CS 59300 – Algorithms for Data Science

Classical and Quantum approaches

Lecture 2 (09/02)

Tensor Methods (II)

https://ruizhezhang.com/course_fall_2025.html

Slides are based on Sitan Chen's lecture

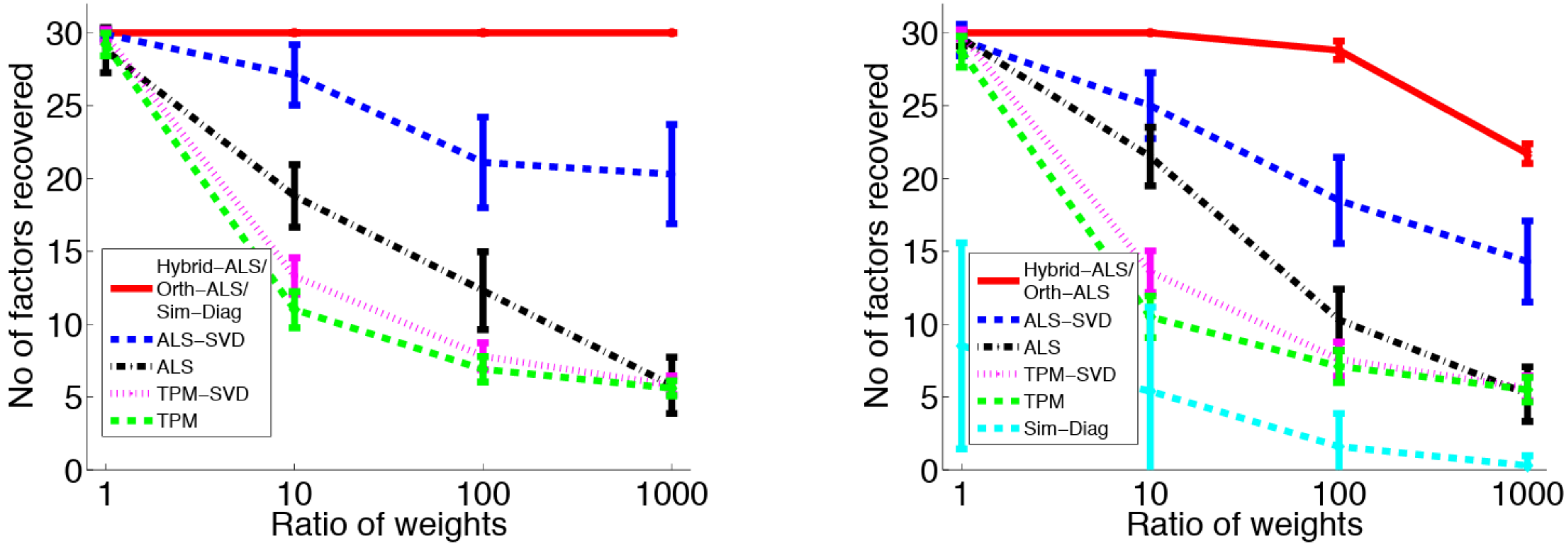
Jennrich's algorithm has good theoretical properties

- In noiseless setting, it is guaranteed to exactly recover the factors if they satisfy the conditions (1-3)

- If the tensor T has small noise $T' = T + E$, we can prove that Jennrich's algorithm is numerically stable (i.e., the output error is $\propto ||E||$)

The ugly truth: it's not a good idea to run Jennrich's algorithm in practice

Jennrich's algorithm is not very noise robust



(a) Noiseless case, ratio of weights equals $\frac{w_{\max}}{w_{\min}}$ (b) Noisy case, ratio of weights equals $\frac{w_{\max}}{w_{\min}}$

(Sharan-Valiant, 2017)

Jennrich's algorithm is not computationally efficient

Bottleneck steps of Jennrich's algorithm:

- Set

$$M_a \coloneqq \sum_{i\in[d]} a_i T(:,:,i) \quad \text{and} \quad M_b \coloneqq \sum_{i\in[d]} b_i T(:,:,i) \qquad \mathcal{O}(d^3) \longrightarrow \mathcal{O}(d^2)$$

- Compute $A \coloneqq M_a M_b^+$ and $B \coloneqq (M_a^+ M_b)^\top$
- Let $\hat{u}_1, \dots, \hat{u}_k$ be eigenvectors of A with eigenvalues $\lambda_1, \dots, \lambda_k$

Suppose $T = \mathbb{E}\left[x^{\otimes 3}\right]$. Then, $M_a = \mathbb{E}[\langle a, x\rangle x x^\top]$ is computable in $\mathcal{O}(d^2)$ time

Jennrich's algorithm is not computationally efficient

Bottleneck steps of Jennrich's algorithm:

- Set

$$M_a \coloneqq \sum_{i\in[d]} a_i T(:,:,i) \quad \text{and} \quad M_b \coloneqq \sum_{i\in[d]} b_i T(:,:,i)$$

$\mathcal{O}(d^3) \longrightarrow \mathcal{O}(d^2)$

- Compute $A \coloneqq M_a M_b^+$ and $B \coloneqq (M_a^+ M_b)^\top$
- Let $\hat{u}_1, \dots, \hat{u}_k$ be eigenvectors of A with eigenvalues $\lambda_1, \dots, \lambda_k$

$\mathcal{O}(d^\omega)$ or $\mathcal{O}(d^3)$

$\omega \approx 2.371$ is the fast matrix multiplication exponent (Alman et al. 2024)

The bottleneck comes from dense matrix operations

Today's plan

We'll explore the iterative methods that heuristic tensor decomposition algorithms build upon:

- Gradient descent
- Power iteration
- Alternating minimization

For simplicity, let's assume T is a symmetric 3-tensor:

$$T = \sum_{i \in [k]} \lambda_i u_i \otimes u_i \otimes u_i$$

where $u_i \in \mathbb{R}^d$ are orthonormal vectors

At the end of this lecture, we'll see how to remove the orthogonality assumption

Gradient descent

Consider the following polynomial optimization problem:

$$\max_{\|x\|=1} p(x) \coloneqq \sum_{a,b,c} T_{abc}\, x_a x_b x_c = T(x,x,x) = \sum_i \lambda_i \, \langle u_i, x\rangle^3$$

- We assume $\{u_i\}$ are orthonormal
- In this case, you can show that $\{u_i\}$ are exactly the local maximizers of $p(x)$ over $\mathbb{S}^{d-1}$
  - → When $x \approx u_i$: $p(x) \approx \lambda_i \langle u_i, x\rangle^3 \approx \lambda_i > 0$ (homework)
  - → When $x \perp u_i \; \forall i \in [k]$: $p(x) \approx 0$

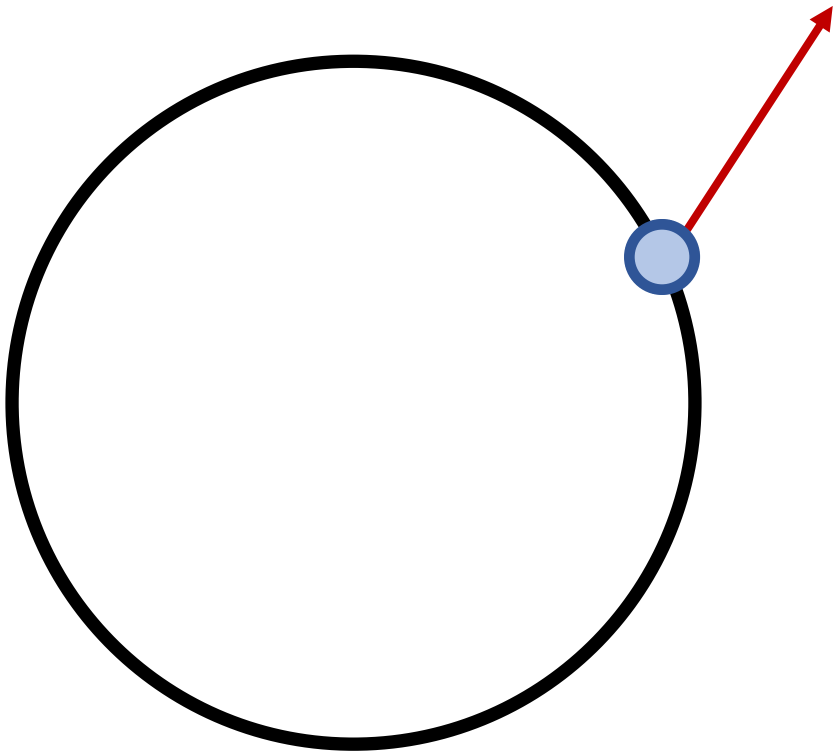
Gradient ascent:

$$\begin{aligned} x^t &= x^{t-1} + \eta \cdot \nabla p(x) \\ &= x^{t-1} + 3\eta \cdot T(:, x, x) \end{aligned}$$

$$T(:, x, x)_a \coloneqq \sum_{b,c} T_{abc} x_b x_c$$

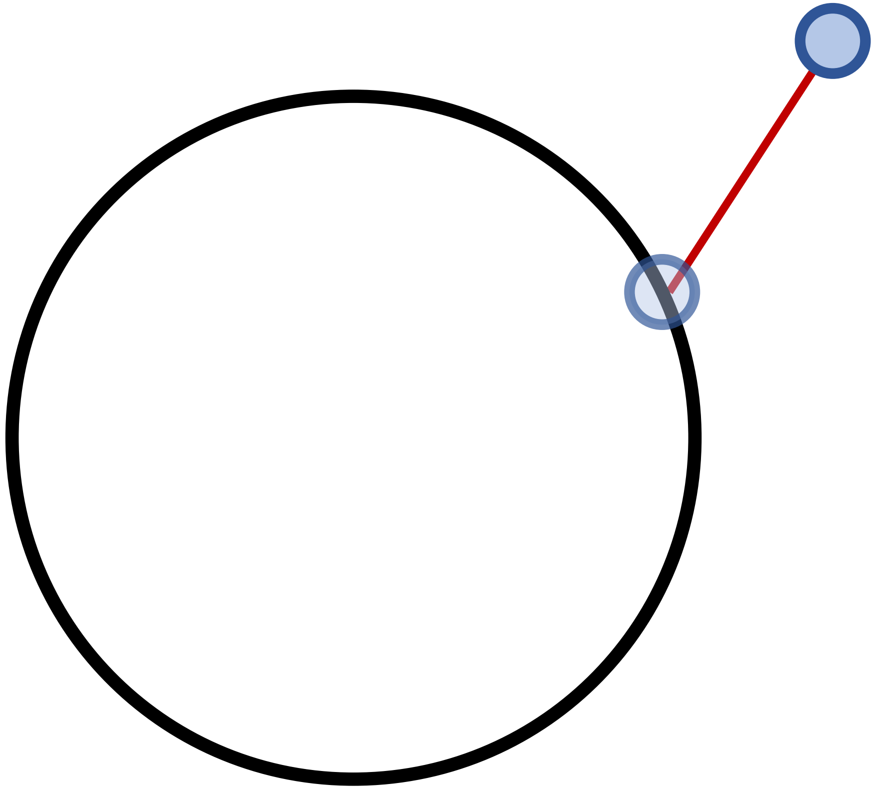
(Riemannian) Gradient descent

$$x^{t+1} = \textcolor{red}{\text{proj}}\left(x^t + 3\eta \cdot T(:, x^t, x^t)\right)$$



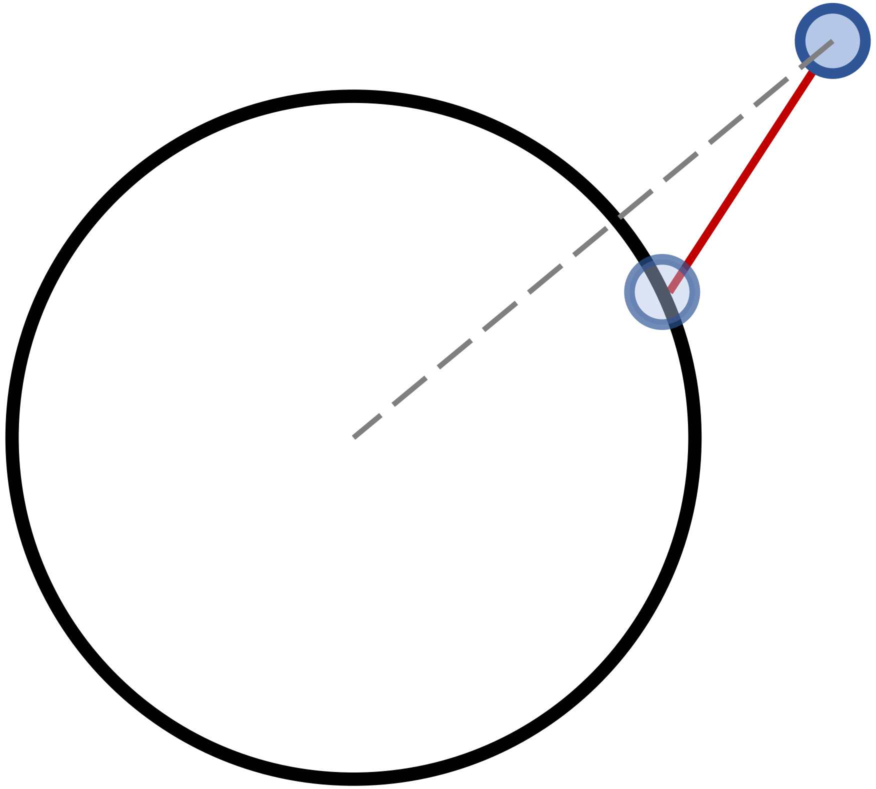
(Riemannian) Gradient descent

$$x^{t+1} = \text{proj}\left(x^t + 3\eta \cdot T(:, x^t, x^t)\right)$$



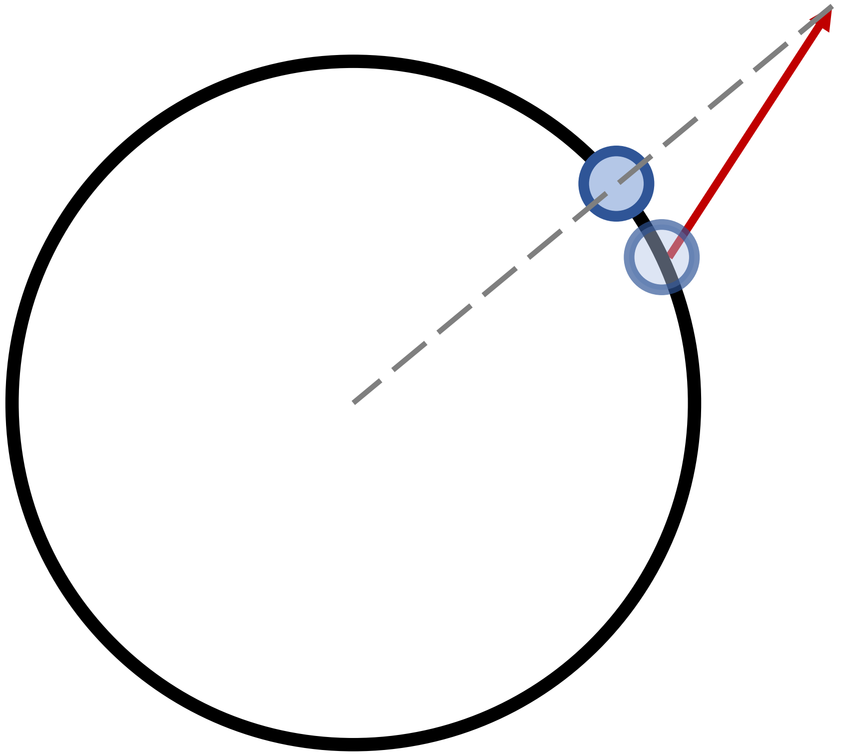
(Riemannian) Gradient descent

$$x^{t+1} = \text{proj}\left(x^t + 3\eta \cdot T(:, x^t, x^t)\right)$$



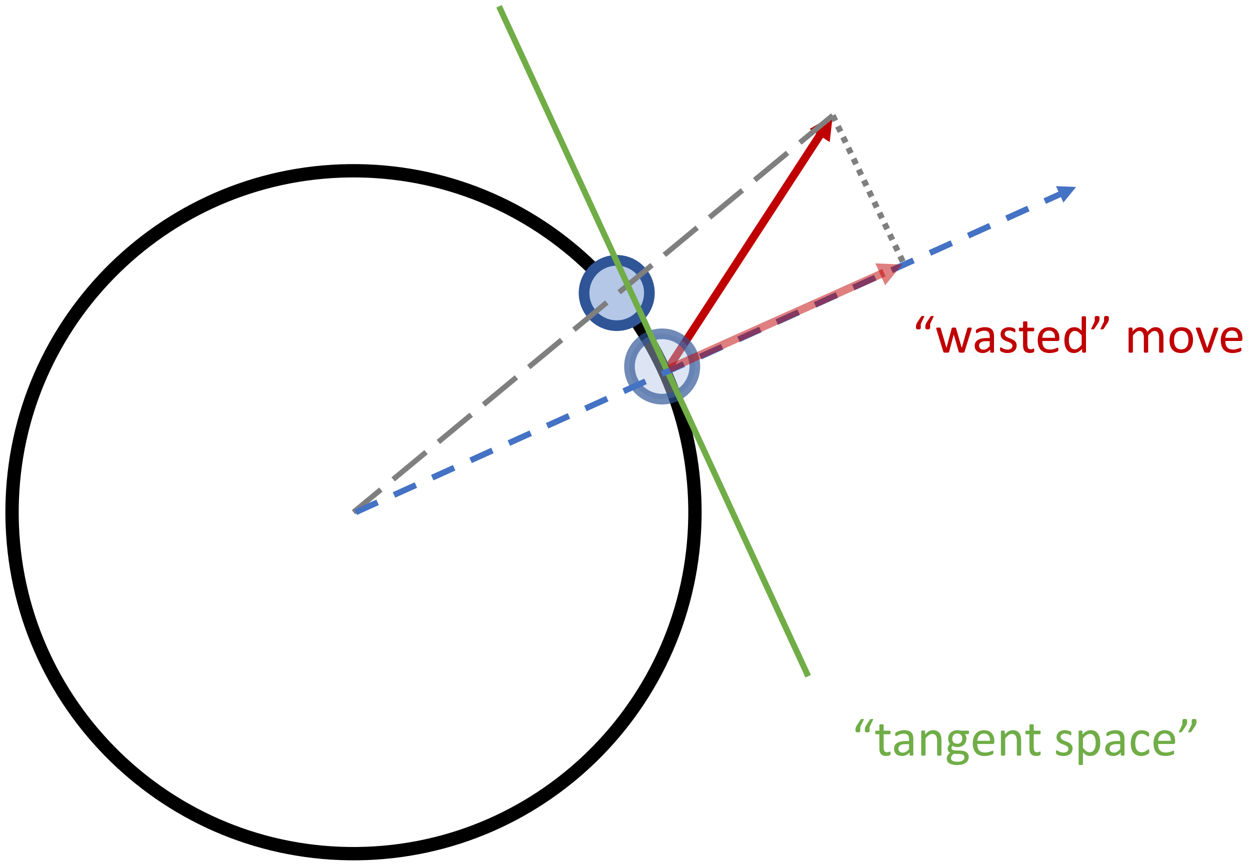
(Riemannian) Gradient descent

$$x^{t+1} = \text{proj}\left(x^t + 3\eta \cdot T(:, x^t, x^t)\right)$$



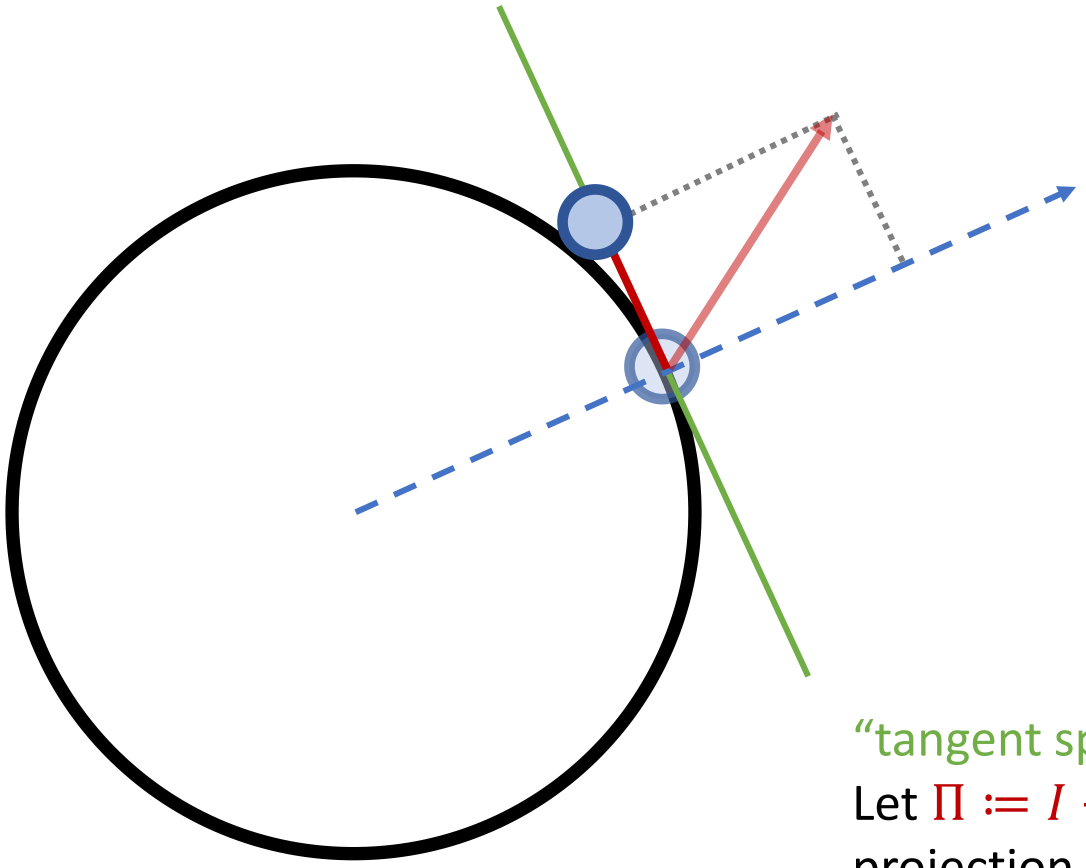
(Riemannian) Gradient descent

$$x^{t+1} = \color{red}{\text{proj}}\left(x^t + 3\eta \cdot T(:, x^t, x^t)\right)$$



(Riemannian) Gradient descent

$$x^{t+1} = \mathrm{proj}\left(x^t + 3\eta \cdot \Pi \cdot T(:, x^t, x^t)\right)$$

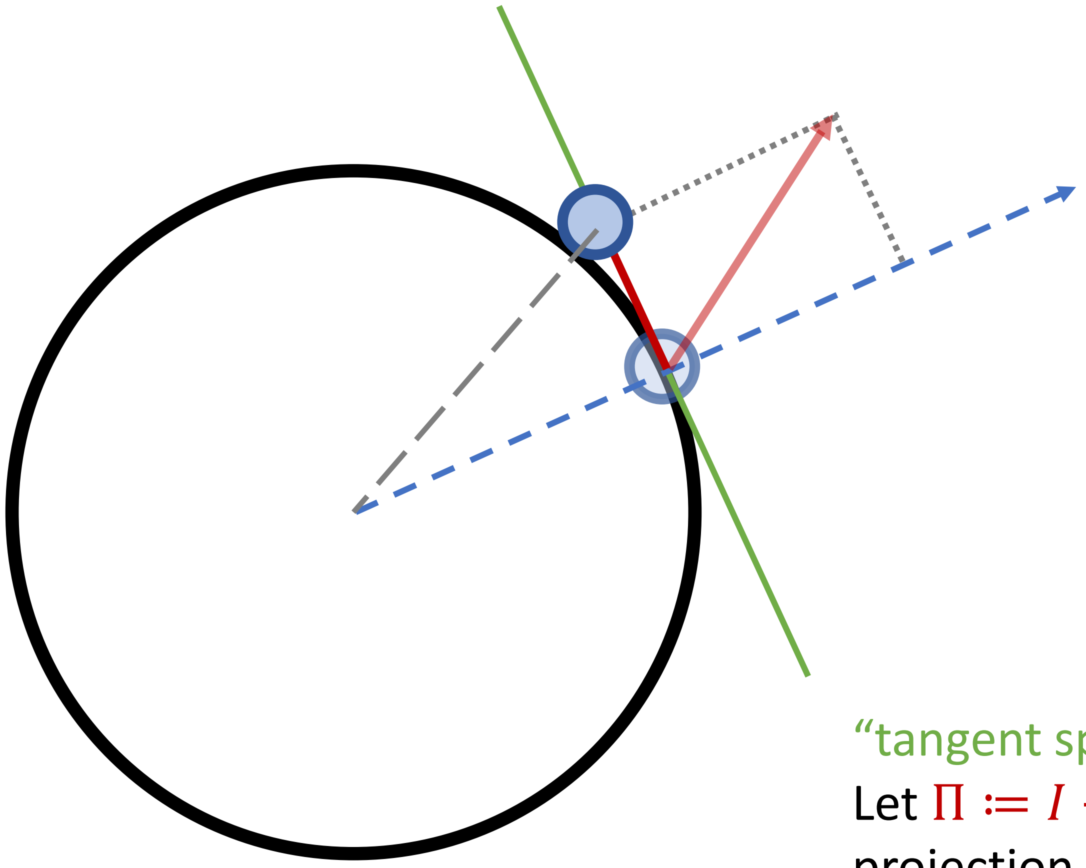


"tangent space"

Let $\Pi \coloneqq I - x^t (x^t)^\top$ be the projection to tangent space

(Riemannian) Gradient descent

$$x^{t+1} = \textcolor{red}{\mathrm{proj}}\left(x^t + 3\eta \cdot \textcolor{red}{\Pi} \cdot T(:, x^t, x^t)\right)$$

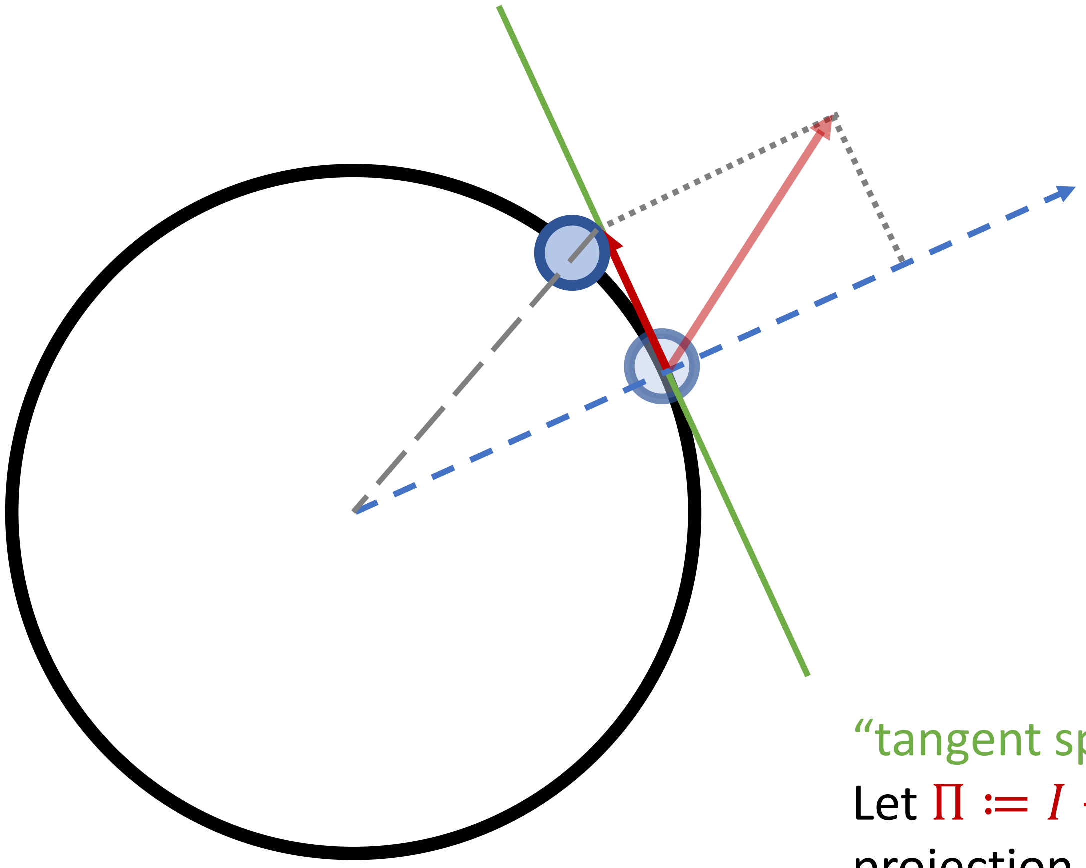


"tangent space"

Let $\Pi := I - x^t (x^t)^\top$ be the projection to tangent space

(Riemannian) Gradient descent

$$x^{t+1} = \text{proj}\left(x^t + 3\eta \cdot \Pi \cdot T(:, x^t, x^t)\right)$$



"tangent space"

Let $\Pi := I - x^t(x^t)^\top$ be the projection to tangent space

(Riemannian) Gradient descent

$$
\begin{aligned}
x^{t+1} &= \mathrm{proj}\left(x^t + 3\eta \cdot \Pi \cdot T(:, x^t, x^t)\right) \\
&= \mathrm{proj}\left(x^t + 3\eta \cdot (I - x^t (x^t)^\top) \cdot T(:, x^t, x^t)\right) \\
&= \mathrm{proj}\left(x^t + 3\eta \cdot T(:, x^t, x^t) - 3\eta \cdot x^t \sum_{abc} (x^t)_a T_{abc} (x^t)_a (x^t)_b\right) \\
&= \mathrm{proj}\left(x^t + 3\eta \cdot T(:, x^t, x^t) - 3\eta \cdot x^t \cdot \underbrace{T(x^t, x^t, x^t)}_{p(x^t)}\right)
\end{aligned}
$$

If we take $\eta \coloneqq \frac{1}{3p(x^t)}$ as the step size,

$$
x^{t+1} = \mathrm{proj}\left(\frac{T(:, x^t, x^t)}{p(x^t)}\right) = \frac{T(:, x^t, x^t)}{\|T(:, x^t, x^t)\|}
$$

- Tensor generalization of the matrix power method for finding top eigenvalue!

Matrix power method

- Let $M = \sum_i \lambda_i u_i \otimes u_i$ ($\equiv u_i u_i^\top$)
- Let $x^0 = \sum_i a_{i,0} u_i$ be the initial point
- Matrix power method update rule:

$$x^{t+1} = \frac{Mx^t}{\|Mx^t\|} = \text{proj}\big(M(:, x^t)\big)$$

- We can track the correlations of x^t and each eigenvector u_i:

$$a_{i,t} := \langle x^t, u_i \rangle = \frac{1}{\|Mx^{t-1}\|} \sum_j \lambda_j a_{j,t-1} \langle u_i, u_j \rangle = \frac{\lambda_i}{\|Mx^{t-1}\|} a_{i,t-1}$$

$$\Rightarrow \quad \frac{a_{i,t}}{a_{i,t}} = \frac{\lambda_i}{\lambda_1} \cdot \frac{a_{i,t-1}}{a_{1,t-1}} = \left(\frac{\lambda_i}{\lambda_1}\right)^2 \frac{a_{i,t-2}}{a_{1,t-2}} \cdots = \left(\frac{\lambda_i}{\lambda_1}\right)^t \frac{a_{i,0}}{a_{1,0}} \to 0 \text{ if } \frac{\lambda_i}{\lambda_1} < 1$$

Exponential ("linear") convergence

Tensor power method

- Let $T = \sum_i \lambda_i u_i \otimes u_i \otimes u_i$
- Let $x^0 = \sum_i a_{i,0} u_i$ be the initial point
- Tensor power method update rule:

$$x^{t+1} = \mathrm{proj}\big(T(:, x^t, x^t)\big) = \frac{T(:, x^t, x^t)}{\|T(:, x^t, x^t)\|}$$

- Track the the correlations of x^t and each orthonormal factor u_i:

$$a_{i,t} \coloneqq \langle x^t, u_i \rangle = \frac{1}{\|T(:, x^{t-1}, x^{t-1})\|} \left\langle \sum_i \lambda_i u_i \langle x^{t-1}, u_i \rangle^2, u_i \right\rangle$$

$$= \frac{\lambda_i a_{i,t-1}^2}{\|T(:, x^{t-1}, x^{t-1})\|}$$

Tensor power method

$$a_{i,t} = \frac{\lambda_i a_{i,t-1}^2}{\|T(:, x^{t-1}, x^{t-1})\|}$$

- We also compute the ratio:

$$\frac{a_{i,t}}{a_{1,t}} = \frac{\lambda_i}{\lambda_1}\frac{a_{i,t-1}^2}{a_{1,t-1}^2} = \left(\frac{\lambda_i}{\lambda_1}\right)^{1+2}\left(\frac{a_{i,t-2}}{a_{1,t-2}}\right)^{2^2} = \cdots = \left(\frac{\lambda_i}{\lambda_1}\right)^{1+2+\cdots+2^{t-1}}\left(\frac{a_{i,0}}{a_{1,0}}\right)^{2^t}$$

$$= \left(\frac{\lambda_i a_{i,0}}{\lambda_1 a_{1,0}}\right)^{2^t}\frac{\lambda_1}{\lambda_i} \quad \to 0 \ \text{ if } \ \frac{\lambda_i a_{i,0}}{\lambda_1 a_{1,0}} < 1$$

doubly exponential ("quadratic") convergence

- However, the probability that $\max_{i \neq 1} \frac{\lambda_i a_{i,0}}{\lambda_1 a_{1,0}} < 1$ for a random initial point is $\sim 1/k$
- But we can just use the above to argue we converge to whichever u_i maximizes $\lambda_i a_{i,0}$

Tensor power method

How to find the remaining factors?

- Deflation
  - Run tensor power method to find one factor $\hat{u}_i$
  - The coefficient $\lambda_i \approx p(\hat{u}_i) = T(\hat{u}_i, \hat{u}_i, \hat{u}_i)$
  - Let $T \leftarrow T - p(\hat{u}_i)\hat{u}_i \otimes \hat{u}_i \otimes \hat{u}_i$, repeat

- Clustering
  - Use different random initial points to run tensor power method
  - Let $\tilde{u}_1, \ldots, \tilde{u}_n$ be the outputs
  - Run a clustering algorithm to estimate $u_1, \ldots, u_k$

Alternating least squares (ALS)

View tensor decomposition as a pure optimization problem:

$$\min_{\hat{u}_1,\dots,\hat{u}_k} \left\| T - \sum_{i\in[k]} \hat{u}_i \otimes \hat{u}_i \otimes \hat{u}_i \right\|_F^2$$

In each iteration, we fix two dimensions and optimize the remaining one:

$$\hat{u}_i^{t+1} = \mathrm{proj}\left(\underbrace{\arg\min_{\{\hat{u}_i\}} \left\| T - \sum_{i\in[k]} \hat{u}_i \otimes \hat{u}_i^t \otimes \hat{u}_i^t \right\|_F^2}_{\text{just a least-squares regression}} \right)$$

just a least-squares regression

Hard to analyze, but very powerful in practice.

Rank-1 ALS is tensor power method

$$\arg\min_{\hat{u}} \|T - \hat{u} \otimes \hat{u}^t \otimes \hat{u}^t\|_F^2 = \arg\min_{\hat{u}} \sum_{abc} (T_{abc} - (\hat{u})_a (\hat{u}^t)_b (\hat{u}^t)_c)^2$$

$$= \arg\min_{\hat{u}} \sum_{abc} T_{abc}^2 - 2T_{abc}(\hat{u})_a (\hat{u}^t)_b (\hat{u}^t)_c + (\hat{u})_a^2 (\hat{u}^t)_b^2 (\hat{u}^t)_c^2$$

$$= \arg\min_{\hat{u}} \sum_{a} -2(\hat{u})_a \sum_{bc} T_{abc}(\hat{u}^t)_b (\hat{u}^t)_c + (\hat{u})_a^2 \sum_{bc} (\hat{u}^t)_b^2 (\hat{u}^t)_c^2$$

$$= \arg\min_{\hat{u}} \sum_{a} -2(\hat{u})_a T(:, \hat{u}^t, \hat{u}^t) + (\hat{u})_a^2$$

Therefore, the lease-squares solution is $\hat{u} = T(:, \hat{u}^t, \hat{u}^t)$, and

$$\hat{u}^{t+1} = \text{proj}\big(T(:, \hat{u}^t, \hat{u}^t)\big)$$

Tensor power method update rule

Removing the orthogonality condition

In our previous discussions, we assume that $T = \sum_i \lambda_i u_i \otimes u_i \otimes u_i$ and $\{u_i\}$ are orthonormal vectors. What if they are non-orthogonal but only linearly independent?

We'll see two solutions:

1. Whitening
2. Directly analyzing the tensor power method for non-orthogonal factors

Whitening

In many practical applications, we not only get access to T, but also to the following matrix:

$$M = \sum_i \lambda_i u_i u_i^\top$$

We can do the following procedure to orthogonalize the factors:

- Let $M = VDV^\top$ be the eigendecomposition of M, where $V \in \mathbb{R}^{d \times k}$ and $D \in \mathbb{R}^{k \times k}$
- Define $W := VD^{-1/2}$ and $\tilde{u}_i := \sqrt{\lambda_i} W^\top u_i \in \mathbb{R}^k$
- Then, we can check that

$$\sum_{i=1}^{k} \tilde{u}_i \tilde{u}_i^\top = \sum_{i=1}^{k} \lambda_i W^\top u_i u_i^\top W = W^\top M W = D^{-1/2} V^\top V D V^\top V D^{-1/2} = D^{-1/2} D D^{-1/2} = I$$

- It implies that $\{\tilde{u}_i\}$ are orthonormal

Whitening

- $\{\tilde{u}_i \coloneqq \sqrt{\lambda_i} W^\top u_i\}$ are orthonormal
- Define a new tensor $T' \coloneqq T(W, W, W) \in \mathbb{R}^{k \times k \times k}$ such that

$$T'_{abc} = \sum_{a'b'c' \in [d]} T_{a'b'c'} W_{a'a} W_{b'b} W_{c'c} \qquad \forall a, b, c \in [k]$$

$$= \sum_{a'b'c'} \sum_i \lambda_i \, (u_i)_{a'} (u_i)_{b'} (u_i)_{c'} W_{a'a} W_{b'b} W_{c'c}$$

$$= \sum_i \lambda_i (W^\top u_i)_a (W^\top u_i)_b (W^\top u_i)_c = \sum_i \lambda_i^{-1/2} (\tilde{u}_i)_a (\tilde{u}_i)_b (\tilde{u}_i)_c$$

- Hence, $T' = \sum_i \lambda_i^{-1/2} \tilde{u}_i \otimes \tilde{u}_i \otimes \tilde{u}_i$
- Transform back:

$$\lambda_i^{-1/2} D^{1/2} V^\top \tilde{u}_i = \lambda_i^{-1/2} D^{1/2} V^\top \lambda_i^{1/2} W^\top u_i = D^{1/2} V^\top V D^{-1/2} u_i = u_i$$

Tensor power method for non-orthogonal factors

Theorem (Sharan-Valiant, 2017).

Consider a d-dimensional rank k tensor $T = \sum_{i\in[k]} u_i \otimes u_i \otimes u_i$.

Let $c_{\max} = \max_{i\neq j}\left|\langle u_i, u_j\rangle\right|$, and assume $c_{\max} \leq 1/k^{1+\epsilon}$.

If the initial point is randomly chosen, then with high probability the tensor power method converge to one of the true factors (say u_1) in $N = \mathcal{O}(\log k + \log\log d)$ steps, and the error at convergence satisfies

$$\|u_1 - x^N\| \leq \mathcal{O}(k\max\{c_{\max}, 1/d\}^2)$$

Proof setups

Tensor power method update:

$$x^t = \frac{\sum_i \langle x^{t-1}, u_i \rangle^2 u_i}{\left\| \sum_i \langle x^{t-1}, u_i \rangle^2 u_i \right\|} = \frac{\sum_i a_{i,t-1}^2 u_i}{\left\| \sum_i a_{i,t-1}^2 u_i \right\|}$$

$$a_{i,t} \coloneqq \langle x^t, u_i \rangle$$

We have

$$\begin{cases} a_{i,t} = \langle x^t, u_i \rangle = \dfrac{\sum_j a_{j,t-1}^2 \langle u_j, u_i \rangle}{\left\| \sum_j a_{j,t-1}^2 u_j \right\|} = \dfrac{a_{1,t-1}^2 \sum_j \hat{a}_{j,t-1}^2 c_{i,j}}{\left\| \sum_j a_{j,t-1}^2 u_j \right\|} \\ a_{1,t} = \dfrac{a_{1,t-1}^2 \sum_j \hat{a}_{j,t-1}^2 c_{1,j}}{\left\| \sum_j a_{j,t-1}^2 u_j \right\|} \end{cases}$$

$$\hat{a}_{i,t} \coloneqq \frac{a_{i,t}}{a_{1,t}}$$

$$\hat{a}_{i,t} = \frac{a_{i,t}}{a_{1,t}} = \frac{\sum_j \hat{a}_{j,t-1}^2 c_{i,j}}{\sum_j \hat{a}_{j,t-1}^2 c_{1,j}} = \frac{c_{i,1} + \hat{a}_{i,t-1}^2 + \sum_{j \neq 1,i} \hat{a}_{j,t-1}^2 c_{i,j}}{1 + \sum_{j \geq 2} \hat{a}_{j,t-1}^2 c_{1,j}}$$

Define a sequence (potential energy):

$$\begin{cases} \beta_0 = \max_{i \neq 1} |\hat{a}_{i,0}| \\ \beta_t = c_{\max} + \beta_{t-1}^2 + 3k c_{\max} \beta_{t-1}^2 \end{cases}$$

Lemma. For all $t \geq 0$ and for all $i \neq 1$,

$$|\hat{a}_{i,t}| \leq \beta_t \qquad (✦)$$

Prove by induction on t:

- $t = 0$: (✦) trivially holds
- Assume (✦) holds for $0, \dots, t-1$

$$\hat{a}_{i,t} = \frac{c_{i,1} + \hat{a}_{i,t-1}^2 + \sum_{j\neq 1,i} \hat{a}_{j,t-1}^2 c_{i,j}}{1 + \sum_{j\geq 2} \hat{a}_{j,t-1}^2 c_{1,j}}$$

- Denominator:

$$\left(1 + \sum_{j\geq 2} \hat{a}_{j,t-1}^2 c_{1j}\right)^{-1} = 1 - \sum_{j\geq 2} \hat{a}_{j,t-1}^2 c_{1j} + R_1 \text{ where } |R_1| \leq \left|\sum_{j\geq 2} \hat{a}_{j,t-1}^2 c_{1j}\right|^2 \leq k^2 c_{\max}^2 \beta_{t-1}^4$$

$(1-x)^{-1} = 1 - x + \mathcal{O}(x^2)\ \ \forall x < 1$ (by induction hypothesis)

$$\left|1 - \sum_{j\geq 2} \hat{a}_{j,t-1}^2 c_{1j} + R_1\right| \leq 1 + k c_{\max} \beta_{t-1}^2 + k^2 c_{\max}^2 \beta_{t-1}^4$$

- Numerator:

$$\left|c_{i1} + \hat{a}_{i,t-1}^2 + \sum_{j\neq 1,i} \hat{a}_{j,t-1}^2 c_{ij}\right| \leq |c_{i1}| + \hat{a}_{i,t-1}^2 + \left|\sum_{j\neq 1,i} \hat{a}_{j,t-1}^2 c_{ij}\right| \leq c_{\max} + \beta_{t-1}^2 + k c_{\max} \beta_{t-1}^2$$

Putting them together:

$$
\begin{aligned}
|\hat{a}_{i,t}| &= \left| c_{i1} + \hat{a}_{i,t-1}^2 + \sum_{j \neq 1,i} \hat{a}_{j,t-1}^2 c_{ij} \right| \cdot \left| 1 - \sum_{j \geq 2} \hat{a}_{j,t-1}^2 c_{1j} + R_1 \right| \\
&\leq (c_{\max} + \beta_{t-1}^2 + k c_{\max} \beta_{t-1}^2)(1 + k c_{\max} \beta_{t-1}^2 + k^2 c_{\max}^2 \beta_{t-1}^4) \\
&\leq c_{\max} + \beta_{t-1}^2 + \big(2 + o(1)\big) k c_{\max} \beta_{t-1}^2 \qquad (c_{\max} \leq 1/k^{1+\epsilon} \text{ and } \beta_{t-1} < 1) \\
&< \beta_t
\end{aligned}
$$

If $\beta_t < 1$, by induction, (✦) holds for any t and $i \neq 1$

Lemma. $\beta_t < 3\eta$ for any $t = \Omega(\log k + \log \log d)$. Moreover, $\beta_t < 1$ for any $t \geq 0$.

$$
\begin{cases}
\beta_0 = \max_{i \neq 1} |\hat{a}_{i,0}| \\
\beta_t = c_{\max} + \beta_{t-1}^2 + 3k c_{\max} \beta_{t-1}^2
\end{cases}
$$

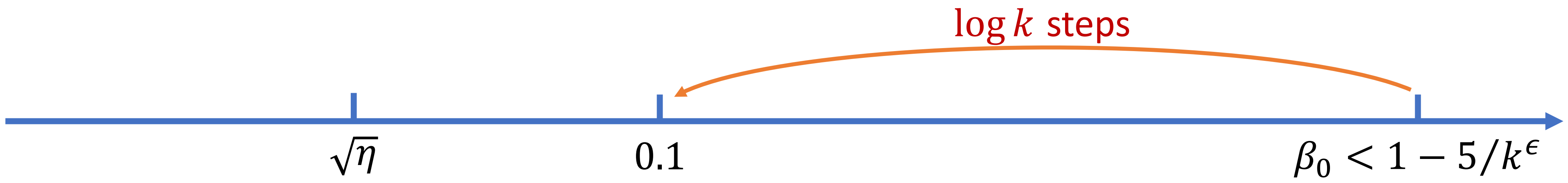
by the random initialization
(proof omitted)

1. $\beta_t \in (0.1, 1 - 5/k^{1+\epsilon})$

We have

$$\begin{aligned}\beta_{t+1} = c_{\max} + \beta_t^2 + 3kc_{\max}\beta_t^2 &\le (1 + 4kc_{\max})\beta_t^2 \\ &\le (1 + 4kc_{\max}) \cdot (1 + 4kc_{\max})^2 \beta_{t-1}^2 \\ &\vdots \\ &\le (1 + 4kc_{\max})^{1+2+2^2+\cdots+2^{t-1}} \beta_0^{2^t} \\ &\le \left((1 + 4kc_{max})(1 - 5/k^{\epsilon})\right)^{2^t} \\ &\le \left((1 + 4/k^{\epsilon})(1 - 5/k^{\epsilon})\right)^{2^t} \\ &\le (1 - 1/k^{\epsilon})^{2^t}\end{aligned}$$

when $t \ge 2 \log k$, $(1 - 1/k^{\epsilon})^{2^t} < 0.1$



2. $\beta_t \in \left(\sqrt{\eta}, 0.1\right)$ where $\eta \coloneqq \max\{c_{\max}, 1/d\}$

We have

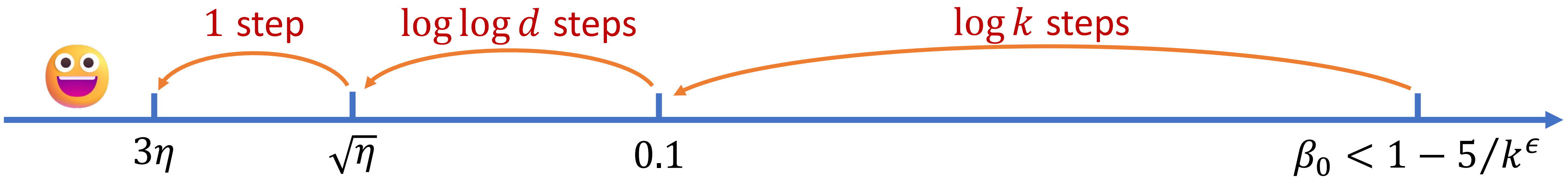
$$\beta_{t+1} = c_{\max} + \beta_t^2 + 3k c_{\max}\beta_t^2 \leq \beta_t^2 + \beta_t^2 + 0.3\beta_t^2 \leq 2.5\beta_t^2$$

Thus, $\beta_t \leq (2.5\beta_0)^{2^t} = 0.25^{2^t} < \sqrt{\eta}$ when $t > \log\log\eta^{-1} = \mathcal{O}(\log\log d)$

3. $\beta_t \leq \sqrt{\eta}$

We have

$$\beta_{t+1} = c_{\max} + \beta_t^2 + 3k c_{\max}\beta_t^2 \leq \eta + \eta + 0.3\beta_t^2 \leq 3\eta$$



Finish the proof of the Theorem

- We have proven that $\beta_t \leq 3\eta$ for $t = \Omega(\log k + \log\log d)$
- Hence, by the previous Lemma,

$$|\hat{a}_{it}| \leq \beta_t = \mathcal{O}(\eta)$$

- Recall the tensor power method update rule:

$$x^{t+1} = \frac{\sum_i \langle x^t, u_i\rangle^2 u_i}{\left\|\sum_i \langle x^t, u_i\rangle^2 u_i\right\|} = \frac{\sum_i a_{1,t}^2 \hat{a}_{i,t}^2 u_i}{\left\|\sum_i a_{1,t}^2 \hat{a}_{i,t}^2 u_i\right\|} = \frac{\sum_i \hat{a}_{i,t}^2 u_i}{\left\|\sum_i \hat{a}_{i,t}^2 u_i\right\|} = \frac{u_1 + \sum_{i>1} \hat{a}_{i,t}^2 u_i}{\left\|u_1 + \sum_{i>1} \hat{a}_{i,t}^2 u_i\right\|}$$

- $\left\|\sum_{i>1} \hat{a}_{i,t}^2 u_i\right\| \leq \mathcal{O}(k\eta^2) \quad \Longrightarrow \quad z \coloneqq \left\|u_1 + \sum_{i>1} \hat{a}_{i,t}^2 u_i\right\| \in 1 \pm \mathcal{O}(k\eta^2)$

$$\Longrightarrow \quad z^{-1} \in 1 \pm \mathcal{O}(k\eta^2)$$

- Thus, we get the desired error bound:

$$\|u_1 - x^{t+1}\| = \left\|(1 - z^{-1})u_1 - z^{-1} \sum_{i>1} \hat{a}_{i,t}^2 u_i\right\| = \mathcal{O}(k\eta^2)$$

Missing piece: randomized initialization creates a large gap in correlations

Suppose x^0 is sampled uniformly from $\mathbb{S}^{d-1}$

Lemma. If $c_{\max} < 1/k^{1+\epsilon}$ for any constant $\epsilon > 0$, then with probability at least $1 - \frac{\log^{\mathcal{O}(1)} k}{k^{\epsilon}}$, for any $i \neq 1$,

$$\left|\hat{a}_{i,0}\right| \leq 1 - \frac{5}{k^{\epsilon}} \quad \forall i \neq 1$$

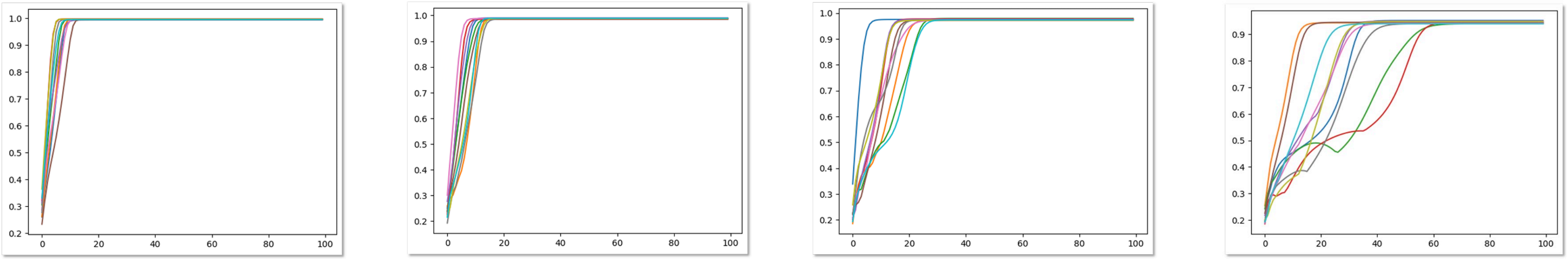
See (Sharan-Valiant 2017, Lemma 1) for the proof.

Recap

- Today, we see the key ideas and some theory behind the heuristic approaches for tensor decompositions
- We prove that the tensor power method converges fast if the factors are sufficiently "incoherent" ($c_{\max} < 1/k^{1+\epsilon}$)
- Up to now, we only consider the underdetermined regime ($k \leq d$)
- What about the overcomplete regime ($d < k \leq d^2$)?

$d = 400$

$k < d^{1.5}$



(a) $k = 400$ (b) $k = 1000$ (c) $k = 2000$ (d) $k = 4000$

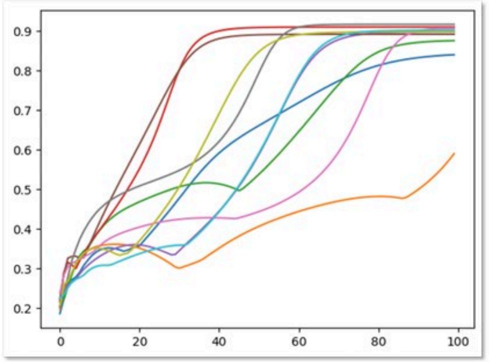
https://www.sitanchen.com/cs224/f24/scribe/lec3_final.pdf

Recap

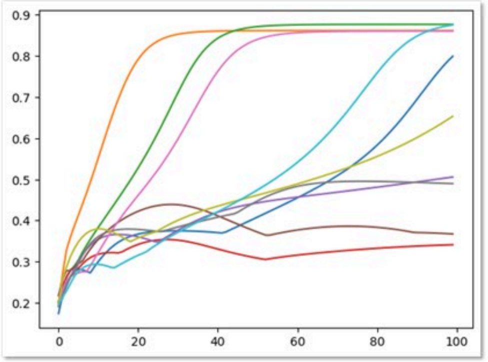
- Today, we see the key ideas and some theory behind the heuristic approaches for tensor decompositions
- We prove that the tensor power method converges fast if the factors are sufficiently "incoherent" ($c_{\max} < 1/k^{1+\epsilon}$)
- Up to now, we only consider the underdetermined regime ($k \leq d$)
- What about the overcomplete regime ($d < k \leq d^2$)?

$d = 400$

$k \simeq d^{1.5}$



(e) $k = 6000$



(f) $k = 8000$

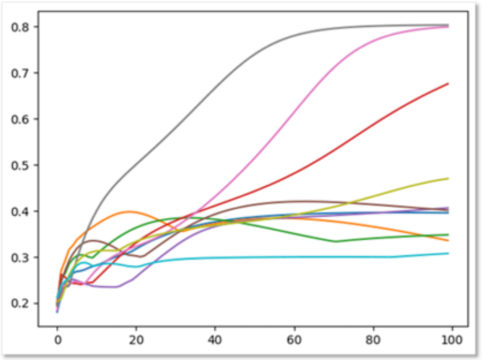
https://www.sitanchen.com/cs224/f24/scribe/lec3_final.pdf

Recap

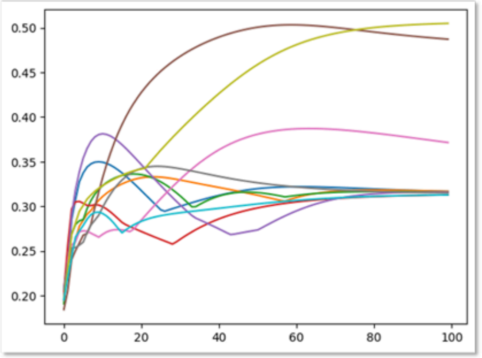
- Today, we see the key ideas and some theory behind the heuristic approaches for tensor decompositions
- We prove that the tensor power method converges fast if the factors are sufficiently "incoherent" ($c_{\max} < 1/k^{1+\epsilon}$)
- Up to now, we only consider the underdetermined regime ($k \leq d$)
- What about the overcomplete regime ($d < k \leq d^2$)?

$d = 400$

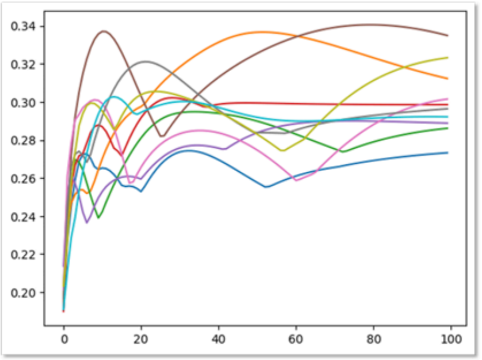
$k > d^{1.5}$



(g) $k = 10000$



(h) $k = 12000$



(i) $k = 15000$

https://www.sitanchen.com/cs224/f24/scribe/lec3_final.pdf