Tensor Factorization via Matrix Factorization

Volodymyr Kuleshov* **Arun Tejasvi Chaganty*** **Percy Liang**
Department of Computer Science
Stanford University
Stanford, CA 94305

Abstract

Tensor factorization arises in many machine learning applications, such as knowledge base modeling and parameter estimation in latent variable models. However, numerical methods for tensor factorization have not reached the level of maturity of matrix factorization methods. In this paper, we propose a new algorithm for CP tensor factorization that uses random projections to reduce the problem to simultaneous matrix diagonalization. Our method is conceptually simple and also applies to non-orthogonal and asymmetric tensors of arbitrary order. We prove that a small number random projections essentially preserves the spectral information in the tensor, allowing us to remove the dependence on the eigengap that plagued earlier tensor-to-matrix reductions. Experimentally, our method outperforms existing tensor factorization methods on both simulated data and two real datasets.

1 Introduction

Given a tensor $\widehat{T} \in \mathbb{R}^{d \times d \times d}$ of the following form:

$$\widehat{T} = \sum_{i=1}^{k} \pi_i a_i \otimes b_i \otimes c_i + \text{noise}, \tag{1}$$

our goal is to estimate the factors $a_i, b_i, c_i \in \mathbb{R}^d$ and factor weights $\pi \in \mathbb{R}^k$. In machine learning and statistics, this tensor $\widehat{T}$ typically represents higher-order relationships among variables, and we would like to uncover the salient factors that explain these relationships. This problem of *tensor factorization* is an important problem rich with applications [1]: modeling knowledge bases [2], topic modeling [3], community detection [4], learning graphical models [5, 6]. The last three fall into a class of procedures based on the method of moments for latent-variable models, which are notable because they provide guarantees of consistent parameter estimation [7].

However, tensors, unlike matrices, are fraught with difficulties: identifiability is a delicate issue [8, 9, 10], and computing Equation 1 is in general NP-hard [11, 12]. In this work, we propose a simple procedure to reduce the problem of factorizing tensors to that of factorizing matrices. Specifically, we first project the tensor $\hat{T}$ onto a set of random vectors, producing a set of matrices. Then we simultaneously diagonalize the matrices, producing an estimate of the factors of the original tensor. We can optionally refine our estimate by running the procedure using the estimated factors rather than random vectors. Our approach applies to orthogonal, non-orthogonal and asymmetric tensors of arbitrary order.

From a practical perspective, this approach enables us to immediately leverage mature algorithms for matrix factorization. Such algorithms often have readily available implementations that are numerically stable and highly optimized. In our experiments, we observed that they contribute to improvements in accuracy and speed over methods that deal directly with a tensor.

From a theoretical perspective, we consider both *statistical* and *optimization* aspects of our method. Most of our results pertain to the former: we provide guarantees on the accuracy of a solution as a function of the noise ϵ (this noise typically comes from the statistical estimation of T from finite data) that are comparable to those of existing methods (Table 1). Algorithms based on matrix diagonalization have been previously criticized [7] to be extremely sensitive to noise due to a dependence on the smallest difference between eigenvalues (the eigengap). We show that this dependence can be entirely avoided using just $O(\log k)$ tensor projections chosen uniformly at random. Furthermore,

Appearing in Proceedings of the 18^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38.

* These authors contributed equally.

our guarantees are independent of the algorithm used for diagonalizing the projection matrices.

The *optimization* aspects of our method, on the other hand, depend on the choice of joint diagonalization subroutine. Most subroutines enjoy local quadratic convergence rates [13, 14, 15] and so does our method. With sufficiently low noise, global convergence guarantees can be established for some joint diagonalization algorithms [16]. More importantly, local optima are not an issue for our method in practice, which is in sharp contrast to some other approaches, such as expectation maximization (EM).

Finally, we show that our method obtains accuracy improvements over alternating least squares and the tensor power method on several synthetic and real datasets. On a community detection task, we obtain up to a 15% reduction in error compared to a recently proposed approach [4], and up to an 8% reduction in error on a crowdsourcing task [17], matching or outperforming a state-of-the-art EM-based estimator on three of the four datasets.

Notation Let $[n] = \{1, \ldots, n\}$ denote the first n positive integers. Let e_i be the indicator vector which is 1 in component i and 0 in all other components. We use $\otimes$ to denote the tensor product: if $u, v, w \in \mathbb{R}^d$, then $u \otimes v \otimes w \in \mathbb{R}^{d \times d \times d}$.[1] For a third order tensor $T \in \mathbb{R}^{d \times d \times d}$ we define vector and matrix application as,

$$T(x, y, z) = \sum_{i=1}^{d} \sum_{j=1}^{d} \sum_{k=1}^{d} T_{ijk} x_i y_j z_k$$

$$T(X, Y, Z)_{ijk} = \sum_{l=1}^{d} \sum_{m=1}^{d} \sum_{n=1}^{d} T_{lmn} X_{li} Y_{mj} Z_{nk},$$

for vectors $x, y, z \in \mathbb{R}^d$ and matrices $X, Y, Z \in \mathbb{R}^{d \times k}$. The partial vector application (or projection) $T(I, I, w)$ of a vector $w \in \mathbb{R}^d$ returns a $d \times d$ matrix: $T(I, I, w)_{ij} = \sum_{k=1}^{d} T_{ijk} w_k$.

We define the CP decomposition of a tensor $T \in \mathbb{R}^{d \times d \times d}$ as $T = \sum_{i=1}^{k} \pi_i a_i \otimes b_i \otimes c_i$, for $a_i, b_i, c_i \in \mathbb{R}^d$. The rank of T is said to be k. When $a_i = b_i = c_i = u_i$ for all i, and the u_i's are orthogonal, we say T has a symmetric orthogonal factorization, $T = \sum_{i=1}^{k} \pi_i u_i^{\otimes 3}$. Projecting a tensor $T = \sum_{i=1}^{k} \pi_i a_i \otimes b_i \otimes c_i$ along w produces a matrix $T(I, I, w) = \sum_{i=1}^{k} \pi_i (c_i^\top w) a_i \otimes b_i$. We use $\lambda_i = \pi_i (c_i^\top w)$ to refer to the factor weights (or eigenvalues in the orthogonal setting) of the projected matrix.

[1] We will only consider third order tensors for the remainder of this paper, though the approach naturally extends to tensors of arbitrary order.

Method	$\mu <$	$\|u_i - \tilde{u}_i\|_2 <$	Conv.
TPM [7]	0	$\frac{\epsilon}{\pi_{\min}}$	G
Givens [18]	0	?	G
ALS [19]	$\frac{\text{polylog}(d)}{\sqrt{d}}$	$\frac{\epsilon}{\pi_{\min}} + \frac{\sqrt{k/d^{p-1}}}{\pi_{\min}}$	L
SD2 [20]	0	$\frac{k^5}{\pi_{\min}(\min_{i \neq j} \lvert\pi_i - \pi_j\rvert)} \epsilon$	G
This paper	1	$\frac{\|U^{-\top}\|_2^2}{(1-\mu^2)\pi_{\min}^2} \epsilon$	L/G

Table 1: Comparison of tensor factorization algorithms (Section 2.1). For a tensor with noise ϵ (Equation 1) and allowed incoherence μ, we show an upper bound on the error in the recovered factors $\|u_i - \tilde{u}_i\|_2$ and whether the convergence is (L)ocal or (G)lobal. The factor weights π are assumed to be normalized ($\|\pi\|_1 = 1$). $\|U^{-\top}\|_2$ is the 2-norm of the inverse of factors U^{-1}. Our method allows for arbitrary incoherence with a sensitivity to noise comparable to existing methods ([20, 7, 19]), and with better empirical performance. In the orthogonal setting, our algorithm is globally convergent for sufficiently small ϵ.

For a vector of values $\pi \in \mathbb{R}^k$, we use $\pi_{\min}$ and $\pi_{\max}$ to denote the minimum and maximum absolute values of the entries, respectively. Finally, we use δ_{ij} to denote the indicator function, which equals 1 when $i = j$ and 0 otherwise.

2 Background

In this section, we establish the context for tensor factorization, method of moments for estimating latent-variable models, and simultaneous matrix diagonalization.

2.1 Tensor factorization algorithms

Existing tensor factorization methods vary in their sensitivity to noise ϵ in the tensor, their tolerance of non-orthogonality (as measured by the incoherence μ) and in their convergence properties (Table 1). The robust tensor power method (TPM, [7]) is a popular algorithm with theoretical guarantees on global convergence. A recently-developed coordinate-descent method for orthogonal tensor factorization based on Givens rotations [18] is empirically more robust than the TPM; however it is limited to the full-rank setting and lacks a sensitivity analysis. A further limitation of both methods is that they only work for symmetric orthogonal tensors. Asymmetric non-orthogonal tensors could be handled by preprocessing and whitening, but this can be a major source of errors in itself [21]. Alternating least squares (ALS) and other gradient-

based methods [22] are simple, popular, and apply to the non-orthogonal setting, but are known to easily get stuck in local optima [23]. Anandkumar et al. [19] explicitly show both local and global convergence guarantees for a slight modification of the ALS procedure under certain assumptions on the tensor $\hat{T}$.

Finally, some authors have also proposed using simultaneous diagonalization for tensor factorization: Lathauwer [23] proposed a reduction, but it requires forming a linear system of size $O(d^4)$ and is quite complex. Anandkumar et al. [20] performed multiple random projections, but only diagonalized two at a time (SD2), leading to unstable results; the method also only applies to orthogonal factors. Anandkumar et al. [7] briefly remarked that using all the projections at once was possible but did not pursue it. In contrast, our method, has comparable bounds to the tensor power method in the orthogonal setting (conventionally $\|\pi\|_1 = 1$ is assumed), and the ALS method in the non-orthogonal setting. Furthermore, in the non-orthogonal setting, our method works for arbitrary incoherence as long as the factors U are non-singular.

2.2 Parameter estimation in mixture models

Tensor factorization can be used for parameter estimation for a wide range of latent-variable models such as Gaussian mixture models, topic models, hidden Markov models, etc. [7]. For illustrative purposes, we focus on the single topic model [7], defined as follows: For each of n documents, draw a latent "topic" $h \in [k]$ with probability $\mathbb{P}[h = i] = \pi_i$ and three observed words $x_1, x_2, x_3 \in \{e_1, \dots, e_d\}$, which are conditionally independent given h with $\mathbb{P}[x_j = w \mid h = i] = u_{iw}$ for each $j \in \{1, 2, 3\}$. The parameter estimation task is to output an estimate of the parameters $(\pi, \{u_i\}_{i=1}^k)$ given n documents $\{(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}\}_{i=1}^n$ (importantly, the topics are unobserved).

Traditional approaches typically use Expectation Maximization (EM) to optimize the marginal log-likelihood, but this algorithm often gets stuck in local optima. The method of moments approach is to cast estimation as tensor factorization: define the empirical tensor $\hat{T} = \frac{1}{n}\sum_{i=1}^n x_1^{(i)} \otimes x_2^{(i)} \otimes x_3^{(i)}$. It can be shown that $\hat{T} = \sum_{i=1}^k \pi_i u_i \otimes u_i \otimes u_i + \epsilon R$ (a refinement of Equation 1), where $\epsilon R \in \mathbb{R}^{d\times d\times d}$ is the statistical noise which goes to zero as $n \to \infty$. A tensor factorization scheme that asymptotically recovers estimates of $(\pi, \{u_i\}_{i=1}^k)$ therefore provides a consistent estimator of the parameters.

2.3 Simultaneous diagonalization

We now briefly review simultaneous matrix diagonalization, the main technical driver in our approach. In simultaneous diagonalization, we are given a set of symmetric matrices $M_1, \dots, M_L \in \mathbb{R}^{d\times d}$ (see Section 6 for a reduction from the asymmetric case), where each matrix can be expressed as

$$M_l = U\Lambda_l U^\top + \epsilon R_l. \tag{2}$$

The diagonal matrix $\Lambda_l \in \mathbb{R}^{k\times k}$ and the noise ϵR_l are individual to each matrix, but the non-singular transform $U \in \mathbb{R}^{d\times k}$ is common to all the matrices. We also define the full-rank extensions,

$$\bar{U} = \begin{bmatrix} U & U^\perp \end{bmatrix} \qquad \bar{\Lambda}_l = \begin{bmatrix} \Lambda_l & 0 \\ 0 & 0 \end{bmatrix}, \tag{3}$$

where the columns of $U^\perp \in \mathbb{R}^{d-k\times d}$ span the orthogonal subspace of U and $\bar{\Lambda}_l \in \mathbb{R}^{d\times d}$ has been appropriately padded with zeros. Note that $\bar{U}\bar{\Lambda}_l\bar{U}^\top = U\Lambda_l U^\top$.

The goal is to find an invertible transform $V^{-1} \in \mathbb{R}^{d\times d}$ such that each $V^{-1}M_lV^{-\top}$ is nearly diagonal. We refer to the V^{-1} as *inverse factors*. When $\epsilon = 0$, this problem admits a unique solution when there are at least two matrices [24]. There are a number of objective functions for finding V [25, 13, 26], but in this paper, we focus on a popular one that penalizes off-diagonal terms:

$$F(X) \triangleq \sum_{l=1}^{L} \mathrm{off}(X^{-1}M_lX^{-\top}),\ \mathrm{off}(A) = \sum_{i\neq j} A_{ij}^2. \tag{4}$$

An important setting of this problem, which we refer to as the *orthogonal case*, is when we know the true factors U to be orthogonal. In this case we constrain our optimization variable X to be orthogonal as well, i.e. $X^{-1} = X^\top$.

In principle, we could just diagonalize one of the matrices, say M_1 (assuming its eigenvalues are distinct) to recover U. However, when $\epsilon > 0$, this procedure is unreliable and simultaneous diagonalization greatly improves on robustness to noise, as we will witness in Section 4.

There exist several algorithms for optimizing $F(X)$. In this paper, we will use the Jacobi method [27, 25] for the orthogonal case and the QRJ1D algorithm [26] for the non-orthogonal case. Both techniques are based on same idea of iteratively constructing X^{-1} via a product of simple matrices $X^{-1} = B_T \cdots B_2B_1$, where at each iteration $t = 1, \dots, T$, we choose B_t to minimize $F(X)$. Typically, this can be done in closed form.

The Jacobi algorithm for the orthogonal case is a simple adaptation of the Jacobi method for diagonalizing a single matrix. Each B_t is chosen to be a *Givens* rotation [27] defined by two of the d axes $i < j \in [d]$: $B_t = (\cos\theta)(\Delta_{ii} + \Delta_{jj}) + (\sin\theta)(\Delta_{ij} - \Delta_{ji})$ for some

angle θ, where Δ_{ij} is a matrix that is 1 in the (i,j)-th entry and 0 elsewhere. We sweep over all $i<j$, compute the best angle θ in closed form using the formula proposed by Cardoso and Souloumiac [25] to obtain B_t, and then update each M_l by $B_t M_l B_t^\top$. The above can be done in $O(d^3 L)$ time per sweep.

For the non-orthogonal case, the QRJ1D algorithm is similar, except that B_t is chosen to be either a lower or upper unit triangular matrix ($B_t = I + a\Delta_{ij}$ for some a and $i \neq j$). The optimal value of a that minimizes $F(X)$ can also be computed in closed form (see [26] for details). The running time per iteration is the same as before.

3 Tensor factorization via simultaneous matrix diagonalization

We now outline our algorithm for symmetric third order tensors. In Section 6, we describe how to generalize our method to arbitrary tensors. Observe that the projection of $T = \sum_i \pi_i u_i^{\otimes 3}$ along a vector w is a matrix $T(I,I,w) = \sum_i \pi_i (w^\top u_i) u_i^{\otimes 2}$ that preserves all the information about the factors u_i (assuming the $\pi_i(w^\top u_i)$'s are distinct). In principle one can recover the u_i through an eigendecomposition of $T(I,I,w)$. However, this method is sensitive to noise: the error $\|u_i - \tilde{u}_i\|_2$ of an estimated eigenvector $\tilde{u}_i$ depends on the reciprocal of the smallest eigengap $\max_{j\neq i} 1/|\lambda_i - \lambda_j|$ of the projected matrix (recall that $\lambda_i = \pi_i(w^\top u_i)$), which can be large and lead to inaccurate estimates.

Instead, let us obtain the factorization of T from projections along multiple vectors $w_1, w_2, \cdots, w_L$. The projections produce matrices of the form $M_l = \sum_i \lambda_{il} u_i^{\otimes 2}$, with $\lambda_{il} = \pi_i w_l^\top u_i$; they have common eigenvectors, and therefore can be simultaneously diagonalized. As we will show later, joint diagonalization is sensitive to the measure $\min_{i\neq j} \sum_{l=1}^L (\lambda_{il} - \lambda_{jl})^2 / \left(\sum_{l=1}^L (\lambda_{il} - \lambda_{jl})^2\right)$, which averages the minimum eigengap across the matrices M_l (here, $\lambda_{il} = \pi_i(w_l^\top u_i)$).

A natural question to ask is along which vectors (w_l) should we project? In Section 4 and Section 5 we show that (a) estimates of the inverse factors (v_i) are a good choice (when the (v_i) are approximately orthogonal, they are close to the factors (u_i)) and that (b) random vectors do almost as well. This suggests a simple two-step method: (i) first, we find approximations of the tensor factors by simultaneously diagonalizing a small number of random projections of the tensor; (ii) then we perform another round of simultaneous diagonalization on projections along the inverse of these approximate factors. Algorithm 1 describes the approach. Its running time is $O(k^2 d^2 s)$, where s is the number of sweeps of the simultaneous diagonalization algorithm.

Algorithm 1 Two-stage tensor factorization algorithm

Require: $\widehat{T} = T + \epsilon R \in \mathbb{R}^{d\times d\times d}$, where T has a CP decomposition $T = \sum_{i=1}^k \pi_i u_i^{\otimes 3}$, $L_0 \geq 2$

Ensure: Estimates of factors, $\tilde{\pi}, \tilde{u}_1, \cdots, \tilde{u}_k$.

1: Define $\mathcal{M}^{(0)} \leftarrow \{\hat{T}(I,I,w_l)\}_{l=1}^{L_0}$ with $\{w_l\}_{l=1}^{L_0}$ are chosen uniformly from the unit sphere $\mathcal{S}^{d-1}$.
2: Obtain factors $\{\tilde{u}_i^{(0)}\}_{i=1}^k$ and their inverse $\{\tilde{v}_i^{(0)}\}_{i=1}^k$ from the simultaneous diagonalization of $\mathcal{M}^{(0)}$.
3: Define $\mathcal{M}^{(1)} \leftarrow \{\hat{T}(I,I,\tilde{v}_i^{(0)})\}_{i=1}^k$.
4: **return** Factors $\{\tilde{u}_i^{(1)}\}_{i=1}^k$ and factor weights $\{\tilde{\pi}_i\}_{i=1}^k$ from simultaneously diagonalizing $\mathcal{M}^{(1)}$.

4 Perturbation analysis for orthogonal tensor factorization

In this section, we will focus on the orthogonal setting, returning to non-orthogonal factors in Section 5. For ease of exposition, we restrict ourselves to symmetric third-order orthogonal tensors: $T = \sum_{i=1}^k \pi_i u_i^{\otimes 3}$. Here the inverse factors (v_i) are equivalent to the factors (u_i), and we do not distinguish between the two. The proofs for this section can be found in Appendix B.

Our sensitivity analysis builds on the perturbation analysis result for the simultaneous diagonalization of matrices in Cardoso [28].

Lemma 1 (Cardoso [28]). *Let $M_l = U\Lambda_l U^\top + \epsilon R_l$, $l \in [L]$, be matrices with common factors $U \in \mathbb{R}^{d\times k}$ and diagonal $\Lambda_l \in \mathbb{R}^{k\times k}$. Let $\bar{U} \in \mathbb{R}^{d\times d}$ be a full-rank extension of U with columns $u_1, u_2, \ldots, u_d$ and let $\tilde{U} \in \mathbb{R}^{d\times d}$ be the orthogonal minimizer of the joint diagonalization objective $F(\cdot)$. Then, for all u_j, $j \in [k]$, there exists a column $\tilde{u}_j$ of $\tilde{U}$ such that*

$$\|\tilde{u}_j - u_j\|_2 \leq \epsilon \sqrt{\sum_{i=1}^d E_{ij}^2} + o(\epsilon), \tag{5}$$

where $E \in \mathbb{R}^{d\times k}$ is

$$E_{ij} \triangleq \frac{\sum_{l=1}^L (\lambda_{il} - \lambda_{jl}) u_j^\top R_l u_i}{\sum_{l=1}^L (\lambda_{il} - \lambda_{jl})^2} \tag{6}$$

when $i \neq j$ and $i \leq k$ or $j \leq k$. We define $E_{ij} = 0$ when $i = j$ and $\lambda_{il} = 0$ when $i > k$.

In the tensor factorization setting, we jointly diagonalize projections $\widehat{M}_l$, $l = 1, 2, \ldots, L$ of the noisy

tensor $\widehat{T}$ along vectors w_l: $\widehat{M}_l = \widehat{T}(I, I, w_l) = \sum_{i=1}^k \pi_i (w_l^\top u_i) u_i^{\otimes 2} + \epsilon R(I, I, w_l)$, where $R_l \triangleq R(I, I, w_l)$ has unit operator norm. Cardoso's lemma provides bounds on the accuracy of recovering the u_i via joint diagonalization; in particular, we can further rewrite Equation 6 in the tensor setting as:

$$E_{ij} = \frac{\sum_{l=1}^L w_l^\top p_{ij} r_{ij}^\top w_l}{\sum_{l=1}^L w_l^\top p_{ij} p_{ij}^\top w_l}, \tag{7}$$

where $p_{ij} \triangleq (\pi_i u_i - \pi_j u_j)$ and $r_{ij} \triangleq R(u_i, u_j, I)$.

Equation 7 tells us that we can control the magnitude of the E_{ij} (and hence the error on recovering the u_i) through appropriate choice of the projections (w_l). Ideally, we would like to ensure that the projected eigengap, $\min_{i \neq j} w_l^\top p_{ij} = \min_{i \neq j} \left(\pi_i (w_l^\top u_i) - \pi_j (w_l^\top u_j)\right)$, is bounded away from zero for at least one M_l so that the denominator of Equation 7 does not blow up.

Random projections The first step of Algorithm 1 projects the tensor along random directions. The form of Equation 7 suggests that the error terms, E_{ij}, should concentrate over several projections and we will show that this is indeed the case. Consequently, the error terms will depend inversely on the mean of $w_l^\top p_{ij}$, $\|p_{ij}\|_2^2 = \pi_i^2 + \pi_j^2 > \pi_{\min}^2$. Our final result is as follows:

Theorem 1 (Tensor factorization with random projections). *Let $w_1, \ldots, w_L$ be i.i.d. Gaussian vectors, $w_l \sim \mathcal{N}(0, I)$, and let the matrices $\widehat{M}_l \in \mathbb{R}^{d \times d}$ be constructed via projection of $\widehat{T}$ along $w_1, \ldots, w_L$. Let $\tilde{u}_i$ be estimates of the u_i derived from the $\widehat{M}_l$. Let $L \geq 16 \log(2d(k-1)/\delta)^2$. Then, with probability at least $1 - \delta$, for every u_i, there exists a $\tilde{u}_i$ such that*

$$\|\tilde{u}_i - u_i\|_2 \leq \left(\frac{2\sqrt{2\|\pi\|_1 \pi_{\max}}}{\pi_i^2} + \frac{C(\delta)}{\pi_i} \right) \epsilon + o(\epsilon),$$

where $C(\delta) \triangleq O\left(\log(kd)/\delta\right) \sqrt{\frac{d}{L}}\Big)$.

The first of the above two terms is the fundamental error in estimating a noisy tensor $\hat{T}$; the second term is due to the concentration of random projections and can be made arbitrarily small by increasing L.

Plug-in projections The next step of our algorithm projects the tensor along the approximate factors from step 2. Intuitively, if the w_l are close to the eigenvectors u_i, then $w_l^\top p_{ij} = w_l^\top (\pi_i u_i - \pi_j u_j) \approx \pi_i \delta_{il}$. Then for each $i \neq j$, there is some projection that ensures that E_{ij} is bounded and does not depend on the projected eigengap $\min_{i \neq j} \left(\pi(w_l^\top u_i) - \pi(w_l^\top u_j)\right)$.

Theorem 2 (Tensor factorization with plug-in projections). *Let $w_1, \ldots, w_k$ be approximations of $u_1, \ldots, u_k$: $\|w_l - u_l\|_2 = O(\epsilon)$, and let $\widehat{M} \in \mathbb{R}^{d \times d}$ be constructed via projection of $\widehat{T}$ along $w_1, \ldots, w_k$. Let $\tilde{u}_i$ be estimates of the u_i derived from the $\widehat{M}_l$. Then, for every u_i, there exists a $\tilde{u}_i$ such that*

$$\|\tilde{u}_i - u_i\|_2 \leq \frac{2\sqrt{\|\pi\|_1 \pi_{\max}}}{\pi_i^2} \epsilon + o(\epsilon).$$

Note that Theorem 1 says that with $O(d)$ random projections, we can recover the eigenvectors u_i with almost the same precision as if we used approximate eigenvectors, with high probability. Moreover, as $L \to \infty$, there is no gap between the precision of the two methods. Theorem 2 on the other hand suggests that we can tolerate errors on the order of $O(\epsilon)$ without significantly affecting the error in recovering $\tilde{u}_i$. In practice, we find that using the plug-in estimates allows us to improve accuracy with fewer random projections.

5 Perturbation analysis for non-orthogonal tensor factorization

We now extend our results to the case when the tensor T has a non-orthogonal symmetric CP decomposition: $T = \sum_{i=1}^k \pi_i u_i^{\otimes 3}$, where the u_i are not orthogonal and $k \leq d$. We parameterize the non-orthogonality using incoherence: $\mu \triangleq \max_{i \neq j} u_i^\top u_j$ and the norm of the inverse factor $\|V^\top\|_2$ where $V \triangleq U^{-1}$. Compared to the orthogonal setting, our bounds reveal an $O\left(\frac{\|V^\top\|_2^2}{1-\mu^2}\right)$ dependence on incoherence. Note that unlike previous work, our algorithm does not require an explicit bound on μ (i.e. any $\mu < 1$ is sufficient), as long as the factors U are non-singular. Proofs for this section are found in Appendix C.

We base our analysis on the perturbation result by Afsari [24].

Lemma 2 (Afsari [24]). *Let $M_l = U \Lambda_l U^\top + \epsilon R_l$, $l \in [L]$, be matrices with common factors $U \in \mathbb{R}^{d \times k}$ and diagonal $\Lambda_l \in \mathbb{R}^{k \times k}$. Let $\bar{U} \in \mathbb{R}^{d \times d}$ be a full-rank extension of U with columns $u_1, u_2, \ldots, u_d$ and let $\bar{V} = \bar{U}^{-1}$, with rows $v_1, v_2, \ldots, v_d$. Let $\tilde{U} \in \mathbb{R}^{d \times d}$ be the minimizer of the joint diagonalization objective $F(\cdot)$ and let $\tilde{V} = \tilde{U}^{-1}$.*

Then, for all u_j, $j \in [k]$, there exists a column $\tilde{u}_j$ of $\tilde{U}$ such that

$$\|\tilde{u}_j - u_j\|_2 \leq \epsilon \sqrt{\sum_{i=1}^d E_{ij}^2} + o(\epsilon), \tag{8}$$

where the entries of $E \in \mathbb{R}^{d\times k}$ *are bounded by*

$$|E_{ij}| \le \frac{1}{1-\rho_{ij}^2}\left(\frac{1}{\|\lambda_i\|_2^2}+\frac{1}{\|\lambda_j\|_2^2}\right) \left(\left|\sum_{l=1}^{L} v_i^\top R_l v_j \lambda_{jl}\right| + \left|\sum_{l=1}^{L} v_i^\top R_l v_j \lambda_{il}\right|\right),$$

when $i \neq j$ *and* $E_{ij} = 0$ *when* $i = j$ *and* $\lambda_{il} = 0$ *when* $i > k$. *Here* $\lambda_i = (\lambda_{i1}, \lambda_{i2}, ..., \lambda_{iL}) \in \mathbb{R}^L$ *and* $\rho_{ij} = \frac{\lambda_i^\top \lambda_j}{\|\lambda_i\|_2\|\lambda_j\|_2}$ *is the modulus of uniqueness, a measure of how ill-conditioned the problem is.*

In the orthogonal case, we had a dependence on the eigengap $\lambda_i - \lambda_j$. Now the error crucially depends on the modulus of uniqueness, ρ_{ij}. The non-orthogonal simultaneous diagonalization problem has a unique solution iff $|\rho_{ij}| < 1$ for all $i \neq j$ [24]. In the orthogonal case, $\rho_{ij} = 0$. It can be shown that ρ_{ij} can once again be controlled by appropriately choosing the projections (w_l).

To get a handle on the difficulty of the problem, let us assume that the vectors u_i are incoherent: $u_i^\top u_j \le \mu$ for all $i \neq j$. Intuitively, the problem is easy when $\mu \approx 0$ and hard when $\mu \approx 1$.

Random projections Intuitively, random projections are isotropic and hence we expect the projections λ_i and λ_j to be nearly orthogonal to each other. This allows us to show that $\rho_{ij} \le O(\mu)$, which matches our intuitions on the difficulty of the problem. Our final result is the following:

Theorem 3 (Non-orthogonal tensor factorization with random projections). *Let* $w_1, \dots, w_L$ *be i.i.d. random Gaussian vectors,* $w_l \sim \mathcal{N}(0, I)$, *and let the matrices* $\widehat{M}_l \in \mathbb{R}^{d\times d}$ *be constructed via projection of* $\widehat{T}$ *along* $w_1, \dots, w_L$. *Assume incoherence* μ *on* (u_i): $u_i^\top u_j \le \mu$. *Let* $L_0 \triangleq \left(\frac{50}{1-\mu^2}\right)^2$ *and let* $L \ge L_0 \log(15d(k-1)/\delta)^2$. *Then, with probability at least* $1-\delta$, *for every* u_i, *there exists a* $\tilde{u}_i$ *such that*

$$\|\tilde{u}_j - u_j\|_2 \le O\left(\frac{\sqrt{\|\pi\|_1 \pi_{\max}}}{\pi_{\min}^2}\frac{\|V^\top\|_2^2}{1-\mu^2}\left(1 + C(\delta)\right)\right)\epsilon + o(\epsilon),$$

where $C(\delta) \triangleq \left(\log(kd/\delta)\sqrt{\frac{d}{L}}\right)$.

Once again, the error decomposes into a fundamental recovery error and a concentration term. Note that the error is sensitive to the smallest factor weight, $\pi_{\min}$. This dependence arises from the sensitivity of the non-orthogonal factorization method to the λ_i with the smallest norm and is unavoidable.

Plug-in projections When using plug-in estimates for the projections, two obvious choices arise: estimates of the columns of the factors, (u_i), or the rows of the inverse, (v_i). Using estimates of (u_i) leads to $\rho_{ij} \le O(\mu)$, similar to what we saw with random projections. However, using estimates of (v_i) ensures that the λ_i are nearly orthogonal, resulting in $\rho_{ij} \approx 0$! This leads to estimates that are less sensitive to the incoherence μ.

Theorem 4 (Non-orthogonal tensor factorization with plug-in projections). *Let* $w_1, \dots, w_k$ *be approximations of* $v_1, \dots, v_k$: $\|w_l - v_l\|_2 \le O(\epsilon)$, *and let the matrices* $\widehat{M}_l \in \mathbb{R}^{d\times d}$ *be constructed via projection of* $\widehat{T}$ *along* $w_1, \dots, w_k$. *Also assume that the* u_i *are incoherent:* $u_i^\top u_j \le \mu$ *when* $j \neq i$. *Then, for every* u_j, *there exists a* $\tilde{u}_j$ *such that*

$$\|\tilde{u}_j - u_j\|_2 \le O\left(\frac{\sqrt{\|\pi\|_1 \pi_{\max}}}{\pi_{\min}^2}\|V^\top\|_2^3\right)\epsilon + o(\epsilon).$$

6 Asymmetric and higher-order tensors

In this section, we present simple extensions to the algorithm to asymmetric and higher order tensors.

Asymmetric tensors We use a reduction to handle asymmetric tensors. Observe that the l-th projection M_l of an asymmetric tensor has the form $M_l = \sum_i \lambda_i u_{il} v_{il}^\top = U\Lambda_l V^\top$, for some diagonal (not necessarily positive) matrix Λ_l and common U, V, not necessarily orthogonal. For each M_l, define another matrix $N_l = \left(\begin{smallmatrix} 0 & M_l^\top \\ M_l & 0 \end{smallmatrix}\right)$ and observe that

$$\begin{bmatrix} 0 & M_l^\top \\ M_l & 0 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} V & V \\ U & -U \end{bmatrix}\begin{bmatrix} \Lambda_l & 0 \\ 0 & -\Lambda_l \end{bmatrix}\begin{bmatrix} V & V \\ U & -U \end{bmatrix}^\top.$$

The (N_l) are symmetric matrices with common (in general, non-orthogonal) factors. Therefore, they can be jointly diagonalized and from their components, we can recover the components of the (M_l). This reduction does not change the modulus of uniqueness of the problem: the factor weights remain unchanged.

Higher order tensors Finally, if we have a higher order (say fourth order) tensor $T = \sum_i \pi_i a_i \otimes b_i \otimes c_i \otimes d_i$ then we can first determine the a_i, b_i by projecting into matrices $T(I, I, w, u) = \sum_i \pi (w^\top c_i)(u^\top d_i) a_i \otimes b_i$, and then determine the c_i, d_i by projecting along the first two components. Our bounds only depend on the dimension of the matrices being simultaneously diagonalized, and thus this reduction does not introduce additional error. Intuitively, we should expect that additional modes of a tensor should provide more information and thus help estimation, not hurt it. However, note that as the tensor order increases, the noise in the tensor will presumably increase as well.

7 Convergence properties.

The convergence of our algorithm depends on the choice of joint diagonalization subroutine. Theoretically, the Jacobi method, the QRJ1D algorithm, and other algorithms are guaranteed to converge to a local minimum at a quadratic rate [27, 14, 29]. The question of global convergence is currently open [30, 25]. Empirically though, these algorithms have been found in the literature to converge reliably to global minima [27, 25, 30] and to corroborate this claim, we conducted a series of experiments [16].

We first examined convergence to global minima in the orthogonal setting. In 1000 trials of the Jacobi algorithm on random sets of matrices for various ϵ and $d = L = 15$, we found that the objective values formed a Gaussian distribution around ϵ (the best accuracy that can be achieved). Then, on each of our real crowdsourcing datasets, we ran our algorithm from 1000 random starting points; in every case, the algorithm converged to the same solution (unlike EM). This suggests that our diagonalization algorithm is not sensitive to local optima. To complement this empirical evidence, we also established that the Jacobi algorithm will converge to the global minimum when ϵ is sufficiently small and when the algorithm is initialized with the eigendecomposition of a single projection matrix [16].

We also performed similar experiments in the non-orthogonal setting using the QRJ1D algorithm. Unlike Jacobi, QRJ1D suffers from local optima, which is expected since the general CP decomposition problem is NP-hard. However, local optima appear to only affect matrices with bad incoherence values, and in several real world experiments (see below), non-orthogonal methods fared better their orthogonal counterparts.

8 Experiments

In the orthogonal setting, we compare our algorithms (**OJD0**, which uses random projections, and **OJD1** which uses with plug-in) with the tensor power method (**TPM**), alternating least squares (**ALS**), and with the method of de **Lathauwer** [23]. In the non-orthogonal setting, we compare de **Lathauwer**, alternating least squares (**ALS**), non-linear least squares (**NLS**), and our non-orthogonal methods (**NOJD0** and **NOJD1**).

Random versus plug-in projections We generated random tensors $T = \sum_{i=1}^{k} \pi u_i^{\otimes 3} + \epsilon R$ with Gaussian entries in π, R and u_i distributed uniformly in the sphere $\mathcal{S}^{d-1}$. In Figure 1, we plot the error $\sum_{i=1}^{k} \frac{1}{k} \|u_i - \tilde{u}_i\|_2$ (averaged over 1000 trials) of using L random projections (blue line), versus using L random projections followed by plug-in (green line). The accuracy of random projections tends to a limit that is immediately achieved by the plug-in projections, as predicted by our theory. In the orthogonal setting, plug-in reduces the total number of projected matrices L required to achieve the limiting error by three-fold (20 vs. 60 when $d = 10$). In the non-orthogonal setting, the difference between the two regimes is much smaller.

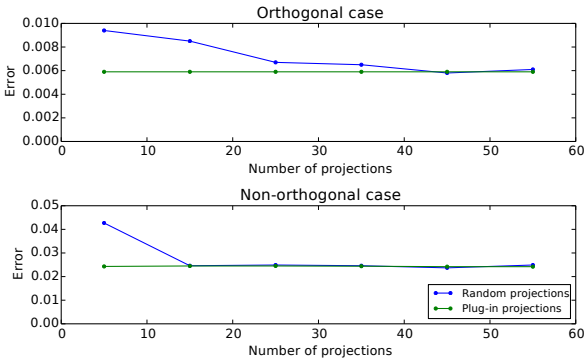

Figure 1: Comparing random vs. plug-in projections ($d = k = 10$, $\epsilon_{\text{ortho}} = 0.05$, $\epsilon_{\text{nonortho}} = 0.01$)

Synthetic accuracy experiments We generated random tensors for various d, k, ϵ using the same procedure as above. We vary ϵ and report the average error $\sum_{i=1}^{k} \frac{1}{k} \|u_i - \tilde{u}_i\|_2$ across 50 trials.

Our method realizes its full potential in the full-rank non-orthogonal setting, where **OJD0** and **OJD1** are up to three times more accurate than alternative methods (Figure 2, top). In the (arguably easier) undercomplete case, our methods do not achieve more than a 10% improvement, and overall, all algorithms fare similarly (Figure 4 in the supplementary material). Alternating least squares displayed very poor performance, and we omit it from our graphs.

In the full rank setting, there is little difference in performance between our method and **Lathauwer** (Figure 2, bottom). In both the full and low-rank cases (Figure 2, bottom and Figure 5 in the supplementary material), we consistently outperform the standard approaches, **ALS** and **NLS**, by 20–50%. Although we do not always outperform **Lathauwer** (a state-of-the-art method), **NOJD0** and **NOJD1** are faster and much simpler to implement.

We also tested our method on the single topic model from Section 2.2. For $d = 50$ and $k = 10$, over 50 trials in which model parameters were generated uniformly at random in $\mathcal{S}^{d-1}$, **OJD0** and **OJD1** obtained error rates of 0.05 and 0.055 respectively, followed by **TPM**

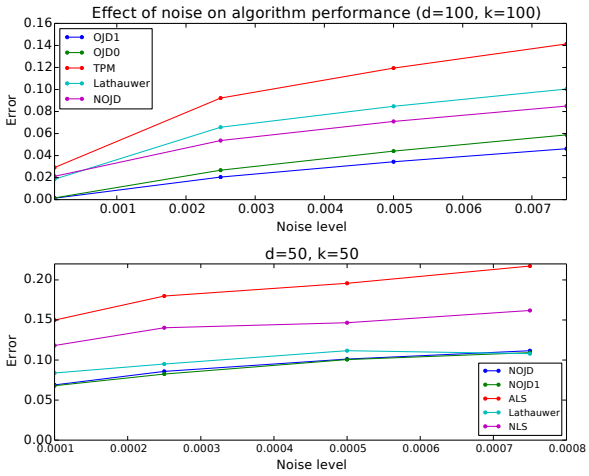

Figure 2: Performance on full-rank synthetic tensors.

(0.62 error), and **Lathauwer** (0.65 error). Additional experiments on asymmetric tensors and on running time are in the supplementary material.

Community detection in a social network Next, we use our method to detect communities in a real Facebook friend network at an American university [31] using a recently developed estimator based on the method of moments [4]. We reproduce a previously proposed methodology for assessing the performance of this estimator on our Facebook dataset [31]: ground truth communities are defined by the known dorm, major, and high school of each student; empirical and true community membership vectors $\hat{c}_i, c_i$ are matched using a similarity threshold $t > 0$; for a given threshold, we define the *recovery ratio* as the number of true c_i to which an empirical $\hat{c}_i$ is matched and we define the *accuracy* to be the average ℓ_1 norm distance between c_i and all the $\hat{c}_i$ that match to it. See [31] for more details. By varying $t > 0$, we obtain a tradeoff curve between the recovery ratio and accuracy (Figure 3). Our **OJD1** method determines the top 10 communities more accurately than **TPM**; finding smaller communities was equally challenging for both methods.

Label prediction from crowdsourcing data Lastly, we predict data labels within several datasets based on real-world crowdsourcing annotations using a recently proposed estimator based on the method of moments [17]. We incorporate our tensor factorization algorithms within the estimator and evaluate the approach on the same datasets as [17] except one, which we could not obtain. In addition to the previously defined methods, we also compare to the expectation maximization algorithm initialized with majority voting by the workers (**MV+EM**). We measure the label prediction accuracy. Overall, **NOJD1** outperforms all other tensor-based methods on three out of four datasets and results in accuracy gains of up to 1.75% (Table 2). **OJD1** outperforms the **TPM** on every dataset but one, and in two cases even outperforms **ALS** and **Lathauwer**, even though they are not affected by whitening. Most interestingly, on two datasets, at least one of our methods matches or outperforms the EM-based estimator.

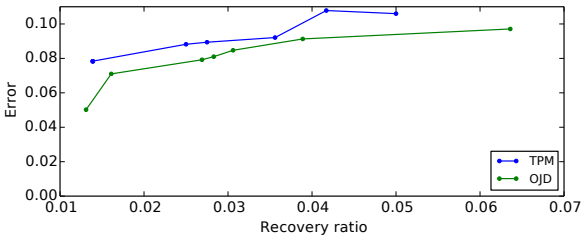

Figure 3: Accuracy/recovery tradeoff for community detection.

Table 2: Crowdsourcing experiment results

Dataset	Web	RTE	Birds	Dogs
TPM	82.25	88.75	87.96	84.01
OJD	82.33	90.00	**89.81**	84.01
NOJD	**83.49**	**90.50**	**89.81**	**84.26**
ALS	83.15	88.75	88.89	**84.26**
LATH	83.00	88.75	88.89	**84.26**
MV+EM	**83.68**	**92.75**	88.89	83.89
Size	2665	800	106	807

9 Discussion

We have presented a simple method for tensor factorization based on three ideas: simultaneous matrix diagonalization, random projections, and plugin estimates. Joint diagonalization methods for tensor factorization have been proposed in the past, but they have either been computationally too expensive [23] or numerically unstable [20]. We overcome both these limitations using multiple random projections of the tensor. Note that our use of random projections is atypical: instead of using projections for dimensionality reduction (e.g. [32]), we use it to reduce the *order* of the tensor. Finally, we improve estimates of the factors retrieved with random projections by using them as plugin estimates, a common technique in statistics to improve statistical efficiency [33]. Extensive experiments show that our factorization algorithm is more accurate than the state-of-the-art.

References

[1] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3): 455–500, 2009.

[2] M. Nickel, V. Tresp, and H. Kriegel. A three-way model for collective learning on multi-relational data. In *International Conference on Machine Learning (ICML)*, pages 809–816, 2011.

[3] A. Anandkumar, D. P. Foster, D. Hsu, S. M. Kakade, and Y. Liu. Two SVDs suffice: Spectral decompositions for probabilistic topic modeling and latent Dirichlet allocation. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[4] A. Anandkumar, R. Ge, D. Hsu, and S. Kakade. A tensor spectral approach to learning mixed membership community models. In *Conference on Learning Theory (COLT)*, pages 867–881, 2013.

[5] Y. Halpern and D. Sontag. Unsupervised learning of noisy-or Bayesian networks. In *Uncertainty in Artificial Intelligence (UAI)*, 2013.

[6] A. Chaganty and P. Liang. Estimating latent-variable graphical models using moments and likelihoods. In *International Conference on Machine Learning (ICML)*, 2014.

[7] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. Tensor decompositions for learning latent variable models. Technical report, arXiv, 2013.

[8] J. B. Kruskal. Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra and Applications*, 18:95–138, 1977.

[9] d. S. V and L. L. Tensor rank and the Ill-Posedness of the best Low-Rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30:1084–1127, 2008.

[10] J. Brachat, P. Comon, B. Mourrain, and E. Tsigaridas. Symmetric tensor decomposition. *Linear Algebra and its Applications*, 433(11):1851–1872, 2010.

[11] J. Hoastad. Tensor rank is NP-complete. *Journal of Algorithms*, 11(4), 1990.

[12] C. J. Hillar and L. Lim. Most tensor problems are NP-Hard. *Journal of the ACM (JACM)*, 60, 2013.

[13] A. Yeredor. Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation. *IEEE Transactions on Signal Processing*, 50(7):1545–1553, 2002.

[14] A. Ziehe, P. Laskov, G. Nolte, and K. Müller. A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation. *Journal of Machine Learning Research (JMLR)*, 5:777–800, 2004.

[15] R. Vollgraf and K. Obermayer. Quadratic optimization for simultaneous matrix diagonalization. *IEEE Transactions on Signal Processing*, 54(9): 3270–3278, 2006.

[16] V. Kuleshov, A. Chaganty, and P. Liang. Simultaneous diagonalization: the asymmetric, low-rank, and noisy settings. Technical report, arXiv, 2015.

[17] Y. Zhang, X. Chen, D. Zhou, and M. I. Jordan. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. Technical report, arXiv, 2014.

[18] U. Shalit and G. Chechik. Coordinate-descent for learning orthogonal matrices through givens rotations. In *International Conference on Machine Learning (ICML)*, 2014.

[19] A. Anandkumar, R. Ge, and M. Janzamin. Guaranteed non-orthogonal tensor decomposition via alternating rank-1 updates. Technical report, arXiv, 2014.

[20] A. Anandkumar, D. Hsu, and S. M. Kakade. A method of moments for mixture models and hidden Markov models. In *Conference on Learning Theory (COLT)*, 2012.

[21] S. A. Joint diagonalization: Is non-orthogonal always preferable to orthogonal? In *Computational Advances in Multi-Sensor Adaptive Processing*, pages 305–308, 2009.

[22] P. Comon, X. Luciani, and A. L. D. Almeida. Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics*, 23(7):393–405, 2009.

[23] L. D. Lathauwer. A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization. *SIAM Journal of Matrix Analysis and Applications*, 28(3):642–666, 2006.

[24] B. Afsari. Sensitivity analysis for the problem of matrix joint diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1148–1171, 2008.

[25] J. Cardoso and A. Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 17(1):161–164, 1996.

[26] B. Afsari. Simple LU and QR based non-orthogonal matrix joint diagonalization. In *In-*

dependent Component Analysis and Blind Signal Separation, pages 1–7, 2006.

[27] A. Bunse-Gerstner, R. Byers, and V. Mehrmann. Numerical methods for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 14(4):927–949, 1993.

[28] J. Cardoso. Perturbation of joint diagonalizers. Technical report, T'el'ecom Paris, 1994.

[29] A. Yeredor, A. Ziehe, and K. Müller. Approximate joint diagonalization using a natural gradient approach. *Independent Component Analysis and Blind Signal Separation*, 1:86–96, 2004.

[30] L. D. Lathauwer, B. D. Moor, and J. Vandewalle. Independent component analysis and (simultaneous) third-order tensor diagonalization. *Signal Processing, IEEE Transactions on*, 49(10):2262–2271, 2001.

[31] F. Huang, U. N. Niranjan, M. U. Hakeem, and A. Anandkumar. Fast detection of overlapping communities via online tensor methods. Technical report, arXiv, 2013.

[32] H. N, M. P, and T. J. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53:217–288, 2011.

[33] A. W. van der Vaart. *Asymptotic statistics*. Cambridge University Press, 1998.

[34] B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, 28(5):1302–1338, 2000.