
Pursuits in Structured Non-Convex Matrix Factorizations

Rajiv Khanna

UT Austin

Michael Tschannen

ETH Zurich

Martin Jaggi

ETH Zurich

RAJIVAK@UTEXAS.EDU

MICHAELT@NARI.EE.ETHZ.CH

JAGGI@INF.ETHZ.CH

Abstract

Efficiently representing real world data in a succinct and parsimonious manner is of central importance in many fields. We present a generalized greedy pursuit framework, allowing us to efficiently solve structured matrix factorization problems, where the factors are allowed to be from arbitrary sets of structured vectors. Such structure may include sparsity, non-negativeness, order, or a combination thereof. The algorithm approximates a given matrix by a linear combination of few rank-1 matrices, each factorized into an outer product of two vector atoms of the desired structure. For the non-convex subproblems of obtaining good rank-1 structured matrix atoms, we employ and analyze a general atomic power method. In addition to the above applications, we prove linear convergence for generalized pursuit variants in Hilbert spaces — for the task of approximation over the linear span of arbitrary dictionaries — which generalizes OMP and is useful beyond matrix problems. Our experiments on real datasets confirm both the efficiency and also the broad applicability of our framework in practice.

1. Introduction

Approximating a matrix using a structured low-rank matrix factorization is a cornerstone problem in a huge variety of data-driven applications. This problem can be seen as projecting a given matrix onto a linear combination of few rank-1 matrices, each of which being an outer product of two vectors, each from a structured set of vectors. Examples of such structure in the vectors can be sparsity, group sparsity, non-negativeness etc. The structure is generally encoded as a constraint on each of the two factors of the factorization problem. Even without imposing structure,

the rank-constrained problem is already NP-hard to solve in general¹. Instead of a rank constraint, convex relaxations are therefore typically applied. However, this involves giving up explicit control over the resulting rank. Nevertheless, there has been a strong body of research studying recovery and performance under several convex relaxations of rank-constrained problems (Candes and Recht, 2009; Candes and Tao, 2010; Toh and Yun, 2009; Pong et al., 2010).

In this paper, we take a different approach. We keep explicit control over the rank of the factorization, as well as the precise structure of the used (vector) factors, which we call atoms. Our approach is a greedy method adding one rank-1 atom per outer iteration, as in matrix variants of matching pursuit (Wang et al., 2014) as well as Frank-Wolfe algorithms on factorizations (Hazan, 2008; Jaggi and Sulovský, 2010; Dudík et al., 2012; Jaggi, 2013; Bach, 2013). By keeping the explicit low-rank factorization into the vector atoms at all times, we can study general algorithms and correction variants applying directly to the original non-convex problems.

Iteratively adding a rank-1 atom for structured matrix factorization falls into the purview of pursuit algorithms, which we here present in general form. Each update is obtained from a linear minimization oracle (LMO), which outputs the best rank-1 atom with respect to a linearized version of the objective. Each iteration hence increases the rank by 1, while improving the approximation quality. We will systematically study this tradeoff between rank and approximation quality, by providing convergence rates of the iterates to the best possible structured approximation of the given matrix.

To study the convergence rates for matrix pursuit over

¹ For example, least-squares matrix completion is NP hard even for a rank-1 factorization, as shown by (Gillis and Glineur, 2011).

structured sets, we adapt the convergence results for pursuit algorithms from the compressed sensing literature. While most of the existing work focuses on assessing the quality of k -greedy selection of atoms vis-a-vis a best possible k -atom selection, there is some work that discusses linear convergence of greedy pursuit approaches in an optimization sense (Davis et al., 1997; Mallat and Zhang, 1993; Gribonval and Vandergheynst, 2006; Blumensath and Davies, 2008; Jones, 1987; Dupé, 2015). However, they are not directly applicable to matrix pursuit for structured factorization, because they typically require the observed vector to lie within the span of the given dictionary (i.e., the structured sets) and make strong structural assumptions about the dictionary (e.g., incoherence, see Section 2.1). We show that even in the more general case when the observation is not in the span of dictionary, greedy pursuit in a Hilbert space converges linearly to the best possible approximation. In the language of functional analysis, the line of research of DeVore and Temlyakov (2014); Temlyakov (2014) is closest to this approach. Note that such results have a wider scope than just the applications to matrix pursuit.

The general nature of our convergence result allows its application to any set of atoms which induce an inner product – which in turn induces a distance function, that can be minimized by greedy pursuit. It is easy to show that the low rank structured matrix completion problem can be cast in this framework as well, and this yields an algorithm that works for any atomic vector set. For the specific case of atomic vector sets being unit 2-norm balls without any further structure, this setup was used by Wang et al. (2014), who showed linear convergence for matrix pursuit on 2-norm balls as vector atomic sets. This is a special case of our framework, because we show linear convergence with *any* compact vector atomic sets. We also present empirical results on real world datasets that show that this generalization is useful in practice.

The linear convergence of the matching pursuit in Hilbert spaces specifies the decay in reconstruction error in terms of number of calls made to the LMO. For the matrix pursuit algorithm, the linear problem being solved by the LMO itself may be a NP-hard, though efficient solutions are available in some cases (Bach et al., 2008; Recht et al., 2010). We also analyze the pursuit using only an approximate version of the LMO, and we show that the linear convergence rate is still maintained, but the decay is less sharp depending on the approximation quality of the LMO.

Related work: There exists a vast literature on structured matrix factorizations. For our cases, the most relevant are the lines of research with iterative rank-1 greedy approximations such as the Frank-Wolfe algorithm (Hazan, 2008; Jaggi and Sulovský, 2010; Dudík et al., 2012; Jaggi, 2013;

Bach, 2013). In the tensor case, a very similar approach has recently been investigated by Yang et al. (2015), but not for the case of structured factorizations like we do here. Their linear convergence result is also a special case of our more general convergence rate result in Hilbert spaces. Similarly, for specific atomic sets, there is a large body of literature, see, e.g., (Yuan and Zhang, 2013; Journée et al., 2010; Pappalopoulos et al., 2013) for Sparse PCA, (Sigg and Buhmann, 2008; Asteris et al., 2014) for sparse non-negative PCA, and references therein.

There is a significant amount of research on pursuit algorithms, even more so on one of its more commonly used flavors known as orthogonal matching pursuit. Davis et al. (1997) prove geometric convergence of matching pursuit and its orthogonal counterpart for finite dictionaries, while Mallat and Zhang (1993); Gribonval and Vandergheynst (2006) give convergence results for (quasi-)incoherent dictionaries in Hilbert spaces of finite or infinite dimension. However, all of these assume the observed vector to lie in the dictionary span, so that the goal is to exactly reconstruct it using as few atoms as possible rather than to approximate it using as few atoms as possible. For infinite-dimensional pursuit, Jones (1987) showed convergence without providing rates.

The matrix completion problem has gained significant interest recently, motivated by powerful applications in recommender systems (e.g. Netflix prize, Koren et al. (2009)), signal processing (robust PCA, Candes et al. (2011)), and most recently word-embeddings in NLP (Levy and Goldberg, 2014). Candes and Recht (2009); Recht (2009) and several subsequent works study the completion problem by convex optimization and provide bounds on exact matrix completion for random matrices. Jain et al. (2013) provide guarantees for low rank matrix completion by alternating minimization under incoherence. These works and several followups cast the matrix completion as minimization of the rank of the matrix (or a convex surrogate) under the constraint that the observed entries are reproduced exactly or approximately. A matrix pursuit view of the problem was taken by Wang et al. (2014) by adding rank-1 updates iteratively to decrease the reproduction error on the observed entries.

Contributions. Our key contributions are as follows:

- We devise and analyze a general algorithmic framework for structured matrix factorizations, where the factors are allowed to be from an arbitrary set of structured vectors. Our method only assumes a constrained linear minimization oracle (LMO), and can be seen as a special case of a more general class of pursuit algorithms, which we analyze in Section 2.
- We prove a linear convergence guarantee for general-

ized pursuit in Hilbert spaces for approximation over the linear span of arbitrary dictionaries, which generalizes OMP and is useful beyond matrix problems.

- For the non-convex rank-one factorization subproblems per iteration, we propose a general atomic power method, allowing to efficiently approximate the LMO for arbitrary structured sets of vector atoms.
- We improve efficiency of the resulting methods in terms of the required rank (number of atoms) by allowing corrective variants of the algorithm.
- Finally, we provide strong experimental results on real world datasets, confirming the efficiency and broad applicability of our framework in practice.

Notation. We represent vectors as small letter bolds, e.g., \mathbf{u} . Matrices are represented by capital bolds, e.g., \mathbf{X} , \mathbf{T} . Vector/matrix transposes are represented by superscript \top . Identity matrix is represented as \mathbf{I} . Sets are represented by calligraphic letters, e.g., \mathcal{S} . For a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and a set of index pairs Ω , \mathbf{A}_Ω stands for the matrix that is equal to \mathbf{A} at the entries indexed by Ω , and 0 elsewhere. Let $[d]$ be the set $\{1, 2, \dots, d\}$. Let $\text{conv}(\mathcal{S})$ be the convex hull of the set \mathcal{S} , and let $\text{lin}(\mathcal{S})$ denote the linear span of the elements in \mathcal{S} . $\mathbf{u} \otimes \mathbf{v}$ represents the rank-1 matrix given by the outer product $\mathbf{u}\mathbf{v}^\top$ of two vectors \mathbf{u}, \mathbf{v} . Analogously we write $\mathcal{A}_1 \otimes \mathcal{A}_2$ for the set of outer products between any pair of elements from two sets \mathcal{A}_1 and \mathcal{A}_2 respectively.

2. Generalized Pursuit in Hilbert Spaces

In this section, we develop a generalized pursuit algorithm for Hilbert spaces. Let \mathcal{H} be a Hilbert space with associated inner product $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{H}}, \forall \mathbf{x}, \mathbf{y} \in \mathcal{H}$. The inner product induces the norm $\|\mathbf{x}\|_{\mathcal{H}}^2 := \langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{H}}, \forall \mathbf{x} \in \mathcal{H}$, as well as the distance function $d_{\mathcal{H}}(\mathbf{x}, \mathbf{y}) := \|\mathbf{x} - \mathbf{y}\|_{\mathcal{H}}, \forall \mathbf{x}, \mathbf{y} \in \mathcal{H}$.

Let $\mathcal{S} \subset \mathcal{H}$ be a bounded set and let $f_{\mathcal{H}}: \mathcal{H} \rightarrow \mathbb{R}$. We would like to solve the optimization problem

$$\min_{\mathbf{x} \in \text{lin}(\mathcal{S})} f_{\mathcal{H}}(\mathbf{x}). \quad (1)$$

We write \mathbf{x}^* for a minimizer of (1). For any $\mathbf{y} \in \mathcal{H}$, and bounded set $\mathcal{S} \subset \mathcal{H}$, a linear minimization oracle (LMO) is defined as

$$\text{LMO}_{\mathcal{S}}(\mathbf{y}) := \arg \min_{\mathbf{x} \in \mathcal{S}} \langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{H}}.$$

Our generalized pursuit algorithm is presented as Algorithm 1. In each iteration r , the linearized objective at the previous iterate is minimized over the set \mathcal{S} (which is purpose of the LMO), in order to obtain the next atom \mathbf{z}_r to be added.

Algorithm 1 Generalized Pursuit (GP) in Hilbert Spaces

Require: $\mathbf{x}_0 \in \mathcal{S}, R, f_{\mathcal{H}}$

- 1: **for** $r = 1..R$ **do**
- 2: $\mathbf{z}_r := (\text{Approx-})\text{LMO}_{\mathcal{S}}(\nabla f_{\mathcal{H}}(\mathbf{x}_{r-1}))$
- 3: $\alpha := \min_{\alpha \in \mathbb{R}^r} f_{\mathcal{H}}(\sum_{i \leq r} \alpha_i \mathbf{z}_i)$
- 4: *Optional:* Correction of some/all \mathbf{z}_i .
- 5: Update iterate $\mathbf{x}_r := \sum_{i \leq r} \alpha_i \mathbf{z}_i$
- 6: **end for**

After that, we do allow for corrections of the weights of all previous atoms, as shown in line 3. In other words, we obtain the next iterate by minimizing $f_{\mathcal{H}}$ over the linear span of the current set of atoms. This idea is similar to the fully-corrective Frank-Wolfe algorithm and its away step variants (Lacoste-Julien and Jaggi, 2015), as well as orthogonal matching pursuit (OMP) (Chen et al., 1989), as we will discuss below. E.g. for distance objective functions $d_{\mathcal{H}}^2$ as of our interest here, this correction step is particularly easy and efficient, and boils down to simply solving a linear system of size $r \times r$.

A second type of (optional) additional correction is shown in line 4, and allows to change some of the actual atoms of the expansion, see e.g. Laue (2012). Both types of corrections have the same motivation, namely to obtain better objective cost while using the same (small) number of atoms r .

The total number of iterations R of Algorithm 1 controls the the tradeoff between approximation quality, i.e., how close $f_{\mathcal{H}}(\mathbf{x}_R)$ is to the optimum $f_{\mathcal{H}}(\mathbf{x}^*)$, and the “structuredness” of \mathbf{x}_R due to the fact that we only use R atoms from \mathcal{S} and through the structure of the atoms themselves (e.g., sparsity). If \mathcal{H} is an infinite dimensional Hilbert space, then f is assumed to be *Fréchet differentiable*.

Note that the main difference between generalized pursuit as in Algorithm 1 and Frank-Wolfe type algorithms (both relying on the same LMO) is that pursuit maintains its respective current iterates as a *linear* combination of the selected atoms, while in FW restricts to *convex* combinations of the atoms.

We next particularize Algorithm 1 to the least squares objective function $f_{\mathcal{H}}(\mathbf{x}) := \frac{1}{2}d_{\mathcal{H}}^2(\mathbf{x}, \mathbf{y})$ for fixed $\mathbf{y} \in \mathcal{H}$. As will be seen later, this variant of Algorithm 1 has many practical applications. The minimization in line 3 of Algorithm 1 hence amounts to orthogonal projection of \mathbf{y} to the linear span of $\{\mathbf{z}_i; i \leq r\}$, and $-\nabla f_{\mathcal{H}}(\mathbf{x}_r)$ writes as $\mathbf{r}_{r+1} := \mathbf{y} - \mathbf{x}_r$, henceforth referred to as the residual at iteration r . We thus recover a variant of the OMP algorithm (Chen et al., 1989). By minimizing only w.r.t. the new weight α_r in line 3 of Algorithm 1 we further recover a variant of the matching pursuit (MP) algorithm (Mallat

and Zhang, 1993)².

Our first main result characterizes the convergence of Algorithm 1 for $f_{\mathcal{H}}(\mathbf{x}) := \frac{1}{2}d_{\mathcal{H}}^2(\mathbf{x}, \mathbf{y})$.

Theorem 1. *Let $f_{\mathcal{H}}(\mathbf{x}) := \frac{1}{2}d_{\mathcal{H}}^2(\mathbf{x}, \mathbf{y})$ for $\mathbf{y} \in \mathcal{H}$. For every $\mathbf{x}_0 \in \mathcal{S}$, Algorithm 1 converges linearly to \mathbf{x}^* .*

Discussion. The linear convergence guarantees $f_{\mathcal{H}}(\mathbf{x}_r)$ to be within ϵ of the optimum value $f(\mathbf{x}^*)$ after $O(\log 1/\epsilon)$ iterations. Moreover, since the atoms selected across iterations are linearly independent, the maximum number of iterations in the finite-dimensional case is equal to the dimension of \mathcal{H} .

Algorithm 1 allows for an *inexact* LMO. In particular for our main interest of matrix factorization problems here, an exact LMO is often very costly, while approximate versions can be much more efficient. We refer to the supplementary material for the proof that the linear convergence rate still holds for approximate LMO, in terms of multiplicative error.

2.1. Relationship with coherence-based rates

Our presented convergence analysis (which holds for infinite sets of atoms) can be seen as a generalization of existing coherence-based rates which were obtained for finite dictionaries $\mathcal{S} = \{\mathbf{s}_i; i \in [n]\} \subset \mathcal{H}$. Throughout this section, we assume that the atoms are normalized according to $\|\mathbf{s}\|_{\mathcal{H}} = 1, \forall \mathbf{s} \in \mathcal{S}$, and that \mathcal{S} is symmetric, i.e., $\mathbf{s} \in \mathcal{S}$ implies $-\mathbf{s} \in \mathcal{S}$.

In the general case, we assume \mathbf{y} to lie in \mathcal{H} (not necessarily in \mathcal{S}). In the sequel, we will restrict \mathbf{y} to lie in $\text{lin}(\mathcal{S})$ in order to recover a known coherence-based convergence result for OMP for finite dictionaries. Our results are based on the cumulative coherence function.

Definition 2 (Cumulative coherence function (Tropp, 2004)). *Let $\mathcal{I} \subset [n]$ be an index set. For an integer m , cumulative coherence function is defined as:*

$$\mu(m) := \max_{|\mathcal{I}|=m} \max_{k \in [n] \setminus \mathcal{I}} \sum_{i \in \mathcal{I}} |\langle \mathbf{s}_k, \mathbf{s}_i \rangle_{\mathcal{H}}|$$

Note that $\mu(m) \leq m\mu(1)$, and $\mu(1)$ can be written as $\max_{j \neq k} |\langle \mathbf{s}_j, \mathbf{s}_k \rangle_{\mathcal{H}}|$. Using the cumulative coherence function to restrict the structure of \mathcal{S} , we obtain the following result.

Theorem 3 (Coherence-based convergence rate). *Let $\mathbf{r}^* := \mathbf{y} - \mathbf{x}^*$.*

If $\mu(n-1) < 1$, where $n = |\mathcal{S}|$, then the residuals follow linear convergence as

$$\|\mathbf{r}_{r+1} - \mathbf{r}^*\|_{\mathcal{H}}^2 \leq \left(1 - \frac{1 - \mu(n-1)}{n}\right) \|\mathbf{r}_r - \mathbf{r}^*\|_{\mathcal{H}}^2.$$

²In OMP and MP, the LMO consist in solving $\arg \max_{\mathbf{x} \in \mathcal{S}} |\langle \mathbf{x}, \mathbf{r}_{r-1} \rangle_{\mathcal{H}}|$ at iteration r .

Writing out the condition $\mu(n-1) < 1$ yields $\max_{k \in [n]} \sum_{i \in [n] \setminus k} |\langle \mathbf{s}_k, \mathbf{s}_i \rangle_{\mathcal{H}}| < 1$, which implies that the atoms in \mathcal{S} need to be close to orthogonal when the number of atoms is on the order of the ambient space dimension. Thus, Theorem 3 gives us an explicit convergence rate at the cost of imposing strong structural conditions on \mathcal{S} .

Considering the special case of our Theorem 1 for achievable $\mathbf{y} \in \text{lin}(\mathcal{S})$, finite dictionary, and $f_{\mathcal{H}}(\mathbf{x}) := \frac{1}{2}d_{\mathcal{H}}^2(\mathbf{x}, \mathbf{y})$, we recover the following known result for the linear convergence of OMP under an analogous coherence bound on \mathcal{S} :

Corollary 4 (see also Gribonval and Vandergheynst (2006, Thm. 2b)). *If $\mathbf{y} \in \text{lin}(\mathcal{S})$, then $\forall 1 \leq r \leq m$ s.t. $\mu(m-1) < 1, m \leq n$,*

$$\|\mathbf{r}_{r+1}\|_{\mathcal{H}}^2 \leq \left(1 - \frac{1 - \mu(m-1)}{m}\right) \|\mathbf{r}_r\|_{\mathcal{H}}^2.$$

Proofs of Theorems 3 and Corollary 4 are provided in the supplementary material.

3. Matrix Pursuit

Motivated from the pursuit algorithms from the previous section, we here present a generalized pursuit framework for structured matrix factorizations.

In order to encode interesting structure for matrix factorizations, we study the following class of matrix atoms, which are simply constructed by arbitrary two sets of vector atoms $\mathcal{A}_1 \subseteq \mathbb{R}^n$ and $\mathcal{A}_2 \subseteq \mathbb{R}^m$. We will study pursuit algorithms on the set of rank-1 matrices $\mathcal{A}_1 \otimes \mathcal{A}_2$, each element being an outer product of two vectors.

Specializing the general optimization problem (1) to sets $\text{lin}(\mathcal{A}_1 \otimes \mathcal{A}_2)$, we obtain the following structured matrix factorization notion: Given an objective function $f: \mathbb{R}^{n \times m} \rightarrow \mathbb{R}$, we want to find a matrix \mathbf{X} optimizing

$$\min_{\mathbf{X} \in \text{lin}(\mathcal{A}_1 \otimes \mathcal{A}_2)} f(\mathbf{X}). \quad (2)$$

When restricting (2) to candidate solutions of rank at most R , we obtain the following equivalent and more interpretable factorized reformulation:

$$\min_{\substack{\mathbf{u}_i \in \mathcal{A}_1 \forall i \in [R], \\ \mathbf{v}_i \in \mathcal{A}_2 \forall i \in [R], \\ \alpha \in \mathbb{R}^R}} f\left(\sum_{i=1}^R \alpha_i \mathbf{u}_i \otimes \mathbf{v}_i\right). \quad (3)$$

Symmetric factorizations. The above problem structure is also interesting in the special case of symmetric matrices, when restricting to just one set of vector atoms \mathcal{A}_1 (and $\mathbf{u} = \mathbf{v}$), which results in symmetric matrix factorizations build from atoms of the form $\mathbf{u}\mathbf{u}^{\top}$.

Table 1. Some example applications for our matrix pursuit framework (Algorithm 2). The table characterizes the applications by the set of atoms used to enforce matrix structure (*rows*), and two prominent optimization objectives (*columns*), being low-rank matrix approximation and low-rank matrix completion (MC). All cases apply for both symmetric as well as general rectangular matrices.

(a) Symmetric Structured Matrix Factorizations, $\mathbf{u}_i \in \mathcal{A}_1 \forall i$

Atoms \mathcal{A}_1	$f = \ \mathbf{Y} - \sum_i \alpha_i \mathbf{u}_i \otimes \mathbf{u}_i\ _F^2$	$f = \ \mathbf{Y} - \sum_i \alpha_i \mathbf{u}_i \otimes \mathbf{u}_i\ _\Omega^2$
$\{\mathbf{u} : \ \mathbf{u}\ _2 = 1\}$	PCA	MC
$\{\mathbf{u} : \ \mathbf{u}\ _2 = 1, \ \mathbf{u}\ _0 = k\}$	sparse PCA	structured MC

(b) Non-Symmetric Structured Matrix Factorizations, $\mathbf{u}_i \in \mathcal{A}_1, \mathbf{v}_i \in \mathcal{A}_2 \forall i$

Atoms \mathcal{A}_1	Atoms \mathcal{A}_2	$f = \ \mathbf{Y} - \sum_i \alpha_i \mathbf{u}_i \otimes \mathbf{v}_i\ _F^2$	$f = \ \mathbf{Y} - \sum_i \alpha_i \mathbf{u}_i \otimes \mathbf{v}_i\ _\Omega^2$
$\{\mathbf{u} : \ \mathbf{u}\ _2 = 1\}$	$\{\mathbf{v} : \ \mathbf{v}\ _2 = 1\}$	SVD	MC
$\{\mathbf{u} : \ \mathbf{u}\ _2 = 1, \ \mathbf{u}\ _0 = k\}$	$\{\mathbf{v} : \ \mathbf{v}\ _2 = 1, \ \mathbf{v}\ _0 = q\}$	sparse SVD	structured MC
$\{\mathbf{u} : \ \mathbf{u}\ _2 = 1, \mathbf{u} \geq \mathbf{0}\}$	$\{\mathbf{v} : \ \mathbf{v}\ _2 = 1, \mathbf{v} \geq \mathbf{0}\}$	NMF	structured MC
$\{\mathbf{u} : \ \mathbf{u}\ _2 = 1, \mathbf{u} \geq \mathbf{0}, \ \mathbf{u}\ _0 = k\}$	$\{\mathbf{v} : \ \mathbf{v}\ _2 = 1, \mathbf{v} \geq \mathbf{0}, \ \mathbf{v}\ _0 = q\}$	sparse NMF	structured MC

Applications. We present some prominent applications of structured matrix factorizations within our pursuit framework in Table 1.

The vector atom sets \mathcal{A}_1 and \mathcal{A}_2 encode the desired matrix factorization structure. For example, in the special case $f(\sum_i \alpha_i \mathbf{u}_i \otimes \mathbf{v}_i) = \|\mathbf{Y} - \sum_i \alpha_i \mathbf{u}_i \otimes \mathbf{v}_i\|_F^2$ for a given matrix \mathbf{Y} , and atoms $\mathcal{A}_1 = \mathcal{A}_2 = \{\mathbf{x} : \|\mathbf{x}\|_2 = 1\}$, problem (2) becomes the standard SVD.

Typical structures of interest for matrix factorizations include sparsity of the factors in various forms, including group/graph structured sparsity (see (Baraniuk et al., 2010) and references therein for more examples). Furthermore, non-negative factorizations are widely used, also ordered vectors and several other structures on the atom vectors. In our framework, it is easy to also use combinations of several different vector structures. Also, note that the sets \mathcal{A}_1 and \mathcal{A}_2 are by no means required to be of the same structure. For the rest of the paper, we assume that the sets \mathcal{A}_1 and \mathcal{A}_2 are compact.

Algorithm. The main matrix pursuit algorithm derived from Algorithm 1 applied to problems of form (3) is presented in Algorithm 2.

Algorithm 2 Generalized Matrix Pursuit (GMP)

Require: \mathbf{X}_0, R

- 1: **for** $r = 1..R$ **do**
 - 2: $\mathbf{u}_r, \mathbf{v}_r := (\text{Approx-})\text{LMO}_{\mathcal{A}_1 \otimes \mathcal{A}_2}(\nabla f(\mathbf{X}_{r-1}))$
 - 3: $\alpha := \min_{\alpha \in \mathbb{R}^r} f(\sum_{i \leq r} \alpha_i \mathbf{u}_i \otimes \mathbf{v}_i)$
 - 4: *Optional:* Correction of some/all $\mathbf{u}_i, \mathbf{v}_i$.
 - 5: Update iterate $\mathbf{X}_r := \sum_{i \leq r} \alpha_i \mathbf{u}_i \mathbf{v}_i^\top$
 - 6: **end for**
-

In practice, the atom-correction step (Step 4) is specially important for maintaining iterates of even smaller rank in

practice, as also highlighted by our experiments. *Local* corrections are made to the already chosen set of atoms to potentially improve the quality of rank- r solution.

Matrix completion. Variants of (structured) matrix completion are obtained for the objective function

$$f\left(\sum_i \alpha_i \mathbf{u}_i \otimes \mathbf{v}_i\right) = \left\| \mathbf{Y} - \sum_i \alpha_i \mathbf{u}_i \otimes \mathbf{v}_i \right\|_\Omega^2, \quad (4)$$

where Ω is set of observed indices. Here the norm on the vector space is defined with respect to only the observed entries. Formally, $\|\mathbf{Z}\|_\Omega^2 = \|\mathbf{Z}_\Omega\|_F^2$ is induced by the inner product $\langle \mathbf{A}, \mathbf{B} \rangle_\Omega := \text{tr}(\mathbf{A}_\Omega^\top \mathbf{B}_\Omega)$.

Convergence. The linear rate of convergence proved in Theorem 1 is directly applicable to Algorithm 2 as well. This is again subject to the availability of a linear oracle (LMO) for the used atoms.

The convergence rate presented by Wang et al. (2014) for the case of matrix completion can be obtained directly as a special case of our Theorem 1, for $\mathcal{A}_1 := \{\mathbf{u} : \|\mathbf{u}\|_2 = 1\}$, $\mathcal{A}_2 := \{\mathbf{v} : \|\mathbf{v}\|_2 = 1\}$ and $\mathbf{R}^* := \mathbf{Y} - \mathbf{X}^* = \mathbf{0}$.

Generalized rank for structured factorizations. For the case of given $\mathbf{y} \in \text{lin}(\mathcal{S})$, the number of iterations performed by Algorithm 2 can be thought of as a complexity measure of generalized matrix rank, specific to our objective function $f_{\mathcal{H}}$ and the atomic sets. As an example, one can directly obtain the analogue of the k - q rank of matrices for sparse SVD defined by Richard et al. (2014).

3.1. Atom Correction Variants

Algorithms 1 and 2 guarantee linear convergence in terms of number of calls made to the LMO oracle, each iteration increasing the rank of the iterate by one. In many applications such as low rank PCA, low rank matrix completion

etc., it is desirable to have iterates being a linear combination of only as few atoms as possible. As discussed in the previous Section 2, we do allow for corrections to obtain a possibly much lower function cost with a given fixed rank approximation. The more severe corrections of atoms themselves in step 4 (as opposed to just their weights) can be made by updating them one or a few atoms at a time, keeping the rest of them fixed. For the symmetric case, the update of the i^{th} atom can be written as

$$\mathbf{u}_i^+ := \arg \min_{\mathbf{u} \in \mathcal{A}_1} f\left(\sum_{j \neq i} \alpha_j \mathbf{u}_j \otimes \mathbf{u}_j + \alpha_i \mathbf{u} \otimes \mathbf{u}\right). \quad (5)$$

The update for non-symmetric case is analogous. The complexity of atom corrections depends very strongly on the structure of the used atomic sets \mathcal{A}_1 and \mathcal{A}_2 . One general way is to call the LMO again, but assuming the current iterate is $\sum_{j \neq i} \alpha_j \mathbf{u}_j \otimes \mathbf{u}_j$. For the non-symmetric case, techniques such as alternating minimization between \mathbf{u}_i and \mathbf{v}_i are also useful if the call to the LMO is more expensive. Note that for the nuclear norm special case $\mathcal{A}_1 = \mathcal{A}_2 = \{\mathbf{x} : \|\mathbf{x}\|_2 = 1\}$, variants of such corrections of the atoms were studied by Laue (2012).

In contrast to the non-convex corrections of atoms \mathbf{u}_i , the optimal updates of the weights α alone as in line 3 can be performed very efficiently after every atom update (as e.g. in OMP), by simply solving a linear system of size $r \times r$.

4. Implementing the Linear Minimization Oracle for Matrix Factorizations

As in the Frank-Wolfe algorithm, the LMO is required to return a minimizer of the function linearized at the current value of \mathbf{X} . As we mentioned in the introduction, this type of *greedy* oracle has been very widely used in several classical algorithms, including many variants of matching pursuit, Frank-Wolfe and related sparse greedy methods (Frank and Wolfe, 1956; Jaggi and Sulovský, 2010; Tewari et al., 2011; Wang et al., 2014; Bach, 2013).

Say $\mathbf{X}_r \in \mathbb{R}^{n \times m}$ is the current iterate of algorithm or Algorithm 2 on a matrix problem, for the objective function $f(\mathbf{X}) = \|\mathbf{Y} - \mathbf{X}\|_F^2$. In the symmetric case (the non-symmetric case is analogous), for arbitrary \mathcal{A}_1 , the LMO can be equivalently written as finding the vector which solves

$$\arg \max_{\mathbf{u} \in \mathcal{A}_1} \langle -\nabla f(\mathbf{X}_r), \mathbf{u} \otimes \mathbf{u} \rangle \quad (6)$$

It is easy to see (6) represents a *generalized* version of the top eigenvector problem (or singular vector, for the non-symmetric case). The problem in general is NP-hard for arbitrary atomic sets (such for example already in the simple case of unit-length non-negative vectors (Murty and Kabadi, 1987)). Specific atomic structures and assumptions can allow for an efficient LMO. Nevertheless, we will here provide a very general technique to design the

LMO for arbitrary vector atoms, namely the *atomic power method*. Our proposed method will iteratively run on the LMO problem (which is non-convex in its variable vector \mathbf{u}), in an ascent fashion. As is the case with ascent methods on non-convex problems, the presented analysis will only show convergence to a fixed point. For hard LMO problems, the atomic power method can be run several times with different initialization, and the best outcome can be chosen as the LMO solution.

4.1. The Atomic Power Method

We will first address the symmetric case of the non-convex LMO problem (6). We use the Frank-Wolfe algorithm (Frank and Wolfe, 1956; Jaggi, 2013) with a fixed step size of 1 to approximate the LMO problem. Although designed for constrained convex *minimization*, it is known that using a fixed step size of 1 can make Frank-Wolfe methods suitable for constrained (non-convex) *maximization* as in our formulation (6) (Journée et al., 2010; Luss and Teboulle, 2013).

To solve (6), say $\mathbf{u}^{(t)}$ is the t^{th} iterate ($\mathbf{u}^{(0)} \in \mathbb{R}^n$ is the initialization). Recall that, $-\nabla f(\mathbf{X}_r) = \mathbf{R}_r$. The next Frank-Wolfe iterate is obtained as

$$\mathbf{u}^{(t+1)} \leftarrow \arg \max_{\mathbf{u} \in \mathcal{A}_1} \langle \mathbf{u}, \mathbf{R}_r \mathbf{u}^{(t)} \rangle \quad (7)$$

We call the update step (7) an *atomic power iteration*. It is easy to see that it recovers the standard power method as a special case, as well as the Truncated Power Method for sparse PCA suggested by Yuan and Zhang (2013), the sparse power methods suggested by Luss and Teboulle (2013), and the cone constrained power method suggested by Deshpande et al. (2014). It can be shown that the iterates monotonically increase the function value.

Analysis. Our analysis is based on the techniques suggested by the work on convex constrained maximization by Journée et al. (2010) and Luss and Teboulle (2013). While their focus is on the sparse PCA setting, we apply their results to general sets of vector atoms. Let $g(\mathbf{u}) := \langle \mathbf{R}_r, \mathbf{u} \otimes \mathbf{u} \rangle$ be the value of the LMO problem for a given vector \mathbf{u} , and $I(t) := \max_{\mathbf{u} \in \mathcal{A}_1} \langle \mathbf{Y}_r \mathbf{u}^{(t)}, \mathbf{u} - \mathbf{u}^{(t)} \rangle$. Note that $I(t) \geq 0$ by definition. We assume $\forall \mathbf{u} \in \mathcal{A}_1$, $\langle \mathbf{R}_r, \mathbf{u} \otimes \mathbf{u} \rangle \geq 0$ so that $g(\cdot)$ is convex on $\text{conv}(\mathcal{A}_1)$. Or, a looser assumption to make is that all atoms in \mathcal{A}_1 are normalized i.e. $\forall \mathbf{u} \in \mathcal{A}_1$, $\mathbf{u}^\top \mathbf{u} = \text{const}$. Note that this assumption holds for most practical applications. If this is the case, $g(\cdot)$ can simply be made convex by defining it as $g(\mathbf{u}) := \langle \mathbf{R}_r + \kappa \mathbf{I}, \mathbf{u} \otimes \mathbf{u} \rangle$ for large enough κ . Adding a term that is constant for all atoms in \mathcal{A}_1 does not change the maximizer.

The following proposition is a consequence of Theorem 3.4 in the work of Luss and Teboulle (2013).

Proposition 5. *If $\forall \mathbf{u} \in \mathcal{A}_1, \mathbf{u}^\top \mathbf{R}_r \mathbf{u} \geq 0$, and \mathcal{A}_1 is compact, then,*

- (a) *The sequence $\{g(\mathbf{u}^{(t)})\}$ is monotonically increasing.*
- (b) *The sequence $\{I(t)\} \rightarrow 0$.*
- (c) *The sequence of iterates of atomic power method $\{\mathbf{u}^{(t)}\}$ converges to a stationary point of $g(\cdot)$.*

Proof sketch. (a) follows because of interval convexity of $g(\cdot)$ on \mathcal{A}_1 which implies

$$\begin{aligned} g(\mathbf{u}^{(t+1)}) &\geq g(\mathbf{u}^{(t)}) + \langle \mathbf{R}_r \mathbf{u}^{(t)}, \mathbf{u}^{(t+1)} - \mathbf{u}^{(t)} \rangle \\ &= g(\mathbf{u}^{(t)}) + I(t), \end{aligned}$$

and because $I(t) \geq 0$ by definition. (b) follows because of (a) and because $g(\cdot)$ is upper bounded on \mathcal{A}_1 (compactness assumption). (c) is a direct consequence from (b) by definition of $I(t)$. \square

Hence at each iteration, the value of the optimization problem increases unless $I(t) = 0$ which is the first order optimality condition till a fixed point is reached.

For sharper analysis, we make further assumptions on $g(\cdot)$ and \mathcal{A}_1 and provide corresponding convergence results. We assume a form of restricted strong convexity of the LMO function on \mathcal{A}_1 . It is easy to see that this is equivalent to assuming $\langle \mathbf{R}_r, \frac{\mathbf{u} \otimes \mathbf{u}}{\|\mathbf{u}\|_2^2} \rangle \geq \sigma_{\min} > 0, \forall \mathbf{u} \in \mathcal{A}_1$. This directly implies that $\text{null}(\mathbf{R}_r) \cap \mathcal{A}_1 = \{\}$. So, $\exists \gamma$ s.t. $\|\mathbf{R}_r \mathbf{u}\|_2 \geq \gamma > 0$. Further, assume that $\text{conv}(\mathcal{A}_1)$ is δ -strongly convex. Using the analysis developed in (Journée et al., 2010, Section 3.4), we can give guarantees for convergence of the sequence $\{\mathbf{u}^{(t)}\}$ developed in Proposition 5. Say g^* a stationary point of the sequence $\{g(\mathbf{u}^{(t)})\}$.

Proposition 6. *If,*

- *$g(\mathbf{u})$ is σ_{\min} -strongly convex on \mathcal{A}_1 , and consequently has $\forall \mathbf{u} \in \mathcal{A}_1, \|\mathbf{R}_r \mathbf{u}\|_2 \geq \gamma > 0$ for some γ , and*
- *$\text{conv}(\mathcal{A}_1)$ is a δ -strongly convex set with $\delta \geq 0$,*

then, for $\gamma, \delta, \sigma_{\min}$ as defined above, for $k \geq \frac{1}{\epsilon^2} \frac{g^ - g(\mathbf{u}^{(0)})}{\gamma \delta + \sigma_{\min}}$, the atomic power iterates converge as $\min_k \|\mathbf{u}^{(t+1)} - \mathbf{u}^{(t)}\|_2 \leq \epsilon$.*

Proof. See (Journée et al., 2010, Theorem 4). \square

We discussed a generic way to design an LMO for the symmetric case. For the non-symmetric case, a natural way to find the maximizing \mathbf{u}, \mathbf{v} is by alternating maximization, which leads to an alternating power method. This is interesting given the success of alternating minimization methods for matrix factorization problems (Hardt, 2013). Alternatively, we can set $\mathbf{T} = \begin{bmatrix} 0 & \mathbf{R} \\ \mathbf{R}^\top & 0 \end{bmatrix}$, $\mathbf{t} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}$ to reduce the

non-symmetric case to the symmetric one, and our previous analysis can be re-applied.

5. Experiments

In this section, we demonstrate some results on real world datasets. We provide empirical evidence for improvements for two algorithms: Truncated Power Method for sparse PCA and matrix completion by rank-1 pursuit, by applying atom corrections suggested in Section 3.1. We also apply to the framework to sparse non-negative vectors as the atomic set to obtain a new algorithm using our proposed LMO design and compare it to existing methods. For the matrix completion, our framework yields a new general algorithm for structured matrix completion. We use two versions of our algorithm - ApproxGMP solves the LMO by power iterations without corrections, while ApproxGMP_r also uses corrections (Section 3.1). Note that GMP (solving the LMO exactly) is NP hard for the cases considered here (except for the setup of matrix completion by Wang et al. (2014) which uses SVD), and is seldom used in practice, hence it is not compared against.

Sparse PCA: Inadequacy of Deflation. Sparse PCA is a special symmetric case of (3), with the respective vector atom set defined as $\mathcal{A}_1 = \{\mathbf{u} : \|\mathbf{u}\|_2 = 1, \|\mathbf{u}\|_0 \leq k\}$, where k is the desired sparsity level defined externally as a parameter.

Since finding the top sparse eigenvector is an NP-hard problem in general, various practical algorithms only solve the problem approximately. As such, any deflation technique (see (Mackey, 2009) for an overview of deflation for sparse PCA) coupled with finding an approximate top-eigenvector of the deflated matrix are still *greedy* approaches similar to the un-corrected variant of Algorithm 2. This suggests that the optional atom corrections could be useful.

To illustrate the utility of performing atom corrections, we consider the Truncated Power Method described by Yuan and Zhang (2013) which can be derived as a special case of the approximate LMO in Algorithm 2. We consider the Leukemia and CBCL face training datasets (Lichman, 2013). The Leukemia dataset has 72 samples, each consisting of expression values for 12582 probe sets. CBCL face training dataset contains 2429 images each represented as a feature vector of size 361 pixels. For $r = 5$ components each with sparsity 300, GMP obtains 0.3472 as the ratio of variance explained as opposed to 0.3438 by TPower+orthogonal deflation. Similarly for CBCL data, GMP and TPower obtain 0.7326 and 0.7231, respectively, for sparsity=200.

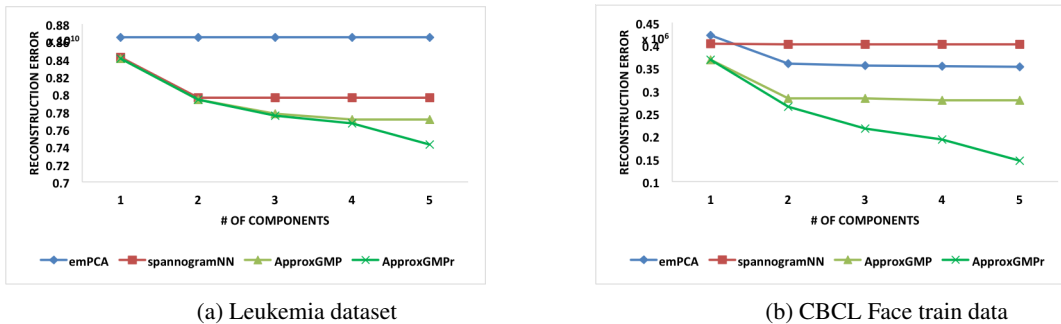


Figure 1. Reconstruction Error

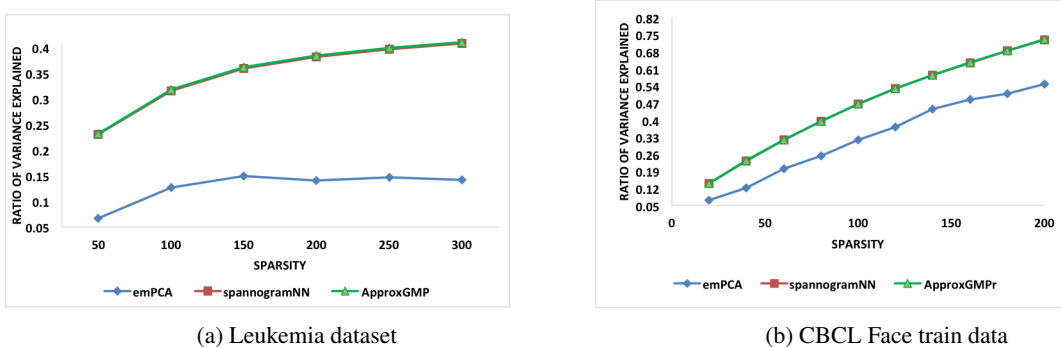


Figure 2. Ratio of variance explained vs. top-eigenvector

Sparse Non-Negative PCA. For Sparse Non-Negative PCA, we use the Leukemia and CBCL Face training datasets as well. The problem is a special symmetric case of (3), with the respective vector atom set defined as $\mathcal{A}_1 = \{\mathbf{u} : \|\mathbf{u}\|_2 = 1, \|\mathbf{u}\|_0 \leq k, \mathbf{u} \geq 0\}$, where k is the desired sparsity level defined externally as a parameter. In this case study, in addition to using corrections as in the previous case study, we use the atomic power method described in Section 4.1 to derive a new algorithm for the atomic set defined above which to our knowledge has not been seen or studied before.

There is little prior work on Sparse Non-negative PCA. We compare against the spannogramNN by Asteris et al. (2014) and emPCA by Sigg and Buhmann (2008). Our algorithm of atomic power iterations is easier to implement, converges faster and gives better results compared to both of these. Figure 1 shows reconstruction error with increasing rank. Figure 2 shows the ratio of variance explained for a rank one approximation for sparse non-negative PCA. We note that ApproxGMP has performance comparable to that of spannogramNN, and both ApproxGMP and spannogramNN outperform emPCA.

Structured Low-Rank Matrix Completion. For $\mathcal{A}_1 = \{\mathbf{u} \in \mathbb{R}^n : \|\mathbf{u}\|_2 = 1\}$, $\mathcal{A}_2 = \{\mathbf{v} \in \mathbb{R}^d : \|\mathbf{v}\|_2 = 1\}$, and $f(\sum_i \alpha_i \mathbf{u}_i \otimes \mathbf{v}_i) := \|\mathbf{T}_\Omega - \sum_i \alpha_i [\mathbf{u}_i \otimes \mathbf{v}_i]_\Omega\|_F$, GMP (Algorithm 2) can be used for matrix completion. This recov-

ers the work of Wang et al. (2014) who study this special case as their algorithm OR1MP and provide linear convergence guarantees. We empirically show that by using corrections, we get significantly better results in terms of reconstruction error. Furthermore, our more general analysis shows that the linear convergence holds for any structured sets. We consider the specific case of sparsity. Soni et al. (2014) study the regularization impact of sparse low rank matrix completion for robustness. Our framework yields a new algorithm for simple rank-one pursuit with alternating atomic power method for sparse factors. We used 3 movie-lens datasets of varying sizes for our experiments. In each dataset, we randomly split the ratings into 50-20-30 training, validation and testing split (we generate 20 different splits). The validation dataset is used for selecting the rank and applying further corrections for better generalization. Our results (averaged over 20 runs) are reported in Table 2. We find that our generalizations of the OR1MP results in better reconstruction error for all three datasets. See the work by Wang et al. (2014) for a comparison of the performance of OR1MP with other matrix completion methods, and the work by Soni et al. (2014) on robustness analysis for sparse factor low rank matrix completion.

Conclusion and Future Work

We presented a pursuit framework for structured low rank matrix factorization and completion. Studying the tradeoff between rank and approximation quality, we proved linear convergence of generalized pursuit in Hilbert Spaces, of which matrix pursuit is a special case. Another direct application would be tensor pursuit for low rank tensor factorization and completion. A general design for the LMO construction for structured sets for tensors is an interesting future direction to explore. Moreover, generalization of the convergence results beyond distance functions is an interesting extension of the present work with many applications. Further, note that both the generalized pursuit and the Frank-Wolfe algorithms solve the same LMO to settle on the next best atom to add. We borrowed the idea of correcting already chosen atoms from the FW framework. Hence, studying the connection between the two frameworks should yield more insights in the future.

References

- Megasthenis Asteris, Dimitris Papailiopoulos, and Alexandros Dimakis. Nonnegative sparse pca with provable guarantees. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1728–1736. JMLR Workshop and Conference Proceedings, 2014.
- Francis Bach. Convex relaxations of structured matrix factorizations. *arXiv.org*, 2013.
- Francis Bach, Julien Mairal, and Jean Ponce. Convex Sparse Matrix Factorizations. Technical report, 2008.
- Richard G. Baraniuk, Volkan Cevher, Marco F. Duarte, and Chinmay Hegde. Model-based compressive sensing. *IEEE Trans. Inf. Theor.*, 56(4):1982–2001, April 2010.
- T Blumensath and M E Davies. Gradient Pursuits. *Signal Processing, IEEE Transactions on*, 56(6):2370–2382, June 2008.
- Emmanuel J Candes and Benjamin Recht. Exact Matrix Completion via Convex Optimization. *Foundations of Computational Mathematics*, 9(6):717–772, April 2009.
- Emmanuel J Candes and Terence Tao. The Power of Convex Relaxation: Near-Optimal Matrix Completion. *IEEE Transactions on Information Theory*, 56(5) 2053–2080, 2010.
- Emmanuel J Candes, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3), May 2011.
- Sheng Chen, Stephen A Billings, and Wan Luo. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of control*, 50(5):1873–1896, 1989.
- G Davis, S Mallat, and M Avellaneda. Adaptive greedy approximations. *Constructive Approximation*, 13(1):57–98, March 1997.
- Yash Deshpande, Andrea Montanari, and Emile Richard. Cone-constrained principal component analysis. In *NIPS - Advances in Neural Information Processing Systems 27*, pages 2717–2725. 2014.
- Ronald A. DeVore and Vladimir N Temlyakov. Convex optimization on Banach Spaces. *arXiv.org*, January 2014.
- Ronald A. DeVore and Vladimir N. Temlyakov. Some remarks on greedy algorithms. *Adv. Comput. Math.*, 5(1):173–187, 1996.
- Miroslav Dudík, Zaid Harchaoui, and Jerome Malick. Lifted coordinate descent for learning with trace-norm regularization. In *AISTATS*, March 2012.
- François-Xavier Dupé. Greed is Fine: on Finding Sparse Zeros of Hilbert Operators. working paper or preprint, February 2015.
- Marguerite Frank and Philip Wolfe. An Algorithm for Quadratic Programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.
- Nicolas Gillis and François Glineur. Low-Rank Matrix Approximation with Weights or Missing Data Is NP-Hard. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1149, 2011.

- Rémi Gribonval and P Vandergheynst. On the exponential convergence of matching pursuits in quasi-incoherent dictionaries. *IEEE Transactions on Information Theory*, 52(1):255–261, 2006.
- Moritz Hardt. Understanding Alternating Minimization for Matrix Completion. *arXiv.org*, December 2013.
- Elad Hazan. Sparse Approximate Solutions to Semidefinite Programs. In *LATIN 2008*, pages 306–316. Springer Berlin Heidelberg, 2008.
- Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML 2013 - Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 427–435, 2013.
- Martin Jaggi and Marek Sulovský. A Simple Algorithm for Nuclear Norm Regularized Problems. *ICML 2010: Proceedings of the 27th international conference on Machine learning*, 2010.
- Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 665–674, New York, NY, USA, 2013. ACM.
- Lee K. Jones. On a conjecture of Huber concerning the convergence of projection pursuit regression. *Ann. Stat.*, 15:880–882, 1987. ISSN 0090-5364; 2168-8966/e.
- Michel Journée, Yurii Nesterov, Peter Richtárik, and Rodolphe Sepulchre. Generalized power method for sparse principal component analysis. *J. Mach. Learn. Res.*, 11:517–553, March 2010. ISSN 1532-4435.
- Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8):30–37, August 2009.
- Simon Lacoste-Julien and Martin Jaggi. On the Global Linear Convergence of Frank-Wolfe Optimization Variants. In *NIPS 2015 - Advances in Neural Information Processing Systems 28*, pages 496–504, 2015.
- Sören Laue. A Hybrid Algorithm for Convex Semidefinite Optimization. In *ICML*, 2012.
- Omer Levy and Yoav Goldberg. Neural Word Embedding as Implicit Matrix Factorization. In *NIPS 2014 - Advances in Neural Information Processing Systems 27*, 2014.
- M. Lichman. UCI machine learning repository, 2013.
- Ronny Luss and Marc Teboulle. Conditional gradient algorithms for rank-one matrix approximations with a sparsity constraint. *SIAM Review Vol. 55 No. 1*, abs/1107.1163, 2013.
- Lester Mackey. Deflation methods for sparse PCA. In *NIPS*, 2009.
- Stéphane Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.
- Katta G Murty and Santosh N Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39(2):117–129, June 1987.
- Dimitris S. Papailiopoulos, Alexandros G. Dimakis, and Stavros Korokythakis. Sparse PCA through low-rank approximations. *ICML*, 2013.
- Ting Kei Pong, Paul Tseng, Shuiwang Ji, and Jieping Ye. Trace Norm Regularization: Reformulations, Algorithms, and Multi-Task Learning. *SIAM Journal on Optimization*, 20(6):3465–3489, 2010.
- Benjamin Recht. A simpler approach to matrix completion. *CoRR*, abs/0910.0651, 2009.
- Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization. *SIAM Review*, 52(3):471–501, 2010.
- Emile Richard, Guillaume R Obozinski, and Jean-Philippe Vert. Tight convex relaxations for sparse matrix factorization. In *Advances in Neural Information Processing Systems 27*, pages 3284–3292. Curran Associates, Inc., 2014.
- Christian D. Sigg and Joachim M. Buhmann. Expectation-maximization for sparse and non-negative pca. In *ICML*, pages 960–967, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4.
- Akshay Soni, Swayambhoo Jain, Jarvis D. Haupt, and Stefano Gonella. Noisy matrix completion under sparse factor models. *CoRR*, abs/1411.0282, 2014.
- Vladimir Temlyakov. Greedy algorithms in convex optimization on Banach spaces. In *48th Asilomar Conference on Signals, Systems and Computers*, pages 1331–1335. IEEE, 2014.
- Ambuj Tewari, Pradeep K. Ravikumar, and Inderjit S. Dhillon. Greedy algorithms for structurally constrained high dimensional problems. In *Advances in Neural Information Processing Systems 24*, pages 882–890, 2011.
- K Toh and S Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Optimization Online*, 2009.
- Joel A Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.
- Zheng Wang, Ming jun Lai, Zhaosong Lu, Wei Fan, Hasan Davulcu, and Jieping Ye. Rank-one matrix pursuit for matrix completion. In *ICML-14 - Proceedings of the 31st International Conference on Machine Learning*, pages 91–99, 2014.
- Yuning Yang, Siamak Mehrkanoon, and Johan A K Suykens. Higher order Matching Pursuit for Low Rank Tensor Learning. *arXiv.org*, March 2015.
- Xiao-Tong Yuan and Tong Zhang. Truncated power method for sparse eigenvalue problems. *J. Mach. Learn. Res.*, 14(1):899–925, April 2013. ISSN 1532-4435.

Dataset	ApproxGMP	ApproxGMPr	ApproxGMPr (Sparse)
Movielens100k	1.778 ± 0.03	1.691 ± 0.03	1.62 ± 0.01
Movielens1M	1.6863 ± 0.01	1.6537 ± 0.01	1.6411 ± 0.01
Movielens10M	1.8634 ± 0.01	1.8484 ± 0.01	1.8452 ± 0.01

Table 2. RMSE on test set : average over 20 runs \pm variance. For ApproxGMPr (Sparse), left singular vector is fully dense while the right one has sparsity 0.6 of its size (chosen by trial and error)

A. Proofs

A.1. Proof of Theorem 1

We prove that Algorithm 1 converges linearly for \mathbf{x}_0 to \mathbf{x}^* . To bound the convergence rate, we need to study how the residual changes over iterations. To this end, we define $\mathbf{q}_i := \mathbf{r}_i - \mathbf{r}_{i+1}$. We start by stating auxiliary results that will be used to prove Theorem 1. Recall from Algorithm 1 that $\mathbf{z}_i = \text{LMO}(-\mathbf{r}_i) = \arg \max_{\mathbf{z} \in \mathcal{S}} \langle \mathbf{z}, \mathbf{r}_i \rangle$.

Proposition 7. *Let $\mathbf{r}_i \in \mathcal{H}$ be a residual, i.e., $\mathbf{r}_i = \mathbf{y} - \sum_{j < i} \alpha_j \mathbf{z}_j$, then $\max_{\mathbf{z} \in \mathcal{S}} \langle \mathbf{r}_i, \mathbf{z} \rangle_{\mathcal{H}} = 0$ iff $\mathbf{r}_i = \mathbf{r}^*$.*

Proof. Follows from the first-order optimality condition for \mathbf{r}^* . \square

Proposition 8. $\langle \mathbf{r}_i, \mathbf{z}_j \rangle = 0, \forall j < i$.

Proof. Follows from the first-order optimality condition for α . \square

Proposition 9. $\langle \mathbf{r}_i, \mathbf{q}_j \rangle = 0, \forall j < i$.

Proof. By definition $\mathbf{q}_j \in \text{lin}(\{\mathbf{z}_1, \dots, \mathbf{z}_j\})$. \square

Proposition 10.

$$\|\mathbf{q}_i\|_{\mathcal{H}}^2 \geq \frac{|\langle \mathbf{z}_i, \mathbf{r}_i \rangle_{\mathcal{H}}|^2}{\|\mathbf{z}_i\|_{\mathcal{H}}^2}$$

Proof. Let P_i be the orthogonal projection operator to $\text{lin}(\{\mathbf{z}_1, \dots, \mathbf{z}_i\})$, i.e., for $\mathbf{x} \in \mathcal{H}$ we have $P_i \mathbf{x} = \arg \min_{\mathbf{z} \in \text{lin}(\{\mathbf{z}_1, \dots, \mathbf{z}_i\})} \|\mathbf{z} - \mathbf{x}\|_{\mathcal{H}}^2$. Hence, $\mathbf{x}_i = P_i \mathbf{y}$ and $\mathbf{r}_i = (\mathbf{I} - P_{i-1}) \mathbf{y}$, where \mathbf{I} designates the identity operator on \mathcal{H} . By the Gram-Schmidt process we get

$$\begin{aligned} \mathbf{r}_{i+1} &= \underbrace{(\mathbf{I} - P_{i-1}) \mathbf{y}}_{=\mathbf{r}_i} \\ &+ \left\langle \frac{(\mathbf{I} - P_{i-1}) \mathbf{z}_i}{\|(\mathbf{I} - P_{i-1}) \mathbf{z}_i\|_{\mathcal{H}}}, \mathbf{y} \right\rangle_{\mathcal{H}} \frac{(\mathbf{I} - P_{i-1}) \mathbf{z}_i}{\|(\mathbf{I} - P_{i-1}) \mathbf{z}_i\|_{\mathcal{H}}} \end{aligned}$$

which in turn implies

$$\begin{aligned} \|\mathbf{q}_i\|_{\mathcal{H}}^2 &= \left\| \left\langle \frac{(\mathbf{I} - P_{i-1}) \mathbf{z}_i}{\|(\mathbf{I} - P_{i-1}) \mathbf{z}_i\|_{\mathcal{H}}}, \mathbf{y} \right\rangle_{\mathcal{H}} \frac{(\mathbf{I} - P_{i-1}) \mathbf{z}_i}{\|(\mathbf{I} - P_{i-1}) \mathbf{z}_i\|_{\mathcal{H}}} \right\|_{\mathcal{H}}^2 \\ &= \left| \left\langle \frac{(\mathbf{I} - P_{i-1}) \mathbf{z}_i}{\|(\mathbf{I} - P_{i-1}) \mathbf{z}_i\|_{\mathcal{H}}}, \mathbf{y} \right\rangle_{\mathcal{H}} \right|^2 \\ &= \frac{|\langle \mathbf{z}_i, (\mathbf{I} - P_{i-1}) \mathbf{y} \rangle_{\mathcal{H}}|^2}{\|(\mathbf{I} - P_{i-1}) \mathbf{z}_i\|_{\mathcal{H}}^2} \\ &= \frac{|\langle \mathbf{z}_i, \mathbf{r}_i \rangle_{\mathcal{H}}|^2}{\|(\mathbf{I} - P_{i-1}) \mathbf{z}_i\|_{\mathcal{H}}^2} \\ &\geq \frac{|\langle \mathbf{z}_i, \mathbf{r}_i \rangle_{\mathcal{H}}|^2}{\|\mathbf{z}_i\|_{\mathcal{H}}^2}, \end{aligned}$$

where we used that $(\mathbf{I} - P_{i-1})$ is a self-adjoint operator to obtain the third equality, and $\|(\mathbf{I} - P_{i-1})\|_{\text{op}} = 1$ to get the inequality. \square

We are now ready to prove Theorem 1.

Theorem 11 (Theorem 1). *Algorithm 1 converges linearly for $f_{\mathcal{H}}(\mathbf{x}) := \frac{1}{2} d_{\mathcal{H}}^2(\mathbf{x}, \mathbf{y})$, $\mathbf{y} \in \mathcal{H}$.*

Proof.

$$\begin{aligned} \|\mathbf{r}_{i+1} - \mathbf{r}^*\|_{\mathcal{H}}^2 &= \|\mathbf{r}_i - \mathbf{r}^* - \mathbf{q}_i\|_{\mathcal{H}}^2 \\ &= \|\mathbf{r}_i - \mathbf{r}^*\|_{\mathcal{H}}^2 + \|\mathbf{q}_i\|_{\mathcal{H}}^2 - 2\langle \mathbf{r}_i - \mathbf{r}^*, \mathbf{q}_i \rangle_{\mathcal{H}} \\ &= \|\mathbf{r}_i - \mathbf{r}^*\|_{\mathcal{H}}^2 + \|\mathbf{q}_i\|_{\mathcal{H}}^2 \\ &\quad - 2\langle \mathbf{r}_{i+1} + \mathbf{q}_i - \mathbf{r}^*, \mathbf{q}_i \rangle_{\mathcal{H}} \\ &= \|\mathbf{r}_i - \mathbf{r}^*\|_{\mathcal{H}}^2 + \|\mathbf{q}_i\|_{\mathcal{H}}^2 - 2\|\mathbf{q}_i\|_{\mathcal{H}}^2 \\ &= \|\mathbf{r}_i - \mathbf{r}^*\|_{\mathcal{H}}^2 - \|\mathbf{q}_i\|_{\mathcal{H}}^2 \\ &\leq \|\mathbf{r}_i - \mathbf{r}^*\|_{\mathcal{H}}^2 - \frac{\langle \mathbf{z}_i, \mathbf{r}_i \rangle_{\mathcal{H}}^2}{\|\mathbf{z}_i\|_{\mathcal{H}}^2} \quad (\text{Prop 10}) \\ &= \|\mathbf{r}_i - \mathbf{r}^*\|_{\mathcal{H}}^2 - \frac{\langle \mathbf{z}_i, \mathbf{r}_i - \mathbf{r}^* \rangle_{\mathcal{H}}^2}{\|\mathbf{z}_i\|_{\mathcal{H}}^2} \quad (8) \\ &= \left(1 - \frac{\langle \mathbf{z}_i, \mathbf{r}_i - \mathbf{r}^* \rangle_{\mathcal{H}}^2}{\|\mathbf{z}_i\|_{\mathcal{H}}^2 \|\mathbf{r}_i - \mathbf{r}^*\|_{\mathcal{H}}^2} \right) \|\mathbf{r}_i - \mathbf{r}^*\|_{\mathcal{H}}^2 \\ &= \mu \|\mathbf{r}_i - \mathbf{r}^*\|_{\mathcal{H}}^2 \quad (9) \end{aligned}$$

$\mu \in [0, 1)$ by Cauchy-Schwarz. \square

We finally note that proving convergence of Algorithm 1 without specifying a rate can be achieved as follows.

Proposition 12. *The sequence of residuals $\{\mathbf{r}_i\}$ produced by Algorithm 1 with $f_{\mathcal{H}}(\mathbf{x}) := \frac{1}{2} d_{\mathcal{H}}^2(\mathbf{x}, \mathbf{y})$, $\mathbf{y} \in \mathcal{H}$, converges to \mathbf{r}^* .*

Proof. We have

$$\begin{aligned} \|\mathbf{r}_{i+1}\|_{\mathcal{H}}^2 &= \|\mathbf{r}_i - \mathbf{q}_i\|_{\mathcal{H}}^2 \\ &= \|\mathbf{r}_i\|_{\mathcal{H}}^2 + \|\mathbf{q}_i\|_{\mathcal{H}}^2 - 2\langle \mathbf{r}_i, \mathbf{q}_i \rangle_{\mathcal{H}} \\ &= \|\mathbf{r}_i\|_{\mathcal{H}}^2 + \|\mathbf{q}_i\|_{\mathcal{H}}^2 - 2\langle \mathbf{r}_{i+1} + \mathbf{q}_i, \mathbf{q}_i \rangle_{\mathcal{H}} \\ &= \|\mathbf{r}_i\|_{\mathcal{H}}^2 + \|\mathbf{q}_i\|_{\mathcal{H}}^2 - 2\langle \mathbf{q}_i, \mathbf{q}_i \rangle_{\mathcal{H}} \\ &= \|\mathbf{r}_i\|_{\mathcal{H}}^2 - \|\mathbf{q}_i\|_{\mathcal{H}}^2 \end{aligned}$$

From Prop 10, $\|\mathbf{r}_{i+1}\|_{\mathcal{H}}^2 \leq \|\mathbf{r}_i\|_{\mathcal{H}}^2 - \frac{\langle \mathbf{z}_i, \mathbf{r}_i \rangle_{\mathcal{H}}^2}{\|\mathbf{z}_i\|_{\mathcal{H}}^2}$. Since $\langle \mathbf{z}_i, \mathbf{r}_i \rangle_{\mathcal{H}}^2 = 0$ iff $\mathbf{r}_i = \mathbf{r}^*$ the sequence $\{\|\mathbf{r}_i\|_{\mathcal{H}}\}$ monotonically decreases with increasing i until it converges to the lower bound $\|\mathbf{r}^*\|_{\mathcal{H}}$. \square

A.2. Proof of Theorem 3

Our proofs rely on the Gram matrix $\mathbf{G}(\mathcal{J})$ of the atoms in \mathcal{S} indexed by $\mathcal{J} \subseteq [n]$, i.e., $(\mathbf{G}(\mathcal{J}))_{i,j} := \langle \mathbf{s}_i, \mathbf{s}_j \rangle_{\mathcal{H}}, i, j \in \mathcal{J}$.

To prove Theorem 3, we use the following known results.

Lemma 13 (Tropp (2004)). *The smallest eigenvalue $\lambda_{\min}(\mathbf{G}(\mathcal{J}))$ of $\mathbf{G}(\mathcal{J})$ obeys $\lambda_{\min}(\mathbf{G}(\mathcal{J})) > 1 - \mu(m-1)$, where $m = |\mathcal{J}|$.*

Lemma 14 (DeVore and Temlyakov (1996)). *For every index set $\mathcal{J} \subseteq [n]$ and every linear combination \mathbf{p} of the atoms in \mathcal{S} indexed by \mathcal{J} , i.e., $\mathbf{p} := \sum_{j \in \mathcal{J}} v_j \mathbf{s}_j$, we have $\max_{j \in \mathcal{J}} |\langle \mathbf{p}, \mathbf{s}_j \rangle_{\mathcal{H}}| \geq \frac{\|\mathbf{p}\|_{\mathcal{H}}^2}{\|\mathbf{v}\|_1} = \frac{\langle \mathbf{v}, \mathbf{G}(\mathcal{J}) \mathbf{v} \rangle_2}{\|\mathbf{v}\|_1}$, where $\mathbf{v} \neq \mathbf{0}$ is the vector having the v_j as entries.*

Proof of Theorem 3. Note that $\mathbf{m}_i := \mathbf{r}_i - \mathbf{r}^* = (\mathbf{y} - \mathbf{x}^{(i-1)}) - (\mathbf{y} - \mathbf{x}^*) = \mathbf{x}^* - \mathbf{x}^{(i-1)} \in \text{lin}(\mathcal{S})$, $\forall i$, by assumption, which implies that there exists a vector $\mathbf{v}_i \neq \mathbf{0}$ s.t. $\mathbf{m}_i = \sum_{j \in [n]} (\mathbf{v}_i)_j \mathbf{s}_j$. Setting $\mathcal{J} = [n]$, we have

$$\begin{aligned}
 |\langle \mathbf{z}_i, \mathbf{r}_i - \mathbf{r}^* \rangle_{\mathcal{H}}| &\stackrel{\text{by def. of } \mathbf{z}_i}{=} \left| \max_{j \in [n]} (\langle \mathbf{s}_j, \mathbf{r}_i \rangle_{\mathcal{H}} - \underbrace{\langle \mathbf{s}_j, \mathbf{r}^* \rangle_{\mathcal{H}}}_{=0}) \right| \\
 &\stackrel{\text{symmetry of } \mathcal{S}}{=} \max_{j \in [n]} |\langle \mathbf{s}_j, \mathbf{r}_i \rangle_{\mathcal{H}} - \langle \mathbf{s}_j, \mathbf{r}^* \rangle_{\mathcal{H}}| \\
 &= \max_{j \in [n]} |\langle \mathbf{s}_j, \mathbf{m}_i \rangle_{\mathcal{H}}| \\
 &\stackrel{\text{Lemma 14}}{\geq} \frac{\|\mathbf{m}_i\|_{\mathcal{H}}^2}{\|\mathbf{v}_i\|_1} \\
 &\geq \frac{\|\mathbf{m}_i\|_{\mathcal{H}}^2}{\sqrt{n} \|\mathbf{v}_i\|_2} \\
 &= \frac{(\sqrt{\langle \mathbf{v}_i, \mathbf{G}(I) \mathbf{v}_i \rangle_2}) \|\mathbf{m}_i\|_{\mathcal{H}}}{\sqrt{n} \|\mathbf{v}_i\|_2} \\
 &\stackrel{\text{Lemma 13}}{\geq} \sqrt{\frac{1 - \mu(n-1)}{n}} \|\mathbf{r}_i - \mathbf{r}^*\|_{\mathcal{H}}.
 \end{aligned}$$

Replacing the second term in (8) in the proof of Theorem 11 with the last expression above, and noting that $\|\mathbf{z}_i\|_{\mathcal{H}} = 1$, we get the desired result.

To get the result in terms of $\mu(1)$, note that $1 - \frac{1 - \mu(n-1)}{n} \leq (1 - 1/n)(1 + \mu(1))$. \square

A.3. Proof of Corollary 4

Proof. The proof is similar to that of Theorem 3 with $\mathbf{r}^* = \mathbf{0}$ (which implies that \mathbf{r}_i lies in $\text{lin}(\mathcal{S})$). \square

A.4. Inexact LMO

Instead of solving the LMO in each iteration exactly (which may be prohibitively expensive), it is often more realistic to obtain a $\delta_i \in (0, 1]$ approximate solution to the LMO at iteration i . In other words, in each iteration our update is $\tilde{\mathbf{z}}_i$ instead of \mathbf{z}_i so that the following holds (for simplicity all other notations including those of the resid-

ual vectors are overloaded)

$$\langle \tilde{\mathbf{z}}_i, \mathbf{r}_i \rangle_{\mathcal{H}} \geq \delta_i \langle \mathbf{z}_i, \mathbf{r}_i \rangle_{\mathcal{H}}. \quad (10)$$

Note that the α update in each iteration is still exact. To make the effect of the inexact LMO on the rate explicit, we assume that the atoms are normalized according to $\|\mathbf{s}\|_{\mathcal{H}} = c, \forall \mathbf{s} \in \mathcal{S}$, for some constant $c > 0$. We emphasize that linear convergence guarantees can be obtained for the inexact LMO even without this assumption. Proceeding as in the proof of Proposition 10, we get a slightly weaker lower bound for $\|\mathbf{q}_i\|_{\mathcal{H}}^2$, namely

$$\|\mathbf{q}_i\|_{\mathcal{H}}^2 \geq \frac{\langle \tilde{\mathbf{z}}_i, \mathbf{r}_i \rangle_{\mathcal{H}}^2}{\|\tilde{\mathbf{z}}_i\|_{\mathcal{H}}^2} \geq \delta_i^2 \frac{\langle \mathbf{z}_i, \mathbf{r}_i \rangle_{\mathcal{H}}^2}{\|\mathbf{z}_i\|_{\mathcal{H}}^2},$$

where we used $\|\tilde{\mathbf{z}}_i\|_{\mathcal{H}} = \|\mathbf{z}_i\|_{\mathcal{H}}$. We now obtain the following.

Theorem 15 (Linear convergence with inexact LMO). *If the LMO in Algorithm 1 is solved within accuracy δ_i as in (10), then Algorithm 1 converges with a linear rate.*

Proof. We proceed as in the proof of Theorem 1 to get

$$\begin{aligned}
 \|\mathbf{r}_{i+1} - \mathbf{r}^*\|_{\mathcal{H}}^2 &\leq \|\mathbf{r}_i - \mathbf{r}^*\|_{\mathcal{H}}^2 - \delta_i^2 \langle \mathbf{z}_i, \mathbf{r}_i \rangle_{\mathcal{H}}^2 \\
 &= \|\mathbf{r}_i - \mathbf{r}^*\|_{\mathcal{H}}^2 - \delta_i^2 \frac{\langle \mathbf{z}_i, \mathbf{r}_i - \mathbf{r}^* \rangle_{\mathcal{H}}^2}{\|\mathbf{z}_i\|_{\mathcal{H}}^2} \\
 &= \left(1 - \delta_i^2 \frac{\langle \mathbf{z}_i, \mathbf{r}_i - \mathbf{r}^* \rangle_{\mathcal{H}}^2}{\|\mathbf{z}_i\|_{\mathcal{H}}^2 \|\mathbf{r}_i - \mathbf{r}^*\|_{\mathcal{H}}^2} \right) \|\mathbf{r}_i - \mathbf{r}^*\|_{\mathcal{H}}^2,
 \end{aligned}$$

from which the result follows. \square