Optimal Per-Coordinate Step-Sizes with Multidimensional Backtracking

"Adaptive" Optimization Methods

In gradient descent, using a step-size for each coordinate can give faster convergence. "Adaptive" methods try to automatically find these step-sizes, and avoid tuning

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \mathbf{P}_t \nabla f(\mathbf{x}_t)$$

Diagonal Matrix

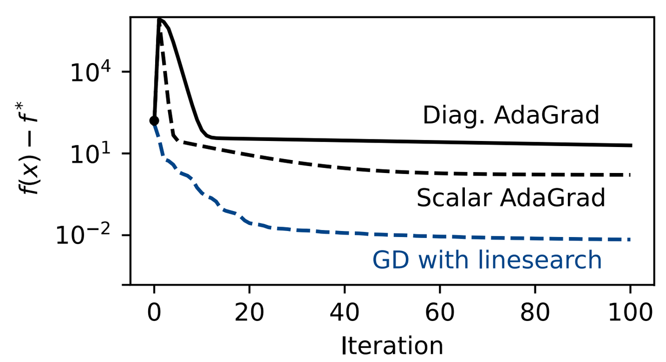
Adaptive Algorithms try to find a **good** matrix automatically

What does **GOOD** mean?

But there is often no definition of what "adaptive" means. This leads to methods with weak or no guarantees even for the "simple" deterministic, smooth, strongly convex f.

Online Learning Underperforms in Nice Problems

The only formal definition of adaptivity comes from **online learning** (OL) with **AdaGrad** as the classic example. Since OL is adversarial, AdaGrad underperforms on deterministic, smooth, strongly convex problems.



Defining our Goal: Optimal (Diagonal) Preconditioner

We define the optimal diagonal preconditioner $\mathbf{P}_*$ as the one that guarantees the best linear convergence rate for the method. This gives a condition between $\mathbf{P}$ and $\nabla f^2(x)$.

$(\mathbf{P}_*, \kappa_*)$ attains $\min_{\mathbf{P},\kappa} \kappa$ such that $\mathbf{P} \succ 0$ is diagonal and

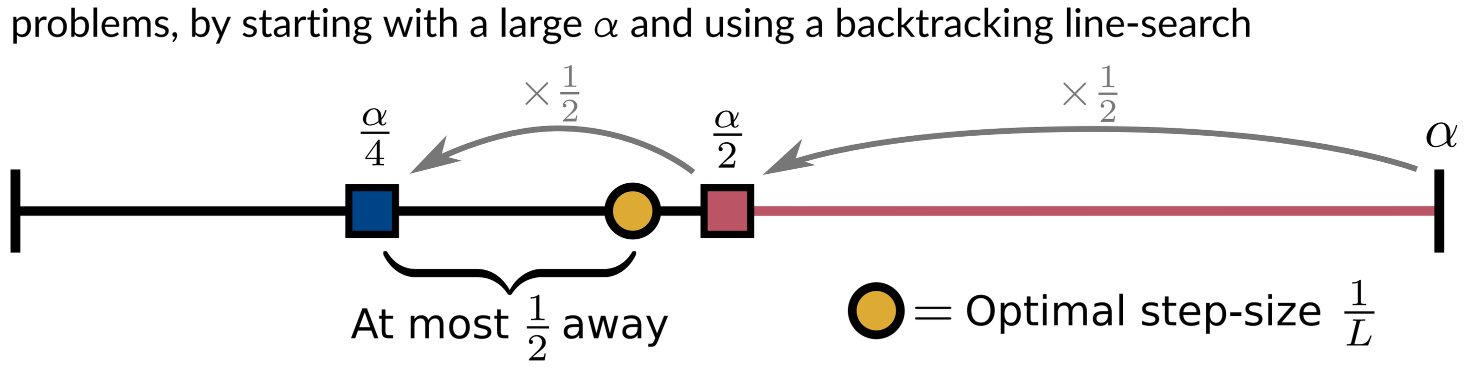
$$\tfrac{1}{\kappa}\mathbf{P}^{-1} \preceq \nabla^2 f(\mathbf{x}) \preceq \mathbf{P}^{-1} \quad \text{for all } \mathbf{x}$$

$\|\nabla f(\mathbf{x})\|^2_{\mathbf{P}}$ is big, i.e., $\tfrac{1}{2}\|\nabla f(\mathbf{x})\|^2_{\mathbf{P}} \geq \tfrac{1}{\kappa}(f(\mathbf{x}) - f(\mathbf{x}_*))$

Progress proportional to $\|\nabla f(\mathbf{x})\|^2_{\mathbf{P}}$, i.e., $f(\mathbf{x} - \mathbf{P}\nabla f(\mathbf{x})) \leq f(\mathbf{x}) - \tfrac{1}{2}\|\nabla f(\mathbf{x})\|^2_{\mathbf{P}}$

The Scalar Case: Backtracking line-search

For a scalar step-size, we know how to automatically find a good step-size in smooth problems, by starting with a large α and using a backtracking line-search



■ satisfies the **sufficient progress** condition:

$$f(\mathbf{x} - \alpha\nabla f(\mathbf{x})) \leq f(\mathbf{x}) - \alpha\tfrac{1}{2}\|\nabla f(\mathbf{x})\|^2_2$$

From Line-search to Preconditioner Search

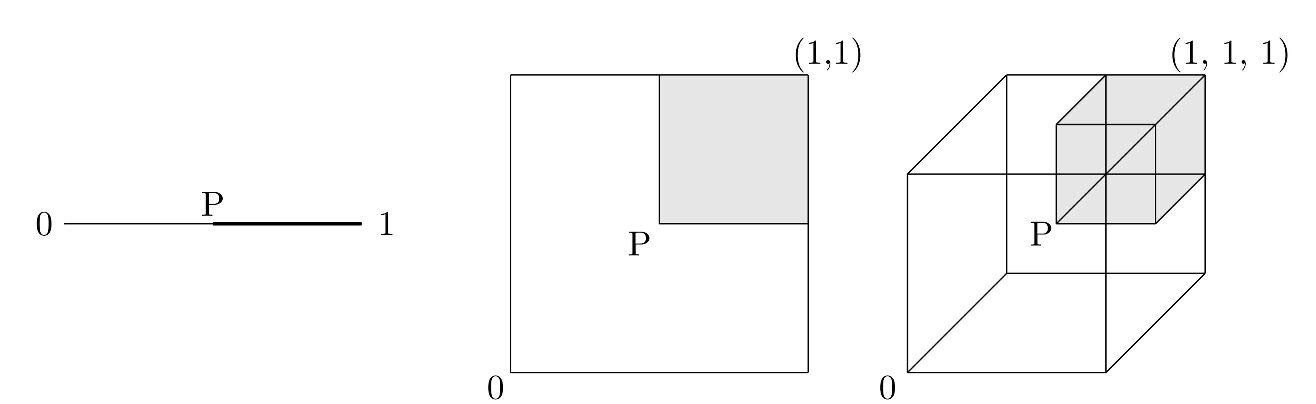
Goal: mimic the guarantees of a line-search when searching for $\mathbf{P}$, checking

$$f(\mathbf{x} - \mathbf{P}\nabla f(\mathbf{x})) \leq f(\mathbf{x}) - \|\nabla f(\mathbf{x})\|^2_{\mathbf{P}}$$

- **Large progress**: When $\mathbf{P}$ is accepted, its progress should be comparable to $\mathbf{P}_*$
- **Volume Shrinkage**: Shrink volume of search space by a constant when $\mathbf{P}$ is rejected.

Curse of dimensionality

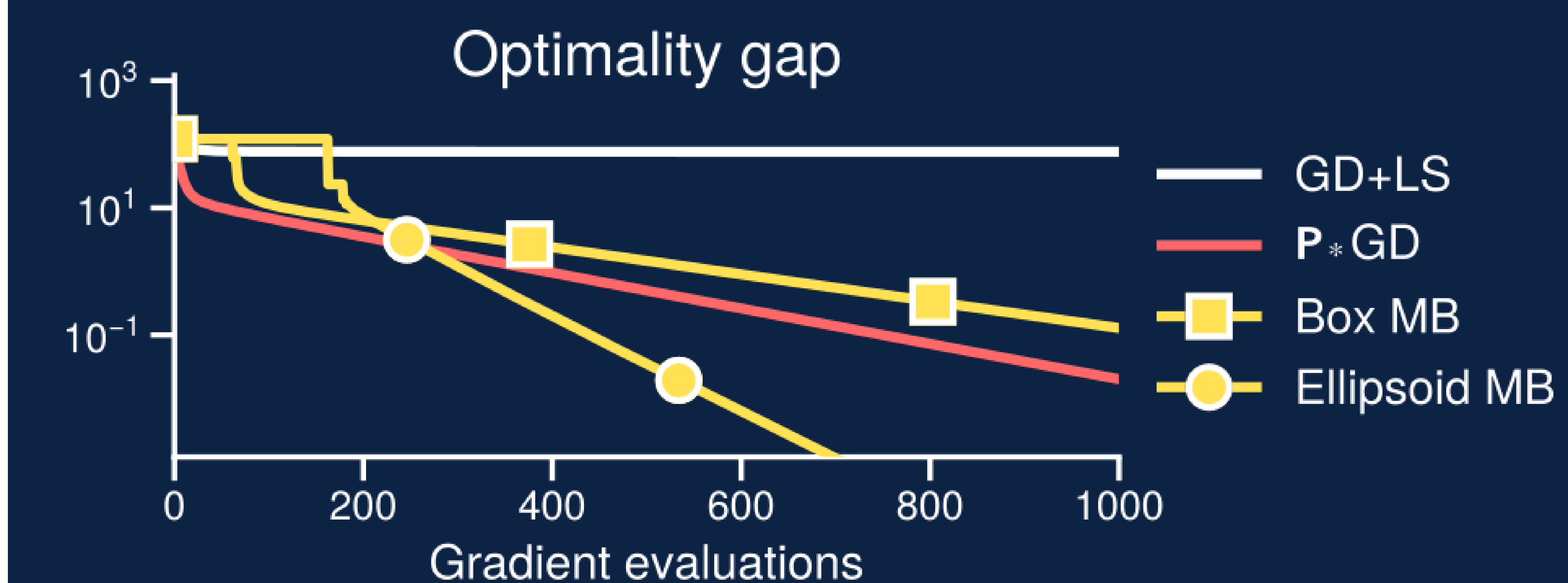
If $\mathbf{P}$ is rejected, removing only candidates $\mathbf{Q} \succeq \mathbf{P}$ does not remove enough volume.



Many optimization methods aim to be *adaptive* **but without defining what that means.**

Adaptive algorithms from online learning (e.g., **AdaGrad**) **need decreasing step-sizes**, making them slow on simple problems.

We define adaptivity for simple problems (deterministic, smooth, strongly-convex) and develop **Multidimensional Backtracking (MB)**, a generalization of a backtracking line-search that finds provably near-optimal per-coordinate step-sizes.



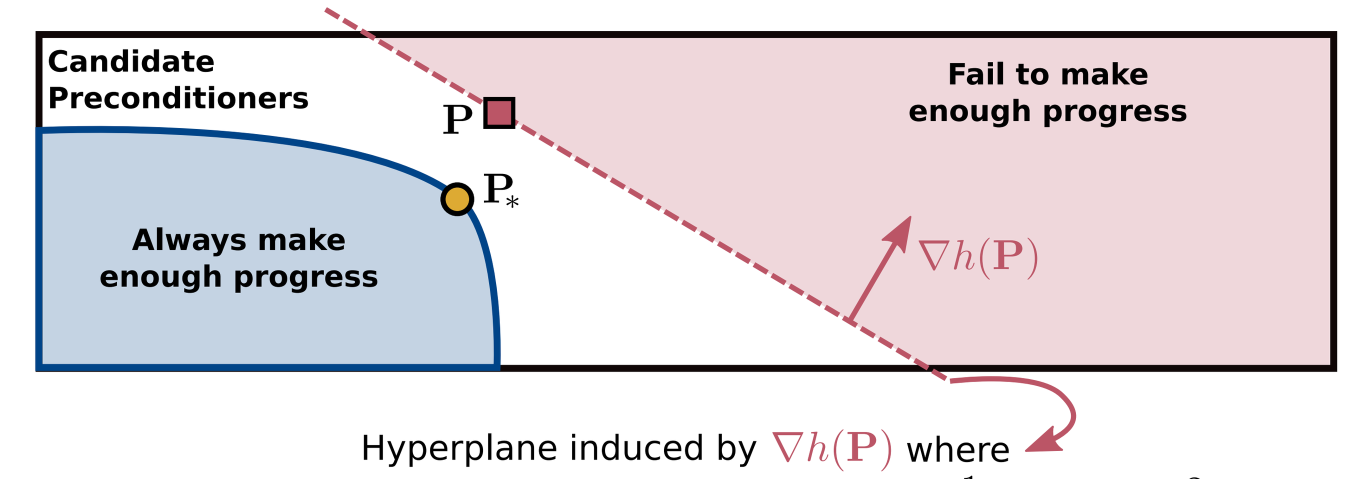
Frederik Kunstner
Victor Sanches Portella
Mark Schmidt
Nick Harvey

THE UNIVERSITY OF BRITISH COLUMBIA
Canada CIFAR AI Chair (Amii)

arxiv.org/pdf/2306.02527

Hypergradient as a Separating Hyperplane

The key idea to circumvent this problem is to use the gradient with respect to $\mathbf{P}$

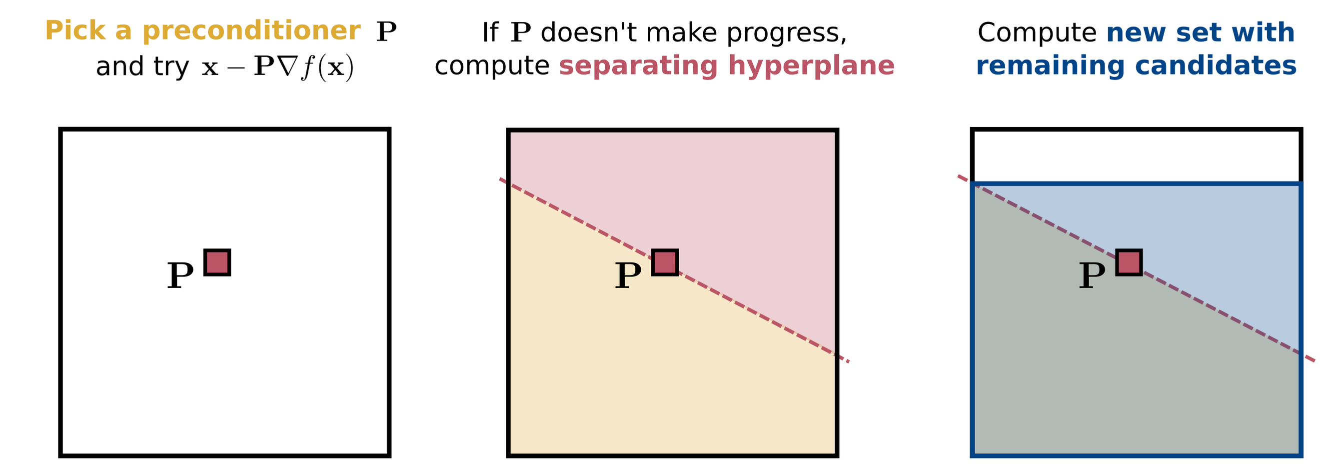


Hyperplane induced by $\nabla h(\mathbf{P})$ where

$$h(\mathbf{P}) = f(\mathbf{x}_t - \mathbf{P}\nabla f(\mathbf{x}_t)) - f(\mathbf{x}) + \tfrac{1}{2}\|\nabla f(\mathbf{x})\|^2_{\mathbf{P}}$$

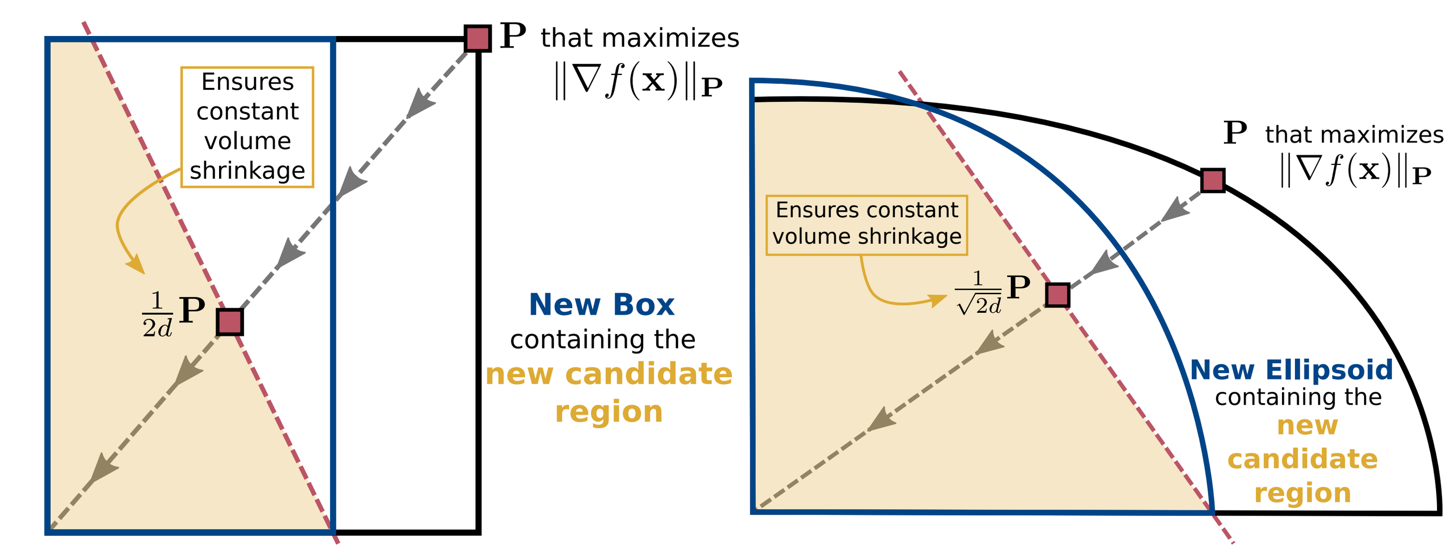
Overview of the Algorithm

With the hyperplane, we can build a cutting plane method to search for a good $\mathbf{P}$



Guarantees Through Cutting Planes

We develop efficient cutting plane methods by using the separating hyperplanes obtained via hypergradient information and using the symmetry of the problem.



Competitive with the optimal preconditioner

Multidimensional backtracking guarantees

$$f(\mathbf{x}_{t+1}) - f(\mathbf{x}_*) \leq \left(1 - \frac{1}{\sqrt{2d}}\frac{1}{\kappa_*}\right)(f(\mathbf{x}_t) - f(\mathbf{x}_*))$$

Searching for a preconditioner has a **cost**

If there is a good preconditioner, then **convergence guarantees are better**

The number of backtracking steps is $O(d\log(L p_0))$ if $\mathbf{P}_0 = p_0\mathbf{I}$.

Empirical Performance

More stable than existing heuristics, still works in high dimensions ($d = 10^6$).