

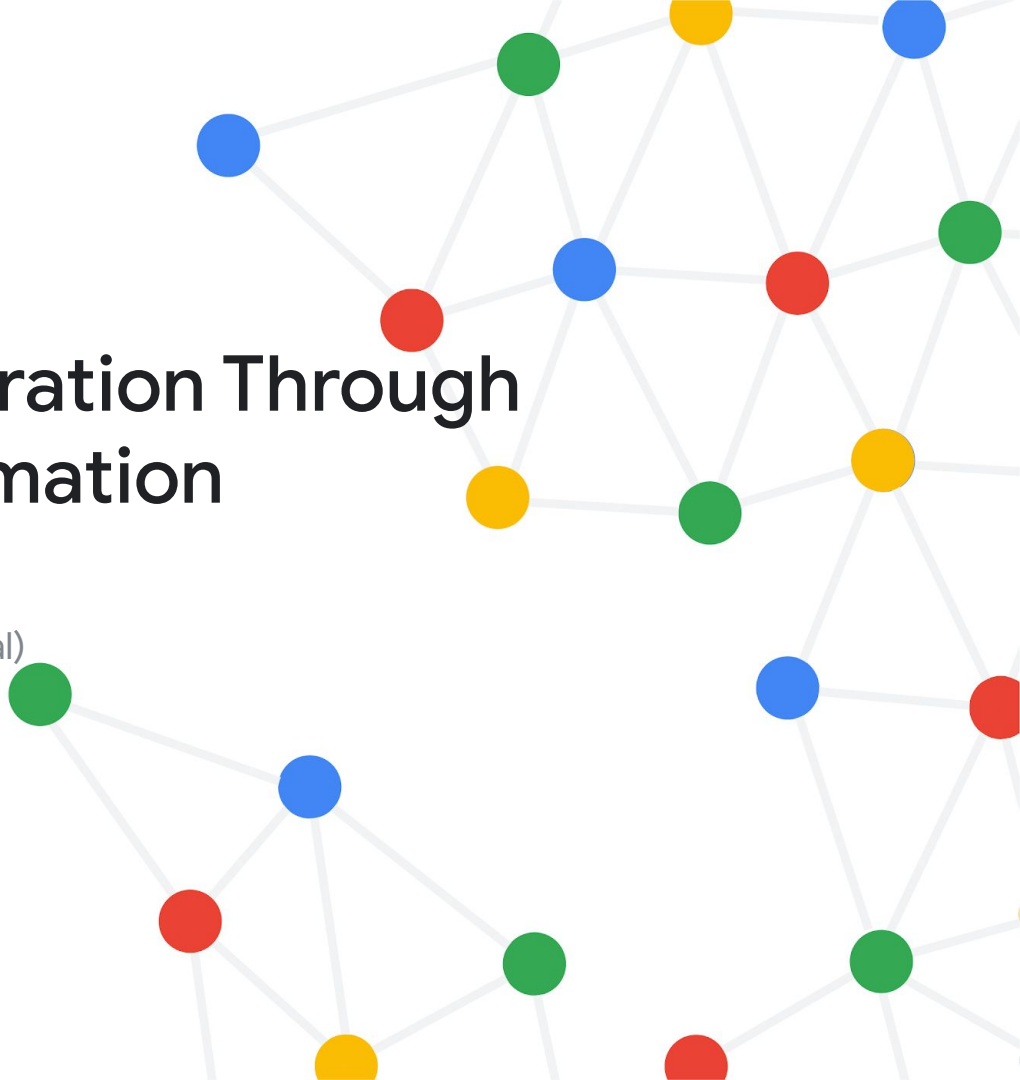
# Average-case Acceleration Through Spectral Density Estimation

Fabian Pedregosa (Google Research)

Damien Scieur (Samsung SAIT AI Lab, Montréal)

 Google Research

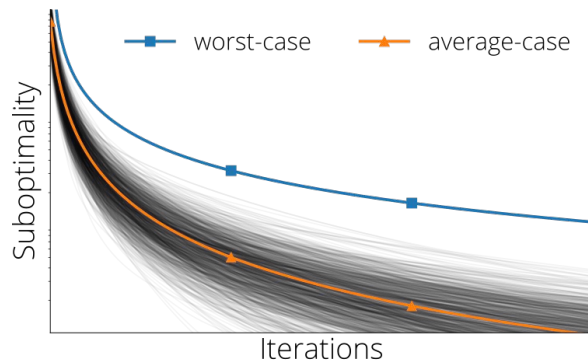
**SAMSUNG**  
Advanced Institute  
of Technology AI Lab  
Montréal



# Complexity Analysis in Optimization

## Worst-case analysis

- ✓ Bound on the complexity for *any* input.
- ✗ Potentially worse than observed runtime.



## Simplex method (Dantzig, '98, Spielman & Teng '04)

- ✗ Exponential worst-case.
- ✓ Runtime typically polynomial.

# Average-case Complexity

- ✓ Complexity *averaged* over all problem instances.
- ✓ Representative of the typical complexity.

Better bounds, sometimes better algorithms

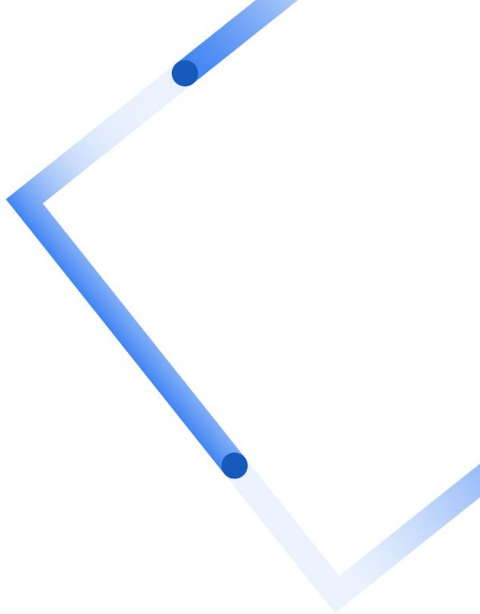
→ Quicksort (Hoare '62): Fast average-case sorting

*Rarely used in optimization*

## Main contributions

Average-case analysis for optimization on quadratics.

Optimal methods under this analysis.



# Problem Distribution: Random Quadratics

$$\min_{\mathbf{x} \in \mathbb{R}^d} \left\{ f(\mathbf{x}) \stackrel{\text{def}}{=} \frac{1}{2} (\mathbf{x} - \mathbf{x}^*)^\top \mathbf{H} (\mathbf{x} - \mathbf{x}^*) \right\},$$

where  $\mathbf{H}$ ,  $\mathbf{x}^*$  are random matrix, vector.

- ✓ exact runtime known depends on eigenvalues( $\mathbf{H}$ ).
- ✓ shares (some) dynamics of real problems, e.g., Neural Tangent Kernel (Jacot et al., 2018).

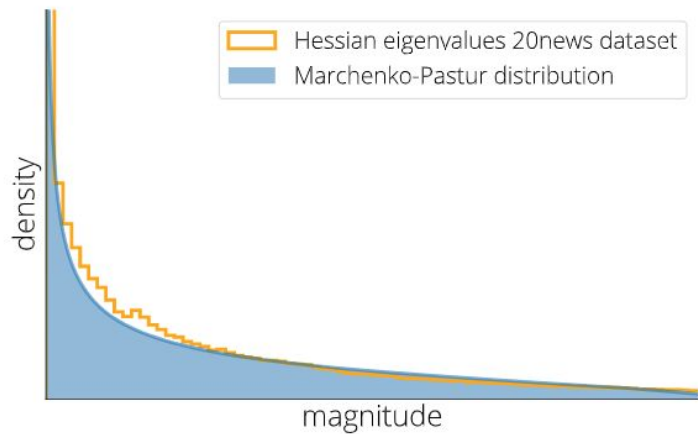
# Example: Random Least Squares

$$\min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2, \text{ with } \mathbf{b} = \mathbf{A}\mathbf{x}^*$$

When elements of  $\mathbf{A}$  are iid, standardized:

Spectrum of  $\mathbf{H}$  will be close to

Marchenko-Pastur.



# Expected Error For Gradient-Based Methods

$$\text{expected error} = \mathbb{E} \|\mathbf{x}_t - \mathbf{x}^*\|^2 = \overbrace{R^2}^{\text{initialization}} \int_{\mathbb{R}} \underbrace{P_t^2}_{\text{algorithm}} \underbrace{d\mu}_{\text{problem}}$$

$R^2$  is the **distance to optimum at initialization**  $\mathbb{E} \|\mathbf{x}_0 - \mathbf{x}^*\|^2$ . Fixed

Problem difficulty represented by **expected density Hessian eigenvalue**  $d\mu$

$P_t$  is a polynomial of degree  $t$  determined from the **optimization algorithm**.

Flexible: algorithm design

# Average-case Optimal Method

**Goal:** Find method with minimal expected error =  $\int_{\mathbb{R}} \underbrace{R^2}_{\text{initialization}} \underbrace{P_t^2}_{\text{algorithm}} \underbrace{d\mu}_{\text{problem}}$

**Algorithms**  $\leftrightarrow$  **Polynomials**

Find **polynomial**  $P_t$  of degree  $t$  that minimizes expected error (with proper normalization).

**Solution:** Polynomial of degree  $t$ , orthogonal wrt to  $\lambda d\mu(\lambda)$ .



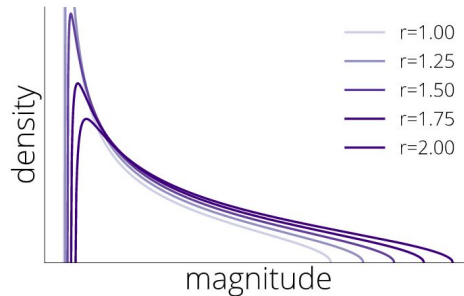
# Marchenko-Pastur Acceleration

Model for  $d\mu = \text{Marchenko-Pastur}(r, \sigma)$ .

$r$  and  $\sigma$  estimated from:

- Largest eigenvalue
- Trace of H

No need to know strong convexity constant.



## Algorithm

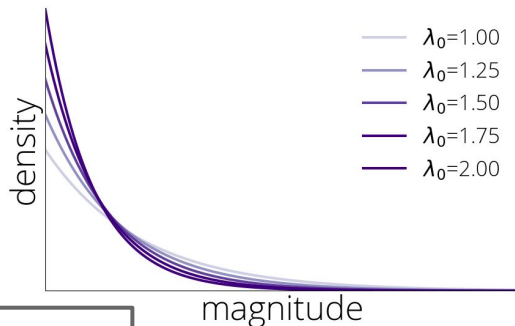
$$\mathbf{x}_t = \mathbf{x}_{t-1} + \underbrace{a_t(r, \sigma)}_{\text{momentum}} (\mathbf{x}_{t-2} - \mathbf{x}_{t-1}) + \underbrace{b_t(r, \sigma)}_{\text{step-size}} \nabla f(\mathbf{x}_{t-1})$$

Simple momentum-like method, low memory requirements.

# Decaying Exponential Acceleration

**Model for**  $d\mu$  = decaying exponential( $\lambda_0$ ).

**Unbounded** largest eigenvalue. Only access to  $\text{Tr}(\mathbf{H})$ .

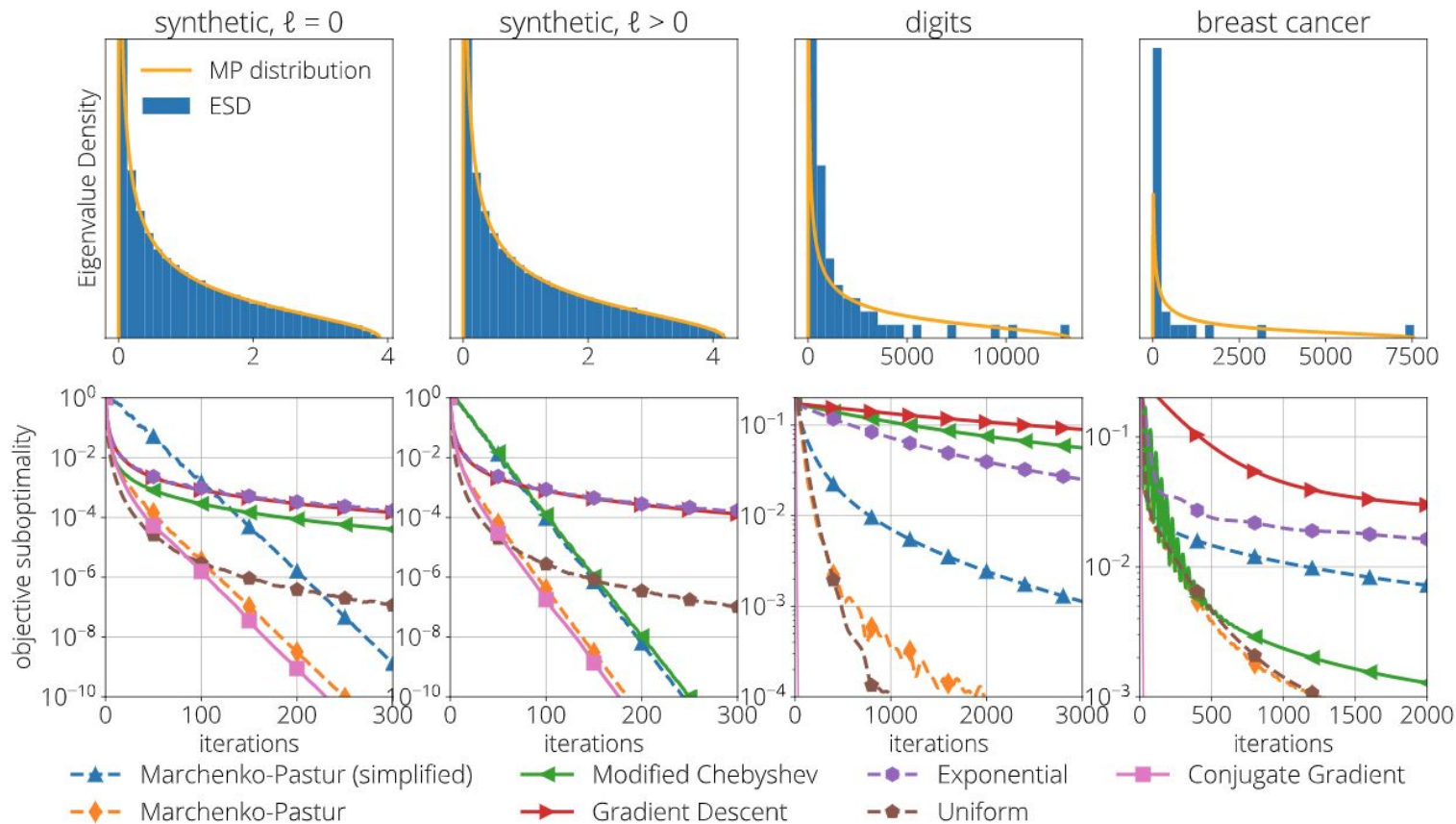


## Algorithm

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \frac{t-1}{t+1}(\mathbf{x}_{t-1} - \mathbf{x}_{t-2}) - \frac{\lambda_0}{t+1} \nabla f(\mathbf{x}_{t-1})$$

- Decaying step-size
- Similar to Polyak averaging

# Benchmarks: Least Squares



# Conclusions

Average-case analysis based on random quadratics.

Optimal methods under different eigenvalue distribution.

✓ Acceleration without knowledge of strong convexity.

In **paper**

+ More methods, convergence rates, empirical extension to non-quadratic objectives.

**Follow-up work** on asymptotic analysis

(Scieur and P., "*Universal Average-Case Optimality of Polyak Momentum*")