Approximability of Constraint Satisfaction Problems

Venkatesan Guruswami

Carnegie Mellon University

October 2009

Constraint Satisfaction Problems (CSPs)

Definition (CSP)

A CSP (denoted $\text{CSP}_q(\mathcal{F})$), specified by

- finite domain $[q] = \{0, 1, \ldots, q-1\}$
- *constraint language* $\mathcal{F}$: a collection of relations over $[q]$, i.e., functions $f : [q]^{a(f)} \to \{0, 1\}$ ($a(f)$ = arity of f)

Constraint Satisfaction Problems (CSPs)

Definition (CSP)

A CSP (denoted $\mathrm{CSP}_q(\mathcal{F})$), specified by

- finite domain $[q] = \{0, 1, \ldots, q-1\}$
- *constraint language* $\mathcal{F}$: a collection of relations over $[q]$, i.e., functions $f : [q]^{a(f)} \to \{0, 1\}$ ($a(f)$ = arity of f)

Definition (CSP instance)

Variable set V.
A collection $\mathcal{C}$ of constraints $\{(f, S)\}$ where $f \in \mathcal{F}$; $S = a(f)$-tuple from V
Question: Is there an assignment $\sigma : V \to [q]$ that satisfies all constraints?
i.e., $f(\sigma_{|S}) = 1$ for each $(f, S) \in \mathcal{C}$.

Boolean CSP, $q = 2$, most basic and of special interest.

CSP examples

CSPs capture many well studied problems in NP.

- $\mathcal{F}$ = CNF formulae: SAT
- $\mathcal{F}$ = not all equal: Graph or hypergraph q-colorability
- $\mathcal{F}$ = affine constraints: Solving linear equations

Rich set of problems based on structure of constraints in underlying $\mathcal{F}$.

Yet, just two possibilities complexity theoretically ...

Schaefer's dichotomy theorem for Boolean CSPs

Theorem

Every Boolean CSP is either in P or NP-complete.

Schaefer's dichotomy theorem for Boolean CSPs

Theorem

Every Boolean CSP is either in P or NP-complete.
More specifically, $CSP_2(\mathcal{F})$ *is polynomial time solvable if every* $f \in \mathcal{F}$ *is*

- 0-*valid*
- 1-*valid*
- *a conjunction of Horn clauses (i.e.,* $(x_1 \wedge \cdots \wedge x_k \rightarrow 0)$ *or* $(x_1 \wedge x_2 \wedge \cdots \wedge x_k \rightarrow x_{k+1})$*)*
- *a conjunction of dual Horn clauses*
- *a 2CNF formula, or*
- *a conjunction of affine equations*

and is NP-complete otherwise.

Dichotomy conjectured for every q [Feder-Vardi], proved for $q = 3$ [Bulatov]

Optimization

- Max $\text{CSP}_q(\mathcal{F})$: Find assignment maximizing fraction of satisfied constraints.
- Can also have weighted constraints, and ask for maximum fractional weight of satisfied constraints

Optimization

- Max $\mathrm{CSP}_q(\mathcal{F})$: Find assignment maximizing fraction of satisfied constraints.
- Can also have weighted constraints, and ask for maximum fractional weight of satisfied constraints
- Minimization version: minimize fraction of unsatisfied constraints.
- Can capture problems with *hard constraints* such as independent set or vertex cover as Max ONES($\mathcal{F}$), Min ONES($\mathcal{F}$):

  satisfy *all* constraints with maximum (minimum) fraction of 1's.

Optimization

- Max $\mathrm{CSP}_q(\mathcal{F})$: Find assignment maximizing fraction of satisfied constraints.
- Can also have weighted constraints, and ask for maximum fractional weight of satisfied constraints
- Minimization version: minimize fraction of unsatisfied constraints.
- Can capture problems with *hard constraints* such as independent set or vertex cover as Max ONES($\mathcal{F}$), Min ONES($\mathcal{F}$):

  satisfy *all* constraints with maximum (minimum) fraction of 1's.
- Let's focus on unweighted Max CSP
  - Example Max CUT. $\mathcal{F} = \{\mathrm{cut}\}$ where $\mathrm{cut}(x, y) = \mathbf{1}(x \neq y)$.
    Note CSP(cut) is in P. Optimization version Max CUT is NP-hard.

Boolean Max CSP

Schaefer's theorem can be strengthened for the following "PCP-like" statement:

Theorem (Khanna,Sudan,Willamson)

For every Boolean constraint language $\mathcal{F}$, either CSP($\mathcal{F}$) is polytime decidable, or there exists $\delta_{\mathcal{F}} < 1$ such that it is NP-hard to distinguish satisfiable instances of CSP($\mathcal{F}$) from instances of Max CSP($\mathcal{F}$) where at most δ_F fraction of constraints are satisfiable.

Boolean Max CSP

Schaefer's theorem can be strengthened for the following "PCP-like" statement:

Theorem (Khanna,Sudan,Willamson)

For every Boolean constraint language $\mathcal{F}$, either CSP($\mathcal{F}$) is polytime decidable, or there exists $\delta_{\mathcal{F}} < 1$ such that it is NP-hard to distinguish satisfiable instances of CSP($\mathcal{F}$) from instances of Max CSP($\mathcal{F}$) where at most δ_F fraction of constraints are satisfiable.

However, even for Schaefer's tractable $\mathcal{F}$ (other than 0-valid and 1-valid cases), Max CSP($\mathcal{F}$) is NP-hard.

Question

Which tractable $\mathcal{F}$ lead to easy optimization versions?

Dichotomy theorem for Boolean Max CSP

Theorem (Creignou;KSW)

For every Boolean constraint language $\mathcal{F}$, Max CSP($\mathcal{F}$) is polynomial time solvable or APX-complete.

Dichotomy theorem for Boolean Max CSP

Theorem (Creignou;KSW)

For every Boolean constraint language $\mathcal{F}$, Max CSP($\mathcal{F}$) is polynomial time solvable or APX-complete.
Max CSP($\mathcal{F}$) is polytime solvable iff $\mathcal{F}$ is 0-valid, 1-valid, or 2-monotone.

($f(x_1, \ldots, x_k)$ is 2-monotone if it is expressible as a 2 term DNF:

$$(x_{i_1} \wedge x_{i_2} \wedge \cdots \wedge x_{i_p}) \vee (\neg x_{j_1} \wedge \cdots \wedge \neg x_{j_q}).$$

$\mathcal{F}$ is 2-monotone if every $f \in F$ is 2-monotone.)

Dichotomy theorem for Boolean Max CSP

Theorem (Creignou;KSW)

For every Boolean constraint language $\mathcal{F}$, Max CSP($\mathcal{F}$) is polynomial time solvable or APX-complete.
Max CSP($\mathcal{F}$) is polytime solvable iff $\mathcal{F}$ is 0-valid, 1-valid, or 2-monotone.

($f(x_1, \ldots, x_k)$ is 2-monotone if it is expressible as a 2 term DNF:

$$(x_{i_1} \wedge x_{i_2} \wedge \cdots \wedge x_{i_p}) \vee (\neg x_{j_1} \wedge \cdots \wedge \neg x_{j_q}).$$

$\mathcal{F}$ is 2-monotone if every $f \in F$ is 2-monotone.)

The 2-monotone case reduces to s-t Min Cut.

Approximating CSPs

Essentially all Max CSP problems are NP-hard, and in fact APX-hard, i.e., hard to approximate within some absolute constant < 1.

Main goal in theory of CSP approximability

Identify **approximation threshold** $\tau_{\mathcal{F}}$ of Max CSP($\mathcal{F}$) for all (or at least interesting?) $\mathcal{F}$!

- Factor $\tau_{\mathcal{F}}$ approximation algorithm (algorithm that finds assignment satisfying a fraction $\geqslant \tau_{\mathcal{F}} \cdot$ Opt of constraints)
- Hardness of obtaining ratio $\tau_{\mathcal{F}} + \varepsilon$ approximation for every $\varepsilon > 0$.

Approximating CSPs

Essentially all Max CSP problems are NP-hard, and in fact APX-hard, i.e., hard to approximate within some absolute constant < 1.

Main goal in theory of CSP approximability

Identify **approximation threshold** $\tau_{\mathcal{F}}$ of Max CSP($\mathcal{F}$) for all (or at least interesting?) $\mathcal{F}$!

- Factor $\tau_{\mathcal{F}}$ approximation algorithm (algorithm that finds assignment satisfying a fraction $\geqslant \tau_{\mathcal{F}} \cdot \text{Opt}$ of constraints)
- Hardness of obtaining ratio $\tau_{\mathcal{F}} + \varepsilon$ approximation for every $\varepsilon > 0$.

Above quest has been very successful

- probably beyond even original expectations
- we now "almost know" the tight answer for *every* CSP.

Two sides

1. Positive results: Efficient algorithms with provable approximation ratios.
2. Negative results: Achieving certain approx. ratio is NP-hard (or hard under some other complexity assumption)

Let's discuss some algorithmic results first.

A mindless approximation algorithm

Random assignment

For each variable independently, assign a value uniformly at random from the domain $[q]$.

Algorithm completely ignores structure of constraints!
In expectation, algorithm satisfies fraction $\geqslant r_{\mathcal{F}} = \min_{f \in \mathcal{F}} r_f$ of constraints. (r_f = prob. that $f(a) = 1$ for random $a \in [q]^{a(f)}$.)
Can be derandomized via conditional expectations.

A mindless approximation algorithm

Random assignment

For each variable independently, assign a value uniformly at random from the domain $[q]$.

Algorithm completely ignores structure of constraints!
In expectation, algorithm satisfies fraction $\geqslant r_{\mathcal{F}} = \min_{f \in \mathcal{F}} r_f$ of constraints. (r_f = prob. that $f(a) = 1$ for random $a \in [q]^{a(f)}$.)
Can be derandomized via conditional expectations.

Examples of random assignment threshold

- $\mathcal{F} = E3SAT$: $7/8$
- $\mathcal{F} =$ 2SAT: $1/2$
- $\mathcal{F} =$ affine constraints over $\mathbb{F}_p$: $1/p$
- $\mathcal{F} = k$-CUT: $1 - 1/k$
- $\mathcal{F} =$ 3MAJ: $1/2$

Approximation Algorithms

- Wasn't much improvement over random assignment algo till early 90s
- In fact, now we know this is sometimes not possible!

- Wasn't much improvement over random assignment algo till early 90s
- In fact, now we know this is sometimes not possible!
- Most pervasive (essentially only) technique:
  1. Solve a convex relaxation of the Max CSP
  2. "Round" the solution to an assignment

We will discuss the simplest case, when the convex relaxation is a linear program (LP), first.

Linear Programming

Integer Linear Program formulation of Max SAT (with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_m$):

Maximize $\frac{1}{m} \cdot \sum_{j=1}^{m} z_j$ subject to

$$\sum_{x_i \in C_j^{\text{pos}}} y_i + \sum_{x_i \in C_j^{\text{neg}}} (1 - y_i) \geqslant z_j \quad \forall j = 1, 2, \ldots, m$$

$$y_i \in \{0, 1\} \quad \forall i = 1, 2, \ldots, n$$

$$0 \leqslant z_j \leqslant 1 \quad \forall j = 1, 2, \ldots, m$$

Linear Programming

Integer Linear Program formulation of Max SAT (with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_m$):

Maximize $\frac{1}{m} \cdot \sum_{j=1}^{m} z_j$ subject to

$$\sum_{x_i \in C_j^{\mathrm{pos}}} y_i + \sum_{x_i \in C_j^{\mathrm{neg}}} (1 - y_i) \geqslant z_j \quad \forall j = 1, 2, \ldots, m$$

$$y_i \in \{0, 1\} \quad \forall i = 1, 2, \ldots, n$$

$$0 \leqslant z_j \leqslant 1 \quad \forall j = 1, 2, \ldots, m$$

Linear program: Relax $y_i \in \{0, 1\}$ to $0 \leqslant y_i \leqslant 1$.
Can solve resulting LP in polynomial time.

Easy exercise

Above LP can decide Horn Satisfiability.

Rounding fractional solution

Need to convert fractional solution y_i to an assignment to x_i.
Can interpret $y_i \in [0, 1]$ as extent to which $x_i = 1$.

Randomized rounding

For each i independently, set $x_i \leftarrow 1$ with prob. y_i.

Rounding fractional solution

Need to convert fractional solution y_i to an assignment to x_i.
Can interpret $y_i \in [0,1]$ as extent to which $x_i = 1$.

Randomized rounding

For each i independently, set $x_i \leftarrow 1$ with prob. y_i.

Prob. that C_j with k literals is satisfied

$$= \quad 1 - \prod_{x_i \in C_j^{\text{pos}}} (1 - y_i) \prod_{x_i \in C_j^{\text{neg}}} y_i$$

Rounding fractional solution

Need to convert fractional solution y_i to an assignment to x_i.
Can interpret $y_i \in [0,1]$ as extent to which $x_i = 1$.

Randomized rounding

For each i independently, set $x_i \leftarrow 1$ with prob. y_i.

Prob. that C_j with k literals is satisfied

$$\begin{aligned} &= 1 - \prod_{x_i \in C_j^{\mathrm{pos}}} (1 - y_i) \prod_{x_i \in C_j^{\mathrm{neg}}} y_i \\ &\geqslant 1 - \left(\frac{k - \sum_{x_i \in C_j^{\mathrm{pos}}} y_i - \sum_{x_i \in C_j^{\mathrm{neg}}} (1 - y_i)}{k} \right)^k \end{aligned}$$

Rounding fractional solution

Need to convert fractional solution y_i to an assignment to x_i.
Can interpret $y_i \in [0,1]$ as extent to which $x_i = 1$.

Randomized rounding

For each i independently, set $x_i \leftarrow 1$ with prob. y_i.

Prob. that C_j with k literals is satisfied

$$\begin{aligned} &= 1 - \prod_{x_i \in C_j^{\mathrm{pos}}} (1 - y_i) \prod_{x_i \in C_j^{\mathrm{neg}}} y_i \\ &\geqslant 1 - \left(\frac{k - \sum_{x_i \in C_j^{\mathrm{pos}}} y_i - \sum_{x_i \in C_j^{\mathrm{neg}}} (1 - y_i)}{k} \right)^k \\ &\geqslant 1 - \left(1 - \frac{z_j}{k}\right)^k \geqslant \left(1 - (1 - 1/k)^k\right) z_j \ . \end{aligned}$$

Max SAT algorithm

- Expected fraction of clauses satisfied $\geqslant \min_k \left(1 - (1 - 1/k)^k\right) \cdot \frac{1}{m} \sum_j z_j$.
- For optimal LP solution, $\frac{1}{m} \sum_j z_j \geqslant \mathrm{Opt}$.

Max SAT algorithm

- Expected fraction of clauses satisfied $\geqslant \min_k \left(1 - (1 - 1/k)^k\right) \cdot \frac{1}{m} \sum_j z_j$.
- For optimal LP solution, $\frac{1}{m} \sum_j z_j \geqslant \mathrm{Opt}$.
- Implies 3/4 approximation algorithm for Max 2SAT. (Random assignment gives 1/2)
- $1 - 1/e$ approximation for Max SAT.

Max SAT algorithm

- Expected fraction of clauses satisfied $\geqslant \min_k \left(1 - (1 - 1/k)^k\right) \cdot \frac{1}{m} \sum_j z_j$.
- For optimal LP solution, $\frac{1}{m} \sum_j z_j \geqslant \mathrm{Opt}$.
- Implies $3/4$ approximation algorithm for Max 2SAT. (Random assignment gives $1/2$)
- $1 - 1/e$ approximation for Max SAT.

- Output better of two randomized algorithms: LP randomized rounding and random assignment
  $\Rightarrow 3/4$ approximation for Max SAT.

Integrality gap

Can we do better than 3/4 by this method (at least for Max 2SAT)?
No, since we get 3/4 times the optimum of the *LP*.

Definition (Integrality gap)

Smallest ratio of integer optimum (Max CSP's true optimum value) to LP optimum.

Integrality gap

Can we do better than 3/4 by this method (at least for Max 2SAT)?
No, since we get 3/4 times the optimum of the *LP*.

Definition (Integrality gap)

Smallest ratio of integer optimum (Max CSP's true optimum value) to LP optimum.

For Max 2SAT instance with 4 clauses

$$(x_1 \vee x_2) \quad (x_1 \vee \neg x_2) \quad (\neg x_1 \vee x_2) \quad (\neg x_1 \neg x_2)$$

- Every assignment satisfies 3 clauses. Integral Opt $= 3/4$
- Assigning $y_1 = y_2 = 1/2$ gives LP solution of value 1.

Thus 3/4 is the best possible approximation factor using this LP.

Note: Closer the integrality gap is to 1, the better the relaxation.

Stronger LPs

Question

Could a smarter, more powerful LP yield a better approximation ratio?

Stronger LPs

Question

Could a smarter, more powerful LP yield a better approximation ratio?
Answer: No.

For Max CUT, integrality gap is close to $1/2$ for basic as well as more powerful LPs. [de la Vega-Mathieu], [Charikar,Makarychev,Makarychev]

- Implies $3/4$ gap for Max 2SAT
- Beating random cut is not possible via LPs!

Let's now digress slightly:

- How does one write a canonical "basic" LP relaxation for every CSP?
- What are these more powerful strengthenings of the basic LP?

A general LP relaxation

CSP asks for a global integral assignment to all variables V.
To make it convex, can allow probability distributions over assignments.

- Same value as integral optimum + Too many variables.

Compromise: Insist on distributions on *local* assignments, say up to s variables ($s \geqslant k$, the arity)

- For each $S \subset V$, $|S| \leqslant s$, a local distribution μ_S over $[q]^{|S|}$.
- Nonnegative variables $y_{i,a}$ for each $i \in V$ and $a \in [q]$, with $\sum_a y_{i,a} = 1$.

A general LP relaxation

CSP asks for a global integral assignment to all variables V.
To make it convex, can allow probability distributions over assignments.

- Same value as integral optimum + Too many variables.

Compromise: Insist on distributions on *local* assignments, say up to s variables ($s \geqslant k$, the arity)

- For each $S \subset V$, $|S| \leqslant s$, a local distribution μ_S over $[q]^{|S|}$.
- Nonnegative variables $y_{i,a}$ for each $i \in V$ and $a \in [q]$, with $\sum_a y_{i,a} = 1$.

Maximize $\sum_{(f,S)\in\mathcal{C}} \mathbb{E}_{x\sim\mu_S}[\ f(x)\ ]$ subject to:

$$y_{i,a} = \sum_{x\in[q]^S, x_i=a} \mu_S(x) \quad \text{for all } S \ni i\ .$$

That is, consistency of marginals of local distributions on each variable.

A general LP relaxation

CSP asks for a global integral assignment to all variables V.
To make it convex, can allow probability distributions over assignments.

- Same value as integral optimum + Too many variables.

Compromise: Insist on distributions on *local* assignments, say up to s variables ($s \geqslant k$, the arity)

- For each $S \subset V$, $|S| \leqslant s$, a local distribution μ_S over $[q]^{|S|}$.
- Nonnegative variables $y_{i,a}$ for each $i \in V$ and $a \in [q]$, with $\sum_a y_{i,a} = 1$.

Maximize $\sum_{(f,S)\in\mathcal{C}} \mathbb{E}_{x\sim\mu_S}[\ f(x)\ ]$ subject to:

$$y_{i,a} = \sum_{x\in[q]^S, x_i=a} \mu_S(x) \quad \text{for all } S \ni i\ .$$

That is, consistency of marginals of local distributions on each variable.

Stronger relaxation: Insist on consistency on all subsets of size r, for some $1 \leqslant r \leqslant s$.

Semidefinite Programming

Max Cut

Input: Graph $G = (\{1, 2, \ldots, n\}, E)$
Find $x_i \in \{-1, 1\}$ for $i = 1, 2, \ldots, n$ that maximizes

$$\frac{1}{|E|} \sum_{(i,j) \in E} \frac{1 - x_i x_j}{2} \ .$$

Max Cut

Input: Graph $G = (\{1, 2, \ldots, n\}, E)$
Find $x_i \in \{-1, 1\}$ for $i = 1, 2, \ldots, n$ that maximizes

$$\frac{1}{|E|} \sum_{(i,j) \in E} \frac{1 - x_i x_j}{2} .$$

Objective function linear in $y_{ij} = x_i x_j$. Matrix $Y = \{y_{ij}\}$, $Y = xx^T$.

Max Cut

Input: Graph $G = (\{1, 2, \ldots, n\}, E)$
Find $x_i \in \{-1, 1\}$ for $i = 1, 2, \ldots, n$ that maximizes

$$\frac{1}{|E|} \sum_{(i,j) \in E} \frac{1 - x_i x_j}{2} \ .$$

Objective function linear in $y_{ij} = x_i x_j$. Matrix $Y = \{y_{ij}\}$, $Y = xx^T$.
Semidefinite Relaxation: Maximize $\frac{1}{|E|} \sum_{(i,j) \in E} \frac{1 - y_{ij}}{2}$ subject to

- Y is positive semidefinite
- $y_{ii} = 1$

Input: Graph $G = (\{1, 2, \ldots, n\}, E)$
Find $x_i \in \{-1, 1\}$ for $i = 1, 2, \ldots, n$ that maximizes

$$\frac{1}{|E|} \sum_{(i,j)\in E} \frac{1 - x_i x_j}{2} \ .$$

Objective function linear in $y_{ij} = x_i x_j$. Matrix $Y = \{y_{ij}\}$, $Y = xx^T$.
Semidefinite Relaxation: Maximize $\frac{1}{|E|} \sum_{(i,j)\in E} \frac{1-y_{ij}}{2}$ subject to

- Y is positive semidefinite
- $y_{ii} = 1$

Can solve above to any desired accuracy in polynomial time [Alizadeh].
Set of PSD matrices is convex, and it is possible to find optimum of linear function over it.

A vector view

Since a positive semidefinite matrix Y admits Cholesky decomposition $Y = V^T V$, the semidefinite program (SDP) finds vectors v_i, $1 \leqslant i \leqslant n$, with $\|v_i\| = 1$ maximizing

$$\frac{1}{|E|} \sum_{(i,j) \in E} \frac{1 - \langle v_i, v_j \rangle}{2} .$$

- SDP allows more general set of solutions: unit vectors in n dimensions instead of one-dimensional ± 1 values.
- In general gives higher objective value (than true Max Cut value)

A vector view

Since a positive semidefinite matrix Y admits Cholesky decomposition $Y = V^T V$, the semidefinite program (SDP) finds vectors v_i, $1 \leqslant i \leqslant n$, with $\|v_i\| = 1$ maximizing

$$\frac{1}{|E|} \sum_{(i,j)\in E} \frac{1 - \langle v_i, v_j \rangle}{2} .$$

- SDP allows more general set of solutions: unit vectors in n dimensions instead of one-dimensional ± 1 values.
- In general gives higher objective value (than true Max Cut value)

Key question

How to "round" vector solution to a Boolean cut?

Random hyperplane rounding

Goemans-Williamson

Pick random hyperplane through the origin. Two hemispheres correspond to two sides of cut.
Pick random vector r and set

$$x_i = \mathrm{sign}(\langle v_i, r \rangle) \ .$$

Random hyperplane rounding

Goemans-Williamson

Pick random hyperplane through the origin. Two hemispheres correspond to two sides of cut.
Pick random vector r and set

$$x_i = \mathrm{sign}(\langle v_i, r \rangle) \ .$$

Intuition: If (i, j) has large contribution $(1 - \langle v_i, v_j \rangle)/2$ to objective function, then angle between v_i, v_j is large, and there is a good chance that v_i, v_j are separated by a random hyperplane.

Rounding analysis

Local analysis for each edge (i, j).
$\theta =$ angle between v_i and v_j.
Contribution to SDP objective function

$$\frac{1 - \langle v_i, v_j \rangle}{2} = \frac{1 - \cos\theta}{2}$$

Probability that we cut edge (i, j)

$$\Pr_r[\mathrm{sign}(\langle v_i, r \rangle) \neq \mathrm{sign}(\langle v_j, r \rangle)] = \frac{\theta}{\pi} \ .$$

Rounding analysis

Local analysis for each edge (i, j).
$\theta =$ angle between v_i and v_j.
Contribution to SDP objective function

$$\frac{1 - \langle v_i, v_j \rangle}{2} = \frac{1 - \cos\theta}{2}$$

Probability that we cut edge (i, j)

$$\Pr_r[\text{sign}(\langle v_i, r \rangle) \neq \text{sign}(\langle v_j, r \rangle)] = \frac{\theta}{\pi} \ .$$

Minimum quotient gives approximation factor

$$\alpha_{GW} = \min_\theta \frac{2\theta}{\pi(1 - \cos\theta)} \approx 0.8785 \ .$$

Other Boolean 2CSPs

SDP based algorithms beat the mindless (random assignment) algorithm for all Boolean 2CSPs.

- Max 2SAT: $\alpha_{GW} = 0.8785...$
  Many subsequent improvements: [Feige-Goemans] 0.931; [Lewin-Livnat-Zwick] 0.94016.
- Max 2CSP: [GW] 0.796.
  [LLZ] improved this to 0.87401.

Natural SDP relaxations; more complicated rounding.

Relaxation for 2SAT

Unit vectors v_i for variables x_i, and a global unit vector b_0 (representing False).

For clause $(x_i \vee x_j)$: contribution to objective function

$$\frac{3 - \langle b_0, v_i \rangle - \langle b_0, v_j \rangle - \langle v_i, v_j \rangle}{4}$$

SDP maximizes sum of above over all clauses.

Relaxation for 2SAT

Unit vectors v_i for variables x_i, and a global unit vector b_0 (representing False).

For clause $(x_i \vee x_j)$: contribution to objective function

$$\frac{3 - \langle b_0, v_i \rangle - \langle b_0, v_j \rangle - \langle v_i, v_j \rangle}{4}$$

SDP maximizes sum of above over all clauses.

Can also add "triangle inequalities"

$$\langle (b_0 \pm v_i), (b_0 \pm v_j) \rangle \geqslant 0 \ .$$

SDP for general CSP

Variables $V = \{x_1, \ldots, x_n\}$, Domain $[q]$. Constraints $\mathcal{C}$.

SDP variables and vectors:

- Vectors $v_{i,a}$ for $1 \leqslant i \leqslant n$ and $a \in [q]$.
- For each constraint $(f, S) \in \mathcal{C}$, a local distribution $\mu_{(f,S)}$ over $[q]^S$ (assignments to variables in S).

SDP for general CSP

Variables $V = \{x_1, \ldots, x_n\}$, Domain $[q]$. Constraints $\mathcal{C}$.

SDP variables and vectors:

- Vectors $v_{i,a}$ for $1 \leqslant i \leqslant n$ and $a \in [q]$.
- For each constraint $(f, S) \in \mathcal{C}$, a local distribution $\mu_{(f,S)}$ over $[q]^S$ (assignments to variables in S).

Maximize $\sum_{(f,S)\in\mathcal{C}} \mathbb{E}_{x\sim\mu_{(f,S)}}[f(x)]$ subject to:

1. $\sum_{a\in[q]} \langle v_{i,a}, v_{i,a} \rangle = 1 \quad \forall i$
2. $\mu_{(f,S)}(x) \geqslant 0$ and $\sum_x \mu_{(f,S)}(x) = 1 \quad \forall (f,S) \in \mathcal{C}$.
3. $\langle v_{i,a}, v_{j,b} \rangle = \Pr_{x\sim\mu_{(f,S)}}\left[x_i = a \text{ and } x_j = b\right]$
$\forall (f,S) \in \mathcal{C}; \quad x_i, x_j \in S; \quad a, b \in [q]$.

In words..

Consistency of local integral distributions on pairs + positive semidefiniteness of pairwise joint probabilities.

Hardness of approximation results

PCP theorem

Starting point for inapproximability results is the famous **PCP theorem**.

Theorem (PCP theorem)

For some absolute constant $\rho < 1$, there is a polynomial time reduction from NP-complete problem 3SAT to Max 3SAT mapping $\phi \mapsto \psi$ such that:

- *(Perfect) completeness: If ϕ is satisfiable, then so is ψ.*
- *Soundness: If ϕ is not satisfiable, then every assignment satisfies at most ρ fraction of ψ's clauses.*

PCP theorem

Starting point for inapproximability results is the famous **PCP theorem**.

Theorem (PCP theorem)

For some absolute constant $\rho < 1$, there is a polynomial time reduction from NP-complete problem 3SAT to Max 3SAT mapping $\phi \mapsto \psi$ such that:

- *(Perfect) completeness: If ϕ is satisfiable, then so is ψ.*
- *Soundness: If ϕ is not satisfiable, then every assignment satisfies at most ρ fraction of ψ's clauses.*

Original proof [Arora-Safra], [Arora-Lund-Motwani-Sudan-Szegedy]: Algebraic techniques: arithmetization, low-degree testing; query parallelization, proof composition, etc.
New proof [Dinur]: expander graphs, iterative amplification of gap.
These give **poor** inapproximability constants ρ.

An optimal result

Subsequent improvements to the constants culminated in the following striking optimal result:

Theorem (Håstad)

For every integer q $\geqslant 2$ and all $\varepsilon, \delta > 0$, it is NP-hard to approximate Max E3-LIN-mod-q within $\frac{1}{q} + \varepsilon$.

- *Hard to tell if linear system is $(1 - \varepsilon)$-satisfiable or at most $(\frac{1}{q} + \delta)$-satisfiable.*

Mindless random assignment algorithm achieves approximation ratio $1/q$.

A powerful result

Gives many other tight (or best known) results by gadgets.

Reduce Max E3-Lin-mod-2 to Max E3SAT

- Replace $x \oplus y \oplus z = 0$ by 4 clauses $(\neg x \vee \neg y \vee \neg z)$, $(\neg x \vee y \vee z)$, $(x \vee \neg y \vee z)$, $(x \vee y \vee \neg z)$.
- Gives $7/8 + \varepsilon$ inapproximability factor for Max E3SAT.

Gives $2/3 + \varepsilon$ inapprox. factor for Max 3MAJ. Also tight.

$21/22 + \varepsilon$ for Max 2SAT, $16/17 + \varepsilon$ for Max CUT, $15/16 + \varepsilon$ for Max NAE3SAT, etc.

- (Probably) not tight, but remain best known NP-hardness bounds.

Perfect completeness

- Reducing 3LIN to 3SAT shows that it is hard to satisfy more than 7/8 of clauses in a $(1-\varepsilon)$-satisfiable formula.
- Inherent for 3LIN
- What about satisfiable 3SAT formulae?

Perfect completeness

- Reducing 3LIN to 3SAT shows that it is hard to satisfy more than $7/8$ of clauses in a $(1-\varepsilon)$-satisfiable formula.
- Inherent for 3LIN
- What about satisfiable 3SAT formulae?

Theorem (Håstad)

For every $\delta > 0$, given an E3SAT formula ϕ, it is NP-hard to distinguish between the cases when ϕ is satisfiable and when ϕ is at most $(\frac{7}{8}+\delta)$-satisfiable.

Next:

1. Some details about such tight hardness results.
2. Approximation resistance

Followed by reductions from Unique Games.

Label Cover

Starting point for strong inapproximability results is almost always the **Label Cover** problem.

Parameterized by integer R. Denote by LabelCover(R).

- 2CSP over large domain (of size R)
- "Projection" constraints

Label Cover

Starting point for strong inapproximability results is almost always the **Label Cover** problem.

Parameterized by integer R. Denote by LabelCover(R).

- 2CSP over large domain (of size R)
- "Projection" constraints

Instance consists of:

1. Bipartite graph $G = (V, W, E)$.
2. For each $e \in E$, a function $\pi_e : [R] \to [R]$.

The value of an assignment (labeling) $\ell : V \cup W \to [R]$ is the fraction of edges $e = (v, w)$ for which $\pi_e(\ell(w)) = \ell(v)$.

Optimization goal: Find labeling with maximum value.

Hardness of Label Cover

Theorem (PCP theorem + Raz's parallel repetition)

There exists an absolute constant $\gamma_0 > 0$ *such that for all* R *it is NP-hard to tell if an instance of* LabelCover(R) *has value* 1 *or value at most* $1/R^{\gamma_0}$.

By picking R large enough, get arbitrarily large gap for a rather nice 2CSP (over a large alphabet).

Reducing from Label Cover

Gadget: "Encode" the projection constraint $\pi(\ell(w)) = \ell(v)$ on labels $\ell(v), \ell(w)$ belonging to large alphabet $[R]$ as (a collection of) simple *tests* on *few bits*.

- Test should correspond to target CSP
- For example, for Max E3-LIN-Mod-2, check parity of 3 bits ($x \oplus y \oplus z = 0/1$)

Must necessarily have larger soundness error, but amazingly can get the optimal bound for many CSPs (3LIN, 3SAT, 4-set-splitting, etc.)

Gadget structure

To reduce projection constraint to some Boolean CSP:

- Expect Boolean tables f and g encoding $\ell(v)$ and $\ell(w)$ respectively (as per some code C).
- Check binary constraints on few locations of f and g (example $f(x) \oplus g(y) \oplus g(z) = 0$)

Property we would like to guarantee:

1. *Completeness*: For a, b satisfying $\pi(b) = a$, legal encoding f, g of a, b satisfies all (or most of) the binary constraints.
2. *Soundness*: If f, g satisfy more than $s + \delta$ fraction of constraints, then can "decode" f, g into "consistent" labels.

Long code

Which "code" to use (for binary encoding of labels)?

Great suggestion by [Bellare-Goldreich-Sudan]: **Long code**

Long code encoding LC maps $[R]$ to $\{0,1\}^{2^R}$

- Long encoding of $a \in [R]$, denoted $\mathrm{LC}^{(a)}$, is a Boolean function $\{0,1\}^R \rightarrow \{0,1\}$
- $\mathrm{LC}^{(a)}(x) = x_a$. "Dictator" function — projection on the a'th coordinate.

Long code is the *most redundant* of all codes!! (When R is a constant, we can afford it.)

Redundancy enables (approximate) checking of global property (namely, the projection constraint on $[R]$) by very local constraints.

Long code testing

- Given tables/functions $f : \{0,1\}^R \to \{0,1\}$ and $g : \{0,1\}^R \to \{0,1\}$, and a projection constraint $\pi : [R] \to [R]$.
- Goal: Check if f and g are long codes of "consistent" values a and b that satisfy $\pi(b) = a$.
- Only allowed to query very few (randomly picked) locations of f, g, and check they obey a local constraint.

Long code testing

- Given tables/functions $f : \{0,1\}^R \to \{0,1\}$ and $g : \{0,1\}^R \to \{0,1\}$, and a projection constraint $\pi : [R] \to [R]$.
- Goal: Check if f and g are long codes of "consistent" values a and b that satisfy $\pi(b) = a$.
- Only allowed to query very few (randomly picked) locations of f, g, and check they obey a local constraint.

A 3-query test (aimed at showing hardness for Max E3LIN-Mod-2):

- Pick $x, y \in \{0,1\}^R$ independently and u.a.r.
- Define $z \in \{0,1\}^R$ by $z_j = y_j \oplus x_{\pi(j)}$.
- With prob. 1/2 check $f(x) \oplus g(y) \oplus g(z) = 0$, and with prob. 1/2 check $f(x) \oplus g(y) \oplus g(\bar{z}) = 1$ (here $\bar{z}$ denotes the coordinate-wise complement of the bit vector z).

Is this a good test?

Completeness: Suppose $f(x) = x_a$ and $g(y) = y_b$ and $\pi(b) = a$. Then

$$g(z) = z_b = y_b \oplus x_{\pi(b)} = y_b \oplus x_a \ .$$

So $f(x) \oplus g(y) \oplus g(z) = x_a \oplus y_b \oplus (y_b \oplus x_a) = 0$.
Similarly $f(x) \oplus g(y) \oplus g(\bar{z}) = 1$

Thus *all* 3LIN constraints are satisfied.

Is this a good test?

Completeness: Suppose $f(x) = x_a$ and $g(y) = y_b$ and $\pi(b) = a$. Then

$$g(z) = z_b = y_b \oplus x_{\pi(b)} = y_b \oplus x_a \ .$$

So $f(x) \oplus g(y) \oplus g(z) = x_a \oplus y_b \oplus (y_b \oplus x_a) = 0$.
Similarly $f(x) \oplus g(y) \oplus g(\bar{z}) = 1$

Thus *all* 3LIN constraints are satisfied.

Soundness?

Question

If most 3LIN constraints are satisfied, does it mean that f, g are "like" long codes (in some reasonable sense)?

Answer: NO.

Linear functions

Polymorphisms: For linear equations mod 2, xor of an odd number of satisfying assignments gives another satisfying assignment.

The functions $g(y) = y_1 \oplus y_2 \oplus \cdots \oplus y_{2k+1}$ and $f(x) = x_{\pi(1)} \oplus x_{\pi(2)} \oplus \cdots \oplus x_{\pi(2k+1)}$ also satisfy all constraints.

For k large, g is "not like" any long code.

Linear functions

Polymorphisms: For linear equations mod 2, xor of an odd number of satisfying assignments gives another satisfying assignment.

The functions $g(y) = y_1 \oplus y_2 \oplus \cdots \oplus y_{2k+1}$ and $f(x) = x_{\pi(1)} \oplus x_{\pi(2)} \oplus \cdots \oplus x_{\pi(2k+1)}$ also satisfy all constraints.

For k large, g is "not like" any long code.

Håstad's insight: add noise to attenuate linear functions of many variables ("dampen high frequencies")

- Must lose perfect completeness as satisfiability of linear equations is in P.

3LIN test with noise

- Sample $x, y \in \{0,1\}^R$ independently and u.a.r.
- Sample $\mu \in \{0,1\}^R$ as follows: for each $j \in [R]$ independently

$$\mu_j = \begin{cases} 0 & \text{with prob. } 1-\varepsilon \\ 1 & \text{with prob. } \varepsilon \end{cases}$$

- Define $z \in \{0,1\}^R$ by $z_j = x_{\pi(j)} \oplus y_j \oplus \mu_j$.
- With prob. 1/2 check $f(x) \oplus g(y) \oplus g(z) = 0$,
  and with prob. 1/2 check $f(x) \oplus g(y) \oplus g(\bar{z}) = 1$

3LIN test with noise

- Sample $x, y \in \{0,1\}^R$ independently and u.a.r.
- Sample $\mu \in \{0,1\}^R$ as follows: for each $j \in [R]$ independently

$$\mu_j = \begin{cases} 0 & \text{with prob. } 1-\varepsilon \\ 1 & \text{with prob. } \varepsilon \end{cases}$$

- Define $z \in \{0,1\}^R$ by $z_j = x_{\pi(j)} \oplus y_j \oplus \mu_j$.
- With prob. $1/2$ check $f(x) \oplus g(y) \oplus g(z) = 0$,
  and with prob. $1/2$ check $f(x) \oplus g(y) \oplus g(\bar{z}) = 1$

Completeness: If $g(y) = y_b$ and $f(x) = x_{\pi(b)}$, $(1-\varepsilon)$ of the 3LIN constraints are satisfied (whenever $\mu_b = 0$).

Easy calculation

For odd k, probability that xor of k long codes (i.e., linear function of k variables) satisfies the tested 3LIN constraint equals $\frac{1}{2} + \frac{(1-2\varepsilon)^k}{2}$.

Soundness for general functions

By expressing f, g (or rather $(-1)^f, (-1)^g$) as a linear combination of linear functions ("Fourier-Walsh" expansion), can prove that if $(1/2+\delta)$ of the 3LIN checks are satisfied, then there must exist

$$S, T \subset [R], \quad |S|, |T| \leqslant c(\delta, \varepsilon), \quad S \cap \pi(T) \neq \emptyset$$

for which f (resp. g) has non-trivial agreement with the linear function $\bigoplus_{i \in S} x_i$ (resp. $\bigoplus_{j \in T} y_j$).

- In fact, $\exists$ distributions D_f and D_g on $2^{[R]}$ for which above happens with good probability (for $(S, T) \in_R D_f \times D_g$).

$\exists$ a *randomized "decoding" procedure* Dec mapping a Boolean function on $\{0,1\}^R$ to $[R]$ such that, when f, g satisfy above condition,

$$\Pr\big[\mathrm{Dec}(f) = \pi(\mathrm{Dec}(g))\big] > \alpha(\delta, \varepsilon) \ .$$

Overall reduction from Label Cover

Plug in long code test on functions f_u, g_v for every edge $e = (u, v)$ with projection constraint π_e.

Completeness $(1 - \varepsilon)$: Just use long codes of a satisfying assignment to Label Cover instance.

Overall reduction from Label Cover

Plug in long code test on functions f_u, g_v for every edge $e = (u, v)$ with projection constraint π_e.

Completeness $(1 - \varepsilon)$: Just use long codes of a satisfying assignment to Label Cover instance.

Soundness $(1/2 + 2\delta)$: Contrapositive: Extract good labeling, satisfying more than $1/R^{\gamma_0}$ constraints of Label Cover, if more than $(1/2 + 2\delta)$ of 3LIN constraints are satisfied.

Overall reduction from Label Cover

Plug in long code test on functions f_u, g_v for every edge $e = (u, v)$ with projection constraint π_e.

Completeness $(1 - \varepsilon)$: Just use long codes of a satisfying assignment to Label Cover instance.

Soundness $(1/2 + 2\delta)$: Contrapositive: Extract good labeling, satisfying more than $1/R^{\gamma_0}$ constraints of Label Cover, if more than $(1/2 + 2\delta)$ of 3LIN constraints are satisfied.

- Run Dec independently for each f_u and each g_v.
- Averaging: $\geqslant \delta$ fraction of edges (u, v) are *good*, i.e., $\geqslant (1/2 + \delta)$ of 3LIN constraints on the long code test for (f_u, g_v) are satisfied.
- For each good edge, decoded labels are consistent with prob. $\alpha(\delta, \varepsilon)$.
- Labeling output by Dec satisfies expected $\delta \cdot \alpha(\delta, \varepsilon)$ fraction of Label Cover constraints.
- Pick R large enough so that $\delta \cdot \alpha(\delta, \varepsilon) > R^{-\gamma_0}$.

Approximation resistance

Beating random assignment

- Max E3LIN-Mod-2, Max E3SAT, Max 4-set splitting, etc. are **approximation resistant**, in the sense that beating the mindless random assignment algorithm is NP-hard.
- Max 2SAT, Max CUT, Max 2CSP admit non-trivial approximations (via semidefinite programming).

Question

Which predicates lead to approximation resistant Max CSPs?

Some results

Every 2CSP (over any domain $[q]$) is **not** approximation resistant. [Goemans-Williamson], [Engebretsen-G], [Håstad]

Bounded occurrence CSPs approximable beyond random assignment threshold [Håstad]

Complete answer for Boolean 3CSPs

- Approximation resistant iff implied by parity or its complement, otherwise admits non-trivial approximation. [Håstad] + [Zwick]

[Hast] classified 354 of the 400 essentially different arity 4 Boolean CSPs (79 approximation resistant).

Approximation resistance results

Large k:

- Boolean kCSP with $2^{O(\sqrt{k})}$ satisfying assignments that is approximation resistant. [Samorodnitsky-Trevisan], [Håstad-Khot]
- No predicate with $\leqslant c \cdot k$ satisfying assignments is approximation resistant [Hast; Charikar-Makarychev-Makarychev]
- Random predicate is approx. resistant w.h.p. [Håstad]*

Approximation resistance results

Large k:

- Boolean kCSP with $2^{O(\sqrt{k})}$ satisfying assignments that is approximation resistant. [Samorodnitsky-Trevisan], [Håstad-Khot]
- No predicate with $\leqslant c \cdot k$ satisfying assignments is approximation resistant [Hast; Charikar-Makarychev-Makarychev]
- Random predicate is approx. resistant w.h.p. [Håstad]*

If there is a pairwise independent distribution supported within the satisfying assignments of the predicate, then it is approximation resistant* [Austrin-Mossel]

- Implies earlier result of [Samorodnitsky-Trevisan] that Max kCSP is hard to approximate with a factor $\Theta(k/2^k)^*$

For more details, go to Per Austrin's talk.

* assuming the Unique Games conjecture

Approximability of 2CSPs

Label Cover + Long Code (LC^2) framework $\Longrightarrow$ many powerful hardness results for CSPs of arity 3 and above.

What about 2CSPs where SDPs give non-trivial (and sometimes bizarre irrational) approximation ratios?

Good 2-query tests for testing consistency (as per projection π) of a pair of purported long codes f, g?

Approximability of 2CSPs

Label Cover + Long Code (LC^2) framework $\Longrightarrow$ many powerful hardness results for CSPs of arity 3 and above.

What about 2CSPs where SDPs give non-trivial (and sometimes bizarre irrational) approximation ratios?

Good 2-query tests for testing consistency (as per projection π) of a pair of purported long codes f, g?

Here's a natural test (that "saves" one query in Håstad's test):

- Pick $x \in \{0,1\}^R$ u.a.r, and noise vector $\mu \in \{0,1\}^R$ s.t. $\mu_I = 0$ with prob. $1-\varepsilon$ for each $I \in [R]$.
- For each $j \in [R]$, set $y_j = x_{\pi(j)} \oplus \mu_j$.
- With prob. $1/2$, check $f(x) \oplus g(y) = 0$,
  with prob. $1/2$, check $f(x) \oplus g(\bar{y}) = 1$.

Good test?

- Query $y \in \{0,1\}^R$ to table g is highly non-uniform.
- $y \approx x \circ \pi$ reveals lot of information about π: $y_k = y_l$ whenever$\pi(k) = \pi(l)$ and independent otherwise.
- Thus can "piece together" many inconsistent g, say a different long code for each part of the hypercube corresponding to the different projection constraints π_e in which w participates.
  - No hope of decoding a single global label $\ell(w)$ for w.
- What would/could fix this?

Unique Games CSP

Khot's insight: This problem goes away if π is a *bijection*.

- In this case, y is uniformly distributed (since x is); gives no clue about π!

Unique Games CSP

Khot's insight: This problem goes away if π is a *bijection*.

- In this case, y is uniformly distributed (since x is); gives no clue about π!

Reduce from special case of LabelCover(R) called UniqueGames(R).

Same as Label Cover, except for each $e \in E$, the projection constraint π_e is a **bijection**. Formally, instance consists of

1. Bipartite graph $G = (V, W, E)$.
2. For each $(v, w) \in E$, a bijection $\pi_{w \to v} : [R] \to [R]$.

Goal: Find labeling $\ell : V \cup W$ with maximum "value", where value = fraction of edges $(v, w) \in E$, $\pi_{w \to v}(\ell(w)) = \ell(v)$.

Example of UniqueGames(R): E2-Lin-mod-R.

- Equations of form $x_i - x_j \equiv c_{ij} \pmod{R}$.

Complexity of Unique Games

Theorem (Easy)

Given a UniqueGames(R) *instance, telling if it is satisfiable (i.e., admits labeling with value* 1*) is in P.*

Complexity of Unique Games

Theorem (Easy)

Given a UniqueGames(R) *instance, telling if it is satisfiable (i.e., admits labeling with value* 1*) is in P.*

Unique Games conjecture — UGC [Khot]

For every $\varepsilon, \delta > 0$, there is a large enough R such that given an instance $\mathcal{I}$ of UniqueGames(R), it is hard to distinguish between the following two cases:

1. $\mathcal{I}$ admits a labeling with value $\geqslant 1 - \varepsilon$.
2. All labelings to $\mathcal{I}$ have value $\leqslant \delta$.

Small amount of noise renders problem inapproximable...

Why UGC?

UGC has some powerful implications:

- Many optimal inapproximability results: Vertex Cover on graphs and hypergraphs, *every* CSP, *every* ordering CSP.
- Led to new integrality gap constructions (and important consequences for metric embeddings)

Why UGC?

UGC has some powerful implications:

- Many optimal inapproximability results: Vertex Cover on graphs and hypergraphs, *every* CSP, *every* ordering CSP.
- Led to new integrality gap constructions (and important consequences for metric embeddings)

Notorious conjecture; no consensus either way ...

- Seemingly no plausible avenue to prove it currently?
  Suffices to prove conjecture for $\delta = 0.99$, or even $\delta = 1 - \varepsilon^{0.51}$.
- Attempts to disprove (based on natural SDP) have failed, but potential of strengthened SDPs not fully ruled out.
- **Not** approximation resistant (it is a 2CSP).
  $\approx 1/R^{\varepsilon/2}$ approximation known. Any improvement would refute UGC.

Back to Boolean 2CSPs

Testing bijection constraint $\pi(b) = a$ given purported long codes f, g of a, b:

- Pick $x \in \{0,1\}^R$ u.a.r, and ε-biased noise vector $\mu \in \{0,1\}^R$.
- Set $y = x \circ \pi \oplus \mu$, i.e., for each $j \in [R]$, $y_j = x_{\pi(j)} \oplus \mu_j$.
- With prob. 1/2, check $f(x) \oplus g(y) = 0$,
  with prob. 1/2, check $f(x) \oplus g(\bar{y}) = 1$.

Now that π is a bijection, turns out it is enough to just test **one** function (essentially assume $\pi = \mathrm{Id}$).

Dictatorship testing

Given access to $f : \{0,1\}^R \to \{0,1\}$.

Make few queries to f, according to some clever distribution, and check constraint Γ on queried bits.

- Γ corresponds to target CSP of interest. Eg. for Max CUT, check $f(x) \neq f(y)$.

Aim: Test must distinguish dictator functions from functions far from every dictator.

Dictatorship testing

Given access to $f : \{0,1\}^R \to \{0,1\}$.

Make few queries to f, according to some clever distribution, and check constraint Γ on queried bits.

- Γ corresponds to target CSP of interest. Eg. for Max CUT, check $f(x) \neq f(y)$.

Aim: Test must distinguish dictator functions from functions far from every dictator.

Completeness: For every $i \in [R]$, if f is the dictator function $f(x) = x_i$, test accepts with probability $\geqslant c$.

Soundness: If $\text{Influence}_i(f)$ is "small" for *every* $i \in [R]$, then test accepts with probability $\leqslant s$.

$$\text{Influence}_i(f) = \Pr_x[f(x) \neq f(x \oplus e_i)]$$

Why dictatorship tests?

Dictatorship test for Max Cut

Parameter $\rho > 1/2$.
Testing a function $f : \{0,1\}^R \to \{0,1\}$

- Pick $x \in \{0,1\}^R$ u.a.r.
- For each $j \in [R]$,

$$y_j = \begin{cases} x_j & \text{with prob. } 1-\rho \\ \overline{x_j} & \text{with prob. } \rho \end{cases}$$

- Check the CUT constraint $f(x) \neq f(y)$, accept if so.

Completeness

When f a dictator, say $f(x) = x_i$,
Probability test accepts $= \rho$.

Dictatorship test for Max Cut

Soundness

What's the best f that has no influential coordinates?

Dictatorship test for Max Cut

Soundness

What's the best f that has no influential coordinates?

Answer: Majority function.

Also, $\Pr_{x,y}[\mathsf{Maj}(x) \neq \mathsf{Maj}(y)] \to \frac{\arccos(1-2\rho)}{\pi}$ for large R.

Theorem (Majority is Stablest (Mossel-O'Donnell-Oleszkiewicz))

For all $\rho > 1/2$ and $\varepsilon > 0$, there is a small enough $\tau = \tau(\rho, \varepsilon) > 0$ s.t. if

$$\Pr_{x,y}[f(x) \neq f(y)] \geqslant \frac{\arccos(1-2\rho)}{\pi} + \varepsilon,$$

then for some $i \in [R]$, $\mathsf{Influence}_i(f) \geqslant \tau$.

Dictatorship test for Max Cut

Soundness

What's the best f that has no influential coordinates?
Answer: Majority function.
Also, $\Pr_{x,y}[\mathsf{Maj}(x) \neq \mathsf{Maj}(y)] \to \frac{\arccos(1-2\rho)}{\pi}$ for large R.

Theorem (Majority is Stablest (Mossel-O'Donnell-Oleszkiewicz))

For all $\rho > 1/2$ and $\varepsilon > 0$, there is a small enough $\tau = \tau(\rho, \varepsilon) > 0$ s.t. if

$$\Pr_{x,y}[f(x) \neq f(y)] \geqslant \frac{\arccos(1-2\rho)}{\pi} + \varepsilon,$$

then for some $i \in [R]$, $\mathsf{Influence}_i(f) \geqslant \tau$.

Therefore, get $\rho - \varepsilon$ vs. $\frac{\arccos(1-2\rho)}{\pi} + \varepsilon$ gap (for any $1/2 < \rho < 1$).

- or $\frac{1-\cos\theta}{2} - \varepsilon$ vs. $\frac{\theta}{\pi} + \varepsilon$ where $\theta = \arccos(1-2\rho) \in (\pi/2, \pi)$.
- Same as SDP optimum vs. cut found by random hyperplane rounding!
- Optimizing over θ, gives 0.8785.. hardness factor for Max CUT [Khot-Kindler-Mossel-O'Donnell]

Approximate polymorphism perspective

Polymorphism combines many satisfying assignments to produce a new satisfying assignment.

- Dictator/projection functions $\Leftrightarrow$ trivial polymorphism

"Approximate polymorphism" combines assignments satisfying Opt fraction of constraints to a new assignment.

- Dictator function: preserves fraction Opt of satisfied constraints.
- What's the best *non-influential* polymorphism?

Approximate polymorphisms for Max CUT

Majority is a (non-trivial) polymorphism for CSP(cut).

Approximate polymorphisms for Max CUT

Majority is a (non-trivial) polymorphism for CSP(cut).

In the context of Max CUT:

- Majority is the *best* "low-influence" approximate polymorphism.
- Given R distributions over assignments that satisfy a specific cut constraint with probability ρ
  - coordinate-wise majority satisfies that constraint with probability $\approx \frac{\arccos(1-2\rho)}{\pi}$
  - And this is largest possible for combining functions with no influential variable.

More about this in Prasad Raghavendra's talk after lunch.

Constructing dictatorship tests

For Max Cut, we "cooked" up a natural test.

In general, how to get a good dictatorship test for a CSP?

Very general answer [Raghavendra]

Can convert **any** integrality gap instance for the "canonical" semidefinite program into dictatorship test with matching parameters!

- Instance with SDP opt c and integral optimum $s \implies$ Dictatorship test with completeness $c - \varepsilon$ and soundness $s + \varepsilon$.

Proof proceeds via a rounding algorithm for the SDP.

Corollary

Assuming UGC, the canonical SDP delivers the best possible approximation ratio, for ***every*** *CSP.*

Recall the SDP

Local integral distributions that are consistent on pairs + positive semidefiniteness of pairwise joint probabilities.

Maximize $\sum_{(h,S)\in\mathcal{C}} \mathbb{E}_{x\sim\mu_{(h,S)}}[h(x)]$ subject to:

1. $\sum_{a\in[q]}\langle v_{i,a}, v_{i,a}\rangle = 1 \quad \forall i$
2. $\mu_{(h,S)}(x) \geqslant 0$ and $\sum_x \mu_{(h,S)}(x) = 1 \quad \forall (h,S)\in\mathcal{C}$.
3. $\langle v_{i,a}, v_{j,b}\rangle = \Pr_{x\sim\mu_{(h,S)}}\left[x_i = a \wedge x_j = b\right]$
   $\forall (h,S)\in\mathcal{C};\quad x_i, x_j \in S;\quad a,b\in[q].$

Recall the SDP

Local integral distributions that are consistent on pairs + positive semidefiniteness of pairwise joint probabilities.

Maximize $\sum_{(h,S)\in\mathcal{C}} \mathbb{E}_{x\sim\mu_{(h,S)}}[h(x)]$ subject to:

1. $\sum_{a\in[q]}\langle v_{i,a}, v_{i,a}\rangle = 1 \quad \forall i$
2. $\mu_{(h,S)}(x) \geqslant 0$ and $\sum_x \mu_{(h,S)}(x) = 1 \quad \forall (h,S)\in\mathcal{C}$.
3. $\langle v_{i,a}, v_{j,b}\rangle = \Pr_{x\sim\mu_{(h,S)}}\left[x_i = a \wedge x_j = b\right]$
   $\forall (h,S)\in\mathcal{C};\quad x_i, x_j \in S;\quad a,b\in[q]$.

Dictatorship test for function $f : [q]^R \to \{0,1\}$:

- Pick a random constraint $(h,S)\in\mathcal{C}$. Let $k = |S|$ be its arity.
- Pick k vectors $y^{(1)}, y^{(2)}, \cdots, y^{(k)} \in [q]^R$ where for each $i\in[R]$ independently, the i'th coordinates $(y_i^{(1)}, y_i^{(2)}, \cdots, y_i^{(k)}) \in_R \mu_{(h,S)}$ are chosen as per the local integral distribution.*
- Check the constraint $h\big(f(y^{(1)}), f(y^{(2)}), \ldots, f(y^{(k)})\big)$

* Actually, one samples from a slightly noisy version of $\mu_{(h,S)}$

Canonical SDP for Boolean case

Unit vectors v_i for variables x_i, and a global unit vector b_0 (representing False).

Value of any constraint on x_i, x_j can be expressed as linear function of $\langle b_0, v_i \rangle$, $\langle b_0, v_j \rangle$, and $\langle v_i, v_j \rangle$.

SDP maximizes sum of this linear function over all constraints, subject to

$$\langle b_0, b_0 \rangle = 1; \quad \langle v_i, v_i \rangle = 1 \ \ \forall i$$

And the "triangle inequalities"

$$\langle (b_0 \pm v_i), (b_0 \pm v_j) \rangle \geqslant 0$$

for all i, j for which x_i, x_j participate in a constraint.

A permutation problem

Topological sorting: Given a directed *acyclic* graph, can order its vertices so that all edges go forward.

- What if digraph is only "nearly" acyclic, say 1% of the edges need to be removed to make it acyclic?
- Can one find an ordering such that most of the edges go forward?
- Equivalently, find acyclic subgraph with maximum fraction of edges.

Picking a random ordering (or better of any ordering and its reverse) finds acyclic subgraph with at least $1/2$ the edges.

Theorem [G.-Manokaran-Raghavendra]

Assuming UGC, this is best possible.
$\forall \varepsilon, \delta > 0$, given a $(1-\varepsilon)$-acyclic graph, it is UG-hard to find an acyclic subgraph with $(1/2+\delta)$ edges.

Max Acyclic Subgraph as a CSP

Max Acyclic Subgraph can be expressed "like" a 2CSP:

- Variables = vertices of graph
- Edge $x \to y$ = constraint $x < y$
- Large domain $[n]$ where $n =$ number of vertices.

Even though it is not a usual 2CSP due to growing domain size, UniqueGames hardness shown by relating it to a "proxy" CSP over a bounded domain.

Ordering CSP

Ordering constraint of arity k = subset Π of $k!$ possible permutations

- MAS: $x_i < x_j$ $\qquad \Pi = \{12\}$
- Betweenness: x_j between x_i and x_ℓ.
  $\Pi = \{123, 321\}$ applied to triple (x_i, x_j, x_ℓ).

Instance of ordering kCSP Π:

- Input: n variables and collection of k-tuples of variables.
- Goal: Find global ordering for which max. fraction of input k-tuples are locally ordered according to a permutation in Π.

Ordering CSP

Ordering constraint of arity k = subset Π of $k!$ possible permutations

- MAS: $x_i < x_j$ $\quad\quad \Pi = \{12\}$
- Betweenness: x_j between x_i and x_ℓ.
  $\Pi = \{123, 321\}$ applied to triple (x_i, x_j, x_ℓ).

Instance of ordering kCSP Π:

- Input: n variables and collection of k-tuples of variables.
- Goal: Find global ordering for which max. fraction of input k-tuples are locally ordered according to a permutation in Π.

Theorem (Charikar, G., Håstad, Manokaran)

Every *ordering CSP is approximation resistant.*

- *UG-hard to distinguish* $(1-\varepsilon)$*-satisfiable instances from at most* $\frac{|\Pi|}{k!} + \delta$*-satisfiable instances, for any* $\varepsilon, \delta > 0$.

Summary

- Lot of progress on approximability of CSPs, both from algorithms and hardness side.
- Natural semidefinite programming relaxation + suitable rounding $\Rightarrow$ best known approximation algorithms for all CSPs.
    - In fact, achieves *the optimal* approximation ratio, under the Unique Games conjecture.
- Many unconditional tight hardness results also known
    - Show approximation resistance of several CSPs
    - A 2CSP called Label Cover is the canonical starting point, of which Unique Games is a particularly nice special case
    - Reduction method: Long code + dictatorship testing.
- "Approximate polymorphisms" (with low influences) give an explanation for the source of a CSP's approximation threshold.

Some Challenges

1. Prove or disprove the Unique Games conjecture.
2. Approximability of *satisfiable* CSPs?
3. Classification of approximation resistant CSPs?