Scaling Up Simultaneous Diagonalization

Volodymyr Kuleshov*
Dept. of Computer Science
Stanford University
kuleshov@stanford.edu

Arun Tejasvi Chaganty*
Dept. of Computer Science
Stanford University
chaganty@cs.stanford.edu

Percy Liang
Dept. of Computer Science
Stanford University
pliang@cs.stanford.edu

Abstract

Simultaneous matrix diagonalization is a key subroutine in many machine learning problems, including blind source separation and parameter estimation in latent variable models. Here, we extend joint diagonalization algorithms to low-rank and asymmetric matrices and also provide extensions to the perturbation analysis of these methods. Our results allow joint diagonalization to scale to larger problem sizes and to new domains; we give a survey of such applications and report improvements relative to the state-of-the-art on a latent variable learning task. We hope that our results will demonstrate the usefulness and versatility of joint diagonalization as a tool in optimization and machine learning.

1 Introduction

Simultaneous or joint diagonalization is a basic problem in numerical analysis, in which we are given a set of $L \geq 2$ matrices $\mathcal{M} = \{M_l\}_{l=1}^L$ of the form

$$M_l = U \Lambda_l U^T, \tag{1}$$

where $U \in \mathbb{R}^{d \times k}$ are factors common to all the matrices, and the diagonal $\Lambda_l \in \mathbb{R}^{k \times k}$ is specific to each matrix M_l. Our task consists in determining the unknown factors and weights from the matrices M_l. Unlike traditional single-matrix diagonalization, the U may be non-orthogonal (such factors are identifiable when $L \geq 2$; see Afsari [1]), and even when the U are orthogonal, simultaneously diagonalizing the entire set $\mathcal{M}$ is often more robust to noise than diagonalizing a single M_l.

In this paper, we extend existing joint diagonalization algorithms to low-rank ($k < d$) and asymmetric matrices; surprisingly, these basic settings have not been discussed in the literature. We also complement these results with a perturbation analysis that gives bounds on the accuracy of a solution for noisy matrices $M_l = U \Lambda_l U^T + \epsilon R_l$ as a function of $\epsilon > 0$ (the R_l being unit-norm matrices).

Our extensions enable joint diagonalization to scale to new settings and to larger problems. We survey several applications, which include blind-source separation [2] and latent variable estimation via tensor factorization [3]. We demonstrate the effectiveness of our extensions via numerical experiments on a tensor decomposition task and show that they are competitive with current state-of-the-art methods. Our results demonstrate that joint diagonalization can be a useful and versatile tool for designing optimization and machine learning algorithms.

Notation Simultaneous diagonalization can be *orthogonal* or *non-orthogonal*; in the former case, the U in Equation 1 are assumed to be orthogonal. We use $\otimes$ to denote the outer product: e.g. if $u, v, w \in \mathbb{R}^d$, then $u \otimes v \otimes w \in \mathbb{R}^{d \times d \times d}$. A (third-order) tensor of rank k is defined as $T = \sum_{i=1}^k \pi_i a_i \otimes b_i \otimes c_i$, where $a_i, b_i, c_i \in \mathbb{R}^d$ are factors and $\pi \in \mathbb{R}^k$ are their weights. Tensor-matrix multiplication $T(X, Y, Z)$ (for $X, Y, Z \in \mathbb{R}^{d \times d}$) is defined as $T(X, Y, Z)_{ijk} = \sum_{l=1}^d \sum_{m=1}^d \sum_{n=1}^d T_{lmn} X_{li} Y_{mj} Z_{nk}$. These definitions naturally extend to higher-order tensors.

* These authors contributed equally.

2 Algorithms and Extensions

Background We consider here simultaneous diagonalization algorithms that minimize the objective $F(V) = \sum_{l=1}^{L} \text{off}(V^{-1} M_l V^{-\top}) = \sum_{l=1}^{L} \sum_{i \neq j} (V^{-1} M_l V^{-\top})_{ij}^2$, which is the sum of squared off-diagonal elements. Two popular methods for minimizing F include the Jacobi method [11, 12] for the orthogonal case and the QRJ1D algorithm [13] for the non-orthogonal case. Both techniques iteratively construct V^{-1} via a product of simple matrices $V^{-1} = B_T \cdots B_2 B_1$. In the Jacobi algorithm, B_t is a *Givens* rotation [11] $G_{ij}(\theta) = \cos\theta(\Delta_{ii} + \Delta_{jj}) + \sin\theta(\Delta_{ij} - \Delta_{ji})$ for some angle θ, with Δ_{ij} being a matrix which is 1 in the (i, j)-th entry and 0 elsewhere. In the QRJ1D algorithm, B_t is $H_{ij}(a) = I + a\Delta_{ij}$ for some a. The a and θ are chosen via a closed-form formula such as to decrease F. Both methods proceed in a number of "sweeps", with $O(d^3 L)$ time per sweep.

Algorithm 1 Low-rank Jacobi

Require: symmetric matrices $(M_l)_{l=1}^L$
  Initialize factors: $U = I$
  while objective is decreasing **do**
    Let $\hat{\theta}$ be the minimizer of $F(G_{ij}(\theta))$
    for $i = 1, 2, ..., k$ **do**
      for $j = i + 1, i + 2, ..., d$ **do**
        Let $\hat{\theta}$ be the minimizer of $F(G_{ij}(\theta))$
        $U \leftarrow U G_{ij}(\hat{\theta})$
        $M_l \leftarrow G_{ij}(\hat{\theta})^T M_l G_{ij}(\hat{\theta})\ \forall l \in [L]$
        if $\sum_{l=1}^{L} |(M_l)_{jj}| > \sum_{l=1}^{L} |(M_l)_{ii}|$ **then**
          Flip columns i and j in U and the M_l

Orthogonal low-rank matrices Notice that the Jacobi algorithm applied to a single matrix M with k non-zero sorted eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_k$ would only need to zero out the entires of M associated with the first k rows and the first k columns, i.e. transform M into $U \left(\begin{smallmatrix} \Lambda & 0 \\ 0 & \times \end{smallmatrix}\right) U^T$. The first k columns of U will correspond to eigenvectors; the remaining columns will contain arbitrary numbers. Doing so requires kd Givens rotations per sweep: one for every row of the eigenvalue block, multiplied by one for every column in the matrix.

If the eigenvalues of the input matrix M are not sorted, then we can sort the diagonal of M after every sweep of Jacobi (while still performing only kd Givens rotations per sweep). When the algorithm terminates, all the k non-zero eigenvalues will find themselves in the top $k \times k$ corner; if that wasn't the case, then they would have been swapped out with another entry from outside that block. This idea is used in current single-matrix low-rank Jacobi algorithms [14]. When there are multiple matrices, the components j over which the rank is positive are ones for which $\sum_{l=1}^{L} |(\Lambda_l)_{jj}| > 0$. This suggests a natural extension of the above idea: choose Givens rotations that push mass on the sum of the absolute values of the matrix diagonals towards the upper left. This idea is implemented in Algorithm 1; although Algorithm 1 is only guaranteed to converge to local optima (rotations don't change the objective F, which otherwise decreases), we show via extensive experiments that it converges globally in practice.

Non-orthogonal low-rank matrices The QRJ1D algorithm has a very similar structure to Jacobi: over the course of a sweep, the matrices M_j are first multiplied by a Givens rotation, and then by a lower triangular matrix. In each case, a rotation only affects two columns and two rows of M_j.

We can similarly sort the diagonal entries of the matrices and zero-out only the top $k \times k$ square of the M_j. This leads to Algorithm 2. Note that both algorithms differ from standard Jacobi and QRJ1D only by column sorting.

Asymmetric matrices Suppose now that we have a set of $L \geq 2$ matrices $\mathcal{M} = \{M_l\}_{l=1}^L$ of the form $M_l = U \Lambda_l V^T$, where $U \in \mathbb{R}^{d_1 \times k}$ and $V \in \mathbb{R}^{d_2 \times k}$ are sets of common factors, possibly non-orthogonal. We propose here a general reduction to the symmetric case. For each M_l, define another matrix $N_l =$

Algorithm 2 Low-rank QRJ1D

Require: symmetric matrices $(M_l)_{l=1}^L$
  Initialize factors: $U = I$
  while objective is decreasing **do**
    for $i = 1, 2, ..., k$ **do**
      for $j = i + 1, i + 2, ..., d$ **do**
        Let $\hat{\theta}$ be the minimizer of $F(G_{ij}(\theta))$
        $U \leftarrow U G_{ij}(\hat{\theta})$
        $M_l \leftarrow G_{ij}(\hat{\theta})^T M_l G_{ij}(\hat{\theta})\ \forall l \in [L]$
        if $\sum_{l=1}^{L} |(M_l)_{jj}| > \sum_{l=1}^{L} |(M_l)_{ii}|$ **then**
          Flip columns i and j in U and the M_l
    for $i = 1, 2, ..., k$ **do**
      for $j = i + 1, i + 2, ..., d$ **do**
        Let $\hat{a}$ be the minimizer of $F(H_{ij}(a))$
        $U \leftarrow U H_{ij}(\hat{a})$
        $M_l \leftarrow H_{ij}(\hat{a})^T M_l H_{ij}(\hat{a})\ \forall l \in [L]$
        if $\sum_{l=1}^{L} |(M_l)_{jj}| > \sum_{l=1}^{L} |(M_l)_{ii}|$ **then**
          Flip columns i and j in U and the M_l

$\begin{pmatrix} 0 & M_l^\top \\ M_l & 0 \end{pmatrix}$ and observe that $\begin{pmatrix} 0 & M_l^\top \\ M_l & 0 \end{pmatrix} = \begin{pmatrix} V & V \\ U & -U \end{pmatrix} \begin{pmatrix} \Lambda_l & 0 \\ 0 & -\Lambda_l \end{pmatrix} \begin{pmatrix} V & V \\ U & -U \end{pmatrix}^\top$. The (N_l) are symmetric matrices with common (in general, non-orthogonal) factors; their diagonalization yields the U, V.

The above approach runs in $O(d_1 d_2^2)$ time (assuming $d_2 \geq d_1$), which is worse than the $O(d_1^2 d_2)$ time complexity of SVD for a single matrix. It remains to be seen if non-orthogonal joint diagonalization admits algorithms are as fast as ones for the ordinary SVD.

Perturbation analysis Given noisy matrices of the form $M_l = U\Lambda_l U^T + \epsilon R_l$ ($\epsilon > 0$ and R_l a unit norm matrix) we can bound the error between the perturbed and unperturbed minimizers of F as a function of ϵ. In brief, we show that for each true component vector u_j, there is a component $\tilde{u}_j$ of the perturbed minimizer of F such that $\|\tilde{u}_j - u_j\|_2 \leq \epsilon\sqrt{\sum_{i=1}^d E_{ij}^2} + o(\epsilon)$. In the orthogonal setting, $E_{ij} = \frac{\sum_{l=1}^L (\lambda_{il} - \lambda_{jl}) u_j^\top R_l u_i}{\sum_{l=1}^L (\lambda_{il} - \lambda_{jl})^2}$ depends on the average eigengap of the M_l. In the non-orthogonal setting, the expression depends on the ratios $\frac{\lambda_i}{\lambda_j}$ of the different eigenvalues. We formally state our bounds in Section B.

Convergence properties We next consider the question of convergence to minimizers of the objective F. Even in the full-rank setting, the global convergence of the simultaneous Jacobi method remains an open problem. [15, 12]. In practice, however, this method is well-known to behave as if it had global convergence [11, 12, 15]. In the appendix, we show numerically that our low-rank orthogonal method converges globally as well, while low-rank QRJ1D may sometimes get stuck in local optima. We complement these results with the formal guarantee that for sufficiently small $\epsilon > 0$, the solution of Algorithm 1 will satisfy our perturbation bounds in the orthogonal setting. Overall, these results can be used to derive theoretical guarantees for algorithms that use joint diagonalization as a subroutine; see [3] for an example.

3 Applications

We now survey several applications of joint diagonalization and explain how our algorithms may improve these strategies. We also hope to give the reader a sense of the usefulness and versatility of joint diagonalization in machine learning.

Independent component analysis In independent component analysis (ICA; [4]), we observe data $\{x_i\}_{i=1}^n$ generated from a model $x_i = As_i$, where the s_i are a set of unknown signals (e.g. n audio signals from speakers at a cocktail party) that were mixed by an unknown matrix A. Our goal is to recover A and the s_i from the x_i. A popular way of solving ICA is via the JADE algorithm, which involves simultaneously diagonalizing eigenmatrices of the fourth order *cumulant* tensor. See [5] for details. Our algorithms may improve the efficiency of ICA algorithms when there are significant differences between the number if microphones and audio signals (i.e. when A is rectangular).

Parameter estimation in general latent variable models Recent work [3] has shown that the connection between tensor factorization and joint diagonalization algorithms extends beyond ICA to a general tensor factorization algorithm that can be applied to learn several latent variable models, such as Gaussian mixture models, topic models, hidden Markov models, etc. [6].

For illustrative purposes, consider the single topic model [6], 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 $x_1, x_2, x_3 \in \{e_1, \ldots, e_d\}$, 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). The method of moments approach casts the parameter estimation problem as one of tensor factorization: define the empirical tensor $\hat{T} = \frac{1}{n}\sum_{i=1}^n x_1^{(i)} \otimes x_2^{(i)} \otimes x_3^{(i)} = \sum_{i=1}^k \pi_i u_i \otimes u_i \otimes u_i + \epsilon R$, where $\epsilon R \in \mathbb{R}^{d \times d \times d}$ is statistical noise that goes to zero as $n \to \infty$.

The factors of this tensor, $(\pi, \{u_i\}_{i=1}^k)$, and hence the parameters of the model, can be recovered using simultaneous diagonalization. Consider of determining the tensor decomposition of

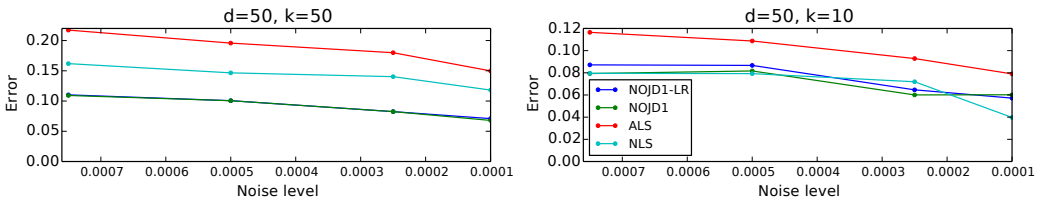

Figure 1: Low rank QRJ1D for tensor factorization compared to state-of-the-art methods.

the form $T = \sum_i \pi_i u_i^{\otimes 3}$. By projecting T along vectors $w_1, \ldots, w_L$, we can produce L matrices $T(I, I, w)_l = \sum_i \pi_i (w_l^\top u_i) u_i^{\otimes 2}$ that can be jointly diagonalized as in Equation 1, with factor U. [3] show that this method is less sensitive to perturbations in T than the popular tensor power method [6] and has natural extensions to non-orthogonal factors w_i. Our low-rank and asymmetric extensions enable the above method to be applies to asymmetric tensors as well as to large low-rank matrices.

Other applications Common component analysis (CCA; [7]) is a technique that generalizes PCA to K groups of data, each having a different covariance matrix Σ_k. We are interested in a set of factors U that explain well the variance in each group: i.e., we would like $\sum_{k=1}^K ||\Sigma_k - U \Lambda_k U^T||_F$ to be small. Such problems arise, for example, in econometrics, where we may observe data from k countries and would like to explain its variance using the same k interpretable factors. Our algorithms extend CCA to asymmetric and low-rank factors.

Other recent applications of joint diagonalization include new Bayesian algorithms for common spatial pattern analysis [8], a generalization of blind source separation to spatial data, kernel-based non-linear blind source separation [9], as well as applications in signal processing [10]. In each case, our methods can scale existing algorithms to larger low-rank matrices.

New applications Besides scaling existing algorithms to much larger matrices or tensors our techniques may also help solve new problem classes, such as kernelized extensions of blind source separation [9] or common principal components analysis. Such problems typically exhibit matrices of high dimension, but relatively low rank. Our methods could be extended to handle new types of problems, such common sparse SVD, where we would represent groups of data (e.g. images) in a common low-rank sparse basis.

4 Experiments

To evaluate our new methods, we performed two series of experiments. First, we examined their convergence properties: we ran each algorithm 1000 times on different sets of L random jointly diagonalizable matrices $U \Lambda_l U^T + \epsilon R$ (for $d = 15, L = k = 5$) corrupted with varying amounts of noise ϵ and plotted the histogram of final objective function values. In the orthogonal setting, we observed a tight distribution around ϵ, suggesting that Algorithm 1 converges to a global optima. In the non-orthogonal setting, we found that Algorithm 2 would occasionally get stuck in local optima. See Section A.1 for more details.

Next, we used Algorithm 2 as a subroutine for factorizing non-orthogonal tensors [3] (Figure 1). In the full-rank setting, Algorithm 2 performed identically to its full-rank counterpart; in the low-rank setting, it was only slightly less accurate due to a higher susceptibility to local optima. Most interestingly, our methods were competitive with alternating and non-linear least squares, two popular state-of-the-art tensor decomposition methods. Finally, in Section A.2, we performed a similar analysis for asymmetric tensors and again found our methods to be competitive with existing alternatives. This suggests our algorithm may improve existing latent variable estimation algorithms.

References

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

[2] 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.

[3] V. Kuleshov, A. Chaganty, and P. Liang. Tensor factorization via random projection and simultaneous matrix diagonalization. In *Artificial Intelligence and Statistics (AISTATS)*, 2015.

[4] A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Netw.*, 13(4-5):411–430, May 2000. ISSN 0893-6080. doi: 10.1016/S0893-6080(00)00026-5. URL http://dx.doi.org/10.1016/S0893-6080(00)00026-5.

[5] Jean-Franois Cardoso and Antoine Souloumiac. Blind beamforming for non gaussian signals. *IEE Proceedings-F*, 140:362–370, 1993.

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

[7] Bernhard N. Flury. Common principal components in k groups. *Journal of the American Statistical Association*, 79(388):pp. 892–898, 1984. ISSN 01621459. URL http://www.jstor.org/stable/2288721.

[8] Mingjun Zhong and Mark A. Girolami. A bayesian approach to approximate joint diagonalization of square matrices. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*, 2012. URL http://icml.cc/discuss/2012/357.html.

[9] S. Harmeling, A. Ziehe, M. Kawanabe, and K-R. Müller. Kernel-based nonlinear blind source separation. *Neural Computation*, 15(5):1089–1124, May 2003.

[10] Alle-Jan van der Veen, Michaela C. Vanderveen, and Arogyaswami Paulraj. Joint angle and delay estimation using shift-invariance techniques. *IEEE Transactions on Signal Processing*, 46(2):405–418, 1998. doi: 10.1109/78.655425. URL http://dx.doi.org/10.1109/78.655425.

[11] 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.

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

[13] B. Afsari. Simple LU and QR based non-orthogonal matrix joint diagonalization. In *Independent Component Analysis and Blind Signal Separation*, pages 1–7, 2006.

[14] D.-C. Xu, Z.-W. Liu, Y.-G. Xu, and J.-L. Cao. A sorted jacobi algorithm and its parallel implementation. *Beijing Ligong Daxue Xuebao/Transaction of Beijing Institute of Technology*, 30(12):1470–1474, 2010. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-78651418318&partnerID=40&md5=df84ea21195322b8417d61de43a9e5a7. cited By 0.

[15] 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.

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