Optimal Diagonal Preconditioner: Theory and Practice

ICIAM 2023

AUGUST 23

Yinyu Ye
Joint work with Qu, Gao, Hinder, and Zhou

Stanford University and CUHKSZ (Sabbatical Leave)

Today’s Talk

I. The Optimal Diagonal Preconditioner via Semidefinite Programming

II. Towards Practical Approximate Optimal Diagonal Preconditioner

Optimal Diagonal Preconditioner [QGHYZ 20]

Given matrix $M = X^\top X \succ 0$, iterative methods are applied to solve

$$Mx = b$$

- Convergence of iterative methods depends on the condition number $\kappa(M)$
- Good performance needs preconditioning and we solve $P^{-1/2}MP^{-1/2}x' = b$

  A good preconditioner reduces $\kappa(P^{-1/2}MP^{-1/2})$
- Diagonal $P = D$ is called diagonal preconditioner

  Most popular in practice: Jacobi, Ruiz, ADAM,…

More generally, we wish to find D (or E) such that $\kappa(D\boldsymbol{X}E)$ is minimized ?

Is it possible to find optimal D^* and E^* ?

SDP works!

Optimal Diagonal Preconditioner

$$\min_{D \text{ diagonal}, D \succeq 0} \kappa(DMD)$$

$$\min_{D \text{ diagonal}, D \succeq 0} \kappa(X^T D X)$$

$$\begin{aligned} \min_{D, \kappa} \quad & \kappa \\ \text{subject to} \quad & I \preceq DMD \preceq \kappa I \end{aligned}$$

$$\begin{aligned} \min_{\kappa, D \succeq 0} \quad & \kappa \\ \text{subject to} \quad & \kappa X^T D X \succeq I \\ & I \succeq X^T D X \end{aligned}$$

$$\begin{aligned} \min_{D, \kappa} \quad & \kappa \\ \text{subject to} \quad & D \preceq M \\ & \kappa D \succeq M \end{aligned}$$

$$\begin{aligned} \max_{\tau, D \succeq 0} \quad & \tau \\ \text{subject to} \quad & X^T D X \succeq \tau \\ & I \succeq X^T D X \end{aligned}$$

- **Finding the optimal diagonal preconditioner is an SDP**
- **Two SDP blocks and sparse coefficient matrices**
- **Trivial dual interior-feasible solution**
- **An ideal formulation for dual SDP methods** $D = \sum d_i e_i e_i^T$

What about two-sided ?

Extension: Optimal preconditioner with arbitrary sparsity pattern

SDP can be generalized to tackle preconditioners with arbitrary sparsity parttern

Given sparsity pattern S, find $P \in S$ such that $\kappa(P^{-1}M)$ minimized

Given sparsity pattern S, find $P^{-1} \in S$ such that $\kappa(P^{-1}M)$ minimized

$$
\begin{aligned}
\max_{\tau,\{p_{ij}\}} \quad & \tau \\
\text{subject to} \quad & M\tau \preceq \sum_{(i,j)\in\mathcal{S}} E_{ij}p_{ij} \\
& M \succeq \sum_{(i,j)\in\mathcal{S}} E_{ij}p_{ij}, \\
& \sum_{(i,j)\in\mathcal{S}} E_{ij}p_{ij} \succeq 0
\end{aligned}
$$

$$
\begin{aligned}
\max_{\tau,\{p_{ij}\}} \quad & \tau \\
\text{subject to} \quad & M^{-1}\tau \preceq \sum_{(i,j)\in\mathcal{S}} E_{ij}p_{ij} \\
& M^{-1} \succeq \sum_{(i,j)\in\mathcal{S}} E_{ij}p_{ij}, \\
& \sum_{(i,j)\in\mathcal{S}} E_{ij}p_{ij} \succeq 0
\end{aligned}
$$

- **Both problems are SDP-representable**
- **Providing benchmark for non-diagonal preconditioners**

e.g., tridiagonal, sparse approximate inverse...

Two-Sided Preconditioner

$$\min_{D_1 \succeq 0,\, D_2 \succeq 0} \quad \kappa(D_1 X D_2)$$

- Common in practice and popular heuristics exist

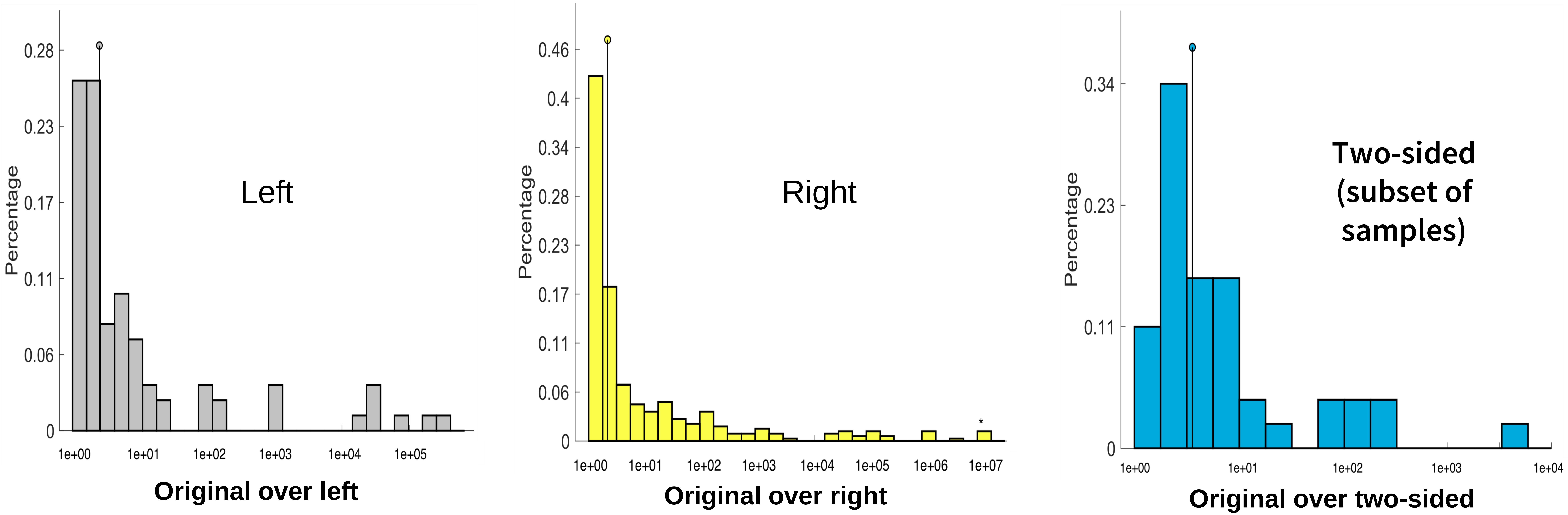
  e.g. Ruiz-scaling, matrix equilibration & balancing
- Not directly solvable using SDP
- Can be solved by iteratively fixing $D_1\,(D_2)$ and optimizing the other side

  Solving a sequence of SDPs
- Benchmark to answer questions:

  How far can diagonal preconditioners go?

  How good are those Heuristics?

Computational results: How far can optimal preconditioner go?



Distribution of condition number improvement on SuiteSparse matrix collection

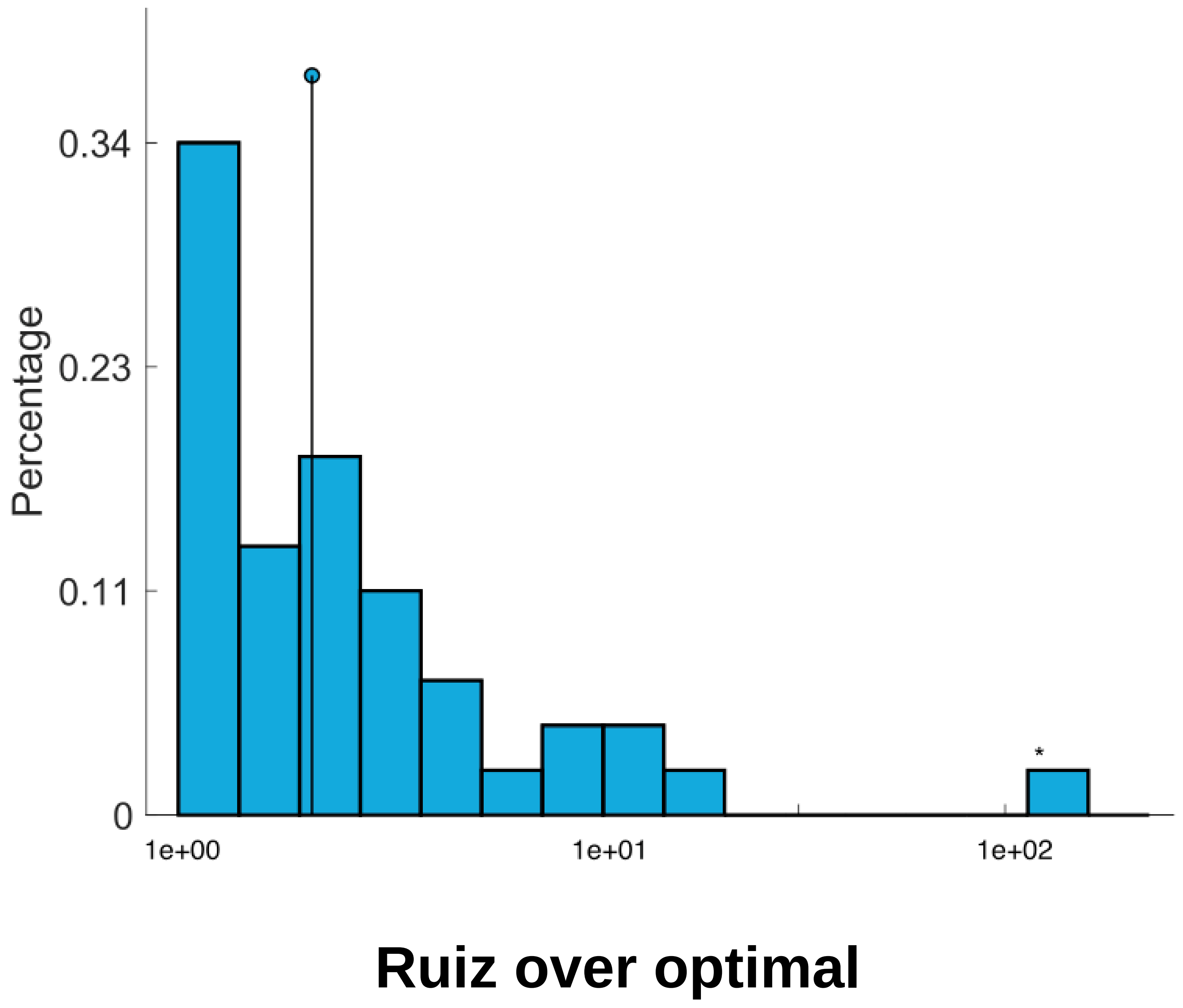
- A median of 2.2 factor of improvement for optimal right preconditioner
- 2.5 factor of improvement for optimal left preconditioner
- 3.6 factor of improvement for optimal two-sided preconditioner

Computational results: How good are the heuristic preconditioners

We use the optimal preconditioner to evaluate
two heuristic preconditioners: one-sided Jacobi and two-sided Ruiz

- A median factor of 1.5 improvement over Jacobi
- A median factor of 2.1 improvement over Ruiz
- For some matrices the improvement reaches >100

heuristics are often good, but sometimes harmful

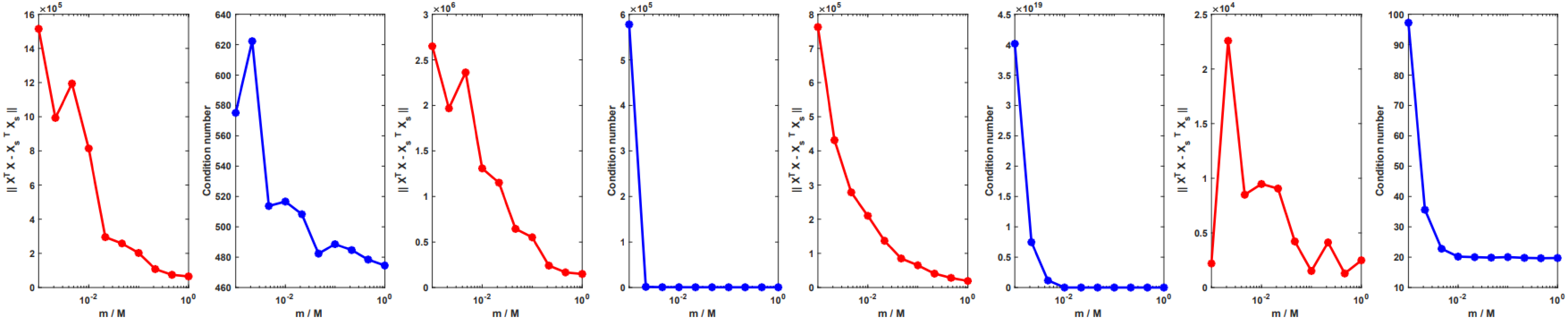


Computational results: Randomized preconditioner

- Many matrices result from statistical datasets
- $M = X^T X$ estimates the covariance matrix
- It suffices to use a few samples to approximate

How few?

As few as $\boldsymbol{O}(\mathbf{log}(\mathbf{sample}))$!



Experiment over regression datasets shows that

- It generally takes 1% to 5% of the samples to approximate well
- Scales well with dimension and saves much time for matrix-matrix multiplication

Takeaways

- Finding optimal (non)diagonal preconditioner can be modeled by SDP
- Optimal preconditioner exhibits nice empirical performance for real-life matrices
- Providing a benchmark for evaluating heuristic preconditioners
- Good for solving systems with fixed left-hand-side matrices

$$\begin{array}{ll} \min_{D,\kappa} & \kappa \\ \text{subject to} & D \preceq M \\ & \kappa D \succeq M \end{array}$$

The theory of optimal preconditioner is attractive, but

- For an $n \times n$ matrix, we need to solve a dual SDP of $n + 1$ variables
- Interior point method solves a $(n+1) \times (n+1)$ dense linear system in a iteration
- Not scalable to matrices of size 5000

Finding the optimal preconditioner seems impractical in a real-time fashion

What about an approximately optimal preconditioner?

Today’s Talk

I. The Optimal Diagonal Preconditioner via Semidefinite Programming

II. Towards Practical Approximate Optimal Diagonal Preconditioner

Approximately optimal preconditioner is acceptable

- Condition number optimization is different from common convex optimization problems
- Peformance of algorithms moderately depends on condition number

  e.g., $\mathcal{O}(\kappa \log(1/\varepsilon))$
- An error of condition number up to moderate **ε does not affect performance**
- We can be aggressive in the trade-off between accuracy and scalability

Our approach:

Step 1: we show that dimension of SDP can be reduced

Step 2: we show that the SDP can be solved via LP with cutting-planes

Step 1: Optimal combination of existing preconditioners

- The bottleneck of optimal diagonal preconditioner comes from $n + 1$ SDP variables
- Each "1" from n corresponds to a column of the identity matrix

  as if we are combining n bases in the space of diagonal preconditioner.

$$D = \sum_{i=1}^{n} E_i d_i$$

Focusing on the whole space is expensive. How about a subspace?

- Pick k "base" preconditioners $D_1, \dots, D_k$ that work well in practice

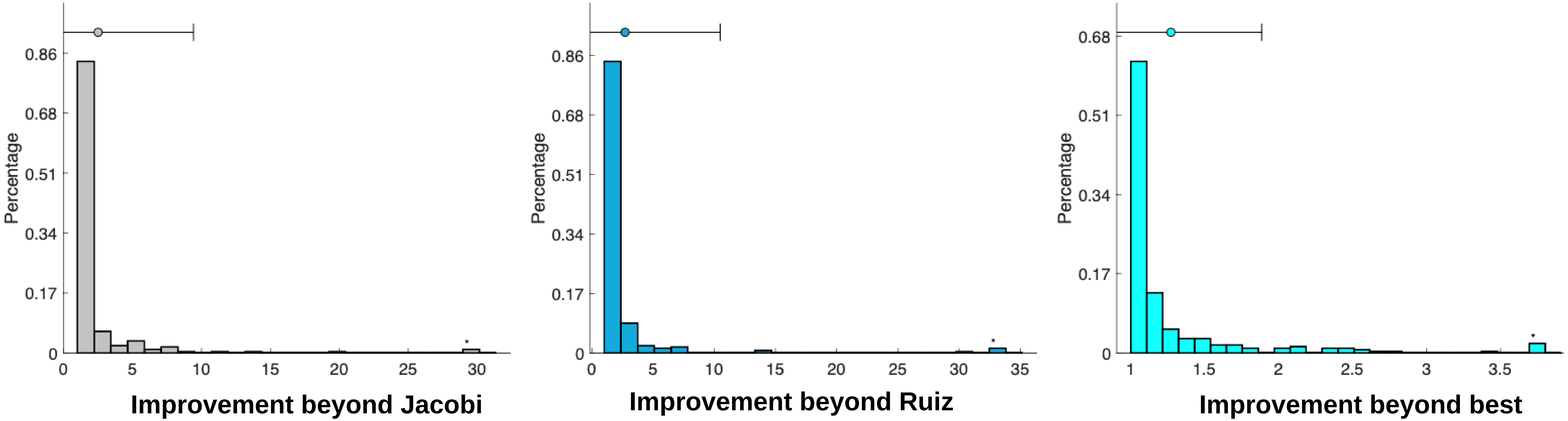
  e.g. Jacobi, Ruiz, Sparse approximate inverse ...
- Restrict preconditioner to lie in the subspace spanned by these bases
- Reducing the SDP to $k + 1$ variables
- Get the optimal combination of the basic preconditioners

  No worse than the best of them

$$D = \sum_{i=1}^{n} D_i \alpha_i$$

$$\begin{aligned} \max_{\alpha,\tau} \quad & \tau \\ \text{subject to} \quad & \sum_{i=1}^{n} D_i \alpha_i \preceq M \\ & \sum_{i=1}^{n} D_i \alpha_i \succeq M\tau \end{aligned}$$

Computational results: optimal combination of preconditioners



- Choosing three basis preconditioners: Jacobi, Ruiz and Identity
- Able to deal with sparse matrices of size up to 20000
- 2.5 factor of improvement beyond Jacobi
- 2.8 factor of improvement beyond Ruiz
- 1.2 factor of improvement beyond best among Jacobi/Ruiz/None

We are much more scalable now.
But solving an SDP is still not ideal
Can we go further?

Yes! We can even be "SDP-free"

Step 2: Semi-infinite linear programming and cutting plane method

We are faced with a dual SDP

- with very few dual variables

  in practice 3 to 10 base preconditioners are needed

- with most constraint matrices diagonal

$$\begin{aligned} \max_{\alpha,\tau} \quad & \tau \\ \text{subject to} \quad & \sum_{i=1}^{n} D_i \alpha_i \preceq M \\ & \sum_{i=1}^{n} D_i \alpha_i \succeq M\tau \end{aligned}$$

Recall that an SDP conic constraint $S \succcurlyeq 0$ can be represented by infinite linear constraints

$$C - \mathcal{A}^* y \succeq 0 \quad \Leftrightarrow \quad \langle a, (C - \mathcal{A}^* y) a \rangle \geq 0, \text{for all } a \in \mathbb{R}^n \quad \Leftrightarrow \quad \langle \mathcal{A}(a a^\top), y \rangle \leq a^\top C a$$

- the SDP can be written as an LP with infinite number of constraints and few variables
- we can employ a cutting plane/constraint generation approach to solve the LP
- similar to the interior point cutting plane method for semi-infinite programming

Cutting plane method for optimal preconditioner

To implement the cutting plane approach

- we initialize with a set of linear constraints
- solve the LP and obtain the LP solution

  the LP has very few variables
- call the separation oracle

  compute the minimum eigenvalue of the dual slack (efficiently computable using Lanczos iteration)

  If $\lambda_{\min}(C - \mathcal{A}^* y) < -\varepsilon$, then there exists $\langle d, (C - \mathcal{A}^* y) d \rangle < 0$

  cutting plane $\langle \mathcal{A}(dd^\top), y \rangle \leq d^\top C d$ is added to the problem
- iterate till convergence

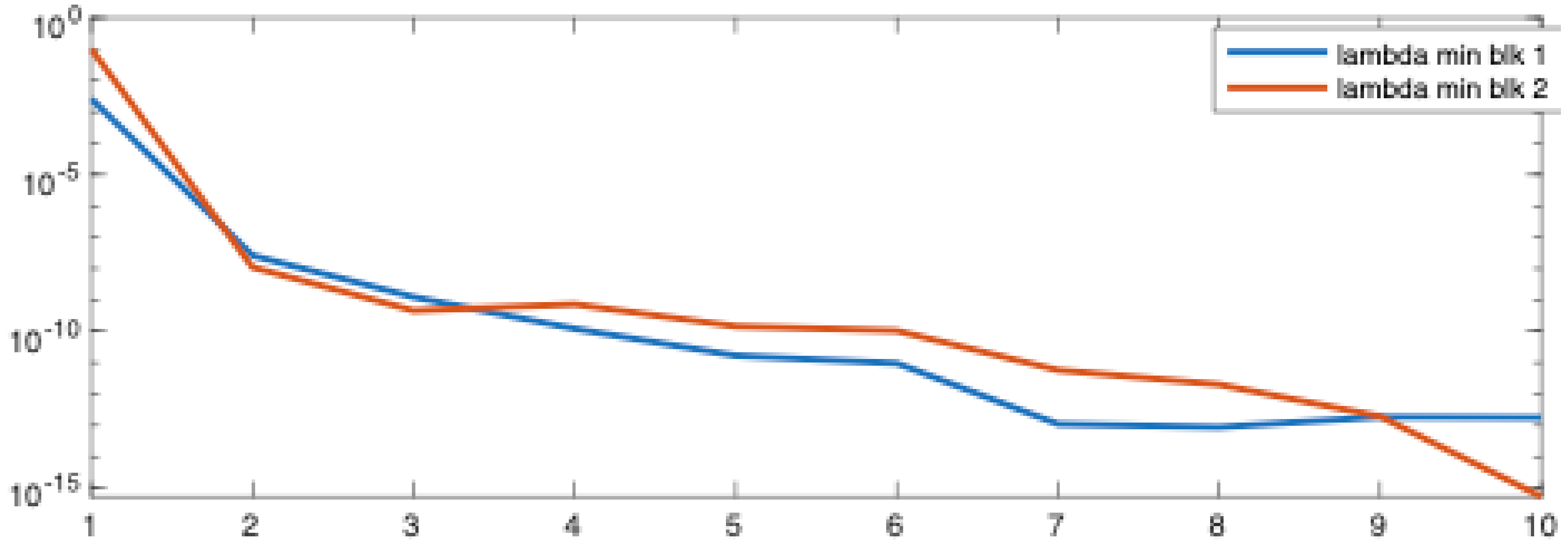
- We solve a sequence of low-dimension LPs rather than the original SDP
- LPs can be efficiently warm-started using dual simplex

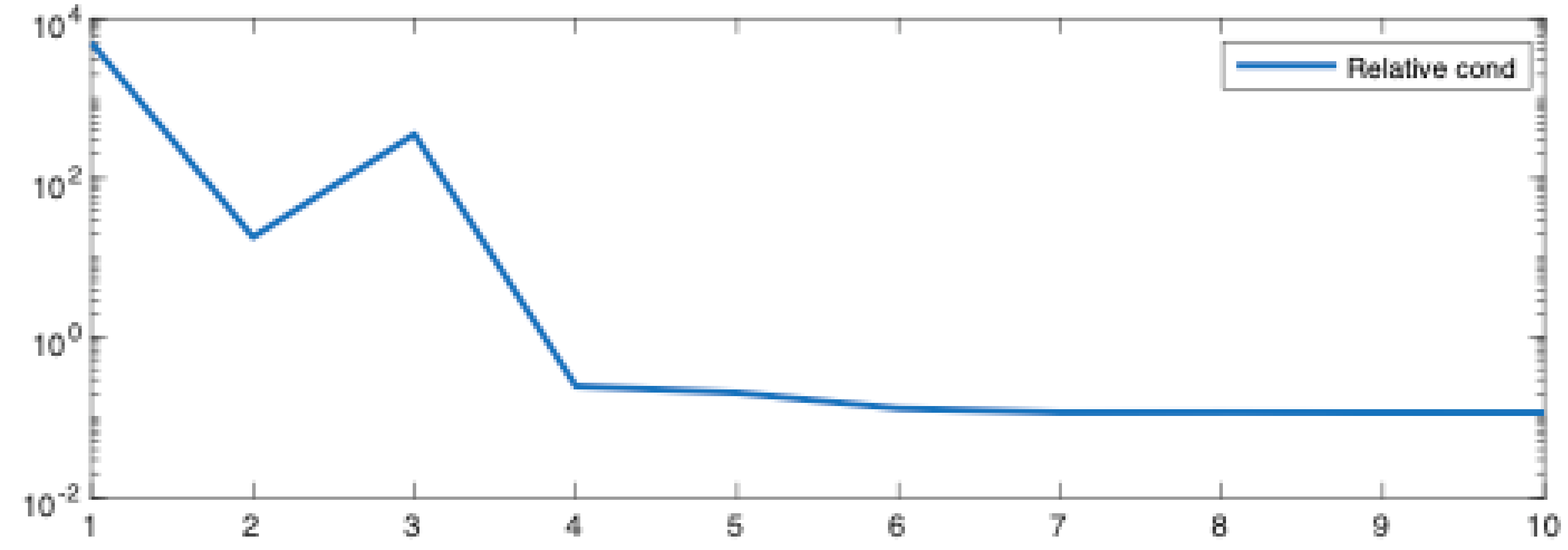
How well does the cutting plane approach work in practice?

Computational results: LP + cutting plane

How does the method work in practice?

- For moderate number (<30) of base preconditioners, only 5~20 LPs are needed to reach good accuracy
- The separation oracle runs very fast when the matrix is sparse
- Dual simplex solves the LPs efficiently
- A 10000 by 10000 sparse matrix needs <5 seconds scalable to very large matrices





x-axis: number of LP iterations
y-axis: up: violation of SDP conic constraint
low: relative optimality in condition number

Summary

- Finding the optimal (non)diagonal preconditioner can be modeled by SDP: another SDP application
- The optimal diagonal preconditioner serves as a benchmark and has desirable empirical performances compared to heuristic approaches

We further show that

- Finding the optimal combination of few heuristic diagonal preconditioners can be modeled by SDP, and it improves scalability of the SDP approach without compromising much performances
- The SDP from optimal combination of preconditioners can be efficiently solved using Semi-infinite optimization + LP dual simplex + cutting plane method,…

Finding approximate optimal diagonal preconditioners may be *scalable?*