Low-Rank Univariate Sum-of-Squares Has No Spurious Local Minima

Presented by Benoît Legat
Based on joint work with Chenyang Yuan and Pablo Parrilo
First-order methods

- Amenability to **parallelization**
- Affordable **per-iteration** computational cost
- Low **storage** requirements

<table>
<thead>
<tr>
<th># nodes</th>
<th>PDLP</th>
<th>SCS</th>
<th>Gurobi Barrier</th>
<th>Gurobi Primal Simp.</th>
<th>Gurobi Dual Simp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^4$</td>
<td>7.4 sec.</td>
<td>1.3 sec.</td>
<td>36 sec.</td>
<td>37 sec.</td>
<td>114 sec.</td>
</tr>
<tr>
<td>$10^5$</td>
<td>35 sec.</td>
<td>38 sec.</td>
<td>7.8 hr.</td>
<td>9.3 hr.</td>
<td>&gt;24 hr.</td>
</tr>
<tr>
<td>$10^6$</td>
<td>11 min.</td>
<td>25 min.</td>
<td>OOM</td>
<td>&gt;24 hr.</td>
<td>-</td>
</tr>
<tr>
<td>$10^7$</td>
<td>5.4 hr.</td>
<td>3.8 hr.</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Deep Learning uses gradient-based solvers on large scale problems

Very successful on various classification and inference tasks

Solved with highly parallelized first-order methods
Nonconvex factorization formulations

- Basin of attraction
  - Initialization
  - Iterative refinement
- Benign Global Landscape
  
  Require statistical/genericity conditions such as Restricted isometry property (RIP)

Matrix sensing, matrix completion, phase retrieval, blind deconvolution, ...

\[
\begin{align*}
\text{minimize} \quad & f(L, R) = \frac{1}{4m} \sum_{i=1}^{m} (\langle A_i, LR^\top \rangle - y_i)^2 \\
\end{align*}
\]

Semidefinite programming (SDP) is a powerful and expressive convex optimization method.

A $n \times n$ positive semidefinite variable $X \succeq 0$ plus $m$ linear constraints.

Applications: Optimal control, Lyapunov analysis, convex relaxations of combinatorial optimization, rank minimization and nuclear norm, ...

Typically solved with expensive interior point methods.

- $O((mn + m^2)n^2)$ operations per iteration
- $O(\sqrt{n \log(\varepsilon)})$ iterations
- $O(m^2 + n^2)$ memory

First-order solver for nonconvex factorization formulation?
Introduction

Burer-Monteiro methods factor PSD constraint \( X = UU^T \), then perform local optimization on resulting non-convex unconstrained problem

\[
\langle A_i, X \rangle = b_i \quad \forall i \\
X \succeq 0
\]

Feasible

\[
\min_U \sum_i (\langle A_i, UU^T \rangle - b_i)^2
\]

Optimum = 0

May get stuck in local optimum (explicit counterexamples where second-order critical point ≠ global minimum)

When is non-convexity benign?
Related work

For general SDP feasibility with $m$ linear constraints, with the factorization $X = UU^T$, where $U$ is a $n \times r$ matrix.

Second-order critical point $\Rightarrow$ Global minimum (non-convexity benign) when:

- $r > n$ [Burer and Monteiro]
- $r = \Omega(\sqrt{m})$, but with smoothed analysis [Cifuentes and Moitra], generic constraints [Bhojanapalli, Boumal, Jain, Netrapalli], or determinant regularization [Burer and Monteiro], (necessary because of counterexamples)

Can we do better if the SDP has special structure?

Sum of Squares Optimization

Given $p(x)$, can we write it as a sum of squares?

Certifies that $p(x) \geq 0$, and can be formulated as SDP

Focus on univariate trigonometric polynomials in this talk (methods can be generalized to multivariate case)

$$p(x) = a_0 + \sum_{k=1}^{d} a_k \cos(kx) + a_{-k} \sin(kx), \quad x \in [0, \pi]$$

Applications in signal processing, filter design and control

$$H(z) = C(zI - A)^{-1}B$$

Univariate to trigonometric basis

\[ x^2 - 2x + 1 \]
\[ x^2 - 2xy + y^2 \]
\[ \cos(\alpha)^2 - 2 \cos(\alpha) \sin(\alpha) + \sin(\alpha)^2 \]
\[ 1 - \sin(2\alpha) \]

*Linear transformation on coefficients:*

*Chebyshev basis*

Contributions

Find sum of squares decomposition of \( p(x) \) by solving

\[
\min_u f(u) = \left\| \sum_{i=1}^{r} u_i(x)^2 - p(x) \right\|
\]

For any norm on polynomials, if \( f(u) = 0 \), sum of squares decomposition agrees with \( p(x) \).

Theorem: when \( r \geq 2 \) (vs \( r = \Omega(\sqrt{m}) \)) first-order methods find sum of squares decomposition for univariate polynomials (non-convexity benign)

If we choose right norm, \( \nabla f(u) \) can be computed in \( O(d \log d) \) time using fast fourier transforms (FFTs)

Sampled basis

Which inner product $\langle p(x), q(x) \rangle$ on polynomials to choose?

Given $p(x)$, $q(x)$ degree $d$, choose $d+1$ points $x_k$

$$\langle p(x), q(x) \rangle = \sum_{k=1}^{d+1} p(x_k)q(x_k), \quad \|p(x)\|^2 = \sum_{k=1}^{d+1} p(x_k)^2$$

Valid inner product: when $x_k$ are distinct points, if $\|p(x)\|^2 = 0$ then $p(x) = 0$.

Sum of squares using a sampled/interpolation basis studied by [Löfberg and Parrilo] and [Cifuentes and Parrilo]

How should we choose $x_k$?

Numerical Implementation

Compute sum of squares decomposition of degree 2d trigonometric polynomial

\[ p(x) = a_0 + \sum_{k=1}^{d} a_k \cos(kx) + a_{-k} \sin(kx) \]

Using basis vectors evaluated at 2d + 1 points

\[ \langle p, q \rangle = \sum_{k=1}^{2d+1} p(x_k)q(x_k), \quad x_k = \frac{2k\pi}{2d + 1} \]

\[ B_k = \begin{bmatrix} 1 & \cos(x_k) & \cdots & \cos\left(\frac{d}{2}x_k\right) & \sin(x_k) & \cdots & \sin\left(\frac{d}{2}x_k\right) \end{bmatrix}^T \]

Matrix-vector products in \( \nabla f(U) \) can be computed by FFT

\[ \nabla f(U) = U^T B \text{diag}(\|U^T B_k\|^2 - p(x_k)) B^T \]
Results

Sum of squares decomposition for random trigonometric polynomial

Convergence rate for LBFGS with random initialization:

<table>
<thead>
<tr>
<th>Degree</th>
<th>Time in seconds</th>
<th>Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,000</td>
<td>2 (1 – 2)</td>
<td>340 (306 – 384)</td>
</tr>
<tr>
<td>10,000</td>
<td>6 (5 – 6)</td>
<td>530 (497 – 592)</td>
</tr>
<tr>
<td>20,000</td>
<td>9 (8 – 10)</td>
<td>632 (587 – 695)</td>
</tr>
<tr>
<td>100,000</td>
<td>53 (46 – 59)</td>
<td>1126 (980 – 1248)</td>
</tr>
<tr>
<td>200,000</td>
<td>160 (139 – 174)</td>
<td>1375 (1212 – 1532)</td>
</tr>
<tr>
<td>1,000,000</td>
<td>1461 (1212 – 1532)</td>
<td>2303 (1934 – 2437)</td>
</tr>
</tbody>
</table>

Running times (stop at 10^{-7} relative error in U):

Use r = 4 with 4 cores.
Comparison with existing algorithms

*Sturm sequence*: Decide positivity of univariate polynomial of degree $d$ in $O(d^2)$

*Interior-point*: Univariate Sum-of-Squares program of degree $d$ in $O(d^4)$ per iteration and $O(\sqrt{d \log(\epsilon)})$ iterations.

*Infeasibility*: Dual certificate.

*Burer-Monteiro*: $O(d \log(d))$ per iteration for degree $d$.

*Infeasibility*: Projection to SOS cone.

Guarantee on number of iterations of Burer-Monteiro for univariate SOS?
Proof Sketch

Assume that $p(x)$ is a univariate polynomial and $r = 2$

$$f(u) = \left\| u_1(x)^2 + u_2(x)^2 - p(x) \right\|^2 = \left\| s(x) - p(x) \right\|^2$$

Given $u$ such that $\nabla f(u)(v) = 0$ and $\nabla^2 f(u)(v,v) \geq 0$ for all $v$, show that $f(u) = 0$

We have inner product $\langle p(x), q(x) \rangle$ on polynomials with associated norm $\| . \|:

$$\nabla f(u)(v) \sim \left\langle \sum_{j=1}^{r} u_j(x) v_j(x), s(x) - p(x) \right\rangle = 0$$

$$\nabla^2 f(u)(v,v) \sim \left\langle \sum_{j=1}^{r} v_j(x)^2, s(x) - p(x) \right\rangle + 2 \left\| \sum_{j=1}^{r} u_j(x) v_j(x) \right\|^2 \geq 0$$
Proof Sketch

Geometrically, we want to show that the only intersection between set with zero gradient and PSD Hessian is when \( f(u) = 0 \).

For fixed \( u \), these sets are convex!

Our proof can be interpreted as finding a certificate of this condition for every \( u \) and \( p \).
Proof Sketch

\[ \nabla f(u)(v) \sim \langle u_1(x)v_1(x) + u_2(x)v_2(x), s(x) - p(x) \rangle = 0 \]
\[ \nabla^2 f(u,v) \sim \langle v_1(x)^2 + v_2(x)^2, s(x) - p(x) \rangle + 2 \|u_1(x)v_1(x) + u_2(x)v_2(x)\|^2 \geq 0 \]

Suppose \( u_1, u_2 \) coprime (true generically)

Bézout's lemma + gradient condition \( \Rightarrow \) exist \( v_1, v_2 \) s.t.

\[ u_1(x)v_1(x) + u_2(x)v_2(x) = s(x) - p(x) \quad \Rightarrow \quad \| s(x) - p(x) \|^2 = 0 \]

Suppose \( u_1 = u_2 \), choose \( v_1 = v \) and \( v_2 = -v \) in Hessian condition so for all \( v \),

\[ \langle v(x)^2, s(x) - p(x) \rangle \geq 0 \quad \Rightarrow \quad \langle p(x), s(x) - p(x) \rangle \geq 0 \]

However, \( \langle s(x), s(x) - p(x) \rangle = 0 \) (gradient condition), so \( \| s(x) - p(x) \|^2 = 0 \)

Interpolate between these two cases with the Positivstellensatz
Numerical Implementation

TrigPolys.jl: a new package for fast manipulation of trigonometric polynomials

```julia
function Base.*(p1::TrigPoly, p2::TrigPoly)
    n = p1.n + p2.n
    interpolate(evaluate(pad_to(p1, n)) .* evaluate(pad_to(p2, n)))
end
```

evaluate, evaluateT and interpolate uses FFTW.jl, enables fast computation of $f(U)$:

```julia
f(u) = sum((evaluate(pad_to(u, p.n)).^2 - evaluate(p)).^2)
```

AutoGrad.jl enables automatic computation of $\nabla f(U)$

```julia
AutoGrad.@primitive evaluate(u::AbstractArray), dy, y evaluateT(dy)
fg = AutoGrad.grad(f)
```

Pass $f(U)$, $\nabla f(U)$ to NLopt.jl to minimize $f(U)$ with first-order optimization algorithms
Conclusion

When does it make sense to solve non-convex formulations of convex problems?

In our setting we can prove non-convexity does not hurt us

Also enables fast implementation in Julia