Subsampled Online Matrix Factorization with Convergence Guarantees

Arthur Mensch(1) Julien Mairal(2) Gaël Varoquaux(1) Bertrand Thirion(1)

(1)Parietal team, Inria, CEA, Neurospin, Paris-Saclay University. Gif-sur-Yvette, France

(2)Thoth team, Inria. Grenoble, France

Matrix factorization

- Decompose $\mathbf{X} \in \mathbb{R}^{p\times n} \approx \mathbf{DA} \in \mathbb{R}^{p\times k} \times \mathbb{R}^{k\times n}$

$$\min_{\mathbf{D}\in\mathcal{C}, \mathbf{A}\in\mathbb{R}^{k\times n}} \|\mathbf{X}-\mathbf{DA}\|_2^2 + \lambda\Omega(\mathbf{A}) \qquad (1)$$

- Sparse/dense and/or positive factors

- Elastic-net constraints and penalties

$$\Omega(\alpha) \triangleq (1-\nu)\|\alpha\|_2^2 + \nu\|\alpha\|_1$$

$$\mathcal{C} \triangleq \{\mathbf{D}\in\mathbb{R}^{p\times k} / \|\mathbf{d}^{(j)}\| \triangleq \mu\|\mathbf{d}^{(j)}\|_1 + (1-\mu)\|\mathbf{d}^{(j)}\|_2^2 \le 1\}$$

How to factorize matrices huge in both directions ?

Hyperspectral patches	fMRI data
103 GB	**2TB**
$p = 2\cdot 10^5$,	$p = 6\cdot 10^4$
$n = 2\cdot 10^6$	$n = 2\cdot 10^6$

- Dictionary learning
- Non-negative matrix factorization
- Sparse components decomposition

Prior: Large number of columns n

- Direct minimization of (1): SGD in $(\mathbf{D}, \mathbf{A})$
- Empirical risk minimization

$$\mathbf{D} = \operatorname*{argmin}_{\mathbf{D}\in\mathcal{C}} \bar{f} \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^n f^{(i)}(\mathbf{D}), \qquad (2)$$

$$f^{(i)}(\mathbf{D}) = \min_{\alpha\in\mathbb{R}^k} \frac{1}{2}\|\mathbf{x}^{(i)} - \mathbf{D}\alpha\|_2^2 + \lambda\,\Omega(\alpha).$$

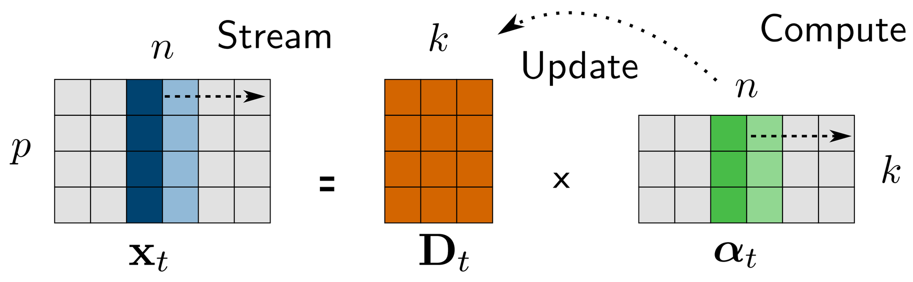
- **Online algorithm** for matrix factorization [2]

$\Rightarrow$ iterate sequence $(\mathbf{D}_t)_t \to$ critical point of $\bar{f}$

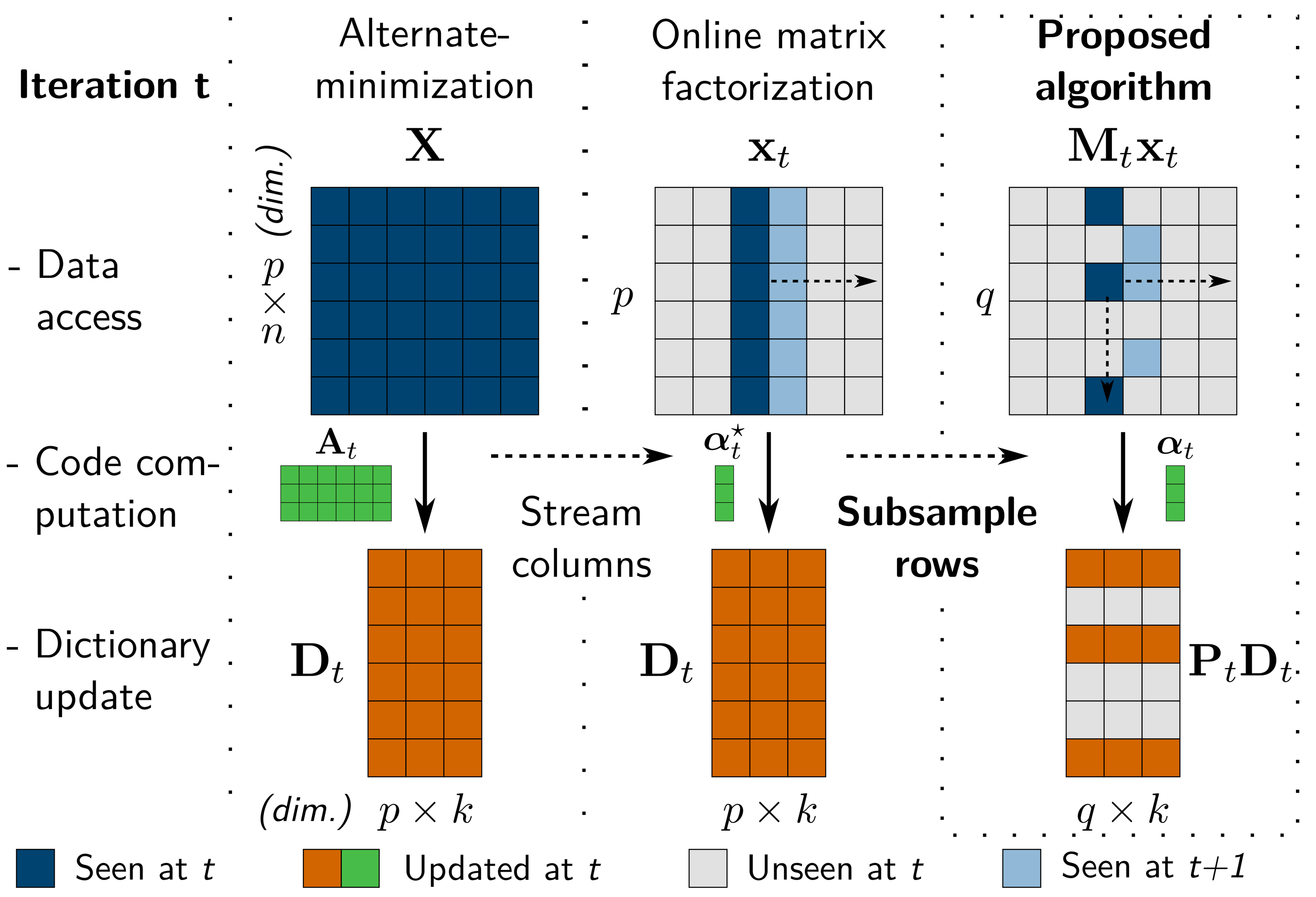
Prior: Large number of rows p

How to reduce the number of features ?

- Random projection to lower dimension (Johnson-Lindenstrauss lemma)
- Randomized linear algebra (*e.g.*, for SVD)



Stochastic access to columns, stochastic subsampling of rows



Introducing subsampling in online matrix factorization [2]

- Sample $\mathbf{x}_t$ from the columns $\{\mathbf{x}^{(i)}\}_i$, defining $f_t(\mathbf{D}) \triangleq \min_{\alpha\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{x}_t - \mathbf{D}\alpha\|_2^2 + \lambda\Omega(\alpha)$
- **Compute code** from dictionary $\mathbf{D}_{t-1}$: $\alpha_t = \operatorname{argmin}_{\alpha\in\mathbb{R}^k} \frac{1}{2}\|\mathbf{x}_t - \mathbf{D}_{t-1}\alpha\|_2^2 + \lambda\Omega(\alpha_t)$ — $\mathcal{O}(p)$

$$g_t(\mathbf{D}) = \frac{1}{2}\|x_t - \mathbf{D}_{t-1}\alpha_t\|_2^2 + \lambda\Omega(\alpha), \quad \text{surrogate of } f_t\text{:} \quad g_t(\mathbf{D}_{t-1}) = f_t(\mathbf{D}_{t-1}), \forall \mathbf{D}, g_t(\mathbf{D}) \ge f_t(\mathbf{D})$$

- **Update aggregated surrogate** and statistics — $\mathcal{O}(p)$

$$\bar{g}_t(\mathbf{D}) \triangleq \Big(\frac{1}{t}\sum_{s=1}^t \frac{1}{2}\|\mathbf{x}_s - \mathbf{D}\alpha_s\|_2^2 + \lambda\Omega(\alpha_s)\Big)$$

$$\ge \bar{f}_t(\mathbf{D}) \triangleq \frac{1}{t}\sum_{s=1}^t f_s(\mathbf{D})$$

$$(\bar{\mathbf{A}}_t, \bar{\mathbf{B}}_t) = (1-\frac{1}{t})(\bar{\mathbf{A}}_{t-1}, \bar{\mathbf{B}}_{t-1}) + \frac{1}{t}(\alpha_t\alpha_t^\top, \mathbf{x}_t\alpha_t^\top)$$

- **Minimize surrogate.** Block coordinate descent — $\mathcal{O}(p)$

$$\mathbf{D}_t \in \operatorname*{argmin}_{\mathbf{D}\in\mathcal{C}} \bar{g}_t(\mathbf{D}) = \operatorname*{argmin}_{\mathbf{D}\in\mathcal{C}} \operatorname{Tr}\big(\frac{1}{2}\mathbf{D}^\top\mathbf{D}\bar{\mathbf{C}}_t - \mathbf{D}^\top\mathbf{B}_t\big)$$

Random subsampling

- Select random rows

$$\mathbf{M}_t \sim \mathit{Bernouilli}\,(\frac{1}{r}\mathbf{I}_p)$$

$$\mathbb{E}[\|\mathbf{M}_t\mathbf{x}_t\|_0] = \frac{p}{r} \triangleq r$$

$$\mathbb{E}[\mathbf{M}_t\mathbf{x}_t] = \mathbf{x}_t$$

- Replace $\mathbf{x}_t \to \mathbf{M}_t\mathbf{x}_t$
- $\mathcal{O}(p) \to \mathcal{O}(q)$ / **iteration**

Adapt online matrix factorization to use $(\mathbf{M}_t\mathbf{x}_t)_t$ instead of $(\mathbf{x}_t)_t$

Code computation — q-costly safe approximations

- Original online matrix factorization dominated by the computation of $\mathbf{G}_t^\star$, $\beta_t^\star$ for high p

$$\alpha_t^\star = \operatorname*{argmin}_{\alpha\in\mathbb{R}^k} \frac{1}{2}\alpha^\top\mathbf{G}_t^\star\alpha - \alpha^\top\beta_t^\star + \lambda\Omega(\alpha), \quad \text{where} \quad \mathbf{G}_t^\star = \mathbf{D}_{t-1}^\top\mathbf{D}_{t-1} \text{ and } \beta_t^\star = \mathbf{D}_{t-1}^\top\mathbf{x}_t$$

- **Estimators** using rows from $\mathbf{M}_t$ only: $\beta_t = \mathbf{D}_{t-1}^\top\mathbf{M}_t\mathbf{x}_t$, $\mathbf{G}_t = \mathbf{D}_{t-1}\mathbf{M}_t\mathbf{D}_{t-1}$. α_t solves (from [3])

$$\alpha_t = \operatorname*{argmin}_{\alpha\in\mathbb{R}^k} \frac{1}{2}\|\mathbf{M}_t(\mathbf{x}^t - \mathbf{D}_{t-1}^\top\alpha)\|_2^2 + \lambda\Omega(\alpha). \quad \mathcal{O}(q) \quad \textbf{breaks convergence guarantees}$$

↑ Define **averaged estimates**, updated online. Keep $2n$ estimators, written $(\mathbf{G}_t^{(i)}, \beta_t^{(i)})_{1\le i\le n}$ such that

$$\begin{cases} \beta_t^{(i)} = (1-\gamma)\mathbf{G}_{t-1}^{(i)} + \gamma\mathbf{D}_{t-1}^\top\mathbf{M}_t\mathbf{x}^{(i)} \\ \mathbf{G}_t^{(i)} = (1-\gamma)\mathbf{G}_{t-1}^{(i)} + \gamma\mathbf{D}_{t-1}^\top\mathbf{M}_t\mathbf{D}_t^{(i)} \end{cases} \text{if } i \text{ is s.t. } \mathbf{x}^{(i)} = \mathbf{x}_t \qquad \begin{cases} \beta_t^{(i)} = \beta_t^{(i)} \\ \mathbf{G}_t^{(i)} = \mathbf{G}_t^{(i)} \end{cases} \text{otherwise}$$

$$\text{and set} \quad \mathbf{G}_t \triangleq \mathbf{G}_t^{(i)} = \sum_{s\le t, \mathbf{x}_s = \mathbf{x}^{(i)}} \gamma_{s,t}^{(i)}\mathbf{D}_{s-1}^\top\mathbf{M}_s\mathbf{D}_{s-1}, \quad \beta_t \triangleq \beta_t^{(i)} = \sum_{s\le t, \mathbf{x}_s = \mathbf{x}^{(i)}} \gamma_{s,t}^{(i)}\mathbf{D}_{s-1}^\top\mathbf{M}_s\mathbf{x}^{(i)},$$

- $(\alpha_t)_t$ closer and closer to $(\alpha_t^\star)_t \to$ **safe approximation** May also maintain $\mathbf{G}_t^\star$ at cost $\propto q$

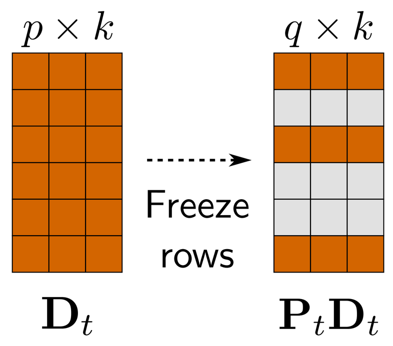
[1] J. Mairal. Stochastic majorization-minimization algorithms for large-scale optimization. In *Advances in Neural Information Processing Systems*, 2013.

[2] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, 11:19–60, 2010.

[3] A. Mensch, J. Mairal, B. Thirion, and G. Varoquaux. Dictionary learning for massive matrix factorization. In *International Conference on Machine Learning*, 2016.

[4] A. Mensch, J. Mairal, G. Varoquaux, and B. Thirion. Subsampled online matrix factorization with convergence guarantees. In *NIPS Workshop on Optimization for Machine Learning*, 2016.

Dictionary update — freezing constraints for partial minimization



- Update only the **rows of $\mathbf{D}_t$ selected by $\mathbf{M}_t$**

$$\mathbf{D}_t \in \operatorname*{argmin}_{\substack{\mathbf{D}\in\mathcal{C} \\ \mathbf{P}_t^\perp\mathbf{D} = \mathbf{P}_t^\perp\mathbf{D}_{t-1}}} \frac{1}{2}\operatorname{Tr}(\mathbf{D}^\top\mathbf{D}\bar{\mathbf{C}}_t) - \operatorname{Tr}(\mathbf{D}^\top\bar{\mathbf{B}}_t),$$

- Projector $\mathbf{P}_t \in \mathbb{R}^{q\times p}$ reduces the problem dimension

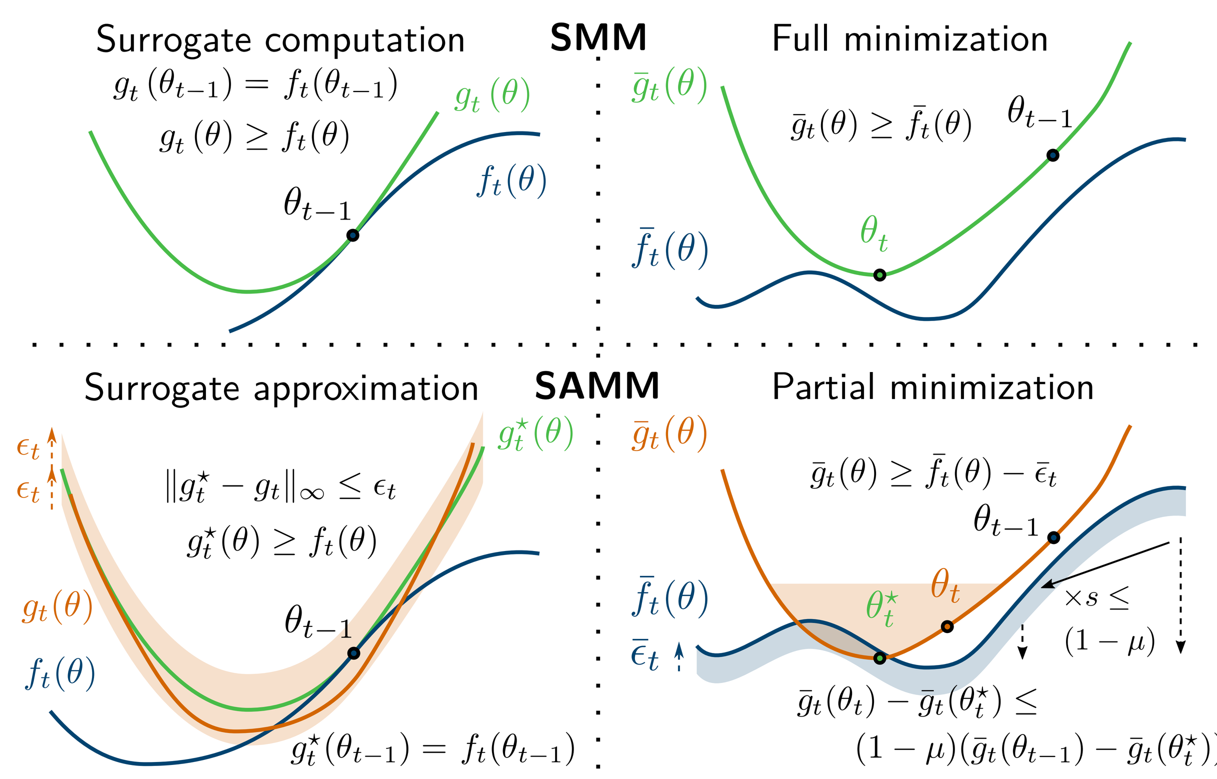
↑ Reduces to constrained minimization of strongly-convex function in the **reduced space** $\mathbb{R}^{q\times k}$:

$$\mathbf{P}_t^\perp\mathbf{D}_t \leftarrow \operatorname*{argmin}_{\mathbf{D}^r\in\mathcal{C}_r} \frac{1}{2}\operatorname{Tr}(\mathbf{D}^{r\top}\mathbf{D}^r\bar{\mathbf{C}}_t) - \operatorname{Tr}(\mathbf{D}^{r\top}\bar{\mathbf{P}}_t\mathbf{B}_t)$$

$$\text{where } \mathcal{C}^r = \{\mathbf{D}^r\in\mathbb{R}^{q\times k} / \forall j\in[0,k-1], \|\mathbf{d}^{r(j)}\| \le 1 - \|\mathbf{d}_{t-1}^{(j)}\| + \|\mathbf{P}_t\mathbf{d}_{t-1}^{(j)}\|\}.$$

- Solvable by **BCD** with k **blocks of dimension** q — one pass only as in [2], **complexity in $\mathcal{O}(q)$**
- Only $\mathbf{P}_t\bar{\mathbf{B}}_t$ is used in BCD: update it before (q-costly) + update $\mathbf{P}_t^\perp\bar{\mathbf{B}}_t$ in **parallel** with $\mathbf{D}_t$ update

Convergence analysis: extending stochastic majorization-minimization [1]



Same assumptions as [2]

- $(\mathbf{D}_t^\top\mathbf{D}_t)_t \succ \nu\mathbf{I}_k$
- Iteration weight seq. $(w_t)_t$
- Sample weight seq. $(\gamma_c)_c$

Asymptotic correctness

- $(\mathbf{D}_t)_t \to \mathbf{D}_\infty$ stationary point of empirical risk (2)

Stochastic approximate majorization-minimization

- Controled approximations of SMM [1]

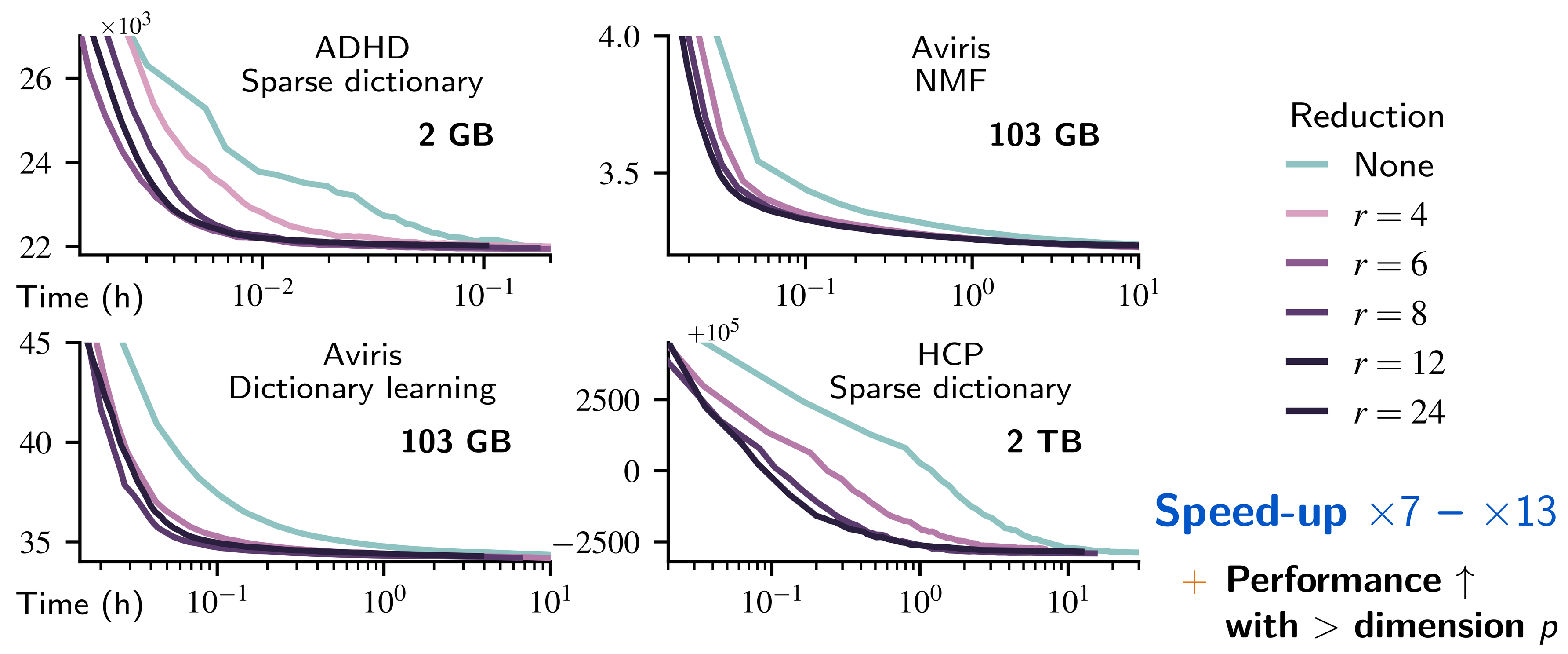
- **Approximate majorization:**

→ Surrogate not tight at $\mathbf{D}_{t-1}$ nor upper-bounding f_t

- **Approximate minimization:**

→ Geometric reduction of suboptimality

Quantitative results: versatile and efficient algorithm

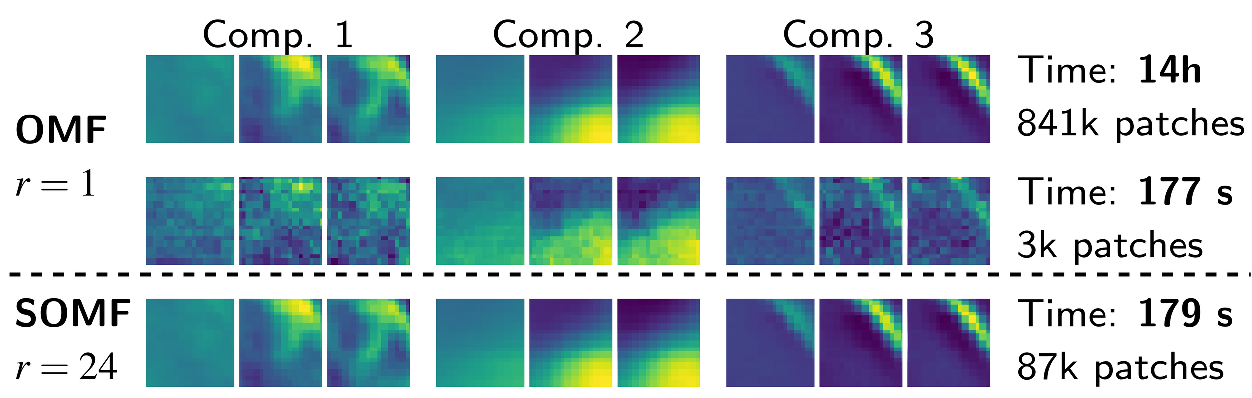


Speed-up ×7 − ×13

+ **Performance ↑ with > dimension p**
- Sparse/dense and/or non-negative dictionary $\mathbf{D}$ and code $\mathbf{A}$

Dataset	ADHD		Aviris (NMF)		Aviris (DL)		HCP	
Algorithm	OMF	SOMF	OMF	SOMF	OMF	SOMF	OMF	SOMF
Conv. time	6 min	**28 s**	2h30	**43 min**	1h16	**11 min**	3h50	**17min**
Speed-up		**11.8**		**3.36**		**6.80**		**13.3**

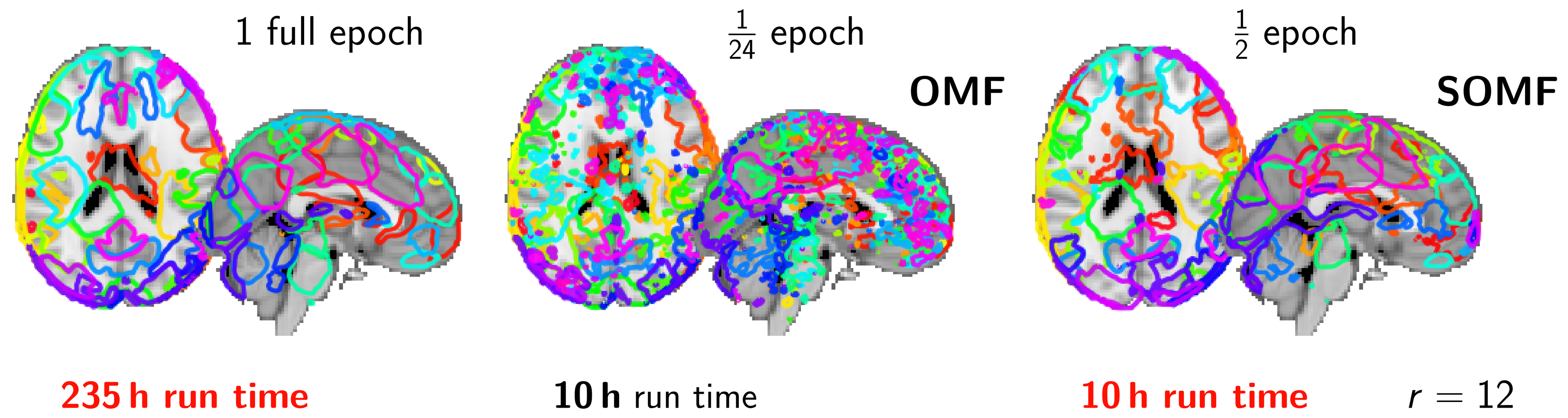
Hyperspectral patches



→ **SOMF atoms are more focal and less noisy**

↑ Given a certain time budget

Functional MRI decomposition



Well defined sparse maps (noiseless contours) are obtained 10× faster

Conclusion

- **Efficient and principled new way to handle high dimensional, numerous data in matrix factorization, using random subsampling**
- Python (fast *Cython* implementation) **package available**: `github.com/arthurmensch/modl`
- Papers and posters available at `http://amensch.fr`. **Preprint to come** in a couple of weeks.
- **Future work.** SAMM for other ERM problem, new applications (text, gradient estimation).