Stochastic Subsampling for Factorizing Huge Matrices

Arthur Mensch, Julien Mairal, Bertrand Thirion, Gaël Varoquaux

▶ **To cite this version:**

Arthur Mensch, Julien Mairal, Bertrand Thirion, Gaël Varoquaux. Stochastic Subsampling for Factorizing Huge Matrices. IEEE Transactions on Signal Processing, 2017. hal-01431618v2

HAL Id: hal-01431618

https://hal.science/hal-01431618v2

Submitted on 26 Jul 2017 (v2), last revised 30 Oct 2017 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Stochastic Subsampling for Factorizing Huge Matrices

Arthur Mensch
Inria Parietal
Saclay, France
arthur.mensch@m4x.org

Julien Mairal
Inria Thoth
Grenoble, France
julien.mairal@inria.fr

Bertrand Thirion
Inria Parietal
Saclay, France
bertrand.thirion@inria.fr

Gaël Varoquaux
Inria Parietal
Saclay, France
gael.varoquaux@inria.fr

July 26, 2017

Abstract

We present a matrix-factorization algorithm that scales to input matrices with both huge number of rows and columns. Learned factors may be sparse or dense and/or non-negative, which makes our algorithm suitable for dictionary learning, sparse component analysis, and non-negative matrix factorization. Our algorithm streams matrix columns while subsampling them to iteratively learn the matrix factors. At each iteration, the row dimension of a new sample is reduced by subsampling, resulting in lower time complexity compared to a simple streaming algorithm. Our method comes with convergence guarantees to reach a stationary point of the matrix-factorization problem. We demonstrate its efficiency on massive functional Magnetic Resonance Imaging data (2 TB), and on patches extracted from hyperspectral images (103 GB). For both problems, which involve different penalties on rows and columns, we obtain significant speed-ups compared to state-of-the-art algorithms.

Index words. Matrix factorization, dictionary learning, non-negative, stochastic optimization, majorization minimization, randomized methods, functional MRI, hyperspectral imaging

1 Introduction

Matrix factorization is a flexible approach to uncover latent factors in low-rank or sparse models. With sparse factors, it is used in dictionary learning, and has proven very effective for denoising and visual feature encoding in signal and computer vision [see e.g., 1]. When the data admit a low-rank structure, matrix factorization has proven very powerful for various tasks such as matrix completion [2, 3], word embedding [4, 5], or network models [6]. It is flexible enough to accommodate a large set of constraints and regularizations, and has gained significant attention in scientific domains where interpretability is a key aspect, such as genetics [7] and neuroscience [8]. In this paper, our goal is to adapt matrix-factorization techniques to huge-dimensional datasets, i.e., with large number of columns n and large number of rows p. Specifically, our work is motivated by the rapid increase in sensor resolution, as in hyperspectral imaging or fMRI, and the challenge that the resulting high-dimensional signals pose to current algorithms.

As a widely-used model, the literature on matrix factorization is very rich and two main classes of formulations have emerged. The first one addresses a convex-optimization problem with a penalty promoting low-rank structures, such as the trace or max norms [2]. This formulation has strong theoretical

The research leading to these results was supported by the ANR (MACARON project, ANR-14-CE23-0003-01 NiConnect project, ANR-11-BINF-0004NiConnect).

guarantees [3], but lacks scalability for huge datasets or sparse factors. For these reasons, our paper is focused on a second type of approach, which relies on nonconvex optimization. Stochastic (or online) optimization methods have been developed in this setting. Unlike classical alternate minimization procedures, they learn matrix decompositions by observing a single matrix column (or row) at each iteration. In other words, they stream data along one matrix dimension. Their cost per iteration is significantly reduced, leading to faster convergence in various practical contexts. More precisely, two approaches have been particularly successful: stochastic gradient descent [9] and stochastic majorization-minimization methods [10, 11]. The former has been widely used for matrix completion [see 12, 13, 14, and references therein], while the latter has been used for dictionary learning with sparse and/or structured regularization [15]. Despite those efforts, stochastic algorithms for dictionary learning are currently unable to deal efficiently with matrices that are large in both dimensions.

We propose a new matrix-factorization algorithm that can handle such matrices. It builds upon the stochastic majorization-minimization framework of [10], which we generalize for our problem. In this framework, the objective function is minimized by iteratively improving an upper-bound surrogate of the function (*majorization* step) and minimizing it to obtain new estimates (*minimization* step). The core idea of our algorithm is to approximate these steps to perform them faster. We carefully introduce and control approximations, so to extend convergence results of [10] when neither the majorization nor the minimization step is performed exactly.

For this purpose, we borrow ideas from *randomized* methods in machine learning and signal processing. Indeed, quite orthogonally to stochastic optimization, efficient approaches to tackle the growth of dataset dimension have exploited random projections [16, 17] or sampling, reducing data dimension while preserving signal content. Large-scale datasets often have an intrinsic dimension which is significantly smaller than their ambient dimension. Good examples are biological datasets [18] and physical acquisitions with an underlying sparse structure enabling compressed sensing [19]. In this context, models can be learned using only random data summaries, also called sketches. For instance, randomized methods [see 20, for a review] are efficient to compute PCA [21], a classic matrix-factorization approach, and to solve constrained or penalized least-square problems [22, 23]. On a theoretical level, recent works on *sketching* [24, 25] have provided bounds on the risk of using random summaries in learning.

Using random projections as a pre-processing step is not appealing in our applicative context since factors learned on reduced data are not interpretable. On the other hand, it is possible to exploit *random sampling* to approximate the steps of online matrix factorization. Factors are learned in the original space whereas the dimension of each iteration is reduced together with the computational cost per iteration.

Contribution. The contribution of this paper is both practical and theoretical. We introduce a new matrix factorization algorithm, called *subsampled online matrix factorization* (SOMF), which is faster than state-of-the-art algorithms by an order of magnitude on large real-world datasets (hyperspectral images, large fMRI data). It leverages random sampling with stochastic optimization to learn sparse and dense factors more efficiently. To prove the convergence of SOMF, we extend the stochastic majorization-minimization framework [10] and make it robust to some time-saving approximations. We then show convergence guarantees for SOMF under reasonable assumptions. Finally, we propose an extensive empirical validation of the subsampling approach.

In a first version of this work [26] presented at the International Conference in Machine Learning (ICML), we proposed an algorithm similar to SOMF, without any theoretical guarantees. The algorithm that we present here has such guarantees, which we express in a more general framework, stochastic majorization-minimization. It is validated for new sparsity settings and a new domain of application. An open-source efficient Python package is provided.

Notations. Matrices are written using bold capital letters and vectors using bold small letters (e.g., $\mathbf{X}$, $\boldsymbol{\alpha}$). We use superscript to specify the column (sample or component) number, and write $\mathbf{X} = [\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}]$. We use subscripts to specify the *iteration* number, as in $\mathbf{x}_t$. The floating bar, as in $\bar{g}_t$, is used to stress that a given value is an average over iterations, or an expectation. The superscript * is used to denote an exact value, when it has to be compared to an inexact value, e.g., to compare $\boldsymbol{\alpha}_t^\star$ (exact) to $\boldsymbol{\alpha}_t$ (approximation).

2 Prior art: matrix factorization with stochastic majorization-minimization

Below, we introduce the matrix-factorization problem and recall a specific stochastic algorithm to solve it observing one column (or a mini-batch) at every iteration. We cast this algorithm in the stochastic majorization-minimization framework [10], which we will use in the convergence analysis.

2.1 Problem statement

In our setting, the goal of matrix factorization is to decompose a matrix $\mathbf{X} \in \mathbb{R}^{p\times n}$ — typically n signals of dimension p — as a product of two smaller matrices:

$$\mathbf{X} \approx \mathbf{D}\mathbf{A} \quad \text{with} \quad \mathbf{D} \in \mathbb{R}^{p\times k} \text{ and } \mathbf{A} \in \mathbb{R}^{k\times n},$$

with potential sparsity or structure requirements on $\mathbf{D}$ and $\mathbf{A}$. In signal processing, sparsity is often enforced on the code $\mathbf{A}$, in a problem called *dictionary learning* [27]. In such a case, the matrix $\mathbf{D}$ is called the "dictionary" and $\mathbf{A}$ the sparse code. We use this terminology throughout the paper.

Learning the factorization is typically performed by minimizing a quadratic data-fitting term, with constraints and/or penalties over the code and the dictionary:

$$\min_{\substack{\mathbf{D}\in\mathcal{C}\\ \mathbf{A}\in\mathbb{R}^{k\times n}}} \quad \sum_{i=1}^{n} \frac{1}{2}\left\|\mathbf{x}^{(i)} - \mathbf{D}\boldsymbol{\alpha}^{(i)}\right\|_2^2 + \lambda\,\Omega(\boldsymbol{\alpha}^{(i)}), \tag{1}$$

where $\mathbf{A} \triangleq [\boldsymbol{\alpha}^{(1)}, \dots, \boldsymbol{\alpha}^{(n)}]$, $\mathcal{C}$ is a column-wise separable convex set of $\mathbb{R}^{p\times k}$ and $\Omega : \mathbb{R}^p \to \mathbb{R}$ is a penalty over the code. Both constraint set and penalty may enforce structure or sparsity, though $\mathcal{C}$ has traditionally been used as a technical requirement to ensure that the penalty on $\mathbf{A}$ does not vanish with $\mathbf{D}$ growing arbitrarily large. Two choices of $\mathcal{C}$ and Ω are of particular interest. The problem of dictionary learning sets $\mathcal{C}$ as the ℓ_2 ball for each atom and Ω to be the ℓ_1 norm. Due to the sparsifying effect of ℓ_1 penalty [28], the dataset admits a *sparse* representation in the dictionary. On the opposite, finding a *sparse set* in which to represent a given dataset, with a goal akin to sparse PCA [29], requires to set as the ℓ_1 ball for each atom and Ω to be the ℓ_2 norm. Our work considers the *elastic-net* constraints and penalties [30], which encompass both special cases. Fixing ν and μ in $[0, 1]$, we denote by $\Omega(\cdot)$ and $\|\cdot\|$ the elastic-net penalty in $\mathbb{R}^p$ and $\mathbb{R}^k$:

$$\Omega(\boldsymbol{\alpha}) \triangleq (1-\nu)\|\boldsymbol{\alpha}\|_1 + \frac{\nu}{2}\|\boldsymbol{\alpha}\|_2^2, \tag{2}$$

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

Following [15], we can also enforce the positivity of $\mathbf{D}$ and/or $\mathbf{A}$ by replacing $\mathbb{R}$ by $\mathbb{R}^+$ in $\mathcal{C}$, and adding positivity constraints on $\mathbf{A}$ in (1), as in non-negative sparse coding [31]. We rewrite (1) as an empirical risk minimization problem depending on the dictionary only. The matrix $\mathbf{D}$ solution of (1) is indeed obtained by minimizing the empirical risk $\bar{f}$

$$\mathbf{D} \in \operatorname*{argmin}_{\mathbf{D}\in\mathcal{C}} \left(\bar{f}(\mathbf{D}) \triangleq \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{D}, \mathbf{x}^{(i)})\right), \tag{3}$$

$$\text{where} \quad f(\mathbf{D}, \mathbf{x}) \triangleq \min_{\boldsymbol{\alpha}\in\mathbb{R}^k} \frac{1}{2}\left\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\right\|_2^2 + \lambda\,\Omega(\boldsymbol{\alpha}),$$

and the matrix $\mathbf{A}$ is obtained by solving the linear regression

$$\min_{\mathbf{A}\in\mathbb{R}^{k\times n}} \sum_{i=1}^{n} \frac{1}{2}\left\|\mathbf{x}^{(i)} - \mathbf{D}\boldsymbol{\alpha}^{(i)}\right\|_2^2 + \lambda\,\Omega(\boldsymbol{\alpha}^{(i)}). \tag{4}$$

The problem (1) is non-convex in the parameters $(\mathbf{D}, \mathbf{A})$, and hence (3) is not convex. However, the problem (1) is convex in both $\mathbf{D}$ and $\mathbf{A}$ when fixing one variable and optimizing with respect to the other. As such, it is naturally solved by alternate minimization over $\mathbf{D}$ and $\mathbf{A}$, which asymptotically provides a stationary point of (3). Yet, $\mathbf{X}$ has typically to be observed hundred of times before obtaining a good dictionary. Alternate minimization is therefore not adapted to datasets with many samples.

Algorithm 1 Online matrix factorization (OMF) [15]

Input: Initial iterate $\mathbf{D}_0$, sample stream $(\mathbf{x}_t)_{t>0}$, number of iterations T.
for t from 1 to T **do**
Draw $\mathbf{x}_t \sim \mathcal{P}$.
Compute $\boldsymbol{\alpha}_t = \operatorname{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{x}_t - \mathbf{D}_{t-1}\boldsymbol{\alpha}\|_2^2 + \lambda\,\Omega(\boldsymbol{\alpha})$.
Update the *parameters* of aggregated surrogate $\bar{g}_t$:

$$\begin{aligned}\bar{\mathbf{C}}_t &= \Big(1-\frac{1}{t}\Big)\bar{\mathbf{C}}_{t-1} + \frac{1}{t}\boldsymbol{\alpha}_t\boldsymbol{\alpha}_t^\top.\\ \bar{\mathbf{B}}_t &= \Big(1-\frac{1}{t}\Big)\bar{\mathbf{B}}_{t-1} + \frac{1}{t}\mathbf{x}_t\boldsymbol{\alpha}_t^\top.\end{aligned} \tag{8}$$

Compute (using block coordinate descent):

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

Output: Final iterate $\mathbf{D}_T$.

2.2 Online matrix factorization

When $\mathbf{X}$ has a large number of columns but a limited number of rows, the stochastic optimization method of [15] outputs a good dictionary much more rapidly than alternate-minimization. In this setting [see 32], learning the dictionary is naturally formalized as an expected risk minimization

$$\min_{\mathbf{D}\in\mathcal{C}} \quad \bar{f}(\mathbf{D}) \triangleq \mathbb{E}_{\mathbf{x}}[f(\mathbf{D},\mathbf{x})], \tag{5}$$

where $\mathbf{x}$ is drawn from the data distribution and forms an i.i.d. stream $(\mathbf{x}_t)_t$. In the finite-sample setting, (5) reduces to (3) when $\mathbf{x}_t$ is drawn uniformly at random from $\{\mathbf{x}^{(i)}, i \in [1,n]\}$. We then write i_t the sample number selected at time t.

The online matrix factorization algorithm proposed in [15] is summarized in Alg. 1. It draws a sample $\mathbf{x}_t$ at each iteration, and uses it to improve the current iterate $\mathbf{D}_{t-1}$. For this, it first computes the code $\boldsymbol{\alpha}_t$ associated to $\mathbf{x}_t$ on the current dictionary:

$$\boldsymbol{\alpha}_t \triangleq \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^k} \frac{1}{2}\|\mathbf{x}_t - \mathbf{D}_{t-1}\boldsymbol{\alpha}\|_2^2 + \lambda\Omega(\boldsymbol{\alpha}). \tag{6}$$

Then, it updates $\mathbf{D}_t$ to make it optimal in reconstructing past samples $(\mathbf{x}_s)_{s\le t}$ from previously computed codes $(\boldsymbol{\alpha}_s)_{s\le t}$:

$$\mathbf{D}_t \in \operatorname*{argmin}_{\mathbf{D}\in\mathcal{C}}\Big(\bar{g}_t(\mathbf{D}) \triangleq \frac{1}{t}\sum_{s=1}^{t}\frac{1}{2}\|\mathbf{x}_s - \mathbf{D}\boldsymbol{\alpha}_s\|_2^2 + \lambda\Omega(\boldsymbol{\alpha}_s)\Big). \tag{7}$$

Importantly, minimizing $\bar{g}_t$ is equivalent to minimizing the quadratic function

$$\mathbf{D} \to \frac{1}{2}\mathrm{Tr}\,(\mathbf{D}^\top\mathbf{D}\bar{\mathbf{C}}_t^\top) - \mathrm{Tr}\,(\mathbf{D}^\top\bar{\mathbf{B}}_t), \tag{9}$$

where $\bar{\mathbf{B}}_t$ and $\bar{\mathbf{C}}_t$ are small matrices that summarize previously seen samples and codes:

$$\bar{\mathbf{B}}_t = \frac{1}{t}\sum_{s=1}^{t}\mathbf{x}_s\boldsymbol{\alpha}_s^\top \qquad \bar{\mathbf{C}}_t = \frac{1}{t}\sum_{s=1}^{t}\boldsymbol{\alpha}_s\boldsymbol{\alpha}_s^\top. \tag{10}$$

As the constraints $\mathcal{C}$ have a separable structure per atom, [15] uses projected block coordinate descent to minimize $\bar{g}_t$. The function gradient writes $\nabla\bar{g}_t(\mathbf{D}) = \mathbf{D}\bar{\mathbf{C}}_t - \bar{\mathbf{B}}_t$, and it is therefore enough to maintain $\bar{\mathbf{B}}_t$ and $\bar{\mathbf{C}}_t$ in memory to solve (7). $\bar{\mathbf{B}}_t$ and $\bar{\mathbf{C}}_t$ are updated online, using the rules (8) (Alg. 1).

The function $\bar{g}_t$ is an upper-bound surrogate of the true current empirical risk, whose definition involves the regression minima computed on current dictionary $\mathbf{D}$:

$$\bar{f}_t(\mathbf{D}) \triangleq \frac{1}{t}\sum_{s=1}^{t}\min_{\boldsymbol{\alpha}\in\mathbb{R}^p}\frac{1}{2}\|\mathbf{x}_s - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\Omega(\boldsymbol{\alpha}) \le \bar{g}_t(\mathbf{D}). \tag{11}$$

Algorithm 2 Stochastic majorization-minimization [SMM 10]

Input: Initial iterate θ_0, weight sequence $(w_t)_{t>0}$, sample stream $(\mathbf{x}_t)_{t>0}$, number of iteration T.
for t from 1 to T **do**
Draw $x_t \sim \mathcal{P}$, get $f_t : \theta \in \Theta \to f(\mathbf{x}_t, \theta)$.
Construct a surrogate of f_t near θ_{t-1}, that meets

$$g_t \geq f_t, \quad g_t(\theta_{t-1}) = f_t(\theta_{t-1}). \tag{12}$$

Update the aggregated surrogate:

$$\bar{g}_t = (1 - w_t)\bar{g}_{t-1} + w_t g_t.$$

Compute

$$\theta_t = \operatorname*{argmin}_{\theta \in \Theta} \bar{g}_t(\theta). \tag{13}$$

Output: Final iterate θ_T.

Using empirical processes theory [33], it is possible to show that minimizing $\bar{f}_t$ at each iteration asymptotically yields a stationary point of the expected risk (5). Unfortunately, minimizing (11) is expensive as it involves the computation of optimal current codes for every previously seen sample at each iteration, which boils down to naive alternate-minimization.

In contrast, $\bar{g}_t$ is much cheaper to minimize than $\bar{f}_t$, using block coordinate descent. It is possible to show that $\bar{g}_t$ converges towards a locally tight upper-bound of the objective $\bar{f}_t$ and that minimizing $\bar{g}_t$ at each iteration also asymptotically yields a stationary point of the expected risk (5). This establishes the correctness of the *online matrix factorization* algorithm (OMF). In practice, the OMF algorithm performs a single pass of block coordinate descent: the minimization step is inexact. This heuristic will be justified by our theoretical contribution in Section 4.

Extensions. For efficiency, it is essential to use mini-batches $\{\mathbf{x}_s,\, s \in \mathcal{T}_t\}$ of size η instead of single samples in the iterations [15]. The surrogate parameters $\bar{\mathbf{B}}_t$, $\bar{\mathbf{C}}_t$ are then updated by the mean value of $\{(\mathbf{x}_s\boldsymbol{\alpha}_s^\top, \boldsymbol{\alpha}_s\boldsymbol{\alpha}_s^\top)\}_{s\in\mathcal{T}_t}$ over the batch. The optimal size of the mini-batches is usually close to k. (8) uses the sequence of weights $(\frac{1}{t})_t$ to update parameters $\bar{\mathbf{B}}_t$ and $\bar{\mathbf{C}}_t$. [15] replaces these weights with a sequence $(w_t)_t$, which can decay more slowly to give more importance to recent samples in $\bar{g}_t$. These weights will prove important in our analysis.

2.3 Stochastic majorization-minimization

Online matrix factorization belongs to a wider category of algorithms introduced in [10] that minimize locally tight upper-bounding surrogates instead of a more complex objective, in order to solve an expected risk minimization problem. Generalizing online matrix factorization, we introduce in Alg. 2 the *stochastic majorization-minimization* (SMM) algorithm, which is at the core of our theoretical contribution.

In online matrix factorization, the true empirical risk functions $\bar{f}_t$ and their *surrogates* $\bar{g}_t$ follow the update rules, with generalized weight $(w_t)_t$ set to $(\frac{1}{t})_t$ in (7) – (11):

$$\bar{f}_t \triangleq (1 - w_t)\bar{f}_{t-1} + w_t f_t, \quad \bar{g}_t \triangleq (1 - w_t)\bar{g}_{t-1} + w_t g_t, \tag{14}$$

where the *pointwise* loss function and its surrogate are

$$\begin{aligned} f_t(\mathbf{D}) &\triangleq \min_{\boldsymbol{\alpha}\in\mathbb{R}^k} \frac{1}{2}\|\mathbf{x}_t - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\Omega(\boldsymbol{\alpha}), \\ g_t(\mathbf{D}) &\triangleq \frac{1}{2}\|\mathbf{x}_t - \mathbf{D}\boldsymbol{\alpha}_t\|_2^2 + \lambda\Omega(\boldsymbol{\alpha}_t). \end{aligned} \tag{15}$$

The function g_t is a majorizing surrogate of f_t: $g_t \geq f_t$, and g_t is tangent to f_t in $\mathbf{D}_{t-1}$, i.e, $g_t(\mathbf{D}_{t-1}) = f_t(\mathbf{D}_{t-1})$ and $\nabla(g_t - f_t)(\mathbf{D}_{t-1}) = 0$. At each step of online matrix factorization:

- The surrogate g_t is computed along with $\boldsymbol{\alpha}_t$, using (6).

- The parameters $\bar{\mathbf{B}}_t, \bar{\mathbf{C}}_t$ are updated following (8). They define the *aggregated* surrogate $\bar{g}_t$ up to a constant.
- The quadratic function $\bar{g}_t$ is minimized efficiently by block coordinate descent, using parameters $\bar{\mathbf{B}}_t$ and $\bar{\mathbf{C}}_t$ to compute its gradient.

The stochastic majorization-minimization framework simply formalizes the three steps above, for a larger variety of loss functions $f_t(\theta) \triangleq f(\theta, \mathbf{x}_t)$, where θ is the parameter we want to learn ($\mathbf{D}$ in the online matrix factorization setting). At iteration t, a surrogate g_t of the loss f_t is computed to update the aggregated surrogate $\bar{g}_t$ following (14). The surrogate functions $(g_t)_t$ should be upper-bounds of loss functions $(f_t)_t$, tight in the current iterate θ_{t-1} (e.g., the dictionary $\mathbf{D}_{t-1}$). This simply means that $f_t(\theta_{t-1}) = g_t(\theta_{t-1})$ and $\nabla(f_t - g_t)(\theta_{t-1}) = 0$. Computing $\bar{g}_t$ can be done if g_t is defined simply, as in OMF where it is linearly parametrized by $(\boldsymbol{\alpha}_t\boldsymbol{\alpha}_t^\top, \mathbf{x}_t\boldsymbol{\alpha}_t^\top)$. $\bar{g}_t$ is then minimized to obtain a new iterate θ_t.

It can be shown following [10] that stochastic majorization-minimization algorithms find asymptotical stationary point of the expected risk $\mathbb{E}_{\mathbf{x}}[f(\theta, \mathbf{x})]$ under mild assumptions recalled in Section 4. SMM admits the same mini-batch and decaying weight extensions (used in Alg. 2) as OMF.

In this work, we extend the SMM framework and allow both majorization and minimization steps to be approximated. As a side contribution, our extension proves that performing a single pass of block coordinate descent to update the dictionary, an important heuristic in [15], is indeed correct. We first introduce the new matrix factorization algorithm at the core of this paper and then present the extended SMM framework.

3 Stochastic subsampling for high dimensional data decomposition

The online algorithm presented in Section 2 is very efficient to factorize matrices that have a large number of columns (i.e., with a large number of samples n), but a reasonable number of rows — the dataset is not very high dimensional. However, it is not designed to deal with very high number of rows: the cost of a single iteration depends linearly on p. On terabyte-scale datasets from fMRI with $p = 2 \cdot 10^5$ features, the original online algorithm requires one week to reach convergence. This is a major motivation for designing new matrix factorization algorithms that scale in *both directions*.

In the large-sample regime $p \gg k$, the underlying dimensionality of columns may be much lower than the actual p: the rows of a single column drawn at random are therefore correlated and redundant. This guides us on how to scale online matrix factorization with regard to the number of rows:

- The online algorithm OMF uses a *single* column of (or mini-batch) of $\mathbf{X}$ at each iteration to enrich the average surrogate and update the *whole* dictionary.
- We go a step beyond and use a *fraction* of a single column of $\mathbf{X}$ to refine a fraction of the dictionary.

More precisely, we draw a column and observe only *some* of its rows at each iteration, to refine these rows of the dictionary, as illustrated in Figure 1. To take into account all features from the dataset, rows are selected at random at each iteration: we call this technique *stochastic subsampling*. Stochastic subsampling reduces the efficiency of the dictionary update *per iteration*, as less information is incorporated in the current iterate $\mathbf{D}_t$. On the other hand, with a correct design, the cost of a single iteration can be considerably reduced, as it grows with the number of observed features. Section 5 shows that the proposed algorithm is an order of magnitude faster than the original OMF on large and redundant datasets.

First, we formalize the idea of working with a fraction of the p rows at a single iteration. We adapt the online matrix factorization algorithm, to reduce the iteration cost by a factor close to the ratio of selected rows. This defines a new online algorithm, called *subsampled online matrix factorization* (SOMF). At each iteration, it uses q rows of the column $\mathbf{x}_t$ to update the sequence of iterates $(\mathbf{D}_t)_t$. As in Section 2, we introduce a more general algorithm, *stochastic approximate majorization-minimization* (SAMM), of which SOMF is an instance. It extends the stochastic majorization-minimization framework, with similar theoretical guarantees but potentially faster convergence.

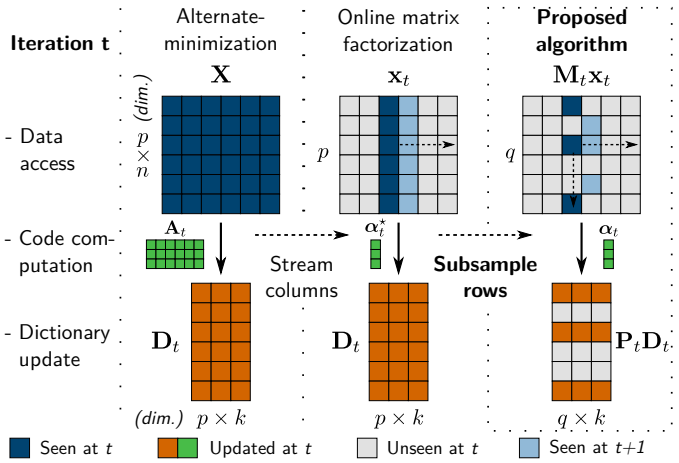

Figure 1: *Stochastic subsampling* further improves online matrix factorization to handle datasets with large number of columns and rows. $\mathbf{X}$ is the input $p \times n$ matrix, $\mathbf{D}_t$ and $\mathbf{A}_t$ are respectively the dictionary and code at time t.

3.1 Subsampled online matrix factorization

Formally, as in online matrix factorization, we consider a sample stream $(\mathbf{x}_t)_t$ in $\mathbb{R}^p$ that cycles onto a finite sample set $\{\mathbf{x}^{(i)}, i \in [1, n]\}$, and minimize the empirical risk (3).[1]

3.1.1 Stochastic subsampling and algorithm outline

We want to reduce the time complexity of a single iteration. In the original algorithm, the complexity depends linearly on the sample dimension p in three aspects:

- $\mathbf{x}_t \in \mathbb{R}^p$ is used to compute the code $\boldsymbol{\alpha}_t$,
- it is used to update the surrogate parameters $\bar{\mathbf{B}}_t \in \mathbb{R}^{p \times k}$,
- $\mathbf{D}_t \in \mathbb{R}^{p \times k}$ is fully updated at each iteration.

Our algorithm reduces the dimensionality of these steps at each iteration, such that p becomes $q = \frac{p}{r}$ in the time complexity analysis, where $r > 1$ is a *reduction factor*. Formally, we randomly draw, at iteration t, a mask $\mathbf{M}_t$ that "selects" a random subset of $\mathbf{x}_t$. We use it to drop a part of the features of $\mathbf{x}_t$ and to "freeze" these features in dictionary $\mathbf{D}$ at iteration t.

It is convenient to consider $\mathbf{M}_t$ as a $\mathbb{R}^{p \times p}$ random diagonal matrix, such that each coefficient is a Bernouilli variable with parameter $\frac{1}{r}$, normalized to be 1 in expectation. $\forall j \in [0, p-1]$,

$$\mathbb{P}\big[\mathbf{M}_t[j,j] = r\big] = \frac{1}{r}, \quad \mathbb{P}\big[\mathbf{M}_t[j,j] = 0\big] = 1 - \frac{1}{r}. \tag{16}$$

Thus, r describes the average proportion of observed features and $\mathbf{M}_t\mathbf{x}_t$ is a non-biased, low-dimensional estimator of $\mathbf{x}_t$:

$$\mathbb{E}\big[\|\mathbf{M}_t\mathbf{x}_t\|_0\big] = \frac{p}{r} = q \qquad \mathbb{E}\big[\mathbf{M}_t\mathbf{x}_t\big] = \mathbf{x}_t. \tag{17}$$

with $\|\cdot\|_0$ counting the number of non-zero coefficients. We define the pair of orthogonal projectors $\mathbf{P}_t \in \mathbb{R}^{q \times p}$ and $\mathbf{P}_t^\perp \in \mathbb{R}^{(p-q) \times p}$ that project $\mathbb{R}^p$ onto $\mathrm{Im}(\mathbf{M}_t)$ and $\mathrm{Ker}(\mathbf{M}_t)$. In other words, $\mathbf{P}_t\mathbf{Y}$ and $\mathbf{P}_t^\perp\mathbf{Y}$ are the submatrices of $\mathbf{Y} \in \mathbb{R}^{p \times y}$ with rows respectively selected and not selected by $\mathbf{M}_t$. In algorithms, $\mathbf{P}_t\mathbf{Y} \leftarrow \mathbf{Z} \in \mathbb{R}^{q \times n}$ assigns the rows of $\mathbf{Z}$ to the rows of $\mathbf{Y}$ selected by $\mathbf{P}_t$, by an abuse of notation.

In brief, subsampled online matrix factorization, defined in Alg. 3, follows the outer loop of online matrix factorization, with the following major modifications at iteration t:

[1] Note that we solve the fully observed problem despite the use of subsampled data, unlike other recent work on low-rank factorization [34].

Algorithm 3 Subsampled online matrix factorization (SOMF)

Input: Initial iterate $\mathbf{D}_0$, weight sequences $(w_t)_{t>0}$, $(\gamma_c)_{c>0}$, sample set $\{\mathbf{x}^{(i)}\}_{i>0}$, number of iterations T.

for t from 1 to T **do**

Draw $\mathbf{x}_t = \mathbf{x}^{(i)}$ at random and $\mathbf{M}_t$ following (16).

Update the regression parameters for sample i:

$$\begin{aligned} c^{(i)} &\leftarrow c^{(i)} + 1, & \gamma &\leftarrow \gamma_{c^{(i)}}. \\ \boldsymbol{\beta}_t^{(i)} &\leftarrow (1-\gamma)\mathbf{G}_{t-1}^{(i)} + \gamma \mathbf{D}_{t-1}^\top \mathbf{M}_t \mathbf{x}^{(i)}, & \boldsymbol{\beta}_t &\leftarrow \boldsymbol{\beta}_t^{(i)}. \\ \mathbf{G}_t^{(i)} &\leftarrow (1-\gamma)\mathbf{G}_{t-1}^{(i)} + \gamma \mathbf{D}_{t-1}^\top \mathbf{M}_t \mathbf{D}_{t-1}, & \mathbf{G}_t &\leftarrow \bar{\mathbf{G}}_t^{(i)}. \end{aligned}$$

Compute the approximate code for $\mathbf{x}_t$:

$$\boldsymbol{\alpha}_t \leftarrow \operatorname*{argmin}_{\boldsymbol{\alpha} \in \mathbb{R}^k} \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{G}_t \boldsymbol{\alpha} - \boldsymbol{\alpha}^\top \boldsymbol{\beta}_t + \lambda\, \Omega(\boldsymbol{\alpha}). \tag{18}$$

Update the parameters of the aggregated surrogate $\bar{g}_t$:

$$\begin{aligned} \bar{\mathbf{C}}_t &\leftarrow (1-w_t)\bar{\mathbf{C}}_{t-1} + w_t \boldsymbol{\alpha}_t \boldsymbol{\alpha}_t^\top. \\ \mathbf{P}_t \bar{\mathbf{B}}_t &\leftarrow (1-w_t)\mathbf{P}_t \bar{\mathbf{B}}_{t-1} + w_t \mathbf{P}_t \mathbf{x}_t \boldsymbol{\alpha}_t^\top. \end{aligned} \tag{19}$$

Compute simultaneously (using Alg. 4 for 1st line):

$$\begin{aligned} \mathbf{P}_t \mathbf{D}_t &\leftarrow \operatorname*{argmin}_{\mathbf{D}^r \in \mathcal{C}^r} \frac{1}{2}\mathrm{Tr}\left(\mathbf{D}^{r\top} \mathbf{D}^r \bar{\mathbf{C}}_t\right) - \mathrm{Tr}\left(\mathbf{D}^{r\top} \mathbf{P}_t \bar{\mathbf{B}}_t\right). \\ \mathbf{P}_t^\perp \bar{\mathbf{B}}_t &\leftarrow (1-w_t)\mathbf{P}_t^\perp \bar{\mathbf{B}}_{t-1} + w_t \mathbf{P}_t^\perp \mathbf{x}_t \boldsymbol{\alpha}_t^\top. \end{aligned} \tag{20}$$

Output: Final iterate $\mathbf{D}_T$.

- it uses $\mathbf{M}_t\mathbf{x}_t$ and low-size statistics instead of $\mathbf{x}_t$ to estimate the code $\boldsymbol{\alpha}_t$ and the surrogate g_t,
- it updates a subset of the dictionary $\mathbf{P}_t\mathbf{D}_{t-1}$ to reduce the surrogate value $\bar{g}_t(\mathbf{D})$. Relevant parameters of $\bar{g}_t$ are computed using $\mathbf{P}_t\mathbf{x}_t$ and $\boldsymbol{\alpha}_t$ only.

We now present SOMF in details. For comparison purpose, we write all variables that would be computed following the OMF rules at iteration t with a * superscript. For simplicity, in Alg. 3 and in the following paragraphs, we assume that we use one sample per iteration —in practice, we use mini-batches of size η. The next derivations are transposable when a batch I_t is drawn at iteration t instead of a single sample i_t.

3.1.2 Code computation

In the OMF algorithm presented in Section 2, $\boldsymbol{\alpha}_t^\star$ is obtained by solving (6), namely

$$\boldsymbol{\alpha}_t^\star \in \operatorname*{argmin}_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{G}_t^\star \boldsymbol{\alpha} - \boldsymbol{\alpha}^\top \boldsymbol{\beta}_t^\star + \lambda \Omega(\boldsymbol{\alpha}), \tag{21}$$

where $\mathbf{G}_t^\star = \mathbf{D}_{t-1}^\top \mathbf{D}_{t-1}$ and $\boldsymbol{\beta}_t^\star = \mathbf{D}_{t-1}^\top \mathbf{x}_t$. For large p, the computation of $\mathbf{G}_t^\star$ and $\boldsymbol{\beta}_t^\star$ dominates the complexity of the regression step, which depends almost linearly on p. To reduce this complexity, we use *estimators* for $\mathbf{G}_t^\star$ and $\boldsymbol{\beta}_t^\star$, computed at a cost proportional to the reduced dimension q. We propose three kinds of estimators with different properties.

Masked loss The most simple *unbiased* estimation of $\mathbf{G}_t^\star$ and $\boldsymbol{\beta}_t^\star$ whose computation cost depends on q is obtained by subsampling matrix products with $\mathbf{M}_t$:

$$\begin{aligned} \mathbf{G}_t &= \mathbf{D}_{t-1}^\top \mathbf{M}_t \mathbf{D}_{t-1} \\ \boldsymbol{\beta}_t &= \mathbf{D}_{t-1}^\top \mathbf{M}_t \mathbf{x}_t. \end{aligned} \tag{a}$$

This is the strategy proposed in [26]. We use $\mathbf{G}_t$ and $\boldsymbol{\beta}_t$ in (18), which amounts to minimize the *masked loss*

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^k} \frac{1}{2}\|\mathbf{M}_t(\mathbf{x}^t - \mathbf{D}_{t-1}^\top\boldsymbol{\alpha})\|_2^2 + \lambda\Omega(\boldsymbol{\alpha}). \tag{22}$$

$\mathbf{G}_t$ and $\boldsymbol{\beta}_t$ are computed in a number of operations proportional to q, which brings a speed-up factor of almost r in the code computation for large p. On large data, using estimators (a) instead of exact $\mathbf{G}_t^\star$ and $\boldsymbol{\beta}_t^\star$ proves very efficient during the first epochs (cycles over the columns).[2] However, due to the masking, $\mathbf{G}_t$ and $\boldsymbol{\beta}_t$ are not *consistent* estimators: they do not converge to $\mathbf{G}_t^\star$ and $\boldsymbol{\beta}_t^\star$ for large t, which breaks theoretical guarantees on the algorithm output. Empirical results in Section 5.5 show that the sequence of iterates approaches a critical point of the risk (3), but may then oscillate around it.

Averaging over epochs At iteration t, the sample $\mathbf{x}_t$ is drawn from a finite set of samples $\{\mathbf{x}^{(i)}\}_i$. This allows to average estimators over previously seen samples and address the non-consistency issue of (a). Namely, we keep in memory $2n$ estimators, written $(\mathbf{G}_t^{(i)}, \boldsymbol{\beta}_t^{(i)})_{1\le i\le n}$. We observe the sample $i = i_t$ at iteration t and use it to update the i-th estimators $\bar{\mathbf{G}}_t^{(i)}$, $\bar{\boldsymbol{\beta}}_t^{(i)}$ following

$$\begin{aligned}\boldsymbol{\beta}_t^{(i)} &= (1-\gamma)\mathbf{G}_{t-1}^{(i)} + \gamma\mathbf{D}_{t-1}^\top\mathbf{M}_t\mathbf{x}^{(i)}\\ \mathbf{G}_t^{(i)} &= (1-\gamma)\mathbf{G}_{t-1}^{(i)} + \gamma\mathbf{D}_{t-1}^\top\mathbf{M}_t\mathbf{D}_t^{(i)},\end{aligned} \tag{23}$$

where γ is a weight factor determined by the number of time the one sample i has been previously observed at time t. Precisely, given $(\gamma_c)_c$ a decreasing sequence of weights,

$$\gamma = \gamma_{c_t^{(i)}} \quad \text{where} \quad c_t^{(i)} = \left|\left\{s \le t, \mathbf{x}_s = \mathbf{x}^{(i)}\right\}\right|.$$

All others estimators $\{\mathbf{G}_t^{(j)}, \boldsymbol{\beta}_t^{(j)}\}_{j\neq i}$ are left unchanged from iteration $t-1$. The set $\{\mathbf{G}_t^{(i)}, \boldsymbol{\beta}_t^{(i)}\}_{1\le i\le n}$ is used to define the *averaged* estimators

$$\begin{aligned}\mathbf{G}_t \triangleq \mathbf{G}_t^{(i)} &= \sum_{s\le t, \mathbf{x}_s=\mathbf{x}^{(i)}} \gamma_{s,t}^{(i)}\mathbf{D}_{s-1}^\top\mathbf{M}_s\mathbf{D}_{s-1}\\ \boldsymbol{\beta}_t \triangleq \boldsymbol{\beta}_t^{(i)} &= \sum_{s\le t, \mathbf{x}_s=\mathbf{x}^{(i)}} \gamma_{s,t}^{(i)}\mathbf{D}_{s-1}^\top\mathbf{M}_s\mathbf{x}^{(i)},\end{aligned} \tag{b}$$

where $\gamma_{s,t}^{(i)} = \gamma_{c_t^{(i)}}\prod_{s<t,\mathbf{x}_s=\mathbf{x}^{(i)}}(1-\gamma_{c_s^{(i)}})$. Using $\boldsymbol{\beta}_t$ and $\mathbf{G}_t$ in (18), $\boldsymbol{\alpha}_t$ minimizes the masked loss averaged over the previous iterations where sample i appeared:

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^k} \sum_{\substack{s\le t\\ \mathbf{x}_s=\mathbf{x}^{(i)}}} \frac{\gamma_{s,t}^{(i)}}{2}\|\mathbf{M}_s(\mathbf{x}^{(i)} - \mathbf{D}_{s-1}^\top\boldsymbol{\alpha})\|_2^2 + \lambda\Omega(\boldsymbol{\alpha}). \tag{24}$$

The sequences $(\mathbf{G}_t)_t$ and $(\boldsymbol{\beta}_t)_t$ are *consistent* estimations of $(\mathbf{G}_t^\star)_t$ and $(\boldsymbol{\beta}_t^\star)_t$ — consistency arises from the fact that a single sample $\mathbf{x}^{(i)}$ is observed with different masks along iterations. Solving (24) is made closer and closer to solving (21), to ensure the correctness of the algorithm (see Section 4). Yet, computing the estimators (b) is no more costly than computing (a) and still permits to speed up a single iteration close to r times. In the mini-batch setting, for every $i \in I_t$, we use the estimators $\mathbf{G}_t^{(i)}$ and $\boldsymbol{\beta}_t^{(i)}$ to compute $\boldsymbol{\alpha}_t^{(i)}$. This method has a memory cost of $\mathcal{O}(n\,k^2)$. This is reasonable compared to the dataset size[3] if $p \gg k^2$.

Exact Gram computation To reduce the memory usage, another strategy is to use the *true* Gram matrix $\mathbf{G}_t$ and the estimator $\boldsymbol{\beta}_t$ from (b):

$$\begin{aligned}\mathbf{G}_t \triangleq \mathbf{G}_t^\star &= \mathbf{D}_{t-1}^\top\mathbf{D}_{t-1}\\ \boldsymbol{\beta}_t \triangleq &\sum_{s\le t, \mathbf{x}_s=\mathbf{x}^{(i)}} \gamma_{s,t}^{(i)}\mathbf{D}_{s-1}^\top\mathbf{M}_s\mathbf{x}^{(i)}\end{aligned} \tag{c}$$

[2] Estimators (a) are also available in the infinite sample setting, when minimizing expected risk (5) from a i.i.d sample stream $(\mathbf{x}_t)_t$.

[3] It is also possible to efficiently swap the estimators $(\mathbf{G}_t^{(i)})_i$ on disk, as they are only accessed for $i = i_t$ at iteration t.

Table 1: Comparison of estimators used for code computation

Est.	$\boldsymbol{\beta}_t$	$\mathbf{G}_t$	Convergence	Extra mem. cost	1st epoch perform.
(a)	Masked	Masked			✓
(b)	Averaged	Averaged	✓	$n\,k^2$	✓
(c)	Averaged	Exact	✓	$n\,k$	

As previously, the consistency of $(\boldsymbol{\beta}_t)_t$ ensures that (5) is correctly solved despite the approximation in $(\boldsymbol{\alpha}_t)_t$ computation. With the partial dictionary update step we propose, it is possible to maintain $\mathbf{G}_t$ at a cost proportional to q. The time complexity of the coding step is thus similarly reduced when replacing (b) or (c) estimators in (21), but the latter option has a memory usage in $\mathcal{O}(n\,k)$. Although estimators (c) are slightly less performant in the first epochs, they are a good compromise between resource usage and convergence. We summarize the characteristics of the three estimators (a)–(c) in Table 1, anticipating their empirical comparison in Section 5.

Surrogate computation. The computation of $\boldsymbol{\alpha}_t$ using one of the estimators above defines a surrogate $g_t(\mathbf{D}) \triangleq \frac{1}{2}\|\mathbf{x}_t - \mathbf{D}\boldsymbol{\alpha}_t\|_2^2 + \lambda\Omega(\boldsymbol{\alpha})$, which we use to update the aggregated surrogate $\bar{g}_t \triangleq (1-w_t)\bar{g}_{t-1} + w_t g_t$, as in online matrix factorization. We follow (8) (with weights $(w_t)_t$) to update the matrices $\bar{\mathbf{B}}_t$ and $\bar{\mathbf{C}}_t$, which define $\bar{g}_t$ up to constant factors. The update of $\bar{\mathbf{B}}_t$ requires a number of operations proportional to p. Fortunately, we will see in the next paragraph that it is possible to update $\mathbf{P}_t\bar{\mathbf{B}}_t$ in the main thread with a number of operation proportional to q and to complete the update of $\mathbf{P}_t^\perp\bar{\mathbf{B}}_t$ in parallel with the dictionary update step.

Weight sequences. Specific $(w_t)_t$ and $(\gamma_c)_c$ in Alg. 3 are required. We provide then in Assumption **(B)** of the analysis: $w_t = \frac{1}{t^u}$ and $\gamma_c = \frac{1}{c^v}$, where $u \in (\frac{11}{12}, 1)$ and $v \in \left(\frac{3}{4}, 3u-2\right)$ to ensure convergence. Weights have little impact on convergence speed in practice.

3.1.3 Dictionary update

In the original online algorithm, the whole dictionnary $\mathbf{D}_{t-1}$ is updated at iteration t. To reduce the time complexity of this step, we add a "freezing" constraint to the minimization (7) of $\bar{g}_t$. Every row r of $\mathbf{D}$ that corresponds to an *unseen* row r at iteration r (such that $\mathbf{M}_t[r,r] = 0$) remains unchanged. This casts the problem (7) into a lower dimensional space. Formally, the freezing operation comes out as a additional constraint in (7):

$$\mathbf{D}_t = \underset{\substack{\mathbf{D}\in\mathcal{C}\\ \mathbf{P}_t^\perp\mathbf{D}=\mathbf{P}_t^\perp\mathbf{D}_{t-1}}}{\operatorname{argmin}} \frac{1}{2}\operatorname{Tr}\left(\mathbf{D}^\top\mathbf{D}\bar{\mathbf{C}}_t\right) - \operatorname{Tr}\left(\mathbf{D}^\top\bar{\mathbf{B}}_t\right). \tag{25}$$

The constraints are separable into two blocks of rows. Recalling the notations of (2), for each atom $\mathbf{d}^{(j)}$, the rules $\|\mathbf{d}^{(j)}\| \leq 1$ and $\mathbf{P}_t^\perp\mathbf{d}^{(j)} = \mathbf{P}_t^\perp\mathbf{d}_{t-1}^{(j)}$ can indeed be rewritten

$$\begin{cases} \|\mathbf{P}_t\mathbf{d}^{(j)}\| & \leq & 1 - \|\mathbf{d}_{t-1}^{(j)}\| + \|\mathbf{P}_t\mathbf{d}_{t-1}^{(j)}\| \triangleq r_t^{(j)} \\ \mathbf{P}_t^\perp\mathbf{d}^{(j)} & = & \mathbf{P}_t^\perp\mathbf{d}_{t-1}^{(j)}. \end{cases} \tag{26}$$

Solving (25) is therefore equivalent to solving the following problem in $\mathbb{R}^{q\times k}$, with $\mathbf{B}_t^r \triangleq \mathbf{P}_t\mathbf{B}_t$,

$$\mathbf{D}^r \in \underset{\mathbf{D}^r\in\mathcal{C}^r}{\operatorname{argmin}} \frac{1}{2}\operatorname{Tr}\left(\mathbf{D}^{r\top}\mathbf{D}^r\bar{\mathbf{C}}_t\right) - \operatorname{Tr}\left(\mathbf{D}^{r\top}\bar{\mathbf{B}}_t^r\right) \tag{27}$$
$$\text{where } \mathcal{C}^r = \{\mathbf{D}^r \in \mathbb{R}^{q\times k} / \forall j \in [0, k-1], \|\mathbf{d}^{r(j)}\| \leq r_t^{(j)}\}.$$

The rows of $\mathbf{D}_t$ selected by $\mathbf{P}_t$ are then replaced with $\mathbf{D}^r$, while the other rows of $\mathbf{D}_t$ are unchanged from iteration $t-1$. Formally, $\mathbf{P}_t\mathbf{D}_t = \mathbf{D}^r$ and $\mathbf{P}_t^\perp\mathbf{D}_t = \mathbf{P}_t^\perp\mathbf{D}_{t-1}$. We solve (27) by a projected block coordinate descent (BCD) similar to the one used in the original algorithm, but performed in a subspace

Algorithm 4 Partial dictionary update

Input: Dictionary $\mathbf{D}_{t-1}$, projector $\mathbf{P}_t$, statistics $\bar{\mathbf{C}}_t$, $\bar{\mathbf{B}}_t$, norms $(n_{t-1}^{(j)})_{0\leq j<k}$, Gram matrix $\mathbf{G}_t$ (optional).
$\mathbf{D}_t \leftarrow \mathbf{D}_{t-1}$, $\quad \mathbf{G}_t \leftarrow \mathbf{G}_t - \mathbf{D}_{t-1}^\top \mathbf{P}_t \mathbf{D}_{t-1}$.
for $j \in$ permutation$([1,k])$ **do**
  $r_t^{(j)} \leftarrow n_{t-1}^{(j)} + \|\mathbf{P}_t \mathbf{d}_{t-1}^{(j)}\|$.
  $\mathbf{u} \leftarrow \mathbf{P}_t \mathbf{d}_{t-1}^{(j)} + \frac{1}{\bar{\mathbf{C}}_t[j,j]}(\mathbf{P}_t \bar{\mathbf{b}}_t^{(j)} - \mathbf{P}_t \mathbf{D}_t \bar{\mathbf{c}}_t^{(j)})$. ▷ in $\mathbb{R}^q$
  $\mathbf{P}_t \mathbf{d}_t^{(j)} \leftarrow$ `enet_projection`$(\mathbf{u}, r_t^{(j)})$. ▷ in $\mathbb{R}^q$
  $n_t^{(j)} \leftarrow r_t^{(j)} - \|\mathbf{P}_t \mathbf{d}_t^{(j)}\|$.
$\mathbf{G}_{t+1} \leftarrow \mathbf{G}_t + \mathbf{D}_t^\top \mathbf{P}_t \mathbf{D}_t$.
Output: Dictionary $\mathbf{D}_t$, norms $(n_t^{(j)})_j$, Gram matrix $\mathbf{G}_{t+1}$.

of size q. We compute each column j of the gradient that we use in the block coordinate descent loop with $q \times k$ operations, as it writes $\mathbf{D}^r \bar{\mathbf{c}}_t^{(j)} - \bar{\mathbf{b}}_t^{r(j)} \in \mathbb{R}^q$, where $\bar{\mathbf{c}}_t^{(j)}$ and $\bar{\mathbf{b}}_t^{r(j)}$ are the j-th columns of $\bar{\mathbf{C}}_t$ and $\bar{\mathbf{B}}_t^r$. Each reduced atom $\mathbf{d}^{r(j)}$ is projected onto the elastic-net ball of radius $r_t^{(j)}$, at an average cost in $\mathcal{O}(q)$ following [15]. This makes the complexity of a single-column update proportional to q. Performing the projection requires to keep in memory the values $\{n_t^{(j)} \triangleq 1 - \|\mathbf{d}_t^{(j)}\|\}_j$, which can be updated online at a negligible cost.

We provide the reduced dictionary update step in Alg. 4, where we use the function `enet_projection`$(\mathbf{u}, r)$ that performs the orthogonal projection of $\mathbf{u} \in \mathbb{R}^q$ onto the elastic-net ball of radius r. As in the original algorithm, we perform a single pass over columns to solve (27). Dictionary update is now performed with a number of operations proportional to q, instead of p in the original algorithm. Thanks to the random nature of $(\mathbf{M}_t)_t$, updating $\mathbf{D}_{t-1}$ into $\mathbf{D}_t$ reduces $\bar{g}_t$ enough to ensure convergence.

Gram matrix computation. Performing partial updates of $\mathbf{D}_t$ makes it possible to maintain the full Gram matrix $\mathbf{G}_t = \mathbf{G}_t^\star$ with a cost in $\mathcal{O}(q\,k^2)$ per iteration, as mentioned in 3.1.2. It is indeed enough to compute the reduced Gram matrix $\mathbf{D}^\top \mathbf{P}_t \mathbf{D}$ before and after the dictionary update:

$$\mathbf{G}_{t+1} = \mathbf{D}_t^\top \mathbf{D}_t = \mathbf{G}_t - \mathbf{D}_{t-1}^\top \mathbf{P}_t \mathbf{D}_{t-1}^\top + \mathbf{D}_t^\top \mathbf{P}_t \mathbf{D}_t^\top. \tag{28}$$

Parallel surrogate computation. Performing block coordinate descent on $\bar{g}_t^r$ requires to access $\bar{\mathbf{B}}_t^r = \mathbf{P}_t \bar{\mathbf{B}}_t$ only. Assuming we may use use more than two threads, this allows to parallelize the dictionary update step with the update of $\mathbf{P}_t^\perp \bar{\mathbf{B}}_t$. In the main thread, we compute $\mathbf{P}_t \bar{\mathbf{B}}_t$ following

$$\mathbf{P}_t \bar{\mathbf{B}}_t \leftarrow (1-w_t)\bar{\mathbf{P}}_t \mathbf{B}_{t-1} + w_t \mathbf{P}_t \mathbf{x}_t \boldsymbol{\alpha}_t^\top. \tag{19 – Alg. 3}$$

which has a cost proportional to q. Then, we update in parallel the dictionary and the rows of $\bar{\mathbf{B}}_t$ that are not selected by $\mathbf{M}_t$:

$$\mathbf{P}_t^\perp \bar{\mathbf{B}}_t \leftarrow (1-w_t)\mathbf{P}_t^\perp \bar{\mathbf{B}}_{t-1} + w_t \mathbf{P}_t^\perp \mathbf{x}_t \boldsymbol{\alpha}_t^\top. \tag{20 – Alg. 3}$$

This update requires $k(p-q)\eta$ operations (one matrix-matrix product) for a mini-batch of size η. In contrast, with appropriate implementation, the dictionary update step requires $4\,k\,q^2$ to $6\,k\,q^2$ operations, among which $2\,k\,q^2$ come from slower matrix-vector products. Assuming $k \sim \eta$, updating $\bar{\mathbf{B}}_t$ is faster than updating the dictionary up to $r \sim 10$, and performing (20) on a second thread is seamless in term of wall-clock time. More threads may be used for larger reduction or batch size.

3.1.4 Subsampling and time complexity

Subsampling may be used in only *some* of the steps of Alg. 3, with the other steps following Alg. 1. Whether to use subsampling or not in each step depends on the trade-off between the computational speed-up it brings and the approximations it makes. It is useful to understand how complexity of OMF evolves with p. We write s the average number of non-zero coefficients in $(\boldsymbol{\alpha}_t)_t$ ($s = k$ when $\Omega = \|\cdot\|_2^2$). OMF complexity has three terms:

(i) $\mathcal{O}(p\,k^2)$: computation of the Gram matrix $\mathbf{G}_t$, update of the dictionary $\mathbf{D}_t$ with block coordinate descent,

(ii) $\mathcal{O}(p\,k\,\eta)$: computation of $\boldsymbol{\beta}_t = \mathbf{D}_{t-1}^\top \mathbf{x}_t$ and of $\bar{\mathbf{B}}_t$ using $\mathbf{x}_t \boldsymbol{\alpha}_t^\top$,

(iii) $\mathcal{O}(k\,s^2\,\eta)$: computation of $\boldsymbol{\alpha}_t$ using $\mathbf{G}_t$ and $\boldsymbol{\beta}_t$, using matrix inversion or elastic-net regression.

Using subsampling turns p into $q = \frac{p}{r}$ in the expressions above. It improves single iteration time when the cost of regression $\mathcal{O}(k\,s^2\,\eta)$ is dominated by another term. This happens whenever $\frac{p}{r} > s^2$, where r is the reduction factor used in the algorithm. Subsampling can bring performance improvement up to $r \sim \frac{p}{s^2}$. It can be introduced in either computations from (i) or (ii), or both. When using small batch size, i.e., when $\eta < k$, computations from (i) dominates complexity, and subsampling should be first introduced in dictionary update (i), and for code computation (ii) beyond a certain reduction ratio. On the other hand, with large batch size $\eta > k$, subsampling should be first introduced in code computation, then in the dictionary update step. The reasoning above ignore potentially large constants. The best trade-offs in using subsampling must be empirically determined, which we do in Section 5.

3.2 Stochastic approximate majorization-minimization

The SOMF algorithm can be understood within the stochastic majorization-minimization framework. The modifications that we propose are indeed perturbations to the first and third steps of the SMM presented in Algorithm 2:

- The code is computed approximately: the surrogate is only an *approximate* majorizing surrogate of f_t near $\mathbf{D}_{t-1}$.
- The surrogate objective is only *reduced* and not minimized, due to the added constraint and the fact that we perform only one pass of block coordinate descent.

We propose a new *stochastic approximate majorization-minimization* (SAMM) framework handling these perturbations:

- A majorization step (12 – Alg. 2), computes an *approximate surrogate* of f_t near θ_{t-1}: $g_t \approx g_t^\star$, where g_t is a true upper-bounding surrogate of $\bar{f}_t$.
- A minimization step (13 – Alg. 2), finds θ_t by reducing *enough* the objective $\bar{g}_t$: $\theta_t \approx \theta_t^\star \triangleq \operatorname{argmin}_{\theta\in\Theta} \bar{g}_t(\theta)$, which implies $\bar{g}_t(\theta_t) \lesssim \bar{g}_t(\theta_t^\star)$.

The SAMM framework is general, in the sense that approximations are not specified. The next section provides a theoretical analysis of the approximation of SAMM and establishes how SOMF is an instance of SAMM. It concludes by establishing Proposition 1, which provides convergence guarantees for SOMF, under the same assumptions made for OMF in [15].

4 Convergence analysis

We establish the convergence of SOMF under reasonable assumptions. For the sake of clarity, we first state our principal result (Proposition 1), that guarantees SOMF convergence. It is a corollary of a more general result on SAMM algorithms. To present this broader result, we recall the theoretical guarantees of the stochastic majorization-minimization algorithm [10] (Proposition 2); then, we show how the algorithm can withstand pertubations (Proposition 3). Proofs are reported in Appendix A. SAMM convergence is proven before establishing SOMF convergence as a corollary of this broader result.

4.1 Convergence of SOMF

Similar to [15, 34], we show that the sequence of iterates $(\mathbf{D}_t)_t$ asymptotically reaches a critical point of the empirical risk (3). We introduce the same hypothesis on the code covariance estimation $\bar{\mathbf{C}}_t$ as in [15] and a similar one on $\mathbf{G}_t$ — they ensure strong convexity of the surrogate and boundedness of $(\boldsymbol{\alpha}_t)_t$. They do not cause any loss of generality as they are met in practice after a few iterations, if r is chosen reasonably low, so that $q > k$. The following hypothesis can also be guaranteed by adding small ℓ_2 regularizations to $\bar{f}$.

(A) There exists $\rho > 0$ such that for all $t > 0$, $\bar{\mathbf{C}}_t, \mathbf{G}_t \succ \rho \mathbf{I}$.

We further assume, that the weights $(w_t)_t$ and $(\gamma_c)_c$ decay at specific rates. We specify simple weight sequences, but the proofs can be adapted for more complex ones.

(B) There exists $u \in (\frac{11}{12}, 1)$ and $v \in \left(\frac{3}{4}, 3u - 2\right)$ such that, for all $t > 0$, $c > 0$, $w_t = t^{-u}$, $\gamma_c \triangleq c^{-v}$.

The following convergence result then applies to any sequence $(\mathbf{D}_t)_t$ produced by SOMF, using estimators (b) or (c). $\bar{f}$ is the empirical risk defined in (3).

*Proposition 1 (*SOMF *convergence).* Under assumptions **(A)** and **(B)**, $\bar{f}(\mathbf{D}_t)$ converges with probability one and every limit point $\mathbf{D}_\infty$ of $(\mathbf{D}_t)_t$ is a stationary point of $\bar{f}$: for all $\mathbf{D} \in \mathcal{C}$

$$\nabla \bar{f}(\mathbf{D}_\infty, \mathbf{D} - \mathbf{D}_\infty) \geq 0 \tag{29}$$

This result applies for any positive subsampling ratio r, which may be set arbitrarily high. However, selecting a reasonable ratio remains important for performance.

Proposition 1 is a corollary of a stronger result on SAMM algorithms. As it provides insights on the convergence mechanisms, we formalize this result in the following.

4.2 Basic assumptions and results on SMM convergence

We first recall the main results on stochastic majorization-minimization algorithms, established in [10], under assumptions that we slightly tighten for our purpose. In our setting, we consider the empirical risk minimization problem

$$\min_{\theta \in \Theta} \left(\bar{f}(\theta) \triangleq \frac{1}{n} \sum_{i=1}^{n} f(\theta, \mathbf{x}^{(i)}) \right), \tag{30}$$

where $f : \mathbb{R}^K \times \mathcal{X} \to \mathbb{R}$ is a loss function and

(C) $\Theta \subset \mathbb{R}^K$ and the support $\mathcal{X}$ of the data are compact.

This is a special case of (5) where the samples $(\mathbf{x}_t)_t$ are drawn uniformly from the set $\{\mathbf{x}^{(i)}\}_i$. The loss functions $f_t \triangleq f(\cdot, \mathbf{x}_t)$ defined on $\mathbb{R}^K$ can be non-convex. We instead assume that they meet reasonable regularity conditions:

(D) $(f_t)_t$ is uniformly R-Lipschitz continuous on $\mathbb{R}^K$ and uniformly bounded on Θ.

(E) The directional derivatives [35] $\nabla f_t(\theta, \theta' - \theta)$ and $\nabla \bar{f}(\theta, \theta' - \theta)$ exist for all θ and θ' in $\mathbb{R}^K$.

Assumption **(E)** allows to characterize the *stationary points* of problem (30), namely $\theta \in \Theta$ such that $\nabla \bar{f}(\theta, \theta' - \theta) \geq 0$ for all $\theta' \in \Theta$ — intuitively a point is stationary when there is no local direction in which the objective can be improved.

Let us now recall the definition of first-order surrogate functions used in the SMM algorithm. $(g_t)_t$ are selected in the set $\mathcal{S}_{\rho,L}(f_t, \theta_{t-1})$, hereby introduced.

Definition 1 (First-order surrogate function). Given a function $f : \mathbb{R}^K \to \mathbb{R}$, $\theta \in \Theta$ and $\rho, L > 0$, we define $\mathcal{S}_{\rho,L}(f, \theta)$ as the set of functions $g : \mathbb{R}^K \to \mathbb{R}$ such that

- g is majorizing f on Θ and g is ρ-strongly convex,
- g and f are tight at θ — i.e., $g(\theta) = f(\theta)$, $g - f$ is differentiable, $\nabla(g - f)$ is L-Lipschitz, $\nabla(g - f)(\theta) = 0$.

In OMF, g_t defined in (15) is a *variational* surrogate[4] of f_t. We refer the reader to [36] for further examples of first-order surrogates. We also ensure that $\bar{g}_t$ should be *parametrized* and thus representable in memory. The following assumption is met in OMF, as $\bar{g}_t$ is parametrized by the matrices $\bar{\mathbf{C}}_t$ and $\bar{\mathbf{B}}_t$.

(F) *Parametrized surrogates.* The surrogates $(\bar{g}_t)_t$ are parametrized by vectors in a compact set $\mathcal{K} \subset \mathbb{R}^P$. Namely, for all $t > 0$, there exists $\boldsymbol{\kappa}_t \in \mathcal{K}$ such that $\bar{g}_t$ is unequivocally defined as $g_t \triangleq \bar{g}_{\boldsymbol{\kappa}_t}$.

[4] In this case as in SOMF, g_t is not ρ-strongly convex but $\bar{g}_t$ is, thanks to assumption **(A)**. This is sufficient in the proofs of convergence.

Finally, we ensure that the weights $(w_t)_t$ used in Alg. 2 decrease at a certain rate.

(G) There exists $u \in (\frac{3}{4}, 1)$ such that $w_t = t^{-u}$.

When $(\theta_t)_t$ is the sequence yielded by Alg. 2, the following result (Proposition 3.4 in [10]) establishes the convergence of $(\bar{f}(\theta_t))_t$ and states that θ_t is asymptotically a stationary point of the finite sum problem (30), as a special case of the *expected* risk minimization problem (5).

Proposition 2 (Convergence of SMM*, from [10]).* Under assumptions **(C)** – **(G)**, $(\bar{f}(\theta_t))_{t\geq 1}$ converges with probability one. Every limit point θ_∞ of $(\theta_t)_t$ is a stationary point of the risk $\bar{f}$ defined in (30). That is,

$$\forall \theta \in \Theta, \quad \nabla \bar{f}(\theta_\infty, \theta - \theta_\infty) \geq 0. \tag{31}$$

The correctness of the online matrix factorization algorithm can be deduced from this proposition.

4.3 Convergence of SAMM

We now introduce assumptions on the approximations made in SAMM, before extending the result of Proposition 2. We make hypotheses on both the surrogate computation (*majorization*) step and the iterate update (*minimization*) step. The principles of SAMM are illustrated in Figure 2, which provides a geometric interpretation of the approximations introduced in the following assumptions **(H)** and **(I)**.

4.3.1 Approximate surrogate computation

The SMM algorithm selects a surrogate for f_t at point θ_{t-1} within the set $\mathcal{S}_{\rho,L}(f_t, \theta_{t-1})$. Surrogates within this set are *tight* at θ_{t-1} and greater than f_t everywhere. In SAMM, we allow the use of surrogates that are only *approximately majorizing* f_t and *approximately tight* at θ_{t-1}. This is indeed what SOMF does when using estimators in the code computation step. For that purpose, we introduce the set $\mathcal{T}_{\rho,L}(f, \theta, \epsilon)$, that contains all functions ϵ-close of a surrogate in $\mathcal{S}_{\rho,L}(f, \theta)$ for the ℓ_∞-norm:

Definition 2 (Approximate first-order surrogate function). Given a function $f : \mathbb{R}^K \to \mathbb{R}$, $\theta \in \Theta$ and $\epsilon > 0$, $\mathcal{T}_{\rho,L}(f, \theta, \epsilon)$ is the set of ρ-strongly convex functions $g : \mathbb{R}^K \to \mathbb{R}$ such that

- g is ϵ-majorizing f on Θ: $\forall\, \kappa \in \Theta$, $g(\kappa) - f(\kappa) \geq -\epsilon$,
- g and f are ϵ-tight at θ — i.e., $g(\theta) - f(\theta) \leq \epsilon$, $g - f$ is differentiable, $\nabla(g - f)$ is L-lipschitz.

We assume that SAMM selects an approximative surrogate in $\mathcal{T}_{\rho,L}(f_t, \theta_{t-1}, \epsilon_t)$ at each iteration, where $(\epsilon_t)_t$ is a deterministic or random non-negative sequence that vanishes at a sufficient rate.

(H) For all $t > 0$, there exists $\epsilon_t > 0$ such that $g_t \in \mathcal{T}_{\rho,L}(f_t, \theta_{t-1}, \epsilon_t)$. There exists a constant $\eta > 0$ such that $\mathbb{E}[\epsilon_t] \in \mathcal{O}(t^{2(u-1)-\eta})$ and $\epsilon_t \to_\infty 0$ almost surely.

As illustrated on Figure 2, given the OMF surrogate $g_t^\star \in \mathcal{S}_{\rho,L}(f_t, \theta_{t-1})$ defined in (15), any function g_t such that $\|g_t - g_t^\star\|_\infty < \epsilon$ is in $\mathcal{T}_{\rho,L}(f_t, \theta_{t-1}, \epsilon)$ — *e.g.*, where g_t uses an approximate $\boldsymbol{\alpha}_t$ in (15). This assumption can also be met in matrix factorization settings with difficult code regularizations, that *require* to make code approximations.

4.3.2 Approximate surrogate minimization

We do not require θ_t to be the minimizer of $\bar{g}_t$ any longer, but ensure that the surrogate objective function $\bar{g}_t$ decreases "fast enough". Namely, θ_t obtained from partial minimization should be closer to a minimizer of $\bar{g}_t$ than θ_{t-1}. We write $(\mathcal{F}_t)_t$ and $(\mathcal{F}_{t-\frac{1}{2}})_t$ the filtrations induced by the past of the algorithm, respectively up to the end of iteration t and up to the beginning of the minimization step in iteration t. Then, we assume

(I) For all $t > 0$, $\bar{g}_t(\theta_t) < \bar{g}_t(\theta_{t-1})$. There exists $\mu > 0$ such that, for all $t > 0$, where $\theta_t^\star = \operatorname{argmin}_{\theta\in\Theta} \bar{g}_t(\theta)$,

$$\mathbb{E}[\bar{g}_t(\theta_t) - \bar{g}_t(\theta_t^\star) | \mathcal{F}_{t-\frac{1}{2}}] \leq (1 - \mu)(\bar{g}_t(\theta_{t-1}) - \bar{g}_t(\theta_t^\star)). \tag{32}$$

Assumption **(I)** is met by choosing an appropriate method for the inner $\bar{g}_t$ minimization step — a large variety of gradient-descent algorithms indeed have convergence rates of the form (32). In SOMF, the block coordinate descent with frozen coordinates indeed meet this property, relying on results from [37]. When both assumptions are met, SAMM enjoys the same convergence guarantees as SMM.

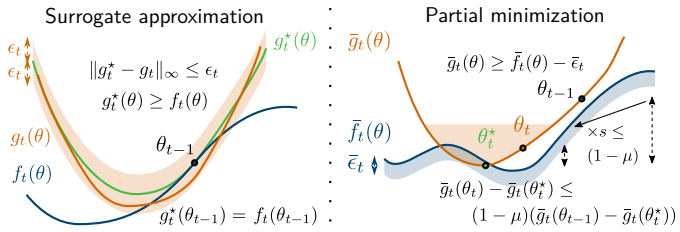

Figure 2: Both steps of SAMM make well-behaved approximations. The operations that are performed in exact SMM are in green and superscripted by *, while the actual computed values are in orange. Light bands recall the bounds on approximations assumed in **(H)** and **(I)**.

4.3.3 Asymptotic convergence guarantee

The following proposition guarantees that the stationary point condition of Proposition 2 holds for the SAMM algorithm, despite the use of approximate surrogates and approximate minimization.

Proposition 3 (Convergence of SAMM*).* Under assumptions **(C)** – **(I)**, the conclusion of Proposition 2 holds for SAMM.

Assumption **(H)** is essential to bound the errors introduced by the sequence $(\epsilon_t)_t$ in the proof of Proposition 3, while **(I)** is the key element to show that the sequence of iterates $(\theta_t)_t$ is stable enough to ensure convergence.The result holds for any subsampling ratio r, provided that **(A)** remains true.

4.3.4 Proving somf convergence

Assumptions **(A)** and **(B)** readily implies **(C)**–**(G)**. With Proposition 3 at hand, proving Proposition 1 reduces to ensure that the surrogate sequence of SOMF meets **(H)** while its iterate sequence meets **(I)**.

5 Experiments

The SOMF algorithm is designed for datasets with large number of samples n and large dimensionality p. Indeed, as detailed in Section 3.1, subsampling removes the computational bottlenecks that arise from high dimensionality. Proposition 1 establishes that the subsampling used in SOMF is safe, as it enjoys the same guarantees as OMF. However, as with OMF, no convergence rate is provided.

We therefore perform a strong empirical validation of subsampling.

We tackle two different problems, in functional Magnetic Resonance Imaging (fMRI) and hyperspectral imaging. Both involve the factorization of very large matrices $\mathbf{X}$ with sparse factors. As the data we consider are huge, subsampling reduces the time of a single iteration by a factor close to $\frac{p}{q}$. Yet it is also much redundant: SOMF makes little approximations and accessing only a fraction of the features per iteration should not hinder much the refinement of the dictionary. Hence high speed-ups are expected — and indeed obtained. All experiments can be reproduced using open-source code.

5.1 Problems and datasets

5.1.1 Functional MRI

Matrix factorization has long been used on functional Magnetic Resonance Imaging [18]. Data are temporal series of 3D images of brain activity and are decomposed into spatial modes capturing regions that activate synchronously. They form a matrix $\mathbf{X}$ where columns are the 3D images, and rows corresponds to voxels. Interesting dictionaries for neuroimaging capture spatially-localized components, with a few brain regions. This can be obtained by enforcing sparsity on the dictionary: we use an ℓ_2 penalty and the elastic-net constraint. SOMF streams subsampled 3D brain records to learn the sparse dictionary $\mathbf{D}$. Data can be huge: we use the whole HCP dataset [38], with $n = 2.4 \cdot 10^6$ (2000 records, 1 200 time points) and $p = 2 \cdot 10^5$, totaling 2 TB of dense data. For comparison, we also use a smaller public dataset (ADHD200 [39]) with 40 records, $n = 7000$ samples and $p = 6 \cdot 10^4$ voxels. Historically, brain decomposition have

Table 2: Summary of experimental settings

Field	Functional MRI		Hyperspectral imaging
Dataset	ADHD	HCP	Patches from AVIRIS
Factors	**D** sparse, **A** dense		**D** dense, **A** sparse
# samples n	$7 \cdot 10^3$	$2 \cdot 10^6$	$2 \cdot 10^6$
# features p	$6 \cdot 10^4$	$2 \cdot 10^5$	$6 \cdot 10^4$
X size	2 GB	2 TB	103 GB
Use case ex.	Extracting predictive feature		Recognition / denoising

been obtained by minimizing the classical dictionary learning objective on *transposed* data [40]: the code **A** holds sparse spatial maps and voxel time-series are streamed. This is not a natural streaming order for fMRI data as **X** is stored columnwise on disk, which makes the sparse dictionary formulation more appealing. Importantly, we seek a *low-rank* factorization, to keep the decomposition interpretable — $k \sim 100 \ll p$.

5.1.2 Hyperspectral imaging

Hyperspectral cameras acquire images with many channels that correspond to different spectral bands. They are used heavily in remote sensing (satellite imaging), and material study (microscopic imaging). They yield digital images with around 1 million pixels, each associated with hundreds of spectral channels. Sparse matrix factorization has been widely used on these data for image classification [41, 42] and denoising [43, 44]. All methods rely on the extraction of full-band patches representing a local image neighborhood with all channels included. These patches are very high dimensional, due to the number of spectral bands. From one image of the AVIRIS project [45], we extract $n = 2 \cdot 10^6$ patches of size 16×16 with 224 channels, hence $p = 6 \cdot 10^4$. A dense dictionary is learned from these patches. It should allow a sparse representation of samples: we either use the classical dictionary learning setting (ℓ_1/elastic-net penalty), or further add positive constraints to the dictionary and codes: both methods may be used and deserved to be benchmarked. We seek a dictionary of reasonable size: we use $k \sim 256 \ll p$.

5.2 Experimental design

To validate the introduction of subsampling and the usefulness of SOMF, we perform two major experiments.

- We measure the performance of SOMF when increasing the reduction factor, and show benefits of stochastic dimension reduction on all datasets.
- We assess the importance of subsampling in each of the steps of SOMF. We compare the different approaches proposed for code computation.

Validation. We compute the objective function (3) over a test set to rule out any overfitting effect — a dictionary should be a good representation of unseen samples. This criterion is always plotted against wall-clock time, as we are interested in the performance of SOMF for practitioners.

Tools. To perform a valid benchmark, we implement OMF and SOMF using Cython [46] We use coordinate descent [47] to solve Lasso problems with optional positivity constraints. Code computation is parallelized to handle mini-batches. Experiments use *scikit-learn* [48] for numerics, and *nilearn* [49] for handling fMRI data. We have released the code in an open-source Python package[5]. Experiments were run on 3 cores of an Intel Xeon 2.6GHz, in which case computing $\mathbf{P}_t^{\perp}\bar{\mathbf{B}}_t$ is faster than updating $\mathbf{P}_t\mathbf{D}_t$.

[5] `https://github.com/arthurmensch/modl`

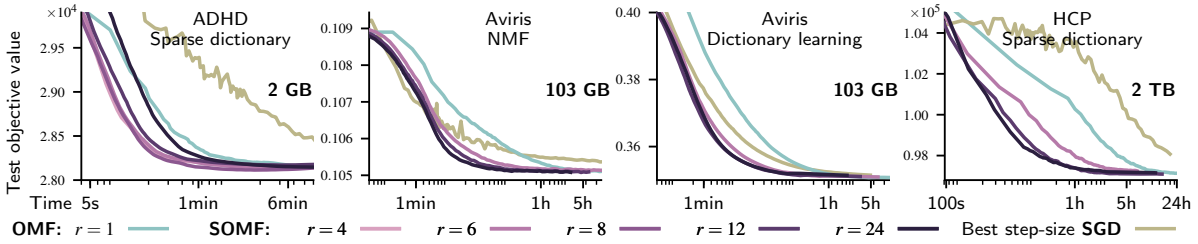

Figure 3: Subsampling provides significant speed-ups on all fMRI and hyperspectral datasets. A reduction factor of 12 is a good overall choice. With larger data, larger reduction factors can be used for better performance — convergence is reached 13× faster than state-of-the-art methods on the 2TB HCP dataset.

Parameter setting. Setting the number of components k and the amount of regularization λ is a hard problem in the absence of ground truth. Those are typically set by cross-validation when matrix factorization is part of a supervised pipeline. For fMRI, we set $k = 70$ to obtain interpretable networks, and set λ so that the decomposition approximately covers the whole brain (i.e., every map is $\frac{k}{70}$) sparse). For hyperspectral images, we set $k = 256$ and select λ to obtain a dictionary on which codes are around 3% sparse. We cycle randomly through the data (fMRI records, image patches) until convergence, using mini-batches of size $\eta = 200$ for HCP and AVIRIS, and $\eta = 50$ for ADHD (small number of samples). Hyperspectral patches are normalized in the dictionary learning setting, but not in the non-negative setting — the classical pre-conditioning for each case. We use $u = 0.917$ and $v = 0.751$ for weight sequences.

5.3 Reduction brings speed-up at all data scales

We benchmark SOMF for various reduction factors against the original online matrix factorization algorithm OMF [15], on the three presented datasets. We stream data in the same order for all reduction factors. Using variant (c) (true Gram matrix, averaged $\boldsymbol{\beta}_t$) performs slightly better on fMRI datasets, whereas (b) (averaged Gram matrix and $\boldsymbol{\beta}_t$) is slightly faster for hyperspectral decomposition. For comparison purpose, we display results using estimators (b) only.

Figure 3 plots the test objective against CPU time. First, we observe that all algorithms find dictionaries with very close objective function values for all reduction factors, on each dataset. This is not a trivial observation as the matrix factorization problem (3) is not convex and different runs of OMF and SOMF may converge towards minima with different values. Second, and most importantly, SOMF provides significant improvements in convergence speed for three different sizes of data and three different factorization settings. Both observations confirm the relevance of the subsampling approach.

Quantitatively, we summarize the speed-ups obtained in Table 3. On fMRI data, on both large and medium datasets, SOMF provides more than an order of magnitude speed-up. Practitioners working on datasets akin to HCP can decompose their data in 20 minutes instead of 4 h previously, while working on a single machine. We obtain the highest speed-ups for the largest dataset — accounting for the extra redundancy that usually appears when dataset size increase. Up to $r \sim 8$, speed-up is of the order of r — subsampling induces little noise in the iterate sequence, compared to OMF. Hyperspectral decomposition is performed near 7× faster than with OMF in the classical dictionary learning setting, and 3× in the non-negative setting, which further demonstrates the versatility of SOMF. Qualitatively, given a certain time budget, Figure 4 compares the results of OMF and the results of SOMF with a subsampling ratio $r = 24$, in the non-negative setting. Our algorithm yields a valid smooth bank of filters much faster. The same comparison has been made for fMRI in [26].

Comparison with stochastic gradient descent. It is possible to solve (3) using the projected stochastic gradient (SGD) algorithm [50]. On all tested settings, for high precision convergence, SGD (with the best step-size among a grid) is slower than OMF and even slower than SOMF. In the dictionary learning setting, SGD is somewhat faster than OMF but slower than SOMF in the first epochs. Compared to SOMF and OMF, SGD further requires to select the step-size by grid search.

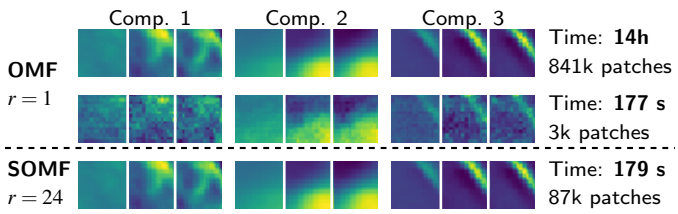

Figure 4: Given a 3 minute time budget, the atoms learned by SOMF are more focal and less noisy that those learned by OMF. They are closer to the dictionary of first line, for which convergence has been reached.

Table 3: Time to reach convergence ($< 1\%$ test objective)

Dataset	ADHD		AVIRIS (NMF)		AVIRIS (DL)		HCP	
Algorithm	OMF	SOMF	OMF	SOMF	OMF	SOMF	OMF	SOMF
Conv. time	6 min	**28 s**	2 h 30	**43 min**	1 h 16	**11 min**	3 h 50	**17 min**
Speed-up	11.8		3.36		6.80		13.31	

Limitations. Table 3 reports convergence time within 1%, which is enough for application in practice. SOMF is less beneficial when setting very high precision: for convergence within 0.01%, speed-up for HCP is 3.4. This is expected as SOMF trades speed for approximation. For high precision convergence, the reduction ratio can be reduced after a few epochs. As expected, there exists an *optimal* reduction ratio, depending on the problem and precision, beyond which performance reduces: $r = 12$ yields better results than $r = 24$ on AVIRIS (dictionary learning) and ADHD, for 1% precision.

Our first experiment establishes the power of stochastic subsampling as a whole. In the following two experiments, we refine our analysis to show that subsampling is indeed useful in the three steps of online matrix factorization.

5.4 For each step of SOMF, subsampling removes a bottleneck

In Section 3, we have provided theoretical guidelines on when to introduce subsampling in each of the three steps of an iteration of SOMF. This analysis predicts that, for $\eta \sim k$, we should first use partial dictionary update, before using approximate code computation and asynchronous parameter aggregation. We verify this by measuring the time spent by SOMF on each of the updates for various reduction factors, on the HCP dataset. Results are presented in Figure 5. We observe that block coordinate descent is indeed the bottleneck in OMF. Introducing partial dictionary update removes this bottleneck, and as the reduction factor increases, code computation and surrogate aggregation becomes the major bottlenecks. Introducing subsampling as described in SOMF overcomes these bottlenecks, which rationalizes all steps of SOMF from a computational point of view.

5.5 Code subsampling becomes useful for high reduction

It remains to assess the performance of approximate code computation and averaging techniques used in SOMF. Indeed, subsampling for code computation introduces noise that may undermine the computational speed-up. To understand the impact of approximate code computation, we compare three strategies to compute $(\boldsymbol{\alpha}_t)_t$ on the HCP dataset. First, we compute $(\boldsymbol{\alpha}_t^\star)_t$ from $(\mathbf{x}_t)_t$ using (21). Subsampling is thus used only in dictionary update. Second, we rely on masked, non-consistent estimators (a), as in [26] — this breaks convergence guarantees. Third, we use averaged estimators $(\boldsymbol{\beta}_t, \mathbf{G}_t)$ from (c) to reduce the variance in $(\boldsymbol{\alpha}_t)_t$ computation.

Fig. 6 compares the three strategies for $r \in \{12, 24\}$. Partial minimization at each step is the most important part to accelerate convergence: subsampling the dictionary updates already allows to outperforms OMF. This is expected, as dictionary update constitutes the main bottleneck of OMF in large-scale settings. Yet, for large reduction factors, using subsampling in code computation is important to further accelerate convergence. This clearly appears when comparing the plain and dashed black curves.

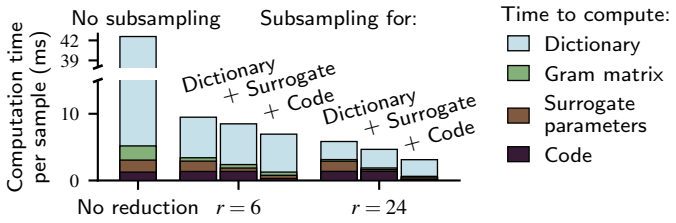

Figure 5: Profiling OMF and SOMF on HCP. Partial dictionary update removes the major bottleneck of online matrix factorization for small reductions. For higher reduction, parameter update and code computation must be subsampled to further reduce the iteration time.

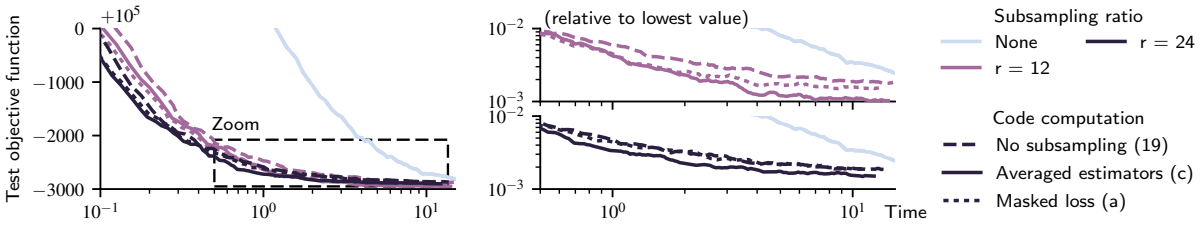

Figure 6: Approximating code computation with the proposed subsampling method further accelerates the convergence of SOMF. Refining code computation using past iterations (averaged estimates) performs better than simply performing a subsampled linear regression as in [26]

Using past estimates to better approximate $(\boldsymbol{\alpha}_t)_t$ yields faster convergence than the non-converging, masked loss strategy (a) proposed in [26].

6 Conclusion

In this paper, we introduce SOMF, a matrix-factorization algorithm that can handle input data with very large number of rows and columns. It leverages subsampling within the inner loop of a streaming algorithm to make iterations faster and accelerate convergence. We show that SOMF provides a stationary point of the non-convex matrix factorization problem. To prove this result, we extend the stochastic majorization-minimization framework to two major approximations. We assess the performance of SOMF on real-world large-scale problems, with different sparsity/positivity requirements on learned factors. In particular, on fMRI and hyperspectral data decomposition, we show that the use of subsampling can speed-up decomposition up to 13 times. The larger the dataset, the more SOMF outperforms state-of-the art techniques, which is very promising for future applications. This calls for adaptation of our approach to learn more complex models.

References

[1] Julien Mairal. Sparse Modeling for Image and Vision Processing. *Foundations and Trends in Computer Graphics and Vision*, 8(2-3):85–283, 2014.

[2] Nathan Srebro, Jason Rennie, and Tommi S. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1329–1336, 2004.

[3] Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[4] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global Vectors for Word Representation. In *Proc. Conf. EMNLP*, volume 14, pages 1532–43, 2014.

[5] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185, 2014.

[6] Yin Zhang, Matthew Roughan, Walter Willinger, and Lili Qiu. Spatio-Temporal Compressive Sensing and Internet Traffic Matrices. 2009.

[7] Hyunsoo Kim and Haesun Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502, 2007.

[8] Gaël Varoquaux et al. Multi-subject dictionary learning to segment an atlas of brain spontaneous activity. In *Proc. IPMI Conf.*, pages 562–573, 2011.

[9] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT*, pages 177–186, 2010.

[10] Julien Mairal. Stochastic majorization-minimization algorithms for large-scale optimization. In *Adv. Neural Inform. Process. Syst.*, pages 2283–2291, 2013.

[11] Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013.

[12] Samuel Burer and Renato D. C. Monteiro. Local Minima and Convergence in Low-Rank Semidefinite Programming. *Math. Program.*, 103(3):427–444, 2004.

[13] Benjamin Recht and Christopher Ré. Parallel stochastic gradient algorithms for large-scale matrix completion. *Math. Program. Comput.*, 5(2):201–226, 2013.

[14] Robert M. Bell and Yehuda Koren. Lessons from the Netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

[15] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *J. Machine Learning Research*, 11:19–60, 2010.

[16] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

[17] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: Applications to image and text data. In *Proc. SIGKDD Conf.*, pages 245–250, 2001.

[18] M. J. McKeown et al. Analysis of fMRI Data by Blind Separation into Independent Spatial Components. *Hum. Brain Mapp.*, 6(3):160–188, 1998.

[19] Emmanuel J. Candès and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[20] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *arXiv:0909.4061 [math]*, 2009.

[21] Vladimir Rokhlin et al. A randomized algorithm for principal component analysis. *SIAM J. Matrix Anal. Appl.*, 31(3):1100–1124, 2009.

[22] Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *Proc. IEEE Symp. Found. Comput. Science*, pages 143–152, 2006.

[23] Yichao Lu et al. Faster ridge regression via the subsampled randomized hadamard transform. In *Adv. Neural Inform. Process. Syst.*, pages 369–377, 2013.

[24] Mert Pilanci and Martin Wainwright. Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares. *JMLR*, 17:1–33, 2015.

[25] Garvesh Raskutti and Michael Mahoney. Statistical and algorithmic perspectives on randomized sketching for ordinary least-squares. In *Proc. ICML*, pages 617–625, 2015.

[26] Arthur Mensch, Julien Mairal, Bertrand Thirion, and Gaël Varoquaux. Dictionary learning for massive matrix factorization. In *Proc. ICML*, pages 1737–1746, 2016.

[27] Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Res.*, 37(23):3311–3325, 1997.

[28] Robert Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Series B Stat. Methodol.*, 58(1):267–288, 1996.

[29] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *J. Comput. Graph. Stat.*, 15(2):265–286, 2006.

[30] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Series B Stat. Methodol.*, 67(2):301–320, 2005.

[31] Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[32] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization Methods for Large-Scale Machine Learning. *arXiv:1606.04838 [stat.ML]*, 2016.

[33] Aad W. Van der Vaart. *Asymptotic Statistics*, volume 3. CUP, 2000.

[34] Morteza Mardani et al. Subspace Learning and Imputation for Streaming Big Data Matrices and Tensors. *IEEE TSP*, 63(10):2663–2677, 2015.

[35] Jonathan M. Borwein and Adrian S. Lewis. *Convex Analysis and Nonlinear Optimization: Theory and Examples*. Springer Science & Business Media, 2010.

[36] Julien Mairal. Optimization with first-order surrogate functions. In *Proceedings of the International Conference on Machine Learning*, pages 783–791, 2013.

[37] Stephen J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

[38] David C. Van Essen et al. The WU-Minn Human Connectome Project: An overview. *NeuroImage*, 80:62–79, 2013.

[39] Michael P. Milham et al. The adhd-200 consortium: a model to advance the translational potential of neuroimaging in clinical neuroscience. *Front. Syst. Neurosci.*, 6(62), 2012.

[40] Gaël Varoquaux, Yannick Schwartz, Philippe Pinel, and Bertrand Thirion. Cohort-level brain mapping: Learning cognitive atoms to single out specialized regions. In *Proceedings of the Information Processing in Medical Imaging Conference*, pages 438–449, 2013.

[41] Yi Chen, Nasser M. Nasrabadi, and Trac D. Tran. Hyperspectral image classification using dictionary-based sparse representation. *IEEE Transactions on Geoscience and Remote Sensing*, 49(10):3973–3985, 2011.

[42] A. Soltani-Farani, H. R. Rabiee, and S. A. Hosseini. Spatial-Aware Dictionary Learning for Hyperspectral Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 53(1):527–541, 2015.

[43] Matteo Maggioni, Vladimir Katkovnik, Karen Egiazarian, and Alessandro Foi. Nonlocal transform-domain filter for volumetric data denoising and reconstruction. *IEEE Trans. Image Process.*, 22(1):119–133, 2013.

[44] Yi Peng et al. Decomposable nonlocal tensor dictionary learning for multispectral image denoising. In *Proc. IEEE Conf. CVPR*, pages 2949–2956, 2014.

[45] Gregg Vane. First results from the airborne visible/infrared imaging spectrometer (AVIRIS). In *Ann. Tech. Symp. Int. Soc. Optics Photonics*, pages 166–175, 1987.

[46] Stefan Behnel et al. Cython: The best of both worlds. *Computing in Science & Engineering*, 13(2):31–39, 2011.

[47] Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[48] Fabian Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[49] Alexandre Abraham et al. Machine learning for neuroimaging with scikit-learn. *Frontiers in Neuroinformatics*, 8:14, 2014.

[50] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l1-ball for learning in high dimensions. In *Proceedings of the International Conference on Machine Learning*, pages 272–279, 2008.

[51] Michel Métivier. *Semimartingales: A Course on Stochastic Processes*, volume 2. Walter de Gruyter, 1982.

[52] Amir Beck and Luba Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060, 2013.

[53] Joseph Leo Doob. *Stochastic processes*. John Wiley & Sons, 1990.

A Proofs of convergence

This appendix contains the detailed proofs of Proposition 3 and Proposition 1. We first introduce three lemmas that will be crucial to prove SAMM convergence, before establishing it by proving Proposition 3. Finally, we show that SOMF is indeed an instance of SAMM (*i.e.* meets the assumptions **(C)**–**(I)**), proving Proposition 1.

A.1 Basic properties of the surrogates, estimate stability

We derive an important result on the stability and optimality of the sequence $(\theta_t)_t$, formalized in Lemma 3 — introduced in the main text. We first introduce a numerical lemma on the boundedness of well-behaved determistic and random sequence. The proof is detailed in Appendix B.

Lemma 1 (Bounded quasi-geometric sequences). Let $(x_t)_t$ be a sequence in $\mathbb{R}^+$, $u : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, $t_0 \in \mathbb{N}$ and $\alpha \in [0, 1)$ such that, for all $t \geq t_0$, $x_t \leq \alpha x_{t-1} + u(x_t, x_{t-1})$, where $u(x, y) \in o(x + y)$ for $x, y \to \infty$. Then $(x_t)_t$ is bounded.

Let now $(X_t)_t$ be a random sequence in $\mathbb{R}^+$, such that $\mathbb{E}[X_t] < \infty$. We define $(\mathcal{F}_t)_t$ the filtration adapted to $(X_t)_t$. If, for all $t > t_0$, there exists a σ-algebra $\mathcal{F}_{t'}$ such that $\mathcal{F}_{t-1} \subseteq \mathcal{F}_{t'} \subseteq \mathcal{F}_t$ and

$$\mathbb{E}[X_t|\mathcal{F}_{t'}] \leq \alpha X_{t-1} + u(X_t, X_{t-1}), \tag{33}$$

then $(X_t)_t$ is bounded almost surely.

We first derive some properties of the approximate surrogate functions used in SAMM. The proof is adapted from [10].

Lemma 2 (Basic properties of approximate surrogate functions). Consider any sequence of iterates $(\theta_t)_t$ and assume there exists $\epsilon > 0$ such that $g_t \in \mathcal{T}_{L,\rho}(f_t, \theta_{t-1}, \epsilon)$ for all $t \geq 1$. Define $h_t \triangleq g_t - f_t$ for all $t \geq 1$, $\bar{h}_0 \triangleq h_0$ and $\bar{h}_t \triangleq (1 - w_t)\bar{h}_{t-1} + w_t h_t$. Under assumptions **(D)** – **(G)**,

(i) $(\nabla h_t(\theta_{t-1}))_{t>0}$ is uniformly bounded and there exists R' such that $\{\nabla h_t\}_t$ is uniformly bounded by R'.

(ii) $(h_t)_t$ and $(\bar{h}_t)_t$ are uniformly R'-Lipschitz, $(g_t)_t$ and $(\bar{g}_t)_t$ are uniformly $(R + R')$-Lipschitz.

Proof. We first prove (i). We set $\alpha > 0$ and define $\theta' = \theta_t - \alpha \frac{\nabla h_t(\theta_t)}{\|\nabla h_t(\theta_t)\|_2}$. As h_t has a L-Lipschitz gradient on $\mathbb{R}^K$, using Taylor's inequality (see Appendix B)

$$\begin{aligned} h_t(\theta') &\leq h_t(\theta_t) - \alpha\|\nabla h_t(\theta_t)\|_2 + \frac{L\alpha^2}{2} \\ \|\nabla h_t(\theta_t)\|_2 &\leq \frac{1}{\alpha}(h_t(\theta_t) - h_t(\theta')) + \frac{L\alpha}{2} \leq \frac{2}{\alpha}\epsilon + \frac{L\alpha}{2}, \end{aligned} \tag{34}$$

where we use $h_t(\theta_t) < \epsilon$ and $-h_t(\theta'_t) \leq \epsilon$ from the assumption $g_t \in \mathcal{T}_{L,\rho}(f_t, \theta_{t-1}, \epsilon)$. Moreover, by definition, ∇h_t exists and is L-lipschitz for all t. Therefore, $\forall\, t \geq 1$,

$$\|\nabla h_t(\theta)\|_2 \leq \|\nabla h_t(\theta_t)\|_2 + L\|\theta_{t-1} - \theta\|_2 \tag{35}$$

Since Θ is compact and $(\|\nabla h_t(\theta_t)\|_2)_{t\geq 1}$ is bounded in (34), ∇h_t is bounded by R' independent of t. (ii) follows by basic considerations on Lipschitz functions. □

Finally, we prove a result on the stability of the estimates, that derives from combining the properties of $(g_t)_t$ and the geometric decrease assumption **(I)**.

Lemma 3 (Estimate stability under SAMM approximation). In the same setting as Lemma 2, with the additional assumption **(I)** (expected linear decrease of $\bar{g}_t$ suboptimality), the sequence $\|\theta_t - \theta_{t-1}\|_2$ converges to 0 as fast as $(w_t)_t$, and θ_t is asymptotically an exact minimizer. Namely, almost surely,

$$\|\theta_t - \theta_{t-1}\|_2 \in \mathcal{O}(w_t) \text{ and } \bar{g}_t(\theta_t) - \bar{g}_t(\theta_t^\star) \in \mathcal{O}(w_t^2). \tag{36}$$

Proof. We first establish the result when a deterministic version of **(I)** holds, as it makes derivations simpler to follow.

A.1.1 Determistic decrease rate

We temporarily assume that decays are deterministic.

($\mathbf{I}_{\text{det}}$) For all $t > 0$, $\bar{g}_t(\theta_t) < \bar{g}_t(\theta_{t-1})$. Moreover, there exists $\mu > 0$ such that, for all $t > 0$

$$\begin{aligned} \bar{g}_t(\theta_t) - \bar{g}_t(\theta_t^\star) &\leq (1-\mu)(\bar{g}_t(\theta_{t-1}) - \bar{g}_t(\theta_t^\star)) \\ \text{where} \quad \theta_t^\star &= \operatorname*{argmin}_{\theta \in \Theta} \bar{g}_t(\theta), \end{aligned} \tag{37}$$

We introduce the following auxiliary positive values, that we will seek to bound in the proof:

$$\begin{aligned} A_t \triangleq \|\theta_t - \theta_{t-1}\|_2, \quad B_t \triangleq \|\theta_t - \theta_t^\star\|_2, \\ C_t \triangleq \|\theta_t^\star - \theta_{t-1}^\star\|_2, \quad D_t \triangleq \bar{g}_t(\theta_t) - \bar{g}_t(\theta_t^\star). \end{aligned} \tag{38}$$

Our goal is to bound A_t. We first relate it to C_t and B_t using convexity of ℓ_2 norm:

$$A_t^2 \leq 3B_t^2 + 3B_{t-1}^2 + 3C_t^2. \tag{39}$$

As $\theta_t^\star$ is the minimizer of $\bar{g}_t$, by strong convexity of $(\bar{g}_t)_t$,

$$\frac{\rho}{2} B_t^2 = \frac{\rho}{2} \|\theta_t - \theta_t^\star\|_2^2 \leq D_t, \tag{40}$$

while we also have

$$\begin{aligned} &\frac{\rho}{2}\|\theta_t^\star - \theta_{t-1}^\star\|_2^2 \leq \bar{g}_t(\theta_{t-1}^\star) - \bar{g}_t(\theta_t^\star) \\ &\leq (1-w_t)\big(\bar{g}_{t-1}(\theta_{t-1}^\star) - \bar{g}_{t-1}(\theta_t^\star)\big) + w_t\big(g_t(\theta_{t-1}^\star) - g_t(\theta_t^\star)\big) \\ &\leq w_t(R+R')\|\theta_t^\star - \theta_{t-1}^\star\|_2, \text{ and thus } C_t \leq w_t \frac{2Q}{\rho}. \end{aligned} \tag{41}$$

The second inequalities holds because $\theta_{t-1}^\star$ is a minimizer of $\bar{g}_{t-1}$ and g_t is Q-Lipschitz, where $Q \triangleq R+R'$, using Lemma 2. Replacing (40) and (41) in (39) yields

$$A_t^2 \leq \frac{6}{\rho}(D_t + D_{t-1}) + \frac{12Q^2}{\rho} w_t^2, \tag{42}$$

and we are left to show that $D_t \in \mathcal{O}(w_t^2)$ to conclude. For this, we decompose the inequality from $\mathbf{(I_{\text{det}})}$ into

$$\begin{aligned} D_t &\leq (1-\mu)(\bar{g}_t(\theta_{t-1}) - \bar{g}_t(\theta_t^\star)) \\ &= (1-\mu)\Big(w_t\big(g_t(\theta_{t-1}) - g_t(\theta_t)\big) + w_t\big(g_t(\theta_t) - g_t(\theta_t^\star)\big)\Big) \\ &\quad + (1-\mu)\Big((1-w_t)\big(\bar{g}_{t-1}(\theta_{t-1}) - \bar{g}_{t-1}(\theta_{t-1}^\star)\big) \\ &\qquad\qquad + (1-w_t)\big(\bar{g}_{t-1}(\theta_{t-1}^\star) - \bar{g}_{t-1}(\theta_t^\star)\big)\Big) \\ &\leq (1-\mu)(w_t Q(A_t + B_t) + D_{t-1}), \end{aligned} \tag{43}$$

where the second inequality holds for the same reasons as in (41). Injecting (40) and (42) in (43), we obtain

$$\tilde{D}_t \leq (1-\mu)\tilde{D}_{t-1}\frac{w_{t-1}^2}{w_t^2} + u(\tilde{D}_t, \tilde{D}_{t-1}), \tag{44}$$

where we define $\tilde{D}_t \triangleq \frac{D_t}{w_t^2}$. It is easy to show (see algebraic details in Appendix B) that the perturbation term $u(\tilde{D}_t, \tilde{D}_{t-1}) \in o(\tilde{D}_t + \tilde{D}_{t-1})$ if $\tilde{D}_t \to \infty$. Using the determistictic result of Lemma 1, this ensures that $\tilde{D}_t$ is bounded, which combined with (40) allows to conclude.

A.1.2 Stochastic decrease rates

In the general case **(I)**, the inequalities (40), (41) and (42) holds, and (44) is replaced by

$$\mathbb{E}[\tilde{D}_t|\mathcal{F}_{t-\frac{1}{2}}] \leq (1-\mu)\tilde{D}_{t-1}\frac{w_{t-1}^2}{w_t^2} + u(\tilde{D}_t, \tilde{D}_{t-1}), \tag{45}$$

Taking the expectation of this inequality and using Jensen inequality, we show that (43) holds when replacing $\tilde{D}_t$ by $\mathbb{E}[\tilde{D}_t]$. This shows that $\mathbb{E}[D_t] \in \mathcal{O}(w_t^2)$ and thus $\mathbb{E}[D_t] < \infty$. The result follows from Lemma 1, that applies as $\mathcal{F}_{t-1} \subseteq \mathcal{F}_{t-\frac{1}{2}} \subseteq \mathcal{F}_t$. □

A.2 Convergence of SAMM — Proof of Proposition 3

We now proceed to prove the Proposition 3, that extends the stochastic majorization-minimization framework to allow approximations in both majorization and minimizations steps.

Proof of Proposition 3. We adapt the proof of Proposition 3.3 from [10] (reproduced as Proposition 2 in our work). Relaxing tightness and majorizing hypotheseses introduces some extra error terms in the derivations. Assumption **(H)** allows to control these extra terms without breaking convergence. The stability Lemma 3 is important in steps 3 and 5.

A.2.1 Almost sure convergence of $(\bar{g}_t(\theta_t))$

We control the positive expected variation of $(g_t(\theta_t))_t$ to show that it is a converging quasi-martingale. By construction of $\bar{g}_t$ and properties of the surrogates $g_t \in \mathcal{T}_{\rho,L}(f_t, \theta_{t-1}, \epsilon_t)$, where ϵ_t is a non-negative sequence that meets **(H)**,

$$\begin{aligned}
&\bar{g}_t(\theta_t) - \bar{g}_{t-1}(\theta_{t-1}) \\
&= (\bar{g}_t(\theta_t) - \bar{g}_t(\theta_{t-1})) + w_t(g_t(\theta_{t-1}) - \bar{g}_{t-1}(\theta_{t-1})) \\
&\leq w_t(g_t(\theta_{t-1}) - \bar{g}_{t-1}(\theta_{t-1})) \\
&\leq w_t(g_t(\theta_{t-1}) - f_t(\theta_{t-1})) + w_t(f_t(\theta_{t-1}) - \bar{f}_{t-1}(\theta_{t-1})) \\
&\quad + w_t(\bar{f}_{t-1}(\theta_{t-1}) - \bar{g}_{t-1}(\theta_{t-1})) \\
&\leq w_t(f_t(\theta_{t-1}) - \bar{f}_{t-1}(\theta_{t-1})) + w_t(\bar{\epsilon}_{t-1} + \epsilon_t),
\end{aligned} \tag{46}$$

where the average error sequence $(\bar{\epsilon}_t)_t$ is defined recursively: $\bar{\epsilon}_0 \triangleq \epsilon_0$ and $\bar{\epsilon}_t \triangleq (1-w_t)\epsilon_{t-1} + w_t\epsilon_t$. The first inequality uses $\bar{g}_t(\theta_t) \leq \bar{g}_t(\theta_{t-1})$. To obtain the forth inequality we observe $g_t(\theta_{t-1}) - f_t(\theta_{t-1}) < \epsilon_t$ by definition of ϵ_t and $\bar{f}_t(\theta_{t-1}) - \bar{g}_t(\theta_{t-1}) \leq \bar{\epsilon}_t$, which can easily be shown by induction on t. Then, taking the conditional expectation with respect to $\mathcal{F}_{t-1}$,

$$\begin{aligned}
&\mathbb{E}[\bar{g}_t(\theta_t) - \bar{g}_{t-1}(\theta_{t-1})|\mathcal{F}_{t-1}] \\
&\leq w_t \sup_{\theta\in\Theta} |f(\theta) - \bar{f}_{t-1}(\theta)| + w_t(\bar{\epsilon}_{t-1} + \mathbb{E}[\epsilon_t|\mathcal{F}_{t-1}]).
\end{aligned} \tag{47}$$

We have used the fact that ϵ_{t-1} is deterministic with respect to $\mathcal{F}_{t-1}$. To ensure convergence, we must bound both terms in (47): the first term is the same as in the original proof with exact surrogate, while the second is the perturbative term introduced by the approximation sequence $(\epsilon_t)_t$. We use Lemma B.7 from [10], issued from the theory of empirical processes: $\mathbb{E}[\sup_{\theta\in\Theta} |f(\theta) - \bar{f}_{t-1}(\theta)|] = \mathcal{O}(w_t t^{1/2})$, and thus

$$\sum_{t=1}^{\infty} w_t \mathbb{E}[\sup_{\theta\in\Theta} |f(\theta) - \bar{f}_{t-1}(\theta)|] < C\sum_{t=1}^{\infty} t^{1/2} w_t^2 < \infty \tag{48}$$

where C is a constant, as $t^{1/2}w_t^2 = t^{1/2-2u}$ and $u > 3/4$ from **(G)**. Let us now focus on the second term of (47). Defining, for all $1 \leq i \leq t$, $w_i^t = w_i \prod_{j=i+1}^t (1-w_j)$,

$$\mathbb{E}[\bar{\epsilon}_t] = \sum_{i=1}^{t} w_i^t \mathbb{E}[\epsilon_t] \leq w_t \sum_{i=1}^{t} \mathbb{E}[\epsilon_t]. \tag{49}$$

We set $\eta > 0$ so that $2(u-1) - \eta > -1$. Assumption **(H)** ensures $\mathbb{E}[\epsilon_t] \in \mathcal{O}(t^{2(u-1)-\eta})$, which allows to bound the partial sum $\sum_{i=1}^{t} \mathbb{E}[\epsilon_i] \in \mathcal{O}(t^{2u-1-\eta})$. Therefore

$$\begin{aligned} & w_t \mathbb{E}[\bar{\epsilon}_{t-1} + \mathbb{E}[\epsilon_t | \mathcal{F}_{t-1}]] = w_t \mathbb{E}[\epsilon_{t-1}] + w_t \mathbb{E}[\epsilon_t] \\ & \leq w_t^2 \Big(\sum_{i=1}^{t} \mathbb{E}[\epsilon_t] \Big) + w_t \mathbb{E}[\epsilon_t] \\ & \leq A t^{2u-2u-1-\eta} + B t^{2u-u-2-\eta} \leq C t^{-1-\eta}, \end{aligned} \tag{50}$$

where we use $u < 1$ on the third line and the definition of $(w_t)_t$ on the second line. Thus $\sum_{t=1}^{\infty} w_t \mathbb{E}[\bar{\epsilon}_{t-1} + \mathbb{E}[\epsilon_t | \mathcal{F}_{t-1}]] < \infty$. We use quasi-martingale theory to conclude, as in [10]. We define the variable δ_t to be 1 if $\mathbb{E}[\bar{g}_t(\theta_t) - \bar{g}_{t-1}(\theta_{t-1}) | \mathcal{F}_{t-1}] \geq 0$, and 0 otherwise. As all terms of (47) are positive:

$$\begin{aligned} & \sum_{t=1}^{\infty} \mathbb{E}[\delta_t (\bar{g}_t(\theta_t) - \bar{g}_{t-1}(\theta_{t-1}))] \\ & = \sum_{t=1}^{\infty} \mathbb{E}[\delta_t \mathbb{E}[\bar{g}_t(\theta_t) - \bar{g}_{t-1}(\theta_{t-1}) | \mathcal{F}_{t-1}]] \\ & \leq \sum_{t=1}^{\infty} w_t \mathbb{E}[\sup_{\theta \in \Theta} |f(\theta) - \bar{f}_{t-1}(\theta)| + \bar{\epsilon}_{t-1} + \mathbb{E}[\epsilon_t | \mathcal{F}_{t-1}]|] < \infty. \end{aligned} \tag{51}$$

As $\bar{g}_t$ are bounded from below ($\bar{f}_t$ is bounded from **(D)** and we easily show that $\bar{\epsilon}_t$ is bounded), we can apply Theorem A.1 from [10], that is a quasi-martingale convergence theorem originally found in [51]. It ensures that $(g_t(\theta_t))_{t \geq 1}$ converges almost surely to an integrable random variable $g^\star$, and that $\sum_{t=1}^{\infty} \mathbb{E}[|\mathbb{E}[\bar{g}_t(\theta_t) - \bar{g}_{t-1}(\theta_{t-1}) | \mathcal{F}_{t-1}]|] < \infty$ almost surely.

A.2.2 Almost sure convergence of $\bar{f}(\theta_t)$

We rewrite the second inequality of (46), adding $\bar{\epsilon}_t$ on both sides:

$$\begin{aligned} 0 & \leq w_t \big(\bar{g}_{t-1}(\theta_{t-1}) - \bar{f}_{t-1}(\theta_{t-1}) + \bar{\epsilon}_{t-1} \big) \\ & \leq w_t \big(g_t(\theta_{t-1}) - f_t(\theta_{t-1}) \big) + w_t \big(f_t(\theta_{t-1}) - \bar{f}_{t-1}(\theta_{t-1}) \big) \\ & \quad + \big(\bar{g}_{t-1}(\theta_{t-1}) - \bar{g}_t(\theta_t) \big) + w_t \bar{\epsilon}_{t-1} \\ & \leq w_t \big(f_t(\theta_{t-1}) - \bar{f}_{t-1}(\theta_{t-1}) \big) + \big(\bar{g}_{t-1}(\theta_{t-1}) - \bar{g}_t(\theta_t) \big) \\ & \quad + w_t (\epsilon_t + \bar{\epsilon}_{t-1}), \end{aligned} \tag{52}$$

where the left side bound has been obtained in the last paragraph by induction and the right side bound arises from the definition of ϵ_t. Taking the expectation of (52) conditioned on $\mathcal{F}_{t-1}$, almost surely,

$$\begin{aligned} 0 \leq w_t & (f(\theta_{t-1}) - \bar{f}_{t-1}(\theta_{t-1})) \\ & - \mathbb{E}[\bar{g}_t(\theta_t) - \bar{g}_{t-1}(\theta_{t-1}) | \mathcal{F}_{t-1}] + w_t (\bar{\epsilon}_{t-1} + \mathbb{E}[\epsilon_t | \mathcal{F}_{t-1}]), \end{aligned} \tag{53}$$

We separately study the three terms of the previous upper bound. The first two terms can undergo the same analysis as in [10]. First, almost sure convergence of $\sum_{t=1}^{\infty} \mathbb{E}\big[|\mathbb{E}[\bar{g}_t(\theta_t) - \bar{g}_{t-1}(\theta_{t-1}) | \mathcal{F}_{t-1}]|\big]$ implies that $\mathbb{E}\big[\bar{g}_t(\theta_t) - \bar{g}_{t-1}(\theta_{t-1}) | \mathcal{F}_{t-1}\big]$ is the summand of an almost surely converging sum. Second, $w_t\big(f(\theta_{t-1}) - \bar{f}_{t-1}(\theta_{t-1})\big)$ is the summand of an absolutely converging sum with probability one, less it would contradict (48). To bound the third term, we have once more to control the perturbation introduced by $(\epsilon_t)_t$. We have $\sum_{t=1}^{\infty} w_t \bar{\epsilon}_{t-1} + w_t \mathbb{E}[\epsilon_t | \mathcal{F}_{t-1}] < \infty$ almost surely, otherwise Fubini's theorem would invalidate (50).

As the three terms are the summand of absolutely converging sums, the positive term $w_t(\bar{g}_{t-1}(\theta_{t-1}) - \bar{f}_{t-1}(\theta_{t-1}) + \bar{\epsilon}_{t-1})$ is the summand of an almost surely convergent sum. This is not enough to prove that $\bar{h}_t(\theta_t) \triangleq \bar{g}_t(\theta_t) - \bar{f}_t(\theta_t) \to_\infty 0$, hence we follow [10] and make use of its Lemma A.6. We define $X_t \triangleq \bar{h}_{t-1}(\theta_{t-1}) + \bar{\epsilon}_{t-1}$. As **(H)** holds, we use Lemma 3, which ensures that $(\bar{h}_t)_{t \geq 1}$ are uniformly

R'-Lipschitz and $\|\theta_t - \theta_{t-1}\|_2 = \mathcal{O}(w_t)$. Hence,

$$\begin{aligned}
&|X_{t+1} - X_t| \leq |\bar{h}_t(\theta_t) - \bar{h}_{t-1}(\theta_{t-1})| + |\bar{\epsilon}_t - \bar{\epsilon}_{t-1}| \\
&\leq R'\|\theta_t - \theta_{t-1}\|_2 + |\bar{\epsilon}_t - \bar{\epsilon}_{t-1}|, \quad \text{as } \bar{h}_t \text{ is } R'\text{-Lipschitz} \\
&\leq \mathcal{O}(w_t) + |\bar{\epsilon}_t - \bar{\epsilon}_{t-1}|, \quad \text{as } \|\theta_t - \theta_{t-1}\|_2 = \mathcal{O}(w_t)
\end{aligned} \tag{54}$$

From assumption **(H)**, $(\epsilon_t)_t$ and $(\bar{\epsilon}_t)_t$ are bounded. Therefore $|\bar{\epsilon}_t - \bar{\epsilon}_{t-1}| \leq w_t(|\epsilon_t| + |\bar{\epsilon}_{t-1}|) \in \mathcal{O}(w_t)$ and hence

$$|X_{t+1} - X_t| \leq \mathcal{O}(w_t). \tag{55}$$

Lemma A.6 from [10] then ensures that X_t converges to zero with probability one. Assumption **(H)** ensures that $\epsilon_t \to_\infty 0$ almost surely, from which we can easily deduce $\bar{\epsilon}_t \to_\infty 0$ almost surely. Therefore $\bar{h}_t(\theta_t) \to 0$ with probability one and $(\bar{f}_t(\theta_t))_{t\geq 1}$ converges almost surely to $g^\star$.

A.2.3 Almost sure convergence of $\bar{f}(\theta_t)$

Lemma B.7 of [10], based on empirical process theory [33], ensures that $\bar{f}_t$ uniformly converges to $\bar{f}$. Therefore, $(\bar{f}(\theta_t))_{t\geq 1}$ converges almost surely to $g^\star$.

A.2.4 Asymptotic stationary point condition

Preliminary to the final result, we establish the asymptotic stationary point condition (57) as in [10]. This requires to adapt the original proof to take into account the errors in surrogate computation and minimization. We set $\alpha > 0$. By definition, $\nabla\bar{h}_t$ is L-Lipschitz over $\mathbb{R}^K$. Following the same computation as in (34), we obtain, for all $\alpha > 0$,

$$\|\nabla\bar{h}_t(\theta_t)\|_2 \leq \frac{2}{\alpha}\bar{\epsilon}_t + \frac{L\alpha}{2}, \tag{56}$$

where we use $|\bar{h}_t(\theta)| \leq \bar{\epsilon}_t$ for all $\theta \in \mathbb{R}^K$. As $\bar{\epsilon}_t \to 0$ and the inequality (56) is true for all α, $\|\nabla\bar{h}_t(\theta_t)\|_2 \to_\infty 0$ almost surely. From the strong convexity of $\bar{g}_t$ and Lemma 3, $\|\theta_t - \theta_t^\star\|_2$ converges to zero, which ensures

$$\|\nabla\bar{h}_t(\theta_t^\star)\|_2 \leq \|\bar{\nabla}h_t(\theta_t)\|_2 + L\|\theta_t - \theta_t^\star\|_2 \to_\infty 0. \tag{57}$$

A.2.5 Parametrized surrogates

We use assumption **(F)** to finally prove the property, adapting the proof of Proposition 3.4 in [10]. We first recall the derivations of [10] for obtaining (58) We define $(\boldsymbol{\kappa}_t)_t$ such that $\bar{g}_t = g_{\boldsymbol{\kappa}_t}$ for all $t > 0$. We assume that θ_∞ is a limit point of $(\theta_t)_t$. As Θ is compact, there exists an increasing sequence $(t_k)_k$ such that $(\theta_{t_k})_k$ converges toward θ_∞. As $\mathcal{K}$ is compact, a converging subsequence of $(\boldsymbol{\kappa}_{t_k})_k$ can be extracted, that converges towards $\boldsymbol{\kappa}_\infty \in \mathcal{K}$. From the sake of simplicity, we drop subindices and assume without loss of generality that $\theta_t \to \theta_\infty$ and $\boldsymbol{\kappa}_t \to \boldsymbol{\kappa}_\infty$. From the compact parametrization assumption, we easily show that $(\bar{g}_{\boldsymbol{\kappa}_t})_t$ uniformly converges towards $\bar{g}_\infty \triangleq \bar{g}_{\boldsymbol{\kappa}_\infty}$. Then, defining $\bar{h}_\infty = \bar{g}_\infty - \bar{f}$, for all $\theta \in \Theta$,

$$\nabla\bar{f}(\theta_\infty, \theta - \theta_\infty) = \nabla\bar{g}_\infty(\theta_\infty, \theta - \theta_\infty) - \nabla\bar{h}_\infty(\theta_\infty, \theta - \theta_\infty) \tag{58}$$

We first show that $\nabla\bar{f}(\theta_\infty, \theta - \theta_\infty) \geq 0$ for all $\theta \in \Theta$. We consider the sequence $(\theta_t^\star)_t$. From Lemma 3, $\|\theta_t - \theta_t^\star\|_2 \to 0$, which implies $\theta_t^\star \to \theta_\infty$. $\bar{g}_t$ converges uniformly towards $\bar{g}_\infty$, which implies $(\bar{g}_t(\theta_t^\star))_t \to \bar{g}_\infty(\theta_\infty)$. Furthermore, as $\theta_t^\star$ minimizes $\bar{g}_t$, for all $t > 0$ and $\theta \in \Theta$, $\bar{g}_t(\theta_t^\star) \leq \bar{g}_t(\theta)$. This implies $\bar{g}_\infty(\theta_\infty) \leq \inf_{\theta\in\Theta} \bar{g}_\infty(\theta)$ by taking the limit for $t \to \infty$. Therefore θ_∞ is the minimizer of $\bar{g}_\infty$ and thus $\nabla\bar{g}_\infty(\theta_\infty, \theta - \theta_\infty) \geq 0$.

Adapting [10], we perform the first-order expansion of $\bar{h}_t$ around $\theta_t^\star$ (instead of θ_t in the original proof) and show that $\nabla\bar{h}_\infty(\theta_\infty, \theta - \theta_\infty) = 0$, as $\bar{h}_t$ differentiable, $\|\nabla\bar{h}_t(\theta_t^\star)\|_2 \to 0$ and $\theta_t^\star \to \theta_\infty$. This is sufficient to conclude. □

A.3 Convergence of SOMF — Proof of Proposition 1

Proof of Proposition 1. From assumption **(D)**, $(x_t)_t$ is ℓ_2-bounded by a constant X. With assumption **(A)**, it implies that $(\boldsymbol{\alpha}_t)_t$ is ℓ_2-bounded by a constant A. This is enough to show that $(g_t)_t$ and $(\theta_t)_t$ meet basic assumptions **(C)**–**(F)**. Assumption **(G)** immediately implies **(B)**. It remains to show that $(g_t)_t$ and $(\theta_t)_t$ meet the assumptions **(H)** and **(I)**. This will allow to cast SOMF as an instance of SAMM and conclude.

A.3.1 The computation of $\mathbf{D}_t$ verifies (I)

We define $\mathbf{D}_t^\star = \operatorname{argmin}_{\mathbf{D}\in\mathcal{C}} \bar{g}_t(\mathbf{D})$. We show that performing subsampled block coordinate descent on $\bar{g}_t$ is sufficient to meet assumption **(I)**, where $\theta_t = \mathbf{D}_t$. We separately analyse the exceptional case where no subsampling is done and the general case.

First, with small but non-zero probability, $\mathbf{M}_t = \mathbf{I}_p$ and Alg. 4 performs a single pass of simple block coordinate descent on $\bar{g}_t$. In this case, as $\bar{g}_t$ is strongly convex from **(A)**, [52, 37] ensures that the sub-optimality decreases at least of factor $1-\mu$ with a single pass of block coordinate descent, where $\mu > 0$ is a constant independent of t. We provide an explicit μ in Appendix B.

In the general case, the function value decreases deterministically at each minimization step: $\bar{g}_t(\mathbf{D}_t) \leq \bar{g}_t(\mathbf{D}_{t-1})$. As a consequence, $\mathbb{E}[\bar{g}_t(\mathbf{D}_t)|\mathcal{F}_{t-\frac{1}{2}}, \mathbf{M}_t \neq \mathbf{I}_p] \leq \bar{g}_t(\mathbf{D}_{t-1})$. Furthermore, $\bar{g}_t$ and hence $\bar{g}_t(\mathbf{D}_t^\star)$ are deterministic with respect to $\mathcal{F}_{t-\frac{1}{2}}$, which implies $\mathbb{E}[\bar{g}_t(\mathbf{D}_t^\star)|\mathcal{F}_{t-\frac{1}{2}}, \mathbf{M}_t \neq \mathbf{I}_p] = \bar{g}_t(\mathbf{D}_t^\star)$. Defining $d \triangleq \mathbb{P}[\mathbf{M}_t = \mathbf{I}_p]$, we split the sub-optimality expectation and combine the analysis of both cases:

$$
\begin{aligned}
&\mathbb{E}[\bar{g}_t(\mathbf{D}_t) - \bar{g}_t(\mathbf{D}_t^\star)|\mathcal{F}_{t-\frac{1}{2}}] \\
&= d\mathbb{E}[\bar{g}_t(\mathbf{D}_t) - \bar{g}_t(\mathbf{D}_t^\star)|\mathcal{F}_{t-\frac{1}{2}}, \mathbf{M}_t = \mathbf{I}_p] \\
&\quad + (1-d)\mathbb{E}[\bar{g}_t(\mathbf{D}_t) - \bar{g}_t(\mathbf{D}_t^\star)|\mathcal{F}_{t-\frac{1}{2}}, \mathbf{M}_t \neq \mathbf{I}_p] \\
&\leq \big(d(1-\mu) + (1-d)\big)(\bar{g}_t(\mathbf{D}_{t-1}) - \bar{g}_t(\mathbf{D}_t^\star)) \\
&= \big(1 - d\mu\big)(\bar{g}_t(\mathbf{D}_{t-1}) - \bar{g}_t(\mathbf{D}_t^\star)).
\end{aligned} \tag{59}
$$

A.3.2 The surrogates $(g_t)_t$ verify (H)

We define $g_t^\star \in \mathcal{S}_{\rho,L}(f_t, \mathbf{D}_{t-1})$ the surrogate used in OMF at iteration t, which depends on the *exact* computation of $\boldsymbol{\alpha}_t^\star$, while the surrogate g_t used in SOMF relies on approximated $\boldsymbol{\alpha}_t$. Formally, using the loss function $\ell(\boldsymbol{\alpha}, \mathbf{G}, \boldsymbol{\beta}) \triangleq \frac{1}{2}\boldsymbol{\alpha}^\top\mathbf{G}\boldsymbol{\alpha} - \boldsymbol{\alpha}^\top\boldsymbol{\beta} + \lambda\Omega(\boldsymbol{\alpha})$, we recall the definitions

$$
\begin{aligned}
\boldsymbol{\alpha}_t^\star \triangleq \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^k} \ell(\boldsymbol{\alpha}, \mathbf{G}_t^\star, \boldsymbol{\beta}_t^\star),\ \boldsymbol{\alpha}_t \triangleq \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^k} \ell(\boldsymbol{\alpha}, \mathbf{G}_t, \boldsymbol{\beta}_t), \\
g_t^\star(\mathbf{D}) \triangleq \ell(\boldsymbol{\alpha}_t^\star, \mathbf{D}^\top\mathbf{D}, \mathbf{D}^\top\mathbf{x}_t),\ g_t(\mathbf{D}) \triangleq \ell(\boldsymbol{\alpha}_t, \mathbf{D}^\top\mathbf{D}, \mathbf{D}^\top\mathbf{x}_t).
\end{aligned} \tag{60}
$$

The matrices $\mathbf{G}_t^\star$, $\boldsymbol{\beta}_t^\star$ are defined in (21) and $\mathbf{G}_t$, $\boldsymbol{\beta}_t$ in either the update rules (b) or (c). We define $\epsilon_t \triangleq \|g_t^\star - g_t\|_\infty$ to be the ℓ_∞ difference between the approximate surrogate of SOMF and the exact surrogate of OMF, as illustrated in Figure 2. By definition, $g_t \in \mathcal{T}_{\rho,L}(f_t, \theta_{t-1}, \epsilon_t)$. We first show that ϵ_t can be bounded by the Froebenius distance between the approximate parameters $\mathbf{G}_t$, $\boldsymbol{\beta}_t$ and the exact parameters $\mathbf{G}_t^\star, \boldsymbol{\beta}_t^\star$. Using Cauchy-Schwartz inequality, we first show that there exists a constant $C' > 0$ such that for all $\mathbf{D} \in \mathcal{C}$,

$$
|g_t(\mathbf{D}) - g_t^\star(\mathbf{D})| \leq C'\|\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^*\|_2. \tag{61}
$$

Then, we show that the distance $\|\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^*\|_2$ can itself be bounded: there exists $C'' > 0$ constant such that

$$
\|\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^\star\|_2 \leq C''(\|\mathbf{G}_t^\star - \mathbf{G}_t\|_F + \|\boldsymbol{\beta}_t^\star - \boldsymbol{\beta}_t\|_2). \tag{62}
$$

We combine both equations and take the supremum over $\mathbf{D} \in \mathcal{C}$, yielding

$$
\epsilon_t \leq C(\|\mathbf{G}_t^\star - \mathbf{G}_t\|_F + \|\boldsymbol{\beta}_t^\star - \boldsymbol{\beta}_t\|_2), \tag{63}
$$

where C is constant. Detailed derivation of (61) to (63) relies on assumption **(A)** and are reported in Appendix B.

In a second step, we show that $\|\mathbf{G}_t^\star - \mathbf{G}_t\|_F$ and $\|\boldsymbol{\beta}_t^\star - \boldsymbol{\beta}_t\|_2$ vanish almost surely, sufficiently fast. We focus on bounding $\|\boldsymbol{\beta}_t - \boldsymbol{\beta}_t^\star\|_2$ and proceed similarly for $\|\mathbf{G}_t - \mathbf{G}_t^\star\|_2$ when the update rules (b) are used. For $t > 0$, we write $i \triangleq i_t$. Then

$$\boldsymbol{\beta}_t \triangleq \boldsymbol{\beta}_t^{(i)} = \sum_{s \leq t, \mathbf{x}_s = \mathbf{x}^{(i)}} \gamma_{s,t}^{(i)} \mathbf{D}_{s-1}^\top \mathbf{M}_s \mathbf{x}^{(i)},$$

where $\gamma_{s,t}^{(i)} = \gamma_{c_t^{(i)}} \prod_{s<t, \mathbf{x}_s = \mathbf{x}^{(i)}} (1 - \gamma_{c_s^{(i)}})$ and $c_t^{(i)} = \big|\big\{s \leq t, \mathbf{x}_s = \mathbf{x}^{(i)}\big\}\big|$. We can then decompose $\boldsymbol{\beta}_t - \boldsymbol{\beta}_t^\star$ as

$$\begin{aligned}\boldsymbol{\beta}_t - \boldsymbol{\beta}_t^\star = &\sum_{s \leq t, \mathbf{x}_s = \mathbf{x}_t = \mathbf{x}^{(i)}} \gamma_{s,t}^{(i)} (\mathbf{D}_{s-1} - \mathbf{D}_{t-1})^\top \mathbf{M}_s \mathbf{x}^{(i)} \\ &+ \mathbf{D}_{t-1}^\top \Big(\sum_{s \leq t, \mathbf{x}_s = \mathbf{x}^{i)}} \gamma_{s,t}^{(i)} \mathbf{M}_s - \mathbf{I} \Big) \mathbf{x}^{(i)}. \end{aligned} \tag{64}$$

The latter equation is composed of two terms: the first one captures the approximation made by using old dictionaries in the computation of $(\boldsymbol{\beta}_t)_t$, while the second captures how the masking effect is averaged out as the number of epochs increases. Assumption **(B)** allows to bound both terms at the same time. Setting $\eta \triangleq \frac{1}{2} \min\big(v - \frac{3}{4}, (3u-2) - v\big) > 0$, a tedious but elementary derivation indeed shows $\mathbb{E}[\|\boldsymbol{\beta}_t - \boldsymbol{\beta}_t^\star\|_2] \in \mathcal{O}(t^{2(u-1)-\eta})$ and $\epsilon_t \to 0$ almost surely — see Appendix B. The SOMF algorithm therefore meets assumption **(H)** and is a convergent SAMM algorithm. Proposition 1 follows. □

B Algebraic details

B.1 Proof of Lemma 1

Proof. We first focus on the deterministic case. Assume that $(x_t)_t$ is not bounded. Then there exists a subsequence of $(x_t)_t$ that diverges towards $+\infty$. We assume without loss of generality that $(x_t)_t \to \infty$. Then, $x_t + x_{t-1} \to \infty$ and for all $\epsilon > 0$, using the asymptotic bounds on u, there exists $t_1 \geq t_0$ such that

$$\begin{aligned} &\forall t \geq t_1,\ x_t \leq \alpha x_{t-1} + \epsilon(x_t + x_{t-1}) \\ \text{and therefore} \quad &x_t \leq \frac{\alpha+\epsilon}{1-\epsilon} x_{t-1}. \end{aligned} \tag{65}$$

Setting ϵ small enough, we obtain that x_t is bounded by a geometrically decreasing sequence after t_1, and converges to 0, which contradicts our hypothesis. This is enough to conclude.

In the random case, we consider a realization of $(X_t)_t$ that is not bounded, and assumes without loss of generality that it diverges to $+\infty$. Following the reasoning above, there exists $\beta < 1$, $t_1 > 0$, such that for all $t > t_1$, $\mathbb{E}[X_t|\mathcal{F}_{t'}] \leq \beta X_{t-1}$, where $\mathcal{F}_{t-1} \subseteq \mathcal{F}_{t'} \subseteq \mathcal{F}_t$. Taking the expectation conditioned on $\mathcal{F}_{t-1}$, $\mathbb{E}[X_t|\mathcal{F}_{t-1}] \leq \beta X_{t-1}$, as X_{t-1} is deterministic conditioned on $\mathcal{F}_{t-1}$. Therefore X_t is a supermartingale beyond a certain time. As $\mathbb{E}[X_t] < \infty$, Doob's forward convergence lemma on discrete martingales [53] ensures that $(X_t)_t$ converges almost surely. Therefore the event $\{(X_t)_t \text{ is not bounded}\}$ cannot happen on a set with non-zero probability, less it would lead to a contradiction. The lemma follows. □

B.2 Taylor's inequality for L-Lipschitz continuous functions

This inequality is useful in the demonstration of Lemma 2 and Proposition 3. Let $f : \Theta \subset \mathbb{R}^K \to \mathbb{R}$ be a function with L-Lipschitz gradient. That is, for all $\theta, \theta' \in \Theta$, $\|\nabla f(\theta) - \nabla f(\theta')\|_2 \leq L\|\theta - \theta'\|_2$. Then, for all $\theta, \theta' \in \Theta$,

$$f(\theta') \leq f(\theta) + \nabla f(\theta)^\top(\theta' - \theta) + \frac{L}{2}\|\theta - \theta'\|_2^2. \tag{66}$$

B.3 Lemma 3: Detailed control of D_t in (44)

Injecting (40) and (42) in (43), we obtain

$$\begin{aligned} &\tilde{D}_t \leq (1-\mu)\tilde{D}_{t-1}\frac{w_{t-1}^2}{w_t^2} + u(\tilde{D}_t, \tilde{D}_{t-1}), \quad \text{where} \\ &u(\tilde{D}_t, \tilde{D}_{t-1}) \triangleq (1-\mu)\tilde{Q}\left(\sqrt{3(\tilde{D}_t + \tilde{D}_{t-1}\frac{w_{t-1}^2}{w_t^2}) + \tilde{Q}} + \sqrt{\tilde{D}_t}\right). \end{aligned} \tag{67}$$

From assumption **(G)**, $\frac{w_{t-1}^2}{w_t^2} \to 1$, and we have, from elementary comparisons, that $u(\tilde{D}_t, \tilde{D}_{t-1}) \in o(\tilde{D}_t + \tilde{D}_{t-1})$ if $D_t \to \infty$. Using the determistictic result of Lemma 1, this ensures that $\tilde{D}_t$ is bounded.

B.4 Detailed derivations in the proof of Proposition 1

Let us first exhibit a scaler $\mu > 0$ independent of t, for which **(I)** is met

B.4.1 Geometric rate for single pass subsampled block coordinate descent

. For $\mathbf{D}^{(j)} \in \mathbb{R}^{p\times k}$ any matrix with non-zero j-th column $\mathbf{d}^{(j)}$ and zero elsewhere

$$\nabla\bar{g}_t(\mathbf{D} + \mathbf{D}^{(j)}) - \nabla\bar{g}_t(\mathbf{D} + \mathbf{D}^{(j)}) = \bar{\mathbf{C}}_t[j,j]\mathbf{d}^{(j)} \tag{68}$$

and hence $\bar{g}_t$ gradient has *component Lipschitz constant* $L_j = \bar{\mathbf{C}}_t[j,j]$ for component j, as already noted in [15]. Using [37] terminology, $\nabla\bar{g}_t$ has *coordinate Lipschitz constant* $L_{\max} \triangleq \max_{0\leq j<k} \bar{\mathbf{C}}_t[j,j] \leq$

$\max_{t>0,0\leq j<k} \boldsymbol{\alpha}_t[j]^2 \leq A^2$, as $(\boldsymbol{\alpha}_t)_t$ is bounded from **(A)**. As a consequence, $\bar{g}_t$ gradient is also L-Lipschitz continuous, where [37] note that $L \leq \sqrt{k}L_{\max}$. Moreover, $\bar{g}_t$ is strongly convex with strong convexity modulus $\rho > 0$ by hypothesis **(A)**. Then, [52] ensures that after one cycle over the k blocks

$$\begin{aligned}\mathbb{E}[\bar{g}_t(\mathbf{D}_t) - \bar{g}_t(\mathbf{D}_t^\star)|\mathcal{F}_{t-1}, \mathbf{M}_t = \mathbf{I}_p] &\leq \big(1 - \frac{\rho}{2L_{\max}(1 + kL^2/L_{\max}^2)}\big)(\bar{g}_t(\mathbf{D}_{t-1}) - \bar{g}_t(\mathbf{D}_t^\star)) \\ &\leq \big(1 - \mu\big)(\bar{g}_t(\mathbf{D}_{t-1}) - \bar{g}_t(\mathbf{D}_t^\star)) \quad \text{where} \quad \mu \triangleq \frac{\rho}{2A^2(1+k^2)} \end{aligned} \tag{69}$$

B.4.2 Controling ϵ_t from $(\mathbf{G}_t, \boldsymbol{\beta}_t), (\mathbf{G}_t^\star, \boldsymbol{\beta}_t^\star)$ — Equations 61–62

We detail the derivations that are required to show that **(H)** is met in the proof of SOMF convergence. We first show that $(\boldsymbol{\alpha}_t)_t$ is bounded. We choose $D > 0$ such that $\|\mathbf{d}^{(j)}\|_2 \leq D$ for all $j \in [k]$ and $\mathbf{D} \in \mathcal{C}$, and X such that $\|\mathbf{x}\|_2 \leq X$ for all $\mathbf{x} \in \mathcal{X}$. From assumption **(A)**, using the second-order growth condition, for all $t > 0$,

$$\begin{aligned} \frac{\rho}{2}\|\boldsymbol{\alpha}_t - 0\|_2^2 &\leq \lambda\Omega(0) - (\frac{1}{2}\boldsymbol{\alpha}_t^\top \mathbf{G}_t\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^\top\boldsymbol{\beta}_t + \lambda\Omega(\boldsymbol{\alpha}_t) \\ \frac{\rho}{2}\|\boldsymbol{\alpha}_t\|_2^2 + \frac{1}{2}\boldsymbol{\alpha}_t^\top \mathbf{G}_t\boldsymbol{\alpha}_t &\leq 0 + \|\boldsymbol{\alpha}_t\|_2\|\boldsymbol{\beta}_t\|_2, \quad \text{hence} \\ \rho\|\boldsymbol{\alpha}_t\|_2^2 &\leq \sqrt{k}rDX\|\boldsymbol{\alpha}_t\|_2, \quad \text{and therefore} \quad \|\boldsymbol{\alpha}_t\|_2 \leq \frac{\sqrt{k}rDX}{\rho} \triangleq A. \end{aligned} \tag{70}$$

We have successively used the fact that $\Omega(0) = 0$, $\Omega(\boldsymbol{\alpha}_t) \geq 0$, and $\|\boldsymbol{\beta}_t\|_2 \leq \sqrt{k}rDX$, which can be shown by a simple induction on the number of epochs. For all $t > 0$, from the definition of $\boldsymbol{\alpha}_t$ and $\boldsymbol{\alpha}_t^\star$, for all $\mathbf{D} \in \mathcal{C}$:

$$\begin{aligned} |g_t(\mathbf{D}) - g_t^\star(\mathbf{D})| &= \Big|\frac{1}{2}\mathrm{Tr}\,\mathbf{D}^\top\mathbf{D}(\boldsymbol{\alpha}_t\boldsymbol{\alpha}_t^\top - \boldsymbol{\alpha}_t^\star\boldsymbol{\alpha}_t^{\star\top}) - (\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^\star)^\top\mathbf{D}^\top\mathbf{x}_t\Big| \\ &\leq \frac{1}{2}\|\mathbf{D}^\top\mathbf{D}\|_F\|\boldsymbol{\alpha}_t\boldsymbol{\alpha}_t^\top - \boldsymbol{\alpha}_t^\star\boldsymbol{\alpha}_t^{\star\top}\|_F + \|\mathbf{D}\|_F\|\mathbf{x}_t\|_2\|\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^\star\|_2 \\ &\leq (kD^2A + \sqrt{k}DX)\|\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^\star\|_2, \end{aligned} \tag{71}$$

where we use Cauchy-Schwartz inequality and elementary bounds on the Froebenius norm for the first inequality, and use $\boldsymbol{\alpha}_t, \boldsymbol{\alpha}_t^\star \leq A$, $\mathbf{x}_t \leq X$ for all $t > 0$ and $\mathbf{d}^{(j)} \leq D$ for all $j \in [k]$ to obtain the second inequality, which is (61) in the main text.

We now turn to control $\|\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^\star\|_2$. We adapt the proof of Lemma B.6 from [36], that states the lipschitz continuity of the minimizers of some parametrized functions. By definition,

$$\boldsymbol{\alpha}_t^\star = \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^k} \ell(\boldsymbol{\alpha}, \mathbf{G}_t^\star, \boldsymbol{\beta}_t^\star) \qquad \boldsymbol{\alpha}_t = \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^k} \ell(\boldsymbol{\alpha}, \mathbf{G}_t, \boldsymbol{\beta}_t), \tag{72}$$

Assumption **(A)** ensures that $\mathbf{G}_t \succ \rho\mathbf{I}_k$, therefore we can write the second-order growth condition

$$\begin{aligned} \frac{\rho}{2}\|\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^\star\|_2^2 &\leq \ell(\boldsymbol{\alpha}_t, \mathbf{G}_t^\star, \boldsymbol{\beta}_t^\star) - \ell(\boldsymbol{\alpha}_t, \mathbf{G}_t, \boldsymbol{\beta}_t) \\ \frac{\rho}{2}\|\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^\star\|_2^2 &\leq \ell(\boldsymbol{\alpha}_t^\star, \mathbf{G}_t, \boldsymbol{\beta}_t) - \ell(\boldsymbol{\alpha}_t^\star, \mathbf{G}_t^\star, \boldsymbol{\beta}_t^\star), \quad \text{and therefore} \\ \rho\|\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^\star\|_2^2 &\leq p(\boldsymbol{\alpha}_t) - p(\boldsymbol{\alpha}_t^\star), \quad \text{where} \quad p(\boldsymbol{\alpha}) \triangleq \ell(\boldsymbol{\alpha}, \mathbf{G}_t, \boldsymbol{\beta}_t) - \ell(\boldsymbol{\alpha}_t, \mathbf{G}_t^\star, \boldsymbol{\beta}_t^\star). \end{aligned} \tag{73}$$

p takes a simple form and can differentiated with respect to $\boldsymbol{\alpha}$. For all $\boldsymbol{\alpha} \in \mathbb{R}^k$ such that $\|\boldsymbol{\alpha}\|_2 \leq A$,

$$\begin{aligned} p(\boldsymbol{\alpha}) &= \frac{1}{2}\boldsymbol{\alpha}^\top(\mathbf{G}_t - \mathbf{G}_t^\star)\boldsymbol{\alpha} - \boldsymbol{\alpha}^\top(\boldsymbol{\beta}_t - \boldsymbol{\beta}_t^\star) \\ \nabla p(\boldsymbol{\alpha}) &= (\mathbf{G}_t - \mathbf{G}_t^\star)\boldsymbol{\alpha} - (\boldsymbol{\beta}_t - \boldsymbol{\beta}_t^\star) \\ \|\nabla p(\boldsymbol{\alpha})\|_2 &\leq A\|\mathbf{G}_t - \mathbf{G}_t^\star\|_F + \|\boldsymbol{\beta}_t - \boldsymbol{\beta}_t^\star\|_2 \triangleq L \end{aligned} \tag{74}$$

Therefore p is L-Lipschitz on the ball of size A where $\boldsymbol{\alpha}_t$ and $\boldsymbol{\alpha}_t^\star$ live, and

$$\begin{aligned}\rho\|\boldsymbol{\alpha}_t-\boldsymbol{\alpha}_t^\star\|_2^2 &\leq L\|\boldsymbol{\alpha}_t-\boldsymbol{\alpha}_t^\star\|_2\\ \|\boldsymbol{\alpha}_t-\boldsymbol{\alpha}_t^\star\|_2 &\leq \frac{A}{\rho}\|\mathbf{G}_t-\mathbf{G}_t^\star\|_F+\frac{1}{\rho}\|\boldsymbol{\beta}_t-\boldsymbol{\beta}_t^\star\|_2,\end{aligned} \tag{75}$$

which is (62) in the main text. The bound (63) on ϵ_t immediately follows.

B.4.3 Bounding $\|\boldsymbol{\beta}_t-\boldsymbol{\beta}_t^\star\|_2$ in equation (64)

Taking the ℓ_2 norm in (64), we have $\|\boldsymbol{\beta}_t-\boldsymbol{\beta}_t^\star\|_2 \leq BL_t+CR_t$, where B and C are positive constants independent of t and we introduce the terms

$$L_t \triangleq \sum_{s\leq t,\mathbf{x}_s=\mathbf{x}_t=\mathbf{x}^{(i)}} \gamma_{s,t}^{(i)}\|\mathbf{D}_{s-1}-\mathbf{D}_{t-1}\|_F, \qquad R_t \triangleq \Big\|\big(\textstyle\sum_{s\leq t,\mathbf{x}_s=\mathbf{x}^{(i)}}\gamma_{s,t}^{(i)}\mathbf{M}_s\big)-\mathbf{I}\Big\|_F. \tag{76}$$

Conditioning on the sequence of drawn indices We recall that $(i_t)_t$ is the sequence of indices that are used to draw $(\mathbf{x}_t)_t$ from $\{\mathbf{x}^{(i)}\}_i$, namely such that $\mathbf{x}_t=\mathbf{x}^{(i_t)}$. $(i_t)_t$ is a sequence of i.i.d random variables, whose law is uniform in $[1,n]$. For each $i\in[n]$, we define the increasing sequence $(t_b^{(i)})_{b>0}$ that record the iterations at which sample (i) is drawn, *i.e.* such that $i_{t_b}=i$ for all $b>0$. For $t>0$, we recall that $c_t^{(i)}>0$ is the integer that counts the number of time sample (i) has appeared in the algorithm, *i.e.* $c_t^{(i)}=\max\{b>0, t_b^{(i)}\leq t\}$. These notations will help us understanding the behavior of $(L_t)_t$ and $(R_t)_t$.

Bounding R_t The right term R_t takes its value into sequences that are running average of masking matrices. Formally, $R_t=\|\bar{\mathbf{M}}_t^{(i_t)}-\mathbf{I}\|_F$, where we define for all $i\in[n]$,

$$\bar{\mathbf{M}}_t^{(i)} \triangleq \sum_{b=1}^{c_t^{(i)}}\gamma^{(i)}_{t_b^{(i)},t_c^{(i)}}\mathbf{M}_{t_b}, \text{ which follows } \begin{cases}\bar{\mathbf{M}}_t^{(i)} &= (1-\gamma_{c_t^{(i)}})\bar{\mathbf{M}}_{t-1}^{(i)}+\gamma_{c_t^{(i)}}\mathbf{M}_t \quad \text{if } i=i_t\\ \bar{\mathbf{M}}_t^{(i)} &= \mathbf{M}_{t-1}^{(i)} \quad \text{if } i\neq i_t\\ \bar{\mathbf{M}}_0^{(i)} &= 0 \quad \text{for all } i\in[n]\end{cases} \tag{77}$$

When sampling a sequence of indices $(i_s)_{s>0}$, the n random matrix sequences $[(\bar{\mathbf{M}}_t^{(i)})_{t\leq 0}]_{i\in[n]}$ follows the same probability law as the sampling is uniform. We therefore focus on controling $(\bar{\mathbf{M}}_t^{(0)})_t$. For simplicity, we write $c_t\triangleq c_t^{(0)}$. When $\mathbb{E}[\cdot]$ is the expectation over the sequence of indices $(i_s)_s$,

$$\begin{aligned}\mathbb{E}[\|\bar{\mathbf{M}}_t^{(0)}-\mathbf{I}\|_F]^2 &\leq \mathbb{E}\big[\sum_{j=1}^p(\bar{\mathbf{M}}_t^{(0)}[j,j]-1)\big] = p\mathbb{E}[(\bar{\mathbf{M}}_t^{(0)}[0,0]-1)]\\ &\leq C\,p(c_t)^{1/2}\gamma_{c_t} = C\,p(c_t)^{1/2-v}, \quad \text{where } C \text{ is a constant independent of } t.\end{aligned} \tag{78}$$

We have simply bounded the Froebenius norm by the ℓ_1 norm in the first inequality and used the fact that all coefficients $\mathbf{M}_t[j,j]$ follows the same Bernouilli law for all $t>0$, $j\in[p]$. We then used Lemma B.7 from [10] for the last inequality. This lemma applies as $\mathbf{M}_t[0,0]$ follows the recursion (77). It remains to take the expectation of (78), over all possible sampling trajectories $(i_s)_{s>0}$:

$$\begin{aligned}\mathbb{E}[R_t] &= \mathbb{E}\big[\mathbb{E}[R_t|(i_s)_s]\big] = \mathbb{E}\big[\mathbb{E}[\|\mathbf{M}_t^{(i_t)}-\mathbf{I}\|_F|(i_s)_s]\big] = \mathbb{E}\big[\mathbb{E}[\|\mathbf{M}_t^{(0)}-\mathbf{I}\|_F|(i_s)_s]\big] = \mathbb{E}[\|\mathbf{M}_t^{(0)}-\mathbf{I}\|_F]\\ &= Cp\mathbb{E}[(c_t)^{1/2-v}] \leq Cp\mathbb{E}[(c_t)^{2(u-1)-\eta}].\end{aligned} \tag{79}$$

The last inequality arises from the definition of $\eta\triangleq\frac{1}{2}\min\big(v-\frac{3}{4},(3u-2)-v\big)$, as follows. First, $\eta>0$ as $u>\frac{11}{12}$. Then, we successively have

$$\begin{aligned}&\frac{5}{2}-2u<\frac{2}{3}<\frac{3}{4}, \quad \text{as } u>\frac{11}{12}, \qquad v\geq\frac{3}{4}+2\eta>\frac{5}{2}-2u+2\eta,\\ &\frac{1}{2}-v<\frac{1}{2}-\frac{5}{2}+2u-2\eta=2(u-1)-2\eta<2(u-1)-\eta, \quad \text{which allows to conclude.}\end{aligned} \tag{80}$$

Lemma B.7 from [10] also ensures that $\mathbf{M}_t[0,0] \to 1$ almost surely when $t \to \infty$. Therefore $(\bar{\mathbf{M}}_t^{(0)} - \mathbf{I})_t$ converges towards 0 almost surely, given any sample sequence $(i_s)_s$. It thus converges almost surely when *all* random variables of the algorithm are considered. This is also true for $(\bar{\mathbf{M}}_t^{(i)} - \mathbf{I})_t$ for all $i \in [n]$ and hence for R_t.

Bounding L_t As above, we define n sequences $[(L_t^{(i)})_t]_{i\in[n]}$, such that $L_t = L_t^{(i_t)}$ for all $t > 0$. Namely,

$$L_t^{(i)} \triangleq \sum_{\substack{s\le t,\\ \mathbf{x}_s=\mathbf{x}_t=\mathbf{x}^{(i)}}} \gamma_{s,t}^{(i)} \|\mathbf{D}_{s-1} - \mathbf{D}_{t-1}\|_F = \sum_{b=1}^{c_t^{(i)}} \gamma^{(i)}_{t_b^{(i)}, t^{(i)}_{c_t^{(i)}}} \left\|\mathbf{D}_{t_b-1} - \mathbf{D}_{t_{c_t^{(i)}}-1}\right\|_F. \tag{81}$$

Once again, the sequences $\left[(L_t^{(i)})_t\right]_i$ all follows the same distribution when sampling over sequence of indices $(i_s)_s$. We thus focus on bounding $(L_t^{(0)})_t$. Once again, we drop the (0) superscripts in the right expression for simplicity. We set $\nu \triangleq 3u - 2 - \eta$. From assumption **(B)** and the definition of η, we have $v < \nu < 1$. We split the sum in two parts, around index $d_t \triangleq c_t - \lfloor (c_t)^\nu \rfloor$, where $\lfloor\cdot\rfloor$ takes the integer part of a real number. For simplicity, we write $d \triangleq d_t$ and $c \triangleq c_t$ in the following.

$$L_t^{(0)} = \sum_{b=1}^{c} \gamma_{t_b,t_c} \left\|\mathbf{D}_{t_b-1} - \mathbf{D}_{t_c-1}\right\|_F \le 2\sqrt{k}D \sum_{b=1}^{d} \gamma_{t_b,t_c} + \sum_{b=d+1}^{c} \gamma_{t_b,t} \sum_{s=t_b-1}^{t_c-1} w_s \triangleq 2\sqrt{k}D L_{t,1}^{(0)} + L_{t,2}^{(0)} \tag{82}$$

On the left side, we have bounded $\|\mathbf{D}_t\|_F$ by $\sqrt{k}D$, where D is defined in the previous section. The right part uses the bound on $\|\mathbf{D}_s - \mathbf{D}_t\|_F$ provided by Lemma 3, that applies here as **(I)** is met and (63) ensures that $(\|g_t - g_t^\star\|_\infty)_t$ is bounded.

We now study both $L_{t,1}^{(0)}$ and $L_{t,2}^{(0)}$. First, for all $t > 0$,

$$\begin{aligned} L_{t,1}^{(0)} \triangleq & \sum_{b=1}^{d} \gamma_{t_b,t_c} = \sum_{b=1}^{d} \gamma_b \prod_{p=b+1}^{c} (1-\gamma_p) \le \sum_{b=1}^{d} \gamma_b (1-\gamma_c)^{c-b} \\ & \le \frac{(1-\gamma_c)^{\lfloor c^\nu \rfloor}}{\gamma_c} \le c^v \exp\left(\log(1 - \frac{1}{c^v}) c^\nu\right) \le C' c^v \exp(c^{\nu-v}) \le C c^{2(u-1)-\eta} = C(c_t)^{2(u-1)-\eta}, \end{aligned} \tag{83}$$

where C and C' are constants independent of t. We have used $\nu > v$ for the third inequality, which ensures that $\exp\left(\log(1 - \frac{1}{c^v}) c^\nu\right) \in \mathcal{O}(c^{\nu-v})$. Basic asymptotic comparison provides the last inequality, as $c_t \to \infty$ almost surely and the right term decays exponentially in $(c_t)_t$, while the left decays polynomially. As a consequence, $L_{t,1}^{(0)} \to 0$ almost surely.

Secondly, the right term can be bounded as $(w_t)_t$ decays sufficiently rapidly. Indeed, as $\sum_{b=1}^{c} \gamma_{t_b,t} = 1$, we have

$$\begin{aligned} L_{t,2}^{(0)} \triangleq \sum_{b=d}^{c} \gamma_{t_b,t} \sum_{s=t_b-1}^{t_c-1} w_s & \le \max_{d\le b\le c} \Big(\sum_{s=t_b-1}^{t_c-1} w_s \Big) = \sum_{s=t_d-1}^{t_c-1} w_s \\ & \le w_{t_d}(t_c - t_d) = \frac{t_c - t_d}{(t_d)^u} = \frac{c_t - d_t}{(d_t)^u} \frac{t_c - t_d}{c_t - d_t} (\frac{d_t}{t_d})^u \end{aligned} \tag{84}$$

from elementary comparisons. First, we use the definition of ν to draw

$$\frac{c_t - d_t}{(d_t)^u} \le \frac{(c_t)^\nu}{(c_t)^u (1 - c_t^{\nu-1})^u} \le C (c_t)^{\nu-u} = C (c_t)^{2(u-1)-\eta}, \tag{85}$$

were we use the fast that $\eta - 1 < 0$. We note that for all $b > 0$, $t_{b+1} - t_b$ follows a geometric law of parameter $\frac{1}{n}$, and expectation n. Therefore, as $c - d \to \infty$ when $t \to 0$, from the strong law of large numbers and linearity of the expectation

$$\frac{t_c - t_d}{c - d} = \frac{1}{c-d} \sum_{b=d}^{c-1} t_{b+1} - t_b \to n, \qquad \frac{t_d}{d} = \frac{1}{d} \sum_{b=0}^{d-1} t_{b+1} - t_b \to n \quad \text{almost surely.} \tag{86}$$

As a consequence, $\frac{t_c-t_d}{c_t-d_t}(\frac{d_t}{t_d})^u \to n^{1-u}$ almost surely. This immediately shows $L_{t,2}^{(0)} \to 0$ and thus $L_t^{(0)} \to 0$ almost surely. As with R_t, this implies that $L_t \to 0$ almost surely and therefore

$$\|\boldsymbol{\beta}_t - \boldsymbol{\beta}_t^\star\|_2 \to 0 \quad \text{almost surely.} \tag{87}$$

Finally, from the dominated convergence theorem, $\mathbb{E}[\frac{t_c-t_d}{c_t-d_t}(\frac{d_t}{t_d})^u] \to n^{1-u}$ for $t \to \infty$. We can use Cauchy-Schartz inequality and write

$$\mathbb{E}[L_{t,2}^{(0)}] = \mathbb{E}[\frac{t_c - t_d}{(t_d)^u}] \leq \mathbb{E}[\frac{c_t - d_t}{(d_t)^u}]\mathbb{E}[\frac{t_c - t_d}{c_t - d_t}(\frac{d_t}{t_d})^u] \leq C'\mathbb{E}[\frac{c_t - d_t}{(d_t)^u}] \leq C\,C'\mathbb{E}[(c_t)^{2(u-1)-\eta}], \tag{88}$$

where C' is a constant independant of t. Then

$$\mathbb{E}[L_t] = \mathbb{E}\big[\mathbb{E}[L_t^{(i_t)}|(i_s)_s]\big] = \mathbb{E}\big[\mathbb{E}[L_t^{(0)}|(i_s)_s]\big] = \mathbb{E}[L_t^{(0)}] \leq 2\sqrt{k}D\mathbb{E}[L_{t,1}^{(0)}] + \mathbb{E}[L_{t,2}^{(0)}] \in \mathcal{O}((c_t)^{2(u-1)-\eta}). \tag{89}$$

Combined with (79), this shows that $\mathbb{E}[\|\boldsymbol{\beta}_t - \boldsymbol{\beta}_t^\star\|_2] \in \mathcal{O}((c_t)^{2(u-1)-\eta})$. As c_t follows a binomial distribution of parameter $(t, \frac{1}{n})$, $\frac{c_t}{t} \to \frac{1}{n}$ almost surely when $t \to 0$. Therefore $\mathbb{E}[(\frac{c_t}{t})^{2(u-1)-\eta})] \to n^{\eta-2(u-1)}$, and from Cauchy-Schwartz inequality,

$$\mathbb{E}[\|\boldsymbol{\beta}_t - \boldsymbol{\beta}_t^\star\|_2] \leq C\mathbb{E}[(\frac{c_t}{t})^{2(u-1)-\eta})]t^{2(u-1)-\eta} \in \mathcal{O}(t^{2(u-1)-\eta}). \tag{90}$$

We have reused the fact that converging sequences are bounded. This is enough to conclude.