$, $\forall A \succeq 0$.

Fact(1): $\langle A, X \rangle = Trace(A^T X)$.
Fact(2): $Trace(AB) = Trace(BA)$ (cyclic permutations).

Proof (Do on board).

($\Leftarrow$) Suppose X is not PSD and obtain a witness $a \in \mathbb{R}^n$ s.t. $a^T X a < 0$.
($\Rightarrow$) Suppose $A \succeq 0$ and obtain the Cholseky decomposition $A = LL^T$. $\square$

Weak Duality

Lemma

If X, y are feasible for (P) and (D), respectively, then $b^T y \leq \langle C, X \rangle$.

Weak Duality

Lemma

If X, y are feasible for (P) and (D), respectively, then $b^T y \leq \langle C, X \rangle$.

Recall: In (P), $X \succeq 0$ and in (D), $\sum_i y_i A_i + S = C$, with $S \succeq 0$.

Weak Duality

Lemma

If X, y are feasible for (P) and (D), respectively, then $b^T y \leq \langle C, X \rangle$.

Recall: In (P), $X \succeq 0$ and in (D), $\sum_i y_i A_i + S = C$, with $S \succeq 0$.

Proof.

$$\begin{aligned}
\langle C, X \rangle &= \langle \sum_i y_i A_i + S, X \rangle = \langle \sum_i y_i A_i, X \rangle + \langle S, X \rangle \\
&= \sum_i y_i \langle A_i, X \rangle + \langle S, X \rangle \\
&= \sum_i y_i \cdot b_i + \cancel{\langle S, X \rangle}^{\geq 0} && (X \text{ is feasible}) \\
&\geq b^T y && (\text{By the technical lemma})
\end{aligned}$$

□

Strong Duality?

When the value of (P) coincides with that of (D).

Strong Duality?

When the value of (P) coincides with that of (D).
Usually holds under some condition.

Strong Duality?

When the value of (P) coincides with that of (D).
Usually holds under some condition.

Definition (Slater's Condition)

Feasible region has an interior point. In other words, $\exists$ feasible $X \succ 0$.

Goemans and Williamson

SDP Formulation of MAXCUT

$$
\begin{aligned}
\max \quad & \sum_{\{u,v\} \in E(G)} \left(\frac{1}{2} - \frac{1}{2} X_{uv}\right) \\
\text{s.t.} \quad & X_{uu} = 1, \quad \forall u \in V(G) \\
& X \succeq 0
\end{aligned}
$$

SDP Formulation of MAXCUT

$$\max \sum_{\{u,v\}\in E(G)} \left(\frac{1}{2} - \frac{1}{2}X_{uv}\right)$$
$$\text{s.t.} \quad X_{uu} = 1, \quad \forall u \in V(G)$$
$$X \succeq 0$$

Intuition: Set $X = aa^T$ for $a \in \mathbb{R}^n$ is defined as

$$a_u = \begin{cases} 1, & \text{if } u \in S, \\ -1, & \text{o.w.} \end{cases}$$

SDP Formulation of MAXCUT

$$\max \sum_{\{u,v\} \in E(G)} \left(\frac{1}{2} - \frac{1}{2} X_{uv}\right)$$
$$\text{s.t.} \quad X_{uu} = 1, \quad \forall u \in V(G)$$
$$X \succeq 0$$

Intuition: Set $X = aa^T$ for $a \in \mathbb{R}^n$ is defined as

$$a_u = \begin{cases} 1, & \text{if } u \in S, \\ -1, & \text{o.w.} \end{cases}$$

Cut edges contribute 1, uncut edges contribute 0.

Rounding by a Random Hyperplane

Theorem (Goemans and Williamson)

There exists an α_{GW}-approximation algorithm for MAXCUT where

$$\alpha_{GW} = \min_{0 \leq \theta \leq \pi} \frac{2}{\pi} \cdot \frac{\theta}{1 - \cos\theta} \approx 0.87856 \ldots .$$

Rounding by a Random Hyperplane

Theorem (Goemans and Williamson)

There exists an α_{GW}-approximation algorithm for MAXCUT where

$$\alpha_{GW} = \min_{0 \le \theta \le \pi} \frac{2}{\pi} \cdot \frac{\theta}{1 - \cos\theta} \approx 0.87856\ldots.$$

- Solve the SDP to get the solution X.

Rounding by a Random Hyperplane

Theorem (Goemans and Williamson)

There exists an α_{GW}-approximation algorithm for MAXCUT where

$$\alpha_{GW} = \min_{0 \leq \theta \leq \pi} \frac{2}{\pi} \cdot \frac{\theta}{1 - \cos\theta} \approx 0.87856\ldots.$$

- Solve the SDP to get the solution X.
- Recover vector variables y_u for which

$$X_{uv} = \langle y_u, y_v \rangle.$$

Rounding by a Random Hyperplane

Theorem (Goemans and Williamson)

There exists an α_{GW}-approximation algorithm for MAXCUT where

$$\alpha_{GW} = \min_{0 \leq \theta \leq \pi} \frac{2}{\pi} \cdot \frac{\theta}{1 - \cos\theta} \approx 0.87856\ldots.$$

- Solve the SDP to get the solution X.
- Recover vector variables y_u for which

$$X_{uv} = \langle y_u, y_v \rangle.$$

- Pick a uniformly random vector $a \in \mathbb{R}^n$.

Rounding by a Random Hyperplane

Theorem (Goemans and Williamson)

There exists an α_{GW}-approximation algorithm for MAXCUT where

$$\alpha_{GW} = \min_{0 \leq \theta \leq \pi} \frac{2}{\pi} \cdot \frac{\theta}{1 - \cos\theta} \approx 0.87856\ldots.$$

- Solve the SDP to get the solution X.
- Recover vector variables y_u for which

$$X_{uv} = \langle y_u, y_v \rangle.$$

- Pick a uniformly random vector $a \in \mathbb{R}^n$.
- Set $x_u = sgn(\langle a, y_u \rangle)$.

Rounding by a Random Hyperplane

Theorem (Goemans and Williamson)

There exists an α_{GW}-approximation algorithm for MAXCUT where

$$\alpha_{GW} = \min_{0 \leq \theta \leq \pi} \frac{2}{\pi} \cdot \frac{\theta}{1 - \cos\theta} \approx 0.87856\ldots.$$

- Solve the SDP to get the solution X.
- Recover vector variables y_u for which

$$X_{uv} = \langle y_u, y_v \rangle.$$

- Pick a uniformly random vector $a \in \mathbb{R}^n$.
- Set $x_u = sgn(\langle a, y_u \rangle)$.
- Edge-by-edge analysis:

Rounding by a Random Hyperplane

Theorem (Goemans and Williamson)

There exists an α_{GW}-approximation algorithm for MAXCUT where

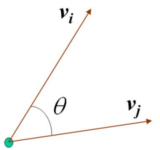
$$\alpha_{GW} = \min_{0 \le \theta \le \pi} \frac{2}{\pi} \cdot \frac{\theta}{1 - \cos\theta} \approx 0.87856\ldots.$$

- Solve the SDP to get the solution X.
- Recover vector variables y_u for which

$$X_{uv} = \langle y_u, y_v \rangle.$$

- Pick a uniformly random vector $a \in \mathbb{R}^n$.
- Set $x_u = sgn(\langle a, y_u \rangle)$.
- Edge-by-edge analysis:

The probability that two vectors are separated by a random hyperplane



$$\frac{\theta}{\pi}$$

Second Generation of SDP Rounding Algorithms

- Beyond random hyperplanes.

Second Generation of SDP Rounding Algorithms

- Beyond random hyperplanes.
- Initiated by Arora, Rao and Vazirani for c-balanced cuts.

Second Generation of SDP Rounding Algorithms

- Beyond random hyperplanes.
- Initiated by Arora, Rao and Vazirani for c-balanced cuts.
- More global analysis.

Second Generation of SDP Rounding Algorithms

- Beyond random hyperplanes.
- Initiated by Arora, Rao and Vazirani for c-balanced cuts.
- More global analysis.
- Different formulation + "triangle inequality".

Second Generation of SDP Rounding Algorithms

- Beyond random hyperplanes.
- Initiated by Arora, Rao and Vazirani for c-balanced cuts.
- More global analysis.
- Different formulation + "triangle inequality".
- Breadth-first search on weighted geometric graph.

SOS Recap

Certifying non-negativity over the hypercube

Problem (Non-negativity)

Given a low-degree polynomial $f : \{0,1\}^n \to \mathbb{R}$, decide if $f \geq 0$ over the hypercube or if there exists a point $x \in \{0,1\}^n$ such that $f(x) < 0$.

Certifying non-negativity over the hypercube

Problem (Non-negativity)

Given a low-degree polynomial $f : \{0,1\}^n \to \mathbb{R}$, decide if $f \geq 0$ over the hypercube or if there exists a point $x \in \{0,1\}^n$ such that $f(x) < 0$.

Example: For an n-vertex graph G, we encode a bipartition of the vertex set by a vector $x \in \{0,1\}^n$ and we let $f_G(x)$ be the number of edges cut by the bipartition x. We get the degree-2 polynomial

$$f_G(x) = \sum_{\{i,j\} \in E(G)} (x_i - x_j)^2.$$

Certifying non-negativity over the hypercube

Problem (Non-negativity)

Given a low-degree polynomial $f : \{0,1\}^n \to \mathbb{R}$, *decide if* $f \geq 0$ *over the hypercube or if there exists a point* $x \in \{0,1\}^n$ *such that* $f(x) < 0$.

Example: For an n-vertex graph G, we encode a bipartition of the vertex set by a vector $x \in \{0,1\}^n$ and we let $f_G(x)$ be the number of edges cut by the bipartition x. We get the degree-2 polynomial

$$f_G(x) = \sum_{\{i,j\} \in E(G)} (x_i - x_j)^2.$$

Deciding if the *polynomial* $c - f_G(x)$ is non-negative over the hypercube is the same as deciding if $MAXCUT(G) \geq c$.

Sum-of-Squares Certificates

Definition

A degree-d sos certificate (of non-negativity) for $f : \{0,1\}^n \to \mathbb{R}$ consists of polynomials $g_1, \ldots, g_r : \{0,1\}^n \to \mathbb{R}$ of degree at most $d/2$ for some $r \in \mathbb{N}$ such that

$$f(x) = g_1^2(x) + \cdots + g_r^2(x),$$

for every $x \in \{0,1\}^n$.

Sum-of-Squares Certificates

Definition

A degree-d sos certificate (of non-negativity) for $f : \{0,1\}^n \to \mathbb{R}$ consists of polynomials $g_1, \ldots, g_r : \{0,1\}^n \to \mathbb{R}$ of degree at most $d/2$ for some $r \in \mathbb{N}$ such that

$$f(x) = g_1^2(x) + \cdots + g_r^2(x),$$

for every $x \in \{0,1\}^n$.

A.k.a *degree-d sos proof* of the inequality $f \geq 0$.

Sum-of-Squares Certificates

Definition

A degree-d sos certificate (of non-negativity) for $f : \{0,1\}^n \to \mathbb{R}$ consists of polynomials $g_1, \ldots, g_r : \{0,1\}^n \to \mathbb{R}$ of degree at most $d/2$ for some $r \in \mathbb{N}$ such that

$$f(x) = g_1^2(x) + \cdots + g_r^2(x),$$

for every $x \in \{0,1\}^n$.

A.k.a *degree-d sos proof* of the inequality $f \geq 0$.

We can assume $r = n^{O(d)}$, thus verifying a certificate takes $n^{O(d)}$ time.

Sum-of-Squares Certificates

Definition

A degree-d sos certificate (of non-negativity) for $f : \{0,1\}^n \to \mathbb{R}$ consists of polynomials $g_1, \ldots, g_r : \{0,1\}^n \to \mathbb{R}$ of degree at most $d/2$ for some $r \in \mathbb{N}$ such that

$$f(x) = g_1^2(x) + \cdots + g_r^2(x),$$

for every $x \in \{0,1\}^n$.

A.k.a *degree-d sos proof* of the inequality $f \geq 0$.

We can assume $r = n^{O(d)}$, thus verifying a certificate takes $n^{O(d)}$ time. Just check that $f - (g_1^2 + \cdots + g_r^2)$ vanishes for every $x \in \{0,1\}^n$.

Finding SOS Certificates

Theorem

There exists an algorithm that given a polynomial $f : \{0,1\}^n \to \mathbb{R}$ and a number $k \in \mathbb{N}$, outputs a degree-k sos certificate for $f + 2^{-n}$ in time $n^{O(k)}$ if f has a degree-k sos certificate.

Finding SOS Certificates

Theorem

There exists an algorithm that given a polynomial $f : \{0,1\}^n \to \mathbb{R}$ *and a number* $k \in \mathbb{N}$, *outputs a degree-k sos certificate for* $f + 2^{-n}$ *in time* $n^{O(k)}$ *if* f *has a degree-k sos certificate.*

Intuition Such polynomials having a degree-d sos certificate form a convex cone, which admits a small semidefinite programming formulation.

PSD Matrices and SOS Certificates

Theorem

A polynomial f has a degree-d sos certificate iff there exists a PSD matrix A such that for all $x \in \{0,1\}^n$,

$$f(x) = \left\langle (1,x)^{\otimes d/2}, A(1,x)^{\otimes d/2} \right\rangle.$$

PSD Matrices and SOS Certificates

Theorem

A polynomial f has a degree-d sos certificate iff there exists a PSD matrix A such that for all $x \in \{0,1\}^n$,

$$f(x) = \langle (1,x)^{\otimes d/2}, A(1,x)^{\otimes d/2} \rangle.$$

Proof (Do on board).

($\Leftarrow$) For a PSD A, extract a degree-d certificate $\{g_1, \ldots g_r\}$.
($\Rightarrow$) For a degree-d sos certificate $f = \sum_{i=1}^{r} g_i^2$ form a PSD matrix A. $\square$

SOS Certificates (Some time later?)

The End

".. often the sign of scientific success is when we eliminate the need for creativity and make boring what was once exciting .. is it just a matter of time until algorithm design will become as boring as solving a single polynomial equation?"

Questions?

akader@cs.umd.edu