Tensor Factorization via Matrix Factorization

(Kuleshov et al., 2015)

Amir Zakeri, Sebastien Henwood

March 24, 2020

Outline

1. Introduction
2. Tensor Factorization via Matrix Factorization(TFMF)
3. Simultaneous diagonalization
4. Experiments
5. Conclusion

Outline

1 Introduction

2 Tensor Factorization via Matrix Factorization(TFMF)

3 Simultaneous diagonalization

4 Experiments

5 Conclusion

Introduction

Given a tensor $\mathcal{T} \in \mathbb{R}^{d \times d \times d}$ with the following CP-decomposition:

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

our goal is to estimate the factors $\mathbf{a_i}, \mathbf{b_i}, \mathbf{c_i}$ and the factor weights $\pi \in \mathbb{R}^k$.

- To solve this, we saw ALS, gradient-based approaches in class
- **This presentation** $\Rightarrow$ Tensor Factorization via Matrix Factorization (TFMF)

The core idea

Projection $\mathcal{T}$ along a vector w to do eigendecomp. repeated L times

Introduction

Given a tensor $\mathcal{T} \in \mathbb{R}^{d \times d \times d}$ with the following CP-decomposition:

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

our goal is to estimate the factors $\mathbf{a_i}, \mathbf{b_i}, \mathbf{c_i}$ and the factor weights $\pi \in \mathbb{R}^k$.

- To solve this, we saw ALS, gradient-based approaches in class
- **This presentation** $\Rightarrow$ Tensor Factorization via Matrix Factorization (TFMF)

The core idea

Projection $\mathcal{T}$ along a vector w to do eigendecomp. repeated L times

Introduction

Given a tensor $\mathcal{T} \in \mathbb{R}^{d \times d \times d}$ with the following CP-decomposition:

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

our goal is to estimate the factors $\mathbf{a_i}, \mathbf{b_i}, \mathbf{c_i}$ and the factor weights $\pi \in \mathbb{R}^k$.

- To solve this, we saw ALS, gradient-based approaches in class
- **This presentation** $\Rightarrow$ Tensor Factorization via Matrix Factorization (TFMF)

The core idea

Projection $\mathcal{T}$ along a vector w to do eigendecomp. repeated L times

Introduction

Given a tensor $\mathcal{T} \in \mathbb{R}^{d \times d \times d}$ with the following CP-decomposition:

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

our goal is to estimate the factors $\mathbf{a_i}, \mathbf{b_i}, \mathbf{c_i}$ and the factor weights $\pi \in \mathbb{R}^k$.

- To solve this, we saw ALS, gradient-based approaches in class
- **This presentation** $\Rightarrow$ Tensor Factorization via Matrix Factorization (TFMF)

The core idea

Projection $\mathcal{T}$ along a vector w to do eigendecomp. repeated L times

Outline

1 Introduction

2 Tensor Factorization via Matrix Factorization(TFMF)

3 Simultaneous diagonalization

4 Experiments

5 Conclusion

Tensor Factorization via Matrix Factorization(TFMF)

TFMF algorithm overview

1. **Input** : L random vectors w, a tensor $\mathcal{T}$
2. Project $\mathcal{T}$ onto a set of random vectors w_L producing $\mathcal{M}$ matrices
3. Simultaneously diagonalize $\mathcal{M}$ producing CP decomp. factors estimates $\tilde{u}_{\mathcal{I}}$
4. Refine by repeating with the factor estimates instead of the random vectors
5. **Output** : CP factor matrices $\tilde{u}_{\mathcal{I}}$

- **Application**: to orthogonal, non-orthogonal and asymmetric tensors of arbitrary order.
- **Novelty**: Simultaneous matrix diagonalization.

Factors u ?

- When $a_i = b_i = c_i = u_i \; \forall i \Rightarrow$ symmetric factorization ! We have :

$$\sum_i \pi_i u_i^{\otimes 3} \tag{1}$$

- Project along a vector w !

$$\sum_i \pi_i (w^T u_i) u_i^{\otimes 2} \tag{2}$$

- Estimate u_i by eigendecomposition of Eq. 2 $\Rightarrow \tilde{u}_i$
- The error $||u_i - \tilde{u}_i||_2$ is *noise sensitive*
- Sensitivity $\approx$ smallest diff. between eigenvalues : the *eigengap*

$$\max_{j \neq i} \frac{1}{|\lambda_i - \lambda_j|}$$

Factors u ?

- When $a_i = b_i = c_i = u_i \; \forall i \Rightarrow$ symmetric factorization ! We have :

$$\sum_i \pi_i u_i^{\otimes 3} \tag{1}$$

- Project along a vector w !

$$\sum_i \pi_i (w^T u_i) u_i^{\otimes 2} \tag{2}$$

- Estimate u_i by eigendecomposition of Eq. 2 $\Rightarrow \tilde{u}_i$
- The error $||u_i - \tilde{u}_i||_2$ is *noise sensitive*
- Sensitivity $\approx$ smallest diff. between eigenvalues : the *eigengap*

$$\max_{j \neq i} \frac{1}{|\lambda_i - \lambda_j|}$$

Factors u ?

- When $a_i = b_i = c_i = u_i \; \forall i \Rightarrow$ symmetric factorization ! We have :

$$\sum_i \pi_i u_i^{\otimes 3} \tag{1}$$

- Project along a vector w !

$$\sum_i \pi_i (w^T u_i) u_i^{\otimes 2} \tag{2}$$

- Estimate u_i by eigendecomposition of Eq. 2 $\Rightarrow \tilde{u}_i$
- The error $||u_i - \tilde{u}_i||_2$ is *noise sensitive*
- Sensitivity $\approx$ smallest diff. between eigenvalues : the *eigengap*

$$\max_{j \neq i} \frac{1}{|\lambda_i - \lambda_j|}$$

Factors u ?

- When $a_i = b_i = c_i = u_i \; \forall i \Rightarrow$ symmetric factorization ! We have :

$$\sum_i \pi_i u_i^{\otimes 3} \tag{1}$$

- Project along a vector w !

$$\sum_i \pi_i (w^T u_i) u_i^{\otimes 2} \tag{2}$$

- Estimate u_i by eigendecomposition of Eq. 2 $\Rightarrow \tilde{u}_i$
- The error $||u_i - \tilde{u}_i||_2$ is *noise sensitive*
- Sensitivity $\approx$ smallest diff. between eigenvalues : the *eigengap*

$$\max_{j \neq i} \frac{1}{|\lambda_i - \lambda_j|}$$

Factors u ?

- When $a_i = b_i = c_i = u_i \; \forall i \Rightarrow$ symmetric factorization ! We have :

$$\sum_i \pi_i u_i^{\otimes 3} \tag{1}$$

- Project along a vector w !

$$\sum_i \pi_i (w^T u_i) u_i^{\otimes 2} \tag{2}$$

- Estimate u_i by eigendecomposition of Eq. 2 $\Rightarrow \tilde{u}_i$
- The error $||u_i - \tilde{u}_i||_2$ is *noise sensitive*
- Sensitivity $\approx$ smallest diff. between eigenvalues : the *eigengap*

$$\max_{j \neq i} \frac{1}{|\lambda_i - \lambda_j|}$$

Solving the *eigengap* with multiple projections (orth. case)

- Using *L random* projections we have the matrices M_ℓ

$$\sum_i \pi_i (w_\ell^T u_i) u_i^{\otimes 2} \tag{3}$$

- The set of matrices M_ℓ has common eigenvectors $\Rightarrow$ simultaneous diagonalization !
- The error bound then follows

$$||\tilde{u}_i - u_i||_2 \leq \left(\frac{2\sqrt{2||\pi||_1 \pi_{max}}}{\pi_i^2} + \frac{C(\delta)}{\pi_i}\right)\epsilon + o(\epsilon) \tag{4}$$

with $C(\delta) = \mathcal{O}(\log(kd/\delta\sqrt{\frac{d}{L}})$

$\Rightarrow$ The bigger L the lower the error bound !

Solving the *eigengap* with multiple projections (orth. case)

- Using *L random* projections we have the matrices M_ℓ

$$\sum_i \pi_i (w_\ell^T u_i) u_i^{\otimes 2} \tag{3}$$

- The set of matrices M_ℓ has common eigenvectors $\Rightarrow$ simultaneous diagonalization !
- The error bound then follows

$$||\tilde{u}_i - u_i||_2 \leq \left(\frac{2\sqrt{2||\pi||_1 \pi_{max}}}{\pi_i^2} + \frac{C(\delta)}{\pi_i}\right)\epsilon + o(\epsilon) \tag{4}$$

with $C(\delta) = \mathcal{O}(\log(kd/\delta\sqrt{\frac{d}{L}})$

$\Rightarrow$ The bigger L the lower the error bound !

Solving the *eigengap* with multiple projections (orth. case)

- Using L *random* projections we have the matrices M_ℓ

$$\sum_i \pi_i (w_\ell^T u_i) u_i^{\otimes 2} \tag{3}$$

- The set of matrices M_ℓ has common eigenvectors $\Rightarrow$ simultaneous diagonalization !
- The error bound then follows

$$||\tilde{u}_i - u_i||_2 \leq \left(\frac{2\sqrt{2||\pi||_1 \pi_{max}}}{\pi_i^2} + \frac{C(\delta)}{\pi_i}\right)\epsilon + o(\epsilon) \tag{4}$$

with $C(\delta) = \mathcal{O}(\log(kd/\delta\sqrt{\frac{d}{L}})$

$\Rightarrow$ The bigger L the lower the error bound !

Using estimates instead of random W

- After a first pass w/ random W the paper proposes to use $\tilde{u}$ as the projection
- The error bound then becomes

$$||\tilde{u}_i - u_i||_2 \leq \frac{2\sqrt{||\pi||_1 \pi_{max}}}{\pi_i^2}\epsilon + o(\epsilon) \tag{5}$$

$\Rightarrow$ same as prev. slide when $L \to \infty$

What about non orth. tensors ?

The papers extends this analysis with a new coef. > 1 (*spoiler*: the error bound grows)

Using estimates instead of random W

- After a first pass w/ random W the paper proposes to use $\tilde{u}$ as the projection
- The error bound then becomes

$$||\tilde{u}_i - u_i||_2 \leq \frac{2\sqrt{||\pi||_1 \pi_{max}}}{\pi_i^2}\epsilon + o(\epsilon) \tag{5}$$

$\Rightarrow$ same as prev. slide when $L \to \infty$

What about non orth. tensors ?

The papers extends this analysis with a new coef. > 1 (*spoiler*: the error bound grows)

Using estimates instead of random W

- After a first pass w/ random W the paper proposes to use $\tilde{u}$ as the projection
- The error bound then becomes

$$||\tilde{u}_i - u_i||_2 \leq \frac{2\sqrt{||\pi||_1 \pi_{max}}}{\pi_i^2}\epsilon + o(\epsilon) \tag{5}$$

$\Rightarrow$ same as prev. slide when $L \to \infty$

What about non orth. tensors ?

The papers extends this analysis with a new coef. > 1 (*spoiler*: the error bound grows)

Outline

1 Introduction

2 Tensor Factorization via Matrix Factorization(TFMF)

3 Simultaneous diagonalization

4 Experiments

5 Conclusion

Simultaneous diagonalization

- Symmetric matrices $\mathbf{M_1}, \cdots, \mathbf{M_L} \in \mathbb{R}^{d \times d}$ as:

$$\mathbf{M_l} = \mathbf{U}\Lambda_l\mathbf{U}^T + \epsilon\mathbf{R_l}.$$

- $U \in \mathbb{R}^{d \times k}$ is common, $\Lambda_l \in \mathbb{R}^{k \times k}$ and ϵR_l are individual.

- **Goal**: find inverse factors $\mathbf{V^{-1}} \in \mathbb{R}^{d \times d}$ such that $\mathbf{V^{-1}M_lV^{-T}}$ is nearly diagonal.
- Optimizing objective function to find $\mathbf{V}$:

$$F(\mathbf{X}) \triangleq \sum_{l=1}^{L} \text{off}(\mathbf{X^{-1}M_lX^{-T}}), \qquad \text{off}(\mathbf{A}) = \sum_{i \neq j} \mathbf{A}_{ij}^2.$$

  $\Rightarrow$ this penalizes the off-diagonal terms!

- Use Jacobi & QRJ1D

Simultaneous diagonalization

- Symmetric matrices $\mathbf{M_1}, \cdots, \mathbf{M_L} \in \mathbb{R}^{d \times d}$ as:

$$\mathbf{M_l} = \mathbf{U}\Lambda_l\mathbf{U}^T + \epsilon\mathbf{R_l}.$$

- $U \in \mathbb{R}^{d \times k}$ is common, $\Lambda_l \in \mathbb{R}^{k \times k}$ and ϵR_l are individual.

- **Goal**: find inverse factors $\mathbf{V^{-1}} \in \mathbb{R}^{d \times d}$ such that $\mathbf{V^{-1}M_lV}^{-T}$ is nearly diagonal.

- Optimizing objective function to find $\mathbf{V}$:

$$F(\mathbf{X}) \triangleq \sum_{l=1}^{L} \text{off}(\mathbf{X}^{-1}\mathbf{M_l}\mathbf{X}^{-T}), \qquad \text{off}(\mathbf{A}) = \sum_{i \neq j} \mathbf{A}_{ij}^2.$$

  $\Rightarrow$ this penalizes the off-diagonal terms!

- Use Jacobi & QRJ1D

Simultaneous diagonalization

- Symmetric matrices $\mathbf{M_1}, \cdots, \mathbf{M_L} \in \mathbb{R}^{d \times d}$ as:

$$\mathbf{M_l} = \mathbf{U}\Lambda_l\mathbf{U}^T + \epsilon\mathbf{R_l}.$$

- $U \in \mathbb{R}^{d \times k}$ is common, $\Lambda_l \in \mathbb{R}^{k \times k}$ and ϵR_l are individual.

- **Goal**: find inverse factors $\mathbf{V^{-1}} \in \mathbb{R}^{d \times d}$ such that $\mathbf{V^{-1}M_lV}^{-T}$ is nearly diagonal.
- Optimizing objective function to find $\mathbf{V}$:

$$F(\mathbf{X}) \triangleq \sum_{l=1}^{L} \text{off}(\mathbf{X}^{-1}\mathbf{M_l}\mathbf{X}^{-T}), \qquad \text{off}(\mathbf{A}) = \sum_{i \neq j} \mathbf{A}_{ij}^2.$$

  $\Rightarrow$ this penalizes the off-diagonal terms!

- Use Jacobi & QRJ1D

Simultaneous diagonalization

- Symmetric matrices $\mathbf{M_1}, \cdots, \mathbf{M_L} \in \mathbb{R}^{d \times d}$ as:

$$\mathbf{M_l} = \mathbf{U}\Lambda_l\mathbf{U}^T + \epsilon\mathbf{R_l}.$$

- $U \in \mathbb{R}^{d \times k}$ is common, $\Lambda_l \in \mathbb{R}^{k \times k}$ and ϵR_l are individual.

- **Goal**: find inverse factors $\mathbf{V^{-1}} \in \mathbb{R}^{d \times d}$ such that $\mathbf{V^{-1}M_lV}^{-T}$ is nearly diagonal.
- Optimizing objective function to find $\mathbf{V}$:

$$F(\mathbf{X}) \triangleq \sum_{l=1}^{L} \text{off}(\mathbf{X}^{-1}\mathbf{M_l}\mathbf{X}^{-T}), \qquad \text{off}(\mathbf{A}) = \sum_{i \neq j} \mathbf{A}_{ij}^2.$$

  $\Rightarrow$ this penalizes the off-diagonal terms!

- Use Jacobi & QRJ1D

For asymmetric and higher-order tensors

Asymmetric tensors:

- The l-th projection($\mathbf{M_l}$) of an asymmetric tensor has the following form:

$$\mathbf{M_l} = \sum_i \lambda_i \mathbf{u_{il}} \mathbf{v_{il}}^T = \mathbf{U} \Lambda_l \mathbf{V}^T,$$

  where Λ_l is diagonal but not necessarily positive matrix, and $\mathbf{U}, \mathbf{V}$ are common but not necessarily orthogonal.

- for each $\mathbf{M_l}$ we define another matrix $\mathbf{N_l}$ as:

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

- The $\mathbf{N_l}$ are symmetric matrices with common (in general, non-orthogonal) factors.

For asymmetric and higher-order tensors

Asymmetric tensors:

- The l-th projection($\mathbf{M_l}$) of an asymmetric tensor has the following form:

$$\mathbf{M_l} = \sum_i \lambda_i \mathbf{u_{il}} \mathbf{v_{il}}^T = \mathbf{U} \Lambda_l \mathbf{V}^T,$$

  where Λ_l is diagonal but not necessarily positive matrix, and $\mathbf{U}, \mathbf{V}$ are common but not necessarily orthogonal.

- for each $\mathbf{M_l}$ we define another matrix $\mathbf{N_l}$ as:

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

- The $\mathbf{N_l}$ are symmetric matrices with common (in general, non-orthogonal) factors.

For asymmetric and higher-order tensors

Asymmetric tensors:

- The l-th projection($\mathbf{M_l}$) of an asymmetric tensor has the following form:

$$\mathbf{M_l} = \sum_i \lambda_i \mathbf{u_{il}} \mathbf{v_{il}}^T = \mathbf{U} \Lambda_l \mathbf{V}^T,$$

  where Λ_l is diagonal but not necessarily positive matrix, and $\mathbf{U}, \mathbf{V}$ are common but not necessarily orthogonal.

- for each $\mathbf{M_l}$ we define another matrix $\mathbf{N_l}$ as:

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

- The $\mathbf{N_l}$ are symmetric matrices with common (in general, non-orthogonal) factors.

For asymmetric and higher-order tensors

Higher order tensors:

- For higher order tensor (say fourth order):

$$\mathcal{T} = \sum_i \pi_i \mathbf{a_i} \otimes \mathbf{b_i} \otimes \mathbf{c_i} \otimes \mathbf{d_i},$$

- We first determine $\mathbf{a_i}, \mathbf{b_i}$ by projecting into matrices:

$$\mathbf{T} = \sum_i \pi (\omega^T \mathbf{c_i})(u^T \mathbf{d_i}) \mathbf{a_i} \otimes \mathbf{b_i},$$

- Then determine $\mathbf{c_i}, \mathbf{d_i}$ by projecting along the first two components.

For asymmetric and higher-order tensors

Higher order tensors:

- For higher order tensor (say fourth order):

$$\mathcal{T} = \sum_i \pi_i \mathbf{a_i} \otimes \mathbf{b_i} \otimes \mathbf{c_i} \otimes \mathbf{d_i},$$

- We first determine $\mathbf{a_i}, \mathbf{b_i}$ by projecting into matrices:

$$\mathbf{T} = \sum_i \pi (\omega^T \mathbf{c_i})(u^T \mathbf{d_i}) \mathbf{a_i} \otimes \mathbf{b_i},$$

- Then determine $\mathbf{c_i}, \mathbf{d_i}$ by projecting along the first two components.

For asymmetric and higher-order tensors

Higher order tensors:

- For higher order tensor (say fourth order):

$$\mathcal{T} = \sum_i \pi_i \mathbf{a_i} \otimes \mathbf{b_i} \otimes \mathbf{c_i} \otimes \mathbf{d_i},$$

- We first determine $\mathbf{a_i}, \mathbf{b_i}$ by projecting into matrices:

$$\mathbf{T} = \sum_i \pi (\omega^T \mathbf{c_i})(u^T \mathbf{d_i}) \mathbf{a_i} \otimes \mathbf{b_i},$$

- Then determine $\mathbf{c_i}, \mathbf{d_i}$ by projecting along the first two components.

Convergence properties

- Convergence depends on the choice of joint diagonalization subroutine.
- Theoretically:
  - Convergence to Local minimum at a quadratic rate guaranteed.
  - Convergence to Global minimum is an open question!
- Empirically, convergence to global minima achieved.

Convergence properties

- Convergence depends on the choice of joint diagonalization subroutine.
- Theoretically:
  - Convergence to Local minimum at a quadratic rate guaranteed.
  - Convergence to Global minimum is an open question!
- Empirically, convergence to global minima achieved.

Convergence properties

- Convergence depends on the choice of joint diagonalization subroutine.
- Theoretically:
  - Convergence to Local minimum at a quadratic rate guaranteed.
  - Convergence to Global minimum is an open question!
- Empirically, convergence to global minima achieved.

Convergence properties

- Convergence depends on the choice of joint diagonalization subroutine.
- Theoretically:
  - Convergence to Local minimum at a quadratic rate guaranteed.
  - Convergence to Global minimum is an open question!
- Empirically, convergence to global minima achieved.

Convergence properties

- Convergence depends on the choice of joint diagonalization subroutine.
- Theoretically:
  - Convergence to Local minimum at a quadratic rate guaranteed.
  - Convergence to Global minimum is an open question!
- Empirically, convergence to global minima achieved.

Outline

1 Introduction

2 Tensor Factorization via Matrix Factorization(TFMF)

3 Simultaneous diagonalization

4 Experiments

5 Conclusion

Experiments

- Examining convergence to global minima in orthogonal setting (Jacobi Algo.): $\Rightarrow$ Using 1000 random starting points, getting the same solution!

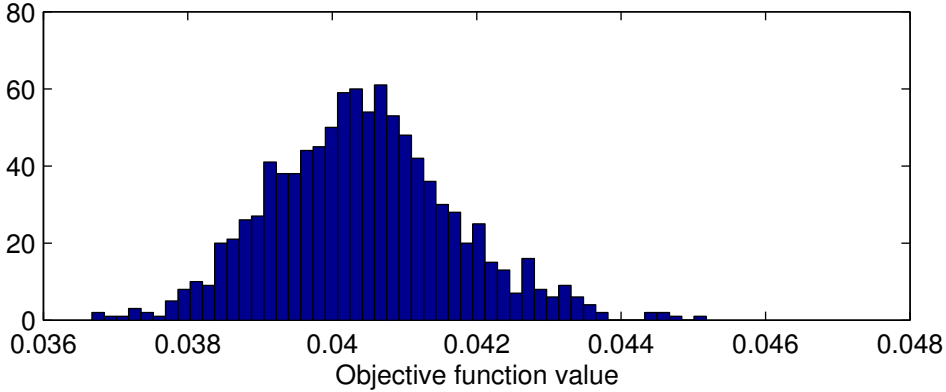


Figure 1: Histogram of objective function values, in orthogonal setting

Experiments

- Plotting histogram for different ϵ values ($\epsilon = 0, \epsilon = 1e-4, \epsilon = 1e-3$)

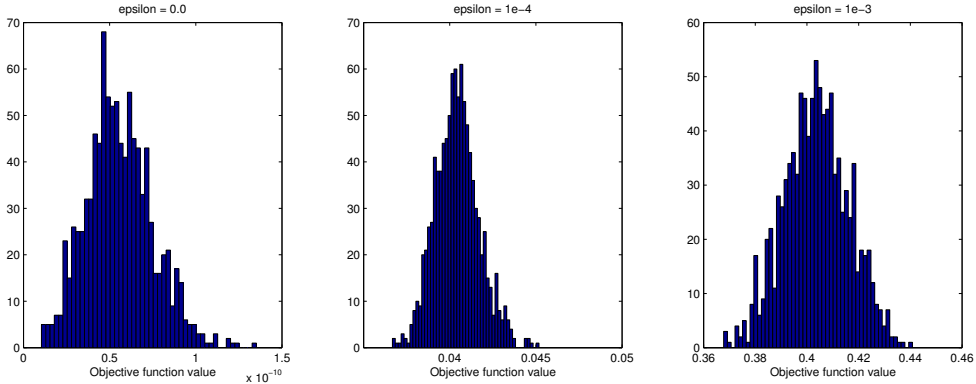


Figure 2: Comparing Histograms for different ϵ sizes, in orthogonal setting

- For small enough ϵ convergence is guaranteed.

Experiments

- Examning convergence to global minimum in Non-orthogonal setting:

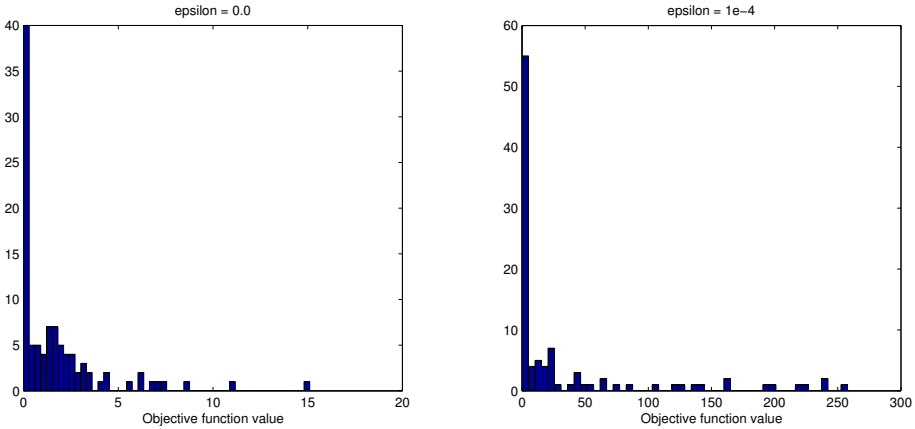


Figure 3: Histograms when μ is big

Experiments

- Examning convergence to global minimum in Non-orthogonal setting,(for small μ)

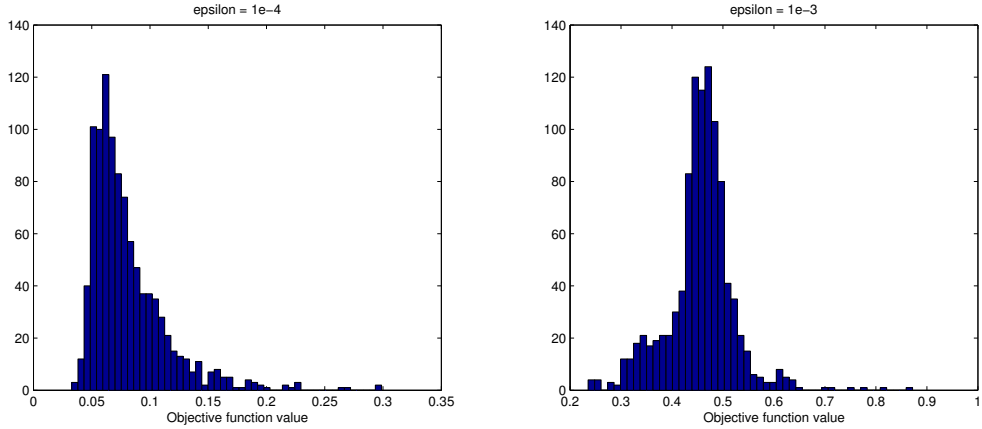
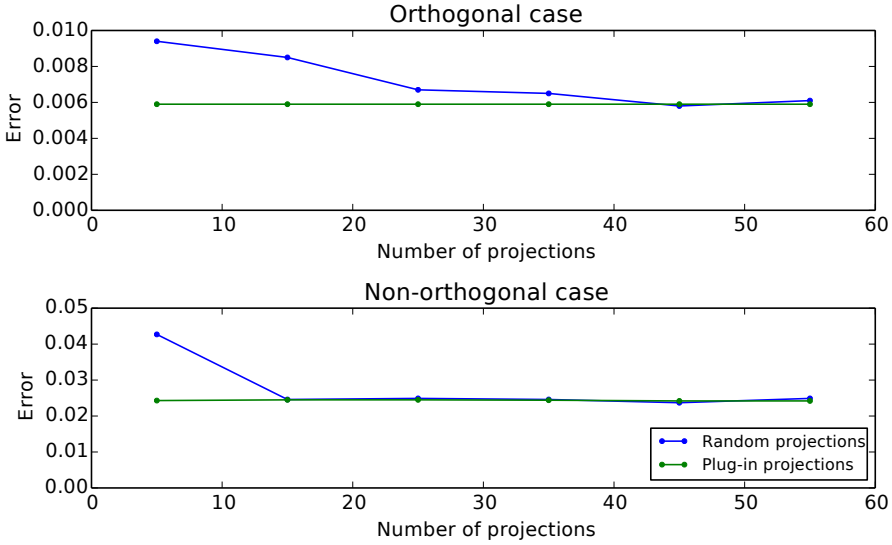


Figure 4: Histogram when μ is small

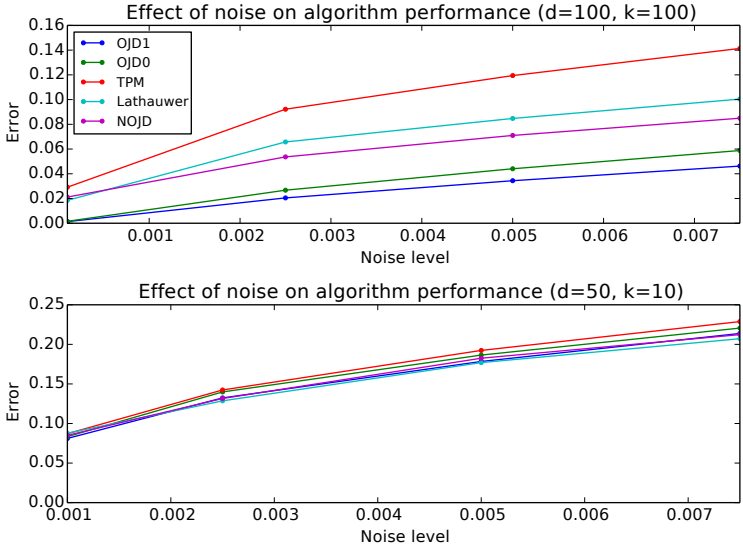
Experiments

- Comparing random vs. plugin projection



Experiments

- Performance comparison:



Outline

1 Introduction

2 Tensor Factorization via Matrix Factorization(TFMF)

3 Simultaneous diagonalization

4 Experiments

5 Conclusion

Conclusion

- TFMF, another take on CP decomposition
- TFMF = random projections + simultaneous diagonalization + plugin estimates
- Works for orthogonal, non-orthogonal, symmetric, asymetric, high order tensors
- Is more accurate that state-of-the-art.