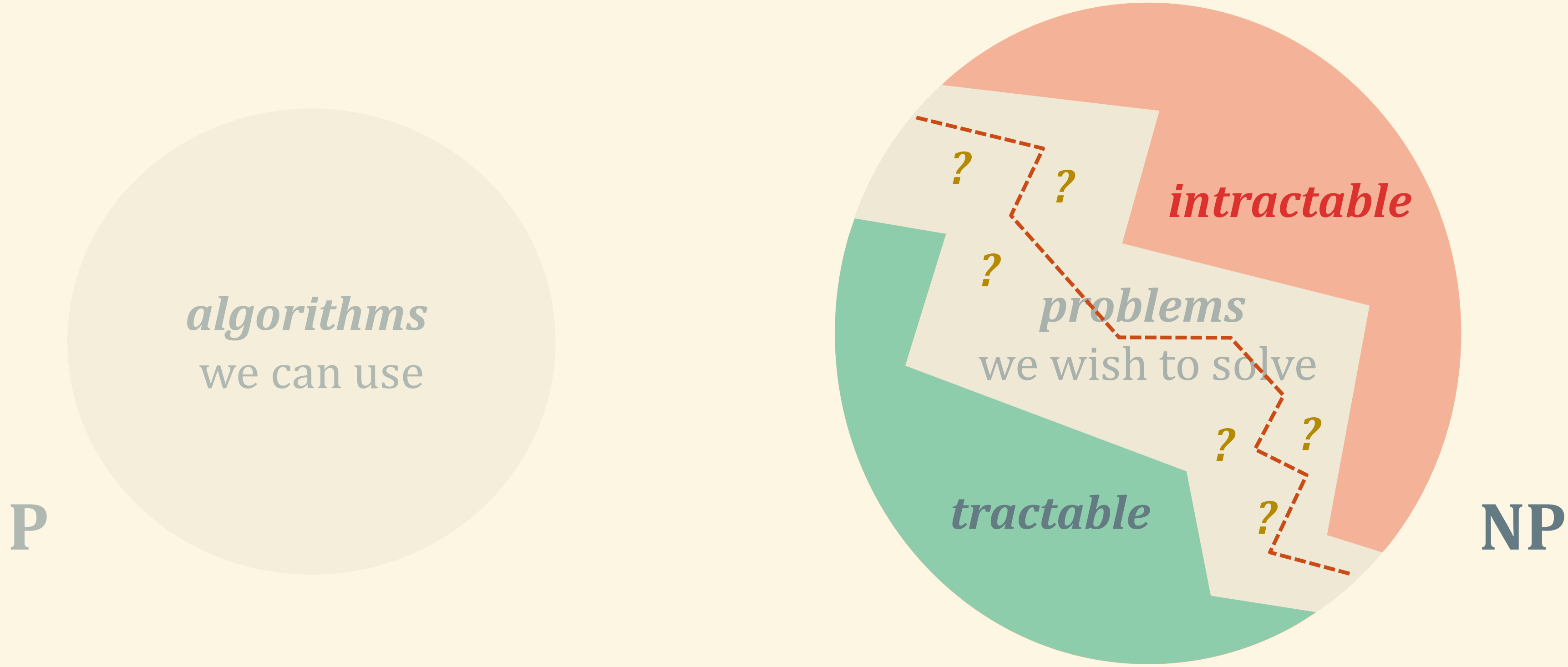
Unique Games, Sum of Squares, and the Quest for Optimal Algorithms

David Steurer
Cornell

Max Planck Institute Informatik, Saarbrücken, July 2014

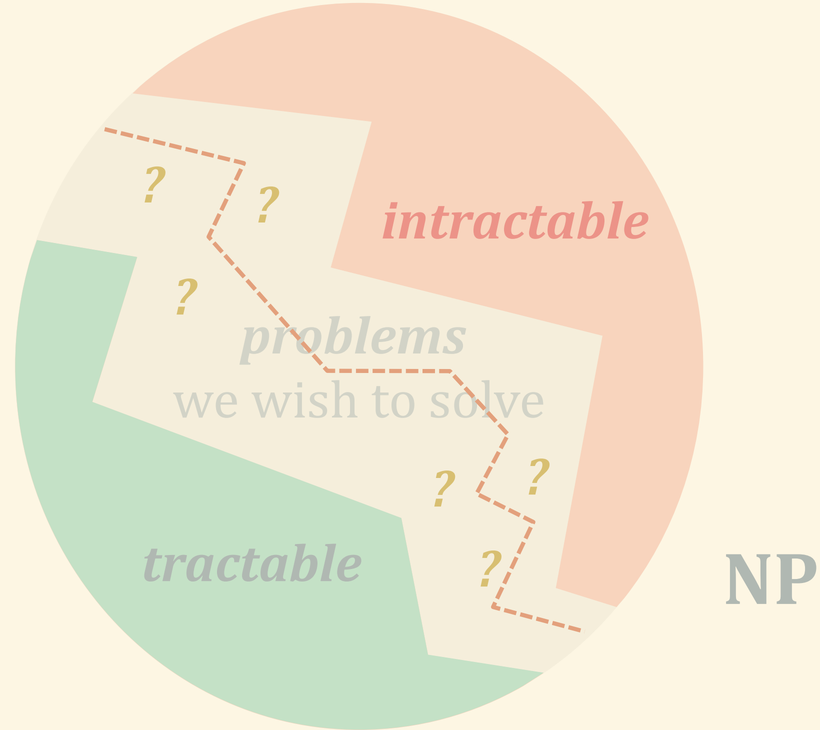

algorithms
we can use
P
problems
we wish to solve
NP


algorithms
we can use
P
intractable
?
?
?
problems
we wish to solve
?
?
?
tractable
NP

why meta-algorithms?

→ problems often change / negotiable
(problem-specific methods of limited use)

→ do not require domain knowledge



theory	***practice***
strong guarantees	weak guarantees
problem-specific	*meta-algorithms*

apply to wide range of problems

examples: gradient descent, EM, belief propagation, SAT solvers;

(all local-search based heuristics)

why meta-algorithms?

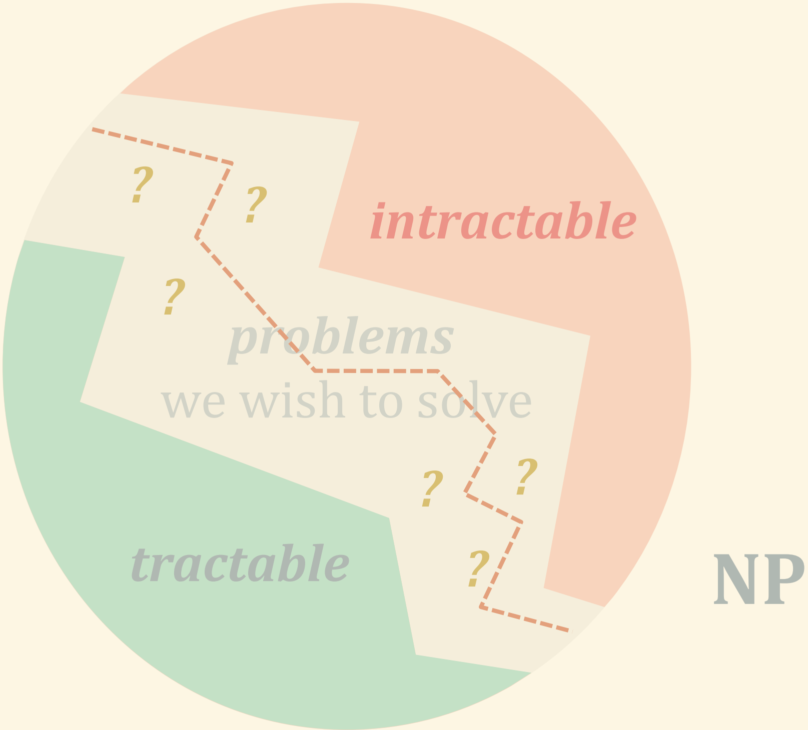
→ problems often change / negotiable
(problem-specific methods of limited use)

→ do not require domain knowledge

vision: unified theory based on new meta-algorithms (non-local)

apply to equally wide range of problems

best-possible guarantees among all efficient algorithms



theory	***practice***
strong guarantees	weak guarantees
problem-specific	*meta-algorithms*

apply to wide range of problems

examples: gradient descent, EM, belief propagation, SAT solvers;

(all local-search based heuristics)

why meta-algorithms?

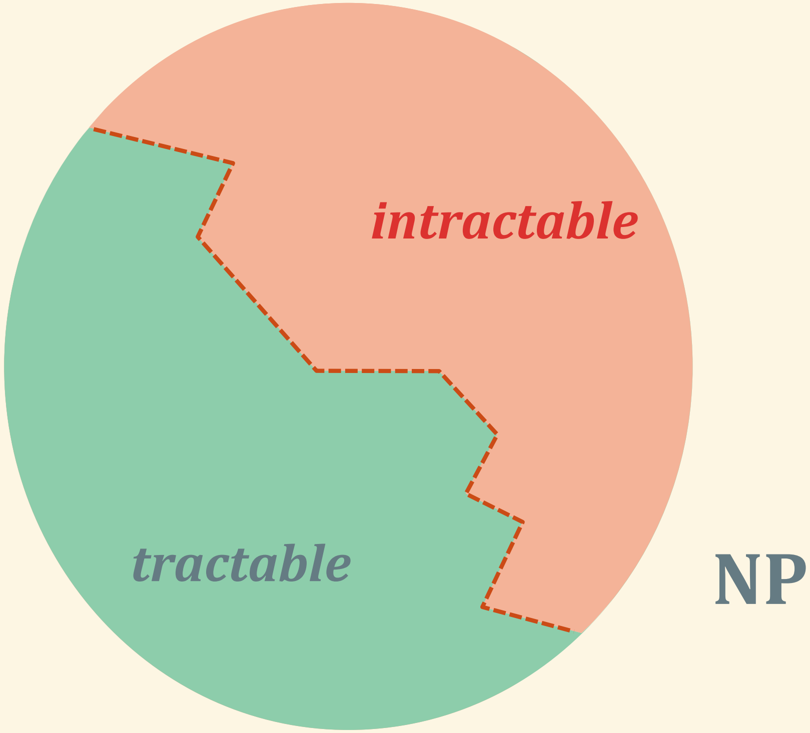
→ problems often change / negotiable
(problem-specific methods of limited use)

→ do not require domain knowledge

vision: unified theory based on
new meta-algorithms (non-local)

apply to equally
wide range of problems

best-possible guarantees
among all efficient algorithms



theory	***practice***
strong guarantees	weak guarantees
problem-specific	*meta-algorithms*

apply to wide range of problems

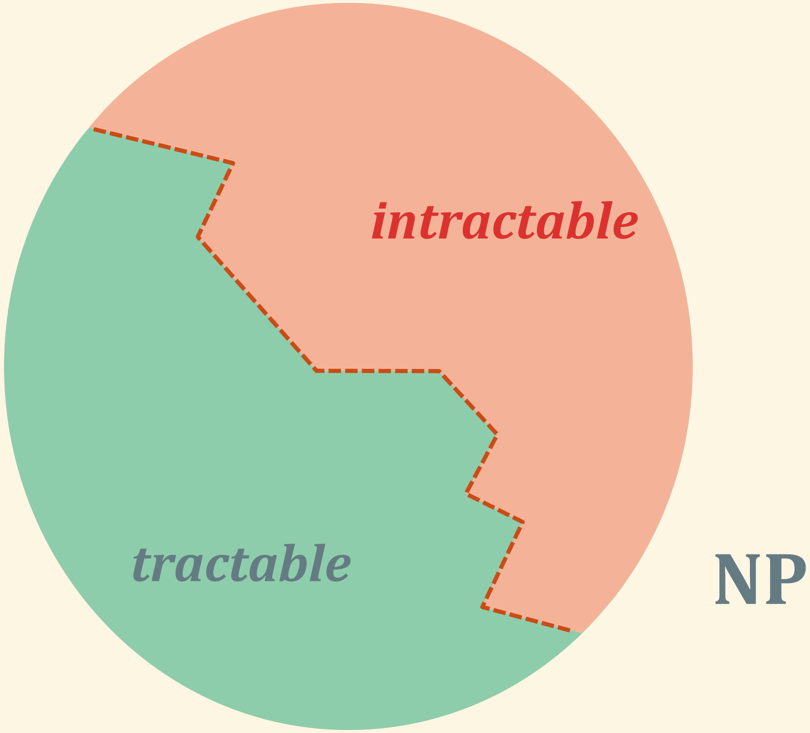
examples: gradient descent, EM,
belief propagation, SAT solvers;

(all local-search based heuristics)

vision: unified theory based on
new meta-algorithms (non-local)

apply to equally
wide range of problems

best-possible guarantees
among all efficient algorithms

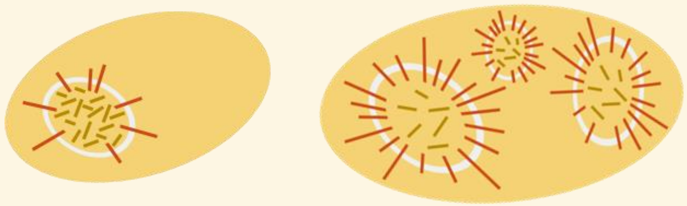


Unique Games Conjecture (UGC)

sum-of-squares (SOS) method

Unique Games Conjecture (UGC)

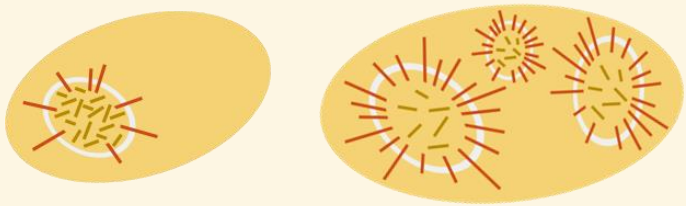
… asserts intractability of detecting small "communities" in networks



sum-of-squares (SOS) method

Unique Games Conjecture (UGC)

... asserts intractability of detecting small "communities" in networks



sum-of-squares (SOS) method

... greatly generalizes algorithms that UGC predicts to be optimal

predictions of UGC

unified theory for optimization problems around *concrete meta-algorithm*
(based on rounding convex relaxations)

ongoing works

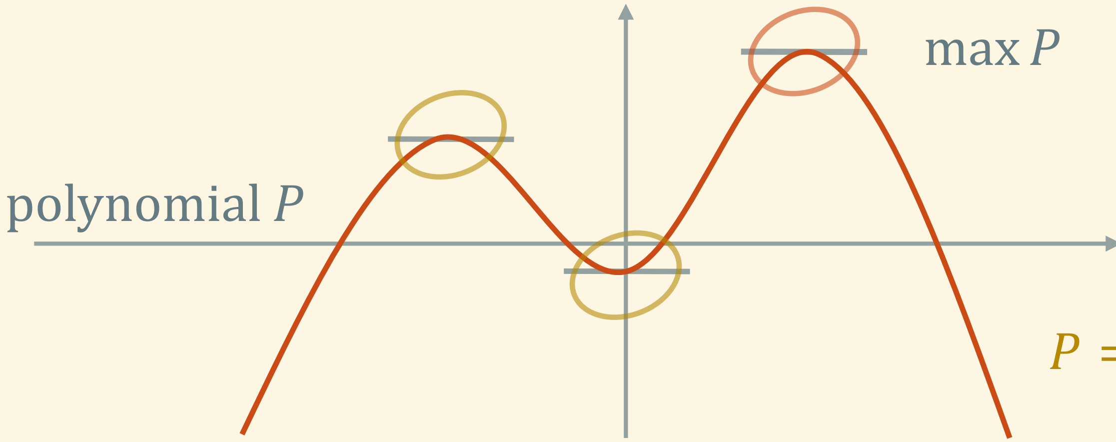
first evidence: UGC-theory false
& SOS right basis for unified theory

***challenge:* which *candidate theory* is correct?**

bigger picture:

gain new insights at frontier of *tractability & intractability*

already: applications to *quantum information, machine learning, ...*



$$P = 1 - (x-1)^2(x+1)^2 - \frac{1}{8}(x-1)^2$$

Newton's approach

search through local optima
(local-search based heuristics)

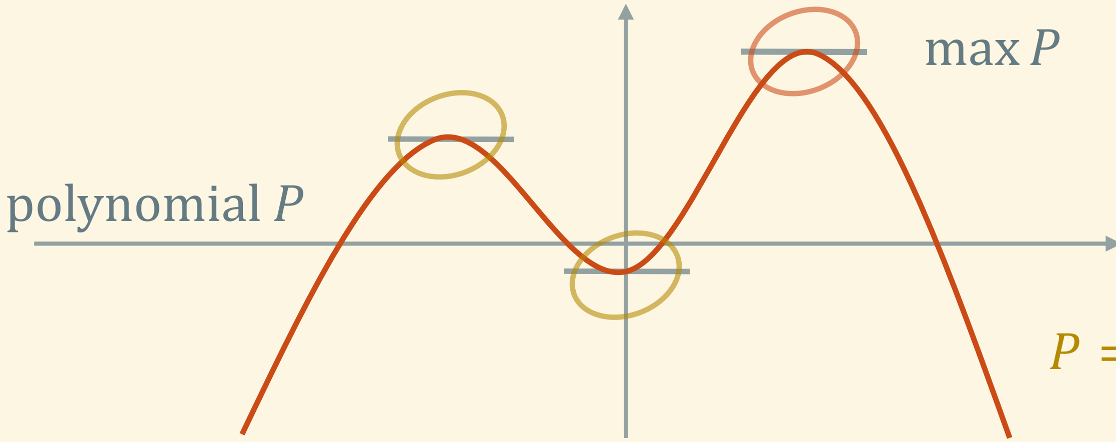
but: non-convex polynomials can have
exponential number of bad local optima

Hilbert's approach

decompose function
into simple pieces

UGC-optimal algorithms:
very restricted special case

sum-of-squares method:
find decomposition efficiently
whenever a small one exists



$$P = 1 - (x-1)^2(x+1)^2 - \frac{1}{8}(x-1)^2$$

Hilbert's approach

decompose function
into simple pieces

UGC-optimal algorithms:
very restricted special case

sum-of-squares method:
find decomposition efficiently
whenever a small one exists

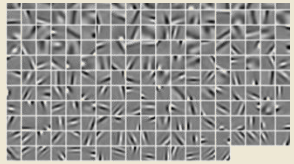
concrete strength of SOS method:

SOS works if *not too many* ***global peaks***

(local-search based algorithms can work
only if *not too many* ***local optima***)

unified theory:
UGC vs. SOS

learning sparse dictionaries



previous works: only very sparse models

SOS method: full sparsity range; *novel, general approach to unsupervised learning*

average-case complexity

[Barak-Kindler-**S.**'13]

intermediate complexity

[Arora-Barak-**S.**'10
Dinur-**S.**'13]

machine learning

[Barak-Kelner-**S.**'14b]

complexity

unified theory: UGC vs. SOS

learning

math

physics

functional analysis

[BBHKZ***S***'13]

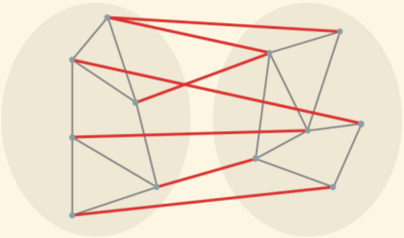
high-dimensional data

[Barak-Kelner-**S.**'14a]

quantum information

[BBHKZ***S***'13]

example: max cut

$x_i = 1$  $x_i = -1$

combinatorial viewpoint: given undirected graph G, bipartition vertex set to cut as many edges as possible

polynomial viewpoint: given polynomial $L_G(x) = \sum_{ij \in E(G)} \frac{1}{4}(x_i - x_j)^2$, find maximum over $x \in \{\pm 1\}^{V(G)}$ *(hypercube)*

how to certify upper bound c on maximum?

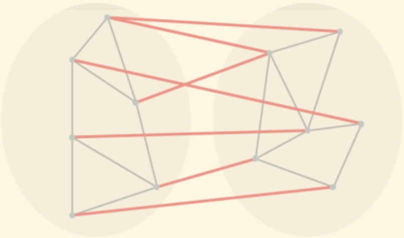
$R_1^2 + \cdots + R_t^2$

decompose $c - L_G$ as *sum of squares* of polynomials
plus quadratic *polynomial vanishing over hypercube*

$\alpha_1 \cdot (X_1^2 - 1) + \cdots + \alpha_n \cdot (X_n^2 - 1)$

} n^2-size semidefinite program

example: max cut



combinatorial viewpoint: given undirected graph G, bipartition vertex set to cut as many edges as possible

polynomial viewpoint: given polynomial $L_G(x) = \sum_{ij \in E(G)} \frac{1}{4}\left(x_i - x_j\right)^2$, find maximum over $x \in \{\pm 1\}^{V(G)}$ *(hypercube)*

how to certify upper bound c on maximum?

$R_1^2 + \cdots + R_t^2$

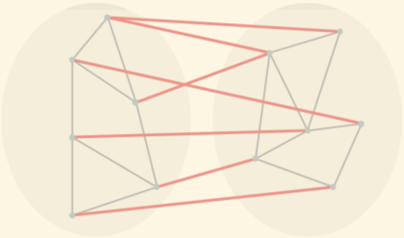
decompose $c - L_G$ as *sum of squares* of polynomials
plus quadratic *polynomial vanishing over hypercube*

$\alpha_1 \cdot (X_1^2 - 1) + \cdots + \alpha_n \cdot (X_n^2 - 1)$

} n^2-size semidefinite program

sum-of-squares polynomial
↔ positive-semidefinite coefficient matrix

example: max cut



combinatorial viewpoint: given undirected graph G, bipartition vertex set to cut as many edges as possible

polynomial viewpoint: given polynomial $L_G(x) = \sum_{ij \in E(G)} \frac{1}{4}(x_i - x_j)^2$, find maximum over $x \in \{\pm 1\}^{V(G)}$ *(hypercube)*

how to certify upper bound c on maximum?

$R_1^2 + \cdots + R_t^2$

decompose $c - L_G$ as *sum of squares* of polynomials plus quadratic *polynomial vanishing over hypercube* } n^2-size semidefinite program

$\alpha_1 \cdot (X_1^2 - 1) + \cdots + \alpha_n \cdot (X_n^2 - 1)$

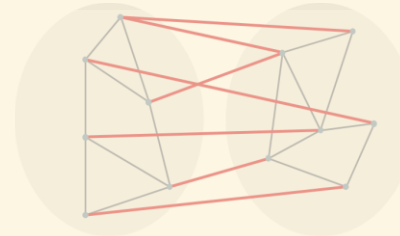
Goemans-Williamson bound: either decomposition exists or max. $\geq 0.868 \cdot c$
(→ current best known approximation guarantee)

spectral method: add restriction $\alpha_1 = \cdots = \alpha_n$ → largest Laplacian eigenvalue

basic object in spectral graph theory

example: max cut

$x_i = 1$



$x_i = -1$

combinatorial viewpoint: given undirected graph G, bipartition vertex set to cut as many edges as possible

polynomial viewpoint: given polynomial $L_G(x) = \sum_{ij \in E(G)} \frac{1}{4}(x_i - x_j)^2$, find maximum over $x \in \{\pm 1\}^{V(G)}$ *(hypercube)*

how to certify upper bound c on maximum?

$R_1^2 + \cdots + R_t^2$

decompose $c - L_G$ as *sum of squares* of polynomials
plus ~~quadratic~~ degree-k *polynomial vanishing over hypercube*

$\alpha_1 \cdot (X_1^2 - 1) + \cdots + \alpha_n \cdot (X_n^2 - 1)$

} ~~n^2~~ n^k-size semidefinite program

degree-n bound is exact *(interpolate $\sqrt{c - L_G}$ as degree-n polynomial over hypercube)*

does degree-$n^{o(1)}$ bound improve over GW in worst-case? *(would refute UGC)*

for every candidate graph construction, degree-16 bound improves over GW

[Brandao-Barak-Harrow-Kelner-S.-Zhou]

multivariate polynomials
$P_1, \dots, P_m \in \mathbb{R}[x_1, \dots, x_n]$

system of equations
$\mathcal{E} = \{P_1 = 0, \dots, P_m = 0\}$

when is $\mathcal{E}$ unsatisfiable over $\mathbb{R}^n$?

idea: derive "obviously unsatisfiable equation" from $\mathcal{E}$

sum-of-squares (SOS) refutation of $\mathcal{E}$

vanishes on $\mathcal{E}$ → $Q_1 \cdot P_1 + \cdots + Q_m \cdot P_m$
$= 1 + R_1^2 + \cdots + R_t^2$ ← *positive over $\mathbb{R}^n$*

intuitive proof system: many common inequalities have proofs in this form, e.g., Cauchy-Schwarz, Hölder, ℓ_p-triangle inequalities

Real Nullstellensatz *linear case:* Gaussian elimination, Farkas lemma [Artin, Krivine, Stengle]
every polynomial system is either satisfiable over $\mathbb{R}^n$ or SOS refutable

SOS method: $n^{O(k)}$-time algorithm to find SOS refutation with degrees $\leq k$ if one exists (uses SDP) [Shor, Nesterov, Parrilo, Lasserre] optimization (e.g., MAX CUT) maximize P_0 over $\{P_1 = 0, \dots, P_m = 0\}$

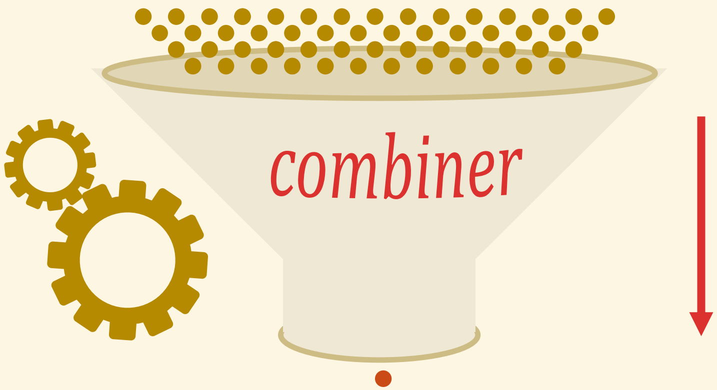
v-vs-v' approximation: *given*: sat. system $\{P_0 = v, P_1 = 0, \dots, P_m = 0\}$
find: solution to $\{P_0 = v', P_1 = 0, \dots, P_m = 0\}$

[Barak-Kelner-S.'14]

claim: SOS reduces approximation in time $n^{O(k)}$ to "deg.-k combining"

subset $\mathcal{X}$ of solutions to $\{P_0 \geq v, P_1 = 0, \dots, P_m = 0\}$

represented by all degree-k moments of $\mathcal{X}$, e.g., $\mathbb{E}_{\mathcal{X}} x_1 \cdots x_k$



use only properties of moments / solutions with degree-k SOS proofs

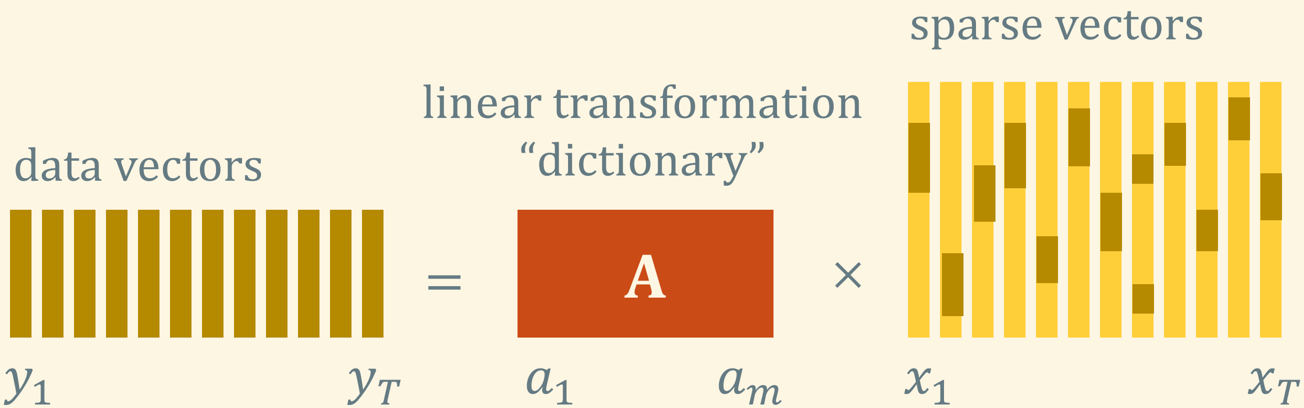
single solution to $\{P_0 \geq v', P_1 = 0, \dots, P_m = 0\}$

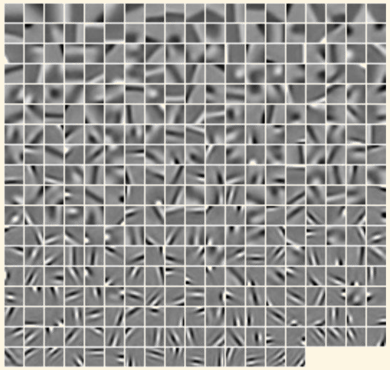
pseudo-moments

"proof:" obstructions to degree-k SOS refutations *indistinguishable* from deg.-k moments *with respect to deg.-k SOS arguments*

dictionary learning (aka sparse coding)

application: machine learning (*feature extraction*)
neuroscience (*model for visual cortex*)





example: dictionary for natural images

[Olshausen-Fields'96]

$a_1, \ldots, a_m$ unknown unit vectors in isotropic position
$x_1, \ldots, x_t$ are i.i.d. samples from unknown "nice" distr. over sparse vectors
(only small correlations between coord's)

goal: given data vectors $y_1, \ldots, y_T$, reconstruct A

[Arora-Ge-Moitra, Agarwal-Anandkumar-Jain-Netrapalli-Tandon]

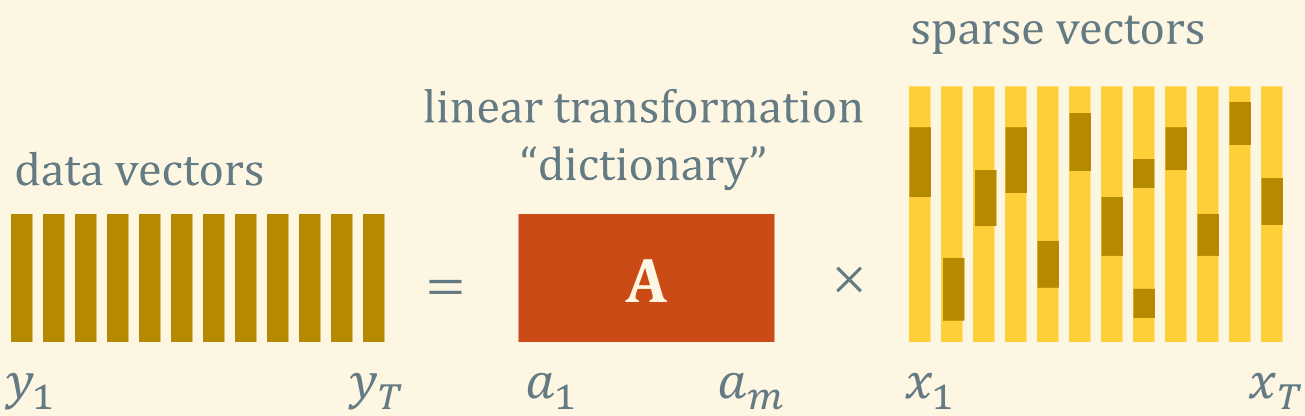
previous methods (local search): only very sparse vectors, up to $\sqrt{n}$ non-zeros

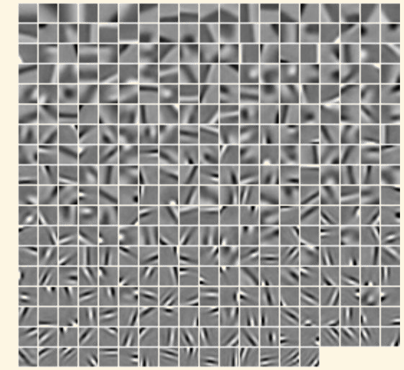
[Barak-Kelner-S.'14]

sum-of-squares method: full sparsity range, up to constant fraction non-zeros
(quasipolynomial-time for $o(1)$; polynomial-time for $n^{-\varepsilon}$)

dictionary learning (aka sparse coding)

application: machine learning (*feature extraction*)
neuroscience (*model for visual cortex*)





example: dictionary for natural images

[Olshausen-Fields’96]

$a_1, \dots, a_m$ unknown unit vectors in isotropic position

$x_1, \dots, x_t$ are i.i.d. samples from unknown “nice” distr. over sparse vectors

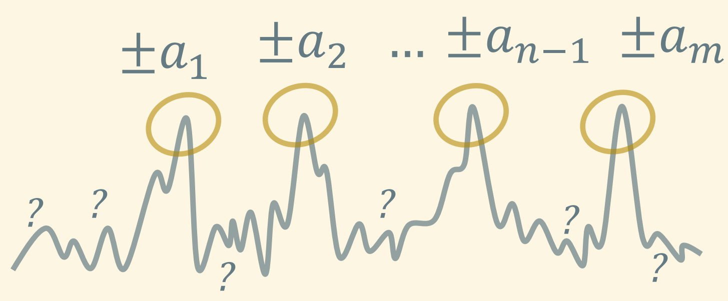
(only small correlations between coord’s)

theorem: [Barak-Kelner-S.’14]

suppose $m = O(n)$ and correlations between coord’s small enough

then, $O(\log n)$-SOS can recover set $A^* \approx \{\pm a_1, \dots, \pm a_m\}$ in Hausdorff distance

theorem:
suppose $m = O(n)$ and correlations between coord's small enough
then, $O(\log n)$-SOS can recover set $A^* \approx \{\pm a_1, \dots, \pm a_m\}$ in Hausdorff distance



1. construct polynomial $P_0(u) = \frac{1}{T}\sum_t \langle y_t, u\rangle^4$ from data vectors

can show: global optima of P_0 correspond to $\pm a_1, \dots, \pm a_m$
(but no control over local optima of P_0)

low-degree SOS proof

2. compute global optima of P_0 …

in general: NP-hard problem *(even approximately)*

approach: use SOS method and degree-$O(\log m)$ combiner

works because every solution set clustered around $\leq m$ points

conclusions

polynomial optimization: often easy when global optima unique (occurs naturally for recovery problems)

unsupervised learning: higher-degree SOS gives better guarantees for recovering hidden structures

low-degree combiner: general way to make proofs into algorithms

UGC and SOS method: opportunity for unified approach to algorithm design for hard problems

thank you!

planted sparse vector recovery

one k-sparse vector among d random vectors

arbitrary / random basis of span W of these vectors

k non-zeros

$$a_0 = \frac{1}{\sqrt{k}}(0, \ldots, 0, \pm 1, \ldots, \pm 1) \in \mathbb{R}^n$$
$$a_1 = \frac{1}{\sqrt{n}}(\pm 1, \ldots, \pm 1)$$
$$\vdots$$
$$a_d = \frac{1}{\sqrt{n}}(\pm 1, \ldots, \pm 1)$$

$$y_0 = (\pm 1, \ldots, \pm 1) \in \mathbb{R}^n$$
$$y_1 = (\pm 1, \ldots, \pm 1)$$
$$\vdots$$
$$y_d = (\pm 1, \ldots, \pm 1)$$

goal: given $y_0, \ldots, y_d$, recover vector $a^* \approx \pm a_0$ [Demanet-Hand'13]

idealized inference problem; subproblem for dictionary learning proxy for sparsity: if vector x is k-sparse then $\frac{\|x\|_\infty}{\|x\|_1} \geq \frac{1}{k}$ and $\frac{\|x\|_4^4}{\|x\|_2^4} \geq \frac{1}{k}$

previous best algorithm

find vector $x \in W$ with maximum ℓ_∞/ℓ_1 ratio [Spielman-Wang-Wright, Demanet-Hand]
(exact using linear programming)

recovers $x \approx \pm a_0$ if and only if $\frac{k}{n} \ll 1/\sqrt{d}$

idea: use ℓ_4/ℓ_2 ratio instead $\left\{\|x\|_4^4 = \frac{1}{k}, \|x\|_2^2 = 1, x \in W\right\}$

good: better proxy for sparsity ($d \ll \sqrt{n}$); system of polynomial equations

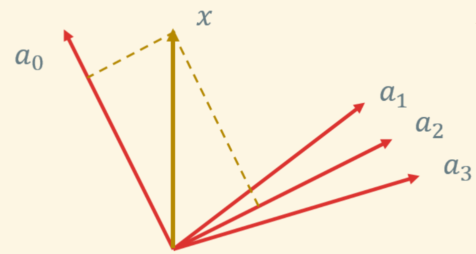
bad: NP-hard to solve exactly; somewhat hard to approximate (SSE-hard)

here: SOS works for this problem (exploit randomness in W)

combining problem: given $\mathcal{X} \subseteq \left\{ \|x\|_4^4 = \frac{1}{k}, \|x\|_2^2 = 1, x \in W \right\}$, find $x^* \approx \pm a_0$

claim: set $\mathcal{X}$ concentrated around planted vector (up to sign)

arguments used in analysis



ℓ_4 triangle inequality

ℓ_4/ℓ_2 ratio bound for random subspaces for $d \ll \sqrt{n}$

deg.-4 SOS proofs

[Barak-Brandao-Harrow-Kelner-S.-Zhou]

combiner: sample Gaussian distr. $\gamma_{\mathcal{X}}$ with same deg.-2 moments as $\mathcal{X}$

$\mathbb{E}_{\gamma_{\mathcal{X}}} xx^\top = \mathbb{E}_{\mathcal{X}} xx^\top \approx a_0 a_0^\top$ → random sample from $\gamma_{\mathcal{X}}$ is close to $\pm a_0$

deg.-2 SOS proof that $\mathrm{Cov}(X) \succcurlyeq 0$

→ *get algorithm via SOS*