Exactness of Relative Entropy Relaxations for Signomial Optimization

Riley Murray

California Institute of Technology

July 6, 2018

Joint work with Venkat Chandrasekaran and Adam Wierman (Caltech)

Preliminary Definitions

Caltech

A **signomial** is a function parameterized by a matrix $\boldsymbol{\alpha}$ and a vector $\boldsymbol{c}$, taking values

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} c_i \exp \boldsymbol{\alpha}_i^{\mathsf{T}} \boldsymbol{x}$$

where $\boldsymbol{\alpha}_i \in \mathbb{R}^n$ are the columns of $\boldsymbol{\alpha}$.

Preliminary Definitions

Caltech

A **signomial** is a function parameterized by a matrix $\boldsymbol{\alpha}$ and a vector $\boldsymbol{c}$, taking values

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} c_i \exp \boldsymbol{\alpha}_i^{\mathsf{T}} \boldsymbol{x}$$

where $\boldsymbol{\alpha}_i \in \mathbb{R}^n$ are the columns of $\boldsymbol{\alpha}$.

"$f = \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})$" means that f takes values as above.

Preliminary Definitions

Caltech

A **signomial** is a function parameterized by a matrix $\boldsymbol{\alpha}$ and a vector $\boldsymbol{c}$, taking values

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} c_i \exp \boldsymbol{\alpha}_i^{\mathsf{T}} \boldsymbol{x}$$

where $\boldsymbol{\alpha}_i \in \mathbb{R}^n$ are the columns of $\boldsymbol{\alpha}$.

"$f = \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})$" means that f takes values as above.

The **relative entropy cone** is

$$K_{\mathrm{rel}} = \mathsf{cl}\{\boldsymbol{v} : \ v_1 \ln(v_1/v_2) \leq v_3, \ \boldsymbol{v}_{\setminus 3} \geq \mathbf{0}\}.$$

K_{rel} has a 4-self-concordant barrier.

Signomial Programming

Caltech

Minimize signomials involving exponents $\boldsymbol{\alpha}$.

Assume $\boldsymbol{\alpha}_1 = \mathbf{0}$, so $\mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c}) - \gamma \equiv \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c} - \gamma \boldsymbol{e}_1)$.

Signomial Programming

Caltech

Minimize signomials involving exponents $\boldsymbol{\alpha}$.

Assume $\boldsymbol{\alpha}_1 = \mathbf{0}$, so $\mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c}) - \gamma \equiv \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c} - \gamma \boldsymbol{e}_1)$.

Unconstrained.

Objective $f = \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})$.

Compute $f^\star = \inf\{f(\boldsymbol{x}) \ : \ \boldsymbol{x} \text{ in } \mathbb{R}^n\}$.

Signomial Programming

Caltech

Minimize signomials involving exponents $\boldsymbol{\alpha}$.

Assume $\boldsymbol{\alpha}_1 = \mathbf{0}$, so $\mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c}) - \gamma \equiv \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c} - \gamma \boldsymbol{e}_1)$.

Unconstrained.

Objective $f = \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})$.

Compute $f^\star = \inf\{f(\boldsymbol{x}) \ : \ \boldsymbol{x} \text{ in } \mathbb{R}^n\}$.

Constrained.

Constraint signomials $g_i = \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{g}_i)$ for i in $[k]$.

Compute $\inf\{f(\boldsymbol{x}) : \boldsymbol{x} \text{ in } \mathbb{R}^n \text{ satisfies } g(\boldsymbol{x}) \geq \mathbf{0}\}$.

Call the above value "$(f, g)^\star$."

Nonnegativity and Optimization

Caltech

Nonnegativity and Optimization

Caltech

Definition. The cone of coefficients for nonnegative signomials involving exponents $\boldsymbol{\alpha}$ is

$$C_{\mathrm{NNG}}(\boldsymbol{\alpha}) \doteq \{\boldsymbol{c} \ : \ \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})(\boldsymbol{x}) \geq 0 \text{ for all } \boldsymbol{x} \text{ in } \mathbb{R}^n\}.$$

Nonnegativity and Optimization

Caltech

Definition. The cone of coefficients for nonnegative signomials involving exponents $\boldsymbol{\alpha}$ is

$$C_{\mathrm{NNG}}(\boldsymbol{\alpha}) \doteq \{\boldsymbol{c} \ : \ \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})(\boldsymbol{x}) \geq 0 \text{ for all } \boldsymbol{x} \text{ in } \mathbb{R}^n\}.$$

Given a signomial $f = \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})$, we have

$$\begin{aligned} f^\star &= \sup\{\gamma : f(\boldsymbol{x}) \geq \gamma \text{ for all } \boldsymbol{x} \text{ in } \mathbb{R}^n\} \\ &= \sup\{\gamma : f - \gamma \text{ nonnegative } \} \\ &= \sup\{\gamma : \boldsymbol{c} - \gamma \boldsymbol{e}_1 \text{ in } C_{\mathrm{NNG}}(\boldsymbol{\alpha})\} \end{aligned}$$

Nonnegativity and Optimization

Caltech

Definition. The cone of coefficients for nonnegative signomials involving exponents $\boldsymbol{\alpha}$ is

$$C_{\mathrm{NNG}}(\boldsymbol{\alpha}) \doteq \{\boldsymbol{c} \ : \ \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})(\boldsymbol{x}) \geq 0 \text{ for all } \boldsymbol{x} \text{ in } \mathbb{R}^n\}.$$

Given a signomial $f = \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})$, we have

$$\begin{aligned} f^\star &= \sup\{\gamma : f(\boldsymbol{x}) \geq \gamma \text{ for all } \boldsymbol{x} \text{ in } \mathbb{R}^n\} \\ &= \sup\{\gamma : f - \gamma \text{ nonnegative } \} \\ &= \sup\{\gamma : \boldsymbol{c} - \gamma \boldsymbol{e}_1 \text{ in } C_{\mathrm{NNG}}(\boldsymbol{\alpha})\} \end{aligned}$$

– where the last equation uses the earlier assumption $\boldsymbol{\alpha}_1 = \mathbf{0}$.

The SAGE Cone

Caltech

Definition. A nonnegative signomial with at most one negative coefficient is an **AM/GM Exponential**, or an **AGE function**.

The SAGE Cone

Caltech

Definition. A nonnegative signomial with at most one negative coefficient is an **AM/GM Exponential**, or an **AGE function**.

We can construct cones of coefficients for AM/GM Exponentials

$$C_{\mathrm{AGE}}(\boldsymbol{\alpha}, k) \doteq \{\boldsymbol{c} \ : \ \boldsymbol{c}_{\setminus k} \geq \mathbf{0} \text{ and } \boldsymbol{c} \text{ in } C_{\mathrm{NNG}}(\boldsymbol{\alpha})\}.$$

Each cone $C_{\mathrm{AGE}}(\boldsymbol{\alpha}, k)$ is relative-entropy representable with barrier complexity $O(m)$.

The SAGE Cone

Caltech

Definition. A nonnegative signomial with at most one negative coefficient is an **AM/GM Exponential**, or an **AGE function**.

We can construct cones of coefficients for AM/GM Exponentials

$$C_{\mathrm{AGE}}(\boldsymbol{\alpha}, k) \doteq \{\boldsymbol{c} \ : \ \boldsymbol{c}_{\setminus k} \geq \mathbf{0} \text{ and } \boldsymbol{c} \text{ in } C_{\mathrm{NNG}}(\boldsymbol{\alpha})\}.$$

Each cone $C_{\mathrm{AGE}}(\boldsymbol{\alpha}, k)$ is relative-entropy representable with barrier complexity $O(m)$. *I won't show that representation in this talk!*

The SAGE Cone

Caltech

Definition. A nonnegative signomial with at most one negative coefficient is an **AM/GM Exponential**, or an **AGE function**.

We can construct cones of coefficients for AM/GM Exponentials

$$C_{\mathrm{AGE}}(\boldsymbol{\alpha}, k) \doteq \{\boldsymbol{c} \ : \ \boldsymbol{c}_{\setminus k} \geq \mathbf{0} \text{ and } \boldsymbol{c} \text{ in } C_{\mathrm{NNG}}(\boldsymbol{\alpha})\}.$$

Each cone $C_{\mathrm{AGE}}(\boldsymbol{\alpha}, k)$ is relative-entropy representable with barrier complexity $O(m)$. *I won't show that representation in this talk!*

Take sums of AGE functions to obtain **SAGE functions**.

$$C_{\mathrm{SAGE}}(\boldsymbol{\alpha}) = \sum_{i=1}^{m} C_{\mathrm{AGE}}(\boldsymbol{\alpha}, i)$$

SAGE Relaxations

Caltech

Replace $C_{\mathrm{NNG}}(\boldsymbol{\alpha})$ with $C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$.

SAGE Relaxations

Caltech

Replace $C_{\mathrm{NNG}}(\boldsymbol{\alpha})$ with $C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$.

Also consider the **dual cone** $C_{\mathrm{SAGE}}(\boldsymbol{\alpha})^{\dagger}$.

SAGE Relaxations

Caltech

Replace $C_{\mathrm{NNG}}(\boldsymbol{\alpha})$ with $C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$.

Also consider the **dual cone** $C_{\mathrm{SAGE}}(\boldsymbol{\alpha})^{\dagger}$.

Primal and dual SAGE relaxations for signomial minimization.

$$f_{\mathrm{SAGE}} = \sup\{\ \gamma\ :\ \boldsymbol{c} - \gamma \boldsymbol{e}_1 \text{ in } C_{\mathrm{SAGE}}(\boldsymbol{\alpha})\}$$

$$f_{\mathrm{SAGE}} = \inf\{\ \boldsymbol{c}^{\mathsf{T}}\boldsymbol{v}\ :\ \boldsymbol{v} \text{ in } C_{\mathrm{SAGE}}(\boldsymbol{\alpha})^{\dagger} \text{ has } \boldsymbol{e}_1^{\mathsf{T}}\boldsymbol{v} = 1\}$$

SAGE Relaxations

Caltech

Replace $C_{\mathrm{NNG}}(\boldsymbol{\alpha})$ with $C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$.

Also consider the **dual cone** $C_{\mathrm{SAGE}}(\boldsymbol{\alpha})^{\dagger}$.

Primal and dual SAGE relaxations for signomial minimization.

$$f_{\mathrm{SAGE}} = \sup\{\ \gamma\ :\ \boldsymbol{c} - \gamma \boldsymbol{e}_1 \text{ in } C_{\mathrm{SAGE}}(\boldsymbol{\alpha})\}$$

$$f_{\mathrm{SAGE}} = \inf\{\ \boldsymbol{c}^{\mathsf{T}}\boldsymbol{v}\ :\ \boldsymbol{v} \text{ in } C_{\mathrm{SAGE}}(\boldsymbol{\alpha})^{\dagger} \text{ has } \boldsymbol{e}_1^{\mathsf{T}}\boldsymbol{v} = 1\}$$

The dual SAGE relaxation for signomial *programming*.

$$(f,g)_{\mathrm{SAGE}} = \inf\{\ \boldsymbol{c}^{\mathsf{T}}\boldsymbol{v}\ : \boldsymbol{v} \text{ in } C_{\mathrm{SAGE}}(\boldsymbol{\alpha})^{\dagger} \text{ has } \boldsymbol{e}_1^{\mathsf{T}}\boldsymbol{v} = 1,$$
$$\text{and satisfies } \boldsymbol{g}_i^{\mathsf{T}}\boldsymbol{v} \geq 0 \text{ for all } i \text{ in } [k]\}$$

We've defined ...

Caltech

We've defined ...

Caltech

1. Signomials and the relative entropy cone.

$$f = \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c}).\ K_{\mathrm{rel}} \subset \mathbb{R}^3.$$

We've defined ...

Caltech

1. Signomials and the relative entropy cone.

   $f = \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})$. $K_{\mathrm{rel}} \subset \mathbb{R}^3$.

2. A standard form for signomial programming.

   Objective f and constraints $\{g_i\}_{i=1}^k$ over common $\boldsymbol{\alpha}$.

   Assume $\boldsymbol{\alpha}_1 = \mathbf{0}$.

We've defined ...

Caltech

1. Signomials and the relative entropy cone.

   $f = \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})$. $K_{\mathrm{rel}} \subset \mathbb{R}^3$.

2. A standard form for signomial programming.

   Objective f and constraints $\{g_i\}_{i=1}^k$ over common $\boldsymbol{\alpha}$.

   Assume $\boldsymbol{\alpha}_1 = \mathbf{0}$.

3. The SAGE cone.

   $C_{\mathrm{SAGE}}(\boldsymbol{\alpha}) \subset C_{\mathrm{NNG}}(\boldsymbol{\alpha})$

We've defined ...

Caltech

1. Signomials and the relative entropy cone.

   $f = \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})$. $K_{\mathrm{rel}} \subset \mathbb{R}^3$.

2. A standard form for signomial programming.

   Objective f and constraints $\{g_i\}_{i=1}^k$ over common $\boldsymbol{\alpha}$.

   Assume $\boldsymbol{\alpha}_1 = \mathbf{0}$.

3. The SAGE cone.

   $C_{\mathrm{SAGE}}(\boldsymbol{\alpha}) \subset C_{\mathrm{NNG}}(\boldsymbol{\alpha})$

4. SAGE relaxations for signomial optimization.

   f_{SAGE}, $(f, g)_{\mathrm{SAGE}}$

Solving SAGE Relaxations

Caltech

You will need a relative entropy solver. Here are your options:

Name	SCS	ECOS	*MOSEK 9
Type	first order	interior point	interior point
Multithreaded	Yes	No	Yes
GPU support	Yes	No	No
Open source	Yes	Yes	No

Solving SAGE Relaxations

Caltech

You will need a relative entropy solver. Here are your options:

Name	SCS	ECOS	*MOSEK 9
Type	first order	interior point	interior point
Multithreaded	Yes	No	Yes
GPU support	Yes	No	No
Open source	Yes	Yes	No

You will also need a way to **construct** the relaxations...

Signomials & SAGE in Python

Caltech

`https://github.com/rileyjmurray/sigpy`

- Adds SAGE support to `cvxpy`.
- Implements entire SAGE "heirarchy."
- Constrained and unconstrained SAGE relaxations.
- Explicit primal and dual formulations.
- Access to ECOS, SCS, and MOSEK 9 (once released).

Signomials & SAGE in Python

Caltech

Suppose you want to compute f_{SAGE} for

```
f = Signomial(alpha, c).
```

Signomials & SAGE in Python

Caltech

Suppose you want to compute f_{SAGE} for

```
f = Signomial(alpha, c).
```

This is trivial with provided helper functions

```
prob = sage_primal(f, level=0)
f_sage = prob.solve().
```

Signomials & SAGE in Python

Caltech

Suppose you want to compute f_{SAGE} for

```
f = Signomial(alpha, c).
```

This is trivial with provided helper functions

```
prob = sage_primal(f, level=0)
f_sage = prob.solve().
```

If you insist, you can do things from scratch

Signomials & SAGE in Python

Caltech

Suppose you want to compute f_{SAGE} for

```
f = Signomial(alpha, c).
```

This is trivial with provided helper functions

```
prob = sage_primal(f, level=0)
f_sage = prob.solve().
```

If you insist, you can do things from scratch

```
gamma = cvxpy.Variable()
```

Signomials & SAGE in Python

Caltech

Suppose you want to compute f_{SAGE} for

```
f = Signomial(alpha, c).
```

This is trivial with provided helper functions

```
prob = sage_primal(f, level=0)
f_sage = prob.solve().
```

If you insist, you can do things from scratch

```
gamma = cvxpy.Variable()
objective = cvxpy.Maximize(gamma)
```

Signomials & SAGE in Python

Caltech

Suppose you want to compute f_{SAGE} for

```
f = Signomial(alpha, c).
```

This is trivial with provided helper functions

```
prob = sage_primal(f, level=0)
f_sage = prob.solve().
```

If you insist, you can do things from scratch

```
gamma = cvxpy.Variable()
objective = cvxpy.Maximize(gamma)
constraints = relative_c_sage(f - gamma)
```

Signomials & SAGE in Python

Caltech

Suppose you want to compute f_{SAGE} for

```
f = Signomial(alpha, c).
```

This is trivial with provided helper functions

```
prob = sage_primal(f, level=0)
f_sage = prob.solve().
```

If you insist, you can do things from scratch

```
gamma = cvxpy.Variable()
objective = cvxpy.Maximize(gamma)
constraints = relative_c_sage(f - gamma)
prob = cvxpy.Problem(objective, constraints)
```

Signomials & SAGE in Python

Caltech

Suppose you want to compute f_{SAGE} for

```
f = Signomial(alpha, c).
```

This is trivial with provided helper functions

```
prob = sage_primal(f, level=0)
f_sage = prob.solve().
```

If you insist, you can do things from scratch

```
gamma = cvxpy.Variable()
objective = cvxpy.Maximize(gamma)
constraints = relative_c_sage(f - gamma)
prob = cvxpy.Problem(objective, constraints)
f_sage = prob.solve().
```

The Trouble in Analyzing SAGE

Caltech

Implementing $\boldsymbol{c} \in C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$ comes down to

1. $\boldsymbol{c} = \sum_{i=1}^{m} \boldsymbol{c}^{(i)}$,
2. $\boldsymbol{c}^{(i)} \in C_{\mathrm{AGE}}(\boldsymbol{\alpha}, i)$ for all i in $[m]$.

The Trouble in Analyzing SAGE

Caltech

Implementing $\boldsymbol{c} \in C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$ comes down to

1. $\boldsymbol{c} = \sum_{i=1}^{m} \boldsymbol{c}^{(i)}$,
2. $\boldsymbol{c}^{(i)} \in C_{\mathrm{AGE}}(\boldsymbol{\alpha}, i)$ for all i in $[m]$.

Constraints for $\boldsymbol{c}^{(i)} \in C_{\mathrm{AGE}}(\boldsymbol{\alpha}, i)$ are nonlinear, and coordinate-system dependent; involve an auxiliary variable $\boldsymbol{\nu}^{(i)}$.

The Trouble in Analyzing SAGE

Caltech

Implementing $\boldsymbol{c} \in C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$ comes down to

1. $\boldsymbol{c} = \sum_{i=1}^{m} \boldsymbol{c}^{(i)}$,
2. $\boldsymbol{c}^{(i)} \in C_{\mathrm{AGE}}(\boldsymbol{\alpha}, i)$ for all i in $[m]$.

Constraints for $\boldsymbol{c}^{(i)} \in C_{\mathrm{AGE}}(\boldsymbol{\alpha}, i)$ are nonlinear, and coordinate-system dependent; involve an auxiliary variable $\boldsymbol{\nu}^{(i)}$.

To devise proofs, we find it useful to ...

1. know exactly which $\boldsymbol{c}^{(i)}$ need actually be used.

The Trouble in Analyzing SAGE

Caltech

Implementing $\boldsymbol{c} \in C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$ comes down to

1. $\boldsymbol{c} = \sum_{i=1}^{m} \boldsymbol{c}^{(i)}$,
2. $\boldsymbol{c}^{(i)} \in C_{\mathrm{AGE}}(\boldsymbol{\alpha}, i)$ for all i in $[m]$.

Constraints for $\boldsymbol{c}^{(i)} \in C_{\mathrm{AGE}}(\boldsymbol{\alpha}, i)$ are nonlinear, and coordinate-system dependent; involve an auxiliary variable $\boldsymbol{\nu}^{(i)}$.

To devise proofs, we find it useful to ...

1. know exactly which $\boldsymbol{c}^{(i)}$ need actually be used.
2. know the sparsity pattern on any $\boldsymbol{c}^{(i)}$ that are used.

The Trouble in Analyzing SAGE

Caltech

Implementing $\boldsymbol{c} \in C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$ comes down to

1. $\boldsymbol{c} = \sum_{i=1}^{m} \boldsymbol{c}^{(i)}$,
2. $\boldsymbol{c}^{(i)} \in C_{\mathrm{AGE}}(\boldsymbol{\alpha}, i)$ for all i in $[m]$.

Constraints for $\boldsymbol{c}^{(i)} \in C_{\mathrm{AGE}}(\boldsymbol{\alpha}, i)$ are nonlinear, and coordinate-system dependent; involve an auxiliary variable $\boldsymbol{\nu}^{(i)}$.

To devise proofs, we find it useful to ...

1. know exactly which $\boldsymbol{c}^{(i)}$ need actually be used.
2. know the sparsity pattern on any $\boldsymbol{c}^{(i)}$ that are used.
3. exploit structure in $\boldsymbol{\alpha}, \boldsymbol{c}$ to inform the choice of $\boldsymbol{\nu}^{(i)}$.

Standard-Form SAGE Decompositions

Caltech

Theorem (2)

Let $f = \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})$ *have* $k > 0$ *negative entries in* $\boldsymbol{c}$. *Then*

Standard-Form SAGE Decompositions

Caltech

Theorem (2)

Let $f = \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})$ have $k > 0$ negative entries in $\boldsymbol{c}$. Then

1. *f is SAGE iff if it can be decomposed into k AGE functions,*

Standard-Form SAGE Decompositions

Caltech

Theorem (2)

Let $f = \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})$ have $k > 0$ negative entries in $\boldsymbol{c}$. Then

1. *f is SAGE iff if it can be decomposed into k AGE functions, and*
2. *without loss of generality, $c_i < 0 \Rightarrow c_i^{(j)} = 0$ when $j \neq i$.*

Standard-Form SAGE Decompositions

Caltech

Theorem (2)

Let $f = \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})$ have $k > 0$ negative entries in $\boldsymbol{c}$. Then

1. *f is SAGE iff if it can be decomposed into k AGE functions, and*
2. *without loss of generality, $c_i < 0 \Rightarrow c_i^{(j)} = 0$ when $j \neq i$.*

A key step in characterizing the extreme rays of $C_{\mathrm{SAGE}}^{\mathrm{POLY}}(\boldsymbol{\alpha})$.

Standard-Form SAGE Decompositions

Caltech

Theorem (2)

Let $f = \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})$ have $k > 0$ negative entries in $\boldsymbol{c}$. Then

1. *f is SAGE iff if it can be decomposed into k AGE functions, and*
2. *without loss of generality, $c_i < 0 \Rightarrow c_i^{(j)} = 0$ when $j \neq i$.*

A key step in characterizing the extreme rays of $C_{\mathrm{SAGE}}^{\mathrm{POLY}}(\boldsymbol{\alpha})$.

Integrated into our "`sage.py`" Python module

reduces barrier complexity from $O(m^2)$ to $O(km)$ (!)

Newton Polytopes and Exponent Vectors

Caltech

A central object in analysis of signomials is the *Newton polytope*.

Definition. Given $f = \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})$, the Newton polytope of f is

$$\mathcal{P}(\boldsymbol{\alpha}) \doteq \mathrm{conv}\{\boldsymbol{\alpha}_i\}_{i=1}^m.$$

Our first order of business should be to find some regularity conditions for Newton polytopes.

Next: a definition leading to a regularity condition.

"Partitioning" Exponent Vectors

Caltech

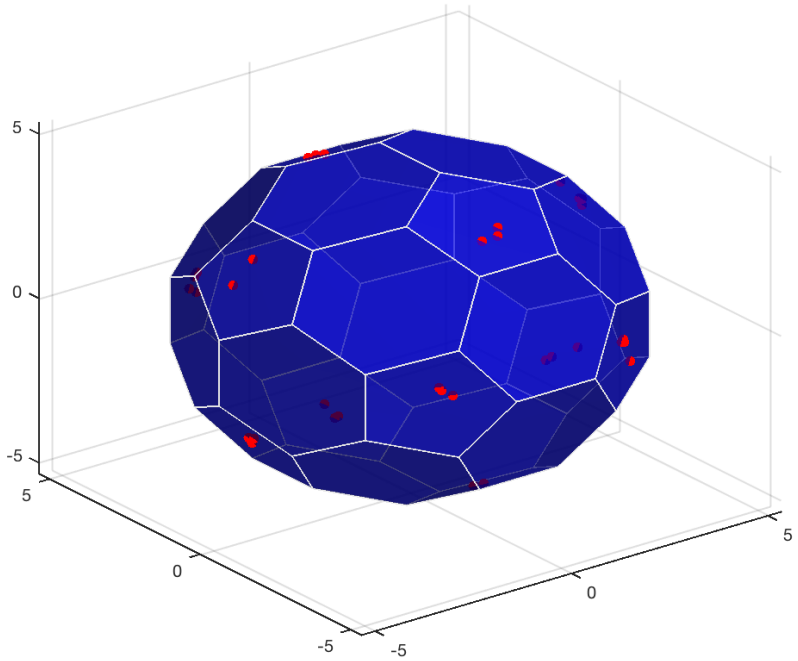
We say that $\boldsymbol{\alpha}$ can be **partitioned into** k **faces** if we can permute its columns so that $\boldsymbol{\alpha} = [\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(k)}]$ where $\{\mathcal{P}(\boldsymbol{\alpha}^{(i)})\}_{i=1}^{k}$ are mutually disjoint faces of $\mathcal{P}(\boldsymbol{\alpha})$.

"Partitioning" Exponent Vectors

Caltech

We say that $\boldsymbol{\alpha}$ can be **partitioned into** k **faces** if we can permute its columns so that $\boldsymbol{\alpha} = [\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(k)}]$ where $\{\mathcal{P}(\boldsymbol{\alpha}^{(i)})\}_{i=1}^{k}$ are mutually disjoint faces of $\mathcal{P}(\boldsymbol{\alpha})$.

A timely example...

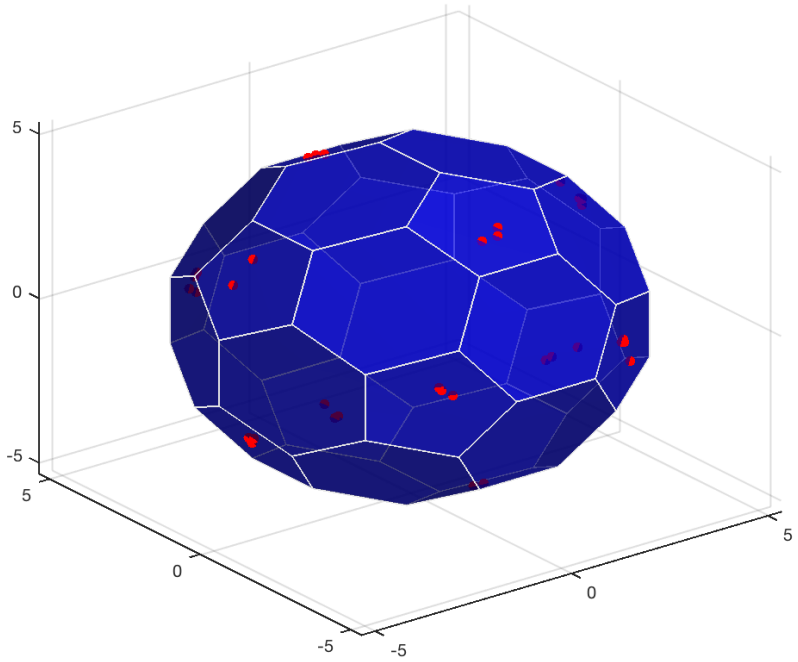


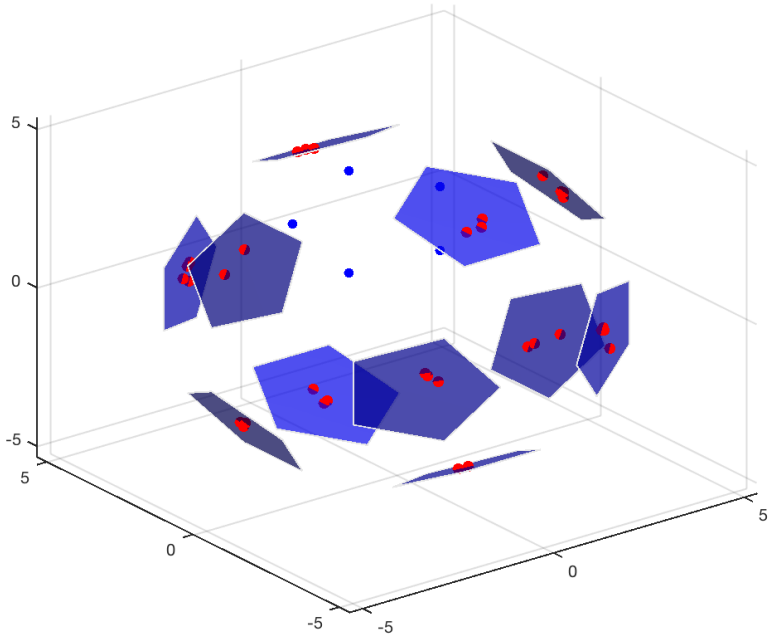
"Partitioning" Exponent Vectors

Caltech

We say that $\boldsymbol{\alpha}$ can be **partitioned into** k **faces** if we can permute its columns so that $\boldsymbol{\alpha} = [\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(k)}]$ where $\{\mathcal{P}(\boldsymbol{\alpha}^{(i)})\}_{i=1}^{k}$ are mutually disjoint faces of $\mathcal{P}(\boldsymbol{\alpha})$.

A timely example...





The Partitioning Theorem

Caltech

Theorem (3)

If $\{\boldsymbol{\alpha}^{(i)}\}_{i=1}^{k}$ *are matrices partitioning* $\boldsymbol{\alpha} = [\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(k)}]$*, then*

$$C_{\mathrm{NNG}}(\boldsymbol{\alpha}) = \oplus_{i=1}^{k} C_{\mathrm{NNG}}(\boldsymbol{\alpha}^{(i)})$$

The Partitioning Theorem

Caltech

Theorem (3)

If $\{\boldsymbol{\alpha}^{(i)}\}_{i=1}^{k}$ *are matrices partitioning* $\boldsymbol{\alpha} = [\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(k)}]$*, then*

$$C_{\mathrm{NNG}}(\boldsymbol{\alpha}) = \oplus_{i=1}^{k} C_{\mathrm{NNG}}(\boldsymbol{\alpha}^{(i)})$$

–and the same is true of $C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$*.*

The Partitioning Theorem

Caltech

Theorem (3)

If $\{\boldsymbol{\alpha}^{(i)}\}_{i=1}^{k}$ *are matrices partitioning* $\boldsymbol{\alpha} = [\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(k)}]$*, then*

$$C_{\mathrm{NNG}}(\boldsymbol{\alpha}) = \oplus_{i=1}^{k} C_{\mathrm{NNG}}(\boldsymbol{\alpha}^{(i)})$$

–and the same is true of $C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$*.*

Sanity checks :

Any $\boldsymbol{\alpha}$ admits a trivial partition with $k = 1$.

If $\boldsymbol{\alpha} = \operatorname{ext} \boldsymbol{\alpha}$, then $C_{\mathrm{NNG}}(\boldsymbol{\alpha}) = C_{\mathrm{SAGE}}(\boldsymbol{\alpha}) = \mathbb{R}_{+}^{m}$.

The Partitioning Theorem

Caltech

Theorem (3)

If $\{\boldsymbol{\alpha}^{(i)}\}_{i=1}^{k}$ *are matrices partitioning* $\boldsymbol{\alpha} = [\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(k)}]$*, then*

$$C_{\mathrm{NNG}}(\boldsymbol{\alpha}) = \oplus_{i=1}^{k} C_{\mathrm{NNG}}(\boldsymbol{\alpha}^{(i)})$$

–and the same is true of $C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$*.*

Sanity checks :

Any $\boldsymbol{\alpha}$ admits a trivial partition with $k = 1$.

If $\boldsymbol{\alpha} = \operatorname{ext} \boldsymbol{\alpha}$, then $C_{\mathrm{NNG}}(\boldsymbol{\alpha}) = C_{\mathrm{SAGE}}(\boldsymbol{\alpha}) = \mathbb{R}^m_+$.

A natural regularity condition: $\boldsymbol{\alpha}$'s *only* partition is trivial.

Sufficient to have at least one $\boldsymbol{\alpha}_i$ in $\operatorname{relint} \mathcal{P}(\boldsymbol{\alpha})$.

Exactness (unconstrained)

Caltech

A result mentioned in Venkat's talk on Tuesday:

Theorem (4)

If $\mathcal{P}(\boldsymbol{\alpha})$ is simplicial, and $c_i \leq 0$ for all nonextremal $\boldsymbol{\alpha}_i$, then $\boldsymbol{c} \in C_{\mathrm{NNG}}(\boldsymbol{\alpha})$ if and only if $\boldsymbol{c} \in C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$.

Exactness (unconstrained)

Caltech

A result mentioned in Venkat's talk on Tuesday:

Theorem (4)

If $\mathcal{P}(\boldsymbol{\alpha})$ is simplicial, and $c_i \leq 0$ for all nonextremal $\boldsymbol{\alpha}_i$, then $\boldsymbol{c} \in C_{\mathrm{NNG}}(\boldsymbol{\alpha})$ if and only if $\boldsymbol{c} \in C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$.

In optimization form, Theorem 4 writes as ...

Corollary (5)

Assume $\mathcal{P}(\boldsymbol{\alpha})$ is simplicial, and nonextremal $\boldsymbol{\alpha}_i \neq \mathbf{0}$ have $c_i \leq 0$.

Exactness (unconstrained)

Caltech

A result mentioned in Venkat's talk on Tuesday:

Theorem (4)

If $\mathcal{P}(\boldsymbol{\alpha})$ is simplicial, and $c_i \leq 0$ for all nonextremal $\boldsymbol{\alpha}_i$, then $\boldsymbol{c} \in C_{\mathrm{NNG}}(\boldsymbol{\alpha})$ if and only if $\boldsymbol{c} \in C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$.

In optimization form, Theorem 4 writes as ...

Corollary (5)

Assume $\mathcal{P}(\boldsymbol{\alpha})$ is simplicial, and nonextremal $\boldsymbol{\alpha}_i \neq \mathbf{0}$ have $c_i \leq 0$.

- *If $\mathbf{0}$ is extremal, then $f_{\mathrm{SAGE}} = f^\star$.*

Exactness (unconstrained)

Caltech

A result mentioned in Venkat's talk on Tuesday:

Theorem (4)

If $\mathcal{P}(\boldsymbol{\alpha})$ is simplicial, and $c_i \leq 0$ for all nonextremal $\boldsymbol{\alpha}_i$, then $\boldsymbol{c} \in C_{\mathrm{NNG}}(\boldsymbol{\alpha})$ if and only if $\boldsymbol{c} \in C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$.

In optimization form, Theorem 4 writes as ...

Corollary (5)

Assume $\mathcal{P}(\boldsymbol{\alpha})$ is simplicial, and nonextremal $\boldsymbol{\alpha}_i \neq \mathbf{0}$ have $c_i \leq 0$.

- *If $\mathbf{0}$ is extremal, then $f_{\mathrm{SAGE}} = f^\star$.*
- *If $\mathbf{0}$ is nonextremal, then $f_{\mathrm{SAGE}} = f^\star$, or $f_{\mathrm{SAGE}} \neq f^\star < c_1$.*

Exactness (unconstrained)

Caltech

The proof of Theorem 4 is actually optimization-based, and it gives rise to a means of solution recovery.

Exactness (unconstrained)

Caltech

The proof of Theorem 4 is actually optimization-based, and it gives rise to a means of solution recovery.

1. Suppose the dual optimal value

$$f_{\mathrm{SAGE}} = \inf\{\boldsymbol{c}^{\mathsf{T}}\boldsymbol{v} : \boldsymbol{v} \text{ in } C_{\mathrm{SAGE}}(\boldsymbol{\alpha})^{\dagger} \text{ has } \boldsymbol{e}_1^{\mathsf{T}}\boldsymbol{v} = 1\}$$

   is attained by $\boldsymbol{v}^{\star}$.

Exactness (unconstrained)

Caltech

The proof of Theorem 4 is actually optimization-based, and it gives rise to a means of solution recovery.

1. Suppose the dual optimal value

$$f_{\mathrm{SAGE}} = \inf\{\boldsymbol{c}^{\mathsf{T}}\boldsymbol{v} : \boldsymbol{v} \text{ in } C_{\mathrm{SAGE}}(\boldsymbol{\alpha})^{\dagger} \text{ has } \boldsymbol{e}_1^{\mathsf{T}}\boldsymbol{v} = 1\}$$

   is attained by $\boldsymbol{v}^{\star}$.

2. Define $\hat{\boldsymbol{v}} = \boldsymbol{v}^{\star}[\boldsymbol{c} > 0]$, and $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}[:,\ \boldsymbol{c} > 0]$.

Exactness (unconstrained)

Caltech

The proof of Theorem 4 is actually optimization-based, and it gives rise to a means of solution recovery.

1. Suppose the dual optimal value

$$f_{\mathrm{SAGE}} = \inf\{\boldsymbol{c}^{\mathsf{T}}\boldsymbol{v} : \boldsymbol{v} \text{ in } C_{\mathrm{SAGE}}(\boldsymbol{\alpha})^{\dagger} \text{ has } \boldsymbol{e}_1^{\mathsf{T}}\boldsymbol{v} = 1\}$$

   is attained by $\boldsymbol{v}^{\star}$.

2. Define $\hat{\boldsymbol{v}} = \boldsymbol{v}^{\star}[\boldsymbol{c} > 0]$, and $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}[:,\ \boldsymbol{c} > 0]$.

3. If $\boldsymbol{x}$ solves $\hat{\boldsymbol{\alpha}}^{\mathsf{T}}\boldsymbol{x} = \log \hat{\boldsymbol{v}}$, then $f(\boldsymbol{x}) = \boldsymbol{c}^{\mathsf{T}}\boldsymbol{v}^{\star} = f^{\star}$.

Exactness (unconstrained)

Caltech

The proof of Theorem 4 is actually optimization-based, and it gives rise to a means of solution recovery.

1. Suppose the dual optimal value

$$f_{\mathrm{SAGE}} = \inf\{\boldsymbol{c}^{\mathsf{T}}\boldsymbol{v} : \boldsymbol{v} \text{ in } C_{\mathrm{SAGE}}(\boldsymbol{\alpha})^{\dagger} \text{ has } \boldsymbol{e}_1^{\mathsf{T}}\boldsymbol{v} = 1\}$$

   is attained by $\boldsymbol{v}^{\star}$.

2. Define $\hat{\boldsymbol{v}} = \boldsymbol{v}^{\star}[\boldsymbol{c} > 0]$, and $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}[:,\ \boldsymbol{c} > 0]$.

3. If $\boldsymbol{x}$ solves $\hat{\boldsymbol{\alpha}}^{\mathsf{T}}\boldsymbol{x} = \log \hat{\boldsymbol{v}}$, then $f(\boldsymbol{x}) = \boldsymbol{c}^{\mathsf{T}}\boldsymbol{v}^{\star} = f^{\star}$.

I.e.- solution recovery only depends on the $\boldsymbol{\alpha}_i$ with $c_i > 0$.

Exactness (constrained)

Caltech

Theorem 4 also extends to constrained optimization.

Exactness (constrained)

Caltech

Theorem 4 also extends to constrained optimization.

Corollary (6)

Suppose that $\mathcal{P}(\boldsymbol{\alpha})$ is simplicial with vertex $\boldsymbol{\alpha}_1 = \mathbf{0}$, and that when $\boldsymbol{\alpha}_i$ is nonextremal we have

Exactness (constrained)

Caltech

Theorem 4 also extends to constrained optimization.

Corollary (6)

Suppose that $\mathcal{P}(\boldsymbol{\alpha})$ is simplicial with vertex $\boldsymbol{\alpha}_1 = \mathbf{0}$, and that when $\boldsymbol{\alpha}_i$ is nonextremal we have

1. *the objective $\boldsymbol{c}^{\mathsf{T}}\boldsymbol{v}$ is decreasing in v_i,*

Exactness (constrained)

Caltech

Theorem 4 also extends to constrained optimization.

Corollary (6)

Suppose that $\mathcal{P}(\boldsymbol{\alpha})$ is simplicial with vertex $\boldsymbol{\alpha}_1 = \mathbf{0}$, and that when $\boldsymbol{\alpha}_i$ is nonextremal we have

1. *the objective $\boldsymbol{c}^{\mathsf{T}}\boldsymbol{v}$ is decreasing in v_i, and*
2. *the constraint functionals $\boldsymbol{v} \mapsto \boldsymbol{g}_i^{\mathsf{T}}\boldsymbol{v}$ are increasing in v_i.*

Exactness (constrained)

Caltech

Theorem 4 also extends to constrained optimization.

Corollary (6)

Suppose that $\mathcal{P}(\boldsymbol{\alpha})$ is simplicial with vertex $\boldsymbol{\alpha}_1 = \mathbf{0}$, and that when $\boldsymbol{\alpha}_i$ is nonextremal we have

1. *the objective $\boldsymbol{c}^{\mathsf{T}}\boldsymbol{v}$ is decreasing in v_i, and*
2. *the constraint functionals $\boldsymbol{v} \mapsto \boldsymbol{g}_i^{\mathsf{T}}\boldsymbol{v}$ are increasing in v_i.*

Then $(f, g)_{\mathrm{SAGE}} = (f, g)^{\star}$.

Bounded Error (unconstrained)

Caltech

Theorem (7)

Suppose there exists an $\epsilon > 0$ *so that* $(1+\epsilon)\boldsymbol{\alpha}_j$ *belongs to* $\mathcal{P}(\boldsymbol{\alpha})$ *for all nonextremal* $\boldsymbol{\alpha}_j$.

Bounded Error (unconstrained)

Caltech

Theorem (7)

Suppose there exists an $\epsilon > 0$ *so that* $(1+\epsilon)\boldsymbol{\alpha}_j$ *belongs to* $\mathcal{P}(\boldsymbol{\alpha})$ *for all nonextremal* $\boldsymbol{\alpha}_j$.

Then $f = \mathrm{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})$ *is bounded below iff* f_{SAGE} *is finite*

Bounded Error (unconstrained)

Caltech

Theorem (7)

Suppose there exists an $\epsilon > 0$ *so that* $(1+\epsilon)\boldsymbol{\alpha}_j$ *belongs to* $\mathcal{P}(\boldsymbol{\alpha})$ *for all nonextremal* $\boldsymbol{\alpha}_j$*.*

Then $f = \operatorname{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})$ *is bounded below iff* f_{SAGE} *is finite .*

A noteworthy case within the theorem's scope:

when all exponent are internal or extremal in $\mathcal{P}(\boldsymbol{\alpha})$.

Bounded Error (unconstrained)

Caltech

Theorem (7)

Suppose there exists an $\epsilon > 0$ *so that* $(1+\epsilon)\boldsymbol{\alpha}_j$ *belongs to* $\mathcal{P}(\boldsymbol{\alpha})$ *for all nonextremal* $\boldsymbol{\alpha}_j$*.*

Then $f = \operatorname{Sig}(\boldsymbol{\alpha}, \boldsymbol{c})$ *is bounded below iff* f_{SAGE} *is finite .*

A noteworthy case within the theorem's scope:

when all exponent are internal or extremal in $\mathcal{P}(\boldsymbol{\alpha})$.

A noteworthy case *outside* of the theorem's scope:

exponent vectors belonging to a homogeneous polynomial.

Additional Results

Caltech

Results presented here form a subset of the work we've done on SAGE in the last year.

In our upcomming paper, you can find...

Additional Results

Caltech

Results presented here form a subset of the work we've done on SAGE in the last year.

In our upcomming paper, you can find...

- Globally nonnegative "SAGE polynomials."

Additional Results

Caltech

Results presented here form a subset of the work we've done on SAGE in the last year.

In our upcomming paper, you can find...

- Globally nonnegative "SAGE polynomials."
- Extreme rays of $C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$ and $C_{\mathrm{SAGE}}^{\mathrm{POLY}}(\boldsymbol{\alpha})$.

Additional Results

Caltech

Results presented here form a subset of the work we've done on SAGE in the last year.

In our upcomming paper, you can find...

- Globally nonnegative "SAGE polynomials."
- Extreme rays of $C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$ and $C_{\mathrm{SAGE}}^{\mathrm{POLY}}(\boldsymbol{\alpha})$.
- A dual characterization of SAGE-vs-NNG.

Additional Results

Caltech

Results presented here form a subset of the work we've done on SAGE in the last year.

In our upcomming paper, you can find...

- Globally nonnegative "SAGE polynomials."
- Extreme rays of $C_{\mathrm{SAGE}}(\boldsymbol{\alpha})$ and $C_{\mathrm{SAGE}}^{\mathrm{POLY}}(\boldsymbol{\alpha})$.
- A dual characterization of SAGE-vs-NNG.
- A conjecture (with evidence) of necessary and sufficient conditions for SAGE-vs-NNG in terms of Newton polytopes.

Opportunities

Caltech

Examples of concrete theoretical questions:

- Can we *recognize* $f_{\mathrm{SAGE}} = f^\star$ when it happens?

Opportunities

Caltech

Examples of concrete theoretical questions:

- Can we *recognize* $f_{\mathrm{SAGE}} = f^\star$ when it happens?
- Is "partitioning" $\boldsymbol{\alpha}$ equivalent to "factoring" $C_{\mathrm{NNG}}(\boldsymbol{\alpha})$?

Opportunities

Caltech

Examples of concrete theoretical questions:

- Can we *recognize* $f_{\mathrm{SAGE}} = f^\star$ when it happens?
- Is "partitioning" $\boldsymbol{\alpha}$ equivalent to "factoring" $C_{\mathrm{NNG}}(\boldsymbol{\alpha})$?
- SAGE leads to a hierarchy. How might we recover solutions from dual formulations at other levels of the heirarchy?

Opportunities

Caltech

Examples of concrete theoretical questions:

- Can we *recognize* $f_{\mathrm{SAGE}} = f^\star$ when it happens?

  I'm not sure!

- Is "partitioning" $\boldsymbol{\alpha}$ equivalent to "factoring" $C_{\mathrm{NNG}}(\boldsymbol{\alpha})$?

- SAGE leads to a hierarchy. How might we recover solutions from dual formulations at other levels of the heirarchy?

Opportunities

Caltech

Examples of concrete theoretical questions:

- Can we *recognize* $f_{\mathrm{SAGE}} = f^\star$ when it happens?

  I'm not sure!
- Is "partitioning" $\boldsymbol{\alpha}$ equivalent to "factoring" $C_{\mathrm{NNG}}(\boldsymbol{\alpha})$?

  Probably (?)
- SAGE leads to a hierarchy. How might we recover solutions from dual formulations at other levels of the heirarchy?

Opportunities

Caltech

Examples of concrete theoretical questions:

- Can we *recognize* $f_{\mathrm{SAGE}} = f^\star$ when it happens?

  I'm not sure!

- Is "partitioning" $\boldsymbol{\alpha}$ equivalent to "factoring" $C_{\mathrm{NNG}}(\boldsymbol{\alpha})$?

  Probably (?)

- SAGE leads to a hierarchy. How might we recover solutions from dual formulations at other levels of the heirarchy?

  Have a hueristic, not sure of theoretical properties.

Thank you!

SAGE as a Hierarchy

Caltech

For unconstrained problems, have

$$f_{\mathrm{SAGE}}^{(\ell)} \doteq \sup\{\gamma : \mathrm{Sig}(\boldsymbol{\alpha}, \mathbf{1})^{\ell}(f - \gamma) \text{ is SAGE } \}.$$

Can show strong duality holds for the above and its conic dual!

For constrained problems, have sequence $(f, g)_{\mathrm{p}}^{(a,b)}$ where

- Constraints $\{g_i\}_{i \in [k]}$ are replaced by $\{\prod_{i \in s} g_i : s \in [k]^b\}$.
- Dual variables are SAGE functions (# terms exponential in a).

Of course, can take duals of $(f, g)_{\mathrm{p}}^{(a,b)}$ to obtain $(f, g)_{\mathrm{d}}^{(a,b)}$.

SAGE as a Hierarchy

Caltech

BoundedFunctionsNeedNotHaveBoundedSAGERelaxations.py

UniqueConvexCoversInsufficient.py