On the Computational Intractability of Exact and Approximate Dictionary Learning

Andreas M. Tillmann

Abstract—The efficient sparse coding and reconstruction of signal vectors via linear observations has received a tremendous amount of attention over the last decade. In this context, the automated learning of a suitable basis or overcomplete dictionary from training data sets of certain signal classes for use in sparse representations has turned out to be of particular importance regarding practical signal processing applications. Most popular dictionary learning algorithms involve NP-hard sparse recovery problems in each iteration, which may give some indication about the complexity of dictionary learning but does not constitute an actual proof of computational intractability. In this technical note, we show that learning a dictionary with which a given set of training signals can be represented as sparsely as possible is indeed NP-hard. Moreover, we also establish hardness of approximating the solution to within large factors of the optimal sparsity level. Furthermore, we give NP-hardness and non-approximability results for a recent dictionary learning variation called the sensor permutation problem. Along the way, we also obtain a new non-approximability result for the classical sparse recovery problem from compressed sensing.

Index Terms—(SAS-MALN, MLSAS-SPARSE) Machine Learning, Compressed Sensing, Computational Complexity

I. INTRODUCTION

As a central problem in compressed sensing (CS) \cite{1, 2}, the task of finding a sparsest exact or approximate solution to an underdetermined system of linear equations has been a strong focus of research during the past decade. Denoting by $\|x\|_0$ the so-called $\ell_0$-norm, i.e., the number of nonzero entries in $x$, the sparse recovery problem reads

$$\min \|x\|_0 \quad \text{s.t.} \quad \|Dx - y\|_2 \leq \delta, \quad (P_0^\delta)$$

for a given matrix $D \in \mathbb{R}^{m \times n}$ with $m \leq n$ and an estimate $\delta \geq 0$ of the amount of error contained in the measurements $y \in \mathbb{R}^m$. Both the noisefree problem $(P_0) := (P_0^0)$ and the error-tolerant variant $(P_\delta^\delta)$ with $\delta > 0$ are well-known to be NP-hard in the strong sense, cf. \cite{4} problem MP5 and \cite{5}, and also difficult to approximate \cite{6, 7}.

Groundbreaking results from CS theory include qualification conditions (on the dictionary $D$ and the solution sparsity level) which yield efficient solvability of the generally hard problems $(P_0^\delta)$ by greedy methods—e.g., orthogonal matching pursuit (OMP) \cite{8}—or (convex) relaxations such as Basis Pursuit \cite{9} (replacing the $\ell_0$-norm by the $\ell_1$-norm); see \cite{2, 3, 10} for overviews. Subsequently, numerous optimization algorithms have been tailored to sparse recovery tasks, and various types of dictionaries were shown or designed to exhibit favorable recoverability properties. In particular, the essential assumption of (exact or approximate) sparse representability of certain signal classes using specific dictionaries has been empirically verified in many practical signal processing applications; for instance, natural images are known to admit sparse approximations over discrete cosine or wavelet bases \cite{2}.

Nevertheless, a predetermined setup typically cannot fully capture the true structure of real-world signals; thus, using a fixed dictionary $D$ naturally restricts the achievable sparsity levels of the representations. Indeed, the simultaneous search for both dictionary and sparse representations of a set of training signals—commonly referred to as dictionary learning—was demonstrated to allow for significantly improved sparsity levels using the learned dictionary instead of an analytical, structured or random one. Successful applications of dictionary learning include diverse tasks such as image inpainting \cite{11}, \cite{12}, denoising \cite{13, 14} and deblurring \cite{15}, or audio and speech signal representation \cite{16, 17}, to name but a few.

Somewhat informally, the dictionary learning (DL) problem can be stated as: Given a collection of training data vectors $y^1, \ldots, y^p \in \mathbb{R}^m$ and a positive integer $n$, find a matrix $D \in \mathbb{R}^{m \times n}$ that allows for the sparsest possible representations $x^j$ such that $Dx^j = y^j$ (for all $j$). This task can be formalized in different ways, and there exist many variants seeking dictionaries with further properties such as incoherence \cite{18} or union-of-bases \cite{19}; see also, e.g., \cite{20, 21, 12}. Moreover, several DL algorithms have been developed over the past years; the frequently encountered hard sparse recovery subproblems are typically treated by classical methods from CS. We refer to \cite{22, 23, 24, 25, 26, 12, 27, 28} (and references therein) for a broader overview of well-established DL techniques and some more recent results.

In this paper, we are concerned with the computational complexity of dictionary learning. Due to its combinatorial nature, it is widely believed to be a very challenging problem, but to the best of our knowledge, a formal proof of this intractability claim was missing. We contribute to the theoretical understanding of the problem by providing an NP-hardness proof as well as a strong non-approximability result for DL, see Section III. Furthermore, we prove NP-hardness and non-approximability of an interesting new DL variant—the sensor permutation problem, where the sought dictionary is constrained to be related to a given sensing matrix via unknown row permutations; see Section III for the details. As a byproduct, we also obtain a new NP-hardness of approximation result for the sparse recovery problem $(P_0^\delta)$.

Remark 1: Recall that NP-hardness implies that no polynomial-time solution algorithm can exist, under the...
most-widely believed theoretical complexity assumption that $P \neq NP$ [4]. Further, strong NP-hardness can be understood, in a nutshell, as an indication that a problem’s intractability does not depend on ill-conditioning of the input coefficients. This additionally implies that (unless $P=NP$) there cannot exist a pseudo-polynomial-time exact algorithm and not even a fully polynomial-time approximation scheme (FPTAS), i.e., an algorithm that solves a minimization problem within a factor of $(1+\varepsilon)$ of the optimal value in polynomial time with respect to the input size and $1/\varepsilon$, see [4]. For a thorough and detailed treatment of complexity theory, we refer to [4], [29].

### II. The Complexity of Dictionary Learning

As mentioned in the introduction, different philosophies or goals lead to different formulations of dictionary learning problems, which are usually captured by the general form

$$\min_{D,X} f(D, X; Y) + g(D) + h(X),$$

(1)

where the variables are the dictionary $D \in \mathbb{R}^{m \times n}$ (for an a priori chosen $n$) and the matrix $X \in \mathbb{R}^{n \times p}$, whose columns are the representation vectors $x^j$ of the given training signals $y^j$ (w.r.t. the linear model assumption $Dx^j \approx y^j$), collected in $Y \in \mathbb{R}^{m \times p}$ as its columns; the functions $f$, $g$, and $h$ express a data fidelity term, and constraints or penalties/regularizers for the dictionary and the representation coefficient vectors, resp.

In the (ideal) noiseless case, the usual approach (see, e.g., [11], [21], [25]) is

$$\min_{D,X} \|X\|_0 \quad \text{s.t.} \quad DX = Y,$$

(2)

which fits the framework (1) by setting $f(D, X; Y) := \chi(DX = Y) \quad (D, X)$ (where $\chi$ is the indicator function, i.e., $f(D, X; Y) = 0$ if $DX = Y$ and $\infty$ otherwise), $g(D) := 0$ and $h(X) := \|X\|_0$ (extending the usual notation to matrices, $\|X\|_0$ counts the nonzero entries in $X$). This problem is a natural extension of (P0), and can also be seen as a matrix-factorization problem. To mitigate scaling ambiguities, one often sets $g(D) := \chi(\|D\|_2 \leq 1) \vee j = 1, \ldots, n)$, i.e., the columns of $D$ are required to have bounded norms; cf. [26], [15].

Note that if $n$ is not fixed a priori to a value smaller than $p$, the dictionary learning task becomes trivial: Then, we could just take $D = [y^1, \ldots, y^p]$ and exactly represent every $y^j$ using only one column. (Clearly, this also holds for variants which allow representation errors, e.g., $\|Dx^j - y^j\|_2 \leq \delta$ for some $\delta > 0$, or minimize such errors under hard sparsity limits $\|x^j\|_0 \leq k$ for some $k \geq 1$.) Thus, requiring $n < p$ is hardly restrictive, in particular since the training data set (and hence, $p$) is usually very large—intuitively, the more samples of a certain signal class are available for learning the dictionary, the better the outcome will be adapted to that signal class—and with respect to storage aspects and efficient (algorithmic) applicability of the learned dictionary, settling for a smaller number of dictionary atoms is well-justified. Similarly, $m \leq n$ is a natural assumption, since sparsity of the coefficient vectors is achieved via appropriate representation bases or redundancy (overcompleteness) of the dictionary; also, at least for large $p$, one can expect $\text{rank}(Y) = m$, in which case $\text{rank}(D) = m \leq n$ becomes necessary to maintain $DX = Y$.

### A. NP-Hardness

As the following results show, finding a dictionary with which the training signals can be represented with optimal sparsity is indeed a computationally intractable problem.

**Theorem 2:** Solving the dictionary learning problem (2) is NP-hard in the strong sense, even when restricting $n = m$.

**Proof:** We reduce from the matrix sparsification (MS) problem: Given a full-rank matrix $M \in \mathbb{Q}^{m \times p}$ ($m < p$), find a regular matrix $B \in \mathbb{R}^{m \times m}$ such that $BM$ has as few nonzero entries as possible. (The full-rank assumption is not mandatory, but can be made w.l.o.g.: If $\text{rank}(M) = k < m$, $m - k$ rows can be zeroed in polynomial time by elementary row operations, reducing the problem to sparsifying the remaining row submatrix.) The MS problem was shown to be NP-hard in [30] Theorem 3.2.1 (see also [31], [10]), by a reduction from simple max cut, cf. [4] problem ND16; since this reduction constructs a binary matrix (of dimensions polynomially bounded by the cut problem’s input graph size), NP-hardness of MS in fact holds in the strong sense, and we may even assume w.l.o.g. that $M \in \{0,1\}^{m \times p}$.

From an instance of MS, we obtain an equivalent instance of (2) as follows: Set $n := m$ and let $Y := M$. Then, the task (2) is to find $D \in \mathbb{R}^{m \times m}$ and $X \in \mathbb{R}^{m \times p}$ such that $DX = Y$ and $\|X\|_0$ is minimal. (Note that the sought dictionary in fact constitutes a basis for $\mathbb{R}^m$, since $M$ has full row-rank $m$, thus requiring this of the dictionary as well, as discussed above.) Clearly, an optimal solution $(D_*, X_*)$ of this dictionary learning instance gives an optimal solution $B_* = D_*^{-1}$ of MS, with $B_*M = X_*$. It remains to note that the reduction is indeed polynomial, since the matrix inversion can be performed in strongly polynomial time by Gaussian elimination, cf. [32]. Thus, (2) is strongly NP-hard.

**Remark 3:** The above NP-hardness result easily extends to variants of (2) with the additional constraint that, for some constant $\epsilon > 0$, $\|D_j\|_2 \leq c$ for all $j$, or $\|D_j\|_2 = \text{tr}(D_j^T D_j) \leq c$ (as treated in [16]): Since the discrete objectives are invariant to scaling in both the dictionary learning and the MS problem, there is always also an optimal $D_*$ (achieving the same number of nonzeros in the corresponding $X_*$) that obeys the norm constraints and yields an associated optimal solution $B_* = (D_*)^{-1}$ of the MS problem. (Clearly, this argument remains valid for a host of similar norm constraints as well.)

It is not known whether the decision version of the MS problem is contained in NP (and thus not only NP-hard but NP-complete) [30]. Similarly, we do not know if the decision problem associated with (2)—“given $Y \in \mathbb{Q}^{m \times p}$ and positive integers $k$ and $n$, decide whether there exist $D \in \mathbb{R}^{m \times m}$ and $X \in \mathbb{R}^{n \times p}$ such that $DX = Y$ and $\|X\|_0 \leq k$”—is contained in NP, even in the square case $n = m$.

### B. Non-Approximability

Since for NP-hard problems, the existence of efficient (polynomial-time) general exact solution algorithms is deemed impossible, it is natural to search for good approximation methods. Indeed, virtually all well-known dictionary learning algorithms can be interpreted (in a vague sense) as “approximation schemes” since, e.g., the $\ell_0$-norm is convexified to
the $\ell_1$-norm, constraints may be turned to penalty terms in the objective (regularization), etc. However, even disregarding the computational costs of the algorithms, little is known about the quality of the obtained approximations; several recent works along these lines started investigating theoretical recovery properties and error guarantees of dictionary learning algorithms, see, e.g., \[33, 34, 28\]; in particular, \[34\] shows the importance of a good dictionary initialization.

The non-existence of an FPTAS (cf. Remark 1) itself does not generally rule out the existence of an efficient algorithm with some constant approximation guarantee. However, we show below that it is almost-NP-hard to approximate the dictionary learning problem \[2\] to within large factors of the optimal achievable sparsity of representations. Almost-NP-hardness means that no polynomial-time algorithm (here, to achieve the desired approximation ratio) can exist so long as \[\text{NP} \not\subseteq \text{DTIME}(N^{\text{poly}(\log N)})\], where \(N\) measures the input size (usually, dimension); cf. \[35\]. This complexity assumption is not completely equivalent to the MS problem, showing strong NP-hardness also \[\not\subseteq \text{DTIME}(N^{\text{poly}(\log m)})\]. Here, it is assumed that the dictionary \(D\) is known to have full rank excludes the possibility of as few sets from \(C\) to \(S\) as possible such that \(\bigcup_{C \in C'} C = S^n\). A cover \(C'\) is called exact if \(C \cap D = \emptyset\) for all \(C, D \in C'\) (in other words, if every element of \(S\) is contained in exactly one set from \(C'\)). We will employ the following very recent result:

**Proposition 8** (\[42 Theorem 2\]): For every \(0 < \alpha < 1\), there exists a polynomial-time reduction from an arbitrary instance of the strongly NP-complete satisfiability problem (SAT, cf. \[4 problem L01\]) to an SC instance \((S, C)\) with a parameter \(k \in \mathbb{N}\) such that if the input SAT instance is satisfiable, there is an exact cover of size \(k\) (and no smaller covers), whereas otherwise, every cover has size at least \((1 - \alpha)\ln(|S|)k\).

Recall also that, for any \(\varepsilon > 0\), approximating the sparse recovery problem \[P_0\] (with any \(\delta \geq 0\)) to within factors \(2^{\log^{1-\varepsilon} m}\) is almost-NP-hard, by \[2 Theorem 3\]. (In fact, although it clearly goes through for \(\delta = 0\) as well, \[7\] states the proof of this only for \(\delta > 0\), because the corresponding result for \(P_0\) had already been shown in \[6\] before.) The proof of \[7 Theorem 3\] is based on a special SC instance construction from \[43\] (see also \[35 Proposition 6\]) similar to that from Proposition 8.

**Remark 9:** In the special SC instances underlying the above results, it holds that \(|C|\) and \(|S|\) are polynomially related, so...
that all non-approximability results stated in this section also hold with \( m (= |S|) \) replaced by \( n (= |C|) \).

We are now ready to prove the main result of this section.

**Proof of Theorem 7** Let \((S, C, k, \alpha)\) be a Set Cover instance as in Proposition 8 and let \( n = |C|, m = |S| \). Following the proof of [Theorem 3], we first transform the task of finding a minimum-cardinality set cover to the sparse recovery problem \( (P_0) \): Define \( D \in \{0, 1\}^{m \times n} \) by setting \( D_{ij} = 1 \) if and only if the \( i \)-th element of \( S \) is contained in the \( j \)-th set from \( C \), and set \( y := 1 \), i.e., the all-ones vector of length \( n \). It is easily seen that the support of every solution \( x \) of \( Dx = y \) induces a set cover (if some element was not covered, at least one row of the equality system would evaluate to 0 = 1, contradicting \( Dx = y \). Conversely, every exact cover induces a solution of the same \( l_0 \)-norm as the cover size (put \( xc = 1 \) for the sets \( C \) contained in the exact cover, and zero in the remaining components). Thus, if there is an exact cover size of \( k \), there is a \( k \)-sparse solution of \( Dx = y \). Conversely, if all set covers have size at least \( \left(1 - \alpha\right) \ln(m) k \), then necessarily all \( x \) with \( Dx = y \) have \( \|x\|_0 \geq \left(1 - \alpha\right) \ln(m) k \) (because otherwise, the support of \( x \) would yield a set cover of size smaller than \( \left(1 - \alpha\right) \ln(m) k \).

This instance of \( (P_0) \) is now easily transformed into one of the sensor permutation problem [3]: We set \( A := D, Y := y \) (thus, \( p = 1 \)). Now, since \( Y = 1 \), \( PY = Y \) for all \( P \in \mathcal{P}_m \) and the choice of \( P \) has no influence on the solution. Thus, indeed, the SP problem [3] for these \( A \) and \( Y \) has precisely the same solution value as the above-constructed instance of \( (P_0) \). Since solving the original Set Cover instance is (strongly) \( NP \)-hard (by Proposition 8), and all constructed numbers and their encoding lengths remain polynomially bounded by the input parameter \( m \) (and \( n \)), this immediately shows the claimed strong \( NP \)-hardness result. In fact, could we approximate, in polynomial time, the optimal solution value of [3] to within a factor of \( \left(1 - \alpha\right) \ln(m) \), then we could also decide the SAT instance underlying the SC problem from Proposition 8 in polynomial time, which is impossible unless \( P = NP \). Therefore, for any \( 0 < \alpha < 1 \), even approximating [3] to within factors \( \left(1 - \alpha\right) \ln(m) \) is \( NP \)-hard.

For the second non-approximability result of Theorem 7 it suffices to note that the construction above is cost-preserving and that the \((P_0)\) instance in the proof of [Theorem 3] also has \( y = 1 \). Hence, we can directly transfer the non-approximability properties, and conclude that there is no polynomial-time algorithm approximating [3] to within factors \( 2^{\log^{1 - 1/n}} \) (for any \( \varepsilon > 0 \), unless \( NP \subseteq \text{DTIME}(n^{\text{polylog}(\log n)}) \).

Finally, the above results extend to the noise-aware SP problem variant by treating the relaxed constraints \( \|AX - PY\|_2 \leq \delta \) for \( \delta > 0 \) completely analogously to the proof of [Theorem 3] (we omit the details) and, by Remark 5 remain valid w.r.t. either \( m \) or \( n \).

**Remark 10:** The decision version of [3] is easily seen to be in \( NP \) (for rational input), and hence \( NP \)-complete.

Note that the first part of the above proof yields a new non-approximability result for sparse recovery:

**Corollary 11:** For any \( \alpha \in (0, 1) \), it is \( NP \)-hard to approximate \((P_0)\) to within a factor of \( \left(1 - \alpha\right) \ln(n) \).

This complements the previously known results from [Theorem 7] and [Theorem 3]: For \( n \) large enough (and some fixed pair \( \alpha, \varepsilon \)), \( 2^{\log^{1 - 1/n}} \) gets lower than \( 1 - \alpha \) and \( \ln(n) \), but the assumption \( P \neq NP \) is weaker than \( NP \subseteq \text{DTIME}(n^{\text{polylog}(\log n)}) \).

## IV. Concluding Remarks

In this note, we gave formal proofs for \( NP \)-hardness and non-approximability of several dictionary learning problems. While perhaps not very surprising, these results provide a complexity-theoretical justification for the common approaches to tackle dictionary learning tasks by inexact methods and heuristics without performance guarantees.

While preparing this manuscript, we became aware of a related result presented at ICASSP 2014, see [46]. In that work, the authors claim \( NP \)-hardness of approximating

\[
\min_{D, X} \|DX - Y\|_F^2 \quad \text{s.t.} \quad \|X_j\|_0 \leq k \; \forall j = 1, \ldots, p, \quad (4)
\]

to within a given additive error w.r.t. the objective (i.e., not within a factor of the optimal value), for the case in which \( Y \) contains only two columns and \( k \) is fixed to 1. Unfortunately, [46] does not contain a proof, and at the time of writing, we could not locate it elsewhere. Note also that, clearly, (4) is also a special case of the general formulation (1)—using \( f(D, X, Y) = \|DX - Y\|_F^2 \), \( h(X) = \chi(|X_j|_0 \leq k \; \forall j \} (X) \) and \( g(D) = 0 \)—but that the results from the present paper and from [46] nevertheless pertain to different problems, both of which are often referred to as “dictionary learning”.

Future research closely related to the present work could include investigating the potential use of matrix sparsification based heuristics for dictionary learning purposes (e.g., when learning a union-of-bases dictionary as in [19]).

Note also that the reduction from [40] does not admit transferring the \( NP \)-hardness of approximating MinULR to within any constant factor (see [35] Theorem 5) to the MS problem. (Similarly, the reduction to MS in [30] apparently does not preserve approximation ratios.) Such non-approximability results under the slightly weaker \( P \neq NP \) assumption hence remain open for problem [3] (and its norm-constrained variants).

Also, the complexities of dictionary learning with \( \ell_1 \)-objective and/or noise-awareness (e.g., constraints \( \|DX - Y\|_F \leq \delta \) for \( \delta > 0 \)) remain important open problems.

On the other hand, one may wish to focus on “good news”, e.g., by designing efficient approximation algorithms that give performance guarantees not too much worse than our intractability thresholds, or by identifying special cases which are notably easier to solve. Also, it would be interesting to develop further “hybrid algorithms” that combine relaxation methods and tools from combinatorial optimization, such as the branch & bound procedure from [43].
D. L. Donoho, “Compressed Sensing,” IEEE Trans. Inform. Theory, vol. 52, no. 4, pp. 1289–1306, 2006.

S. Foucart and H. Rauhut, A Mathematical Introduction to Compressive Sensing, ser. Applied and Numerical Harmonic Analysis. Birkhäuser, 2013.

G. Kutyniok and Y. C. Eldar, Eds., Compressed Sensing: Theory and Applications. Cambridge University Press, 2012.

M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness. W. H. Freeman and Company, 1979.

B. K. Natarajan, “Sparse Approximate Solutions to Linear Systems,” SIAM J. Comput., vol. 24, no. 2, pp. 227–234, 1995.

E. Amaldi and V. Kann, “On the Approximability of Minimizing Nonzero Variables or Unsatisfied Relations in Linear Systems,” Theor. Comput. Sci., vol. 209, no. 1–2, pp. 237–260, 1998.

E. Amaldi, “On the complexity of designing compact perceptrons and some consequences,” in El. Proc. 5th Internat. Symp. on Artificial Intelligence and Math., 1999.

Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, “Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition,” in Proc. 27th Ann. Asilomar Conference on Signals, Systems and Computers. IEEE Computer Society Press, 1993, vol. 1, pp. 40–44.

S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic Decomposition by Basis Pursuit,” SIAM J. Sci. Comput., vol. 20, no. 1, pp. 33–61, 1998.

A. M. Tillmann, “Computational Aspects of Compressed Sensing,” Doctoral dissertation, TU Darmstadt, Germany, 2013.

M. Aharon, M. Elad, and A. M. Bruckstein, “The Need for spikes in signal representations: The k-SVD Algorithm and Sparse Signal Approximation,” IEEE Trans. Signal Process., vol. 54, no. 11, pp. 4311–4322, 2006.

R. Rubinstein, T. Peleg, and M. Elad, “Analysis K-SVD: A Dictionary-Learning Algorithm for the Analysis Sparse Model,” IEEE Trans. Signal Process., vol. 56, no. 3, pp. 661–677, 2013.

M. Yaghoobi, S. Nam, R. Gribonval, and M. E. Davies, “Analysis Operator Learning for Overcomplete Cosparse Representations,” in Proc. EUSIPCO’11, 2011.

V. Emiya, A. Bonnefoy, L. Daudet, and R. Gribonval, “Compressed Sensing with Unknown Sensor Permutation,” in Proc. IEEE ICASSP’14, 2014, pp. 1040–1044.

D. Moshkovitz, “The Projection Games Conjecture and the NP-Hardness of Inapproximating Set-Cover,” Preprint, 2014. [Online]. Available: http://people.csail.mit.edu/dmoshkov/papers/set-cover/set-cover-full.pdf

M. Razaviyayn, H.-W. Tseng, and Z.-Q. Luo, “Dictionary Learning for Sparse Representation: Complexity and Algorithms,” in Proc. IEEE ICASSP’14, 2014, pp. 5247–5251.