Sparse and Low-Rank Tensor Decomposition

Authored by:

Nikhil Rao
Parikshit Shah
Gongguo Tang

Abstract

Motivated by the problem of robust factorization of a low-rank tensor, we study the question of sparse and low-rank tensor decomposition. We present an efficient computational algorithm that modifies Leurgans' algoirthm for tensor factorization. Our method relies on a reduction of the problem to sparse and low-rank matrix decomposition via the notion of tensor contraction. We use well-understood convex techniques for solving the reduced matrix sub-problem which then allows us to perform the full decomposition of the tensor. We delineate situations where the problem is recoverable and provide theoretical guarantees for our algorithm. We validate our algorithm with numerical experiments.

1 Paper Body

Tensors are useful representational objects to model a variety of problems such as graphical models with latent variables [1], audio classification [20], psychometrics [8], and neuroscience [3]. One concrete example proposed in [1] involves topic modeling in an exchangeable bag-of-words model wherein given a corpus of documents one wishes to estimate parameters related to the different topics of the different documents (each document has a unique topic associated to it). By computing the empirical moments associated to (exchangeable) bi-grams and tri-grams of words in the documents, [1] shows that this problem reduces to that of a (low rank) tensor decomposition. A number of other machine learning tasks, such as Independent Component Analysis [11], and learning Gaussian mixtures [2] are reducible to that of tensor decomposition. While most tensor problems are computationally intractable [12] there has been renewed interest in developing tractable and principled approaches for the same [4, 5, 12, 15, 19, 21, 24?27]. In this paper we consider the problem of performing tensor decompositions when a subset of the entries of a low-rank tensor X are corrupted adversarially, so that the tensor observed is Z = X +Y where Y is the corruption. One may view this problem as the tensor version of a sparse and low-rank matrix decomposition problem as studied in [6, 9, 10, 13]. We develop an algo-

rithm for performing such a decomopsition and provide theoretical guarantees as to when such decomposition is possible. Our work draws on two sets of tools: (a) The line of work addressing the Robust PCA problem in the matrix case [6, 9], and (b) Application of Leaurgans? algorithm for tensor decomposition and tensor inverse problems [4, 17, 24]. Our algorithm is computationally efficient and scalable, it relies on the key notion of tensor contraction which effectively reduces a tensor problem of dimension n ? n ? n to four decompostion problems for matrices of size n?n. One can then apply convex methods for sparse and low-rank matrix decomposition followed by certain linear algebraic operations to recover the constituent tensors. Our algorithm not only produces the correct decomposition of Z into X and Y , but also produces the low rank factorization of X. We are able to avoid tensor unfolding based approaches [14,21,26] which are expensive and would lead to solving convex problems that are larger by orders of magnitude; in the 3rd order case the unfolded matrix would be n2 ? n. Furthermore, our method is 1

conceptually simple, to impelement as well as to analyze theoretically. Finally our method is also modular ? it can be extended to the higher order case as well as to settings where the corrupted tensor Z has missing entries, as described in Section 5. 1.1

Problem Setup

In this paper, vectors are denoted using lower case characters (e.g. x, y, a, b, etc.), matrices by uppercase characters (e.g. X, Y, etc,) and tensors by upper-case bold characters (e.g. X, T , A etc.). We will work with tensors of third order (representationally to be thought of as three-way arrays), and the term mode refers to one of the axes of the tensor. A slice of a tensor refers to a two dimensional matrix generated from the tensor by varying indices along two modes while keeping the third mode fixed. For a tensor X we will refer to the indices of the ith mode-1 slice (i.e., the slice corresponding (1) to the indices {i} ? [n2] ? [n3]) by Si , where [n2] = {1, 2, . . . , n2 } and [n3] is defined similarly. (1) We denote the matrix corresponding to Si by Xi1 . Similarly the indices of the k th mode-3 slice (3) will be denoted by Sk and the matrix by Xk3 . Given a tensor of interest X, consider its decomposition into rank one tensors r X ? i u i ? vi ? w i , X=

(1)

i=1

where {ui }i=1,...,r ? Rn1 , {vi }i=1,...,r ? Rn2 , and {wi }i=1,...,r ? Rn3 are unit vectors. Here ? denotes the tensor product, so that X ? Rn1 ?n2 ?n3 is a tensor of order 3 and dimension n1 ? n2 ? n3 . Without loss of generality, throughout this paper we assume that n1 ? n2 ? n3 . We will present our results for third order tensors, and analogous results for higher orders follow in a transparent manner. We will be dealing with low-rank tensors, i.e. those tensors with r ? n1 . Tensors can have rank larger than the dimension, indeed r ? n3 is an interesting regime, but far more challenging and is a topic left for future work. Kruskal?s Theorem [16] guarantees that tensors satisfying Assumption 1.1 below have a unique minimal decomposition into rank one terms of the form (1). The number of terms is called the (Kruskal) rank. Assumption 1.1. {ui

$\}_{i=1,\ldots,r} \in R^{n_1}$, $\{v_i\}_{i=1,\ldots,r} \in R^{n_2}$, and $\{w_i\}_{i=1,\ldots,r} \in R^{n_3}$ are sets of linearly independent vectors. While rank decomposition of tensors in the worst case is known to be computationally intractable [12], it is known that the (mild) assumption stated in Assumption 1.1 above suffices for an algorithm known as Leurgans? algorithm [4, 18] to correctly identify the factors in this unique decomposition. In this paper, we will make this assumption about our tensor X throughout. This assumption may be viewed as a ?genericity? or ?smoothness? assumption [4]. In (1), r is the rank, $\lambda_i \in R$ are scalars, and $u_i \in R^{n_1}$, $v_i \in R^{n_2}$, $w_i \in R^{n_3}$ are the tensor factors. Let $U \in R^{n_1 \times r}$ denote the matrix whose columns are u_i, and correspondingly define $V \in R^{n_2 \times r}$ and $W \in R^{n_3 \times r}$. Let $Y \in R^{n_1 \times n_2 \times n_3}$ be a sparse tensor to be viewed as a ?corruption? or adversarial noise added to X, so that one observes: $Z = X + Y$. The problem of interest is that of decomposition, i.e. recovering X and Y from Z. For a tensor X, we define its mode-3 contraction with respect to a contraction vector $a \in R^{n_3}$, denoted by $Xa^3 \in R^{n_1 \times n_2}$, as the following matrix:

$$Xa^3{}_{ij} = \sum_{k=1}^{n_3} X_{ijk} a_k, \quad (2)$$

so that the resulting $n_1 \times n_2$ matrix is a weighted sum of the mode-3 slices of the tensor X. Under this notation, the k th mode-3 slice matrix X_k^3 is a mode-3 contraction with respect to the k th canonical basis vector. We similarly define the mode-1 contraction with respect to a vector $c \in R^{n_1}$ as

$$X^1 c{}_{jk} = \sum_{i=1}^{n_1} X_{ijk} c_i. \quad (3)$$

2

In the subsequent discussion we will also use the following P notation. For a matrix M, $\|M\|$ refers to the spectral norm, $\|M\|_*$ the nuclear norm, $\|M\|_1 := \sum_{i,j} |M_{ij}|$ the elementwise ℓ_1 norm, and $\|M\|_\infty := \max_{i,j} |M_{i,j}|$ the elementwise ℓ_∞ norm. 1.2

Incoherence

The problem of sparse and low-rank decomposition for matrices has been studied in [6, 9, 13, 22], and it is well understood that exact decomposition is not always possible. In order for the problem to be identifiable, two situations must be avoided: (a) the low-rank component X must not be sparse, and (b) the sparse component Y must not be low-rank. In fact, something stronger is both necessary and sufficient: the tangent spaces of the low-rank matrix (with respect to the rank variety) and the sparse matrix (with respect to the variety of sparse matrices) must have a transverse intersection [9]. For the problem to be amenable to recovery using comptationally tractable (convex) methods, somewhat stronger, incoherence assumptions are standard in the matrix case [6,7,9]. We will make similar assumptions for the tensor case, which we now describe. Given the decomposition (1) of X we define the following subspaces of matrices:

$T_{U,V} = \{U A^T + B V^T : A \in R^{n_2 \times r}, B \in R^{n_1 \times r}\}$

$T_{V,W} = \{V C^T + D W^T : C \in R^{n_3 \times r}, D \in R^{n_2 \times r}\}$.

(4)

Thus $T_{U,V}$ is the set of rank r matrices whose column spaces are contained in span(U) or row spaces are contained in span(V) respectively, and a similar definition holds for $T_{V,W}$ and matrices V, W. If Q is a rank r matrix with column space span(U) and row space span(V), $T_{U,V}$ is the tangent space at

3

Q with respect to the variety of rank r matrices. For a tensor Y , the support of Y refers to the indices corresponding to the non-zero entries of Y . (3) Let ? ? [n1] ? [n2] ? [n3] denote the support of Y . Further, for a slice Yi3 , let ?i ? [n1] ? [n2] (k) denote the corresponding sparsity pattern of the slice Yi3 (more generally ?i can be defined as th the sparsity of the matrix resulting from the i mode k slice). When a tensor contraction of Y is computed along mode k, the sparsity of the resulting matrix is the union of the sparsity patterns of

Snk (k) each (matrix) slice, i.e. ?(k) = i=1 ?i . Let S ?(k) denote the set of (sparse) matrices with support ?(k) . We define the following incoherence parameters: ? (U, V) :=

max

kM k?

? (V, W) :=

M ?TU,V :kM k?1

? ?(k) :=

max

max

kM k?

M ?TV,W :kM k?1

kN k.

N ?S (?(k)):kN k? ?1

The quantities ? (U, V) and ? (V, W) being small implies that for contractions of the tensor Z, all matrices in the tangent space of those contractions with respect to the variety of rank r matrices are ?diffuse?, i.e. do not have sparse elements [9]. Similarly, ? ?(k) being small implies that all matrices with the contracted sparsity pattern ?(k) are such that their spectrum is ?diffuse?, i.e. they do not have low rank. We will see specific settings where these forms of incoherence hold for tensors in Section 3.

2

Algorithm for Sparse and Low Rank Tensor Decomposition

We now introduce our algorithm to perform sparse and low rank tensor decompositions. We begin with a Lemma: Lemma 2.1. Let X ? Rn1 ?n2 ?n3 , with n1 ? n2 ? n3 be a tensor of rank r ? n1 . Then the rank of Xa3 is at most r. Similarly the rank of Xc1 is at most r. Pr Proof. Consider a tensor X = i=1 ?i ui ? vi ? wi . The reader may verify in a straightforward manner that Xa3 enjoys the decomposition: Xa3 =

r X

?i hwi , aiui viT .

i=1

3

(5)

The proof for the rank of Xc1 is analogous. Note that while (5) is a matrix decomposition of the contraction, it is not a singular value decomposition (the components need not be orthogonal, for instance). Recovering the factors needs an application of simultaneous diagonalization, which we describe next. Pr Lemma 2.2. [4, 18] Suppose we are given an order 3 tensor X = i=1 ?i ui ? vi

? wi of size n1 ? n2 ? n3 satisfying the conditions of Assumption 1.1. Suppose the contractions Xa3 and Xb3 are computed with respect to unit vectors a, b ? Rn3 distributed independently and uniformly on the unit sphere Sn3 ?1 and consider the matrices M1 and M2 formed as: M1 = Xa3 (Xb3)?

M2 = (Xb3)? Xa3 .

Then the eigenvectors of M1 (corresponding to the non-zero eigenvalues) are {ui }i=1,...,r , and the eigenvectors of M2T are {vi }i=1,...,r . Remark Note that while the eigenvectors {ui } , {vj } are thus determined, a source of ambiguity remains. For a fixed ordering of {ui } one needs to determine the order in which {vj } are to be arranged. This can be (generically) achieved by using the (common) eigenvalues of M1 and M2 for pairing i(f the contractions Xa3 , Xb3 are computed with respect to random vectors a, b the eigenvalues are distinct almost surely). Since the eigenvalues of M1 , M2 are distinct they can be used to pair the columns of U and V . Lemma 2.2 is essentially a simultaneous diagonalization result [17] that facilitates tensor decomposition [4]. Given a tensor T , one can compute two contractions for mode 1 and apply simultaneous diagonalization as described in Lemma 2.2 - this would yield the factors vi , wi (up to sign and reordering). One can then repeat the same process with mode 3 contractions to obtain ui , vi . In the final step one can then obtain ?i by solving a system of linear equations. The full algorithm is described in Algorithm 2 in the supplementary material. For a contraction Zvk of a tensor Z with respect to a vector v along mode k, consider solving the convex problem: minimize X ,Y

kX k? + ?k kYk1
subject to
Zvk = X + Y.
(6)

Our algorithm, stated in Algorithm 1, proceeds as follows: Given a tensor Z = X + Y , we perform (3) (3) two random contractions (w.r.t. vectors a, b) of the tensor along mode 3 to obtain matrices Za , Zb (3) (3) . Since Z is a sum of sparse and low-rank components, by Lemma 2.1 so are the matrices Za , Zb . We thus use (6) to decompose them into constituent sparse and low-rank components, which are the (3) (3) (3) (3) (3) (3) contractions of the matrices Xa , Xb , Ya , Yb . We then use Xa , Xb and Lemma 2.2 to obtain the factors U, V . We perform the same operations along mode 1 to obtain factors V, W . In the last step, we solve for the scale factors ?i (a system of linear equations). Algorithm 2 in the supplementary material, which we adopt for our decomposition problem in Algorithm 1, essentially relies on the idea of simultaneous diagonalization of matrices sharing common row and column spaces [17]. In this paper we do not analyze the situation where random noise is added to all the entries, but only the sparse adversarial noise setting. We note, however, that the key algorithmic insight of using contractions to perform tensor recovery is numerically stable and robust with respect to noise, as has been studied in [4, 11, 17]. Parameters that need to be picked to implement our algorithm are the regularization coefficients ?1 , ?3 . In the theoretical guarantees we will see that this can be picked in a stable manner, and that a range of values guarantee exact decomposition when

the suitable incoherence conditions hold. In practice these coefficents would need to be determined by a cross-validation method. Note also that under suitable random sparsity assumptions [6], the regularization coefficient may be picked to be the inverse of the square-root of the dimension. 2.1

Computational Complexity

The computational complexity of our algorithm is dominated by the complexity of perfoming the sparse and low-rank matrix decomposition of the contractions via (6). For simplicity, let us consider 4

Algorithm 1 Algorithm for sparse and low rank tensor decomposition 1: 2: 3: 4: 5: 6:

7: 8: 9: 10:

11:

12:

Input: Tensor Z, parameters ?1 , ?3 . Generate contraction vectors a, b ? Rn3 independently and uniformly distributed on unit sphere. Compute mode 3 contractions Za3 and Zb3 respectively. Solve the convex problem (6) with v = a, k = 3. Call the resulting solution matrices Xa3 , Ya3 , and regularization parameter ?1 . Solve the convex problem (6) with v = b, k = 3. Call the resulting solution matrices Xb3 , Yb3 and regularization parameter ?3 . Compute eigendecomposition of M1 := Xa3 (Xb3)? and M2 := (Xb3)? Xa3 . Let U and V denote the matrices whose columns are the eigenvectors of M1 and M2T respectively corresponding to the non-zero eigenvalues, in sorted order. (Let r be the (common) rank of M1 and M2 .) The eigenvectors, thus arranged are denoted as {ui }i=1,...,r and {vi }i=1,...,r . Generate contraction vectors c, d ? Rn1 independently and uniformly distributed on unit sphere. Solve the convex problem (6) with v = c, k = 1. Call the resulting solution matrices Xc1 , Yc1 and regularization parameter ?3 . Solve the convex problem (6) with v = d, k = 1. Call the resulting solution matrices Xd1 , Yd1 and regularization parameter ?4 . ? denote Compute eigen-decomposition of M3 := Xc1 (Xd1)? and M4 := (Xc1)? Xd1 . Let V? and W T the matrices whose columns are the eigenvectors of M3 and M4 respectively corresponding to the non-zero eigenvalues, in sorted order. (Let r be the (common) rank of M3 and M4 .) ? , also performing simultaneous sign reversals as Simultaneously reorder the columns of V? , W ? necessary so that the columns of V and V are equal, call the resulting matrix W with columns {wi }i=1,...,r . Solve for ?i in the linear system Xa3 =

r X

?i ui viT hwi , ai.

i=1

? := 13: Output: Decomposition X

Pr

i=1

? ?i ui ? vi ? wi , Y? := Z ? X.

the case where the target tensor Z ? Rn?n?n has equal dimensions in different modes. Using 3 a standard first order method, the solution of (6) has a per iteration complexity of O(n), and to 1 achieve an accuracy of , O iterations arerequired [22]. Since only four such steps need be

3

performed, the complexity of the method is O n where is the accuracy to which (6) is solved. Another alternative is to reformulate (6) such that it is amenable to greedy atomic approaches [23], which yields an order of magnitude improvement. We note that in contrast, a tensor unfolding for this problem [14, 21, 26] results in the need to solve much larger convex programs. For instance, for Z ? Rn?n?n , the resulting flattened matrix be of size n2 ? n and the resulting convex prob 4would lem would then have a complexity of O n . For higher order tensors, the gap in computational complexity would increase by further orders of n. 2.2

Numerical Experiments

We now present numerical results to validate our approach. We perform experiments for tensors of size 50 ? 50 ? 50 (non-symmetric). A tensor Z is generated as the sum of a low rank tensor X and a sparse tensor Y . The low-rank component is generated as follows: Three sets of r unit vecots ui , vi , wi ? R50 are generated randomly, independently and uniformly distributed on the unit sphere. P Also a random positive scale factor (uniformly distributed on [0, 1] is chosen and the r tensor X = i=1 ?i ui ? vi ? wi . The tensor Y is generated by (Bernoulli) randomly sampling its entries with probability p. For each such p, we perform 10 trials and apply our algorithm. In all our experiments, the regularization parameter was picked to be ? = ?1n . The optimization problem (6) is solved using CVX in MATLAB. We report success if the MSE is smaller than 10?5 , separately for both the X and Y components. We plot the empirical probability of success as a function of p in Fig. 1 (a), (b), for multiple values of the true rank r. In Fig. 1 (c), (d) we test the scalability 5

0.8 0.6

P(recovery)

0.6 0.4

0.4

0.2 0 0

r=1 r=2 r=3 r=4

0.2

0.5

1 sparsity x 100

1.5

2

0 0

0.5

1 sparsity x 100

1.5

5

5

4

4

3

2

7
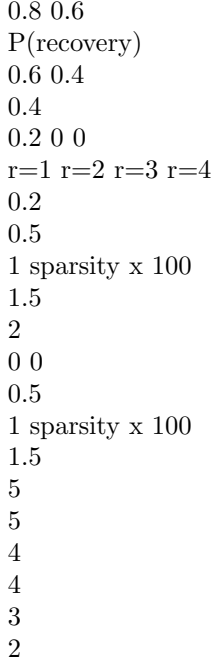
1
0 0
2
Inexact Recoveries
0.8 P(recovery)
1
r=1 r=2 r=3 r=4
# Inexact Recoveries
1
0.05
0.1
0.15
0.2
3
2
1
0 0
0.05
Corruption Sparsity
0.1
0.15
0.2
Corruption Sparsity

(a) Low Rank Compo- (b) Sparse Component (c) Low Rank Compo- (d) Sparse Component nent nent

Figure 1: Recovery of the low rank and sparse components from our proposed methods. In figures (a) and (b) we see that the probability of recovery is high when both the rank and sparsity are low. In figures (c) and (d) we study the recovery error for a tensor of dimensions 300 ? 300 ? 300 and rank 50. of our method, by generating a random 300 ? 300 ? 300 tensor of rank 50, and corrupting it with a sparse tensor of varying sparsity level. We run 5 independent trials and see that for low levels of corruption, both the low rank and sparse components are accurately recovered by our method.

3

Main Results

We now present the main rigorous guarantees related to the performance of our algorithm. Due to space constraints, the proofs are deferred to the supplementary materials. Pr Theorem 3.1. Suppose Z = X + Y , where X = i=1 ?i ui ? vi ? wi , has rank r ? n1 and such that the factors satisfy Assumption 1.1. Suppose Y has support ? and the following condition is satisfied:

1 1 ? ?(1) ? (V, W) ¡ . ? ?(3) ? (U, V) ? 6 6 Then Algoritm 1 succeeds in exactly recovering the component tensors, i.e. (X, Y) =

1?3?(U,V)?(?(3)) ?(U,V) ? Y?) whenever ?k are picked so that ?3 ? , (X, and 1?4?(U,V)?(?(3)) ?(?(3))

1?3?(V,W)?(?(1)) ?(V,W)))p , ?1 ? . Specifically, choice of ?3 = (3?(U,V 1?p and (3) 1?4?(V,W)?(?(1)) ?(?(1)) (?(?)) p)) ?1 = (3?(V,W 1?p for any

8

p ? [0, 1] in these respective intervals guarantees exact recovery. (?(?(1))) For a matrix M , the degree of M , denoted by deg(M), is the maximum number of non-zeros in any row or column of M . For a tensor Y , we define the degree along mode k, denoted by $\deg_k$ (Y) to be the maximum number of non-zero entries in any row or column of a matrix supported on ?(k) (defined in Section 1.2). The degree of Y is denoted by deg(Y) := $\max_{k?\{1,2,3\}}$ $\deg_k$ (Y). Lemma 3.2. We have:

? ?(k) ? deg(Y), for all k. For a subspace S ? Rn , let us define the incoherence of the subspace as: ?(S) := $\max_{k PS}$ e_i k2 , i

where P_S denotes the projection operator onto S, e_i is a standard unit vector and k ? k2 is the Euclidean norm of a vector. Let us define: inc(X) := max {? (span(U)) , ? (span(V)) , ? (span(W))} inc_3 (X) := max {? (span(U)) , ? (span(V))} inc_1 (X) := max {? (span(V)) , ? (span(W))} . 6

Note that inc(X) ¡ 1, always. For many random ensembles qof interest, we have that the incoherence scales gracefully with the dimension n, i.e.: inc(X) ? K

max{r,log n} . n

Lemma 3.3. We have ? (U, V) ? 2 inc(X)

? (V, W) ? 2 inc(X). Pr

Corollary 3.4. Let Z = X + Y , with X = $\sum_{i=1}$ $?_i$ u_i ? v_i ? w_i and rank r ? n1 , the factors satisfy Assumption 1.1 and incoherence inc(X). Suppose Y is sparse and has degree deg(Y). If the condition 1 inc(X)deg(Y) ¡ 12 ? Y?) when the holds then Algorithm 1 successfully recovers the true solution, i.e. . (X, Y) = (X, parameters

2inc_3 (X) 1 ? 6deg_3 (Y)inc_3 (X) $?_3$? , 1 ? 8deg_3 (Y)inc_3 (X) deg_3 (Y)

2inc_1 (X) 1 ? 6deg_1 (Y)inc_1 (X) $?_1$? , . 1 ? 8deg_1 (Y)inc_1 (X) deg_1 (Y)

p

(6inc_3 (X)) Specifically, a choice of $?_3$ = (2deg 1?p , $?_1$ = 3 (Y)) that guarantees exact recovery.

(6inc_1 (X))p (2deg_1 (Y))1?p

for any p ? [0, 1] is a valid choice

Remark Note that Corollary 3.4 presents a deterministic guarantee on the recoverability of a sparse corruption of a low rank tensor, and can be viewed as a tensor extension of [9, Corollary 3]. We now consider, for the sake of simplicity, tensors of uniform dimension, i.e. X, Y , Z ? Rn?n?n . We show that when the low-rank and sparse components are suitably random, the approach outlined in Algorithm 1 achieves exact recovery. We define the random sparsity model to be one where each entry of the tensor Y is non-zero independently and with identical probability ?. We make no assumption about the mangitude of the entries of Y , only that its non-zero entries are thus sampled. Pr n Lemma 3.5. Let X = $\sum_{i=1}$ $?_i$ u_i ? v_i ? w_i , where u_i , v_i , w_i ? R are uniformly randomly n?1 distributed on the unit sphere S . Then the incoherence of the tensor X satisifies: r max {r, log n} inc(X) ? c_1 n with probability exceeding 1 ? c_2 n?3 log n for some constants c_1 , c_2 . Lemma3.6. Suppose the entriesof Y are sampled according to the random sparsity model, and $?_1$ 3 ? n . Then the tensor Y satisfies: deg(Y) ? $12c_1$ max(log ? = O n 2 max(log n, r) n,r) with

9

? n probability exceeding 1 ? exp ?c3 max(log n,r) for some constant c3 ¿ 0. Corollary 3.7. Let Z = X + Y where X is low rank with random factors as per the conditions of Lemma 1 3.5 and Y is sparse with random support as per the conditions in Lemma 3.6. Provided ? Y?) = (X, Y) 2 r ? o n , Algorithm 1 successfully recovers the correct decomposition, i.e. (X, with probability exceeding 1 ? n?? for some ? ¿ 0. Remarks 1) Under this sampling model, the cardinality of the support of Y is allowed to be as large 3 as m = O(n 2 log?1 n) when the rank r is constant (independent of n). 2) We could equivalently have looked at a uniformly random sampling model, i.e. one where a support set of size m is chosen uniformly randomly from the set of all possible support sets of cardinality at most m, and our results for exact recovery would have gone through. This follows from the equivalence principle for successful recovery between Bernoulli sampling and uniform sampling, see [6, Appendix 7.1]. 3) Note that for the random sparsity ensemble, [6] shows that a choice of ? = ?1n ensures exact recovery (an additional condition regarding the magnitudes of the factors is needed, however). By extension, the same choice can be shown to work for our setting. 7

4

Extensions

The approach described in Algorithm 1 and the analysis is quite modular and can be adapted to various settings to account for different forms of measurements and robustness models. We do not present an analysis of these situations due to space constraints, but outline how these extensions follow from the current development in a straightforward manner. 1) Higher Order Tensors: Algorithm 1 can be extended naturally to the higher order setting. Recall that in the third order case, one needs to recover two contractions along the third mode to discover factors U, V and then two contractions along the first mode to discover factors V, W . For an order K tensor of the form Z ? Rn1 ?...?nK which is the sum of a low rank component Pr NK (l) X = i=1 ?i l=1 ui and a sparse component Y , one needs to compute higher order contractions of Z along K ? 1 different modes. For each of these K ? 1 modes the resulting contraction is the sum of a sparse and low-rank matrix, and thus pairs of matrix problems of the form (6) reveal the sparse and low-rank components of the contractions. The low-rank factors can then be recovered via application of Lemma 2.2 and the full decomposition can thus be recovered. The same guarantees as in Theorem 3.1 and Corollary 3.4 hold verbatim (the notions of incoherence inc(X) and degree deg(Y) of tensors need to be extended to the higher order case in the natural way) 2) Block sparsity: Situations where entire slices of the tensor are corrupted may happen in recommender systems with adversarial ratings [10]. A natural approach in this case is to use a convex relaxation of the form minimize M1 ,M2

?k kM1 k? + kM2 k1,2

Zvk = M1 + M2 P in place of (6) in Algorithm 1. In the above, kM k1,2 := i kMi k2 , where Mi is the ith column of M . Since exact recovery of the block-sparse and low-rank components of the contractions are guaranteed via this relaxation under suitable assumptions [10], the algorithm would inherit associated provable guarantees. subject to

3) Tensor completion: In applications such as recommendation systems, it may be desirable to perform tensor completion in the presence of sparse corruptions. In [24], an adaptation of Leurgans? algorithm was presented for performing completion from measurements restricted to only four slices of the tensor with near-optimal sample complexity (under suitable genericity assumptions about the tensor). We note that it is straightforward to blend Algorithm 1 with this method to achieve completion with sparse corruptions. Recalling that Z = X + Y and therefore Zk3 = Xk3 + Yk3 (i.e. the k th mode 3 slice of Z isa sum of sparse and low rank slices of X and Y), if only a subset of elements of Zk3 (say P? Zk3) is observed for some index set ?, we can replace (6) in Algorithm 1 with

minimize ?k kM1 k? + kM2 k1 subject to P? Zvk = P? (M1 + M2) . M1 ,M2

Under suitable incoherence assumptions [6, Theorem 1.2], the above will achieve exact recovery of the slices. Once four slices are accurately recovered, one can then use Leurgans? algorithm to recover the full tensor [24, Theorem 3.6]. Indeed the above idea can be extended more generally to the concept of deconvolving a sum of sparse and low-rank tensors from separable measurements [24]. 4) Non-convex approaches: A basic primitive for sparse and low-rank tensor decomposition used in this paper is that of using (6) for matrix decomposition. More efficient non-convex approaches such as the ones described in [22] may be used instead to speed up Algorithm 1. These alternative

1 2 nonconvex methods [22] requre O(rn) steps per iterations, and O log iterations resulting in a

1 2 total complexity of O rn log for solving the decomposition of the contractions to an accuracy of .

2 References

[1] A. A NANDKUMAR , R. G E , D. H SU , AND S. M. K AKADE, A tensor approach to learning mixed membership community models, The Journal of Machine Learning Research, 15 (2014), pp. 2239?2312. [2] A. A NANDKUMAR , R. G E , D. H SU , S. M. K AKADE , AND M. T ELGARSKY, Tensor decompositions for learning latent variable models, Tech. Rep. 1, 2014.
8
[3] C. B ECKMANN AND S. S MITH, Tensorial extensions of independent component analysis for multisubject FMRI analysis, NeuroImage, 25 (2005), pp. 294?311. [4] A. B HASKARA , M. C HARIKAR , A. M OITRA , AND A. V IJAYARAGHAVAN, Smoothed analysis of tensor decompositions, in Proceedings of the 46th Annual ACM Symposium on Theory of Computing, ACM, 2014, pp. 594?603. [5] S. B HOJANAPALLI AND S. S ANGHAVI, A new sampling technique for tensors, arXiv preprint arXiv:1502.05023, (2015). [6] E. J. C AND E' S , X. L I , Y. M A , AND J. W RIGHT, Robust principal component analysis?, Journal of the ACM, 58 (2011), pp. 11?37. [7] E. J. C AND E' S AND B. R ECHT, Exact matrix completion via convex optimization, Foundations of

Computational Mathematics, 9 (2009), pp. 717?772. [8] R. B. C ATTELL, Parallel proportional profiles and other principles for determining the choice of factors by rotation, Psychometrika, 9 (1944), pp. 267?283. [9] V. C HAN- DRASEKARAN , S. S ANGHAVI , P. A. PARRILO , AND A. S. W ILLSKY, Rank-sparsity incoherence for matrix decomposition, SIAM Journal on Optimization, 21 (2011), pp. 572?596. [10] Y. C HEN , H. X U , C. C ARAMANIS , AND S. S ANGHAVI, Robust matrix completion and corrupted columns, in Proceedings of the 28th International Conference on Machine Learning (ICML-11), L. Getoor and T. Scheffer, eds., New York, NY, USA, 2011, ACM, pp. 873?880. [11] N. G OYAL , S. V EMPALA , AND Y. X IAO, Fourier PCA and robust tensor decomposition, in Proceedings of the 46th Annual ACM Symposium on Theory of Computing, ACM, 2014, pp. 584?593. [12] C. J. H ILLAR AND L.-H. L IM, Most tensor problems are NP-hard, Journal of the ACM, 60 (2013), pp. 45:1?45:39. [13] D. H SU , S. K AKADE , AND T. Z HANG, Robust matrix decomposition with sparse corruptions, Information Theory, IEEE Transactions on, 57 (2011), pp. 7221?7234. [14] B. H UANG , C. M U , D. G OLDFARB , AND J. W RIGHT, Provable models for robust low-rank tensor completion, Pacific Journal of Optimization, 11 (2015), pp. 339?364. [15] A. K RISHNAMURTHY AND A. S INGH, Low-rank matrix and tensor completion via adaptive sampling, in Advances in Neural Information Processing Systems, 2013. [16] J. B. K RUSKAL, Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics, Linear Algebra Applicat., 18 (1977). [17] V. K ULESHOV, A. C HAGANTY, AND P. L IANG, Tensor factorization via matrix factorization, arXiv.org, (2015). [18] S. L EURGANS , R. ROSS , AND R. A BEL, A decomposition for three-way arrays, SIAM Journal on Matrix Analysis and Applications, 14 (1993), pp. 1064?1083. [19] Q. L I , A. P RATER , L. S HEN , AND G. TANG, Overcomplete tensor decomposition via convex optimization, in IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), Cancun, Mexico, Dec. 2015. [20] N. M ESGARANI , M. S LANEY, AND S. A. S HAMMA, Discrimination of speech from non-speech based on multiscale spectro-temporal modulations, Audio, Speech and Language Processing, IEEE Transactions on, 14 (2006), pp. 920?930. [21] C. M U , B. H UANG , J. W RIGHT, AND D. G OLDFARB, Square deal: Lower bounds and improved relaxations for tensor recovery, preprint arXiv:1307.5870, 2013. [22] P. N ETRAPALLI , U. N IRANJAN , S. S ANGHAVI , A. A NANDKUMAR , PCA, in Advances in Neural Information Processing Systems, 2014.

AND

P. JAIN, Non-convex robust

[23] N. R AO , P. S HAH , AND S. W RIGHT, Forward-backward greedy algorithms for signal demixing, in Signals, Systems and Computers, 2013 Asilomar Conference on, IEEE, 2014. [24] P. S HAH , N. R AO , AND G. TANG, Optimal low-rank tensor recovery from separable measurements: Four contractions suffice, arXiv.org, (2015). [25] G. TANG AND P. S HAH, Guaranteed tensor decomposition: A moment approach, International Conference on Machine Learning (ICML 2015), (2015), pp. 1491?1500. [26] R. T OMIOKA , K. H

AYASHI , AND H. K ASHIMA, Estimation of low-rank tensors via convex opti-
mization, preprint arXiv:1010.0789, 2011. [27] M. Y UAN AND C.-H. Z HANG,
On tensor completion via nuclear norm minimization, preprint arXiv:1405.1773,
2014.

9