Communication-efficient distributed eigenspace estimation

Vasilis Charisopoulos

Joint work with Austin Benson & Anil Damle

SIAM Annual Meeting, 2022

Outline

Problem setting

Communication-efficient eigenspace estimation

Robustness to node failures

Problem setting

Motivating example: principal component analysis (PCA).

Problem setting

Motivating example: principal component analysis (PCA).

- Given samples $X_1, \ldots, X_n \in \mathbb{R}^d$, reduce dimension to $r \ll d$.

Problem setting

Motivating example: principal component analysis (PCA).

- Given samples $X_1, \ldots, X_n \in \mathbb{R}^d$, reduce dimension to $r \ll d$.
- Solution: top-r eigenspace of *empirical covariance* matrix:

$$\Sigma_n := \frac{1}{n} \sum_{j=1}^{n} X_j X_j^\mathsf{T} = V \Lambda V^\mathsf{T} + V_\perp \Lambda_\perp V_\perp^\mathsf{T}, \qquad V \in O(d, r).$$

Problem setting

Motivating example: principal component analysis (PCA).

- Given samples $X_1, \ldots, X_n \in \mathbb{R}^d$, reduce dimension to $r \ll d$.
- Solution: top-r eigenspace of *empirical covariance* matrix:

$$\Sigma_n := \frac{1}{n} \sum_{j=1}^{n} X_j X_j^\mathsf{T} = V \Lambda V^\mathsf{T} + V_\perp \Lambda_\perp V_\perp^\mathsf{T}, \qquad V \in O(d, r).$$

This talk: two challenges.

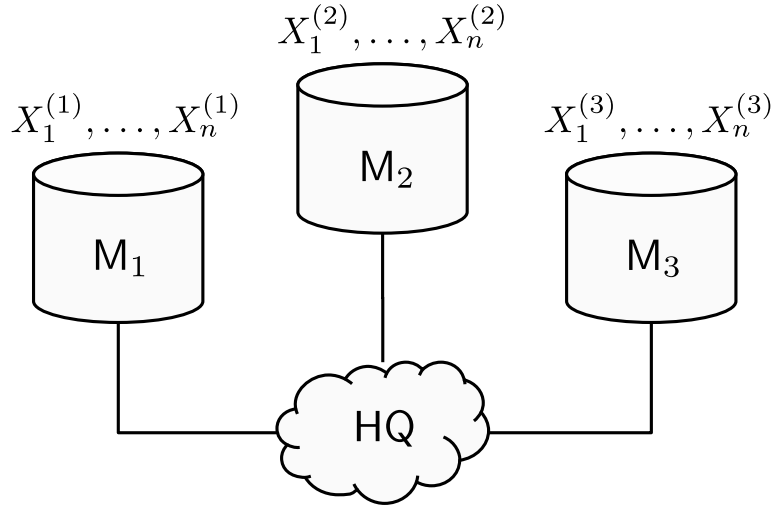
Problem setting

Motivating example: principal component analysis (PCA).

- Given samples $X_1, \ldots, X_n \in \mathbb{R}^d$, reduce dimension to $r \ll d$.
- Solution: top-r eigenspace of *empirical covariance* matrix:

$$\Sigma_n := \frac{1}{n} \sum_{j=1}^{n} X_j X_j^\mathsf{T} = V \Lambda V^\mathsf{T} + V_\perp \Lambda_\perp V_\perp^\mathsf{T}, \qquad V \in O(d, r).$$

1. What if data are *distributed*?



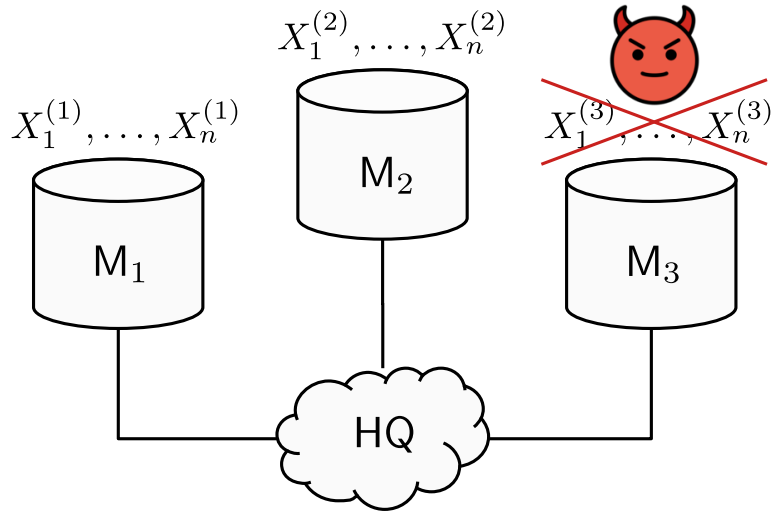
Problem setting

Motivating example: principal component analysis (PCA).

- Given samples $X_1, \ldots, X_n \in \mathbb{R}^d$, reduce dimension to $r \ll d$.
- Solution: top-r eigenspace of *empirical covariance* matrix:

$$\Sigma_n := \frac{1}{n} \sum_{j=1}^{n} X_j X_j^\mathsf{T} = V \Lambda V^\mathsf{T} + V_\perp \Lambda_\perp V_\perp^\mathsf{T}, \qquad V \in O(d, r).$$

1. What if data are *distributed*?
2. What if some machines are *compromised*?



Problem setting

General setting and assumptions:

- **Unknown** symmetric matrix $A \in \mathbb{R}^{d \times d}$ with decomposition

$$A = V \Lambda V^{\mathsf{T}} + V_{\perp} \Lambda_{\perp} V_{\perp}^{\mathsf{T}}, \quad V \in O(d, r).$$

Problem setting

General setting and assumptions:

- **Unknown** symmetric matrix $A \in \mathbb{R}^{d \times d}$ with decomposition

$$A = V \Lambda V^{\mathsf{T}} + V_{\perp} \Lambda_{\perp} V_{\perp}^{\mathsf{T}}, \quad V \in O(d, r).$$

- **Eigengap**: we have $\delta_r := \lambda_r(A) - \lambda_{r+1}(A) > 0$.

Problem setting

General setting and assumptions:

- **Unknown** symmetric matrix $A \in \mathbb{R}^{d\times d}$ with decomposition

$$A = V\Lambda V^{\mathsf{T}} + V_\perp \Lambda_\perp V_\perp^{\mathsf{T}}, \quad V \in O(d, r).$$

- **Eigengap**: we have $\delta_r := \lambda_r(A) - \lambda_{r+1}(A) > 0$.
- **Local errors**: machine i observes symmetric $A^{(i)} \in \mathbb{R}^{d\times d}$ such that

$$\left\|A^{(i)} - A\right\|_2 \leq \frac{\delta_r}{8}, \quad i = 1, \ldots, m, \quad (m := \text{number of machines.})$$

Problem setting

General setting and assumptions:

- **Unknown** symmetric matrix $A \in \mathbb{R}^{d \times d}$ with decomposition

$$A = V \Lambda V^{\mathsf{T}} + V_{\perp} \Lambda_{\perp} V_{\perp}^{\mathsf{T}}, \quad V \in O(d, r).$$

- **Eigengap**: we have $\delta_r := \lambda_r(A) - \lambda_{r+1}(A) > 0$.
- **Local errors**: machine i observes symmetric $A^{(i)} \in \mathbb{R}^{d \times d}$ such that

$$\left\| A^{(i)} - A \right\|_2 \leq \frac{\delta_r}{8}, \quad i = 1, \ldots, m, \quad (m := \text{number of machines.})$$

Goals:

1. *Communication-efficient* algorithms to estimate V.
2. *Robustness* to corrupted or compromised nodes.

Problem setting

General setting and assumptions:

- **Unknown** symmetric matrix $A \in \mathbb{R}^{d\times d}$ with decomposition

$$A = V\Lambda V^{\mathsf{T}} + V_\perp \Lambda_\perp V_\perp^{\mathsf{T}}, \quad V \in O(d,r).$$

- **Eigengap**: we have $\delta_r := \lambda_r(A) - \lambda_{r+1}(A) > 0$.
- **Local errors**: machine i observes symmetric $A^{(i)} \in \mathbb{R}^{d\times d}$ such that

$$\left\|A^{(i)} - A\right\|_2 \le \frac{\delta_r}{8}, \quad i = 1,\dots,m, \quad (m := \text{number of machines.})$$

Goals:

1. *Communication-efficient* algorithms to estimate V.
2. *Robustness* to corrupted or compromised nodes.

Quality of approximation measured in ℓ_2-subspace distance:

$$\mathrm{dist}_2(V,U) := \left\|(I - VV^{\mathsf{T}})U\right\|_2 = \left\|(I - UU^{\mathsf{T}})V\right\|_2.$$

Outline

Problem setting

Communication-efficient eigenspace estimation

Robustness to node failures

Existing approaches

Note: throughout this section, assume no machines are compromised.

[1] Garber et al., 2016
[2] Fan et al., 2019

Existing approaches

Note: throughout this section, assume no machines are compromised.

Option I: "Distribute" an existing iterative algorithm.

- Distributed power method: "map-reduce" eval of

$$v_{k+1} \leftarrow \frac{1}{m} \sum_{i=1}^{m} A^{(i)} v_k.$$

.

[1] Garber et al., 2016

[2] Fan et al., 2019

Existing approaches

Note: throughout this section, assume no machines are compromised.

Option I: "Distribute" an existing iterative algorithm.

- Distributed power method: "map-reduce" eval of

$$v_{k+1} \leftarrow \frac{1}{m} \sum_{i=1}^{m} A^{(i)} v_k.$$

.

- Shift-and-invert power method[1]:
  - Accelerated version of power method via shifts;
  - Reduces to distributed linear system solves.

[1] Garber et al., 2016
[2] Fan et al., 2019

Existing approaches

Note: throughout this section, assume no machines are compromised.

Option I: "Distribute" an existing iterative algorithm.

- Distributed power method: "map-reduce" eval of

$$v_{k+1} \leftarrow \frac{1}{m} \sum_{i=1}^{m} A^{(i)} v_k.$$

.

- Shift-and-invert power method[1]:
  - Accelerated version of power method via shifts;
  - Reduces to distributed linear system solves.

Drawback: "outer" algorithm sequential $\rightarrow \omega(1)$ communication rounds.

---

[1] Garber et al., 2016

[2] Fan et al., 2019

Existing approaches

Note: throughout this section, assume no machines are compromised.

Option II: Average spectral projectors[2] + SVD

1. Machine i computes local principal eigenvectors $V^{(i)}$, sends to HQ;

[1] Garber et al., 2016
[2] Fan et al., 2019

Existing approaches

Note: throughout this section, assume no machines are compromised.

Option II: Average spectral projectors[2] + SVD

1. Machine i computes local principal eigenvectors $V^{(i)}$, sends to HQ;
2. HQ forms and computes principal r-dim. eigenspace of
$$\frac{1}{m}\sum_{i=1}^{m} V^{(i)}(V^{(i)})^{\mathsf{T}}.$$

[1] Garber et al., 2016
[2] Fan et al., 2019

Existing approaches

Note: throughout this section, assume no machines are compromised.

Option II: Average spectral projectors[2] + SVD

1. Machine i computes local principal eigenvectors $V^{(i)}$, sends to HQ;
2. HQ forms and computes principal r-dim. eigenspace of
$$\frac{1}{m}\sum_{i=1}^{m} V^{(i)}(V^{(i)})^{\mathsf{T}}.$$

Drawback: existing analysis relies on distributional assumptions.

[1] Garber et al., 2016
[2] Fan et al., 2019

Proposed method

Algorithm: average **eigenvectors** of $A^{(i)}$.

Naive implementation:

- Machine i computes and sends $V^{(i)}$ to HQ
- HQ forms $\frac{1}{m}\sum_{i=1}^{m} V^{(i)}$, computes principal r-dim. eigenspace

Proposed method

Algorithm: average **eigenvectors** of $A^{(i)}$.

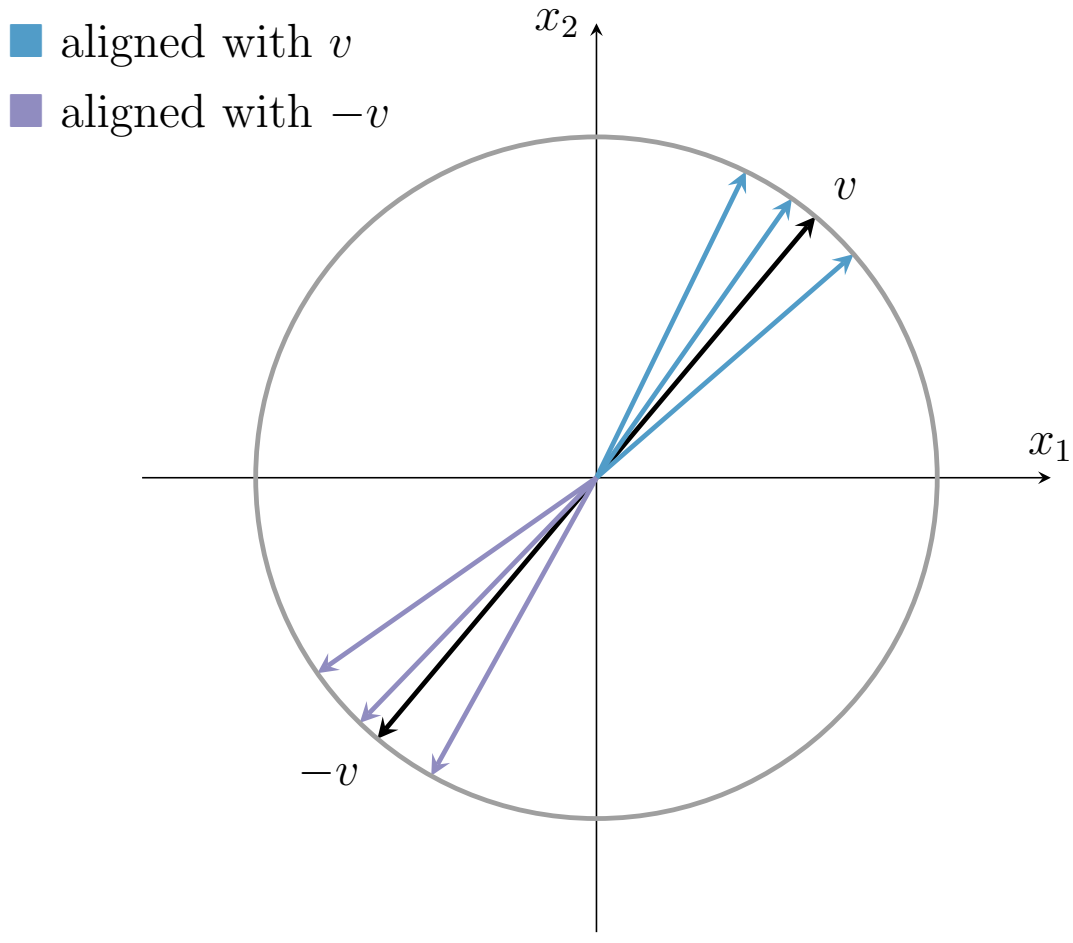
Naive implementation:

- Machine i computes and sends $V^{(i)}$ to HQ
- HQ forms $\frac{1}{m}\sum_{i=1}^{m} V^{(i)}$, computes principal r-dim. eigenspace

Challenge: Local solutions $V^{(i)}$ are defined up to symmetry. *Unclear* if naive averaging sufficient to reduce the error.

Averaging with symmetries

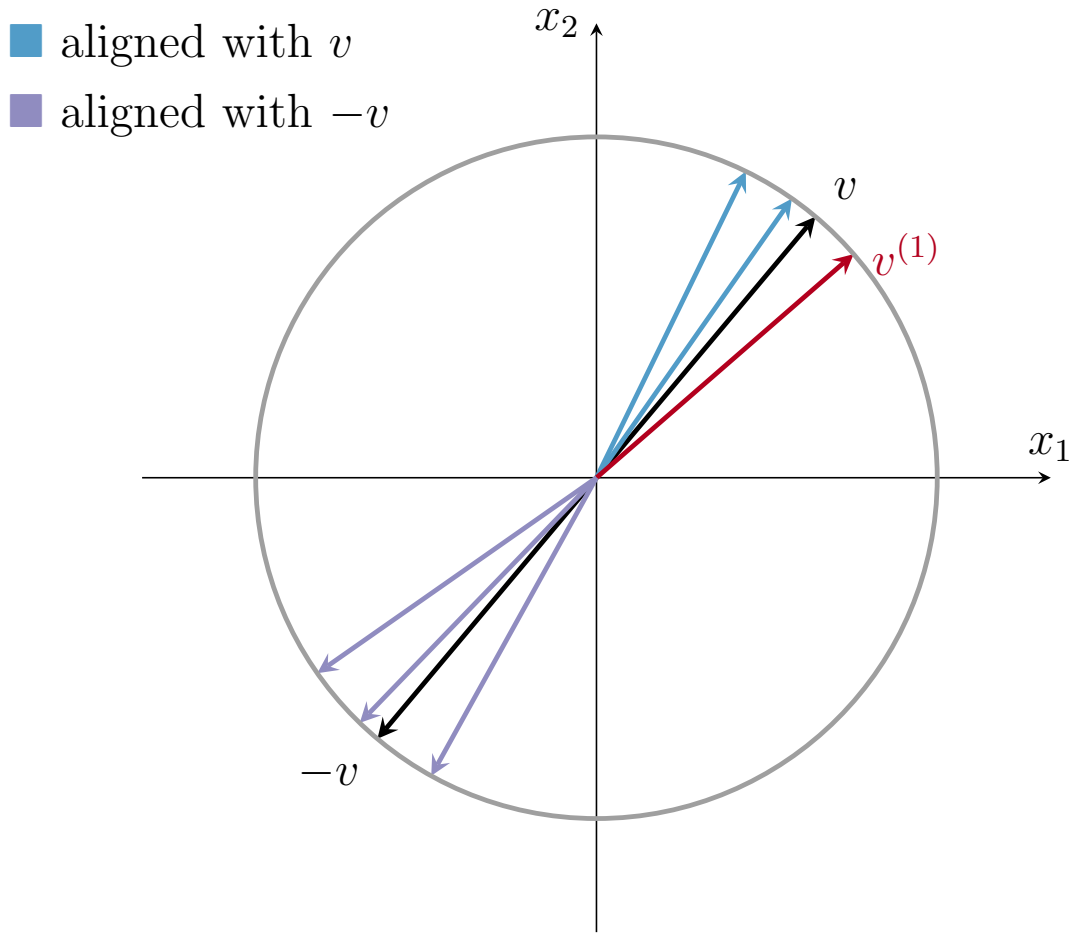
Problem (for $r = 1$): $v^{(i)}$ only defined up to sign.



Question: can we fix the sign ambiguity?

Averaging with symmetries

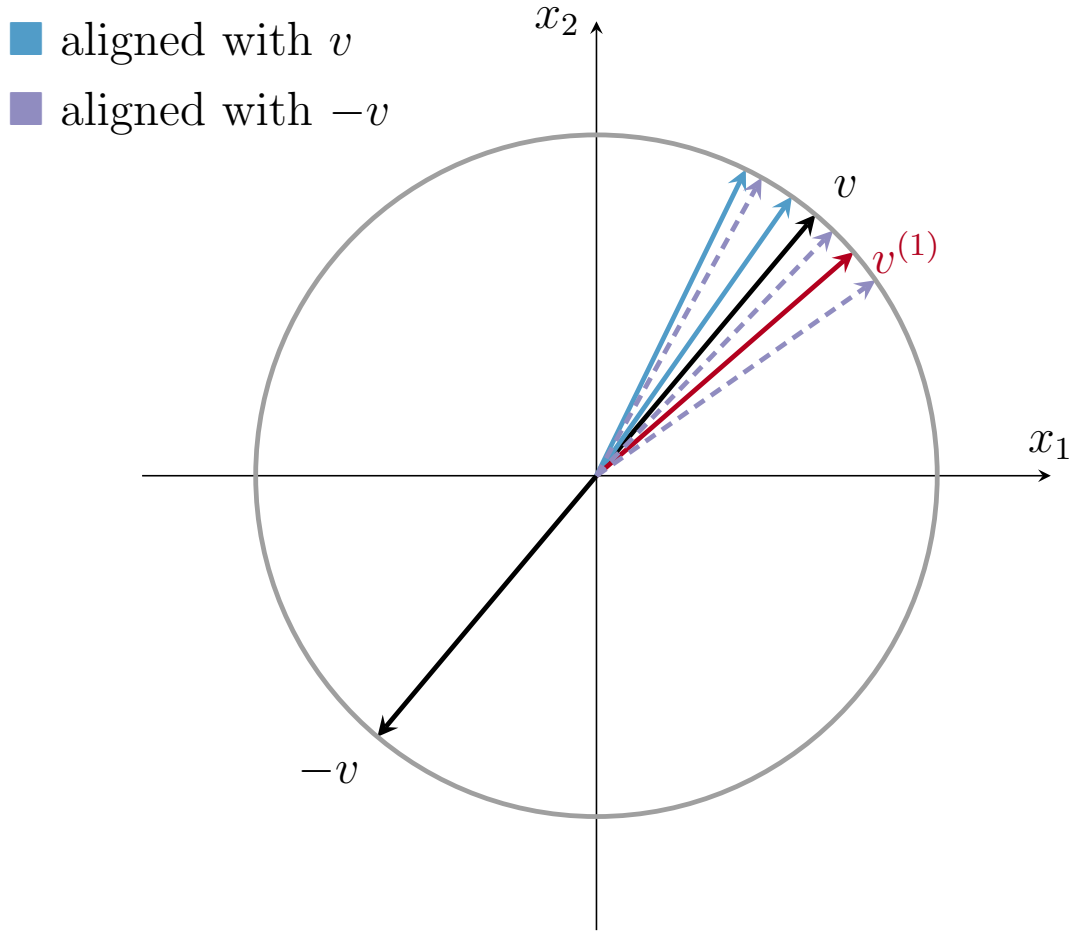
Problem (for $r = 1$): $v^{(i)}$ only defined up to sign.



Algorithm: $v^{(1)}$

Averaging with symmetries

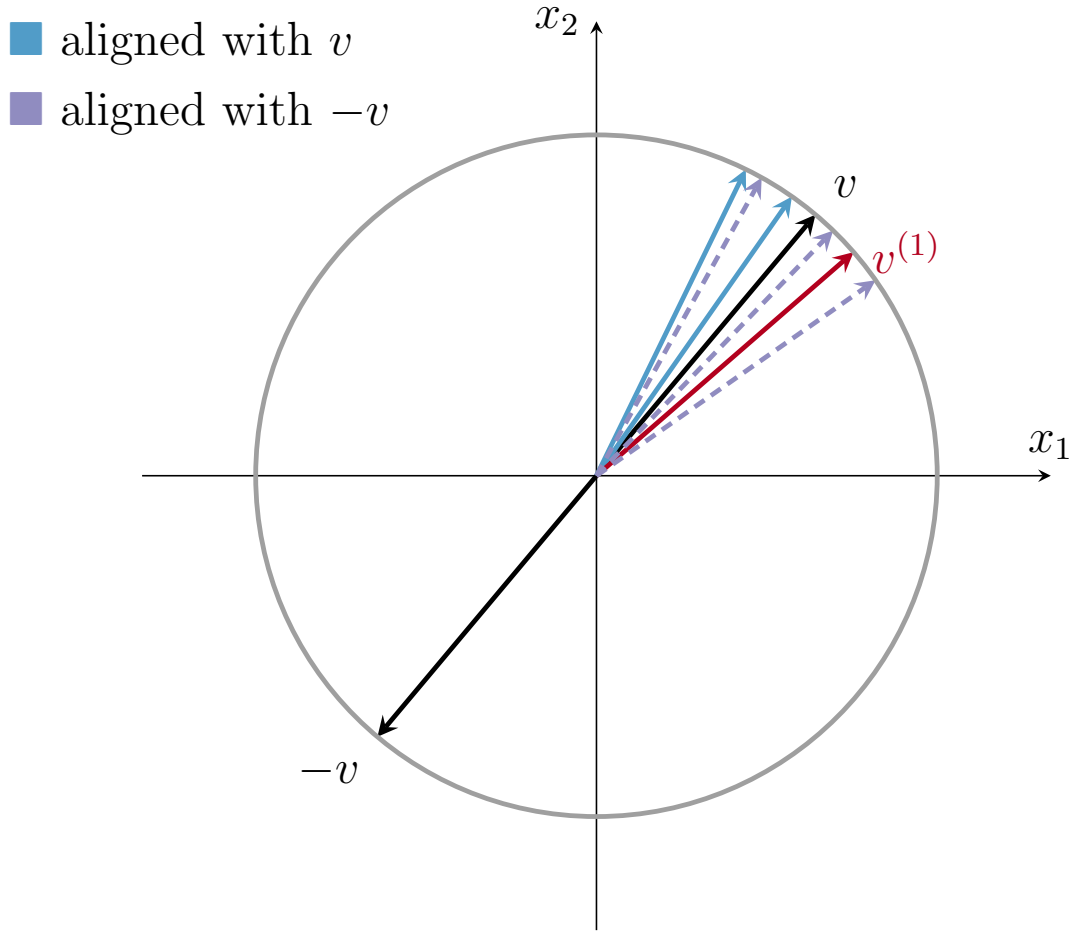
Problem (for $r = 1$): $v^{(i)}$ only defined up to sign.



Algorithm: $\operatorname{sign}(\langle v^{(i)}, v^{(1)} \rangle) \cdot v^{(i)}$

Averaging with symmetries

Problem (for $r = 1$): $v^{(i)}$ only defined up to sign.



Algorithm: $\tilde{v} := \frac{1}{m} \sum_{i=1}^{m} \operatorname{sign}(\langle v^{(i)}, v^{(1)} \rangle) \cdot v^{(i)}$

Averaging with symmetries

When $r > 1$, solutions invariant to arbitrary $Z \in O(r)$. **How should we align?**

[1] Higham '88

Averaging with symmetries

When $r > 1$, solutions invariant to arbitrary $Z \in O(r)$. **How should we align?**

Solution: align with minimizer of Procrustes problem:

Algorithm:

$$Z_i := \operatorname*{argmin}_{U \in O(r)} \left\| V^{(1)} - V^{(i)} U \right\|_{\mathrm{F}}, \quad \tilde{V}_i := V^{(i)} Z_i$$

$$\tilde{V} := \frac{1}{m} \sum_{i=1}^{m} \tilde{V}_i, \quad \tilde{V}, \_ \leftarrow \mathtt{qr}(\tilde{V})$$

[1] Higham '88

Averaging with symmetries

When $r > 1$, solutions invariant to arbitrary $Z \in O(r)$. **How should we align?**

Solution: align with minimizer of Procrustes problem:

Algorithm:

$$Z_i := \operatorname*{argmin}_{U \in O(r)} \left\| V^{(1)} - V^{(i)} U \right\|_{\mathrm{F}}, \quad \tilde{V}_i := V^{(i)} Z_i$$

$$\tilde{V} := \frac{1}{m} \sum_{i=1}^{m} \tilde{V}_i, \quad \tilde{V}, \_ \leftarrow \mathtt{qr}(\tilde{V})$$

- **Recovers sign-fixing** algorithm when $r = 1$.
- **Closed-form solution** for the Procrustes problem.[1]
- Can solve *general aggregation problems* under orthogonal symmetries.

[1] Higham '88

Averaging with symmetries

When $r > 1$, solutions invariant to arbitrary $Z \in O(r)$. **How should we align?**

Solution: align with minimizer of Procrustes problem:

Algorithm:

$$Z_i := \operatorname*{argmin}_{U \in O(r)} \left\| V^{(1)} - V^{(i)} U \right\|_{\mathrm{F}}, \quad \tilde{V}_i := V^{(i)} Z_i$$

$$\tilde{V} := \frac{1}{m} \sum_{i=1}^{m} \tilde{V}_i, \quad \tilde{V}, \_ \leftarrow \mathtt{qr}(\tilde{V})$$

- **Recovers sign-fixing** algorithm when $r = 1$.
- **Closed-form solution** for the Procrustes problem.[1]
- Can solve *general aggregation problems* under orthogonal symmetries.

Question: how does the error of the "Procrustes-fixed" average scale?

[1] Higham '88

Procrustes fixing: approximation error

We provide a **deterministic** result:

Theorem 1 (C., Benson, Damle '20)

With $\delta := \lambda_r(A) - \lambda_{r+1}(A) > 0$, the output $\tilde{V}$ of the algorithm satisfies

$$\operatorname{dist}_2(\tilde{V}, V) \leq \frac{1}{m} \sum_{i=1}^{m} \left(\frac{\left\| A^{(i)} - A \right\|_2}{\delta} \right)^2 + \frac{1}{\delta} \left\| \frac{1}{m} \sum_{i=1}^{m} A^{(i)} - A \right\|_2$$

Procrustes fixing: approximation error

We provide a **deterministic** result:

Theorem 1 (C., Benson, Damle '20)

With $\delta := \lambda_r(A) - \lambda_{r+1}(A) > 0$, the output $\tilde{V}$ of the algorithm satisfies

$$\operatorname{dist}_2(\tilde{V}, V) \leq \frac{1}{m} \sum_{i=1}^{m} \left(\frac{\left\| A^{(i)} - A \right\|_2}{\delta} \right)^2 + \frac{1}{\delta} \left\| \frac{1}{m} \sum_{i=1}^{m} A^{(i)} - A \right\|_2$$

Error bound: matches centralized algorithm up to **quadratic** local error.

Procrustes fixing: approximation error

We provide a **deterministic** result:

Theorem 1 (C., Benson, Damle '20)

With $\delta := \lambda_r(A) - \lambda_{r+1}(A) > 0$, the output $\tilde{V}$ of the algorithm satisfies

$$\operatorname{dist}_2(\tilde{V}, V) \leq \frac{1}{m} \sum_{i=1}^{m} \left(\frac{\left\| A^{(i)} - A \right\|_2}{\delta} \right)^2 + \frac{1}{\delta} \left\| \frac{1}{m} \sum_{i=1}^{m} A^{(i)} - A \right\|_2$$

Error bound: matches centralized algorithm up to **quadratic** local error.

Communication cost (PCA example):

Centralized: $m \cdot dn$ numbers (send all $m \cdot n$ samples)
Distributed: $m \cdot dr$ numbers (send m matrices $d \times r$)

Typically, $r \ll \min(d, n)$.

Procrustes fixing: approximation error

We provide a **deterministic** result:

Theorem 1 (C., Benson, Damle '20)

With $\delta := \lambda_r(A) - \lambda_{r+1}(A) > 0$, the output $\tilde{V}$ of the algorithm satisfies

$$\operatorname{dist}_2(\tilde{V}, V) \leq \frac{1}{m} \sum_{i=1}^{m} \left(\frac{\left\| A^{(i)} - A \right\|_2}{\delta} \right)^2 + \frac{1}{\delta} \left\| \frac{1}{m} \sum_{i=1}^{m} A^{(i)} - A \right\|_2$$

Error bound: matches centralized algorithm up to **quadratic** local error.

Communication cost (PCA example):

Centralized: $m \cdot dn$ numbers (send all $m \cdot n$ samples)
Distributed: $m \cdot dr$ numbers (send m matrices $d \times r$)

Typically, $r \ll \min(d, n)$.

Question: can we save on communication without compromising error?

Application: distributed PCA for subgaussian data

Setting: $A := \mathbb{E}_{\mathcal{D}}\left[XX^{\mathsf{T}}\right]$ and $A^{(i)} = \frac{1}{n}\sum_{j=1}^{n} X_j^{(i)}(X_j^{(i)})^{\mathsf{T}}$.

Here, $X \sim \mathcal{D}$ satisfy $\|X\|_{\psi_2} \leq \sigma$ (Gaussian-like tail), with $\|A\|_2 \asymp \sigma^2$.

Corollary 1 (C., Benson, Damle '20)

With $\delta := \lambda_r(A) - \lambda_{r+1}(A)$, the output $\tilde{V}$ of the algorithm satisfies

$$\mathrm{dist}_2(\tilde{V}, V) \lesssim \frac{\|A\|_2}{\delta}\sqrt{\frac{\mathsf{sr}(A) + \log n}{mn}} + \left(\frac{\|A\|_2}{\delta}\right)^2 \frac{\mathsf{sr}(A) + \log m}{n},$$

where $\mathsf{sr}(A) \leq \mathrm{rank}(A)$ is the *stable rank* of A.

Application: distributed PCA for subgaussian data

Setting: $A := \mathbb{E}_{\mathcal{D}}\left[XX^{\mathsf{T}}\right]$ and $A^{(i)} = \frac{1}{n}\sum_{j=1}^{n} X_j^{(i)}(X_j^{(i)})^{\mathsf{T}}$.

Here, $X \sim \mathcal{D}$ satisfy $\|X\|_{\psi_2} \leq \sigma$ (Gaussian-like tail), with $\|A\|_2 \asymp \sigma^2$.

Corollary 1 (C., Benson, Damle '20)

With $\delta := \lambda_r(A) - \lambda_{r+1}(A)$, the output $\tilde{V}$ of the algorithm satisfies

$$\mathrm{dist}_2(\tilde{V}, V) \lesssim \frac{\|A\|_2}{\delta}\sqrt{\frac{\mathsf{sr}(A) + \log n}{mn}} + \left(\frac{\|A\|_2}{\delta}\right)^2 \frac{\mathsf{sr}(A) + \log m}{n},$$

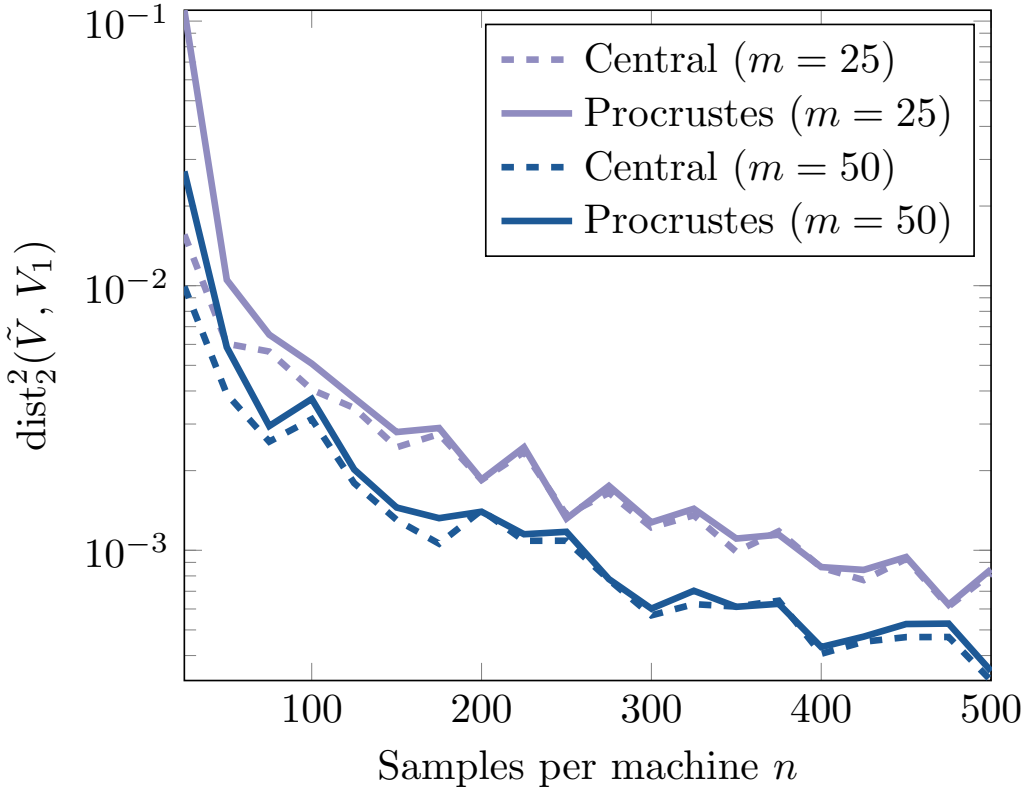
where $\mathsf{sr}(A) \leq \mathrm{rank}(A)$ is the *stable rank* of A.

Competitive with central algorithm when $n = \Omega(m)$.

Application: distributed PCA

Experiment: learn eigenvectors of covariance matrix of distribution $\mathcal{D}$, where:

- $\mathcal{D} = \mathcal{N}(0, A)$, where A has eigengap $\delta = 0.2$
- $\mathrm{sr}(A) \approx 16$, $d = 300$, machines m and samples n are varied.



Proof sketch

Theorem: if $\lambda_r(A) - \lambda_{r+1}(A) \geq \delta > 0$, have

$$\mathrm{dist}_2(\tilde{V}, V) \leq \frac{1}{\delta^2 m} \sum_{i=1}^{m} \left\| A^{(i)} - A \right\|_2^2 + \frac{1}{\delta} \left\| \frac{1}{m} \sum_{i=1}^{m} A^{(i)} - A \right\|_2 \qquad (1)$$

High level proof idea:

[1] Stewart, 2012.

Proof sketch

Theorem: if $\lambda_r(A) - \lambda_{r+1}(A) \geq \delta > 0$, have

$$\mathrm{dist}_2(\tilde{V}, V) \leq \frac{1}{\delta^2 m} \sum_{i=1}^{m} \left\| A^{(i)} - A \right\|_2^2 + \frac{1}{\delta} \left\| \frac{1}{m} \sum_{i=1}^{m} A^{(i)} - A \right\|_2 \qquad (1)$$

High level proof idea:

1. **Idealized case**: assume it is possible to align with V instead

$$Z_i^{\text{ideal}} \leftarrow \operatorname*{argmin}_{U \in O(r)} \left\| V - V^{(i)} U \right\|_{\mathrm{F}}, \quad \tilde{V}_i^{\text{ideal}} \leftarrow V^{(i)} Z_i^{\text{ideal}}$$

   Then, we can show that $\mathrm{dist}_2(\tilde{V}^{\text{ideal}}, V)$ follows (1).

[1] Stewart, 2012.

Proof sketch

Theorem: if $\lambda_r(A) - \lambda_{r+1}(A) \geq \delta > 0$, have

$$\operatorname{dist}_2(\tilde{V}, V) \leq \frac{1}{\delta^2 m} \sum_{i=1}^{m} \left\| A^{(i)} - A \right\|_2^2 + \frac{1}{\delta} \left\| \frac{1}{m} \sum_{i=1}^{m} A^{(i)} - A \right\|_2 \qquad (1)$$

High level proof idea:

1. **Idealized case**: assume it is possible to align with V instead

$$Z_i^{\text{ideal}} \leftarrow \operatorname*{argmin}_{U \in O(r)} \left\| V - V^{(i)} U \right\|_{\mathrm{F}}, \quad \tilde{V}_i^{\text{ideal}} \leftarrow V^{(i)} Z_i^{\text{ideal}}$$

   Then, we can show that $\operatorname{dist}_2(\tilde{V}^{\text{ideal}}, V)$ follows (1).

2. **Path independence**:[1] for $V^{(1)}$ "near" V, alignment essentially as good:

$$Z_i \leftarrow \operatorname*{argmin}_{U \in O(r)} \left\| V^{(1)} - V^{(i)} U \right\|_{\mathrm{F}}, \quad \tilde{V}_i \leftarrow V^{(i)} Z_i$$

$$\Rightarrow \left\| \tilde{V}_i - \tilde{V}_i^{\text{ideal}} \right\|_2 \lesssim \frac{1}{\delta^2} \left\| A^{(i)} - A \right\|_2^2.$$

[1] Stewart, 2012.

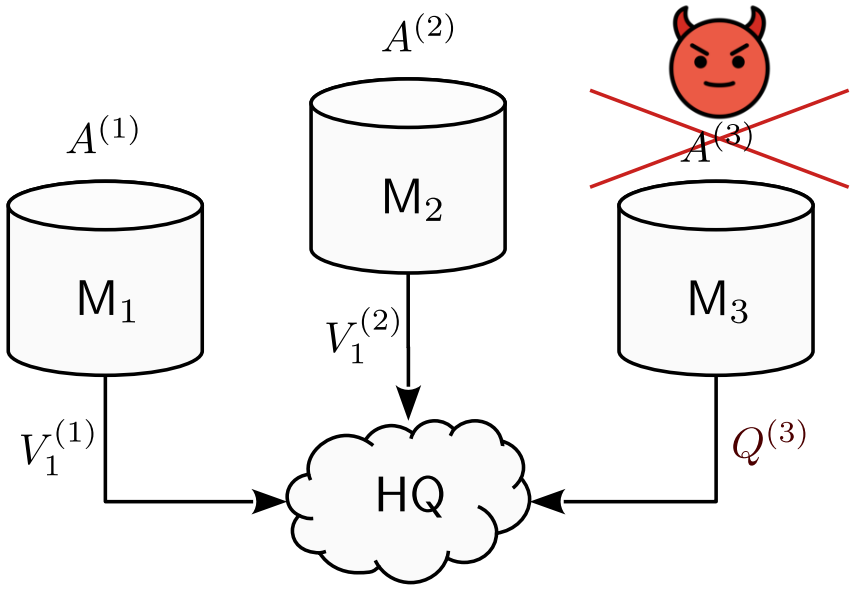
Outline

Problem setting

Communication-efficient eigenspace estimation

Robustness to node failures

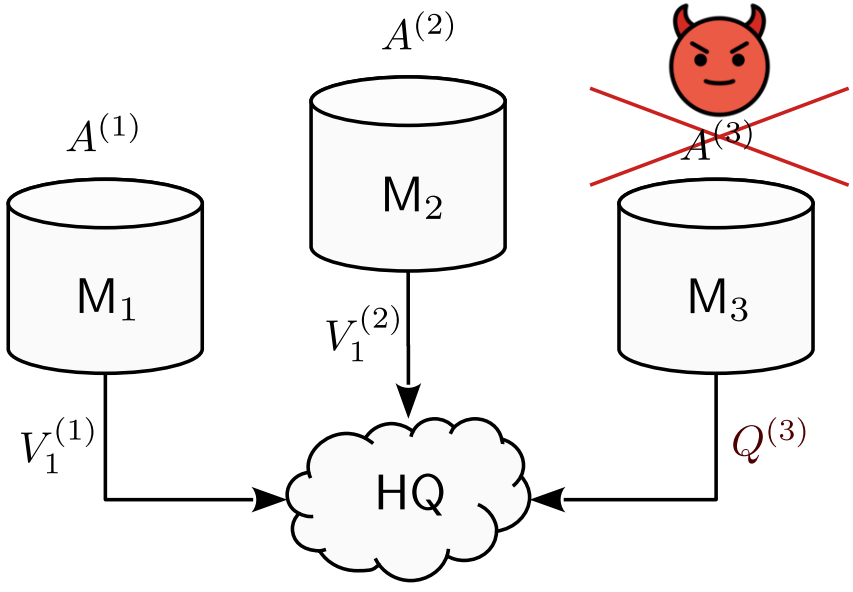
Node failures



Corruption model: unknown index set $\mathcal{I}_{\text{bad}} \subset [m]$ such that:

- $|\mathcal{I}_{\text{bad}}| \leq \alpha m$, for $\alpha \in (0, 1/2)$.
- All nodes $i \in \mathcal{I}_{\text{bad}}$ return *arbitrary*, but *structurally valid* $Q^{(i)} \in O(d, r)$.

Node failures

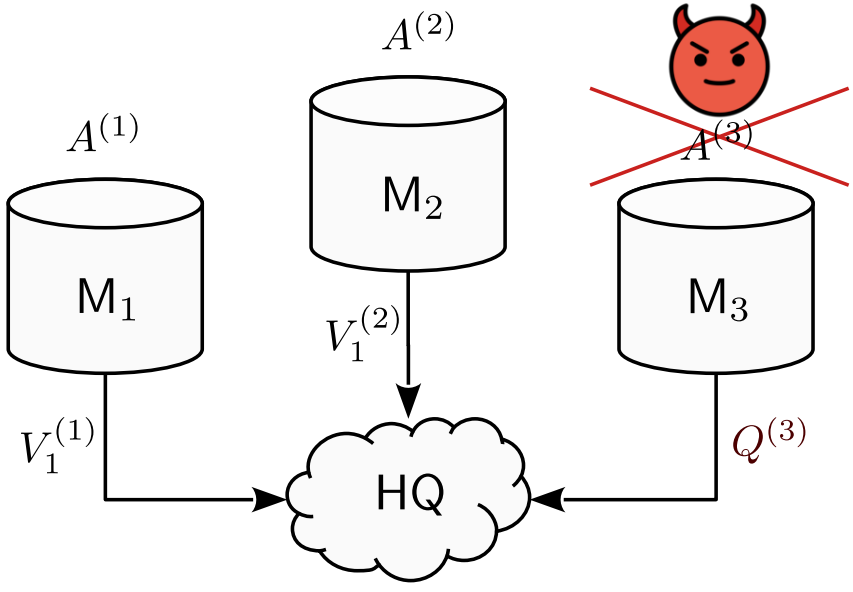


Corruption model: unknown index set $\mathcal{I}_{\text{bad}} \subset [m]$ such that:

- $|\mathcal{I}_{\text{bad}}| \leq \alpha m$, for $\alpha \in (0, 1/2)$.
- All nodes $i \in \mathcal{I}_{\text{bad}}$ return *arbitrary*, but *structurally valid* $Q^{(i)} \in O(d, r)$.

Sources of corruption:

Node failures



Corruption model: unknown index set $\mathcal{I}_{\text{bad}} \subset [m]$ such that:

- $|\mathcal{I}_{\text{bad}}| \leq \alpha m$, for $\alpha \in (0, 1/2)$.
- All nodes $i \in \mathcal{I}_{\text{bad}}$ return *arbitrary*, but *structurally valid* $Q^{(i)} \in O(d, r)$.

Sources of corruption:

- *Silent* / *soft* errors (e.g., insufficient eigensolver tolerance);
- Outliers / corrupted data (e.g., too few samples in machine in PCA);
- Adversarial responses.

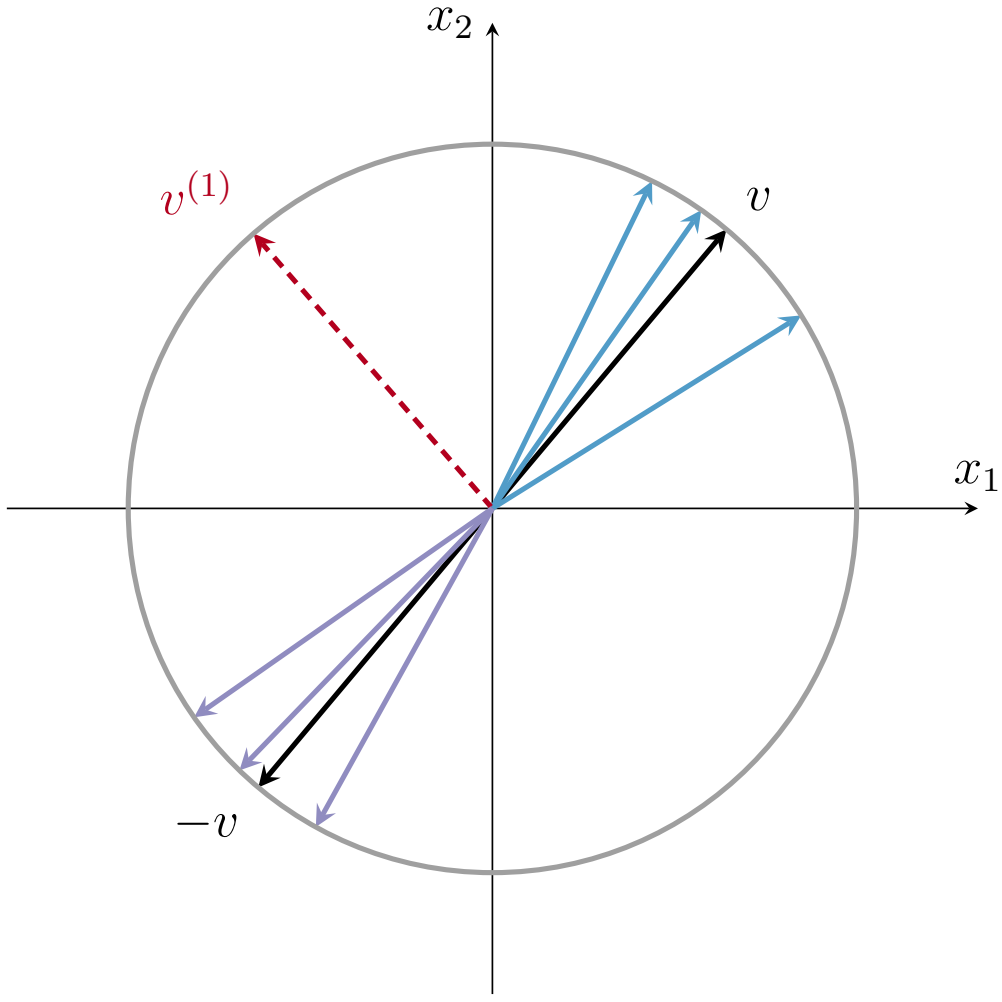
A robust algorithm

Strategy: "robustify" two-stage algorithm from noiseless setting.

A robust algorithm

Strategy: "robustify" two-stage algorithm from noiseless setting.

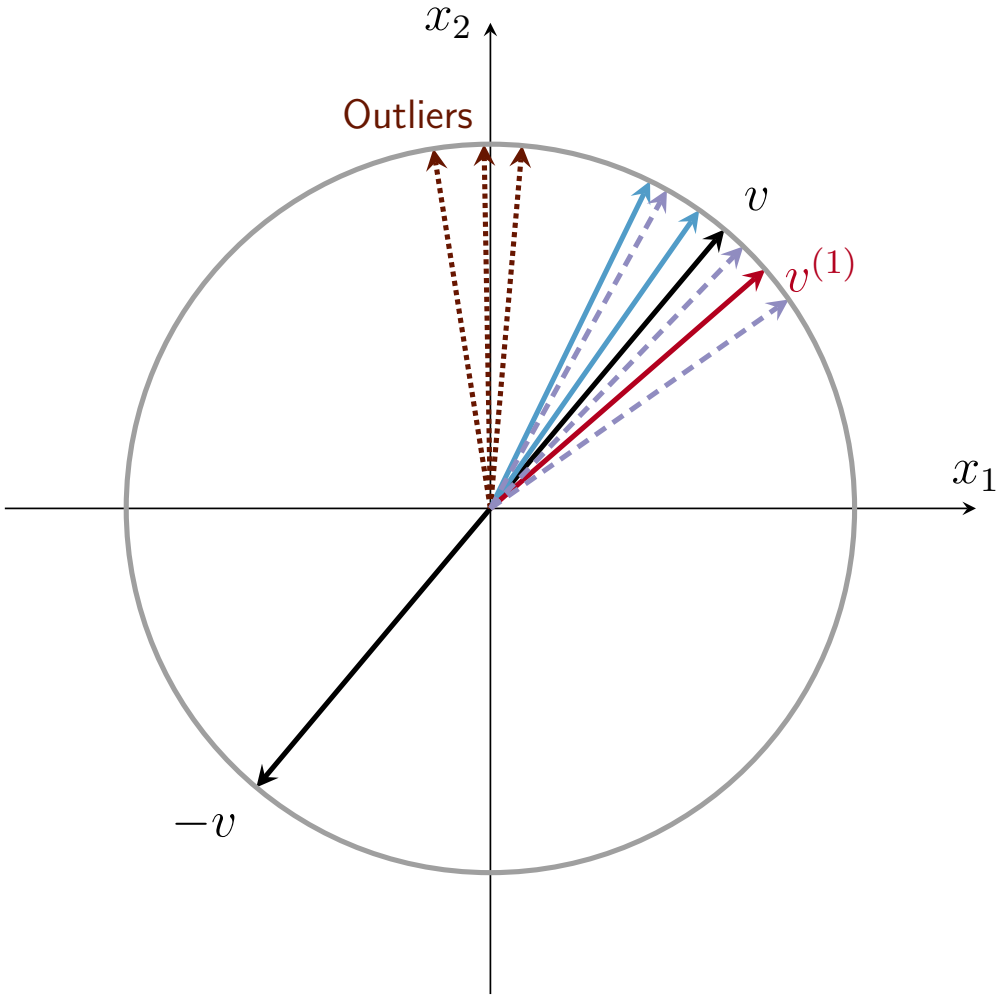
Challenge I: reference solution could be chosen among outliers.



A robust algorithm

Strategy: "robustify" two-stage algorithm from noiseless setting.

Challenge II: Even with "good" reference, we could average over outliers.



A robust algorithm

Strategy: "robustify" two-stage algorithm from noiseless setting.

Algorithm Robust procrustes fixing

1: **Input**: responses $\left\{\widehat{V}^{(i)} \mid i \in [m]\right\}$, corruption fraction α.
2: $V_{\text{ref}} := \texttt{RobustReferenceEstimator}(\left\{\widehat{V}^{(i)}\right\}_{i=1}^{m})$
3: $\left\{\widetilde{V}^{(i)}\right\}_{i=1}^{m} := \texttt{ProcrustesFixing}(\left\{\widehat{V}^{(i)}\right\}_{i=1}^{m}, V_{\text{ref}})$
4: $\tilde{V} := \texttt{RobustMeanEstimation}(\left\{\widetilde{V}^{(i)}\right\}_{i=1}^{m}, \alpha)$.

Step 1: The robust reference estimator

Idea: adapt standard robust distance estimation technique.[1]

Algorithm Robust reference estimation

Input: $Y^{(1)}, \ldots, Y^{(m)} \in O(d, r)$.
for $i = 1, \ldots, m$ **do**
$\epsilon_i := \min \left\{ r \geq 0 \mid \left| \left\{ j : \mathrm{dist}_2(Y^{(i)}, Y^{(j)}) < r \right\} \right| > \frac{m}{2} \right\}$
return $Y^{(i_\star)}$, where $i_\star \in \mathrm{argmin}_{i=1}^{m} \epsilon_i$.

---

[1] Nemirovski & Yudin, '83.

Step 1: The robust reference estimator

Idea: adapt standard robust distance estimation technique.[1]

Algorithm Robust reference estimation

Input: $Y^{(1)}, \ldots, Y^{(m)} \in O(d, r)$.
for $i = 1, \ldots, m$ **do**
$\epsilon_i := \min \left\{ r \geq 0 \mid \left| \left\{ j : \mathrm{dist}_2(Y^{(i)}, Y^{(j)}) < r \right\} \right| > \frac{m}{2} \right\}$
return $Y^{(i_\star)}$, where $i_\star \in \mathrm{argmin}_{i=1}^m \epsilon_i$.

Guarantee: if at least $\frac{m}{2} + 1$ points satisfy $\mathrm{dist}_2(Y^{(i)}, V) \leq \epsilon$, then

$$\mathrm{dist}_2(Y^{(i_\star)}, V) \leq 3\epsilon.$$

[1] Nemirovski & Yudin, '83.

Step 2: Procrustes alignment with robust reference

Idea: argue that when $\mathrm{dist}_2(V_{\mathsf{ref}}, V) < \epsilon$, average over "inliers" has small error.

Algorithm Procrustes fixing

1: **Input**: responses $\left\{\widehat{V}^{(i)}\right\}_{i=1}^{m}$, robust reference V_{ref}.
2: **for** $i = 1, \ldots, m$ **do**
3: $\quad \widehat{V}^{(i)}_{\mathsf{aligned}} := \widehat{V}^{(i)} \cdot \mathrm{argmin}_{U \in O(r)} \left\| V_{\mathsf{ref}} - \widehat{V}^{(i)} U \right\|_{\mathrm{F}}$.
4: **return** $\left\{\widehat{V}^{(i)}_{\mathsf{aligned}}\right\}_{i=1}^{m}$.

Step 2: Procrustes alignment with robust reference

Idea: argue that when $\mathrm{dist}_2(V_{\mathsf{ref}}, V) < \epsilon$, average over "inliers" has small error.

Algorithm Procrustes fixing

1: **Input**: responses $\left\{\widehat{V}^{(i)}\right\}_{i=1}^{m}$, robust reference V_{ref}.
2: **for** $i = 1, \ldots, m$ **do**
3: $\quad \widehat{V}^{(i)}_{\mathsf{aligned}} := \widehat{V}^{(i)} \cdot \mathrm{argmin}_{U \in O(r)} \left\| V_{\mathsf{ref}} - \widehat{V}^{(i)} U \right\|_{\mathrm{F}}$.
4: **return** $\left\{\widehat{V}^{(i)}_{\mathsf{aligned}}\right\}_{i=1}^{m}$.

Guarantee: if $\mathrm{dist}_2(V_{\mathsf{ref}}, V) := \epsilon < \frac{\delta_r(A)}{8}$, then:

$$\left\| \frac{1}{|\mathcal{I}_{\mathsf{good}}|} \sum_{i \in \mathcal{I}_{\mathsf{good}}} \widehat{V}^{(i)}_{\mathsf{aligned}} - V \right\|_2 \lesssim \frac{1}{\delta^2 \, |\mathcal{I}_{\mathsf{good}}|} \sum_{i \in \mathcal{I}_{\mathsf{good}}} \max(\|A_i - A\|_2^2 \, , \|A\|_2^2 \, \epsilon^2) + \frac{1}{\delta} \left\| \frac{1}{|\mathcal{I}_{\mathsf{good}}|} \sum_{i \in \mathcal{I}_{\mathsf{good}}} A_i - A \right\|_2$$

Step 3: Robust mean estimation

Idea: use a spectral filtering algorithm to remove outliers.[1]

Algorithm `Filter`$(S = \{Y_i\}_{i=1}^m, \lambda_{\mathsf{ub}})$

1: Compute empirical mean and covariance:

$$\theta_S := \frac{1}{|S|}\sum_{i\in S} X_i, \quad \Sigma_S := \frac{1}{|S|}\sum_{i\in S}(X_i - \theta_S)(X_i - \theta_S)^{\mathsf{T}}.$$

2: Compute leading eigenpair (λ, v) of Σ_S.
3: **if** $\lambda < \lambda_{\mathsf{ub}}$ **then**
4: **return** θ_S
5: **else**
6: Compute outlier scores $\tau_i := \|(X_i - \theta_S)^{\mathsf{T}} v\|^2$.
7: Sample Z using $\mathbb{P}(Z = X_i) = \frac{\tau_i}{\sum_{j\in S}\tau_j}$.
8: **return** `Filter`$(S \setminus \{Z\}, \lambda_{\mathsf{ub}})$.

---

[1] Kamath et al., 2016

Step 3: Robust mean estimation

Idea: use a spectral filtering algorithm to remove outliers.[1]

Algorithm `Filter`$(S = \{Y_i\}_{i=1}^m, \lambda_{\text{ub}})$

1: Compute empirical mean and covariance:

$$\theta_S := \frac{1}{|S|}\sum_{i\in S} X_i, \quad \Sigma_S := \frac{1}{|S|}\sum_{i\in S}(X_i - \theta_S)(X_i - \theta_S)^\mathsf{T}.$$

2: Compute leading eigenpair (λ, v) of Σ_S.
3: **if** $\lambda < \lambda_{\text{ub}}$ **then**
4: **return** θ_S
5: **else**
6: Compute outlier scores $\tau_i := \|(X_i - \theta_S)^\mathsf{T} v\|^2$.
7: Sample Z using $\mathbb{P}(Z = X_i) = \frac{\tau_i}{\sum_{j\in S}\tau_j}$.
8: **return** `Filter`$(S \setminus \{Z\}, \lambda_{\text{ub}})$.

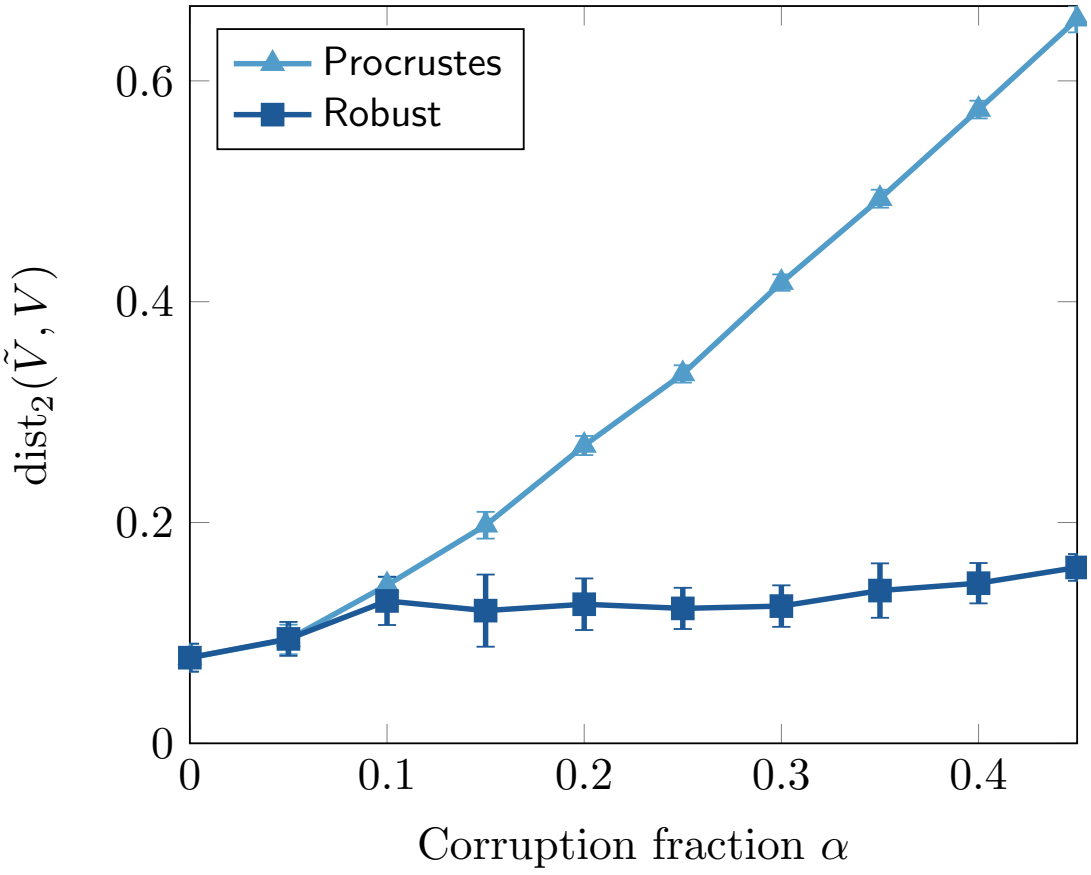
- λ_{ub} is upper bound on (unknown) $\|\Sigma_{\mathcal{I}_{\text{good}}}\|_2$;
- Can be made adaptive to $\|\Sigma_{\mathcal{I}_{\text{good}}}\|_2$ at logarithmic cost.
- Practical implementation: remove point with largest outlier score.

---

[1] Kamath et al., 2016

Experiment

Setup: distributed PCA with $\lfloor \alpha m \rfloor$ responses replaced by a $V_{\text{adv}} \in O(d, r)$.



✗ "Baseline" solution almost orthogonal to V as $\alpha \to 1/2$.

✓ Robust solution: natural breakdown point at $\alpha = 1/2$.

Closing remarks

Main takeaway. Distributed eigenspace estimation algorithm with:

- Only 1 round of communication;
- Only $d \times r$ numbers transmitted per machine;
- Robustness to adversarial, structurally valid node responses.

Closing remarks

Main takeaway. Distributed eigenspace estimation algorithm with:

- Only 1 round of communication;
- Only $d \times r$ numbers transmitted per machine;
- Robustness to adversarial, structurally valid node responses.

Potential extensions:

- Distributed PCA with *heavy-tailed data*?
- What are other interesting corruption models in distributed environments?
- Applications with orthogonal symmetry? (e.g., distributed node embeddings)

Thanks for listening!

arXiv:2009.02436, arXiv:2206.00127