Global Optimality of Sparse Dictionary Learning with Applications to Subspace Clustering

by

Imke Mayer

A thesis submitted to École Normale Supérieure Paris Saclay in partial fulfillment of the requirements for the degree of Master of Science.

Master's program "Mathématiques Vision Apprentissage"

Center for Imaging Science
Johns Hopkins University, Baltimore

September 2018

Abstract

Most signal processing applications require a good choice of representation of the data for numerous reasons. For instance, classical tasks such as compression, denoising or classification can be performed more easily when the data is represented in terms of certain bases that exploit special structure in high-dimensional data. One specific representation is obtained by decomposing the data in terms of a dictionary, whose elements are either predefined such as wavelets or learned directly from the data. When the representation is assumed to be sparse, i.e., when each data point is a linear combination of a few elements from the dictionary, the latter problem is called Sparse Dictionary Learning (SDL). This class of data modeling methods exploits certain structure in the data which can be formulated as: *the data lies in a union of low-dimensional subspaces* and therefore the dictionary elements, called *atoms*, can be seen as basis elements of these subspaces. Since the dictionary is typically overcomplete, the atoms are actually dependent and should be considered as frames for the subspaces that capture the main directions in terms of compact representations. While a variety of efficient algorithms that give approximate solutions for the SDL problem have been proposed, the theoretical analysis of this problem remains a key challenge due to its non-convexity. In this work we present a step towards a theoretical understanding of global optimality for the SDL problem. In particular, we show that by regularizing the SDL problem with a certain function that penalizes the number of dictionary atoms it is possible to derive conditions under which an optimal dictionary can be found. We also demonstrate this analysis with an application of SDL to the problem of Subspace Clustering.

Primary Reader and Advisor: Prof. René Vidal
Secondary Readers: Dr. Benjamin Haeffele, Dr. Benjamín Béjar Haro

Acknowledgments

This thesis would not have been possible without the guidance of my supervisors Prof. René Vidal and Dr. Benjamin Haeffele. I would like to thank you for giving me the opportunity to contribute to your project.

Furthermore, I would like to say thank you to Connor Lane who provided me with valuable advices on theoretic as well as more technical aspects of this work.

That said, I would like to give thanks to the entire Vision, Dynamics and Learning Lab for giving me the chance to gain insights into their research projects and into their day-to-day work as researchers in a multidisciplinary environment.

Contents

CONTENTS

1 Introduction

Modeling data as sparse linear combinations of some basis elements is a widely used approach to deal with complex data and is motivated by different applications in various domains. For example in signal processing, sparse representations can allow for better results in signal denoising and compression [10, 27], machine learning sparse representations can lead to better results in classi cation [26, 37], and in statistics sparse modeling can be used for variable selection resulting in more interpretable statistical models [35]. The main assumption and goal of such models can be described as follows: the relevant information contained in the (often high-dimensional) data often lies in a low-dimensional structure and nding this low-dimensional structure can therefore allow for more ef cient, robust and interpretable data processing. If one assumes that the data can be modeled by a linear low-dimensional structure, i.e. by one or more linear subspaces, the model which allows recovering such structure is called structured matrix factorization . In this section, we give the formal de nition of structured matrix factorization, some of its well-known applications and we cast a speci c class of sparse models called sparse dictionary learning as a special case of structured matrix factorization. In sparse dictionary learning each data point is decomposed as a sparse linear combination of elements from a dictionary, and both the dictionary and the sparse representations are learned from the data. We will discuss its applicability to another special case of structured matrix factorization, the problem of subspace clustering .

1.1 Problem formulation

Let $X \in \mathbb{R}^{D \times N}$ be the data matrix which can be considered as the concatenation of N vectorized signals $\{x_i, \ldots, x_N\} \subset \mathbb{R}^D$. Under the linear data model, i.e. the assumption that the data can be approximated by one or more linear subspaces, structured matrix factorization consists in decomposing the data X into two factors U and V in a way that translates or exploits the low-dimensionality assumptions on the data. More formally, given a loss function ℓ which measures the quality of approximation of the factorization UV^T w.r.t. the initial data X and a regularization function ψ which enforces specific structure on the factors U and V, we can write the structured matrix factorization problem as:

$$\min_{\substack{U \in \mathbb{R}^{D \times r} \\ V \in \mathbb{R}^{N \times r}}} \ell(X, UV^T) + \psi(U, V), \tag{1.1}$$

where we usually assume $r < \min\{D, N\}$. The interest of structured matrix factorization is two-fold: especially in the case of high-dimensional data it can considerably reduce the representation complexity of the data (i.e. $O((D+N)r)$ instead of $O(DN)$). Furthermore, it can allow to separate the information contained in the data into the different factors by enforcing certain constraints on each factor, such as spatial and temporal information in the case of videos as demonstrated in [18] and therefore reveal interesting patterns in the data. The generic form of (1.1) allows to derive other possible applications and to cast other well-known models as special cases of this problem.

We will now turn to the problem of sparse dictionary learning and explain the choices of loss and regularization function for (1.1) in this case. Given a set of N signals $\{x_1, \ldots, x_N\} \subset \mathbb{R}^D$, the dictionary learning problem consists of finding a set of dictionary atoms $\{d_1, \ldots, d_r\} \subset \mathbb{R}^D$ and N sparse codes $\{a_1, \ldots, a_N\} \subset \mathbb{R}^r$ that allow to recover, either exactly or approximately, the initial set of signals (which can be one-dimensional but also vectorized forms of multi-dimensional data such as images or videos). In other words, it requires solving the following problem:

$$\min_{\substack{d_1,\dots,d_r \in \mathbb{R}^D \\ a_1,\dots,a_N \in \mathbb{R}^r}} \frac{1}{2} \sum_{i=1}^{N} \left\| x_i - \sum_{j=1}^{r} (a_i)_j d_j \right\|_2^2 + \lambda \, \Omega(\{d_k\}_{k=1}^r, \{a_k\}_{k=1}^r); \qquad (1.2)$$

where Ω is defined on the dictionary atoms and the sparse codes and $\lambda > 0$ is a parameter adjusting the trade-off between data fidelity and regularization.

In matrix form this problem can be rewritten as

$$\min_{\substack{D \in \mathbb{R}^{D \times r} \\ A \in \mathbb{R}^{r \times N}}} \frac{1}{2} \|X - DA\|_F^2 + \lambda \, \Omega(D, A); \qquad (1.3)$$

where the r columns of D are the dictionary atoms and the N columns of A contain the sparse representation of the data $X = [x_1 \dots x_N] \in \mathbb{R}^{D \times N}$ in terms of the dictionary D.

Without any regularization, i.e. if $\lambda = 0$, this problem can be an ill-posed signal reconstruction problem. For instance, given a signal $x \in \mathbb{R}^D$ and a dictionary D with non-trivial null-space, there are infinitely many solutions $a \in \mathbb{R}^r$ to the problem $\min_a \frac{1}{2}\|x - Da\|_2^2$. But assuming that the decomposition of the data into D and A is such that each signal only "uses" very few atoms, i.e. the columns of A are sparse, and that the dictionary is overcomplete ($r > D$), the problem becomes the Sparse Dictionary Learning problem (SDL), first introduced in [28] to efficiently process natural image patches. Hence the regularization function Ω can be considered to be defined on the $\ell_{1,1}$ norm of matrix A, i.e. on the sum of the absolute values of all its entries, since it is well known that the ℓ_1 norm is a sparsity-inducing convex relaxation of the pseudo-norm ℓ_0. The resulting decomposition of the data into dictionary and codes is illustrated in Figure 1.1.

Another well-known class of problems which exploits (and recovers) a certain structure of the – often high-dimensional – data and which can be cast as a matrix factorization model is Subspace Clustering (SC). For this class of problems the goal is to recover a low-dimensional structure which the data comes

Figure 1.1: Sparse dictionary learning model.

from, more speci cally a union of multiple low-dimensional subspaces, allowing for a more robust and compact representation of the data. Since this is a combinatorial problem which is NP-hard, current subspace clustering methods (see [36] for a review) only seek to nd a possible representation of the subspaces (a large dictionary with linearly dependent atoms) and to identify which subspace a data point comes from, based on such a dictionary. The subspaces' representation being redundant, one cannot hope for a unique sparse representation.

While the applications of structured matrix factorization, and of sparse dictionary learning and subspace clustering as special cases, are numerous, the main issue is that it is a non-convex problem due to the matrix multiplication in the data delity term of the objective function. Therefore most optimization strategies require good heuristics for the choice of initialization and there are no guarantees on global optimality. But if we keep the dictionary D in (1.3) xed and assume that is convex w.r.t. A, the problem reduces to a regularized least squares problem in A and is therefore a convex problem (for speci c choices of such as the ℓ_1 norm numerous ef cient algorithms solving this lasso-type problem, known as sparse coding, have been proposed [9, 25]). However, analyzing and solving the joint optimization problem is much more challenging and is still an open eld of research. Motivated by the work of [4] and

more speci cally [18] who proposed a general matrix factorization framework to characterize global optimality and several explicit methods to nd globally optimal solutions in certain cases of structured matrix factorization, we follow the same line of approach to analyze the non-convex SDL problem.

1.2 Related work

MATRIX FACTORIZATION AND SPARSE DICTIONARY LEARNING

The analysis of the optimization landscape of non-convex decomposition or signal recovery problems has recently gained much attention due to the recurring empirical observation that many apparently hard problems can be reasonably well solved by approximate local strategies which do not come with a guarantee to converge to a “good” solution of non-convex problems but only to some local optima or critical points. This suggests that the optimization landscape of these problems is favorable in the sense that there are no spurious local minima or bad saddle points and therefore any descent strategy can achieve a good result without getting stuck in a bad critical point.

For the case of non-convex low-rank matrix optimization problems [38] showed that there are no spurious local minima and extend the concept of strict saddle property introduced by [13] implying that most optimization strategies such as gradient descent converge to a global solution from any initialization. The problem of optimal dictionary recovery has been studied by [34] in the complete case ($r = D$) over the sphere and they propose an algorithm that provably recovers an optimal sparsifying dictionary, again by using the argument that there are no spurious local minima. Extending the ideas of [17,18], the authors of [32] propose a method with guarantees on global optimality to solve the separable dictionary learning problem applied to tensors. Another approach to (approximately) solve such non-convex matrix factorization problems is to introduce some convex relaxation (such as the nuclear norm in the low-rank matrix factorization case, [6]) to transform the factorization problem into a convex

matrix approximation problem. Convex relaxations are an appealing solution from the methodological point of view since they allow an easier analysis of problem and its optimal solutions and can often be solved by using standard optimization tools (ef cient MATLAB implementations are provided for instance by [15]). A caveat however is that these formulations often do not scale to larger dataset, i.e. their application to large scale datasets is challenging and often inadvisable from a practical point of view. Furthermore the use of (tight) convex relaxations cannot capture the entire modeling exibility of the structured matrix factorization formulation. For instance, certain structures on the factors U and V, such as non-negativity [22], cannot be promoted by regularizing their product UV^T.

SUBSPACE CLUSTERING

Many different approaches have been proposed for solving the subspace clustering problem, ranging from a generalization of the well-known k-means to the multiple subspace setting, to algebraic, statistical and spectral methods (see [36] for a review). Among the spectral methods the Sparse Subspace Clustering (SSC, [11]) is particularly ef cient (it only requires to solve a convex $`_1$ minimization problem) and robust to different types of contaminated data such as noise, missing data and sparse outliers. The problem of sparse subspace clustering is also non-convex but under certain assumptions on the data distribution and subspace con guration the solution to the convex surrogate problem induced by the $`_1$ norm provides the desired sparse representation of the data. Another assumption which can be made is that the data allows a low-rank representation [23] and in the case of noise-free data drawn from independent subspaces, the optimal solution to the corresponding optimization problem can be written in closed form from the SVD of the data. These methods consider the data as a self-expressive dictionary, the factorization of the initial data X into two factors therefore does not come with any (memory) complexity reduction. [3] however propose a subspace clustering method which is

based on a smaller dictionary and the clustering is then made possible through bipartite graph modeling.

1.3 Contributions

In this work we present a theoretical and experimental study of the problem of structured matrix factorization in the case of sparse dictionary learning. We focus on the problem of solving an auxiliary optimization problem, the polar problem, which serves as a certificate of global optimality for the initial SDL problem. We give some intuitions and first steps towards a theoretical analysis of the polar problem for the sparse dictionary learning problem and discuss the impact of approximation errors on the initial problem. We also experimentally demonstrate its relevance for solving the overall matrix factorization formulation, especially for recovering the "generating" factorization size on data drawn from a union of subspaces.

In Chapter 2 we provide the theoretical analysis of global optimality for the SDL problem, discuss and analyze the challenges of verifying the derived global optimality conditions and propose an iterative algorithm for SDL based on the previous analysis. In Chapter 3 we draw a connection of our SDL model to existing subspace clustering models to motivate a novel approach to subspace clustering via our SDL matrix factorization formulation.

Notations

In the next chapters, we will use the following fairly standard notations: $[n]$ denotes the set of positive integers up to n, $\{1, \dots, n\}$; for a set S we denote its complement by $S^c = [n] \setminus S$; scalars are denoted by regular lower case letters, vectors by bold lower case letters; matrices are denoted by capital letters. Columns and rows of a matrix $X \in \mathbb{R}^{m \times n}$ are referenced by X_j with $j \in [n]$ and $X_{\langle i \rangle}$ with $i \in [m]$ respectively. For a scalar a, we define $(a)_+ := \max(a, 0)$.

Unless speci ed differently, all vectors are considered as column vectors. The notation $k\ k$ will be used indifferently for both vector and matrix norms. For instance $k\ k_F$ denotes the Frobenius norm on matrices, $k\ k_{k!\ l}$ the $k\ !\ l$ operator norm and $k\ k_{k;l}$ the $L_{k;l}$ norm ($k; l\ 2\ N$).

2 Global Optimality for Sparse Dictionary Learning

In this chapter we will analyze the problem of Sparse Dictionary Learning as a particular case of regularized matrix factorization problem and characterize its global optima. Based on these characterizations of global optimality we propose an iterative algorithm for solving the non-convex optimization problem. We will focus on an auxiliary optimization problem which arises in these characterizations of global optima. We empirically assess the difficulty of solving this problem, propose a conjecture on its optimization landscape and present initialization and optimization strategies which empirically lead to good approximate solutions.

2.1 Review of Conditions for Global Optimality in Structured Matrix Factorization

We recall the standard formulation of sparse dictionary learning which we want to analyze as a special case of the structured matrix factorization problem.

$$\min_{\substack{U \in \mathbb{R}^{D \times r} \\ V^T \in \mathbb{R}^{r \times N}}} \frac{1}{2}\|X - UV^T\|_F^2 + \|V\|_{1,1} \quad \text{subject to } \|U_i\|_2 \leq 1 \text{ for all } i \in [r]: \tag{2.1}$$

The objective is a sum of a differentiable loss function and a convex but non-differentiable regularization function. The loss function is convex w.r.t. the

product UV^T and w.r.t. each factor but it is not jointly convex in (U, V), making the overall optimization problem non-convex. Minimization is defined over U and V which are of fixed size, i.e. the size r of the factorization is part of the problem dimensions $(D, N, r) \in (\mathbb{N}_+)^3$.

Before we derive our sparse dictionary learning model from the above formulation and delve into its analysis we begin by defining some relevant notations and quantities, following the work of [17, 18].

Definition 1 (Rank-1 regularizer, [18]). A function $\Omega : \mathbb{R}^D \times \mathbb{R}^N \to \mathbb{R}_+ \cup \{\infty\}$ is said to be a rank-1 regularizer if

1. $\Omega(u, v)$ is positively homogeneous with degree $p > 0$, i.e. $\Omega(\alpha u, \alpha v) = \alpha^p \Omega(u, v), \forall \alpha \geq 0, \forall (u, v) \in \mathbb{R}^D \times \mathbb{R}^N$.
2. $\Omega(u, v)$ is positive semi-definite, i.e. $\Omega(0, 0) = 0$ and $\Omega(u, v) \geq 0$ for all (u, v).
3. For any sequence (u_n, v_n) such that $\|u_n v_n^T\| \to \infty$ we have that $\Omega(u_n, v_n) \to \infty$.

Definition 2 (Elemental mapping, r-element factorization mapping, [18]). An elemental mapping $\Phi : \mathbb{R}^D \times \mathbb{R}^N \to \mathbb{R}^{D \times N}$ is any mapping which is positively homogeneous with degree $p > 0$. The r-element factorization mapping $\Phi_r : \mathbb{R}^{D \times r} \times \mathbb{R}^{N \times r} \to \mathbb{R}^{D \times N}$ is defined as $\Phi_r(U, V) = \sum_{i=1}^{r} \Phi(U_i, V_i)$.

Unless specified differently we will use $\Phi : (u, v) \mapsto uv^T$ and $\Phi_r : (U, V) \mapsto UV^T = \sum_i U_i V_i^T$ as elemental mapping and r-element factorization mapping respectively.

These definitions can be extended to higher dimensions, for instance to analyze more general tensor decomposition problems as detailed in [18], but here we limit the presentation to the matrix factorization case as it is sufficient to analyze the problem of sparse dictionary learning.

Definition 3 (Matrix factorization regularizer, [18]). Given a rank-1 regularizer which is positively homogeneous with degree 2, the matrix factorization

regularizer $: \mathbb{R}^{D \times N} \to \mathbb{R}_+ \cup \{+\infty\}$ is defined as

$$\Omega(Y) = \inf_{r \in \mathbb{N}} \inf_{\substack{U \in \mathbb{R}^{D \times r} \\ V \in \mathbb{R}^{N \times r}}} \sum_{i=1}^{r} \omega(u_i, v_i) \quad \text{subject to } Y = UV^T. \tag{2.2}$$

An important remark on this last definition is that such a matrix factorization regularizer will be convex w.r.t. Y. And we can also note that if ω is such that $\omega(-u, v) = \omega(u, v)$ or $\omega(u, -v) = \omega(u, v)$ for all (u, v), then $\|\cdot\|$ is a norm on X, called atomic norm or decomposition norm. For instance, taking $\omega(u, v) = \|u\|_2 \|v\|_2$, this definition corresponds to the variational form of the nuclear norm $\|\cdot\|_*$ for matrices which is defined as the sum of the singular values of a matrix.

These definitions allow to introduce the framework proposed in [18] with the following key idea: the matrix factorization regularizer allows to tightly couple the structured matrix factorization problem which is non-convex in (U, V) to a problem that is convex in Y. Exploiting the convexity of the latter allows to analyze the former and to derive conditions of globally optimality for it. An important aspect which allows this coupling is to consider the factorization size r as one of the variables and therefore to optimize over factorizations of all possible sizes $r \in \mathbb{N}_+$.

We can now go on to analyze how the sparse dictionary learning problem can be cast into this matrix factorization framework in order to give global optimality conditions for the non-convex sparse dictionary learning problem. As pointed out in [5], constraining the dictionary atoms in problem (2.1) to have ℓ_2-norm less than 1 is equivalent to penalizing their ℓ_2-norm by integrating these terms in the regularization term of the objective function. Indeed, this becomes clear if we write the variational form of the $\ell_{2,1}$ norm of some matrix $M = [M_1 \, \vdots \, M_N]$

$$\|M\|_{2,1} = \sum_{i=1}^{N} \|M_i\|_2 = \frac{1}{2} \min_{\lambda_i \geq 0} \sum_{i=1}^{N} \frac{\|M_i\|_2^2}{\lambda_i} + \lambda_i; \tag{2.3}$$

which is obtained by a simple application of the Cauchy-Schwarz inequality. Additionally we will allow the factorization size r to vary, since we argued earlier that this will allow to couple this non-convex factorization problem to a closely related convex problem. This leads to a first formulation:

$$\min_{\substack{U \in \mathbb{R}^{D \times r} \\ V \in \mathbb{R}^{N \times r} \\ r \in \mathbb{N}_+}} \frac{1}{2}\|X - UV^T\|_F^2 + \sum_{i=1}^{r} \|U_i\|_2 \|V_i\|_1 \tag{2.4}$$

Observe that the factorization of a matrix Y as the product UV^T is not unique due to a rotational ambiguity, i.e. if $(U, V) \in \mathbb{R}^{D \times r} \times \mathbb{R}^{N \times r}$ are such that $Y = UV^T$, then for any orthogonal matrix $R \in O(r)$, we have $Y = URR^TV^T$. The regularizer can eliminate this ambiguity by enforcing special structure on the factors U and V, which in turn enforces special structure on the column and row spaces of UV^T, respectively. For instance, using the ℓ_1 norm on the columns of V, i.e. in the row space, the decomposition of the data X over the dictionary will be very sparse and the dictionary will consequently be large since we optimize over all possible factorization sizes. Conversely, using the ℓ_2 norm on the columns of V will lead to a compact dictionary but the codes (rows of V) will not be sparse. In order to define a trade-off between these two boundary cases we modify the regularization function and propose to solve the alternative problem:

$$\min_{\substack{U \in \mathbb{R}^{D \times r} \\ V \in \mathbb{R}^{N \times r} \\ r \in \mathbb{N}_+}} \underbrace{\frac{1}{2}\|X - UV^T\|_F^2}_{\ell(X;\, r(U,V))} + \underbrace{\sum_{i=1}^{r} \|U_i\|_2 (\|V_i\|_1 + (1-\lambda)\|V_i\|_2)}_{\Omega(U,V)} \tag{2.5}$$

Again we draw the attention to the fact that in the above formulation minimization is carried out over the factors and the factorization size r. This implies that the optimality conditions we will present next are also defined on the factorization size, superseding the model selection step that requires the use of some heuristics in most applications to define a fixed size of the factors prior to solving the factorization problem. Instead, the size of the factorization

is data-driven through the use of the regularization function .

The loss term of the objective function (2.5) remains unchanged from the previous formulations (2.1) and (2.4). But we introduce a new regularization function that depends on the parameter 2 [0; 1]. Observe that the new regularization function is de ned as the sum of a rank-1 regularizer over the columns of U and V. Indeed we easily see that

$$: (u; v) \in R^D \quad R^N \mapsto \|u\|_2(\ \|v\|_1 + (1 \quad)\|v\|_2) \in R_+$$

is positively homogeneous with degree 2, positive semi-de nite and such that for any sequence $(u_n; v_n)$ such that $\|u_n v_n^T\| \to 1$, we have $(u_n; v_n) \to 1$ (since for all $n \in N$, $\|u_n v_n^T\| = \|u_n\|_2\|v_n\|_2 \ \|u_n\|_2\|v_n\|_1$ and therefore $\|u_n v_n^T\| \ (u_n; v_n)$).

We remark that the regularization function is closely related to the matrix factorization regularizer (2.2), also referred to as decomposition norm or atomic norm in the sense that we always have

$$\underbrace{`(X; Y) + \quad (Y)}_{=: F(Y)} \quad \underbrace{`(X; \ _r(U; V))}_{=: f(U;V)} + \quad (U; V) \tag{2.6}$$

$\forall (U; V)$ such that $Y = \ _r(U; V) = UV^T$.

The main result of [18] makes use of this global lower bound of the non-convex objective function f (in the factor space) by the convex function F (in the product space) to prove the following results:

Theorem 1 (Theorem 1 in [18]) . Given a function $`(X; Y)$ that is convex in Y and once differentiable w.r.t. Y, a rank-1 regularizer and a constant > 0, local minima $(U; V)$ of (2.5) are globally optimal if $(U_i; V_i) = (0; 0)$ for some $i \in [r]$. Moreover, $\hat{Y} = \ _r(U; V) = UV^T$ is global minima of $F(Y)$ and UV^T is an optimal factorization of $\hat{Y}$, i.e. it achieves the in mum in the de nition (2.2) of the atomic norm $(\hat{Y})$.

This theorem roughly states that under the given assumptions any local

minima of f which is sufficiently large is a global minima. And the following corollary suggests a way to verify that a given point is a local minima and sufficiently large, hence a global minima.

Corollary 1 (Corollary 1 in [18]) . Under the same assumptions as in the previous theorem, a local minima $(U^*; V^*)$ of $f(U; V)$ is globally optimal if it satisfies:

1. $u_i^T \in \partial_1 r_{\lambda}(X; \sigma_r(U^*; V^*))\, v_i^* = \partial(u_i; v_i)$ for all $i = 1; \ldots; r$

2. $u^T \in \partial_1 r_{\lambda}(X; \sigma_r(U^*; V^*))\, v \subseteq \partial(u; v)$ for all $(u; v)$.

It can be shown that for our choice of rank-1 regularizer, Ω, any first-order optimal point of (2.5) satisfies condition 1 [18, Proposition 3] [1]. The second condition implies that at $(U^*; V^*)$, the addition of any new direction $(u; v)$, i.e. $(U; V) \leftarrow ([U^* \; \alpha u]; [V^* \; \beta v])$ for some $\alpha > 0$, will only increase the objective function f. This can be seen more easily by rearranging the terms of the inequality above:

$$u^T \in \partial_1 r_{\lambda}(X; \sigma_r(U^*; V^*))\, v \subseteq \partial(u; v), \quad 0 \leq \alpha \in r_{\lambda}(X; \sigma_r(U^*; V^*)); uv^T \rangle + \partial(u; v);$$

where the right hand-side of the second equivalent term corresponds to the directional derivative of the objective function in the direction $(u; v)$. Therefore if condition 2 of the corollary holds we have $f([U^* \; \alpha u]; [U^* \; \beta v]) \geq f(U^*; V^*)$, since by using the convexity of the loss ℓ w.r.t. to its second argument we have:

$$\begin{aligned} f([U^* \; \alpha u]; [V^* \; \beta v]) &= \ell(X; U^* V^{*T} + \alpha uv^T) + \sum_{i=1}^{r} \partial(u_i^*; v_i^*) + \partial(u; v) \\ &\geq \ell(X; U^* V^{*T}) + \alpha \langle r_{\lambda}(X; \sigma_r(U^*; V^*)); uv^T \rangle \\ &\quad + \sum_{i=1}^{r} \partial(u_i^*; v_i^*) + \partial(u; v) \\ &\geq f(U^*; V^*). \end{aligned}$$

[1] This proposition actually covers a broader class of possible rank-1 regularizers that encompasses the product of norms as a special case.

If condition 2 is violated, i.e. we find a pair $(u;v)$ that does not satisfy the inequality in Corollary 1, we have found a descent direction that allows to decrease the objective by exactly $\text{hr}_r\ell(X;\ _r(D;W));uv^T\text{i} + \ (u;v)$ where is some (small) positive step-size.

The authors of [18] point out that condition 2 of the corollary can be verified by evaluating the so-called polar function at $Z = \frac{1}{}r_r\ell(X;\ _r(D;W))$ and testing whether (Z) 1 where (Z) is defined as

$$(Z) = \sup_{u;v} \ u^T Z v \quad \text{s.t.} \quad (u;v) \quad 1: \tag{2.7}$$

Indeed, this becomes clear when writing out the following equivalences:

$$u^T Z v \quad (u;v) \quad \text{for all } (u;v) \tag{2.8}$$

$$, \quad \frac{u^T Z v}{(u;v)} \quad 1 \quad \text{for all } (u;v) \tag{2.9}$$

$$, \quad \sup_{\substack{(u;v)\\ (u;v)\ 1}} u^T Z v \quad 1 \tag{2.10}$$

Solving this last problem is referred to as solving the polar problem associated to the function .

Before we discuss this auxiliary optimization problem in more detail we first cite an algorithm proposed in [18] and built upon the above results for solving the structured matrix factorization problem. We present the algorithm in the case of the sparse dictionary learning problem.

2.2 Structured matrix factorization algorithm for SDL

We will now present the high-level structure of the SDL matrix factorization algorithm as well as the key elements of each sub-routine, especially the different local optimization strategies.

2.2.1 Meta-algorithm

An adaptation of the proposed structured matrix factorization Meta-Algorithm (Algorithm 1, [18]) for the sparse dictionary learning problem has been implemented by Connor Lane [2] in MATLAB. In what follows we will present the details of this adapted algorithm and contributed modifications to this implementation.

Algorithm 1: SDL Matrix Factorization

Input: data X, initial factorization (U_{init}, V_{init}) of size r_{init}. Result: optimal factorization (U_{final}, V_{final}) of size r_{final}.
while not converged do
  1) Local descent to a first-order optimal point (U, V).
  2) Solve the SDL polar problem at $Z = \frac{1}{r}\nabla_r \ell(X; \sigma_r(U, V))$.
    if polar value ≥ -1 then
      Algorithm has converged.
    else
      Append the polar solution (u, v) to (U, V):
      $(U, V) \leftarrow ([U \ \sqrt{\gamma}u], [V \ \sqrt{\gamma}v])$ for some $\gamma > 0$.
    end
end

The intuition behind the meta-algorithm in [18] is built upon Corollary 1. Specifically, we begin with an initial solution (U, V) of a certain size r. If such a solution is not a local minimum, then we can perform local descent to arrive at a critical point, which needs to satisfy condition 1 of the corollary. We can then verify if condition 2 is satisfied, in which case we know that we cannot further reduce the objective by increasing r. Alternatively, if condition 2 is not satisfied, then we know that there exists a descent direction (u, v) such that if we augment U by $\sqrt{\gamma}u$ and V by $\sqrt{\gamma}v$, where γ is the optimal step size computed in closed form, we can reduce the objective. We can then repeat the process for a factorization of size $r + 1$ until the conditions in Corollary 1 hold, which is guaranteed to happen for finite r, as shown in [18].

[2] VisionLab/Center for Imaging Science, Johns Hopkins University

2.2.2 Details on local optimization strategies

LOCAL DESCENT

Due to the non-differentiability of the regularization term , we cannot use optimization methods that require smoothness of the cost function such as simple gradient descent. However, it is possible to compute the proximal operator of w.r.t. to U and w.r.t. V, which allows us to de ne an alternating proximal gradient descent strategy to minimize the following function

$$\begin{aligned} h(U;V) &= \ell(X;\ _r(U;V)) + \quad (U;V) \\ &= \frac{1}{2}\|X \quad UV^T\|_F^2 + \sum_{i=1}^{r} \|U_i\|_2(\ \|V_i\|_1 + (1 \quad)\|V_i\|_2) \end{aligned} \tag{2.11}$$

with respect to U and V, i.e. the factorization size r is kept xed in this step. Using the de nition and basic properties of the proximal operator we have:

For all V, the proximal operator of the continuous convex mapping $U \mapsto (U;V)$ is de ned by

$$\begin{aligned} \text{prox}_{\ (\cdot;V)}(W) &= \arg\min_U \quad (U;V) + \frac{1}{2}\|U \quad W\|_F^2 \\ &= \sum_i \quad \arg\min_{U_i} \quad (U_i;V_i) + \frac{1}{2}\|U_i \quad W_i\|_2^2 \\ &= \sum_i \quad \text{prox}_{\ (\cdot;V_i)}(W_i); \end{aligned} \tag{2.12}$$

i.e. requiring the computation of the proximal operator of the ℓ_2 norm.

For all U, the proximal operator of the continuous convex mapping $V \mapsto$

$(U;V)$ is de ned by

$$\begin{aligned}\text{prox}_{(U;\cdot)}(W) &= \arg\min_{V} \quad (U;V) + \frac{1}{2}\|V - W\|_F^2 \\ &= \sum_i \arg\min_{V_i} \quad (U_i;V_i) + \frac{1}{2}\|V_i - W_i\|_2^2 \\ &= \sum_i \text{prox}_{(U_i;\cdot)}(W_i)\end{aligned} \tag{2.13}$$

i.e. requiring the computation of the proximal operator of a weighted sum of ℓ_1 norm and ℓ_2 norm.

From [18, Proposition 5], we know that

$$\text{prox}_{\|\cdot\|_1+(1-)\|\cdot\|_2}(y) = \text{prox}_{(1-)\|\cdot\|_2}(\text{prox}_{\|\cdot\|_1}(y)): \tag{2.14}$$

This relation is useful since we know the proximal operators of the ℓ_1 and ℓ_2 norms:

$$\text{prox}_{\|\cdot\|_2}(y) = \left(1 - \frac{}{\|y\|_2}\right)_+ y \tag{2.15}$$

and

$$(\text{prox}_{\|\cdot\|_1}(y))_i = \text{sign}(y_i)(|y_i| -)_+ \quad 8\, i: \tag{2.16}$$

These remarks on the proximal operators $\text{prox}_{(U;\cdot)}$ and $\text{prox}_{(\cdot;V)}$ allow an ef cient implementation of an alternating proximal gradient descent algorithm to solve $\min_{U;V} h(U;V)$; at each iteration we are guaranteed a decrease of the objective but this alternation scheme comes without theoretical guarantees on convergence.

APPROXIMATION OF THE POLAR VALUE

Ideally, in the previous step we would use a descent strategy which is guaranteed to converge to a local minimum and not only to a critical point. Once we are at a local minimum we know from Theorem 1 that we only need to check whether this local minimum contains a zero slice (some $i \in [r]$ such that

$(U_i; V_i) = (0; 0)$) but unfortunately in the case of non-convex optimization, usual minimization strategies only guarantee convergence to a critical point which can be a local minimum, local maximum or a saddle point. Therefore we need to have an alternative test for optimality which applies to all critical points. From Corollary 1 we know that the polar value tells us whether there exists a descent direction that allows to further decrease the objective or whether we are at the global optimum.

If we substitute in problem (2.7) the de nition of the SDL rank-1 regularizer, we get the following problem:

$$\max_{u;v} \quad u^T Z v \qquad \text{s.t.} \quad \|u\|_2 (\ \|v\|_1 + (1 \quad) \|v\|_2) \quad 1 \qquad (2.17)$$

for some xed matrix Z.

Examining problem (2.17) we remark that we can eliminate one of the variables by noticing that at the optimum we obtain the optimal u as $u = \frac{Zv}{\|Zv\|_2}$. Therefore we have the following equivalent problem(s):

$$\max_{v \in R^N} \quad v^T Z^T Z v \qquad \text{subject to} \qquad \|v\|_1 + (1 \quad) \|v\|_2 \quad 1 \qquad (2.18)$$

$$\max_{v \in R^N} \quad \|Zv\|_2 \qquad \text{subject to} \qquad \|v\|_1 + (1 \quad) \|v\|_2 \quad 1 \qquad (2.19)$$

We note that in some cases it reduces to well known problems which we can solve in closed form:

$= 0$: v = top eigenvector of $H = Z^T Z$.

$= 1$: $v = e_i$ where $i = \arg\max_i \|Z_i\|_2$ and e_j denotes an element of the canonical basis of R^N.

For other cases of , i.e. for all $\in (0; 1)$, however we do not have a closed form solution and need to de ne an optimization scheme to solve the non-convex polar problem. There are several iterative approximation methods we can de ne for one of the equivalent formulations above:

projected power iteration : given a diagonalizable matrix H and an initial vector v_0, the power iteration method is defined as:

$$v_{k+1} = \frac{Hv_k}{\|Hv_k\|_2}$$

and converges to the top eigenvector/eigenvalue of H (under some mild assumption on the initial vector v_0). Similarly to [8], we replace the renormalization in the recurrence relation (the projection onto the Euclidean ball) by a projection on our constraint set to obtain the following relation:

$$v_{k+1} = \frac{Hv_k}{P_C(Hv_k)} \tag{2.20}$$

where $C = \{v : \|v\|_1 + (1-\lambda)\|v\|_2 \leq 1\}$ and the projection onto C, P_C, is obtained by iteratively solving the proximal operator of the mapping $v \mapsto \|v\|_1 + (1-\lambda)\|v\|_2$.

(accelerated) proximal gradient ascent : We can rewrite the polar problem using the indicator function :

$$\max_{v \in \mathbb{R}^N} \|Zv\|_2^2 - \{\|v\|_1 + (1-\lambda)\|v\|_2 \leq 1\} \tag{2.21}$$

and use the proximal gradient algorithm where the proximal operator of $\{\cdot\}_C$ is simply the projection onto the set C. An acceleration can be achieved by adding an extrapolation step which translates as described in [29].

alternating maximization (or generalized power iteration) : Instead of deriving the optimal u from the result v of the optimization over v, we optimize iteratively over both variables and alternate between the two steps

1. $u_{k+1} = \frac{Zv_k}{\|Zv_k\|_2}$
2. $v_{k+1} = \arg\max_v \langle Z^T u_{k+1}, v\rangle$ subject to $\|v\|_1 + (1-\lambda)\|v\|_2 \leq 1$,

where step 2 is a convex problem which can be solved efficiently by bisection search. This alternating maximization is an adaptation from the method proposed in [21] in the context of sparse principal component analysis.

We compared these three strategies on two types of randomly generated matrices: (1) Gaussian matrix and (2) random matrix with a sparse eigen-vector (different from the top eigen-vector). The results are reported in Figures 2.1 and 2.2 and suggest that the proximal gradient algorithm performs best in both scenarios, even for a small number of random initializations. However, repeating this experiment several times, the alternating maximization yields similar performance as the proximal gradient method. The projected power iteration breaks down for in the case of a Gaussian matrix with dense eigenvectors.

Instead of generating a large number of random initializations, another strategy is to use spectral initialization, i.e. take the eigenvectors of $H = Z^T Z$, since we are trying to solve a generalized eigenvalue problem. We do not claim that the solutions to the generalized eigenvalue problem can be expressed in terms of these eigenvectors, but the empirical results we obtain with this initialization strategy on small-scale problems are satisfactory. Indeed on Figure 2.3 we can see that for both proximal gradient and alternating maximization the spectral initialization leads to similar or better results than using as many random initializations.

Following these results we choose to keep the accelerated proximal gradient and alternating maximization and spectral initialization for our meta-algorithm. Once the polar problem is solved, the polar value is compared to 1: if it is larger, then we append the solution (u, v), scaled by a scalar $= \frac{(u)^T(X - \sigma uv^T)v}{\|u\|(v)^T\|_F^2} > 0$ which guarantees a decrease of the objective, to the current factorization.

(a) 50 random initializations.

(b) 100 random initializations.

(c) 250 random initializations.

Figure 2.1: Random Gaussian matrix $H = Z^T Z \in \mathbb{R}^{50 \times 50}$ with small condition number (1:1). The x-axis corresponds to the sparsity trade-off parameter . The y-axis gives the value of the cost function $u^T Z v$. All convergence results are represented as cyan crosses, the largest solution is marked by blue circles. Left: projected power iteration. Middle: accelerated proximal gradient. Right: alternating maximization.

(a) 50 random initializations.

(b) 100 random initializations.

(c) 250 random initializations.

Figure 2.2: Sparse eigenvector matrix $H = Z^T Z \in \mathbb{R}^{50 \times 50}$ with small condition number (1:1). The x-axis corresponds to the sparsity trade-off parameter . The y-axis gives the value of the cost function $u^T Z v$. All convergence results are represented as cyan crosses, the largest solution is marked by blue circles. Left: projected power iteration. Middle: accelerated proximal gradient. Right: alternating maximization.

(a) 50 random initializations.

(b) 100 random initializations.

(c) 250 random initializations.

Figure 2.3: Relative difference between the best solution found with spectral initialization (50 eigenvectors of random $H = Z^T Z$ 2 $R^{50\ 50}$) and the best solution found with random initializations. For each pair of plots, left: accelerated proximal gradient, right: alternating maximization. (red: mean difference; magenta: median difference, taken over 10 experiments).

2.3 Analysis of the polar problem for SDL

In this part we will provide some intuition about the meaning of the polar problem for our overall SDL problem and analyze it as a non-convex optimization problem with a conjectured small number of local minima.

The polar problem that arises in the analysis of the non-convex optimization problem of structured matrix factorization is itself an optimization problem which can be more or less easily solved. Indeed the problem difficulty depends on the rank-1 regularizer Ω. For instance, if $\Omega(u,v) = \|u\|_2\|v\|_2$, then the problem reduces to finding the largest eigenvector of Z^TZ whereas it has been shown that for $\Omega(u,v) = \|u\|_1\|v\|_1$ the problem is NP-hard [20].

The condition on the polar value can be understood as a general higher-order optimality condition in the sense that if we are at a saddle point (of the optimization problem over (U,V,r)) it allows to find a descent direction; it is a higher order non-smooth saddle point problem. For instance in the low-rank matrix factorization case ($\Omega(u,v) = \|u\|_2\|v\|_2$) we can show the following result:

Proposition 1. In the low-rank matrix factorization case,

$$\min_{U,V,r} f(U,V) = \|X - UV^T\|_F^2 + \sum_{i=1}^{r} \|U_i\|_2\|V_i\|_2 \tag{2.22}$$

the condition $\Omega^\circ(Z) \leq 1$ on the polar function at $Z = \frac{1}{r}\partial_r(X, \Omega_r(U,V))$ is equivalent to the second order optimality condition of the initial problem (2.22).

Proof. Let $r = \mathrm{rank}(X)$. We know that problem (2.22) is equivalent to

$$\min_{U,V,r} \tilde{f}(U,V) = \|X - UV^T\|_F^2 + \frac{1}{2}\sum_{i=1}^{r} \|U_i\|_2^2 + \|V_i\|_2^2 \tag{2.23}$$

A formal derivation for this equivalence for can be found in [5, 7].

We define $W := \begin{bmatrix} U \\ V \end{bmatrix}$ and $h : \mathbb{R}^{(D+N)\times(r+1)} \to \mathbb{R}_+$ such that $h(W) = \tilde{f}(U,V)$

and $:= \begin{bmatrix} U \\ V \end{bmatrix} \in \mathbb{R}^{(D+N)\times(r+1)}$.

Using standard matrix derivations, we obtain an expression of the Hessian of h: $\nabla^2 h(W)[\Delta;\Delta] = \|\Delta_U V^T + U\Delta_V^T\|_F^2 + 2\langle UV^T - X, \Delta_U\Delta_V^T\rangle + \|\Delta_U\|_F^2 + \|\Delta_V\|_F^2$. Let $(\bar{U},\bar{V}) \in W \in \mathbb{R}^{(D+N)\times(r+1)}$ be a local minimum of h. The second-order optimality condition is then: $\nabla^2 h(\bar{W})[\Delta;\Delta] \geq 0$. We know from theorem 1 that it is a global optimum if for some $i \in [r]$, $(\bar{U}_i,\bar{V}_i) = (0,0)$. So the second-order optimality condition reduces to

$$2\langle \bar{U}\bar{V}^T - X, uv^T\rangle + \|u\|_2^2 + \|v\|_2^2 \geq 0$$

which is equivalent to the condition on the polar function corresponding to problem (2.23). □

This result in the Low-Rank Matrix Factorization case, which is a boundary case of our Sparse Dictionary Learning problem (2.5), gives us an intuition of the meaning of the polar problem for the more general case. This general formulation with $\in (0,1)$ has no closed-form solution but we found some empirical evidence that the polar problem for this general SDL problem still has a structured optimization geometry as detailed later on.

We recall that we previously derived an equivalent formulation, (2.19), of the initial polar problem (2.17). A first attempt to solve this constrained extrema problem (2.19) approximately is to use one of the previously introduces initialization and optimization strategies and to pick the largest solution. The question however is how difficult is it to reach the global solution, or how many local maxima does this problem have? If this number does not explode exponentially in the dimension one might hope to find (a good approximation of) the global optimum by finding sufficiently many local maxima by local ascent strategies and picking the largest one.

2.3.1 Empirical estimation of the optimization geometry

To get a first intuition of the difficulty of this problem, we study the problem for different values of $\lambda \in (0, 1)$ and on randomly generated matrices Z of rather small dimensions ($D, N \leq 50$). A first observation is that, as expected from the non-convexity of the problem, there always are multiple critical points that the optimization method converges to (see Figures 2.1 and 2.2). To get a better estimation of the number of local optima we run the approximate solvers on $O(N^2)$ random initializations [3]. The optimization strategy used to obtain the results in Figure 2.4 is an accelerated proximal gradient ascent, where the proximal operator is simply the projection onto the constraint set $\{v \in \mathbb{R}^N : \|v\|_1 + (1/\lambda)\|v\|_2 \leq 1\}$. Alternatively we also used the alternating maximization scheme over u and v which lead to very similar results. We drew the matrix Z in two different ways: (1) from a centered Gaussian distribution with covariance $\Sigma = I$ and (2) such that $Z^T Z$ has a sparse eigenvector that does not correspond to the largest eigenvalue. We also controlled the condition number of the matrix by choosing different values for the reciprocal condition number $rc \in (0, 1]$ (defined by the ratio of the smallest singular value over the largest singular value of the matrix Z). In fact, for the boundary case $\lambda = 0$, we know that the convergence speed of iterative optimization methods, for instance by using the power method, depend on the eigen-gap of $Z^T Z$ and therefore the conditioning of the matrix plays a role. What we observe on Figure 2.4 is that the number of local optima that we find after many random initializations ($\leq 2N^2$) appears to be bounded linearly in the dimension N (we removed the sign ambiguity in the reported results, including the sign in the counting leads to an empirical bound of $2N$). We know that for $\lambda = 0$ we have exactly 1 local minimum (the top eigenvector of $Z^T Z$ which is therefore globally min-

[3] Assuming that there are of the order N local minima, that they are uniformly distributed over the set of feasible points and that they have basins of attraction of similar size, we need $O(N \log(N))$ initializations to visit all of them at least once, according to the coupon-collector's problem.

imal) and for = 1 there are N local minima (every element of the canonical basis of R^N). Letting tend to each of these boundary cases we observe that the number of detected local minima is consistent with these known results which might suggest that the results for the remaining cases are good estimates of the actual number of local minima. However we cannot exclude the possibility of further local minima which have very small basins of attraction and are therefore unlikely to be reached from an arbitrary point. Additionally the choice of the optimization method might bias the empirical estimation of the optimization geometry. The latter has been pointed out by [16] in the case of under-determined linear regression and separable linear classi cation problems. A more thorough empirical and theoretic study of the optimization geometry of the studied polar problem is left for future work.

2.3.2 Restricted necessary optimality conditions

With these rst empirical results in hand we now go on to the theoretical analysis: rst we will give a geometric interpretation of the problem and its solutions, then we will derive an equivalent formulation of the SDL polar problem which on the one hand allows a simpler derivation of second-order optimality conditions and on the other hand allows to draw connections to the well-known problem of sparse PCA.

The objective function of (2.18) is a quadratic function and the constraint is an interpolation of the $`_1$ and $`_2$ norm balls as illustrated (in blue) in Figure 2.5. We know that the solutions of this constrained extrema problem lie on the boundary of the constraint. Furthermore, at rst-order (FO) optimal points, the constraint boundary is tangent to the level surface. However, since we want to estimate the number of local maxima and the problem being non-convex, we are interested in second-order (SO) optimal points. The SO optimality condition depends on the curvature of the surface of level and constraint set and informally it requires the constraint surface to have strictly larger curvature at a rst-order optimal point than the level surface for it to be second-order

Figure 2.4: Estimated number of local maxima of the SDL polar problem (y-axis) for different problem sizes as a function of the dimension N (x-axis). Top: $H = Z^T Z$ where each Z_i N (0; I). Bottom: H is generated such that it has at least one sparse eigenvector using a similar strategy as [21]. For each plot, 10 matrices with a xed condition number were generated per dimension and the problem solved by accelerated proximal gradient ascent. Left to right column: results for matrices with reciprocal condition number equal to 0.1, 0.5 and 0.9 respectively.

optimal. An example in R^2 is given in Figure 2.5. Due to the non-differential $`_1$ term in the constraint and the boundary case $= 1$ we have to take some precautions to formalize this informal characterization which uses the surfaces' curvatures. Before we give the formal expression of the FO and SO optimality conditions we derive an equivalent formulation of the SDL polar problem in Proposition 2.

(a) (b)

Figure 2.5: Level and constraint sets of an SDL polar problem in R^2. FO optimal points are indicated by small circles. (a) The FO optimal points are also second-order optimal; (b) The FO optimal points violate the SO optimality condition.

Proposition 2. The sparse dictionary learning polar problem is equivalent to solving the optimization problem

$$\min_{\substack{v \in R^N \\ \|v\|_2 = 1}} \frac{\|v\|_1 + 1}{\|Zv\|_2} \tag{2.24}$$

where $Z = \pm r\,`(X;\ (\ U; V)) \in R^{D\ \ N}$.

Proof. We start with formulation (2.19) of the polar problem derived earlier. Using a result on constrained extrema that states that the solutions of constrained maximization of a convex function are located at the boundaries of

the constraint set we have:

$$\max_{v \in \mathbb{R}^N} \frac{\|Zv\|_2}{\|v\|_1 + (1-\lambda)\|v\|_2} \quad \text{subject to} \quad \|v\|_1 + (1-\lambda)\|v\|_2 = 1 \tag{2.25}$$

Assuming that $\text{null}(Z) \neq \mathbb{R}^N$, we can add the constraint $v \notin \text{null}(Z)$. The positivity of the objective function and invariance to the scaling of v allow us to write

$$\min_{v \in \mathbb{R}^N} \frac{\|v\|_1 + (1-\lambda)\|v\|_2}{\|Zv\|_2} \quad \text{subject to} \quad \|v\|_2 = 1 \tag{2.26}$$

□

The objective function of (2.26) is a convex combination of the objective functions of the variational definition of the $\|\cdot\|_{1,2}$ and $\|\cdot\|_{2,2}$ operator norms on Z. We point out that [19] use this formulation to compute approximate solutions for the well-known sparse PCA problem [39] by using an algorithm designed to solve non-linear eigenvalue problems (the problem being non-linear in the eigenvector). This connection to the sparse PCA problem provides an additional motivation to study the SDL polar problem more deeply to understand its optimization landscape.

We will now present the derivations of first- and second-order optimality conditions for problem (2.24) to further develop the previously mentioned geometric intuitions for this polar problem. In order to eliminate problems due to the non-differentiable ℓ_1 term, we compute the gradient and Hessian of the objective function for a fixed support $\# \subseteq [N]$ and signature $s \in \{-1, 0, 1\}^N$. Let $Z \in \mathbb{R}^{D \times N}$ and $v \in \mathbb{R}^N$ with support $\#$ and signature s, without loss of generality we can assume $v = \begin{bmatrix} v_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \forall_1 s_1 \\ 0 \end{bmatrix}$ where $s = \begin{bmatrix} s_1 \\ 0 \end{bmatrix}$. We also partition the matrix $Z_\#$ accordingly: $Z = [Z_1 \; Z_2]$. We denote $S_1 = \text{diag}(s_1)$, $H = Z^T Z = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}$ and $H_{11} = S_1 Z_1^T Z_1 S_1$. With these notations in hand, we give the following expression for the first-order and second-order optimality conditions:

Proposition 3. Let $Z \in \mathbb{R}^{D \times N}$ and $v \in \mathbb{R}^N$ with support $\quad$ and signature s. v is a local minimum of (2.24) if it satisfies the following conditions:

$$1) \quad (FO) \qquad H_{11}v_1 = -(\ 1 + (1 \quad)v_1) \tag{2.27}$$

$$2) \quad (FO_c) \qquad H_{21}S_1v_1 \in \left[\quad ; \quad\right]_{N \times j} \tag{2.28}$$

$$3) \quad (SO) \qquad \frac{^2}{2}(1^T\ )^2 + \frac{1}{\quad} \quad {}^T H_{11} \tag{2.29}$$

$$\text{for all} \quad \in T_{v_1}S^{j \times j-1} \text{ such that } \|\ \|_2 = 1$$

where we defined $\quad = v_1^T H_{11} v_1$ and $\quad = (1^T v_1) + (1 \quad)$. Given a manifold M, $T_u M$ denotes the tangent space at $u \in M$ to the manifold.

The proof of this proposition uses notions and results of differential geometry and manifold optimization from [1] and is given in the Appendix.

The FO necessary condition (FO) expresses the previous geometric observation that at a first-order optimal point, the normals to the surface of the constraint and level sets are parallel. And in the boundary case $\quad = 0$ we recover the result that the solution to (2.24) is given by the top eigen-vector of H. However for other choices of $\quad$, a geometric interpretation of condition (SO) in terms of curvature of the surface of level and constraint sets is less straightforward. But these optimality conditions on the (Riemannian) gradient and Hessian could possibly be used to bound the number of local optima of the SDL polar problem, by invoking a similar proof as in [14] for the problem of tensor decomposition. This direction is left for future work.

2.4 Extension to discriminative sparse dictionary learning

In this section we present a potential extension of our SDL formulation to discriminative SDL and explain how the theoretical results and our algorithm

easily extend to this new problem, using the fact that the matrix factorization mapping $_r$ can be chosen different from the regular matrix-matrix product (UV^T).

Learning a sparse dictionary and decomposition directly from the data has several different applications, for instance classi cation. In order to classify the data, a possibility is to consider the sparse codes as a nonlinear transformation of the data and to learn a classi er on these features. Indeed it is admitted that sparse codes are well-suited for classi cation of static data ([24] and [33] also show how they can be used for classi cation of time-series data). Since these features do not depend on some pre-de ned dictionary but on a data-dependent dictionary it seems natural to "design" the dictionary and the sparse codes for the classi cation task, i.e. to make them discriminative. Given a training set with known class labels, this can be achieved by adding a discriminative loss term to the model, as it has been proposed in [26]. The authors call their model Supervised Dictionary Learning to emphasize the explicit use of the class labels during dictionary learning as opposed to an unsupervised dictionary learning approach.

In order to integrate this idea of learning discriminative sparse codes into the SDL matrix factorization framework we assume that the data can be separated by a linear classi er in the latent sparse coding coef cient space. Consider that there are L different classes and that for each data point x_i we know its class label y_i; without loss of generality we can assume $y_i \in [L]$ for all $i \in [N]$. Given the sparse code $V_{<i>} \in R^r$ of x_i and a classi er $f W; b g \in R^{L \times r} \times R^L$, the predicted label $\hat{y}_i$ can be de ned via the softmax function (assuming $\hat{y}_i = \arg\max_{l \in [L]} W_{<l>}^T V_{<i>} + b_l$). Under these assumptions we can naturally use the cross-entropy loss on the softmax classi er in our objective function:

$$\min_{\substack{U\in\mathbb{R}^{D\times r}\\ V\in\mathbb{R}^{N\times r};r\in\mathbb{N}\\ W\in\mathbb{R}^{L\times r}\\ b\in\mathbb{R}^{L}}} \underbrace{\frac{1}{2}\|X - UV^T\|_F^2 + \sum_{i=1}^{N} K(y_i, f(V_{<i>}, f W, b))}_{=\ell((X,Y),\Omega(U,V,W),b)} + \left\| \begin{bmatrix} U_j \\ W_j \end{bmatrix} \right\|_{2,1} \quad (2.30)$$

where $f(\cdot, f W, b) = W_{<l>}^T + b_l$ is the score function and K the cross-entropy loss.

This model fits into the matrix factorization framework with the following elements:

The elemental mapping

$$\varphi : \mathbb{R}^D \times \mathbb{R}^N \times \mathbb{R}^L \to \mathbb{R}^{(D+L)\times N}, \quad (u, v, w) \mapsto \begin{bmatrix} uv^T \\ wv^T \end{bmatrix} \quad (2.31)$$

The r-element factorization mapping

$$\varphi_r : \mathbb{R}^{D\times r} \times \mathbb{R}^{N\times r} \times \mathbb{R}^{L\times r} \to \mathbb{R}^{(D+L)\times N}, \quad (U, V, W) \mapsto \sum_{i=1}^{r} \varphi(U_i, V_i, W_i) \quad (2.32)$$

The rank-1 regularizer

$$\Omega : \mathbb{R}^D \times \mathbb{R}^N \times \mathbb{R}^L \to \mathbb{R}_+, \quad (u, v, w) \mapsto \left\| \begin{bmatrix} u \\ w \end{bmatrix} \right\|_2 (\gamma\|v\|_1 + (1-\gamma)\|v\|_2) \quad (2.33)$$

The loss function $\ell((X, Y), Z, b)$ is differentiable and jointly convex w.r.t.

$(Z; b)$. (Here we consider that the bias b as an auxiliary variable which is not part of the factorization; the theoretical framework of our main reference [17] covers this case, but for simplicity we omitted this detail in theorem 1).

Proposition 4. Solving the polar problem for the sparse dictionary learning problem (2.5) is equivalent to solving the polar problem for the discriminative sparse dictionary learning (2.30).

Proof. If we de ne $\tilde{u} = \begin{bmatrix} u \\ w \end{bmatrix}$ and adjust the problem dimensions, then we obtain the same formulation of the SDL polar problem as in (2.17). □

A motivation for integrating the classi er in the sparse dictionary learning matrix factorization framework is twofold: this additional variable does not make the optimization scheme more dif cult to solve than the regular SDL formulation as shown above and it allows to make statements about the optimal size for the dictionary trained for a discriminative (and also reconstructive) task. Indeed, when reviewing the literature on classi cation via sparse coding the heuristics for the dictionary size vary from under-completeness (to prevent over- tting, for instance [26]) to larger dictionaries (if the dictionary is also required to be reconstructive [31]) and the dictionary size is then also chosen via cross-validation which can come at a high computational cost.

However, empirically validating our model on synthetic data and developing some intuition on the formulation's sensitivity to changing parameters turned out to be challenging because of the dif culty to generate data such that we know the joint ground truth $f U ; V ; W ; b g$, or at least a good proxy of it. Furthermore the impact of the inexact polar solutions (except for the boundary cases $2 f 0; 1g$ is not clear yet and still needs to be understood in the regular SDL case. And we conjecture that, in order to obtain interpretable results, we need to add a positivity constraint on the coding variables to guarantee a linear classi er W on these variables.

3 Subspace Clustering and Sparse Dictionary Learning

In this chapter we review two state-of-the-art methods for subspace clustering, belonging to the broader class of spectral methods. They promote different structures to provide compact representations of the data (sparsity or low-rank) which allow to recover a representation of the low-dimensional structure of the data drawn form multiple subspaces by finding a specific decomposition over a self-expressive dictionary (i.e. the data matrix serves as dictionary). As pointed out in the introduction, both sparse dictionary learning and subspace clustering assume the same data model and we will now position our SDL model relatively to the two previously mentioned subspace clustering methods and demonstrate how subspace clustering can be achieved with our formulation while providing additional information on the optimality of the dictionary size w.r.t. the generating factorization size.

3.1 Subspace clustering by sparse or low-rank representation

Given a large amount of high-dimensional data, stacked as columns of a matrix $X \in \mathbb{R}^{D \times N}$ (for instance D could be the number of pixels in an image and N the number of images), it is often preferable to invoke a low-dimensional representation of the data or to cluster the data such that each cluster is well approximated by a low-dimensional subspace in order to perform different data processing and learning tasks more efficiently. In computer vision for instance these could be denoising, classification or inpainting.

If one assumes that the data is well represented by a single low-dimensional subspace, then a classical approach to dimensionality reduction is Principal Component Analysis (PCA). This method and its variants perform well as long as the data can indeed be approximated by a single subspace. But if the structure becomes more complex, i.e. the data rather belongs to multiple subspaces as illustrated in Figure 3.1 and is potentially corrupted by noise or outliers, more advanced techniques of tting the data to such low-dimensional structure are required. Given such a collection of datapoints coming from multiple subspaces, the subspace clustering problem consists of recovering the following information: number and dimensions of the subspaces, a basis for each subspace and the segmentation of the data, ideally without requiring any prior on the number and dimensions of the subspaces.

Figure 3.1: Data $X = [x_1 ::: x_N]$ lies on a union of linear low-dimensional subspaces $S = (S_1 [::: [S_n)$ R^D.

One way to tackle this problem is to nd a self-expressive representation of the data as illustrated in Figure 3.2 and to use this representation to infer the clustering of the points w.r.t. the different subspaces: given $X 2 R^{D N}$, one can assume the self-expressiveness property and seek a matrix $C 2 R^{N N}$ such that $X = XC$. To prevent the trivial solution $C = I$ one needs to enforce $diag(C) = 0$. which does not contribute to the aim of recovering the low-

dimensional structure of the data, one enforces specific constraints onto this matrix C which translate different assumptions on the form of the data, for instance sparsity and low-rank. We will consider these two different situations as proposed in [11] and [12, 23] respectively.

Figure 3.2: Data $X = [x_1 : : : x_N]$ serves as a self-expressive dictionary. The coefficient matrix C is constrained to be zero on its main diagonal.

3.1.1 Sparse subspace clustering

The idea behind the sparse subspace clustering formulation (SSC) is the following: a datapoint x can be approximated by a linear combination of the other datapoints from the set. Such a representation is not unique – there are infinitely many solutions –, but if one requires it to be sparse and if the subspaces are well arranged then it is likely that only points lying in same subspace as x will be selected to represent the given point or at least most of the selected points will belong to the same subspace as x. Such a representation is referred to as subspace-sparse Therefore one gets to solve the following problem:

$$\min \quad \|C\|_0 \qquad \text{subject to } X = XC \text{ and } \operatorname{diag}(C) = 0 \tag{3.1}$$

where the second constraint prevents the trivial solution $C = I$, i.e. each point being represented by itself. This problem is shown to be NP-hard due to the pseudo-norm ℓ_0 but the classical tight relaxation using the sparsity inducing ℓ_1

norm allows to solve a convex problem [11]:

$$\min \quad \|C\|_1 \qquad \text{subject to } X = XC \text{ and } \operatorname{diag}(C) = 0 \tag{3.2}$$

The above formulation requires an exact reconstruction of the data ($X = XC$) but if one assumes that the data is contaminated by some random noise E, then it is preferable to minimize the reconstruction error $\|X + E - XC\|$, which gives the extension of SSC to the presence of noise:

$$\min \quad \|C\|_1 + \frac{\lambda}{2}\|E\|_F^2 \qquad \text{subject to } X = XC + E \text{ and } \operatorname{diag}(C) = 0 \tag{3.3}$$

The authors of [11] show that under appropriate assumptions on the distribution of the data and the con guration of the subspaces the solution of the above formulation is subspace-sparse and can therefore be used to build a similarity graph to deduce the segmentation of the data.

3.1.2 Low-rank subspace clustering

Instead of arguing in terms of sparse representation, one can consider that if the subspaces are well arranged such that the rank r of the data is equal to the sum of the dimensions of the subspaces, therefore potentially $r < \min\{D, N\}$ (if D and N are indeed suf ciently large). This leads to the following non-convex problem

$$\min \quad \operatorname{rank}(C) \qquad \text{subject to } X = XC \tag{3.4}$$

which can be relaxed into a convex problem by noticing that similarly to the ℓ_0/ℓ_1 substitution, the nuclear norm $\|\cdot\|_*$ is a convex envelope for the matrix rank function [1].

If in addition one assumes the data to be corrupted by random noise, i.e. X is the sum of the clean data A and the noise E, the following convex problem

[1] Given a square or rectangular matrix M the nuclear norm is de ned as the sum of the singular values of M, i.e. $\|M\|_* = \sum_{i=1}^{r} \sigma_i$, where the singular values are obtained from the singular value decomposition of M: $M = U\operatorname{diag}(\sigma)V^T$.

can be solved to get a subspace-sparse representation of the data under the appropriate assumptions.

$$\min \quad \|C\|_* + \frac{\lambda}{2}\|E\|_F^2 \qquad \text{subject to } A = AC \text{ and } X = A + E \qquad (3.5)$$

Under appropriate assumptions the authors of [12] show that for clean data and in the presence of noise the problem can be solved in closed form.

These two methods both yield state-of-the-art results for subspace clustering for clean as well as corrupted data and only require fairly simple tools from optimization and linear algebra. But their major drawback lies in their complexity in the number of signals. And the self-expressiveness assumption, while leading to efficient clustering methods, does not provide any information on the optimality of the decomposition w.r.t. the set of sparse decompositions $\{(D, C, r) \in \mathbb{R}^{D \times r} \times \mathbb{R}^{r \times N} : X = DC\}$ since the self-expressiveness assumption implies the use of the fairly large dictionary $D = X$.

Given our analysis from the previous chapters we can therefore ask whether our SDL matrix factorization framework can provide such information for the problem of subspace clustering. In other words we are interested in the "compactness" of the factorization in the problem setting of subspace modeling and clustering.

3.2 Subspace clustering by sparse dictionary learning

The underlying assumptions for sparse dictionary learning are to some extend related to those of subspace clustering: by attempting to sparsely decompose the data over some (overcomplete) set of basis elements one implicitly assumes that the data lies in a union of linear low-dimensional subspaces which allows to express each signal as a linear combination of some basis elements of

the subspace it belongs to, in other words one attempts to recover a subspace-sparse representation. Hence, one choice for the dictionary is to form it as the concatenation of such basis elements leading to subspace-preserving codes (decomposition coef cients) which are used to derive a similarity graph and the segmentation of the data. The data model and its approximated decomposition are illustrated in Figures 3.1 and 3.3.

Figure 3.3: Data X expressed as (sparse) linear combinations of subspace speci c basis elements (B_j , j 2 [n]).

The two previously presented models, SSC and LR-SC, can be considered as special cases of this general formulation, where the signals are considered as a concatenation of (overcomplete) subspace bases. The question now is whether one can recover a more compact and still subspace-preserving decomposition of the data by jointly learning the dictionary and decomposition coef cients without xing the factorization size in advance. In other words, is there a matrix factorization regularization function and associated structured matrix factorization problem which allows to capture the speci c low-dimensional structure

of the data and to find an optimal generating factorization size? If one chooses to drop the self-expressiveness in (3.3) and (3.5) in order to find an optimal subspace-sparse representation of the data, one can obtain the following formulations [2]:

$$\min_{\substack{U,V \\ \|U_i\|_2 \leq 1, \forall i}} \|V\|_1 + \frac{\lambda}{2}\|X - UV^T\|_F^2 \tag{3.6}$$

$$\min_{\substack{U,V \\ \|U_i\|_2 \leq 1, \forall i}} \|V\| + \frac{\lambda}{2}\|X - UV^T\|_F^2 \tag{3.7}$$

For problem (3.7), since we want to find a compact factorization of the data into U and V, instead of assuming only V to be low-rank, we can assume UV^T to have low-rank, leading to

$$\min_{U,V,r} \|UV^T\| + \frac{\lambda}{2}\|X - UV^T\|_F^2 \tag{3.8}$$

We can now establish the link to our SDL formulation by using the well-known variational form of the nuclear norm: given a matrix $M \in \mathbb{R}^{D \times N}$ two variational forms of its nuclear norm are defined as:

$$\|M\|_* = \inf_r \inf_{\substack{U \in \mathbb{R}^{D \times r} \\ V \in \mathbb{R}^{N \times r} \\ X = UV^T}} \sum_{i=1}^{r} \|U_i\|_2 \|V_i\|_2 \tag{3.9}$$

$$= \inf_r \inf_{\substack{U \in \mathbb{R}^{D \times r} \\ V \in \mathbb{R}^{N \times r} \\ X = UV^T}} \frac{1}{2}\sum_{i=1}^{r} \|U_i\|_2^2 + \|V_i\|_2^2 \tag{3.10}$$

This alternative definition of the nuclear norm allows us to recover one boundary case of our SDL model:

$$\min_{\substack{U \in \mathbb{R}^{D \times r} \\ V \in \mathbb{R}^{N \times r}; r \in \mathbb{N}}} \frac{1}{2}\|X - UV^T\|_F^2 + \sum_{i=1}^{r} \|U_i\|_2 \|V_i\|_2 \tag{3.11}$$

[2] As before, we add the norm constraint on U to prevent unbounded solutions.

which is equivalent to the rank minimization problem

$$\min_{Y \in \mathbb{R}^{D \times N}} \quad \frac{1}{2}\|X - Y\|_F^2 + \lambda \|Y\|_* ; \tag{3.12}$$

as showed in [30]. This convex low-rank minimization problem is known to have a closed form solution. But it does not provide any information on the data segmentation.

Similarly for problem (3.6), we use again the remark that constraining the norm of the columns of U is equivalent to penalizing their sum:

$$\min_{U,V,r} \quad \frac{1}{2}\|X - UV^T\|_F^2 + \lambda \sum_{i=1}^{r} \|U_i\|_2 \|V_i\|_1 \tag{3.13}$$

which corresponds to the second extreme case of our problem (2.5) ($\lambda = 1$). Here again the varying factorization size leads to a (trivial) solution ($U = X, V = I$) which does not give any insight into the underlying structure of the data.

But since we assume that there exists a sparse decomposition for the data X (by construction), we can interpolate these two problems to attempt to obtain a compact (low-rank) and subspace-sparse factorization. In other words, since our SDL matrix factorization formulation also involves optimizing over the factorization size, by choosing a suitable regularization function, i.e. a suitable choice for the sparsity parameter λ, one can attempt to recover the number of subspaces, their dimensions and a basis for each subspace in form of a dictionary as well as the representation of the data over this dictionary, i.e. the segmentation of the data collection. If we denote the dimension of subspace S_j by d_j and if we have $\sum_{j=1}^{n} d_j < N$, then we expect the resulting factorization of the data to be more compact as opposed to the self-expressive dictionary from SSC, LRR or LR-SC.

3.3 Experiments

With this framing of our SDL formulation w.r.t. two existing subspace clustering methods we now proceed to the veri cation of our claim on the recovery of a compact subspace-preserving factorization of the data, i.e. for some value for can we perform subspace clustering while obtaining an optimal factorization size?
First we will explain our data model, discuss our initialization strategy and stopping criteria which are necessary due to the use of approximate polar solutions and nally we compare the results of our approach to existing subspace clustering methods.

3.3.1 Preliminaries

SYNTHETIC DATA

We analyze our method by applying it on synthetic data: We generate signals according to two different models: (1) the collection of subspaces is independent[3] and (2) the subspaces are disjoint [4]. Requiring the subspaces to be independent is a strong assumption but intuitively it makes the subspace estimation and clustering easier. We choose these two data models since the methods we are comparing our model to (SSC and LR-SC) give guarantees of subspace recovery for these cases.

For each experiment we generate 100 signals per subspace in the ambient space R^{30}. Each subspace is of dimension R^3. For model (1) we generate signals from 9 independent subspaces, for model (2) we draw 12 subspace bases $f U^j g_{j=1}^{12}$ at random from a normal distribution [5].

[3] A collection of subspaces $f S_j g_{j=1}^n$ is said to be independent if $\dim (\bigoplus_{j=1}^n S_j) = \sum_{j=1}^n \dim (S_j)$ where $\bigoplus$ denotes the direct sum operator. [11]

[4] A collection of subspaces $f S_j g_{j=1}^n$ is said to be disjoint if every pair of subspaces intersect only at the origin. [11]

[5] The generated subspaces are disjoint with probability 1.

REMARK ON THE CHOICE OF INITIALIZATION AND STOPPING CRITERION

Assuming that every step of our algorithm 1 is solved exactly, the algorithm is guaranteed to converge to a global solution, starting from any initial factorization. But as pointed out in Part 2 we cannot solve the polar problem exactly and we therefore have to take this approximation error into account. A phenomenon we observe throughout our experiments (on synthetic and real data) is that the final factorization produced by the algorithm always contains a lot of "small" factors with a vanishing rank-1 regularizer and some factors with much larger (see Figure 3.4 for an example of the evolution of the ratio between the smallest and largest rank-1 regularizer throughout the meta-iterations of our algorithm using approximate polar solutions). This could be due to numerical issues in any of the subroutines of the overall algorithm [6] but it could also be the case that the global solution for our problem (2.5) indeed contains many small rank-1 factors. An argument for the former is that when looking at the low-rank matrix factorization case (= 0) for some matrix X which is exactly low rank, the algorithm recovers the optimal solution (which we also know in closed form), starting from an empty factorization, i.e. it optimizes over the successive polar directions. But if we corrupt each solution of the polar problem with some small white Gaussian noise, we obtain a similar behavior as in Figure 3.4, i.e. a very large factorization containing many small additional factors as reported in Figure 3.5. In parallel we can compare the evolution of the polar and objective functions, see Figure 3.6: in all cases the polar and the objective functions have two stages of evolution: a fast decrease until the factorization size reaches r_{data} and a very slow decrease throughout the rest of the optimization.

These observations suggest that instead of defining the convergence criterion for our algorithm on the polar value (converged ((Z) = 1)) which is likely to lead to an overestimation of the factorization size, we rather need to

[6] In Section 2.2 we empirically convinced ourselves that the non-convex polar problem has a nice optimization landscape and that spectral initialization leads to good estimations of the true solution.

identify the “knee” in either the polar function or the objective which occur at the generating factorization size r_{data}.

Figure 3.4: Ratio of smallest and largest rank-1 regularizer throughout the meta-iterations of algorithm 1, initialized with the empty factorization. The x-axis represents the factorization size r. We stop the algorithm once the polar value drops below 1.1. Problem dimensions: $X = UV^T \in \mathbb{R}^{30 \times 900}$ and $\text{rank}(X) = 27$, $= 0.17$. The red dotted line corresponds to $y = \log(0.001)$. The ratios and reference value y are reported in log-scale.

FORWARD SDL MF

Our algorithm 1 is designed such that it can be initialized with any factorization; for instance we could use initialization strategies used in other approaches to solve the problem: initialize the dictionary U with a random subset of the data matrix X and random sparse coefficients V. However we choose to initialize the algorithm with a zero factorization and the algorithm will therefore run for at least r iterations, where r is the factorization size of the global solution. This specific choice of initialization also appears in a similar algorithmic setting in [4]. We will refer to this choice zero initialization for our algorithm as forward SDL MF.

(a) Ratio $\log(\frac{\min}{\max})$ (for exact polar solutions)

(b) Ratio $\log(\frac{\min}{\max})$ (for noisy polar solutions)

Figure 3.5: Ratio of smallest and largest rank-1 regularizer throughout the meta-iterations of algorithm 1, initialized with the empty factorization. The x-axis represents the factorization size r, which coincides with the meta-iterations in this case. We stop the algorithm once the polar value drops below 1.1. Problem dimensions: $X = UV^T \in \mathbb{R}^{30 \times 900}$ and $\text{rank}(X) = 27$. The red dotted line corresponds to $y = \log(0.001)$.

3.3.2 Recovering the generating factorization size

The phase transition that we describe above for the polar and objective function in the case of noise-free low-rank data still appears for noisy full-rank data drawn from either the independent or disjoint model as illustrated in Figure 3.7.

In order to get a better understanding of this sensitivity of the matrix factorization formulation to the generating factorization size, denoted by r_{data}, we consider the case $\lambda = 1$ (i.e. explicit column regularization on U and sparsity on V) for two reasons: first, as mentioned in the previous section, we can solve the associated polar problem in closed form; this allows us to verify that the described phenomenon is not due to approximation errors of the polar problem discussed earlier, and second, we know a (trivial) global solution to the SDL problem, $X = XI_N$, but there might exist a more compact global solution which

(a) Polar function, $= 0:167$.

(b) Objective (log-scale), $= 0:167$.

(c) Polar function, $= 0$ (noisy polar solutions).

(d) Objective (log-scale), $= 0$ (noisy polar solutions).

Figure 3.6: Evolution of polar and objective functions (when using approximate/noisy polar solutions). The x-axis represents the factorization size r, which coincides with the meta-iterations in this case. Problem dimensions: $X = UV^T \ 2 \ R^{30 \ 900}$ and $\text{rank}(X) = 27$. The red dotted line corresponds to $r_{data} = 27$.

provides more information on the actual structure of the data and which can potentially be discovered by the forward SDL MF algorithm. Intuitively such a compact solution should resemble the generating factorization, for instance in our synthetic data setting, these generating factors are U_{data} and V_{data} which we used to generate the data X. More formally, at each iteration we evaluate

(a) Polar (b) Objective (log-scale)

Figure 3.7: Polar and objective function throughout the meta-iterations of algorithm 1, initialized with the empty factorization. The x-axis represents the factorization size r. Problem dimensions: $X = UV^T + E \in \mathbb{R}^{30\ 960}$, $(U;V) \in \mathbb{R}^{30\ 36} \times \mathbb{R}^{960\ 36}$, $E_{ij} \sim \mathcal{N}(0;0.1^2)$. $= 0.17$. Data drawn from 12 disjoint low-dimensional subspaces of dimension 3.

$_1 \overline{\ }^1 r_{k}`(X; {}_r(U^{(k)};V^{(k)})) = {}_1 \overline{\ }^1(X - U^{(k)}V^{(k)T})$ where we recall that ${}_1(Z) = \max_i \|Z_i\|_2$. Therefore we are in fact evaluating the amplitude of the largest direction in the residual, $X - U^{(k)}V^{(k)T}$. Assume that $k < r_{data}$ and that $U^{(k)}$ is the concatenation of k basis elements of the n subspaces. Without loss of generality, assume that $U^{(k)}$ contains the bases of exactly $m < n$ subspaces $S_1;\dots;S_m$. Given $U^{(k)}$, trying to sparsely decompose points X_i which belong to subspaces $S_{m+1};\dots;S_n$ will then lead to large residuals since in non degenerate cases these points cannot be expressed as sparse linear combinations of elements from subspaces $S_1;\dots;S_m$.

Let us illustrate this behavior on a small example: we take U_{data} as the concatenation of 12 bases in ambient space $\mathbb{R}^D$ (generated at random) and V_{data} as a block-diagonal matrix such that each signal comes from one of the 12 subspaces (each of dimension $d = 3$), i.e. $r_{data} = 36$. We report the results in Figure 3.8 where we observe that, up to a permutation of the columns, the factor $V^{(36)}$ (i.e. after 36 iterations) is block-diagonal and each signal is encoded by 3 atoms. This can be seen more explicitly if we compute the matrix $A = |V||V|^T$ as an

affinity matrix and a variant of the average subspace-sparse recovery error[7]: consider X_i and denote $J_i = \{k \in [N] : X_i \text{ and } X_k \text{ are from the same subspace}\}$.

$$\text{ssr error} = \frac{1}{N}\sum_{i=1}^{N} 1 - \frac{\|(A_i)_{J_i^c}\|_1}{\|A_i\|_1}$$

This error becomes zero after r_{data} iterations of forward SDL MF, which translates as: the dictionary $U^{(k)}$ and sparse codes $V^{(k)}$, $k \geq r_{data}$, are such that points from different subspaces do not share any atoms.

In parallel we note a rapid decrease of the polar value for the first 36 iterations and a stagnation throughout the remaining iterations. A similar behavior also can be observed on the objective function which decreases much slower after the first 36 iterations. We expect this similarity since from [17] we know that the distance of the objective function at a given point (U, V) to the global optimum can be bounded by some linear function of the polar gap at this point (i.e. $\frac{1}{r}\ell_1(X, (U, V)) \geq 1$). Similar results can be observed in the case of unbalanced subspace dimensions (i.e. there exist $(i, j) \in [n]^2$ such that $d_i \neq d_j$, not reported here).

Even though we know that after r_{data} iterations we have not reached the global optimum of our SDL formulation yet (the polar value is still larger than 1 in the given examples) there is empirical evidence that this forward selection strategy for factorizing some matrix X (under the union of subspaces model) allows to recover the generating factorization size in the case of independent and disjoint subspaces.

In order to detect the "knee" on the polar graph (for instance on Figures 3.7 and 3.8a), providing a trade-off between the complexity (factorization size) and the distance to the optimum (polar value), we minimize a simple objec-

[7] The subspace-sparse recovery error was introduced in the context of sparse subspace clustering where the data is used as a self-expressive dictionary and ideally every point is to be encoded exclusively by points from the subspace it belongs to. The ssr error measures the fraction of ℓ_1-norm of the sparse representation which comes from points of other subspaces.