IPM-LSTM: A Learning-Based Interior Point Method for Solving Nonlinear Programs

Xi Gao[1] Jinxin Xiong[2,3] Akang Wang[2,3] Qihong Duan[1] Jiang Xue[1] Qingjiang Shi[2,4]

[1]School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China

[2]Shenzhen Research Institute of Big Data, China

[3]School of Data Science, The Chinese University of Hong Kong, Shenzhen, China

[4]School of Software Engineering, Tongji University, Shanghai, China

38th Conference on Neural Information Processing Systems, December 2024

Table of Contents

Table of Contents

The Interior Point Method

Problem

We focus on solving the following NLP (1):

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & h(x) = 0 \\ & x \geq 0 \end{aligned} \tag{1}$$

where the functions $f : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}^m$ are all assumed to be twice continuously differentiable.

The Interior Point Method

The Classic IPM

By introducing a decreasing sequence of parameters μ converging to zero, the perturbed *Karush-Kuhn-Tucker* (KKT) conditions can be represented as:

$$\begin{aligned} &\nabla f(x) + \lambda^\top \nabla h(x) - z = 0 \qquad && h(x) = 0 \\ &\mathrm{diag}(z)\mathrm{diag}(x)e = \mu e && x, z \geq 0 \end{aligned} \tag{2}$$

A one-step Newton's method is employed to solve such a system, aiming to solve systems of linear equations (3).

$$\underbrace{\begin{bmatrix} \nabla^2 f(x) + \lambda^\top \nabla^2 h(x) & \nabla h^\top(x) & -I \\ \nabla h(x) & & \\ \mathrm{diag}(z) & & \mathrm{diag}(x) \end{bmatrix}}_{J} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta z \end{bmatrix} = -F(x, \lambda, z) \tag{3}$$

The IPM commences with an initial solution (x^0, λ^0, z^0) such that $x^0, z^0 > 0$. At iteration k, the linear system (3) defined by the current iterate (x^k, λ^k, z^k) is solved.

The Interior Point Method

The Classic IPM

Algorithm 1 The classic IPM

Inputs: An initial solution (x^0, λ^0, z^0), $\sigma \in (0, 1)$, $k \leftarrow 0$
Outputs: The optimal solution (x^*, λ^*, z^*)

1: **while** not converged **do**
2: Update μ^k
3: Solve the system $J^k \left[(\Delta x^k)^\top, (\Delta \lambda^k)^\top, (\Delta z^k)^\top\right]^\top = -F^k$
4: Choose α^k via a line-search filter method
5: $(x^{k+1}, \lambda^{k+1}, z^{k+1}) \leftarrow (x^k, \lambda^k, z^k) + \alpha^k(\Delta x^k, \Delta \lambda^k, \Delta z^k)$
6: $k \leftarrow k + 1$
7: **end while**

- Solving linear systems is the main computational bottleneck.
- IPM is difficult to be warm-started.

Can we leverage L2O techniques to expedite IPMs for NLPs?

The Interior Point Method

Approximating Solutions to Linear Systems

To avoid high computational costs, the least squares problem (4) is employed to obtain the approximate solution of the IPM linear system.

$$\min_{y} \frac{1}{2} \left\| J^k y + F^k \right\|^2 \tag{4}$$

This perspective is similar to the inexact IPM[1].

Assumption 1

At iteration k, we could identify some y^k such that

$$\left\| J^k y^k + F^k \right\| \leq \eta \left[(z^k)^\top x^k \right] / n \tag{5}$$

$$\|y^k\| \leq (1 + \sigma + \eta) \|F_0(x^k, \lambda^k, z^k)\|. \tag{6}$$

where $\eta \in (0, 1)$ and $F_0(x^k, \lambda^k, z^k)$ denotes $F(x^k, \lambda^k, z^k)$ with $\mu = 0$.

[1] Stefania Bellavia. "Inexact interior-point method". In: *Journal of Optimization Theory and Applications* 96 (1998), pp. 109–121.

The Interior Point Method

Approximating Solutions to Linear Systems

To satisfy Assumption 1, the approximate solution y^k has to be **bounded** and **accurate enough**, regardless of whether J^k is invertible.

Proposition 1

If (x^k, λ^k, z^k) is generated such that Assumption 1 is satisfied, let (x^*, λ^*, z^*) denote a limit point of the sequence $\{(x^k, \lambda^k, z^k)\}$, then $\{(x^k, \lambda^k, z^k)\}$ converges to (x^*, λ^*, z^*) and $F_0(x^*, \lambda^*, z^*) = 0$.

Table of Contents

The IPM-LSTM Approach

Architecture

LSTM networks are considered suitable for solving the least squares problem due to the resemblance between LSTM recurrent calculations and iterative algorithms.

$$y_t := \mathrm{LSTM}_\theta\left(\left[y_{t-1}, (J^k)^\top (J^k y_{t-1} + F^k)\right]\right). \tag{7}$$

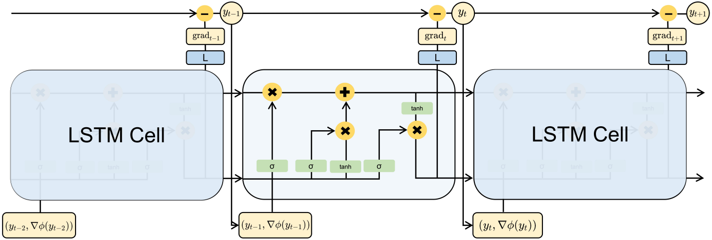


Figure 1: The LSTM architecture for solving the least quares problem.

The IPM-LSTM Approach

Model Training

Base on the least squares problem, we propose a new self-supervised loss function:

$$\min_{\theta} \frac{1}{|\mathcal{M}|} \sum_{M \in \mathcal{M}} \left(\frac{1}{K} \sum_{k=1}^{K} \frac{1}{T} \sum_{t=1}^{T} \frac{1}{2} \left\| J^k y_t^k(\theta) + F^k \right\|^2 \right)_M ,$$

where the subscript M indicates that the corresponding term is associated with instance M. *Truncated backpropagation through time* is employed to mitigate memory issues.

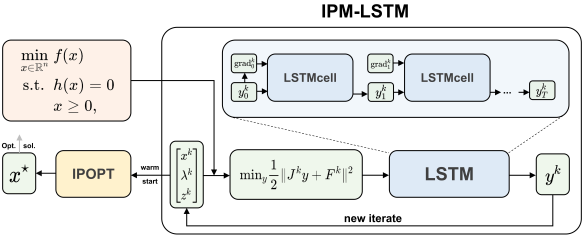


Figure 2: An illustration of the IPM-LSTM approach.

Table of Contents

Experiments

Experimental Settings

Datasets:

The dataset used in our work includes randomly generated benchmarks[2][3][4] as well as real-world instances from Globallib. These benchmarks encompass **convex QPs**, **convex QCQPs**, **nonconvex QPs**, and **simple non-convex programs**.

Baselines:

- Traditional optimizer: OSQP, IPOPT.
- L2O algorithms : NN, DC3, DeepLDE, PDL, LOOP-LC, H-Proj.

[2] Jieqiu Chen and Samuel Burer. "Globally solving nonconvex quadratic programming problems via completely positive programming". In: *Mathematical Programming Computation* 4.1 (2012), pp. 33–52.

[3] Priya L Donti, David Rolnick, and J Zico Kolter. "DC3: A learning method for optimization with hard constraints". In: (2021).

[4] Enming Liang, Minghua Chen, and Steven Low. "Low complexity homeomorphic projection to ensure neural-network Solution feasibility for optimization over (non-) convex set". In: (2023).

Experiments

QP

$$
\begin{aligned}
\min_{x \in \mathbb{R}^n} \quad & \frac{1}{2} x^\top Q_0 x + p_0^\top x \\
\text{s.t.} \quad & p_j^\top x \leq q_j && j = 1, \cdots, l \\
& p_j^\top x = q_j && j = l+1, \cdots, m \\
& x_i^L \leq x_i \leq x_i^U && i = 1, \cdots, n
\end{aligned} \tag{8}
$$

Table 1: Computational results on convex QPs.

Method	End-to-End						IPOPT (warm start)		Total Time (s) ↓	Gain (Ite./ Time) ↑
	Obj. ↓	Max ineq. ↓	Mean ineq. ↓	Max eq. ↓	Mean eq. ↓	Time (s) ↓	Ite. ↓	Time (s) ↓		
Convex QPs (RHS)										
OSQP	-29.176	0.000	0.000	0.000	0.000	0.009	-	-	-	-
IPOPT	-29.176	0.000	0.000	0.000	0.000	0.642	12.5	-	-	-
NN	-26.787	0.000	0.000	0.631	0.235	<0.001	10.5	0.560	0.560	16.0%/12.8%
DC3	-26.720	0.002	0.000	0.000	0.000	<0.001	10.2	0.535	0.535	18.4%/16.7%
DeepLDE	-3.697	0.000	0.000	0.000	0.000	<0.001	12.5	0.648	0.648	0.0%/-0.9%
PDL	-28.559	0.421	0.122	0.024	0.000	<0.001	9.7	0.514	0.514	22.4%/**19.9%**
LOOP-LC	-28.512	0.000	0.000	0.000	0.000	<0.001	10.8	0.565	0.565	13.6%/12.0%
H-Proj	-23.257	0.000	0.000	0.000	0.000	<0.001	11.2	0.605	0.605	10.4%/5.8%
IPM-LSTM	-29.050	0.000	0.000	0.002	0.001	0.175	7.2	0.370	0.545	**42.4%**/15.1%
Convex QPs (ALL)										
OSQP	-33.183	0.000	0.000	0.000	0.000	0.009	-	-	-	-
IPOPT	-33.183	0.000	0.000	0.000	0.000	0.671	12.9	-	-	-
IPM-LSTM	-32.600	0.000	0.000	0.003	0.001	0.195	8.3	0.426	0.621	**35.7%/7.5%**

Experiments

QP

Table 3: Computational results on non-convex QPs.

Instance	IPOPT			IPM-LSTM			IPOPT (warm-start)			Total Time (s)	Gain (Ite./ Time)
	Obj.	Ite.	Time (s)	Obj.	Max Vio.	Time (s)	Obj.	Ite.	Time (s)		
qp1	0.001	52.0	0.707	0.045	0.008	0.017	0.001	42.0	0.559	0.576	19.2%/18.5%
qp2	0.001	69.0	0.674	0.034	0.008	0.029	0.001	40.0	0.347	0.376	42.0%/ 44.2%
st_rv1	-58.430	215.0	0.955	-34.563	0.000	0.009	-58.867	168.0	0.626	0.635	21.9%/33.5%
st_rv2	-67.083	190.8	0.956	-30.955	0.000	0.011	-67.083	120.5	0.482	0.494	36.8%/38.1%
st_rv3	0.000	55.0	0.781	0.818	0.000	0.017	0.000	47.0	0.616	0.634	14.5%/18.8%
st_rv7	-132.019	449.0	2.445	-61.428	0.000	0.016	-131.756	162.0	0.705	0.721	63.9%/70.5%
st_rv9	-126.945	655.0	3.457	-58.415	0.000	0.026	-127.652	408.0	1.830	1.856	37.7%/46.3%
qp30_15_1_1	37.767	16.0	0.198	37.787	0.002	0.021	37.767	9.0	0.083	0.104	43.7%/47.5%

Max Vio. denotes the maximum constraint violation.

Experiments

Convex QCQP

$$\min_{x \in \mathbb{R}^n} \quad \frac{1}{2} x^\top Q_0 x + p_0^\top x$$

$$\text{s.t.} \quad x^\top Q_j x + p_j^\top x \leq q_j \qquad j = 1, \cdots, l$$

$$p_j^\top x = q_j \qquad j = l+1, \cdots, m$$

$$x_i^L \leq x_i \leq x_i^U \qquad i = 1, \cdots, n$$

Table 2: Computational results on convex QCQPs.

Method	End-to-End						IPOPT (warm start)		Total Time (s) ↓	Gain (Ite./ Time) ↑
	Obj. ↓	Max ineq. ↓	Mean ineq. ↓	Max eq. ↓	Mean eq. ↓	Time (s) ↓	Ite. ↓	Time (s) ↓		
Convex QCQPs (RHS)										
IPOPT	-39.162	0.000	0.000	0.000	0.000	1.098	12.5	-	-	-
NN	-2.105	0.000	0.000	0.552	0.169	<0.001	12.1	1.311	1.311	3.2%/-19.4%
DC3	-35.741	0.000	0.000	0.000	0.000	0.005	9.6	1.051	1.051	20.7%/4.8%
DeepLDE	-15.132	0.000	0.000	0.000	0.000	<0.001	11.5	1.222	1.222	8.0%/-11.3%
PDL	-39.089	0.005	0.000	0.015	0.005	<0.001	8.9	1.013	1.013	28.8%/**7.7%**
H-Proj	-36.062	0.000	0.000	0.000	0.000	<0.001	9.8	1.070	1.070	21.6%/2.6%
IPM-LSTM	-38.540	0.000	0.000	0.004	0.001	0.205	8.0	0.825	1.030	**36.0%**/6.2%
Convex QCQPs (ALL)										
IPOPT	-39.868	0.000	0.000	0.000	0.000	0.801	12.4	-	-	-
IPM-LSTM	-38.405	0.004	0.000	0.001	0.000	0.203	8.3	0.507	0.710	**33.1%/11.4%**

Experiments

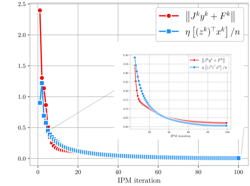
Simple Non-convex Program

Table 8: Computational results on non-convex programs

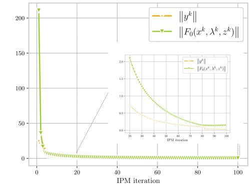
Method	End-to-End						IPOPT (warm start)		Total Time (s) ↓	Gain (Ite./ Time) ↑
	Obj. ↓	Max ineq. ↓	Mean ineq. ↓	Max eq. ↓	Mean eq. ↓	Time (s) ↓	Ite. ↓	Time (s) ↓		
Non-convex Programs (RHS): $n = 200, m_{\text{ineq}} = 100, m_{\text{eq}} = 100$										
IPOPT	-22.375	0.000	0.000	0.000	0.000	0.717	13.1	-	-	-
DC3	-20.671	0.000	0.000	0.000	0.000	<0.001	10.9	0.603	0.603	16.8%/15.9%
NN	-20.736	0.000	0.000	0.632	0.235	<0.001	11.0	0.607	0.607	16.0%/20.7%
DeepLDE	-20.074	0.000	0.000	0.000	0.000	<0.001	10.5	0.576	0.576	19.8%/19.7%
PDL	-21.859	0.589	0.167	0.026	0.000	<0.001	10.9	0.600	0.600	16.8%/16.3%
LOOP-LC	-21.932	0.000	0.000	0.000	0.000	<0.001	10.2	0.558	0.558	22.1%/**22.2%**
H-Proj	-19.097	0.000	0.000	0.006	0.000	<0.001	11.5	0.634	0.634	12.2%/11.6%
IPM-LSTM	-22.213	0.000	0.000	0.002	0.001	0.175	9.5	0.533	0.708	**27.5%**/1.3%
Non-convex Programs (ALL): $n = 200, m_{\text{ineq}} = 100, m_{\text{eq}} = 100$										
IPOPT	-25.1043	0.000	0.000	0.000	0.000	0.768	14.3	-	-	-
IPM-LSTM	-20.288	0.000	0.000	0.006	0.002	0.195	12.1	0.639	0.834	**15.4%**/-8.6%
Non-convex Programs (RHS): $n = 100, m_{\text{ineq}} = 50, m_{\text{eq}} = 50$										
IPOPT	-11.590	0.000	0.000	0.000	0.000	0.289	12.9	-	-	-
DC3	-10.660	0.000	0.000	0.000	0.000	<0.001	11.6	0.259	0.259	11.6%/10.4%
NN	-10.020	0.000	0.000	0.350	0.130	<0.001	11.4	0.253	0.253	11.6%/12.5%
DeepLDE	4.870	0.000	0.000	0.008	0.000	<0.001	13.1	0.294	0.294	-1.6%/-1.7%
PDL	-11.385	0.006	0.002	0.001	0.000	<0.001	9.6	0.207	0.207	25.6%/**28.4%**
LOOP-LC	-11.296	0.000	0.000	0.000	0.000	<0.001	10.1	0.217	0.217	21.7%/24.9%
H-Proj	-9.616	0.000	0.000	0.000	0.000	<0.001	11.3	0.252	0.252	12.4%/12.8%
IPM-LSTM	-11.421	0.000	0.000	0.002	0.001	0.044	8.9	0.181	0.225	**31.0%**/22.1%
Non-convex Programs (ALL): $n = 100, m_{\text{ineq}} = 50, m_{\text{eq}} = 50$										
IPOPT	-12.508	0.000	0.000	0.000	0.000	0.305	13.2	-	-	-
IPM-LSTM	-12.360	0.000	0.000	0.001	0.000	0.044	8.0	0.149	0.193	**39.4%/36.7%**

Experiments

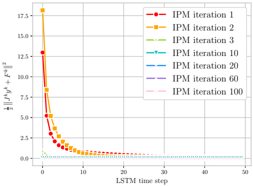
Performance Analysis of IPM-LSTM



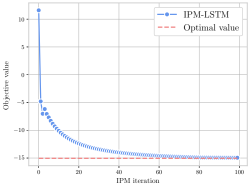
(a) Condition (5)



(b) Condition (6)



(c) Residual



(d) Objective value

Figure 3: The performance analysis of IPM-LSTM on a convex QP (RHS).

Table of Contents

Conclusions

Our work can be summarized as follows:

- Approximating Solutions to Linear Systems via LSTM.
- A new self-supervised loss function.
- A new learning-based method based on IPM that can simultaneously keep feasibility and optimality.
- Two-Stage Framework.
- Better performance in end-to-end solutions and warm-starting IPOPT.