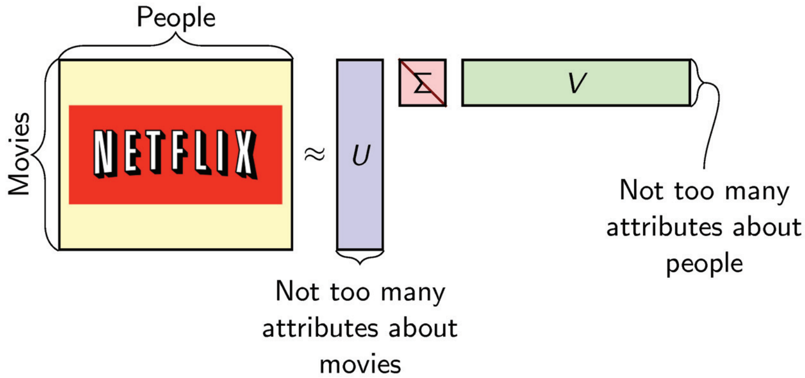
Fast Matrix Completion without the condition number

Moritz Hardt and Mary Wootters

IBM Almaden and University of Michigan→Carnegie Mellon

COLT 2014

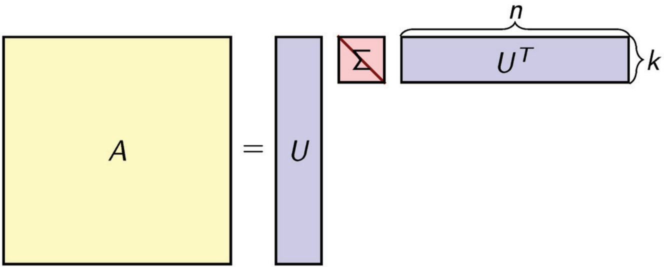
Low rank structure



Of interest:

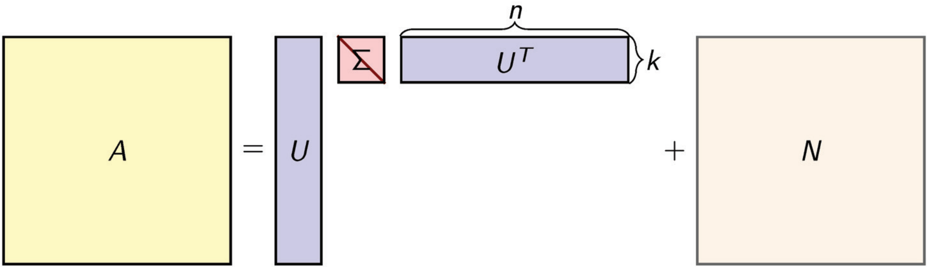
- The original matrix
- U, V

Matrix Completion



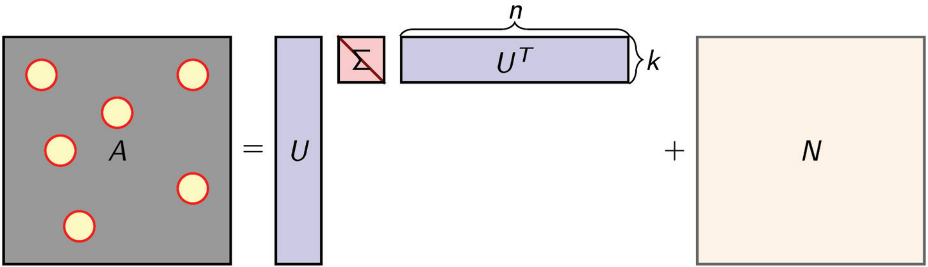
- $A \in \mathbb{R}^{n \times n}$ is (close to) a symmetric rank-k matrix, with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$.

Matrix Completion



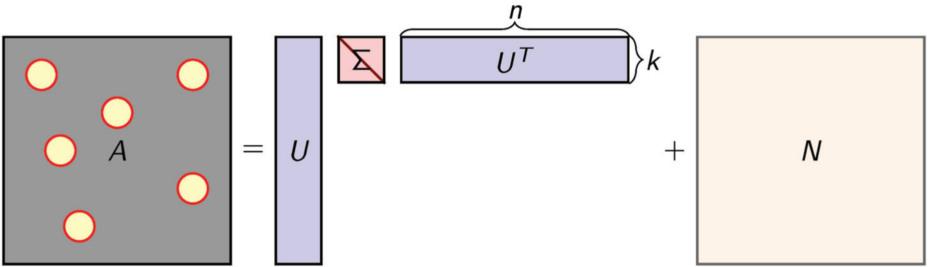
- $A \in \mathbb{R}^{n \times n}$ is (close to) a symmetric rank-k matrix, with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$.

Matrix Completion



- $A \in \mathbb{R}^{n \times n}$ is (close to) a symmetric rank-k matrix, with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$.
- See m entries, $\Omega \subset [n] \times [n]$ of A.

Matrix Completion



- $A \in \mathbb{R}^{n \times n}$ is (close to) a symmetric rank-k matrix, with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$.
- See m entries, $\Omega \subset [n] \times [n]$ of A.
- Goal(s):
  - Recover $\hat{A}$ so that $\left\| A - \hat{A} \right\| \leq \varepsilon \|A\| + \|N\|$.
  - Recover $\hat{U}$, so that $\sin\theta(\hat{U}, U) \leq \varepsilon$.
- Would like: $m \approx kn$, fast algorithm, provable guarantees.

Algorithms that guarantee recovery

- Convex programming:
  [Candès-Recht '09, Candès-Tao '10, Recht et al. '10, Recht '11...].
  - Exact recovery
  - $m = \tilde{O}(nk)$, Running time= $\Omega(n^2)$.

Algorithms that guarantee recovery

- Convex programming:
  [Candès-Recht '09, Candès-Tao '10, Recht et al. '10, Recht '11...].
  - Exact recovery
  - $m = \tilde{O}(nk)$, Running time$= \Omega(n^2)$.
- Alternating Minimization:
  [Keshavan '12, Jain et al. '13, Hardt '13].
  - Approximate (ε) recovery
  - $m = \Omega\left(nk^3\left(\frac{\sigma_1}{\sigma_k}\right)^2 \log(1/\varepsilon)\right)$, running time $= m\mathrm{poly}(k)$.

Algorithms that guarantee recovery

- Convex programming:
  [Candès-Recht '09, Candès-Tao '10, Recht et al. '10, Recht '11...].
  - Exact recovery
  - $m = \tilde{O}(nk)$, Running time= $\Omega(n^2)$.
- Alternating Minimization:
  [Keshavan '12, Jain et al. '13, Hardt '13].
  - Approximate (ε) recovery
  - $m = \Omega\left(nk^3\left(\frac{\sigma_1}{\sigma_k}\right)^2 \log(1/\varepsilon)\right)$, running time $= m\mathrm{poly}(k)$.

Other fast algorithms

- (Online) Frank-Wolfe, (Stochastic) Gradient Descent, ...
  [Mazumder et al. '10, Jaggi-Sulovský '10, Avron et al. '12, Hazan-Kale '12, Recht-Re '13, Hsieh-Olsen '14]

Other fast algorithms

- (Online) Frank-Wolfe, (Stochastic) Gradient Descent, ...
  [Mazumder et al. '10, Jaggi-Sulovský '10, Avron et al. '12, Hazan-Kale '12, Recht-Re '13, Hsieh-Olsen '14]
- Generally guarantee:
  - error on observed entries is small: $\left\| (A - \hat{A})_\Omega \right\|_F \leq \varepsilon$.
  - sample/time complexity like $1/\varepsilon$ (rather than $\log(1/\varepsilon)$).

Other fast algorithms

- (Online) Frank-Wolfe, (Stochastic) Gradient Descent, ...
  [Mazumder et al. '10, Jaggi-Sulovský '10, Avron et al. '12, Hazan-Kale '12, Recht-Re '13, Hsieh-Olsen '14]
- Generally guarantee:
  - error on observed entries is small: $\left\| (A - \hat{A})_\Omega \right\|_F \leq \varepsilon$.
    - ★ Does not imply the sort of reconstruction we're after
  - sample/time complexity like $1/\varepsilon$ (rather than $\log(1/\varepsilon)$).

Other fast algorithms

- (Online) Frank-Wolfe, (Stochastic) Gradient Descent, ...
  [Mazumder et al. '10, Jaggi-Sulovský '10, Avron et al. '12, Hazan-Kale '12, Recht-Re '13, Hsieh-Olsen '14]
- Generally guarantee:
  - error on observed entries is small: $\left\| (A - \hat{A})_\Omega \right\|_F \leq \varepsilon$.
    - ★ Does not imply the sort of reconstruction we're after
  - sample/time complexity like $1/\varepsilon$ (rather than $\log(1/\varepsilon)$).
    - ★ If we want to recover U, this is $1/\sigma_k$.

Either slow or ill-conditioned

- Existing work either

  is slow: Running time $\Omega(n^2)$

  –or–

  depends polynomially on the condition number:

$$m = \Omega\left(n \cdot k \cdot \left(\frac{\sigma_1}{\sigma_k}\right)^2\right).$$

Either slow or ill-conditioned

- Existing work either

  is slow: Running time $\Omega(n^2)$

  –or–

  depends polynomially on the condition number:

$$m = \Omega\left(n \cdot k \cdot \left(\frac{\sigma_1}{\sigma_k}\right)^2\right).$$

- This work: a variant of Alternating Minimization that

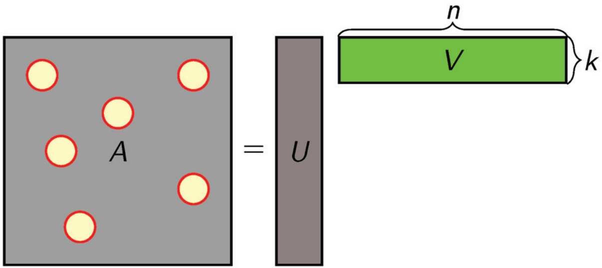
  is fast: Running time is $\tilde{O}\left(\text{poly}(k)m\right)$

  –and–

  depends logarithmically on the condition number:

$$m = \tilde{O}\left(n \cdot k^c \cdot \log\left(\frac{\sigma_1}{\sigma_k + \varepsilon\sigma_1}\right)\right)$$

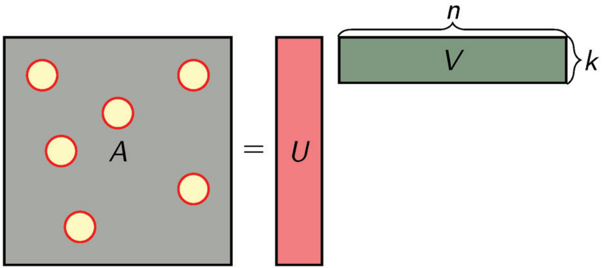
Alternating Minimization



Alternating Minimization:

- Fix U, find V to minimize $\|(A - UV^T)_\Omega\|_F^2$
- Fix V find U to minimize $\|(A - UV^T)_\Omega\|_F^2$

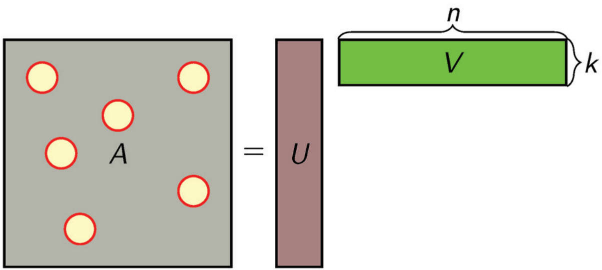
Alternating Minimization



Alternating Minimization:

- Fix U, find V to minimize $\left\|(A - UV^T)_\Omega\right\|_F^2$
- Fix V find U to minimize $\left\|(A - UV^T)_\Omega\right\|_F^2$

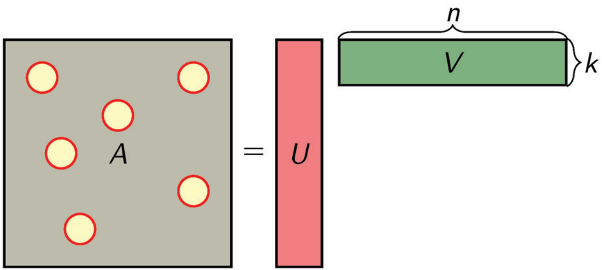
Alternating Minimization



Alternating Minimization:

- Fix U, find V to minimize $\left\|(A - UV^T)_\Omega\right\|_F^2$
- Fix V find U to minimize $\left\|(A - UV^T)_\Omega\right\|_F^2$

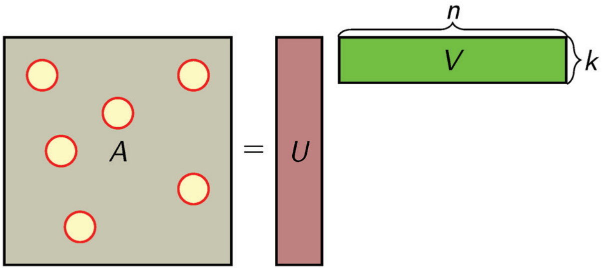
Alternating Minimization



Alternating Minimization:

- Fix U, find V to minimize $\left\|(A - UV^T)_\Omega\right\|_F^2$
- Fix V find U to minimize $\left\|(A - UV^T)_\Omega\right\|_F^2$

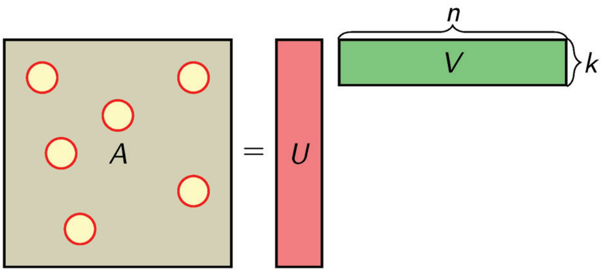
Alternating Minimization



Alternating Minimization:

- Fix U, find V to minimize $\left\|(A - UV^T)_\Omega\right\|_F^2$
- Fix V find U to minimize $\left\|(A - UV^T)_\Omega\right\|_F^2$

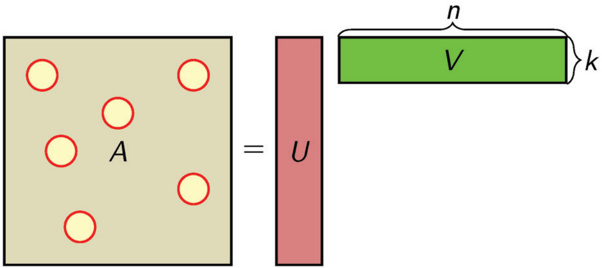
Alternating Minimization



Alternating Minimization:

- Fix U, find V to minimize $\left\| (A - UV^T)_\Omega \right\|_F^2$
- Fix V find U to minimize $\left\| (A - UV^T)_\Omega \right\|_F^2$

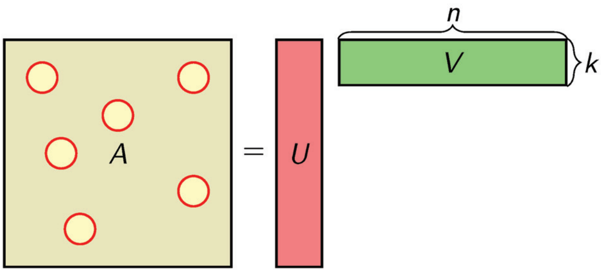
Alternating Minimization



Alternating Minimization:

- Fix U, find V to minimize $\left\|(A - UV^T)_\Omega\right\|_F^2$
- Fix V find U to minimize $\left\|(A - UV^T)_\Omega\right\|_F^2$

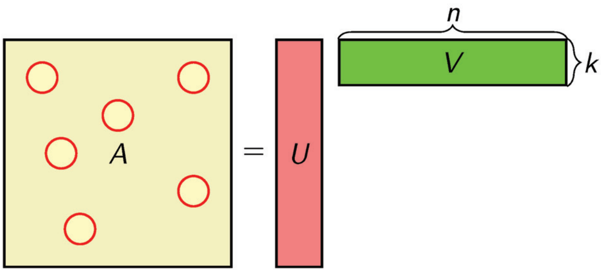
Alternating Minimization



Alternating Minimization:

- Fix U, find V to minimize $\left\|(A - UV^T)_\Omega\right\|_F^2$
- Fix V find U to minimize $\left\|(A - UV^T)_\Omega\right\|_F^2$

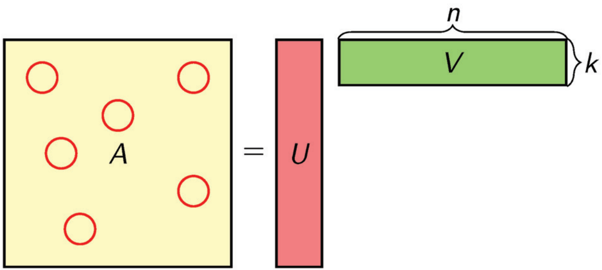
Alternating Minimization



Alternating Minimization:

- Fix U, find V to minimize $\|(A - UV^T)_\Omega\|_F^2$
- Fix V find U to minimize $\|(A - UV^T)_\Omega\|_F^2$

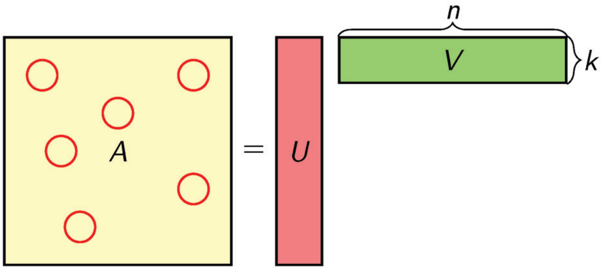
Alternating Minimization



Alternating Minimization:

- Fix U, find V to minimize $\left\|(A - UV^T)_\Omega\right\|_F^2$
- Fix V find U to minimize $\left\|(A - UV^T)_\Omega\right\|_F^2$

Alternating Minimization



Alternating Minimization:

- Fix U, find V to minimize $\left\|(A - UV^T)_\Omega\right\|_F^2$
- Fix V find U to minimize $\left\|(A - UV^T)_\Omega\right\|_F^2$

Stop after about $\log(1/\varepsilon)$ steps.

Why the condition number?

- Typically, AM is initialized by taking the SVD of A_Ω.

Why the condition number?

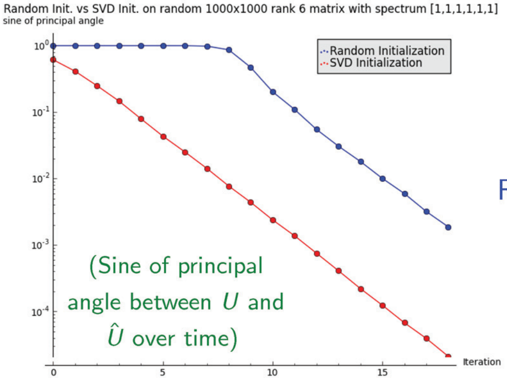
- Typically, AM is initialized by taking the SVD of A_Ω.
  - To prove AM converges, need to start "close enough."

Why the condition number?

- Typically, AM is initialized by taking the SVD of A_Ω.
  - To prove AM converges, need to start "close enough."
  - In practice, there is some effect of initialization.
    (Although AM does usually eventually converge from a random start.)

Why the condition number?

- Typically, AM is initialized by taking the SVD of A_Ω.
  - To prove AM converges, need to start "close enough."
  - In practice, there is some effect of initialization.
    (Although AM does usually eventually converge from a random start.)

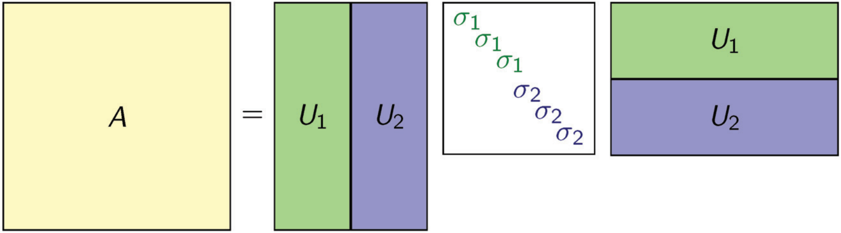


Random initialization.

SVD initialization.

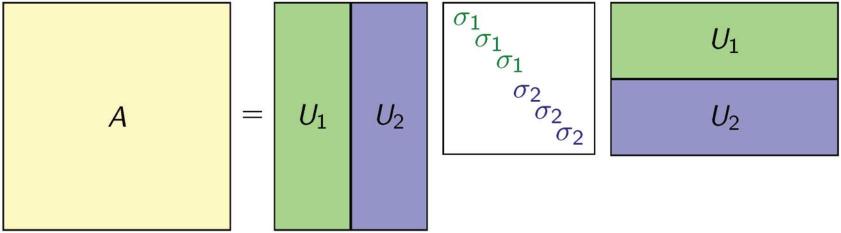
SVD depends on the condition number

- Suppose $\sigma_1 \gg \sigma_2$.



SVD depends on the condition number

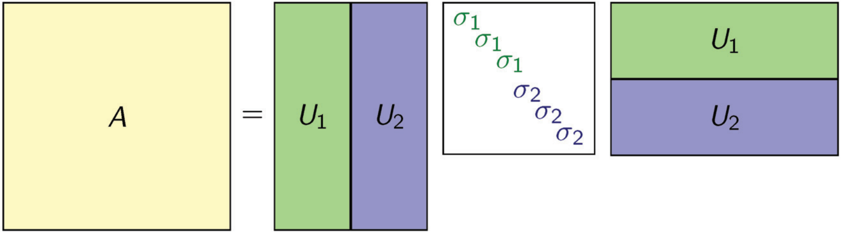
- Suppose $\sigma_1 \gg \sigma_2$.



- To approximate $[U_1|U_2]$ via SVD, need $|\Omega| \approx \left(\frac{\sigma_1}{\sigma_2}\right)^2 kn$ samples.
  - That depends on the condition number.

SVD depends on the condition number

- Suppose $\sigma_1 \gg \sigma_2$.

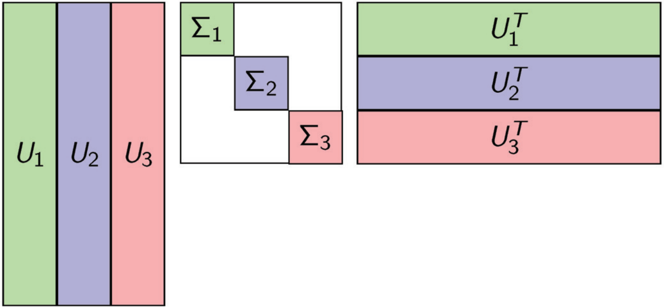


- To approximate $[U_1|U_2]$ via SVD, need $|\Omega| \approx \left(\frac{\sigma_1}{\sigma_2}\right)^2 kn$ samples.
  - That depends on the condition number.
- To approximate U_1 via SVD, need $|\Omega| \approx kn$ samples.
  - U_2 may as well have not been initialized: same problem as before.

First try: Deflation

Say the spectrum of A is $[\sigma_1, \sigma_1, \sigma_1, \sigma_2, \sigma_2, \sigma_2, \sigma_3, \sigma_3, \sigma_3, \ldots]$, with associated subspaces spanned by U_1, U_2, $U_3 \ldots$

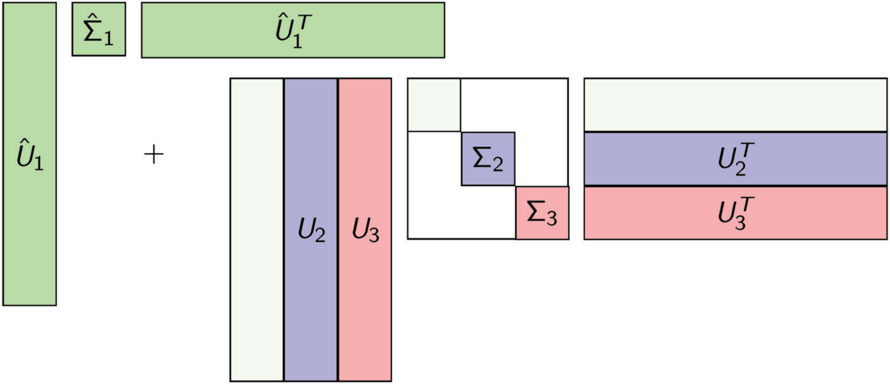
- Estimate $\hat{U}_1, \hat{\sigma_1}$ using SVD-initialized AM ($m \approx kn$).
- Subtract off $\hat{U}_1 \hat{\Sigma}_1 \hat{U}_1{}^T$.



First try: Deflation

Say the spectrum of A is $[\sigma_1, \sigma_1, \sigma_1, \sigma_2, \sigma_2, \sigma_2, \sigma_3, \sigma_3, \sigma_3, \ldots]$, with associated subspaces spanned by U_1, U_2, $U_3 \ldots$

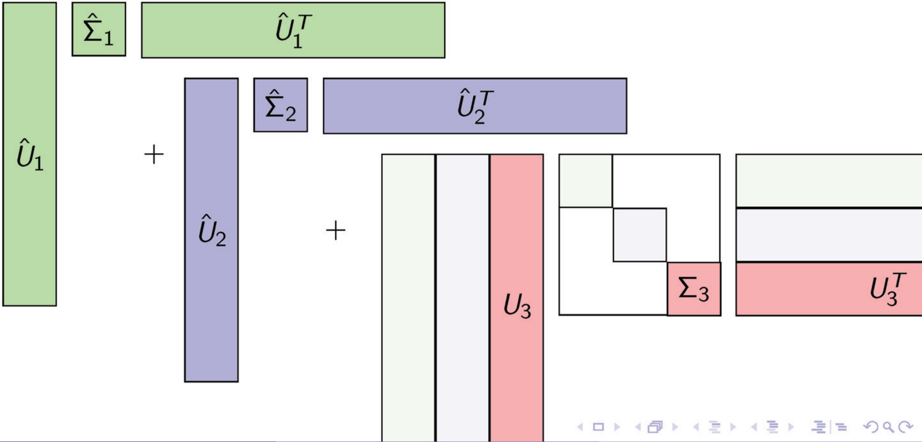
- Estimate $\hat{U}_1, \hat{\sigma_1}$ using SVD-initialized AM ($m \approx kn$).
- Subtract off $\hat{U}_1 \hat{\Sigma}_1 \hat{U}_1{}^T$.



First try: Deflation

Say the spectrum of A is $[\sigma_1, \sigma_1, \sigma_1, \sigma_2, \sigma_2, \sigma_2, \sigma_3, \sigma_3, \sigma_3, \ldots]$, with associated subspaces spanned by U_1, U_2, $U_3 \ldots$

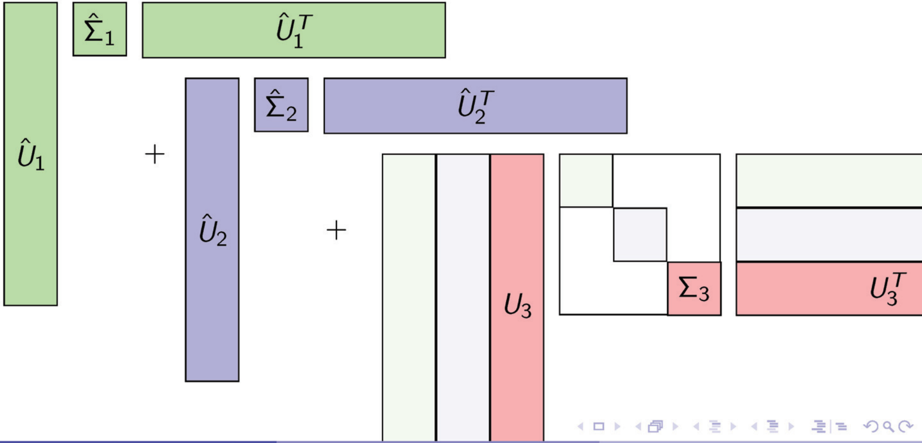
- Estimate $\hat{U}_2, \hat{\sigma_2}$ using SVD-initialized AM ($m \approx kn$).
- Subtract off $\hat{U}_2 \hat{\Sigma}_2 \hat{U}_2^T$.



First try: Deflation

Say the spectrum of A is $[\sigma_1, \sigma_1, \sigma_1, \sigma_2, \sigma_2, \sigma_2, \sigma_3, \sigma_3, \sigma_3, \ldots]$, with associated subspaces spanned by U_1, U_2, $U_3 \ldots$

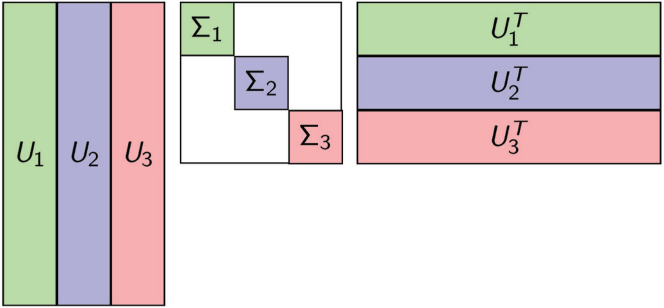
Etc...



Deflation doesn't work

Say the spectrum of A is $[\sigma_1, \sigma_1, \sigma_1, \sigma_2, \sigma_2, \sigma_2, \sigma_3, \sigma_3, \sigma_3, \ldots]$, with associated subspaces spanned by U_1, U_2, $U_3 \ldots$

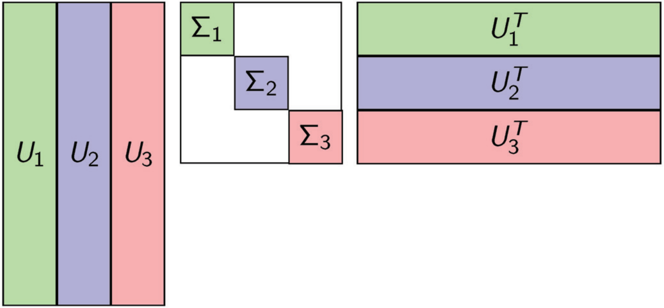
What actually happens:



Deflation doesn't work

Say the spectrum of A is $[\sigma_1, \sigma_1, \sigma_1, \sigma_2, \sigma_2, \sigma_2, \sigma_3, \sigma_3, \sigma_3, \ldots]$, with associated subspaces spanned by U_1, U_2, $U_3 \ldots$

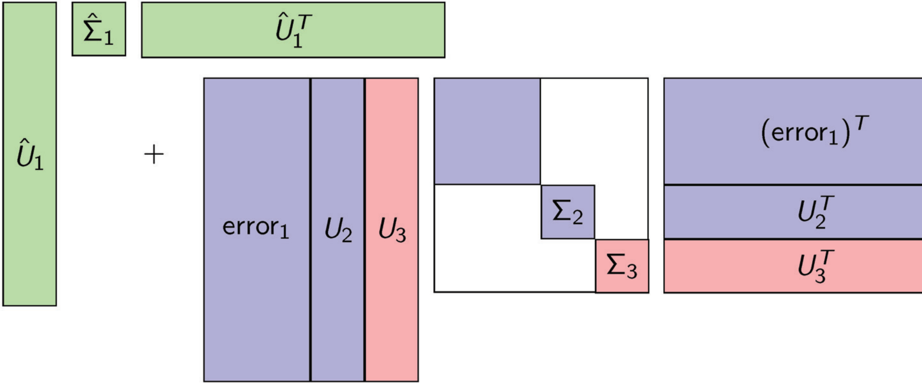
- Estimate $\hat{U}_1, \hat{\sigma_1}$ using SVD-initialized AM ($m \approx kn$).
- Subtract off $\hat{U}_1 \hat{\Sigma}_1 \hat{U}_1^{\,T}$: error is on the order of σ_2.



Deflation doesn't work

Say the spectrum of A is $[\sigma_1, \sigma_1, \sigma_1, \sigma_2, \sigma_2, \sigma_2, \sigma_3, \sigma_3, \sigma_3, \ldots]$, with associated subspaces spanned by U_1, U_2, $U_3 \ldots$

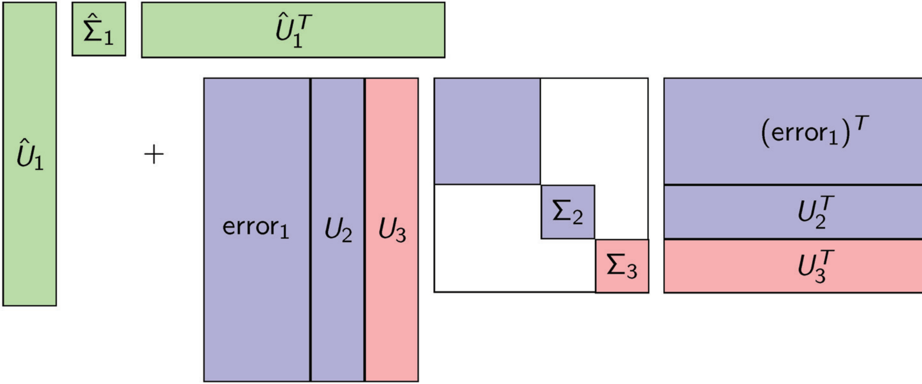
- Estimate $\hat{U}_1, \hat{\sigma_1}$ using SVD-initialized AM ($m \approx kn$).
- Subtract off $\hat{U}_1 \hat{\Sigma}_1 \hat{U}_1{}^T$: error is on the order of σ_2.



Deflation doesn't work

Say the spectrum of A is $[\sigma_1, \sigma_1, \sigma_1, \sigma_2, \sigma_2, \sigma_2, \sigma_3, \sigma_3, \sigma_3, \ldots]$, with associated subspaces spanned by U_1, U_2, $U_3 \ldots$

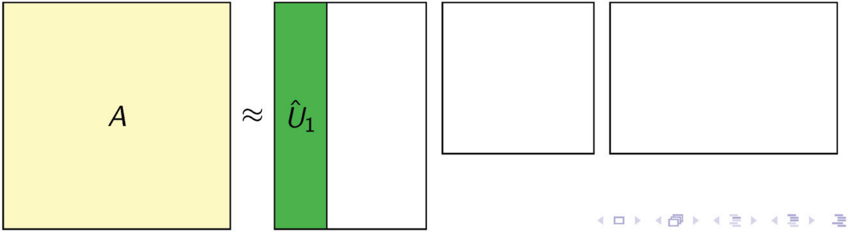
- Estimate the stuff of magnitude σ_2 using SVD-initialized AM.
- But now the rank is much bigger :(



Instead: Soft Deflation

Say the spectrum of A is $[\sigma_1, \sigma_1, \sigma_1, \sigma_2, \sigma_2, \sigma_2, \sigma_3, \sigma_3, \sigma_3, \ldots]$, with associated subspaces spanned by U_1, U_2, $U_3 \ldots$

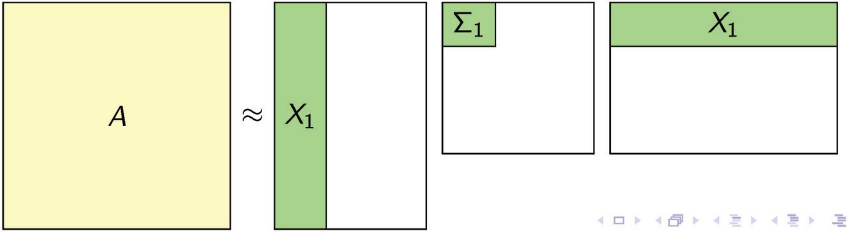
- Use SVD on A_Ω to form initial guess $\hat{U}_1$ for U_1.
- Run AM starting from $\hat{U}_1$ to get $A_1 = X_1 \hat{\Sigma}_1 X_1^T \approx U_1 \Sigma_1 U_1^T$.



Instead: Soft Deflation

Say the spectrum of A is $[\sigma_1, \sigma_1, \sigma_1, \sigma_2, \sigma_2, \sigma_2, \sigma_3, \sigma_3, \sigma_3, \ldots]$, with associated subspaces spanned by U_1, U_2, $U_3 \ldots$

- Use SVD on A_Ω to form initial guess $\hat{U}_1$ for U_1.
- Run AM starting from $\hat{U}_1$ to get $A_1 = X_1 \hat{\Sigma}_1 X_1^T \approx U_1 \Sigma_1 U_1^T$.

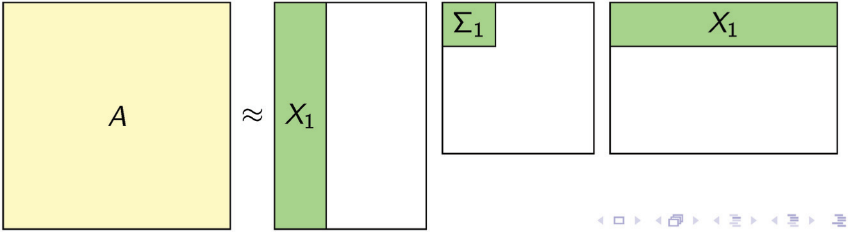


Instead: Soft Deflation

Say the spectrum of A is $[\sigma_1, \sigma_1, \sigma_1, \sigma_2, \sigma_2, \sigma_2, \sigma_3, \sigma_3, \sigma_3, \ldots]$, with associated subspaces spanned by U_1, U_2, $U_3 \ldots$

$\sigma_1 \sin \Theta(\hat{U}_1, U_1) \approx \sigma_1/10$
$\sigma_1 \sin \Theta(X_1, U_1) \approx \sigma_2/10$

- Use SVD on A_Ω to form initial guess $\hat{U}_1$ for U_1.
- Run AM starting from $\hat{U}_1$ to get $A_1 = X_1 \hat{\Sigma}_1 X_1^T \approx U_1 \Sigma_1 U_1^T$.



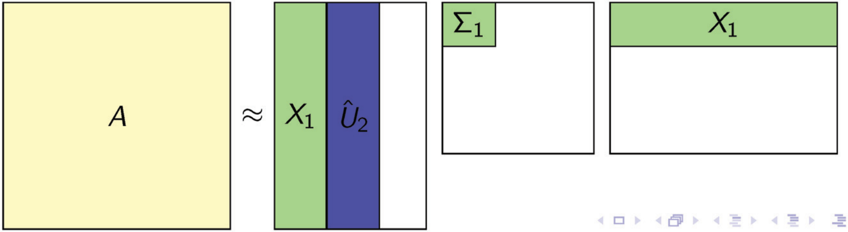
Instead: Soft Deflation

Say the spectrum of A is $[\sigma_1, \sigma_1, \sigma_1, \sigma_2, \sigma_2, \sigma_2, \sigma_3, \sigma_3, \sigma_3, \ldots]$, with associated subspaces spanned by U_1, U_2, $U_3 \ldots$

$\sigma_1 \sin \Theta(\hat{U}_1, U_1) \approx \sigma_1/10$
$\sigma_1 \sin \Theta(X_1, U_1) \approx \sigma_2/10$

- Use SVD on A_Ω to form initial guess $\hat{U}_1$ for U_1.
- Run AM starting from $\hat{U}_1$ to get $A_1 = X_1 \hat{\Sigma}_1 X_1^T \approx U_1 \Sigma_1 U_1^T$.

- Use SVD on $(A - A_1)_\Omega$ to guess $\hat{U}_2$ for U_2.
- Run AM from $[X_1 | \hat{U}_2]$ to get $A_2 = X_2 \hat{\Sigma}_2 X_2^T$.



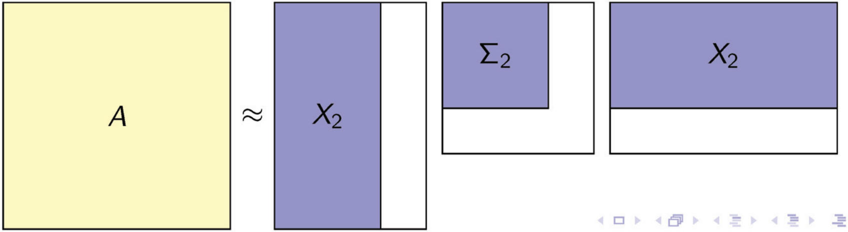
Instead: Soft Deflation

Say the spectrum of A is $[\sigma_1, \sigma_1, \sigma_1, \sigma_2, \sigma_2, \sigma_2, \sigma_3, \sigma_3, \sigma_3, \ldots]$, with associated subspaces spanned by U_1, U_2, $U_3 \ldots$

$\sigma_1 \sin \Theta(\hat{U}_1, U_1) \approx \sigma_1/10$
$\sigma_1 \sin \Theta(X_1, U_1) \approx \sigma_2/10$

- Use SVD on A_Ω to form initial guess $\hat{U}_1$ for U_1.
- Run AM starting from $\hat{U}_1$ to get $A_1 = X_1 \hat{\Sigma}_1 X_1^T \approx U_1 \Sigma_1 U_1^T$.

- Use SVD on $(A - A_1)_\Omega$ to guess $\hat{U}_2$ for U_2.
- Run AM from $[X_1 | \hat{U}_2]$ to get $A_2 = X_2 \hat{\Sigma}_2 X_2^T$.



Instead: Soft Deflation

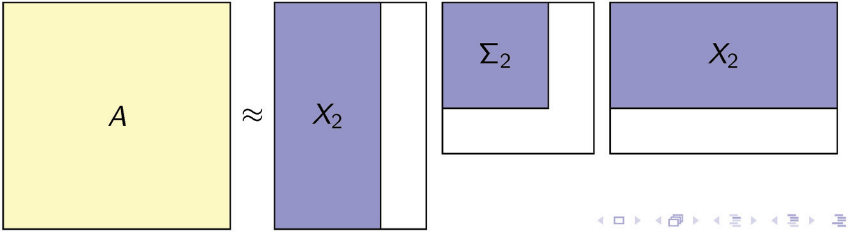
Say the spectrum of A is $[\sigma_1, \sigma_1, \sigma_1, \sigma_2, \sigma_2, \sigma_2, \sigma_3, \sigma_3, \sigma_3, \ldots]$, with associated subspaces spanned by U_1, U_2, $U_3 \ldots$

$\sigma_1 \sin \Theta(\hat{U}_1, U_1) \approx \sigma_1/10$
$\sigma_1 \sin \Theta(X_1, U_1) \approx \sigma_2/10$

- Use SVD on A_Ω to form initial guess $\hat{U}_1$ for U_1.
- Run AM starting from $\hat{U}_1$ to get $A_1 = X_1 \hat{\Sigma}_1 X_1^T \approx U_1 \Sigma_1 U_1^T$.

$\sigma_2 \sin \Theta(\hat{U}_2, U_2) \approx \sigma_2/10$
$\sigma_2 \sin \Theta(X_2, U_2) \approx \sigma_3/10$

- Use SVD on $(A - A_1)_\Omega$ to guess $\hat{U}_2$ for U_2.
- Run AM from $[X_1 | \hat{U}_2]$ to get $A_2 = X_2 \hat{\Sigma}_2 X_2^T$.



Instead: Soft Deflation

Say the spectrum of A is $[\sigma_1, \sigma_1, \sigma_1, \sigma_2, \sigma_2, \sigma_2, \sigma_3, \sigma_3, \sigma_3, \ldots]$, with associated subspaces spanned by U_1, U_2, $U_3 \ldots$

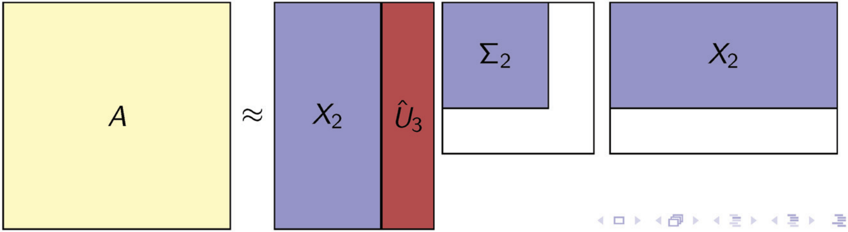
- Use SVD on A_Ω to form initial guess $\hat{U}_1$ for U_1.
- Run AM starting from $\hat{U}_1$ to get $A_1 = X_1 \hat{\Sigma}_1 X_1^T \approx U_1 \Sigma_1 U_1^T$.

$\sigma_1 \sin\Theta(\hat{U}_1, U_1) \approx \sigma_1/10$
$\sigma_1 \sin\Theta(X_1, U_1) \approx \sigma_2/10$

- Use SVD on $(A - A_1)_\Omega$ to guess $\hat{U}_2$ for U_2.
- Run AM from $[X_1 | \hat{U}_2]$ to get $A_2 = X_2 \hat{\Sigma}_2 X_2^T$.

$\sigma_2 \sin\Theta(\hat{U}_2, U_2) \approx \sigma_2/10$
$\sigma_2 \sin\Theta(X_2, U_2) \approx \sigma_3/10$

- Use SVD on $(A - A_2)_\Omega$ to form initial guess $\hat{U}_3$ for U_3.
- Run AM starting from $[X_2 | \hat{U}_3]$ to get $A_3 = X_3 \hat{\Sigma}_3 X_3^T$.



Instead: Soft Deflation

Say the spectrum of A is $[\sigma_1, \sigma_1, \sigma_1, \sigma_2, \sigma_2, \sigma_2, \sigma_3, \sigma_3, \sigma_3, \ldots]$, with associated subspaces spanned by U_1, U_2, $U_3 \ldots$

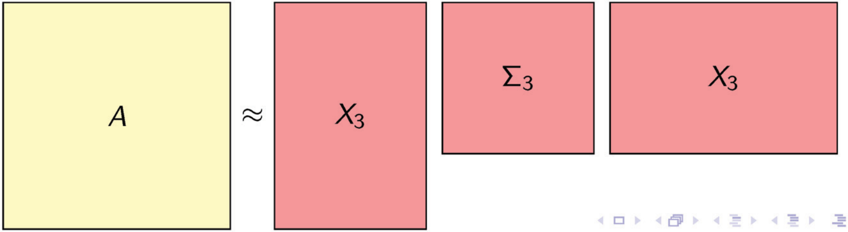
$\sigma_1 \sin\Theta(\hat{U}_1, U_1) \approx \sigma_1/10$
$\sigma_1 \sin\Theta(X_1, U_1) \approx \sigma_2/10$

- Use SVD on A_Ω to form initial guess $\hat{U}_1$ for U_1.
- Run AM starting from $\hat{U}_1$ to get $A_1 = X_1 \hat{\Sigma}_1 X_1^T \approx U_1 \Sigma_1 U_1^T$.

$\sigma_2 \sin\Theta(\hat{U}_2, U_2) \approx \sigma_2/10$
$\sigma_2 \sin\Theta(X_2, U_2) \approx \sigma_3/10$

- Use SVD on $(A - A_1)_\Omega$ to guess $\hat{U}_2$ for U_2.
- Run AM from $[X_1 | \hat{U}_2]$ to get $A_2 = X_2 \hat{\Sigma}_2 X_2^T$.

- Use SVD on $(A - A_2)_\Omega$ to form initial guess $\hat{U}_3$ for U_3.
- Run AM starting from $[X_2 | \hat{U}_3]$ to get $A_3 = X_3 \hat{\Sigma}_3 X_3^T$.



Theorem (Exact)

Suppose

- *Each entry in Ω is included independently with probability p*
- *A is* incoherent
- *$A = UV^T$ is exactly rank k.*

There is some

$$m \lesssim nk^c \log\left(\frac{\sigma_1}{\sigma_k + \varepsilon\sigma_1}\right)$$

so that if $\mathbb{E}|\Omega| = pn^2 \geq m$, then SoftDeflate returns X, Y so that

$$\|A - XY\| \leq \varepsilon \|A\| \qquad .$$

Theorem (Noisy)

Suppose

- *Each entry in Ω is included independently with probability p*
- *A is* incoherent
- $A = UV^T + N$.

There is some

$$m \lesssim n \left(\frac{k}{\gamma_k} \right)^c \log \left(\frac{\sigma_1}{\sigma_k + \varepsilon \sigma_1} \right) \left(1 + \left(\frac{\|N\|_F}{\varepsilon \sigma_1} \right)^2 \right)^2$$

so that if $\mathbb{E}|\Omega| = pn^2 \geq m$, then SoftDeflate returns X, Y so that

$$\|A - XY\| \leq \varepsilon \|A\| + (1 + o(1)) \|N\|.$$

Theorem (Noisy)

Suppose

- *Each entry in* Ω *is included independently with probability* p
- A *is* incoherent
- $A = UV^T + N$.

There is some

$$m \lesssim n \left(\frac{k}{\gamma_k}\right)^c \log\left(\frac{\sigma_1}{\sigma_k + \varepsilon\sigma_1}\right) \left(1 + \left(\frac{\|N\|_F}{\varepsilon\sigma_1}\right)^2\right)^2$$

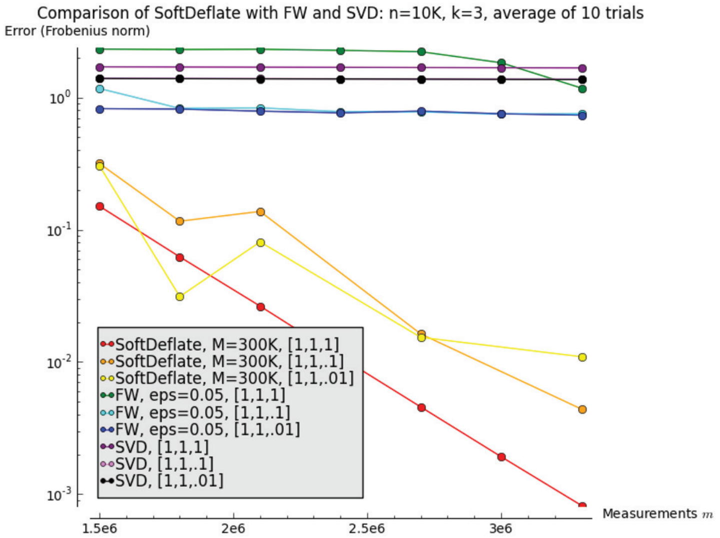
so that if $\mathbb{E}|\Omega| = pn^2 \geq m$, *then SoftDeflate returns* X, Y *so that*

$$\|A - XY\| \leq \varepsilon \|A\| + (1 + o(1)) \|N\|.$$

$$\gamma_k := 1 - \frac{\sigma_k}{\sigma_{k+1}} = \begin{cases} 1 & N = 0 \\ \text{big} & \|N\| \approx \sigma_k \end{cases}$$

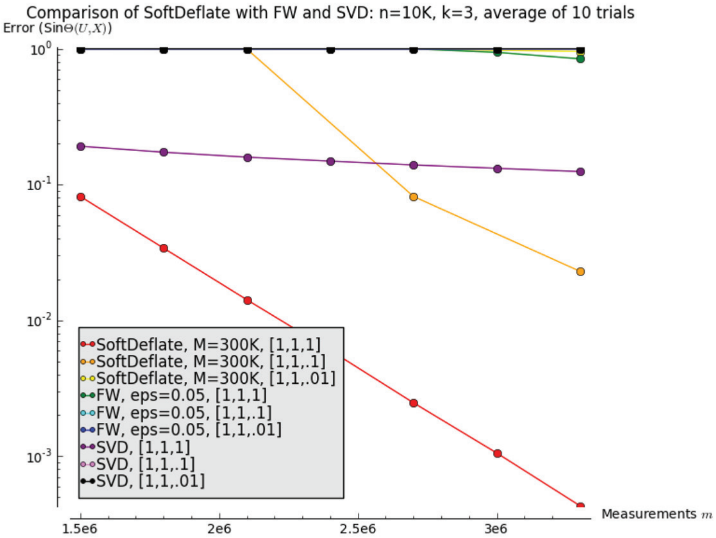
Some pictures

Comparing SoftDeflate to FW, SVD



Some pictures

Comparing SoftDeflate to FW, SVD



Summary

- New "Soft Deflation" variant of Alternating Minimization
- Fast:

  runtime linear in n
- Works on ill-conditioned matrices:

  sample and time complexity is logarithmic in σ_1/σ_k.

Summary

- New "Soft Deflation" variant of Alternating Minimization
- Fast:

  runtime linear in n
- Works on ill-conditioned matrices:

  sample and time complexity is logarithmic in σ_1/σ_k.

- Open Questions:
  - How badly does Alternating Minimization itself actually depend on the condition number? On a "typical" matrix?
  - (How much) can you reduce the power k in our analysis?

The end

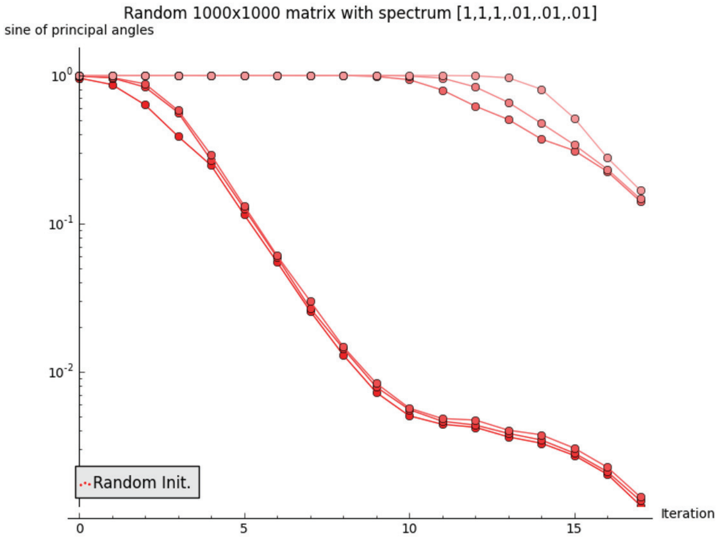
Thanks!

Under the rug

- How do we know where the “gaps” are?
  - Use a good enough approximation to detect this with the SVD.
- The gaps could be pretty small.
  - If $\sigma_i/\sigma_{i+1} = (1 - 1/\sqrt{k})$, then $\sigma_1/\sigma_k \approx e^{\sqrt{k}}$ is still big.
  - This makes us pay extra factor(s) of k.
- Need to ensure incoherence between the iterations.
  - Carefully truncate entry-wise before/after SVD.
- Need to ensure incoherence during Alternating Minimization.
  - Borrow from [Hardt'13]: Add some noise to “smooth” AM, and take some medians to control outliers.

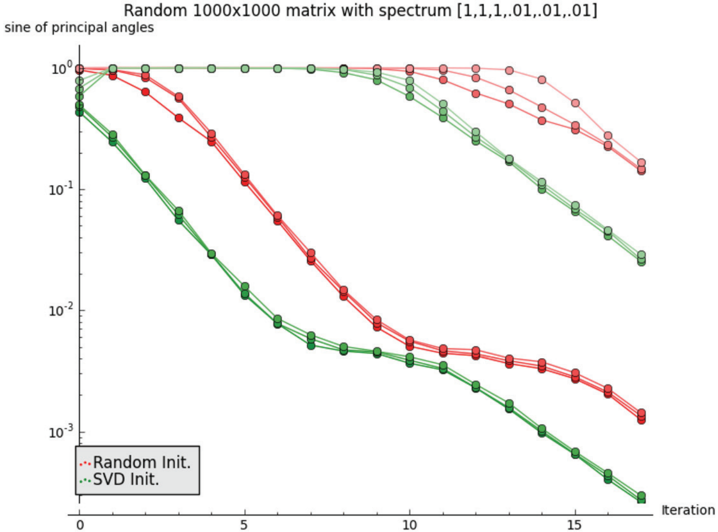
Is SoftDeflate better than AM in practice?

Plotting all 6 principal angles as the algorithms run



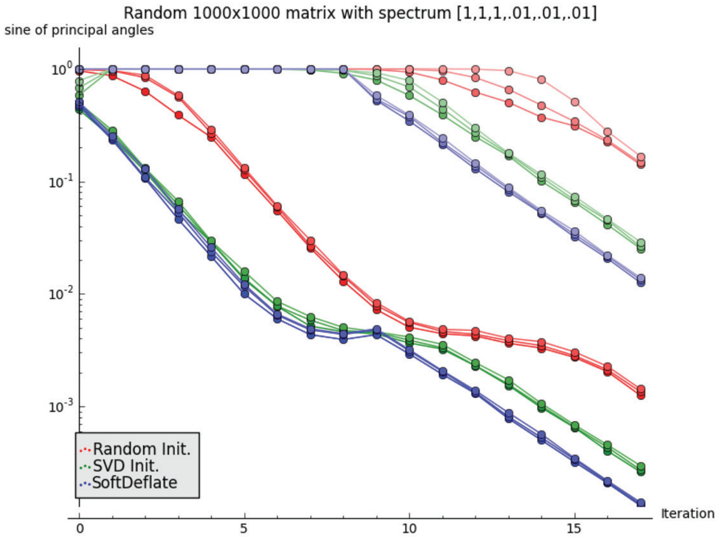
Is SoftDeflate better than AM in practice?

Plotting all 6 principal angles as the algorithms run

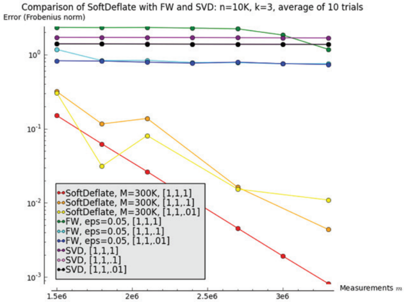


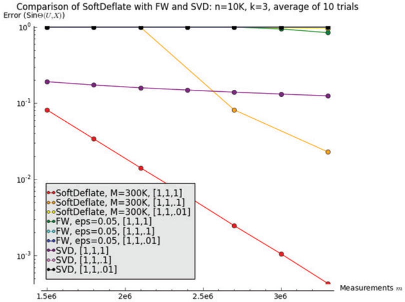
Is SoftDeflate better than AM in practice?

Plotting all 6 principal angles as the algorithms run

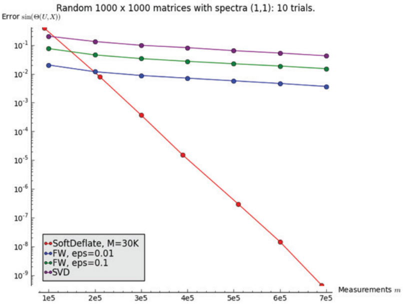


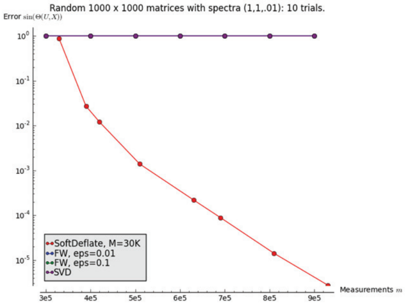
How does this compare to FW? Or just taking the SVD of the observations?





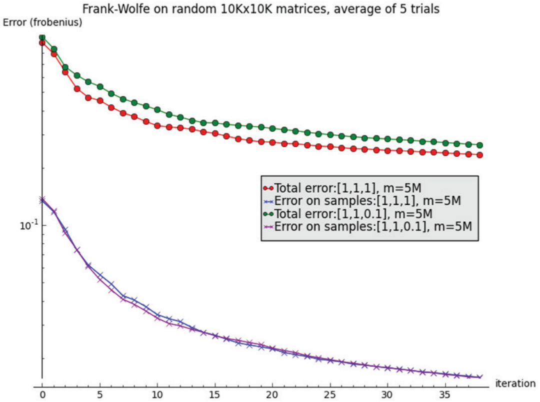
Does FW/SVD get better with more observations?





What about some of the provable guarantees for, say, Frank-Wolfe?

- Running time depends on ε like $1/\varepsilon$, not like $\log(1/\varepsilon)$, so if we want to recover all of U, need $\varepsilon < \sigma_k/\sigma_1$.
- Convergence guarantees are on observed entries, not on whole matrix.



Basic proof idea*

- Maintain the inductive hypothesis

$$\forall i, \sigma_i \sin \Theta(U_i, X_{t-1}[i]) \leq \frac{\sigma_t}{100}$$

$*$ Hiding many details.

Basic proof idea*

- Maintain the inductive hypothesis

$$\forall i, \sigma_i \sin \Theta(U_i, X_{t-1}[i]) \leq \frac{\sigma_t}{100}$$

$X_{t-1} \approx [U_1 | \cdots | U_{t-1}]$.
In particular, $A - X_{t-1} Y_{t-1}^T$ is a good approximation of the leftovers.

$*$ Hiding many details.

Basic proof idea*

- Maintain the inductive hypothesis

$$\forall i, \sigma_i \sin \Theta(U_i, X_{t-1}[i]) \leq \frac{\sigma_t}{100}$$

$X_{t-1} \approx [U_1 | \cdots | U_{t-1}]$. In particular, $A - X_{t-1} Y_{t-1}^T$ is a good approximation of the leftovers.

- Then SVD will find $\hat{U}_t$ so that

$$\sigma_t \sin \Theta(U_t, \hat{U}_t) \leq \frac{1}{100}$$

$*$ Hiding many details.

Basic proof idea*

- Maintain the inductive hypothesis

$$\forall i, \sigma_i \sin\Theta(U_i, X_{t-1}[i]) \leq \frac{\sigma_t}{100}$$

$X_{t-1} \approx [U_1|\cdots|U_{t-1}]$.
In particular, $A - X_{t-1}Y_{t-1}^T$ is a good approximation of the leftovers.

- Then SVD will find $\hat{U}_t$ so that

$$\sigma_t \sin\Theta(U_t, \hat{U}_t) \leq \frac{1}{100}$$

Since the part of $A - X_{t-1}Y_{t-1}^T$ associated with U_t has a flat spectrum, we don't pay for the condition number.

$*$ Hiding many details.

Basic proof idea*

- Maintain the inductive hypothesis

$$\forall i, \sigma_i \sin\Theta(U_i, X_{t-1}[i]) \leq \frac{\sigma_t}{100}$$

$X_{t-1} \approx [U_1|\cdots|U_{t-1}]$.
In particular, $A - X_{t-1}Y_{t-1}^T$ is a good approximation of the leftovers.

- Then SVD will find $\hat{U}_t$ so that

$$\sigma_t \sin\Theta(U_t, \hat{U}_t) \leq \frac{1}{100}$$

Since the part of $A - X_{t-1}Y_{t-1}^T$ associated with U_t has a flat spectrum, we don't pay for the condition number.

- Then AM started at $[X_{t-1}|\hat{U}_t]$ will find X_t with

$$\sigma_t \sin\Theta([U_1|\cdots|U_t], X_t) \leq \frac{\sigma_{t+1}}{100}$$

$*$ Hiding many details.

Basic proof idea*

- Maintain the inductive hypothesis

$$\forall i, \sigma_i \sin\Theta(U_i, X_{t-1}[i]) \leq \frac{\sigma_t}{100}$$

$X_{t-1} \approx [U_1 | \cdots | U_{t-1}]$.
In particular, $A - X_{t-1} Y_{t-1}^T$ is a good approximation of the leftovers.

- Then SVD will find $\hat{U}_t$ so that

$$\sigma_t \sin\Theta(U_t, \hat{U}_t) \leq \frac{1}{100}$$

Since the part of $A - X_{t-1} Y_{t-1}^T$ associated with U_t has a flat spectrum, we don't pay for the condition number.

- Then AM started at $[X_{t-1} | \hat{U}_t]$ will find X_t with

$$\sigma_t \sin\Theta([U_1 | \cdots | U_t], X_t) \leq \frac{\sigma_{t+1}}{100}$$

AM converges until it "hits" the next part of the spectrum, σ_{t+1}

$*$ Hiding many details.

How well does this scale?

SoftDeflate vs. AltMin on random 10Kx10K matrix with spectrum [1,1,1,1,.001,.001,.001,.001], 500K samples per iteration