Simple, efficient, and neural algorithms for sparse coding

Simple, Efficient, and Neural Algorithms for Sparse Coding

Sanjeev Arora[*] ARORA@CS.PRINCETON.EDU
Princeton University, Computer Science Department

Rong Ge RONGGE@MICROSOFT.COM
Microsoft Research

Tengyu Ma [†] TENGYU@CS.PRINCETON.EDU
Princeton University, Computer Science Department

Ankur Moitra[‡] MOITRA@MIT.EDU
MIT, Department of Mathematics and CSAIL

Abstract

Sparse coding is a basic task in many fields including signal processing, neuroscience and machine learning where the goal is to learn a basis that enables a sparse representation of a given set of data, if one exists. Its standard formulation is as a non-convex optimization problem which is solved in practice by heuristics based on alternating minimization. Recent work has resulted in several algorithms for sparse coding with provable guarantees, but somewhat surprisingly these are outperformed by the simple alternating minimization heuristics. Here we give a general framework for understanding alternating minimization which we leverage to analyze existing heuristics and to design new ones also with provable guarantees. Some of these algorithms seem implementable on simple neural architectures, which was the original motivation of Olshausen and Field (1997a) in introducing sparse coding. We also give the first efficient algorithm for sparse coding that works almost up to the information theoretic limit for sparse recovery on incoherent dictionaries. All previous algorithms that approached or surpassed this limit run in time exponential in some natural parameter. Finally, our algorithms improve upon the sample complexity of existing approaches. We believe that our analysis framework will have applications in other settings where simple iterative algorithms are used.

1. Introduction

Sparse coding or *dictionary learning* consists of learning to express (i.e., *code*) a set of input vectors, say image patches, as linear combinations of a *small* number of vectors chosen from a large *dictionary*. It is a basic task in many fields. In signal processing, a wide variety of signals turn out to be sparse in an appropriately chosen basis (see references in Mallat (1998)). In neuroscience, sparse representations are believed to improve energy efficiency of the brain by allowing most neurons to be inactive at any given time. In machine learning, imposing sparsity as a constraint on the representation is a useful way to avoid *over-fitting*. Additionally, methods for sparse coding can be thought of as a tool for *feature*

[‡] Supported by the NSF and the Simons Foundation.
[‡] Supported by the NSF and the Simons Foundation.
[‡] Supported by NEC Corporation and Google.

extraction and are the basis for a number of important tasks in image processing such as segmentation, retrieval, de-noising and super-resolution (see references in Elad (2010)), as well as a building block for some deep learning architectures Ranzato et al. (2007). It is also a basic problem in linear algebra itself since it involves finding a better basis.

The notion was introduced by neuroscientists Olshausen and Field (1997a) who formalized it as follows: Given a dataset $y^{(1)}, y^{(2)}, \ldots, y^{(p)} \in \Re^n$, our goal is to find a set of basis vectors $A_1, A_2, \ldots, A_m \in \Re^n$ and sparse coefficient vectors $x^{(1)}, x^{(2)}, \ldots, x^{(p)} \in \Re^m$ that minimize the *reconstruction error*

$$\sum_{i=1}^{p} \|y^{(i)} - A \cdot x^{(i)}\|_2^2 + \sum_{i=1}^{p} S(x^{(i)}) \tag{1}$$

where A is the $n \times m$ *coding matrix* whose jth column is A_j and $S(\cdot)$ is a nonlinear penalty function that is used to encourage sparsity. This function is nonconvex because both A and the $x^{(i)}$'s are unknown. Their paper as well as subsequent work chooses m to be larger than n (so-called *overcomplete* case) because this allows greater flexibility in adapting the representation to the data. We remark that sparse coding should not be confused with the related — and usually easier — problem of finding the sparse representations of the $y^{(i)}$'s *given* the coding matrix A, variously called *compressed sensing* or *sparse recovery* Candes et al. (2006); Candes and Tao (2005).

Olshausen and Field also gave a local search/gradient descent heuristic for trying to minimize the nonconvex energy function (1). They gave experimental evidence that it produces coding matrices for image patches that resemble known features (such as Gabor filters) in $V1$ portion of the visual cortex. A related paper of the same authors Olshausen and Field (1997b) (and also Lewicki and Sejnowski (2000)) places sparse coding in a more familiar *generative model* setting whereby the data points $y^{(i)}$'s are assumed to be probabilistically generated according to a model $y^{(i)} = A^* \cdot x^{*(i)} + \text{noise}$ where $x^{*(1)}, x^{*(2)}, \ldots, x^{*(p)}$ are samples from some appropriate distribution and A^* is an unknown code. Then one can define the maximum likelihood estimate, and this leads to a different and usually more complicated energy function — and associated heuristics — compared to (1).

Surprisingly, maximum likelihood-based approaches seem unnecessary in practice and local search/gradient descent on the energy function (1) with hard constraints works well, as do related algorithms such as MOD Aharon et al. (2006) and k-SVD Engan et al. (1999). In fact these methods are so effective that sparse coding is considered in practice to be a solved problem, even though it has no polynomial time algorithm per se.

Efficient Algorithms vs Neural Algorithms. Recently, there has been rapid progress on designing polynomial time algorithms for sparse coding with provable guarantees (the relevant papers are discussed below). All of these adopt the generative model viewpoint sketched above. But the surprising success of the simple descent heuristics has remained largely unexplained. Empirically, these heuristics far out perform — in running time, sample complexity, and solution quality — the new algorithms, and this (startling) observation was in fact the starting point for the current work.

Of course, the famous example of *simplex* vs *ellipsoid* for linear programming reminds us that it can be much more challenging to analyze the behavior of an empirically successful algorithm than it is to to design a new polynomial time algorithm from scratch! But

for sparse coding the simple intuitive heuristics are important for another reason beyond just their algorithmic efficiency: they appear to be implementable in neural architectures. (Roughly speaking, this means that the algorithm stores the code matrix A as synapse weights in a neural network and updates the entries using differences in potentials of the synapse's endpoints.) Since neural computation — and also deep learning — have proven to be difficult to analyze in general, analyzing sparse coding thoroughly seems to be a natural first step for theory. Our algorithm is a close relative of the Olshausen-Field algorithm and thus inherits its neural implementability; see Appendix H for further discussion.

Here we present a rigorous analysis of the simple energy minimization heuristic, and as a side benefit this yields bounds on running time and sample complexity for sparse coding that are better (in some cases, dramatically so) than the algorithms in recent papers. This adds to the recent literature on analyzing alternating minimization Jain et al. (2013); Hardt (2013); Netrapalli et al. (2013, 2014) but these work in a setting where there is a convex program that is known to work too, and in our setting, the only known convex program runs in time exponential in a natural parameter Barak et al. (2014).

1.1. Recent Work

A common thread in recent work on sparse coding is to assume a generative model; the precise details vary, but each has the property that given enough samples the solution is essentially unique. Spielman et al. (2012) gave an algorithm that succeeds when A^* has full column rank (in particular $m \leq n$) which works up to sparsity roughly $\sqrt{n}$. However this algorithm is not applicable in the more prevalent overcomplete setting. Arora et al. (2014) and Agarwal et al. (2013, 2014) independently gave algorithms in the overcomplete case assuming that A^* is μ-incoherent (which we define in the next section). The former gave an algorithm that works up to sparsity $n^{1/2-\gamma}/\mu$ for any $\gamma > 0$ but the running time is $n^{\Theta(1/\gamma)}$; Agarwal et al. (2013, 2014) gave an algorithm that works up to sparsity either $n^{1/4}/\mu$ or $n^{1/6}/\mu$ depending on the particular assumptions on the model. These works also analyze alternating minimization but assume that it starts from an estimate A that is column-wise $1/\mathrm{poly}(n)$-close to A^*, in which case the objective function is essentially convex. We remark that the key feature that distinguishes our paper with previous works Agarwal et al. (2014); Schnass (2014b,a); Jenatton et al. (2012); Geng and Wright (2014) on non-convex approaches for sparse coding is that we only require an $O(1/\log n)$-close initialization while before $1/\mathrm{poly}(n)$-closeness was needed for provably non-convex local search algorithms.

Barak et al. (2014) gave a new approach based on the sum-of-squares hierarchy that works for sparsity up to $n^{1-\gamma}$ for any $\gamma > 0$. But in order to output an estimate that is column-wise ϵ-close to A^* the running time of the algorithm is $n^{1/\epsilon^{O(1)}}$. In most applications, one needs to set (say) $\epsilon = 1/k$ in order to get a useful estimate. However in this case their algorithm runs in exponential time. The sample complexity of the above algorithms is also rather large, and is at least $\Omega(m^2)$ if not much larger. Here we will give simple and more efficient algorithms based on alternating minimization whose column-wise error decreases geometrically, and that work for sparsity up to $n^{1/2}/\mu \log n$. We remark that even empirically, alternating minimization does not appear to work much beyond this bound.

1.2. Model, Notation and Results

We will work with the following family of generative models (similar to those in earlier papers)[1]:

Our Model Each sample is generated as $y = A^*x^* + \text{noise}$ where A^* is a ground truth dictionary and x^* is drawn from an unknown distribution $\mathcal{D}$ where

(1) the support $S = \text{supp}(x^*)$ is of size at most k, $\mathbf{Pr}[i \in S] = \Theta(k/m)$ and $\mathbf{Pr}[i, j \in S] = \Theta(k^2/m^2)$

(2) the distribution is normalized so that $\mathbf{E}[x_i^*|x_j^* \neq 0] = 0$; $\mathbf{E}[x_i^{*2}|x_i^* \neq 0] = 1$ and when $x_i^* \neq 0$, $|x_i^*| \geq C$ for some constant $C \leq 1$ and

(3) the non-zero entries are pairwise independent and subgaussian, conditioned on the support.

(4) The noise is Gaussian and independent across coordinates.

Such models are natural since the original motivation behind sparse coding was to discover a code whose representations have the property that the coordinates are almost independent. We can relax most of the requirements above, at the expense of further restricting the sparsity, but will not detail such tradeoffs.

The rest of the paper ignores the iid noise: it has little effect on our basic steps like computing inner products of samples or taking singular vectors, and easily tolerated so long as it stays smaller than the "signal."

We assume A^* is an incoherent dictionary, since these are widespread in signal processing Elad (2010) and statistics Donoho and Huo (1999), and include various families of wavelets, Gabor filters as well as randomly generated dictionaries.

Definition 1 *An $n \times m$ matrix A whose columns are unit vectors is μ-incoherent if for all $i \neq j$ we have $\langle A_i, A_j \rangle \leq \mu/\sqrt{n}$.*

We also require that $\|A^*\| = O(\sqrt{m/n})$. However this can be relaxed within polylogarithmic factors by tightening the bound on the sparsity by the same factor. Throughout this paper we will say that A^s is (δ, κ)-near to A^* if after a permutation and sign flips its columns are within distance δ and we have $\|A^s - A^*\| \leq \kappa\|A^*\|$. See also Definition 7. We will use this notion to measure the progress of our algorithms. Moreover we will use $g(n) = O^*(f(n))$ to signify that $g(n)$ is upper bounded by $Cf(n)$ for some small enough constant C.

Regime of parameters: Finally, throughout this paper we will assume that $k \leq O^*(\sqrt{n}/\mu \log n)$ and $m = O(n)$. Again, m can be allowed to be higher by lowering the sparsity. *We assume all these conditions in our main theorems.*

1. The casual reader should just think of x^* as being drawn from some distribution that has independent coordinates. Even in this simpler setting —which has polynomial time algorithms using *Independent Component Analysis*—we do not know of any rigorous analysis of heuristics like Olshausen-Field. The earlier papers were only interested in polynomial-time algorithms, so did not wish to assume independence.

Main Theorems In Section 2 we give a general framework for analyzing alternating minimization. Instead of thinking of the algorithm as trying to minimize a known non-convex function, we view it as trying to minimize an *unknown* convex function. Various update rules are shown to provide good approximations to the gradient of the unknown function. See Lemma 10, Lemma 26 and Lemma 31 for examples. We then leverage our framework to analyze existing heuristics and to design new ones also with provable guarantees. In Section 3, we prove:

Theorem 2 *There is a* neurally plausible *algorithm which when initialized with an estimate A^0 that is $(\delta, 2)$-near to A^* for $\delta = O^*(1/\log n)$, converges at a geometric rate to A^* until the column-wise error is $O(\sqrt{k/n})$. Furthermore the running time is $O(mnp)$ and the sample complexity is $p = \widetilde{O}(mk)$ for each step.*

Additionally we give a neural architecture implementing our algorithm in Appendix H. To the best of our knowledge, this is the first neurally plausible algorithm for sparse coding with provable convergence. We also remark that when the coefficients x^* have independent entries then the theorem above can be strengthened to work for nearly linear sparsity $k = O^*(n/\text{polylog}n)$, although we don't have a good initialization procedure to achieve $O^*(1/\log n)$-closeness in this regime.

Having set up our general framework and analysis technique we can use it on other variants of alternating minimization. Section 4.2 gives a new update rule whose bias (i.e., error) is negligible:

Theorem 3 *There is an algorithm which when initialized with an estimate A^0 that is $(\delta, 2)$-near to A^* for $\delta = O^*(1/\log n)$, converges at a geometric rate to A^* until the column-wise error is $O(n^{-\omega(1)})$. Furthermore each step runs in time $O(mnp)$ and the sample complexity p is polynomial*[2]*.*

This algorithm is based on a modification where we carefully project out components along the column currently being updated. We complement the above theorems by revisiting the Olshausen-Field rule and analyzing a variant of it in Section 4.1 (Theorem 11). However its analysis is more complex because we need to bound some quadratic error terms. It uses convex programming.

What remains is to give a method to initialize these iterative algorithms. We give a new approach based on pair-wise reweighting and we prove that it returns an estimate A^0 that is $(\delta, 2)$-near to A^* for $\delta = O^*(1/\log n)$ with high probability. As an additional benefit, this algorithm can be used even in settings where m is not known and this could help solve another problem in practice — that of *model selection*. In Section 5 we prove:

Theorem 4 *There is an algorithm which returns an estimate A^0 that is $(\delta, 2)$-near to A^* for $\delta = O^*(1/\log n)$. Furthermore the running time is $\widetilde{O}(mn^2p)$ and the sample complexity $p = \widetilde{O}(mk)$.*

2. In principle, the sample complexity of this algorithm should be similar to that of 2. A careful analysis is left to future work

Algorithm 1 Generic Alternating Minimization Approach

Initialize A^0
Repeat for $s = 0, 1, ..., T$

Decode: Find a sparse solution to $A^s x^{(i)} = y^{(i)}$ for $i = 1, 2, ..., p$
Set X^s such that its columns are $x^{(i)}$ for $i = 1, 2, ..., p$
Update: $A^{s+1} = A^s - \eta g^s$ where g^s is the gradient of $\mathcal{E}(A^s, X^s)$ with respect to A^s

This algorithm also admits a neural implementation, which is sketched in Appendix H. The proof currently requires a projection step that increases the run time though we suspect it is not needed.

We remark that these algorithms work up to sparsity $O^*(\sqrt{n}/\mu \log n)$ which is within a logarithmic factor of the information theoretic threshold for sparse recovery on incoherent dictionaries Donoho and Huo (1999); Gribonval and Nielsen (2003). All previous known algorithms that approach Arora et al. (2014) or surpass this sparsity Barak et al. (2014) run in time exponential in some natural parameter. Moreover, our algorithms are simple to describe and implement, and involve only basic operations. We believe that our framework will have applications beyond sparse coding, and could be used to show that simple, iterative algorithms can be powerful in other contexts as well by suggesting new ways to analyze them.

2. Our Framework, and an Overview

Here we describe our framework for analyzing alternating minimization. The generic scheme we will be interested in is given in Algorithm 1 and it alternates between updating the estimates A and X. It is a heuristic for minimizing the non-convex function in (1) where the penalty function is a hard constraint. The crucial step is if we fix X and compute the gradient of (1) with respect to A, we get:

$$\nabla_A \mathcal{E}(A, X) = \sum_{i=1}^{p} -2(y^{(i)} - Ax^{(i)})(x^{(i)})^T.$$

We then take a step in the opposite direction to update A. Here and throughout the paper η is the learning rate, and needs to be set appropriately. The challenge in analyzing this general algorithm is to identify a suitable "measure of progress"— called a Lyapunov function in dynamical systems and control theory — and show that it improves at each step (with high probability over the samples). We will measure the progress of our algorithms by the maximum column-wise difference between A and A^*.

In the next subsection, we identify sufficient conditions that guarantee progress. They are inspired by proofs in convex optimization. We view Algorithm 1 as trying to minimize an *unknown* convex function, specifically $f(A) = \mathcal{E}(A, X^*)$, which is strictly convex and hence has a unique optimum that can be reached via gradient descent. This function is unknown since the algorithm does not know X^*. The analysis will show that the direction of movement is correlated with $A^* - A^s$, which in turn is the gradient of the above function. An independent paper of Balakrishnan et al. (2014) proposes a similar framework for

analysing EM algorithms for hidden variable models. The difference is that their condition is really about the geometry of the objective function, though ours is about the property of the direction of movement. Therefore we have the flexibility to choose different decoding procedures. This flexibility allows us to have a closed form of X^s and obtain a useful functional form of g^s. The setup is reminiscent of *stochastic gradient descent*, which moves in a direction whose expectation is the gradient of a *known* convex function. By contrast, here the function $f()$ is unknown, and furthermore the expectation of g^s is not the true gradient and has *bias*. Due to the bias, we will only be able to prove that our algorithms reach an approximate optimum up to some error whose magnitude is determined by the bias. We can make the bias negligible using more complicated algorithms.

Approximate Gradient Descent

Consider a general iterative algorithm that is trying to get to a desired solution z^* (in our case $z^* = A_i^*$ for some i). At step s it starts with a guess z^s, computes some direction g^s, and updates its estimate as: $z^{s+1} = z^s - \eta g^s$. The natural progress measure is $\|z^* - z^s\|^2$, and below we will identify a sufficient condition for it to decrease in each step:

Definition 5 *A vector g^s is $(\alpha, \beta, \epsilon_s)$-*correlated with z^* *if*

$$\langle g^s, z^s - z^* \rangle \geq \alpha \|z^s - z^*\|^2 + \beta \|g^s\|^2 - \epsilon_s.$$

Remark: The traditional analysis of convex optimization corresponds to the setting where z^* is the global optimum of some convex function f, and $\epsilon_s = 0$. Specifically, if $f(\cdot)$ is 2α-strongly convex and $1/(2\beta)$-smooth, then $g^s = \nabla f(z^s)$ $(\alpha, \beta, 0)$-correlated with z^*. Also we will refer to ϵ_s as the bias.

Theorem 6 *Suppose g^s satisfies Definition 5 for $s = 1, 2, \ldots, T$, and η satisfies $0 < \eta \leq 2\beta$ and $\epsilon = \max_{s=1}^T \epsilon_s$. Then for any $s = 1, \ldots, T$,*

$$\|z^{s+1} - z^*\|^2 \leq (1 - 2\alpha\eta)\|z^s - z^*\|^2 + 2\eta\epsilon_s$$

In particular, the update rule above converges to z^* geometrically with systematic error ϵ/α in the sense that

$$\|z^s - z^*\|^2 \leq (1 - 2\alpha\eta)^s \|z^0 - z^*\|^2 + \epsilon/\alpha.$$

Furthermore, if $\epsilon_s < \frac{\alpha}{2}\|z^s - z^*\|^2$ for $s = 1, \ldots, T$, then

$$\|z^s - z^*\|^2 \leq (1 - \alpha\eta)^s \|z^0 - z^*\|^2.$$

We defer the proof to Appendix C; it closely follows existing proofs in convex optimization, and we also give an analysis for approximate projected gradient descent in Corollary 15.

An Overview of Applying Our Framework

Our framework clarifies that any improvement step meeting Definition 5 will also converge to an approximate optimum, which enables us to engineer other update rules that turn out to be easier to analyze. Indeed we first analyze a simpler update rule with $g^s =$

$\mathbf{E}[(y - A^s x)\mathrm{sgn}(x^T)]$ in Section 3. Here $\mathrm{sgn}(\cdot)$ is the coordinate-wise sign function. We then return to the Olshausen-Field update rule and analyze a variant of it in Section 4.1 using approximate projected gradient descent. Finally, we design a new update rule in Section 4.2 where we carefully project out components along the column currently being updated. This has the effect of replacing one error term with another and results in an update rule with negligible bias. The main steps in showing that these update rules fit into our framework are given in Lemma 10, Lemma 26 and Lemma 31.

How should the algorithm update X? The usual approach is to solve a sparse recovery problem with respect to the current code matrix A. However many of the standard basis pursuit algorithms (such as solving a linear program with an ℓ_1 penalty) are difficult to analyze when there is error in the code itself. This is in part because the solution does not have a closed form in terms of the code matrix. Instead we take a much simpler approach to solving the sparse recovery problem which uses matrix-vector multiplication followed by thresholding: In particular, we set $x = \mathrm{threshold}_{C/2}((A^s)^T y)$, where $\mathrm{threshold}_{C/2}(\cdot)$ keeps only the coordinates whose magnitude is at least $C/2$ and zeros out the rest. Recall that the non-zero coordinates in x^* have magnitude at least C. This decoding rule recovers the signs and support of x correctly provided that A is column-wise δ-close to A^* for $\delta = O^*(1/\log n)$. See Lemma 9.

The rest of the analysis can be described as follows: If the signs and support of x are recovered correctly, then alternating minimization makes progress in each step. In fact this holds each for much larger values of k than we consider; as high as $n/(\log n)^{O(1)}$. (However, the explicit decoding rule fails for $k > \sqrt{n}/\mu \log n$.) Thus it only remains to properly initialize A^0 so that it is close enough to A^* to let the above decoding rule succeed. In Section 5 we give a new initialization procedure based on pair-wise reweighting that we prove works with high probability. This section may be of independent interest, since this algorithm can be used even in settings where m is not known and could help solve another problem in practice — that of *model selection*. See Lemma 14.

3. A Neurally Plausible Algorithm with Provable Guarantees

Here we will design and analyze a neurally plausible algorithm for sparse coding which is given in Algorithm 2, and we give a neural architecture implementing our algorithm in Appendix H. The fact that such a simple algorithm provably works sheds new light on how sparse coding might be accomplished in nature. Here and throughout this paper we will work with the following measure of closeness:

Definition 7 *A is δ-*close to A^* *if there is a permutation $\pi : [m] \to [m]$ and a choice of signs $\sigma : [m] \to \{\pm 1\}$ such that $\|\sigma(i)A_{\pi(i)} - A_i^*\| \leq \delta$ for all i We say A is (δ, κ)-*near to A^* *if in addition $\|A - A^*\| \leq \kappa \|A^*\|$ too.*

This is a natural measure to use, since we can only hope to learn the columns of A^* up to relabeling and sign-flips. In our analysis, we will assume throughout that $\pi(\cdot)$ is the identity permutation and $\sigma(\cdot) \equiv +1$ because our family of generative models is invariant under this relabeling and it will simplify our notation.

Algorithm 2 Neurally Plausible Update Rule

Initialize A^0 that is $(\delta_0, 2)$-near to A^*
Repeat for $s = 0, 1, ..., T$

Decode: $x^{(i)} = \text{threshold}_{C/2}((A^s)^T y^{(i)})$ for $i = 1, 2, ..., p$

Update: $A^{s+1} = A^s - \eta \widehat{g}^s$ where $\widehat{g}^s = \frac{1}{p} \cdot \sum_{i=1}^{p} (y^{(i)} - A^s x^{(i)}) \text{sgn}(x^{(i)})^T$

Let $\text{sgn}(\cdot)$ denote the coordinate-wise sign function and recall that η is the learning rate, which we will soon set. Also we fix both $\delta, \delta_0 = O^*(1/\log n)$. We will also assume that in each iteration, our algorithm is given a fresh set of p samples. Our main theorem is:

Theorem 8 *Suppose that A^0 is $(2\delta, 2)$-near to A^* and that $\eta = \Theta(m/k)$. Then if each update step in Algorithm 2 uses $p = \widetilde{\Omega}(mk)$ fresh samples, we have*

$$\mathbf{E}[\|A_i^s - A_i^*\|^2] \leq (1-\tau)^s \|A_i^0 - A_i^*\|^2 + O(k/n)$$

for some $0 < \tau < 1/2$ and for any $s = 1, 2, ..., T$. In particular it converges to A^* geometrically, until the column-wise error is $O(\sqrt{k/n})$.

Our strategy is to prove that $\widehat{g}^s$ is $(\alpha, \beta, \epsilon)$-correlated (see Definition 5) with the desired solution A^*, and then to prove that $\|A\|$ never gets too large. We will first prove that if A is somewhat close to A^* then the estimate x for the representation almost always has the correct support. Here and elsewhere in the paper, we use "very high probability" to mean that an event happens with probability at least $1 - 1/n^{\omega(1)}$.

Lemma 9 *Suppose that A^s is δ-close to A^*. Then with very high probability over the choice of the random sample $y = A^* x^*$:*

$$\text{sgn}(\text{threshold}_{C/2}((A^s)^T y)) = \text{sgn}(x^*)$$

We prove a more general version of this lemma (Lemma 16) in Appendix C; it is an ingredient in analyzing all of the update rules we consider in this paper. However this is just one step on the way towards proving that $\widehat{g}^s$ is correlated with the true solution.

The next step in our proof is to use the properties of the generative model to derive a new formula for $\widehat{g}^s$ that is more amenable to analysis. We define g^s to be the expectation of $\widehat{g}^s$

$$g^s := \mathbf{E}[\widehat{g}^s] = \mathbf{E}[(y - A^s x)\text{sgn}(x)^T] \tag{2}$$

where $x := \text{threshold}_{C/2}((A^s)^T y)$ is the decoding of y. Let $q_i = \mathbf{Pr}[x_i^* \neq 0]$ and $q_{i,j} = \mathbf{Pr}[x_i^* x_j^* \neq 0]$, and define $p_i = \mathbf{E}[x_i^* \text{sgn}(x_i^*) | x_i^* \neq 0]$.

Here and in the rest of the paper, we will let γ denote any vector whose norm is negligible (i.e. smaller than $1/n^C$ for any large constant $C > 1$). This will simplify our calculations. Also let A^*_{-i} denote the matrix obtained from deleting the ith column of A^*. The following lemma is the main step in our analysis.

Lemma 10 *Suppose that A^s is $(2\delta, 2)$-near to A^*. Then the update step in Algorithm 2 takes the form $\mathbf{E}[A_i^{s+1}] = A_i^s - \eta g_i^s$ where $g_i^s = p_i q_i \left(\lambda_i^s A_i^s - A_i^* + \epsilon_i^s \pm \gamma\right)$, and $\lambda_i^s = \langle A_i^s, A_i^* \rangle$ and*

$$\epsilon_i^s = \left(A_{-i}^s diag(q_{i,j}) \left(A_{-i}^s\right)^T\right) A_i^* / q_i$$

Moreover the norm of ϵ_i^s can be bounded as $\|\epsilon_i^s\| \leq O(k/n)$.

Note that $p_i q_i$ is a scaling constant and $\lambda_i \approx 1$; hence from the above formula we should expect that g_i^s is well-correlated with $A_i^s - A_i^*$. We defer the proof to Section A.

In Appendix D we complete the analysis of Algorithm 2 in the infinite sample setting. In particular, in Appendix D.1, we prove that if A^s is $(2\delta, 2)$-near to A^* then g_i^s is indeed $(\alpha, \beta, \epsilon)$-correlated with A_i (Lemma 23). Finally we prove that if A^s is $(2\delta, 2)$-near to A^* then $\|A^{s+1} - A^*\| \leq 2\|A^*\|$ (Lemma 24). These lemmas together with Theorem 6 imply Theorem 21, the simplified version of Theorem 8 where the number of samples p is assumed to be infinite (i.e. we have access to the true expectation g^s). In Appendix G we prove the sample complexity bounds we need and this completes the proof of Theorem 8.

4. Further Applications

Here we apply our framework to design and analyze further variants of alternating minimization.

4.1. Revisiting Olshausen-Field

In this subsection we analyze a variant of the Olshausen-Field update rule. However there are quadratic error terms that arise in the expressions we derive for g^s and bounding them is more challenging. We will also need to make (slightly) stronger assumptions on the distributional model that for distinct i_1, i_2, i_3 we have $q_{i_1,i_2,i_3} = O(k^3/m^3)$ where $q_{i_1,i_2,i_3} = \mathbf{Pr}[i_1, i_2, i_3 \in S]$.

Theorem 11 *Suppose that A^0 is $(2\delta, 2)$-near to A^* and that $\eta = \Theta(m/k)$. There is a variant of Olshausen-Field (given in Algorithm 4 in Appendix E.1) for which at each step s we have*

$$\|A^s - A^*\|_F^2 \leq (1-\tau)^s \|A^0 - A^*\|_F^2 + O(mk^2/n^2)$$

for some $0 < \tau < 1/2$ and for any $s = 1, 2, ..., T$. In particular it converges to A^* geometrically until the error in Frobenius norm is $O(\sqrt{m}k/n)$.

We defer the proof of the main theorem to Appendix E.1. Currently it uses a projection step (using convex programming) that may not be needed but the proof requires it.

4.2. Removing the Systemic Error

In this subsection, we design and analyze a new update rule that converges geometrically until the column-wise error is $n^{-\omega(1)}$. The basic idea is to engineer a new decoding matrix that projects out the components along the column currently being updated. This has the effect of replacing a certain error term in Lemma 10 with another term that goes to zero as A gets closer to A^* (the earlier rules we have analyzed do not have this property).

We will use $B^{(s,i)}$ to denote the decoding matrix used when updating the ith column in the sth step. Then we set $B_i^{(s,i)} = A_i$ and $B_j^{(s,i)} = \text{Proj}_{A_i^\perp} A_j$ for $j \neq i$. Note that $B_{-i}^{(s,i)}$

(i.e. $B^{(s,i)}$ with the ith column removed) is now orthogonal to A_i. We will rely on this fact when we bound the error. We defer the proof of the main theorem to Appendix E.2.

Theorem 12 *Suppose that A^0 is $(2\delta, 2)$-near to A^* and that $\eta = \Theta(m/k)$. There is an algorithm (given in Algorithm 5 given in Appendix E.2) for which at each step s, we have*

$$\|A_i^s - A_i^*\|^2 \leq (1-\tau)^s \|A_i^0 - A_i^*\|^2 + n^{-\omega(1)}$$

for some $0 < \tau < 1/2$ and for any $s = 1, 2, ..., T$. In particular it converges to A^* geometrically until the column-wise error is $n^{-\omega(1)}$.

5. Initialization

There is a large gap between theory and practice in terms of how to initialize alternating minimization. The usual approach is to set A randomly or to populate its columns with samples $y^{(i)}$. These often work but we do not know how to analyse them. Here we give a novel method for initialization which we show succeeds with very high probability. Our algorithm works by pairwise reweighting. Let $u = A^*\alpha$ and $v = A^*\alpha'$ be two samples from our model whose supports are U and V respectively. The main idea is that if we reweight fresh samples y with a factor $\langle y, u\rangle\langle y, v\rangle$ and compute

$$\widehat{M}_{u,v} = \frac{1}{p_2}\sum_{i=1}^{p_2} \langle y^{(i)}, u\rangle\langle y^{(i)}, v\rangle y^{(i)}(y^{(i)})^T$$

then the top singular vectors will correspond to columns A_j^* where $j \in U \cap V$. (This is reminiscent of ideas in recent papers on dictionary learning, but more sample efficient.)

Throughout this section we will assume that the algorithm is given two sets of samples of size p_1 and p_2 respectively. Let $p = p_1 + p_2$. We use the first set of samples for the pairs u, v that are used in reweighting and we use the second set to compute $\widehat{M}_{u,v}$ (that is, the same set of p_2 samples is used for each u, v throughout the execution of the algorithm). Our main theorem is:

Theorem 13 *Suppose that Algorithm 3 is given $p_1 = \widetilde{\Omega}(m)$ and $p_2 = \widetilde{\Omega}(mk)$ fresh samples and moreover (a) A^* is μ-incoherent with $\mu = O^*(\frac{\sqrt{n}}{k\log^3 n})$, (b) $m = O(n)$ and (c) $\|A^*\| \leq O(\sqrt{\frac{m}{n}})$. Then with high probability A is $(\delta, 2)$-near to A^* where $\delta = O^*(1/\log n)$.*

We will defer the proof of this theorem, and the following main lemma to Appendix F.

Lemma 14 *Suppose $u = A^*\alpha$ and $v = A^*\alpha'$ are two random samples with supports U, V respectively. Let $\beta = A^{*T}u$ and $\beta' = A^{*T}v$. Let $y = A^*x^*$ be random sample that is independent of u,v, then*

$$M_{u,v} := \mathbf{E}[\langle u, y\rangle\langle v, y\rangle yy^T] = \sum_{i\in U\cap V} q_i c_i \beta_i \beta_i' A_i^* A_i^{*T} + E_1 + E_2 + E_3, \tag{3}$$

where $q_i = \mathbf{Pr}[i \in S]$, $c_i = \mathbf{E}[x_i^{*4} | i \notin S]$, and the error terms are:

$$\begin{aligned} E_1 &= \textstyle\sum_{i\notin U\cap V} q_i c_i \beta_i \beta_i' A_i^* A_i^{*T} \\ E_2 &= \textstyle\sum_{i,j\in[m], i\neq j} q_{i,j} \beta_i \beta_i' A_j^* A_j^{*T} \\ E_3 &= \textstyle\sum_{i,j\in[m], i\neq j} q_{i,j}(\beta_i A_i^* \beta_j' A_j^{*T} + \beta_i' A_i^* \beta_j A_j^{*T}). \end{aligned}$$

Algorithm 3 Pairwise Initialization

Set $L = \emptyset$
While $|L| < m$ choose samples u and v

Set $\widehat{M}_{u,v} = \frac{1}{p_2}\sum_{i=1}^{p_2}\langle y^{(i)}, u\rangle\langle y^{(i)}, v\rangle y^{(i)}(y^{(i)})^T$

Compute the top two singular values σ_1, σ_2 and top singular vector z of $\widehat{M}_{u,v}$

If $\sigma_1 \geq \Omega(k/m)$ and $\sigma_2 < O^*(k/m \log m)$

If z is not within distance $1/\log m$ of any vector in L (even after sign flip), add z to L

Set $\widetilde{A}$ such that its columns are $z \in L$ and output $A = \text{Proj}_{\mathcal{B}}\widetilde{A}$ where $\mathcal{B}$ is the convex set defined in Definition 28

Moreover the error terms $E_1 + E_2 + E_3$ has spectral norm bounded by $O^*(k/m \log n)$, $|\beta_i| \geq \Omega(1)$ for all $i \in \text{supp}(\alpha)$ and $|\beta'_i| \geq \Omega(1)$ for all $i \in \text{supp}(\alpha')$.

We will invoke this lemma several times in order to analyze Algorithm 3 to verify whether or not the supports of u and v share a common element, and again to show that if they do we can approximately recover the corresponding column of A^* from the top singular vector of $M_{u,v}$.

Conclusions

Going beyond $\sqrt{n}$ sparsity requires new ideas as alternating minimization appears to break down. Mysterious properties of alternating minimization are also left to explore, such as why a random initialization works. Are these heuristics information theoretically optimal in terms of their sample complexity? Finally, can we analyse energy minimization in other contexts as well?

Acknowledgements

We are grateful to Dmitri Chklovskii and Sebastian Seung for useful discussions about neural computation.

References

A. Agarwal, A. Anandkumar, and P. Netrapalli. Exact recovery of sparsely used overcomplete dictionaries. In *arXiv:1309.1952*, 2013.

A. Agarwal, A. Anandkumar, P. Jain, P. Netrapalli, and R. Tandon. Learning sparsely used overcomplete dictionaries via alternating minimization. In *COLT*, pages 123–137, 2014.

M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. In *IEEE Trans. on Signal Processing*, pages 4311–4322, 2006.

S. Arora, R. Ge, and A. Moitra. New algorithms for learning incoherent and overcomplete dictionaries. In *COLT*, pages 779–806, 2014.

Sivaraman Balakrishnan, Martin J. Wainwright, and Bin Yu. Statistical guarantees for the EM algorithm: From population to sample-based analysis. *CoRR*, abs/1408.2156, 2014. URL http://arxiv.org/abs/1408.2156.

Boaz Barak, John Kelner, and David Steurer. Dictionary learning using sum-of-square hierarchy. 2014.

E. Candes and T. Tao. Decoding by linear programming. In *IEEE Trans. on Information Theory*, pages 4203–4215, 2005.

E. Candes, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. In *Communications of Pure and Applied Math*, pages 1207–1223, 2006.

D. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. In *IEEE Trans. on Information Theory*, pages 2845–2862, 1999.

M. Elad. Sparse and redundant representations. In *Springer*, 2010.

K. Engan, S. Aase, and J. Hakon-Husoy. Method of optimal directions for frame design. In *ICASSP*, pages 2443–2446, 1999.

Quan Geng and John Wright. On the local correctness of l1-minimization for dictionary learning. In *2014 IEEE International Symposium on Information Theory, Honolulu, HI, USA, June 29 - July 4, 2014*, pages 3180–3184, 2014. doi: 10.1109/ISIT.2014.6875421. URL http://dx.doi.org/10.1109/ISIT.2014.6875421.

R. Gribonval and M. Nielsen. Sparse representations in unions of bases. In *IEEE Transactions on Information Theory*, pages 3320–3325, 2003.

M. Hardt. On the provable convergence of alternating minimization for matrix completion. In *arxiv:1312.0925*, 2013.

R. Horn and C. Johnson. Matrix analysis. In *Cambridge University Press*, 1990.

P. Jain, P. Netrapalli, and S. Sanghavi. Low rank matrix completion using alternating minimization. In *STOC*, pages 665–674, 2013.

Rodolphe Jenatton, Rémi Gribonval, and Francis R. Bach. Local stability and robustness of sparse dictionary learning in the presence of noise. *CoRR*, abs/1210.0685, 2012. URL http://arxiv.org/abs/1210.0685.

M. Lewicki and T. Sejnowski. Learning overcomplete representations. In *Neural Computation*, pages 337–365, 2000.

S. Mallat. A wavelet tour of signal processing. In *Academic-Press*, 1998.

Praneeth Netrapalli, Prateek Jain, and Sujay Sanghavi. Phase retrieval using alternating minimization. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 2796–2804, 2013. URL http://papers.nips.cc/paper/5041-phase-retrieval-using-alternating-minimization.

Praneeth Netrapalli, Niranjan U. N, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain. Non-convex robust PCA. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 1107–1115, 2014. URL http://papers.nips.cc/paper/5430-non-convex-robust-pca.

Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: a strategy employed by v1. *Vision Research*, 37:3311–3325, 1997a.

Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997b.

Marc'Aurelio Ranzato, Y-Lan Boureau, and Yann LeCun. Sparse feature learning for deep belief networks. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 1185–1192, 2007. URL http://papers.nips.cc/paper/3363-sparse-feature-learning-for-deep-belief-networks.

Karin Schnass. Local Identification of Overcomplete Dictionaries. *ArXiv e-prints*, January 2014a.

Karin Schnass. On the identifiability of overcomplete dictionaries via the minimisation principle underlying k-svd. *Applied and Computational Harmonic Analysis*, 37(3):464 – 491, 2014b. ISSN 1063-5203. doi: http://dx.doi.org/10.1016/j.acha.2014.01.005. URL http://www.sciencedirect.com/science/article/pii/S1063520314000207.

D. Spielman, H. Wang, and J. Wright. Exact recovery of sparsely-used dictionaries. In *Journal of Machine Learning Research*, 2012.

Appendix A. Proof of Lemma 10

Since A^s is $(2\delta, 2)$-near to A^*, A^s is 2δ-close to A^*. We can now invoke Lemma 9 and conclude that with high probability, $\mathrm{sgn}(x^*) = \mathrm{sgn}(x)$. Let $\mathcal{F}_{x^*}$ be the event that $\mathrm{sgn}(x^*) = \mathrm{sgn}(x)$, and let $\mathbf{1}_{\mathcal{F}_{x^*}}$ be the indicator function of this event.

To avoid the overwhelming number of appearances of the superscripts, let $B = A^s$ throughout this proof. Then we can write $g_i^s = \mathbf{E}[(y - Bx)\mathrm{sgn}(x_i)]$. Using the fact that $\mathbf{1}_{\mathcal{F}_{x^*}} + \mathbf{1}_{\bar{\mathcal{F}}_{x^*}} = 1$ and that $\mathcal{F}_{x^*}$ happens with very high probability:

$$\begin{aligned} g_i^s &= \mathbf{E}[(y - Bx)\mathrm{sgn}(x_i)\mathbf{1}_{\mathcal{F}_{x^*}}] + \mathbf{E}[(y - Bx)\mathrm{sgn}(x_i)\mathbf{1}_{\bar{\mathcal{F}}_{x^*}}] \\ &= \mathbf{E}[(y - Bx)\mathrm{sgn}(x_i)\mathbf{1}_{\mathcal{F}_{x^*}}] \pm \gamma \end{aligned} \tag{4}$$

The key is that this allows us to essentially replace $\mathrm{sgn}(x)$ with $\mathrm{sgn}(x^*)$. Moreover, let $S = \mathrm{supp}(x^*)$. Note that when $\mathcal{F}_{x^*}$ happens S is also the support of x. Recall that according to the decoding rule (where we have replaced A^s by B for notational simplicity) $x = \mathrm{threshold}_{C/2}(B^T y)$. Therefore, $x_S = (B^T y)_S = B_S^T y = B_S^T A^* x^*$. Using the fact that the support of x is S again, we have $Bx = B_S^T B_S A^* x^*$. Plugging it into equation (4):

$$\begin{aligned} g_i^s &= \mathbf{E}[(y - Bx)\mathrm{sgn}(x_i)\mathbf{1}_{\mathcal{F}_{x^*}}] \pm \gamma = \mathbf{E}[(I - B_S B_S^T)A^* x^* \cdot \mathrm{sgn}(x_i^*)\mathbf{1}_{\mathcal{F}_{x^*}}] \pm \gamma \\ &= \mathbf{E}[(I - B_S B_S^T)A^* x^* \cdot \mathrm{sgn}(x_i^*)] - \mathbf{E}[(I - B_S B_S^T)A^* x^* \cdot \mathrm{sgn}(x_i)\mathbf{1}_{\bar{\mathcal{F}}_{x^*}}] \pm \gamma \\ &= \mathbf{E}[(I - B_S B_S^T)A^* x \cdot \mathrm{sgn}(x_i^*)] \pm \gamma \end{aligned}$$

where again we have used the fact that $\mathcal{F}_{x^*}$ happens with very high probability. Now we rewrite the expectation above using subconditioning where we first choose the support S of x^*, and then we choose the nonzero values x_S^*.

$$\begin{aligned} \mathbf{E}[(I - B_S B_S^T)A^* x^* \cdot \mathrm{sgn}(x_i^*)] &= \mathop{\mathbf{E}}_{S}\Big[\mathop{\mathbf{E}}_{x_S^*}[(I - B_S B_S^T)A^* x^* \cdot \mathrm{sgn}(x_i^*)|S]\Big] \\ &= \mathbf{E}[p_i(I - B_S B_S^T)A_i^*] \end{aligned}$$

where we use the fact that $\mathbf{E}[x_i^* \cdot \mathrm{sgn}(x_i^*)|S] = p_i$. Let $R = S - \{i\}$. Using the fact that $B_S B_S^T = B_i B_i^T + B_R B_R^T$, we can split the quantity above into two parts,

$$\begin{aligned} g_i^s &= p_i \mathbf{E}[(I - B_i B_i^T)A_i^* + p_i \mathbf{E}[B_R B_R^T]A_i^* \\ &= p_i q_i \Big(I - B_i B_i^T\Big)A_i^* + p_i \Big(B_{-i}\mathrm{diag}(q_{i,j})B_{-i}^T\Big)A_i^* \pm \gamma. \end{aligned}$$

where $\mathrm{diag}(q_{i,j})$ is a $m \times m$ diagonal matrix whose (j,j)-th entry is equal to $q_{i,j}$, and B_{-i} is the matrix obtained by zeroing out the ith column of B. Here we used the fact that $\mathbf{Pr}[i \in S] = q_i$ and $\mathbf{Pr}[i, j \in S] = q_{ij}$.

Now we set $B = A^s$, and rearranging the terms, we have $g_i^s = p_i q_i \left(\langle A_i^s, A_i^* \rangle A_i^s - A_i^* + \epsilon_i^s \pm \gamma\right)$ where $\epsilon_i^s = \Big(A_{-i}^s \mathrm{diag}(q_{i,j}) \left(A_{-i}^s\right)^T\Big)A_i^*/q_i$, which can be bounded as follows

$$\|\epsilon_i^s\| \le \|A_{-i}^s\|^2 \max_{j \neq i} q_{i,j}/q_i \le O(k/m)\|A^s\|^2 = O(k/n)$$

where the last step used the fact that $\frac{\max_{i \neq j} q_{i,j}}{\min q_i} \le O(k/m)$, which is an assumption of our generative model.

Appendix B. Approximate Gradient Descent

Here we prove Theorem 6:

Proof:[Proof of Theorem 6] We expand the error as

$$\begin{aligned}
\|z^{s+1} - z^*\|^2 &= \|z^s - z^*\|^2 - 2\eta g^{sT}(z^s - z^*) + \eta^2\|g^s\|^2 \\
&= \|z^s - z^*\|^2 - \eta\left(2g^{sT}(z^s - z^*) - \eta\|g^s\|^2\right) \\
&\leq \|z^s - z^*\|^2 - \eta\left(2\alpha\|z^s - z^*\|^2 + (2\beta - \eta)\|g^s\|^2 - 2\epsilon_s\right) \quad \text{(Definition 5 and } \eta \leq 2\beta) \\
&\leq \|z^s - z^*\|^2 - \eta\left(2\alpha\|z^s - z^*\|^2 - 2\epsilon_s\right) \\
&\leq (1 - 2\alpha\eta)\|z^s - z^*\|^2 + 2\eta\epsilon_s
\end{aligned}$$

Then solving this recurrence we have $\|z^{s+1} - z^*\|^2 \leq (1-2\alpha\eta)^{s+1}R^2 + \frac{\epsilon}{\alpha}$ where $R = \|z^0 - z^*\|$. And furthermore if $\epsilon_s < \frac{\alpha}{2}\|z^s - z^*\|^2$ we have instead

$$\|z^{s+1} - z^*\|^2 \leq (1 - 2\alpha\eta)\|z^s - z\|^2 + \alpha\eta\|z^s - z\|^2 = (1 - \alpha\eta)\|z^s - z\|^2$$

and this yields the second part of the theorem too. ■

In fact, we can extend the analysis above to obtain identical results for the case of constrained optimization. Suppose we are interested in optimizing a convex function $f(z)$ over a convex set $\mathcal{B}$. The standard approach is to take a step in the direction of the gradient (or g^s in our case) and then project into $\mathcal{B}$ after each iteration, namely, replace z^{s+1} by $\text{Proj}_{\mathcal{B}}\, z^{s+1}$ which is the closest point in $\mathcal{B}$ to z^{s+1} in Euclidean distance. It is well-known that if $z^* \in \mathcal{B}$, then $\|\text{Proj}_{\mathcal{B}}\, z - z^*\| \leq \|z - z^*\|$. Therefore we obtain the following as an immediate corollary to the above analysis:

Corollary 15 *Suppose g^s satisfies Definition 5 for $s = 1, 2, \ldots, T$ and set $0 < \eta \leq 2\beta$ and $\epsilon = \max_{s=1}^{T} \epsilon_s$. Further suppose that z^* lies in a convex set $\mathcal{B}$. Then the update rule $z^{s+1} = Proj_{\mathcal{B}}(z^s - \eta g^s)$ satisfies that for any $s = 1, \ldots, T$,*

$$\|z^s - z^*\|^2 \leq (1 - 2\alpha\eta)^s\|z^0 - z^*\|^2 + \epsilon/\alpha$$

In particular, z^s converges to z^* geometrically with systematic error ϵ/α. Additionally if $\epsilon_s < \frac{\alpha}{2}\|z^s - z^*\|^2$ for $s = 1, \ldots, T$, then

$$\|z^s - z^*\|^2 \leq (1 - \alpha\eta)^s\|z^0 - z^*\|^2$$

What remains is to derive a functional form for various update rules and show that these rules move in a direction g^s that approximately points in the direction of the desired solution z^* (under the assumption that our data is generated from a stochastic model that meets certain conditions).

Appendix C. Threshold Decoding

Here we show that a simple thresholding method recovers the support of each sample with high probability (over the randomness of x^*). This corresponds to the fact that sparse recovery for incoherent dictionaries is much easier when the non-zero coefficients do not take on a wide range of values; in particular, one does not need iterative pursuit algorithms in this case. As usual let $y = A^*x^*$ be a sample from the model, and let S be the support of x^*. Moreover suppose that A^* is μ-incoherent and let A be column-wise δ-close to A. Then

Lemma 16 *If $\frac{\mu}{\sqrt{n}} \leq \frac{1}{2k}$ and $k = \Omega^*(\log m)$ and $\delta = O^*(1/\sqrt{\log m})$, then with high probability (over the choice of x^*) we have $S = \{i : |\langle A_i, y\rangle| > C/2\}$. Also for all $i \in S$ $sgn(\langle A_i, y\rangle) = sgn(x_i^*)$.*

Consider $\langle A_i, y\rangle = \langle A_i, A_i^*\rangle x_i^* + Z_i$ where $Z_i = \sum_{j\neq i}\langle A_i, A_j^*\rangle x_j^*$ is a mean zero random variable which measures the contribution of the cross-terms. Note that $|\langle A_i, A_i^*\rangle| \geq (1 - \delta^2/2)$, so $|\langle A_i, A_i^*\rangle x_i^*|$ is either larger than $(1-\delta^2/2)C$ or equal to zero depending on whether or not $i \in S$. Our main goal is to show that the variable Z_i is much smaller than C with high probability, and this follows by standard concentration bounds.

Proof: Intuitively, Z_i has two source of randomness: the support S of x^*, and the random values of x^* conditioned on the support. We prove a stronger statement that only requires second source of randomness. Namely, even conditioned on the support S, with high probability $S = \{i : |\langle A_i, y\rangle| > C/2\}$.

We remark that Z_i is a sum of independent subgaussian random variables and the variance of Z_i is equal to $\sum_{j\in S\setminus\{i\}}\langle A_i, A_j^*\rangle^2$. Next we bound each term in the sum as

$$\langle A_i, A_j^*\rangle^2 \leq 2(\langle A_i^*, A_j^*\rangle^2 + \langle A_i - A_i^*, A_j^*\rangle^2) \leq 2\mu^2 + 2\langle A_i - A_i^*, A_j^*\rangle^2.$$

On the other hand, we know $\|A^*_{S\setminus\{i\}}\| \leq 2$ by Gershgorin's Disk Theorem. Therefore the second term can be bounded as $\sum_{j\in S\setminus\{i\}}\langle A_i - A_i^*, A_j^*\rangle^2 = \|{A^*_{S\setminus\{i\}}}^T(A_i - A_i^*)\|^2 \leq O^*(1/\log m)$. Using this bound, we know the variance is at most $O^*(1/\log m)$:

$$\sum_{j\in S\setminus\{i\}}\langle A_i, A_j^*\rangle^2 \leq 2\mu^2 k + 2\sum_{j\in S\setminus\{i\}}\langle A_i - A_i, A_j^*\rangle^2 \leq O^*(1/\log m).$$

Hence we have that Z_i is a subgaussian random variable with variance at most $O^*(1/\log m)$ and so we conclude that $Z_i \leq C/4$ with high probability. Finally we can take a union bound over all indices $i \in [m]$ and this completes the proof of the lemma. ■

In fact, even if k is much larger than $\sqrt{n}$, as long as the spectral norm of A^* is small and the support of x^* is random enough, the support recovery is still correct.

Lemma 17 *If $k = O(n/\log n)$, $\mu/\sqrt{n} < 1/\log^2 n$ and $\delta < O^*(1/\sqrt{\log m})$, the support of x^* is a uniformly k-sparse set, then with high probability (over the choice of x) we have $S = \{i : |\langle A_i, y\rangle| > C/2\}$. Also for all $i \in S$ $sgn(\langle A_i, y\rangle) = sgn(x_i^*)$.*

The proof of this lemma is very similar to the previous one. However, in the previous case we only used the randomness after conditioning on the support, but to prove this stronger lemma we need to use the randomness of the support.

First we will need the following elementary claim:

Claim 18 $\|A^{*T}A_i\| \leq O(\sqrt{m/n})$ *and* $|A^{*T}_j A_i| \leq O^*(1/\sqrt{\log m})$ *for all* $j \neq i$.

Proof: The first part follows immediately from the assumption that A^* and A are columnwise close and that $\|A^*\| = O(\sqrt{m/n})$. The second part follows because $|A^{*T}_j A_i| \leq |A^{*T}_j A^*_i| + |A^{*T}_j (A^*_i - A_i)| \leq O^*(1/\sqrt{\log m})$. ■

Let $R = S \backslash \{i\}$. Recall that conditioned on choice of S, we have $\text{var}(Z_i) = \sum_{j \in R} \langle A_i, A^*_j \rangle^2$. We will bound this term with high probability over the choice of R. First we bound its expectation:

Lemma 19 $\mathbf{E}_R[\sum_{j \in R} \langle A_i, A^*_j \rangle^2] \leq O(k/n)$

Proof: By assumption R is a uniformly random subset of $[m] \setminus \{i\}$ of size $|R|$ (this is either k or $k-1$). Then

$$\mathbf{E}[\sum_{j \in R} \langle A_i - A^*_i, A^*_j \rangle^2] = \frac{|R|}{m-1} \|A^{*T} A_i\|^2 = O(k/n),$$

where the last step uses Claim 18. ■

However bounding the expected variance of Z_i is not enough; we need a bound that holds with high probability over the choice of the support. Intuitively, we should expect to get bounds on the variance that hold with high probability because each term in the sum above (that bounds $\text{var}(Z_i)$) is itself at most $O^*(1/\log m)$, which easily implies Theorem 16.

Lemma 20 $\sum_{j \in R} \langle A_i, A^*_j \rangle^2 \leq O^*(1/\log m)$ *with high probability over the choice of* R.

Proof: Let $a_j = \langle A_i, A^*_j \rangle^2$, then $a_j = O^*(1/\log m)$ and moreover $\sum_{j \neq i} a_j = O(m/n)$ using the same idea as in the proof of Lemma 19. Hence we can apply Chernoff bounds and conclude that with high probability $\sum_{j \neq i} a_j X_j = \sum_{j \in R} \langle A_i, A^*_j \rangle^2 \leq O^*(1/\log m)$ where X_j is an indicator variable for whether or not $j \in R$. ■

Proof:[Proof of Lemma 17] Using Lemma 20 we have that with high probability over the choice of R, $var(Z_i) \leq O^* 1/\log m$. In particular, conditioned on the support R, Z_i is the sum of independent subgaussian variables and so with high probability (using Theorem **??**)

$$|Z_i| \leq O(\sqrt{var Z_i \log n}) = O^*(1).$$

Also as we saw before that $|\langle A_i, A^*_i \rangle x_i| > (1 - \delta^2/2)C$ if $i \in S$ and is zero otherwise. So we conclude that $|\langle A_i, y \rangle| > C/2$ if and only if $i \in S$ which completes the proof. ■

Remark: In the above lemma we only needs the support of x satisfy concentration inequality in Lemma 20. This does not really require S to be uniformly random.

Appendix D. Analysis of the Neural Algorithm

In Lemma 10 we gave a new (and more useful) expression that describes the update direction under the assumptions of our generative model. Here we will make crucial use of Lemma 10 in order to prove that g_i^s is $(\alpha, \beta, \epsilon)$-correlated with A_i (Lemma 22). Moreover we use Lemma 10 again to show that $\|A^{s+1} - A^*\| \leq 2\|A^*\|$ (Lemma 24). Together, these auxiliary lemmas imply that the column-wise error decreases in the next step and moreover the errors across columns are uncorrelated.

We assume that each iteration of Algorithm 2 takes infinite number of samples, and prove the corresponding simplified version of Theorem 8. The proof of this Theorem highlights the essential ideas of behind the proof of the Theorem 8, which can be found at Section G.

Theorem 21 *Suppose that A^0 is $(2\delta, 2)$-near to A^* and that $\eta = \Theta(m/k)$. Then if each update step in Algorithm 2 uses infinite number of samples at each iteration, we have*

$$\|A_i^s - A_i^*\|^2 \leq (1-\tau)^s \|A_i^0 - A_i^*\|^2 + O(k^2/n^2)$$

for some $0 < \tau < 1/2$ and for any $s = 1, 2, ..., T$. In particular it converges to A^* geometrically until the column-wise error is $O(k/n)$.

The proof is deferred to the end of this section.

D.1. Making Progress

In Lemma 10 we showed that $g_i^s = p_i q_i(\lambda_i A_i^s - A_i^* + \epsilon_i^s + \gamma)$ where $\lambda_i = \langle A_i, A_i^* \rangle$. Here we will prove that g_i^s is $(\alpha, \beta, \epsilon)$-correlated with A_i^*. Recall that we fixed $\delta = O^*(1/\log n)$. The main intuition is that g_i^s is mostly equal to $p_i q_i(A_i^s - A_i^*)$ with a small error term.

Lemma 22 *If a vector g_i^s is equal to $4\alpha(A_i^s - A_i^*) + v$ where $\|v\| \leq \alpha\|A_i^s - A_i^*\| + \zeta$, then g_i^s is $(\alpha, 1/100\alpha, \zeta^2/\alpha)$-correlated with A_i^*, more specifically,*

$$\langle g_i^s, A_i^s - A_i^* \rangle \geq \alpha\|A_i^s - A_i^*\|^2 + \frac{1}{100\alpha}\|g_i\|^2 - \zeta^2/\alpha.$$

In particular, g_i^s is $(\alpha, \beta, \epsilon)$-correlated with A_i^*, where $\alpha = \Omega(k/m)$, $\beta \geq \Omega(m/k)$ and $\epsilon = O(k^3/mn^2)$. We can now apply Theorem 6 and conclude that the column-wise error gets smaller in the next step:

Corollary 23 *If A^s is $(2\delta, 2)$-near to A^* and $\eta \leq \min_i(p_i q_i(1-\delta)) = O(m/k)$, then $g_i^s = p_i q_i(\lambda_i A_i^s - A_i^* + \epsilon_i^s + \gamma)$ is $(\Omega(k/m), \Omega(m/k), O(k^3/mn^2))$-correlated with A_i^*, and further*

$$\|A_i^{s+1} - A_i^*\|^2 \leq (1 - 2\alpha\eta)\|A_i^s - A_i^*\|^2 + O(\eta k^2/n^2)$$

Proof:[Proof of Lemma 22] Throughout this proof s is fixed and so we will omit the superscript s to simplify notations. By the assumption, g_i already has a component that is pointing to the correct direction $A_i - A_i^*$, we only need to show that the norm of the extra term v is small enough. First we can bound the norm of g_i by triangle inequality: $\|g_i\| \leq \|4\alpha(A_i - A_i^*)\| + \|v\| \leq 5\alpha\|(A_i - A_i^*)\| + \zeta$, therefore $\|g_i\|^2 \leq 50\alpha^2\|(A_i - A_i^*)\|^2 + 2\zeta^2$.

Also, we can bound the inner-product between g_i and $A_i - A_i^*$ by $\langle g_i, A_i - A_i^* \rangle \geq 4\alpha \|A_i - A_i^*\|^2 - \|v\| \|A_i - A_i^*\|$.

Using these bounds, we will show $\langle g_i, A_i - A_i^* \rangle - \alpha \|A_i - A_i^*\|^2 - \frac{1}{100\alpha}\|g_i\|^2 + \zeta^2/\alpha \geq 0$. Indeed we have

$$\begin{aligned}
&\langle g_i, A_i - A_i^* \rangle - \alpha \|A_i - A_i^*\|^2 - \frac{1}{100\alpha}\|g_i\|^2 + \zeta^2/\alpha \\
&\quad \geq \quad 4\alpha \|A_i - A_i^*\|^2 - \|v\| \|A_i - A_i^*\| - \alpha \|A_i - A_i^*\|^2 - \frac{1}{100\alpha}\|g_i\|^2 + \zeta^2/\alpha \\
&\quad \geq \quad 3\alpha \|A_i - A_i^*\|^2 - (\alpha \|A_i - A_i^*\| + \zeta) \|A_i - A_i^*\| - \frac{1}{100\alpha}(50\alpha^2 \|(A_i - A_i^*)\|^2 + 2\zeta^2) + \zeta^2/\alpha \\
&\quad \geq \quad \alpha \|A_i - A_i^*\|^2 - \zeta \|A_i - A_i^*\| + \frac{1}{4}\zeta^2/\alpha \\
&\quad = \quad (\sqrt{\alpha}\|A_i - A_i^*\| - \zeta/2\sqrt{\alpha})^2 \geq 0.
\end{aligned}$$

This completes the proof of the lemma. ■

Proof:[Proof of Corollary 23] We use the form in Lemma 10, $g_i^s = p_i q_i(\lambda_i A_i^s - A_i^* + \epsilon_i^s + \gamma)$ where $\lambda_i = \langle A_i, A_i^* \rangle$. We can write $g_i^s = p_i q_i(A_i^s - A_i^*) + p_i q_i((1-\lambda_i)A_i^s + \epsilon_i^s + \gamma)$, so when applying Lemma 22 we can use $4\alpha = p_i q_i = \Theta(k/m)$ and $v = p_i q_i((1-\lambda_i)A_i^s + \epsilon_i^s + \gamma)$. The norm of v can be bounded in two terms, the first term $p_i q_i(1-\lambda_i)A_i^s$ has norm $p_i q_i(1-\lambda_i)$ which is smaller than $p_i q_i \|A_i^s - A_i^*\|$, and the second term has norm bounded by $\zeta = O(k^2/mn)$.

By Lemma 22 we know the vector g_s^i is $(\Omega(k/m), \Omega(m/k), O(k^3/mn^2))$-correlated with A^s. Then by Theorem 6 we have the last part of the corollary. ■

D.2. Maintaining Nearness

Lemma 24 *Suppose that A^s is $(2\delta, 2)$-near to A^*. Then $\|A^{s+1} - A^*\| \leq 2\|A^*\|$ in Algorithm 2.*

Proof: As in the proof of the previous lemma, we will make crucial use of Lemma 10. Substituting and rearranging terms we have:

$$\begin{aligned}
A_i^{s+1} - A_i^* &= A_i^s - A_i^* - \eta g_i^s \\
&= (1 - \eta p_i q_i)(A_i^s - A_i^*) + \eta p_i q_i (1 - \lambda_i^s) A_i^s - \eta p_i \Big(A_{-i}^s \mathrm{diag}(q_{i,j}) \left(A_{-i}^s\right)^T \Big) A_i^* \pm \gamma
\end{aligned}$$

Our first goal is to write this equation in a more convenient form. In particular let U and V be matrices such that $U_i = p_i q_i (1 - \lambda_i^s) A_i^s$ and $V_i = p_i \Big(A_{-i}^s \mathrm{diag}(q_{i,j}) \left(A_{-i}^s\right)^T \Big) A_i^*$. Then we can re-write the above equation as:

$$A^{s+1} - A^* = (A^s - A^*)\mathrm{diag}(1 - \eta p_i q_i) + \eta U - \eta V \pm \gamma$$

where $\mathrm{diag}(1 - \eta p_i q_i)$ is the $m \times m$ diagonal matrix whose entries along the diagonal are $1 - \eta p_i q_i$.

We will bound the spectral norm of $A^{s+1} - A$ by bounding the spectral norm of each of the matrices of right hand side. The first two terms are straightforward to bound:

$$\|(A^s - A^*)\mathrm{diag}(1 - \eta p_i q_i)\| \leq \|A^s - A^*\| \cdot (1 - \eta \min_i p_i q_i) \leq 2(1 - \Omega(\eta k/m))\|A^*\|$$

where the last inequality uses the assumption that $p_i = \Theta(1)$ and $q_i \leq O(k/m)$, and the assumption that $\|A^s - A^*\| \leq 2\|A^*\|$.

From the definition of U it follows that $U = A^s \mathrm{diag}(p_i q_i (1 - \lambda_i^s))$, and therefore

$$\|U\| \leq \delta \max_i p_i q_i \|A^s\| = o(k/m) \cdot \|A^*\|$$

where we have used the fact that $\lambda_i^s \geq 1 - \delta$ and $\delta = o(1)$, and $\|A^s\| \leq \|A^s - A^*\| + \|A^*\| = O(\|A^*\|)$.

What remains is to bound the third term, and let us first introduce an auxiliary matrix Q which we define as follows: $Q_{ii} = 0$ and $Q_{i,j} = q_{i,j}\langle A_i^s, A_i^*\rangle$ for $i \neq j$. It is easy to verify that the following claim:

Claim 25 *The ith column of $A^s Q$ is equal to $\left(A^s_{-i} diag(q_{i,j}) \left(A^s_{-i}\right)^T\right) A_i^*$*

Therefore we can write $V = A^s Q \mathrm{diag}(p_i)$. We will bound the spectral norm of Q by bounding its Frobenus norm instead. Then from the definition of A, we have that:

$$\|Q\|_F \leq \left(\max_{i \neq j} q_{ij}\right) \sum_{i \neq j} \sqrt{\langle A_i^s, A_j^*\rangle^2} = O(k^2/m^2)\|A^{*T} A^s\|_F$$

Moreover since $A^{*T} A^s$ is an $m \times m$ matrix, its Frobenius norm can be at most a $\sqrt{m}$ factor larger than its spectral norm. Hence we have

$$\begin{aligned} \|V\| \leq \left(\max_i p_i\right) \|A^s\|\|Q\| &\leq O(k^2\sqrt{m}/m^2)\|A^s\|^2\|A^*\| \\ &\leq o(k/m)\|A^*\| \end{aligned}$$

where the last inequality uses the fact that $k = O(\sqrt{n}/\log n)$ and $\|A^s\| \leq O(\|A^*\|)$.

Therefore, putting the pieces together we have:

$$\begin{aligned} \|A^{s+1} - A^*\| &\leq \|(A^s - A^*)\mathrm{diag}(1 - \eta p_i q_i)\| + \|\eta U\| + \|\eta V\| \pm \gamma \\ &\leq 2(1 - \Omega(\eta k/m))\|A\| + o(\eta k/m)\|A^*\| + o(\eta k/m)\|A^*\| \pm \gamma \\ &\leq 2\|A^*\| \end{aligned}$$

and this completes the proof of the lemma. ■

D.3. Proof of Theorem 21

We prove by induction on s. Our induction hypothesis is that the theorem is true at each step s and A^s is $(2\delta, 2)$-near to A^*. The hypothesis is trivially true for $s = 0$. Now assuming the inductive hypothesis is true. Recall that Corollary 23 of Section D.1 says that if A^s is $(2\delta, 2)$-near to A^*, which is guaranteed by the inductive hypothesis, by then

g_i^s is indeed $(\Omega(k/m), \Omega(m/k), O(k^3/mn^2))$-correlated with A_i^*. Invoking our framework of analysis (Theorem 6), we have that

$$\|A_i^{s+1} - A_i^*\|^2 \leq (1-\tau)\|A_i^s - A_i^*\|^2 + O(k^2/n^2) \leq (1-\tau)^{s+1}\|A_i^0 - A_i^*\|^2 + O(k^2/n^2)$$

Therefore it also follows that A^{s+1} is 2δ-close to A^*. Then we invoke Lemma 24 to prove A^{s+1} has not too large spectral norm $\|A^{s+1} - A^*\| \leq 2\|A^*\|$, which completes the induction.

Appendix E. More Alternating Minimization

Here we prove Theorem 11 and Theorem 12. Note that in Algorithm 4 and Algorithm 5, we use the expectation of the gradient over the samples instead of the empirical average. We can show that these algorithms would maintain the same guarantees if we used $p = \widetilde{\Omega}(mk)$ to estimate g^s as we did in Algorithm 2. However these proofs would require repeating very similar calculations to those that we performed in Appendix G, and so we only claim that these algorithms maintain their guarantees if they use a polynomial number of samples to approximate the expectation.

E.1. Proof of Theorem 11

We give a variant of the Olshausen-Field update rule in Algorithm 4. Our first goal is to prove that each column of g^s is $(\alpha, \beta, \epsilon)$-correlated with A_i^*. The main step is to prove an analogue of Lemma 10 that holds for the new update rule.

Lemma 26 *Suppose that A^s is $(2\delta, 5)$-near to A^* Then each column of g^s in Algorithm 4 takes the form*

$$g_i^s = q_i\left((\lambda_i^s)^2 A_i^s - \lambda_i A_i^s + \epsilon_i^s\right)$$

where $\lambda_i = \langle A_i, A_i^*\rangle$. Moreover the norm of ϵ_i^s can be bounded as $\|\epsilon_i^s\| \leq O(k^2/mn)$.

We remark that unlike the statement of Lemma 22, here we will not explicitly state the functional form of ϵ_i^s because we will not need it.

Proof: The proof parallels that of Lemma 10, although we will use slightly different conditioning arguments as needed. Again, we define $\mathcal{F}_{x^*}$ as the event that $\mathrm{sgn}(x^*) = \mathrm{sgn}(x)$, and let $\mathbf{1}_{\mathcal{F}_{x^*}}$ be the indicator function of this event. We can invoke Lemma 16 and conclude that this event happens with high probability. Moreover let $\mathcal{F}_i$ be the event that i is in the set $S = \mathrm{supp}(x^*)$ and let $\mathbf{1}_{\mathcal{F}_i}$ be its indicator function.

When event $\mathcal{F}_{x^*}$ happens, the decoding satisfies $x_S = A_S^T A_S^* x_S^*$ and all the other entries are zero. Throughout this proof s is fixed and so we will omit the superscript s for notational convenience. We can now rewrite g_i as

$$\begin{aligned}
g_i &= \mathbf{E}[(y - Ax)x^T] = \mathbf{E}[(y - Ax)x_i^T \mathbf{1}_{\mathcal{F}_{x^*}}] + \mathbf{E}[(y - Ax)x_i^T(1 - \mathbf{1}_{\mathcal{F}_{x^*}})] \\
&= \mathbf{E}\left[(I - A_S^T A_S)A_S^* x_S^* x_S^{*T} A_S^{*T} A_i \mathbf{1}_{\mathcal{F}_{x^*}} \mathbf{1}_{\mathcal{F}_i}\right] \pm \gamma \\
&= \mathbf{E}\left[(I - A_S^T A_S)A_S^* x_S^* x_S^{*T} A_S^{*T} A_i \mathbf{1}_{\mathcal{F}_i}\right] \\
&= \mathbf{E}\left[(I - A_S^T A_S)A_S^* x_S^* x_S^{*T} A_S^{*T} A_i \mathbf{1}_{\mathcal{F}_i}\right] \pm \gamma
\end{aligned}$$

Algorithm 4 Olshausen-Field Update Rule

Initialize A^0 that is $(\delta_0, 2)$-near to A^*
Repeat for $s = 0, 1, ..., T$

Decode: $x = \text{threshold}_{C/2}((A^s)^T y)$ for each sample y

Update: $A^{s+1} = A^s - \eta g^s$ where $g^s = \mathbf{E}[(y - A^s x)x^T]$

Project: $A^{s+1} = \text{Proj}_{\mathcal{B}} A^{s+1}$(where $\mathcal{B}$ is defined in Definition 28)

Once again our strategy is to rewrite the expectation above using subconditioning where we first choose the support S of x^*, and then we choose the nonzero values x^*_S.

$$\begin{aligned}
g_i &= \mathop{\mathbf{E}}_{S}\left[\mathop{\mathbf{E}}_{x^*_S}[(I - A_S^T A_S)A^*_S x^*_S x^{*T}_S A^{*T}_S A_i \mathbf{1}_{\mathcal{F}_i}|S]\right] \pm \gamma \\
&= \mathbf{E}\left[(I - A_S A_S^T)A^*_S A^{*T}_S A_i \mathbf{1}_{\mathcal{F}_i}\right] \pm \gamma \\
&= \mathbf{E}\left[(I - A_i A_i^T - A_R {A_R}^T)(A^*_i A^{*T}_i + A^*_R {A^*_R}^T)A_i \mathbf{1}_{\mathcal{F}_i}\right] \pm \gamma \\
&= \mathbf{E}\left[(I - A_i A_i^T)(A^*_i A^{*T}_i)A_i \mathbf{1}_{\mathcal{F}_i}\right] + \mathbf{E}\left[(I - A_i A_i^T)A^*_R {A^*_R}^T A_i \mathbf{1}_{\mathcal{F}_i}\right] \\
&\quad - \mathbf{E}\left[A_R {A_R}^T A^*_i A^{*T}_i A_i \mathbf{1}_{\mathcal{F}_i}\right] - \mathbf{E}\left[A_R A_R^T A^*_R {A^*_R}^T A_i \mathbf{1}_{\mathcal{F}_i}\right] \pm \gamma
\end{aligned}$$

Next we will compute the expectation of each of the terms on the right hand side. This part of the proof will be somewhat more involved than the proof of Lemma 10, because the terms above are quadratic instead of linear. The leading term is equal to $q_i(\lambda_i A^*_i - \lambda_i^2 A_i)$ and the remaining terms contribute to ϵ_i. The second term is equal to $(I - A_i A_i^T)A^*_{-i}\text{diag}(q_{i,j}){A^*_{-i}}^T A_i$ which has spectral norm bounded by $O(k^2/mn)$. The third term is equal to $\lambda_i A_{-i}\text{diag}(q_{i,j}){A^*_{-i}}^T A^*_i$ which again has spectral norm bounded by $O(k^2/mn)$. The final term is equal to

$$\begin{aligned}
\mathbf{E}\left[A_R A_R^T A^*_R {A^*_R}^T A_i \mathbf{1}_{\mathcal{F}_i}\right] &= \sum_{j_1, j_2 \neq i} \mathbf{E}[(A_{j_1} A_{j_1}^T)(A^*_{j_2} {A^*_{j_2}}^T)A_i \mathbf{1}_{\mathcal{F}_i} \mathbf{1}_{\mathcal{F}_{j_1}} \mathbf{1}_{\mathcal{F}_{j_2}}] \\
&= \sum_{j_1 \neq i} \left(\sum_{j_2 \neq i} q_{i,j_1,j_2} \langle A^*_{j_2}, A_i \rangle \langle A^*_{j_2}, A_{j_1} \rangle \right) A_{j_1} \\
&= A_{-i} v.
\end{aligned}$$

where v is a vector whose j_2-th component is equal to $\sum_{j_2 \neq i} q_{i,j_1,j_2} \langle A^*_{j_2}, A_i \rangle \langle A^*_{j_2}, A_{j_1} \rangle$. The absolute value of v_{j_2} is bounded by

$$\begin{aligned}
|v_{j_2}| &\leq O(k^2/m^2)|\langle A^*_{j_2}, A_i \rangle| + O(k^3/m^3)\Big(\sum_{j_2 \neq j_1, i} (\langle A^*_{j_2}, A_i \rangle^2 + \langle A^*_{j_2}, A_{j_1} \rangle^2)\Big) \\
&\leq O(k^2/m^2)|\langle A^*_{j_2}, A_i \rangle| + O(k^3/m^3)\|A^*\|^2 = O(k^2/m^2)(|\langle A^*_{j_2}, A_i \rangle| + k/n).
\end{aligned}$$

The first inequality uses bounds for q's and the AM-GM inequality, the second inequality uses the spectral norm of A^*. We can now bound the norm of v as follows

$$\|v\| \leq O(k^2/m^2 \cdot \sqrt{m/n})$$

and this implies that the last term satisfies $\|A_{-i}\|\|v\| \leq O(k^2/mn)$. Combining all these bounds completes the proof of the lemma. ■

We are now ready to prove that the update rule satisfies Definition 5. This again uses Lemma 22, except that we invoke Lemma 26 instead. Combining these lemmas we obtain:

Lemma 27 *Suppose that A^s is $(2\delta, 5)$-near to A^*. Then for each i, g_i^s as defined in Algorithm 4 is $(\alpha, \beta, \epsilon)$-correlated with A_i^*, where $\alpha = \Omega(k/m)$, $\beta \geq \Omega(m/k)$ and $\epsilon = O(k^3/mn^2)$.*

Notice that in the third step in Algorithm 4 we project back (with respect to Frobenius norm of the matrices) into a convex set $\mathcal{B}$ which we define below. Viewed as minimizing a convex function with convex constraints, this projection can be computed by various convex optimization algorithm, e.g. subgradient method (see Theorem 3.2.3 of Section 3.2.4 of Nesterov's seminal Book **?** for more detail). Without this modification, it seems that the update rule given in Algorithm 4 does not necessarily preserve nearness.

Definition 28 *Let $\mathcal{B} = \{A | A$ is δ_0 close to A^0 and $\|A\| \leq 2\|A^*\|\}$*

The crucial properties of this set are summarized in the following claim:

Claim 29 *(a) $A^* \in \mathcal{B}$ and (b) for each $A \in \mathcal{B}$, A is $(2\delta_0, 5)$-near to A^**

Proof: The first part of the claim follows because by assumption A^* is δ_0-close to A^0 and $\|A^* - A^0\| \leq 2\|A^*\|$. Also the second part follows because $\|A - A^*\| \leq \|A - A^0\| + \|A^0 - A^*\| \leq 4\|A^*\|$. This completes the proof of the claim. ■

By the convexity of $\mathcal{B}$ and the fact that $A^* \in \mathcal{B}$, we have that projection doesn't increase the error in Frobenius norm.

Claim 30 *For any matrix A, $\|Proj_{\mathcal{B}} A - A^*\|_F \leq \|A - A^*\|_F$.*

We now have the tools to analyze Algorithm 4 by fitting it into the framework of Corollary 15. In particular, we prove that it converges to a globally optimal solution by connecting it to an approximate form of *projected* gradient descent:

Proof: [Proof of Theorem 11] We note that projecting into $\mathcal{B}$ ensures that at the start of each step $\|A^s - A^*\| \leq 5\|A^*\|$. Hence g_i^s is $(\Omega(k/m), \Omega(m/k), O(k^3/mn^2))$-correlated with A_i^* for each i, which follows from Lemma 27. This implies that g^s is $(\Omega(k/m), \Omega(m/k), O(k^3/n^2))$-correlated with A^* in Frobenius norm. Finally we can apply Corollary 15 (on the matrices with Frobenius) to complete the proof of the theorem. ■

Algorithm 5 Unbiased Update Rule

Initialize A^0 that is $(\delta_0, 2)$-near to A^*
Repeat for $s = 0, 1, ..., T$

Decode: $x = \text{threshold}_{C/2}((A^s)^T y)$ for each sample y

$\bar{x}^i = \text{threshold}_{C/2}((B^{(s,i)})^T y)$ for each sample y, and each $i \in [m]$

Update: $A_i^{s+1} = A_i^s - \eta g_i^s$ where $g_i^s = \mathbf{E}[(y - B^{(s,i)}\bar{x}^i)\text{sgn}(x)_i^T]$ for each $i \in [m]$

E.2. Proof of Theorem 12

The proof of Theorem 12 is parallel to that of Theorem 21 and Theorem 11. As usual, our first step is to show that g_s is correlated with A^*:

Lemma 31 *Suppose that A^s is $(\delta, 5)$-near to A^*. Then for each i, g_i^s as defined in Algorithm 5 is $(\alpha, \beta, \epsilon)$-correlated with A_i^*, where $\alpha = \Omega(k/m)$, $\beta \geq \Omega(m/k)$ and $\epsilon \leq n^{-\omega(1)}$.*

Proof: We chose to write the proof of Lemma 10 so that we can reuse the calculation here. In particular, instead of substituting B for A^s in the calculation we can substitute $B^{(s,i)}$ instead and we get:

$$g^{(s,i)} = p_i q_i(\lambda_i^s A_i^s - A_i^* + B_{-i}^{(s,i)}\text{diag}(q_{i,j})B_{-i}^{(s,i)T}A_i^*) + \gamma.$$

Recall that $\lambda_i^s = \langle A_i^s, A_i^* \rangle$. Now we can write $g^{(s,i)} = p_i q_i(A_i^s - A_i^*) + v$, where

$$v = p_i q_i(\lambda_i^s - 1)A_i^s + p_i q_i B_{-i}^{(s,i)}\text{diag}(q_{i,j})B_{-i}^{(s,i)T}A_i^* + \gamma$$

Indeed the norm of the first term $p_i q_i(\lambda_i^s - 1)A_i^s$ is smaller than $p_i q_i \|A_i^s - A_i^*\|$.

Recall that the second term was the main contribution to the systemic error, when we analyzed earlier update rules. However in this case we can use the fact that $B_{-i}^{(s,i)T}A_i^s = 0$ to rewrite the second term above as

$$p_i q_i B_{-i}^{(s,i)}\text{diag}(q_{i,j})B_{-i}^{(s,i)T}(A_i^* - A_i^s)$$

Hence we can bound the norm of the second term by $O(k^2/mn)\|A_i^* - A_i^s\|$, which is also much smaller than $p_i q_i \|A_i^s - A_i^*\|$.

Combining these two bounds we have that $\|v\| \leq p_i q_i \|A_i^s - A_i^*\|/4 + \gamma$, so we can take $\zeta = \gamma = n^{-\omega(1)}$ in Lemma 22. We can complete the proof by invoking Lemma 22 which implies that the $g^{(s,i)}$ is $(\Omega(k/m), \Omega(m/k), n^{-\omega(1)})$-correlated with A_i. ■

This lemma would be all we would need, if we added a third step that projects onto $\mathcal{B}$ as we did in Algorithm 4. However here we do not need to project at all, because the update rule maintains nearness and thus we can avoid this computationally intensive step.

Lemma 32 *Suppose that A^s is $(\delta, 2)$-near to A^*. Then $\|A^{s+1} - A^*\| \leq 2\|A^*\|$ in Algorithm 5.*

This proof of the above lemma parallels that of Lemma 24. We will focus on highlighting the differences in bounding the error term, to avoid repeating the same calculation.

Proof: [sketch] We will use A to denote A^s and $B^{(i)}$ to denote $B^{(s,i)}$ to simplify the notation. Also let $\bar{A}_i$ be normalized so that $\bar{A}_i = A_i/\|A_i\|$ and then we can write $B^{(i)}_{-i} = (I - \bar{A}_i\bar{A}_i^T)A_{-i}$. Hence the error term is given by

$$(I - \bar{A}_i\bar{A}_i^T)A_{-i}\text{diag}(q_{i,j})A_{-i}^T(I - \bar{A}_i\bar{A}_i^T)A_i^*$$

Let C be a matrix whose columns are $C_i = (I - \bar{A}_i\bar{A}_i^T)A_i^* = A_i - \langle \bar{A}_i, A_i^* \rangle \bar{A}_i$. This implies that $\|C\| \leq O(\sqrt{m/n})$. We can now rewrite the error term above as

$$A_{-i}\text{diag}(q_{i,j})A_{-i}^T C_i - (\bar{A}_i\bar{A}_i)^T A_{-i}\text{diag}(q_{i,j})A_{-i}^T C_i$$

It follows from the proof of Lemma 24 that the first term above has spectral norm bounded by $O(k/m \cdot \sqrt{m/n})$. This is because in Lemma 24 we bounded the term $A_{-i}\text{diag}(q_{i,j})A_{-i}^T A_i^*$ and in fact it is easily verified that all we used in that proof was the fact that $\|A^*\| = O(\sqrt{m/n})$, which also holds for C.

All that remains is to bound the second term. We note that its columns are scalar multiples of $\bar{A}_i$, where the coefficient can be bounded as follows: $\|\bar{A}_i\|\|A_{-i}\|^2\|\text{diag}(q_{i,j})\|\|A_i^*\| \leq O(k^2/mn)$. Hence we can bound the spectral norm of the second term iby $O(k^2/mn)\|\bar{A}\| = O^*(k/m \cdot \sqrt{m/n})$. We can now combine these two bounds, which together with the calculation in Lemma 24 completes the proof. ■

These two lemmas directly imply Theorem 12.

Appendix F. Analysis of Initialization

Here we prove an infinite sample version of Theorem 13 by repeatedly invoking Lemma 14. We give sample complexity bounds for it in AppendixG.3 where we complete the proof of Theorem 13.

Theorem 33 *Under the assumption of Theorem 13, if Algorithm 3 has access to $M_{u,v}$ (defined in Lemma 14) instead of the empirical average $\widehat{M}_{u,v}$, then with high probability A is $(\delta, 2)$-near to A^* where $\delta = O^*(1/\log n)$.*

Our first step is to use Lemma 14 to show that when u and v share a unique dictionary element, there is only one large term in $M_{u,v}$ and the error terms are small. Hence the top singular vector of $M_{u,v}$ must be close to the corresponding dictionary element A_i.

Lemma 34 *Under the assumptions of Theorem 13, suppose $u = A^*\alpha$ and $v = A^*\alpha'$ are two random samples with supports U, V respectively. When $U \cap V = \{i\}$ the top singular vector of $M_{u,v}$ is $O^*(1/\log n)$-close to A_i^*.*

Proof: When u and v share a unique dictionary element i, the contribution of the first term in (3) is just $q_i c_i \beta_i \beta_i' A_i^* A_i^{*T}$. Moreover the coefficient $q_i c_i \beta_i \beta_i'$ is at least $\Omega(k/m)$ which follows from Lemma 14 and from the assumptions that $c_i \geq 1$ and $q_i = \Omega(k/m)$.

On the other hand, the error terms are bounded by $\|E_1 + E_2 + E_3\| \leq O^*(k/m \log m)$ which again by Lemma 14. We can now apply Wedin's Theorem (see e.g. Horn and Johnson (1990)) to

$$M_{u,v} = q_i c_i \beta_i \beta_i' A_i^* A_i^{*T} + \underbrace{(E_1 + E_2 + E_3)}_{\text{perturbation}}$$

and conclude that its top singular vector must be $O^*(k/m \log m)/\Omega(k/m) = O^*(1/\log m)$-close to A_i^*, and this completes the proof of the lemma. ■

Using (3) again, we can verify whether or not the supports of u and v share a unique element.

Lemma 35 *Suppose $u = A^*\alpha$ and $v = A^*\alpha'$ are two random samples with supports U, V respectively. Under the assumption of Theorem 13, if the top singular value of $M_{u,v}$ is at least $\Omega(k/m)$ and the second largest singular value is at most $O^*(k/m \log m)$, then with high probability u and v share a unique dictionary element.*

Proof: By Lemma 14 we know with high probability the error terms have spectral norm $O^*(k/m \log m)$. Here we show when that happens, and the top singular value is at least $\Omega(k/m)$, second largest singular value is at most $O^*(k/m \log m)$, then u and v must share a unique dictionary element.

If u and v share no dictionary element, then the main part in Equation (3) is empty, and the error term has spectral norm $O^*(k/m \log m)$. In this case the top singular value of $M_{u,v}$ cannot be as large as $\Omega(k/m)$.

If u and v share more than one dictionary element, there are more than one terms in the main part of (3). Let $S = U \cap V$, we know $M_{u,v} = A_S^* D_S A_S^{*T} + E_1 + E_2 + E_3$ where D_S is a diagonal matrix whose entries are equal to $q_i c_i \beta_i \beta_i'$. All diagonal entries in D_S have magnitude at least $\Omega(k/m)$. By incoherence we know A_S^* have smallest singular value at least $1/2$, therefore the second largest singular value of $A_S^* D_S A_S^{*T}$ is at least:

$$\sigma_2(A_S^* D_S A_S^{*T}) \geq \sigma_{min}(A_S^*)^2 \sigma_2(D_S) \geq \Omega(k/m).$$

Finally by Weyl's theorem (see e.g. Horn and Johnson (1990)) we know $\sigma_2(M_{u,v}) \geq \sigma_2(A_S^* D_S A_S^{*T}) - \|E_1 + E_2 + E_3\| \geq \Omega(k/m)$. Therefore in this case the second largest singular value cannot be as small as $O^*(k/m \log m)$.

Combining the above two cases, we know when the top two singular values satisfy the conditions in the lemma, and the error terms are small, u and v share a unique dictionary element. ■

Finally, we are ready to prove Theorem 13. The idea is every vector added to the list L will be close to one of the dictionary elements (by Lemma 35), and for every dictionary element the list L contains at least one close vector because we have enough random samples.

Proof:[Proof of Theorem 33] By Lemma 35 we know every vector added into L must be close to one of the dictionary elements. On the other hand, for any dictionary element A_i^*,

by the bounded moment condition of $\mathcal{D}$ we know

$$\begin{aligned}\mathbf{Pr}[|U \cap V| = \{i\}] &= \mathbf{Pr}[i \in U]\,\mathbf{Pr}[i \in V]\,\mathbf{Pr}[(U \cap V)\backslash\{i\} = \emptyset | i \in U, j \in U] \\ &\geq \mathbf{Pr}[i \in U]\,\mathbf{Pr}[i \in V](1 - \textstyle\sum_{j \neq i, j \in [m]} \mathbf{Pr}[j \in U \cap V | i \in U, j \in V]) \\ &= \Omega(k^2/m^2) \cdot (1 - m \cdot O(k^2/m^2)) \\ &= \Omega(k^2/m^2).\end{aligned}$$

Here the inequality uses union bound. Therefore given $O(m^2 \log^2 n/k^2)$ trials, with high probability there is a pair of u,v that intersect uniquely at i for all $i \in [m]$. By Lemma 34 this implies there must be at least one vector that is close to A_i^* for all dictionary elements.

Finally, since all the dictionary elements have distance at least $1/2$ (by incoherence), the connected components in L correctly identifies different dictionary elements. The output A must be $O^*(1/\log m)$ close to A^*. ■

We now come to the proof of the main lemma:

Proof:[Proof of Lemma 14] We will prove this lemma in three parts. First we compute the expectation and show it has the desired form. Recall that $y = A^*x^*$, and so:

$$\begin{aligned}M_{u,v} &= \mathop{\mathbf{E}}_{S}\Big[\mathop{\mathbf{E}}_{x_S^*}[\langle u, A_S^* x_S^*\rangle\langle v, A_S^* x_S^*\rangle A_S^* x_S^* x_S^{*T} A_S^{*T} | S]\Big] \\ &= \mathop{\mathbf{E}}_{S}\Big[\mathop{\mathbf{E}}_{x_S^*}[\langle \beta, x_S^*\rangle\langle \beta', x_S^*\rangle A_S^* x_S^* x_S^{*T} A_S^{*T} | S]\Big] \\ &= \mathop{\mathbf{E}}_{S}\Big[\sum_{i\in S} c_i\beta_i\beta_i' A_i^* A_i^{*T} + \sum_{i,j\in S, i\neq j}\Big(\beta_i\beta_i' A_j^* A_j^{*T} + \beta_i\beta_j' A_i^* A_j^{*T} + \beta_i'\beta_j A_i^* A_j^{*T}\Big)\Big] \\ &= \sum_{i\in[m]} q_i c_i \beta_i\beta_i' A_i^* A_i^{*T} + \sum_{i,j\in[m], i\neq j} q_{i,j}\Big(\beta_i\beta_i' A_j^* A_j^{*T} + \beta_i\beta_j' A_i^* A_j^{*T} + \beta_i'\beta_j A_i^* A_j^{*T}\Big)\end{aligned}$$

where the second-to-last line follows because the entries in x_S^* are independent and have mean zero, and the only non-zero terms come from x_i^{*4} (whose expectation is c_i) and $x_i^{*2}x_j^{*2}$ (whose expectation is one). Equation (3) now follows by rearranging the terms in the last line. What remains is to bound the spectral norm of E_1, E_2 and E_3.

Next we establish some useful properties of β and β':

Claim 36 *With high probability it holds that (a) for each i we have $|\beta_i - \alpha_i| \leq \frac{\mu k \log m}{\sqrt{n}}$ and (b) $\|\beta\| \leq O(\sqrt{mk/n})$.*

In particular since the difference between β_i an α_i is $o(1)$ for our setting of parameters, we conclude that if $\alpha_i \neq 0$ then $C - o(1) \leq |\beta_i| \leq O(\log m)$ and if $\alpha_i = 0$ then $|\beta_i| \leq \frac{\mu k \log m}{\sqrt{n}}$.

Proof: Recall that U is the support of α and let $R = U\backslash\{i\}$. Then:

$$\beta_i - \alpha_i = A_i^{*T} A_U^* \alpha_U - \alpha_i = A_i^{*T} A_R^* \alpha_R$$

and since A^* is incoherent we have that $\|A_i^{*T} A_R^*\| \leq \mu\sqrt{k/n}$. Moreover the entries in α_R are independent and subgaussian random variables, and so with high probability $|\langle A_i^{*T} A_R^*, \alpha_R\rangle| \leq \frac{\mu k \log m}{\sqrt{n}}$ and this implies the first part of the claim.

For the second part, we can bound $\|\beta\| \leq \|A^*\|\|A_U^*\|\|\alpha\|$. Since α is a k-sparse vector with independent and subgaussian entries, with high probability $\|\alpha\| \leq O(\sqrt{k})$ in which case it follows that $\|\beta\| \leq O(\sqrt{mk/n})$. ■

Now we are ready to bound the error terms.

Claim 37 *With high probability each of the error terms E_1, E_2 and E_3 in (3) has spectral norm bounded at most $O^*(k/m \log m)$.*

Proof: Let $S = [m]\backslash(U \cap V)$, then $E_1 = A_S^* D_1 A_S^{*T}$ where D_1 is a diagonal matrix whose entries are $q_i c_i \beta_i \beta_i'$. We first bound $\|D_1\|$. To this end, we can invoke the first part of Claim 36 to conclude that $|\beta_i \beta_i'| \leq \frac{\mu^2 k^2 \log^2 m}{n}$. Also $q_i c_i = \Theta(k/m)$ and so

$$\|D_1\| \leq O\left(\frac{\mu^2 k^3 \log^2 m}{mn}\right) = O\left(\frac{\mu k^2 \log^2 n}{m\sqrt{n}}\right) = O^*(k/m \log m)$$

Finally $\|A_S\| \leq \|A\| \leq O(1)$ (where we have used the assumption that $m = O(n)$), and this yields the desired bound on $\|E_1\|$.

The second term E_2 is a sum of positive semidefinite matrices and we will make crucial use of this fact below:

$$E_2 = \sum_{i \neq j} q_{i,j} \beta_i \beta_i' A_j^* A_j^{*T} \preceq O(k^2/m^2)\Big(\sum_i \beta_i \beta_i'\Big)\Big(\sum_j A_j^* A_j^{*T}\Big) \preceq O(k^2/m^2)\|\beta\|\|\beta'\|A^* A^{*T}.$$

Here the first inequality follows by using bounds on $q_{i,j}$ and then completing the square. The second inequality uses Cauchy-Schwartz. We can now invoke the second part of Claim 36 and conclude that $\|E_2\| \leq O(k^2/m^2)\|\beta\|\|\beta'\|\|A^*\|^2 \leq O^*(k/m \log m)$ (where we have used the assumption that $m = O(n)$).

For the third error term E_3, by symmetry we need only consider terms of the form $q_{i,j}\beta_i \beta_j' A_i^* A_j^{*T}$. We can collect these terms and write them as $A^* Q A^{*T}$, where $Q_{i,j} = 0$ if $i = j$ and $Q_{i,j} = q_{i,j}\beta_i \beta_j'$ if $i \neq j$. First, we bound the Frobenius norm of Q:

$$\|Q\|_F = \sqrt{\sum_{i \neq j, i, j \in [m]} q_{i,j}^2 \beta_i^2 (\beta_j')^2} \leq \sqrt{O(k^4/m^4)(\sum_{i \in [m]} \beta_i^2)(\sum_{j \in [m]} (\beta_j')^2)} \leq O(k^2/m^2)\|\beta\|\|\beta'\|.$$

Finally $\|E_3\| \leq 2\|A^*\|^2\|Q\| \leq O(m/n \cdot k^2/m^2)\|\beta\|\|\beta'\| \leq O^*(k/m \log m)$, and this completes the proof of the claim. ■

The proof of the main lemma is now complete. ■

Appendix G. Sample Complexity

In the previous sections, we analyzed various update rules assuming that the algorithm was given the exact expectation of some matrix-valued random variable. Here we show that these algorithms can just as well use approximations to the expectation (computed by taking a small number of samples). We will focus on analyzing the sample complexity of Algorithm 2, but a similar analysis extends to the other update rules as well.

G.1. Generalizing the $(\alpha, \beta, \epsilon)$-correlated Condition

We first give a generalization of the framework we presented in Section 2 that handles random update direction g^s.

Definition 38 *A random vector g^s is* $(\alpha, \beta, \epsilon_s)$-correlated-whp *with a desired solution z^* if with probability at least $1 - n^{-\omega(1)}$,*

$$\langle g^s, z^s - z^* \rangle \geq \alpha \|z^s - z^*\|^2 + \beta \|g^s\|^2 - \epsilon_s.$$

This is a strong condition as it requires the random vector is well-correlated with the desired solution with very high probability. In some cases we can further relax the definition as the following:

Definition 39 *A random vector g^s is* $(\alpha, \beta, \epsilon_s)$-correlated-in-expectation *with a desired solution z^* if*

$$\mathbf{E}[\langle g^s, z^s - z^* \rangle] \geq \alpha \|z^s - z^*\|^2 + \beta \, \mathbf{E}[\|g^s\|^2] - \epsilon_s.$$

We remark that $\mathbf{E}[\|g^s\|^2]$ can be much larger than $\|\mathbf{E}[g^s]\|^2$, and so the above notion is still stronger than requiring (say) that the expected vector $\mathbf{E}[g^s]$ is $(\alpha, \beta, \epsilon_s)$-correlated with z^*.

Theorem 40 *Suppose random vector g^s is $(\alpha, \beta, \epsilon_s)$-correlated-whp with z^* for $s = 1, 2, \ldots, T$ where $T \leq \text{poly}(n)$, and η satisfies $0 < \eta \leq 2\beta$, then for any $s = 1, \ldots, T$,*

$$\mathbf{E}[\|z^{s+1} - z^*\|^2] \leq (1 - 2\alpha\eta)\|z^s - z^*\|^2 + 2\eta\epsilon_s$$

In particular, if $\|z^0 - z^*\| \leq \delta_0$ and $\epsilon_s \leq \alpha \cdot o((1 - 2\alpha\eta)^s)\delta_0^2 + \epsilon$, then the updates converge to z^* geometrically with systematic error ϵ/α in the sense that

$$\mathbf{E}[\|z^s - z^*\|^2] \leq (1 - 2\alpha\eta)^s \delta_0^2 + \epsilon/\alpha.$$

The proof is identical to that of Theorem 6 except that we take the expectation of both sides.

G.2. Proof of Theorem 8

In order to prove Theorem 8, we proceed in two steps. First we show when A^s is $(\delta_s, 2)$-near to A^*, the approximate gradient is $(\alpha, \beta, \epsilon_s)$-correlated-whp with optimal solution A^*, with $\epsilon_s \leq O(k^2/mn) + \alpha \cdot o(\delta_s^2)$. This allows us to use Theorem 40 as long as we can guarantee the spectral norm of $A^s - A^*$ is small. Next we show a version of Lemma 24 which works even with the random approximate gradient, hence the nearness property is preserved during the iterations. These two steps are formalized in the following two lemmas, and we defer the proofs until the end of the section.

Lemma 41 *Suppose A^s is $(2\delta, 2)$-near to A^* and $\eta \leq \min_i(p_i q_i(1 - \delta)) = O(m/k)$, then $\widehat{g}_i^s$ as defined in Algorithm 2 is $(\alpha, \beta, \epsilon_s)$-correlated-whp with A_i^* with $\alpha = \Omega(k/m)$, $\beta = \Omega(m/k)$ and $\epsilon_s \leq \alpha \cdot o(\delta_s^2) + O(k^2/mn)$.*

Lemma 42 *Suppose A^s is $(\delta_s, 2)$-near to A^* with $\delta_s = O^*(1/\log n)$, and number of samples used in step s is $p = \widetilde{\Omega}(mk)$, then with high probability A^{s+1} satisfies $\|A^{s+1} - A^*\| \leq 2\|A^*\|$.*

We will prove these lemmas by bounding the difference between $\widehat{g}_i^s$ and g_i^s using various concentration inequalities. For example, we will use the fact that $\widehat{g}_i^s$ is close to g_i^s in Euclidean distance.

Lemma 43 *Suppose A_i^s is $(\delta_s, 2)$-near to A^* with $\delta_s = O^*(1/\log n)$, and number of samples used in step s is $p = \widetilde{\Omega}(mk)$, then with high probability $\|\widehat{g}_i^s - g_i^s\| \leq O(k/m) \cdot (o(\delta_s) + O(\sqrt{k/n}))$.*

Using the above lemma, Lemma 41 now follows the fact that g_i^s is correlated with A_i^*. The proof of Lemma 42 mainly involves using matrix Bernstein's inequality to bound the fluctuation of the spectral norm of A^{s+1}.

Proof:[Proof of Theorem 8] The theorem now follows immediately by combining Lemma 41 and Lemma 42, and then applying Theorem 40. ■

G.3. Sample Complexity for Algorithm 3

For the initialization procedure, when computing the reweighted covariance matrix $M_{u,v}$ we can only take the empirical average over samples. Here we show with only $\widetilde{\Omega}(mk)$ samples, the difference between the true $M_{u,v}$ matrix and the estimated $M_{u,v}$ matrix is already small enough.

Lemma 44 *In Algorithm 3, if $p = \widetilde{\Omega}(mk)$ then with high probability for any pair u, v consider by Algorithm 3, we have $\|M_{u,v} - \widehat{M}_{u,v}\| \leq O^*(k/m \log n)$.*

The proof of this Lemma is deferred to Section G.4.3. notice that although in Algorithm 3, we need to estimate $M_{u,v}$ for many pairs u and v, the samples used for different pairs do not need to be independent. Therefore we can partition the data into two parts, use the first part to sample pairs u, v, and use the second part to estimate $M_{u,v}$. In this way, we know that for each pair u, v the whole initialization algorithm also takes $\widetilde{\Omega}(mk)$ samples. Now we are ready to prove Theorem 13.

Proof:[Proof of Theorem 13] First of all, the conclusion of Lemma 34 is still true for $\widehat{M}_{u,v}$ when $p = \widetilde{\Omega}(mk)$. To see this, we could simply write

$$\widehat{M}_{u,v} = q_i c_i \beta_i \beta_i' A_i^* A_i^{*T} + \underbrace{(E_1 + E_2 + E_3) + (\widehat{M}_{u,v} - M_{u,v})}_{\text{perturbation}}$$

where E_1, E_2, E_3 are the same as the proof of Lemma 34. We can now view $\widehat{M}_{u,v} - M_{u,v}$ as an additional perturbation term with the same magnitude. We have that when $U \cap V = \{i\}$ the top singular vector of $M_{u,v}$ is $O^*(1/\log n)$-close to A_i^*. Similarly, we can prove the conclusion of Lemma 35 is also true for $\widehat{M}_{u,v}$. Note that we actually choose p such that the perturbation of $\widehat{M}_{u,v}$ matches noise level in Lemma 35. Finally, the proof of the theorem follows exactly that of the infinite sample case given in Theorem 33, except that we invoke the finite sample counterparts of Lemma 14 and Lemma 35 that we gave above. ■

G.4. Proofs of Auxiliary Lemmas

Here we prove Lemma 41, Lemma 42, and Lemma 44 which will follow from various versions of the Bernstein inequality. We first recall Bernstein's inequality that we are going to use several times in this section. Let Z be a random variable (which could be a vector or a matrix) chosen from some distribution $\mathcal{D}$ and let $Z^{(1)}, Z^{(2)}, ..., Z^{(p)}$ be p independent and identically distributed samples from $\mathcal{D}$. Bernstein's inequality implies that if $\mathbf{E}[Z] = 0$ and for each j, $\|Z^{(j)}\| \leq R$ almost surely and $\mathbf{E}[(Z^{(j)})^2] \leq \sigma^2$, then

$$\frac{1}{p}\left\|\sum_{i=1}^{p} Z^{(i)}\right\| \leq \widetilde{O}\left(\frac{R}{p} + \sqrt{\frac{\sigma^2}{p}}\right) \tag{5}$$

with high probability. The proofs below will involve computing good bounds on R and σ^2. However in our setting, the random variables will not be bounded almost surely. We will use the following technical lemma to handle this issue.

Lemma 45 *Suppose that the distribution of Z satisfies $\mathbf{Pr}[\|Z\| \geq R(\log(1/\rho))^C] \leq 1 - \rho$ for some constant $C > 0$, then*

(a) If $p = n^{O(1)}$ then $\|Z^{(j)}\| \leq \widetilde{O}(R)$ holds for each j with high probability and

(b) $\|\mathbf{E}[Z\mathbf{1}_{\|Z\| \geq \widetilde{\Omega}(R)}]\| = n^{-\omega(1)}$.

In particular, if $\frac{1}{p}\sum_{j=1}^{p} Z^{(j)}(1 - \mathbf{1}_{\|Z^{(j)}\| \geq \widetilde{\Omega}(R)})$ is concentrated with high probability, then $\frac{1}{p}\sum_{j=1}^{p} Z^{(j)}$ is too.

Proof: The first part of the lemma follows from choosing $\rho = n^{-\log n}$ and applying a union bound. The second part of the lemma follows from

$$\begin{aligned}\mathbf{E}[Z\mathbf{1}_{\|Z\| \geq R\log^{2c} n}] &\leq \mathbf{E}[\|Z\|\mathbf{1}_{\|Z\| \geq R\log^{2c} n}] \\ &= R\log^{2c} n\,\mathbf{Pr}[\|Z\| \geq R\log^{2c} n] + \int_{R\log^{2c} n}^{\infty} \mathbf{Pr}[\|Z\| \geq t]dt = n^{-\omega(1)}.\end{aligned}$$

and this completes the proof. ■

All of the random variables we consider are themselves products of subgaussian random variables, so they satisfy the tail bounds in the above lemma. In the remaining proofs we will focus on bounding the norm of these variables with high probability.

G.4.1. Proof of Lemma 43 and Lemma 41

Since s is fixed throughout, we will use A to denote A^s. Also we fix i in this proof. Let S denote the support of x^*. Note that $\widehat{g}_i$ is a sum of random variable of the form $(y - Ax)\text{sgn}(x_i)$. Therefore we are going to apply Bernstein inequality for proving $\widehat{g}_i$ concentrates around its mean g_i. Since Bernstein is typically not tight for sparse random varaibles like in our case. We study the concentration of the random variable $Z := (y - Ax)\text{sgn}(x_i) \mid i \in S$ first. We prove the following technical lemma at the end of this section.

Claim 46 *Let $Z^{(1)}, \ldots, Z^{(\ell)}$ be i.i.d random variables with the same distribution as $Z := (y - Ax)sgn(x_i) \mid i \in S$. Then when $\ell = \widetilde{\Omega}(k^2)$,*

$$\|\frac{1}{\ell}\sum_{j=1}^{\ell} Z^{(j)} - \mathbf{E}[Z]\| \leq o(\delta_s) + O(\sqrt{k/n})$$

We begin by proving Lemma 43.

Proof:[Proof of Lemma 43] Let $W = \{j : i \in \text{supp}(x^{*(j)})\}$ and then we have that

$$\widehat{g}_i = \frac{|W|}{p} \cdot \frac{1}{|W|}\sum_j (y^{(j)} - Ax^{(j)})\text{sgn}(x_i^{(j)})$$

Note that $\frac{1}{|W|}\sum_j (y^{(j)} - Ax^{(j)})\text{sgn}(x_i^{(j)})$ has the same distribution as $\frac{1}{\ell}\sum_{j=1}^{\ell} Z^{(j)}$ for $\ell = |W|$, and indeed by concentration we have $\ell = |W| = \widetilde{\Omega}(k^2)$ when $p = \widetilde{\Omega}(mk)$. Also note that $\mathbf{E}[(y - Ax)\text{sgn}(x_i)] = q_i \cdot \mathbf{E}[Z]$ with $q_i = O(k/m)$. Therefore by Lemma 46 we have that

$$\|\widehat{g}_i - g_i\| \leq O(k/m) \cdot \|\frac{1}{\ell}\sum_{j=1}^{\ell} Z^{(j)} - \mathbf{E}[Z]\| \leq O(k/m) \cdot (o(\delta_s) + O(\sqrt{k/n}))$$

and this completes the proof. ■

Proof:[Proof of Lemma 41] Therefore using Lemma 10 we can write $\widehat{g}_i^s$ (whp) as $\widehat{g}_i = \widehat{g}_i - g_i + g_i = 4\alpha(A_i^s - A_i^*) + v$ with $\|v\| \leq \alpha\|A_i^s - A_i^*\| + O(k/m) \cdot (o(\delta_s) + O(\sqrt{k/n}))$. By Lemma 22 we have $\widehat{g}_i$ is $(\Omega(k/m), \Omega(m/k), o(k/m \cdot \delta_s^2) + O(k^2/mn))$-correlated-whp with A_i^*. ■

Then it suffices to prove Claim 46. To this end, we apply the Bernstein's inequality stated in equation 5 with the additional technical lemma 45. We are going to control the maximum norm of Z and as well as the variance of Z using Claim 47 and Claim 48 as follows:

Claim 47 $\|Z\| = \|(y - Ax)sgn(x_i)\| \leq \widetilde{O}(\mu k/\sqrt{n} + k\delta_s^2 + \sqrt{k}\delta_s)$ *holds with high probability*

Proof: We write $y - Ax = (A_S^* - A_S A_S^T A_S^*)x_S^* = (A_S^* - A_S)x_S^* + A_S(I - A_S^T A_S^*)x_S^*$ and we will bound each term. For the first term, since A is δ_s-close to A^* and $|S| \leq O(k)$, we have that $\|A_S^* - A_S\|_F \leq O(\delta_s\sqrt{k})$. And for the second term, we have

$$\begin{aligned}
\|A_S(A_S^T A_S^* - I)\|_F &\leq \|A_S\|\|(A_S^T A_S^* - I)\|_F \\
&\leq (\|A_S^*\| + \delta_s\sqrt{k})(\|(A_S - A_S^*)^T A_S^*\|_F + \|{A_S^*}^T A_S^* - I\|_F) \\
&\leq (2 + \delta_s\sqrt{k})(\|A_S^*\|\|A_S - A_S^*\|_F + \mu k/\sqrt{n}) \leq O(\mu k/\sqrt{n} + \delta_s^2 k + \sqrt{k}\delta_s).
\end{aligned}$$

Here we have repeatedly used the bound $\|UV\|_F \leq \|U\|\|V\|_F$ and the fact that A^* is μ incoherent which implies $\|A_S^*\| \leq 2$. Recall that the entries in x_S^* are chosen independently

of S and are subgaussian. Hence if M is fixed then $\|Mx_S^*\| \leq \widetilde{O}(\|M\|_F)$ holds with high probability. And so

$$\|(y - Ax)\mathrm{sgn}(x_i)\| \leq \widetilde{O}(\|A_S^* - A_S\|_F + \|A_S(A_S^T A_S^* - I)\|_F) \leq \widetilde{O}(\mu k/\sqrt{n} + k\delta_s^2 + \sqrt{k}\delta_s)$$

which holds with high probability and this completes the proof. ■

Next we bound the variance.

Claim 48 $\mathbf{E}[\|Z\|^2] = \mathbf{E}[\|(y - Ax)sgn(x_i)\|^2 | i \in S] \leq O(k^2\delta_s^2) + O(k^3/n)$

Proof: We can again use the fact that $y - Ax = (A_S^* - A_S A_S^T A_S^*)x_S^*$ and that x_S^* is conditionally independent of S with $\mathbf{E}[x_S^*(x_S^*)^T] = I$ and conclude

$$\mathbf{E}[\|(y - Ax)\mathrm{sgn}(x_i)\|^2 | i \in S] = \mathbf{E}[\|A_S^* - A_S A_S^T A_S^*\|_F^2 \mid i \in S]$$

Then again we write $A_S^* - A_S A_S^T A_S^*$ as $(A_S^* - A_S) + A_S(I_{k\times k} - A_S^T A_S^*)$, and the bound the Frobenius norm of the two terms separately. First, since A is δ_s-close to A^*, we have that $A_S^* - A_S$ has column-wise norm at most δ_s and therefore $\|A_S^* - A_S\|_F \leq \sqrt{k}\delta_s$. Second, note that $\|A_S\|_F \leq O(\sqrt{k})$ since each column of A has norm $1 \pm \delta_s$, we have that

$$\begin{aligned}
\mathbf{E}[\|A_S(I - A_S^T A_S^*)\|_F^2 \mid i \in S] &\leq O(k)\,\mathbf{E}[\|(I_{k\times k} - A_S^T A_S^*)\|_F^2 \mid i \in S] \\
&\leq O(k)\,\mathbf{E}\left[\sum_{j\in S}(1 - A_j^T A_j^*)^2 + \sum_{j\neq \ell \in S}\langle A_j, A_\ell^*\rangle^2 \mid i \in S\right]
\end{aligned}$$

We can now use the fact that A is δ_s-close to A^*, expand out the expectation, and use the fact that $\mathbf{Pr}[j \in S, \ell \in S \mid i \in S] \leq O(k^2/m^2)$, to obtain

$$\begin{aligned}
&\mathbf{E}[\|A_S(I - A_S^T A_S^*)\|_F^2 \mid i \in S] \\
\leq\ & O(k^2\delta_s^2) + O(k^3/m^2)\cdot \sum_{j,\ell\in[m]\setminus i}\langle A_j, A_\ell^*\rangle^2 + O(k^2/m)\|A_i^T A_{-i}^*\|^2 + O(k^2/m)\|A_{-i}^T A_i^*\|^2 \\
\leq\ & O(k^2\delta_s^2) + O(k^3/n)
\end{aligned}$$

and this completes the proof. ■

Proof:[Proof of Claim 46] We apply first Bernstein's inequality (5) with $R = \widetilde{O}(\mu k/\sqrt{n} + k\delta_s^2 + \sqrt{k}\delta_s)$ and $\sigma^2 = O(k^2\delta_s^2) + O(k^3/n)$ on random variable $Z^{(j)}(1 - \mathbf{1}_{\|Z^{(j)}\|\geq\Omega(R)})$. Then by claim 47, claim 48 and Bernstein Inequality, we know that the truncated version of Z concentrates when $\ell = \Omega(k^2)$,

$$\left\|\frac{1}{\ell}\sum_{j=1}^{\ell} Z^{(j)}(1 - \mathbf{1}_{\|Z^{(j)}\|\geq\Omega(R)}) - \mathbf{E}[Z(1 - \mathbf{1}_{\|Z\|\geq\Omega(R)})]\right\| \leq \widetilde{O}\left(\frac{R}{\ell}\right) + \widetilde{O}\left(\sqrt{\frac{\sigma^2}{\ell}}\right) = o(\delta_s) + O(\sqrt{k/n})$$

Note that we choose $\ell = k^2\log^c n$ for a large constant c so that it kills the log factors caused by Bernstein's inequality. Then by Lemma 45, we have that $\sum_j Z^{(j)}$ also concentrates:

$$\|\frac{1}{\ell}\sum_{j=1}^{\ell} Z^{(j)} - \mathbf{E}[Z]\| \leq o(\delta_s) + O(\sqrt{k/n})$$

and this completes the proof. ■

G.4.2. Proof of Lemma 42

Proof:[Proof of Lemma 42] We will apply the matrix Bernstein inequality. In order to do this, we need to establish bounds on the spectral norm and on the variance. For the spectral norm bound, we have $\|(y - A^s x)\mathrm{sgn}(x)^T\| = \|(y - A^s x)\|\|\mathrm{sgn}(x)\| = \sqrt{k}\|(y - A^s x)\|$. We can now use Claim 47 to conclude that $\|(y - A^s x)\| \leq \widetilde{O}(k)$, and hence $\|(y - A^s x)\mathrm{sgn}(x)\| \leq \widetilde{O}(k^{3/2})$ holds with high probability.

For the variance, we need to bound both $\mathbf{E}[(y - A^s x)\mathrm{sgn}(x)^T\mathrm{sgn}(x)(y - A^s x)^T]$ and $\mathbf{E}[\mathrm{sgn}(x)(y - A^s x)^T(y - A^s x)\mathrm{sgn}(x)^T]$. The first term is equal to $k\,\mathbf{E}[(y - A^s x)(y - A^s x)^T]$. Again, the bound follows from the calculation in Lemma 26 and we conclude that

$$\|\,\mathbf{E}[(y - A^s x)\mathrm{sgn}(x)^T\mathrm{sgn}(x)(y - A^s x)^T]\| \leq O(k^2/n)$$

To bound the second term we note that

$$\mathbf{E}[\mathrm{sgn}(x)(y - A^s x)^T(y - A^s x)\mathrm{sgn}(x)^T] \preceq \widetilde{O}(k^2)\,\mathbf{E}[\mathrm{sgn}(x)\mathrm{sgn}(x)^T] \preceq \widetilde{O}(k^3/m)I$$

Moreover we can now apply the matrix Bernstein inequality and conclude that when the number of samples is at least $\widetilde{\Omega}(mk)$ we have

$$\|\frac{1}{p}\sum_{j=1}^{p}(y^{(j)} - A^s x^{*(j)})\mathrm{sgn}(x^{*(j)})^T - \mathbf{E}[(y - A^s x)\mathrm{sgn}(x)^T\| \leq O^*(k/m \cdot \sqrt{m/n})$$

and this completes the proof. ■

G.4.3. Proof of Lemma 44

Again in order to apply the matrix Bernstein inequality we need to bound the spectral norm and the variance of each term of the form $\langle u, y\rangle\langle v, y\rangle yy^T$. We make use of the following claim to bound the magnitude of the inner product:

Claim 49 *$|\langle u, y\rangle| \leq \widetilde{O}(\sqrt{k})$ and $\|y\| \leq \widetilde{O}(\sqrt{k})$ hold with high probability*

Proof: Since $u = A^*\alpha$ and because α is k-sparse and has subgaussian non-zero entries we have that $\|u\| \leq \widetilde{O}(\sqrt{k})$, and the same bound holds for y too. Next we write $|\langle u, y\rangle| = |\langle A^{*T}_S u, x^*_S\rangle|$ where S is the support of x^*. Moreover for any set S, we have that

$$\|A^{*T}_S u\| \leq \|A^*_S\|\|u\| \leq \widetilde{O}(\sqrt{k})$$

holds with high probability, again because the entries of x^*_S are subgaussian we conclude that $|\langle u, y\rangle| \leq \widetilde{O}(\sqrt{k})$ with high probability. ■

This implies that $\|\langle u, y\rangle\langle v, y\rangle yy^T\| \leq \widetilde{O}(k^2)$ with high probability.

Now we need to bound the variance:

Claim 50 $\|\,\mathbf{E}[\langle u, y\rangle^2\langle v, y\rangle^2 yy^T yy^T]\| \leq \widetilde{O}(k^3/m)$

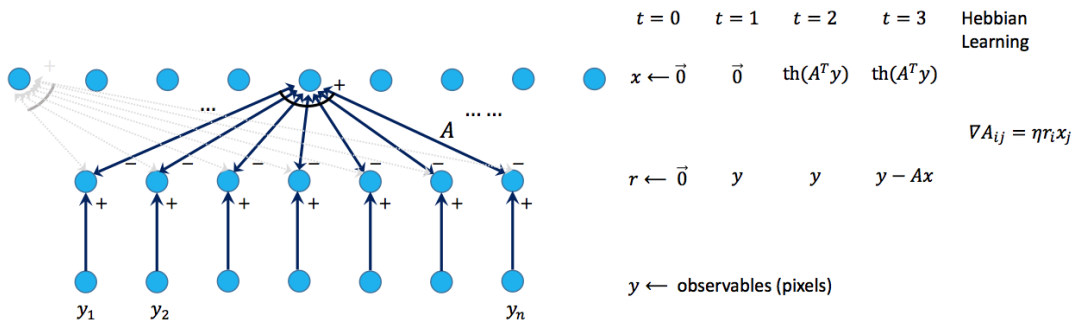

Figure 1: A neural implementation of Algorithm 2, which mimics that of Olshausen-Field (Figure 5 in Olshausen and Field (1997a))

Proof: We have that with high probability $\|y\|^2 \leq \widetilde{O}(k)$ and $\langle u, y\rangle^2 \leq \widetilde{O}(k)$, and we can apply these bounds to obtain

$$\mathbf{E}[\langle u, y\rangle^2 \langle v, y\rangle^2 yy^T yy^T] \preceq \widetilde{O}(k^2)\, \mathbf{E}[\langle v, y\rangle^2 yy^T]$$

On the other hand, notice that $\mathbf{E}[\langle v, y\rangle^2 yy^T] = M_{v,v}$ and using Lemma 14 we have that $\|\mathbf{E}[\langle v, y\rangle^2 yy^T]\| \leq O(k/m)$. Hence we conclude that the variance term is bounded by $\widetilde{O}(k^3/m)$. ■

Now we can apply the matrix Bernstein inequality and conclude that when the number of samples is $p = \widetilde{\Omega}(mk)$ then

$$\|\widehat{M}_{u,v} - M_{u,v}\| \leq \widetilde{O}(k^2)/p + \sqrt{\widetilde{O}(k^3/mp)} \leq O^*(k/m \log n)$$

with high probability, and this completes the proof.

Appendix H. Neural Implementation

Neural Implementation of Alternating Minimization: Here we sketch a neural architecture implementing Algorithm 2, essentially mimicking Olshausen-Field (Figure 5 in Olshausen and Field (1997a)), except our decoding rule is much simpler and takes a single neuronal step.

(a) The bottom layer of neurons take input y and at the top layer neurons output the decoding x of y with respect to the current code. The middle layer labeled r is used for intermediate computation. The code A is stored as the weights between the top and middle layer on the synapses. Moreover these weights are set via a Hebbian rule, and upon receiving a new sample y, we update the weight A_{ij} on the synapses by the product of the values of the two endpoint neurons, x_j and r_i.

(b) The top layer neurons are equipped with a threshold function. The middle layer ones are equipped with simple linear functions (no threshold). The bottom layer can only be changed by stimulation from outside the system.

(c) We remark that the updates require some attention to timing, which can be accomplished via spike timing. In particular, when a new image is presented, the value of all neurons are updated to a (nonlinear) function of the weighted sum of the values of its neighbors with weights on the corresponding synapses. The execution of the network is shown at the right hand side of the figure. Upon receiving a new sample at time $t = 0$, the values of bottom layer are set to be y and all the other layers are reset to zero. At time $t = 1$, the values in the middle layer are updated by the weighted sum of their neighbors, which is just y. Then at time $t = 2$, the top layer obtains the decoding x of y by calculating $\text{threshold}_{C/2}(A^T y)$. At time $t = 3$ the middle layer calculates the residual error $y - Ax$ and then at time $t = 4$ the synapse weights that store A are updated via Hebbian rule (update proportional to the product of the endpoint values). Repeating this process with many images indeed implements Algorithm 2 and succeeds provided that the network is appropriately initialized to A^0 that is $(\delta, 2)$ near to A^*.

Neural Implementation of Initialization: Here we sketch a neural implementation of Algorithm 13. This algorithm uses simple operations that can be implemented in neurons and composed, however unlike the above implementation we do not know of a two layer network, but note that this procedure need only be performed once so it need not have a particularly fast or short neural implementation.

(a) It is standard to compute the inner products $\langle y, u\rangle$ and $\langle y, v\rangle$ using neurons, and even the top singular vector can be computed using the classic Oja's Rule **?** in an online manner where each sample y is received sequentially. There are also generalizations to computing other principle components **?**. However, we only need the top singular vector and the top two singular values.

(b) Also, the greedy clustering algorithm which preserves a single estimate z in each equivalence class of vectors that are $O^*(1/\log m)$-close (after sign flips) can be implemented using inner products. Finally, projecting the estimate $\widetilde{A}$ onto the set $\mathcal{B}$ may not be required in real life (or even for correctness proof), but even if it is it can be accomplished via stochastic gradient descent where the gradient again makes use of the top singular vector of the matrix.