Frequent Directions and its applicatoin in Efficient Anomaly Detection - a practice

謝幸娟 (Hsing-Chuan Hsieh)
connie0915549431@hotmail.com

1. Introduction

Source paper:

Ghashami, M., Liberty, E., Phillips, J. M., & Woodruff, D. P. (2016). Frequent directions: Simple and deterministic matrix sketching. *SIAM Journal on Computing*, *45*(5), 1762-1792.

1. Introduction

- **Context**

➢ Streaming paradigm of large data set $A \in R^{n \times d}$

- Data items arriving in an arbitrary order, are processed, and then never seen again.

➢ Only a small amount of memory is available at any given time

➢ In data mining tasks, Low rank approximations for A are used

  ➢ E.g. of techs: PCA, SVD, k-means clustering, latent semantic indexing (LSI)

- **Computational challenge**

➢ Computing full SVD takes $O(d^2)$ of memory/time. → unacceptable if d ↑

1. Introduction

- **Solutions of Approximating Streaming**

1. Make matrix rows become available over time
2. Only a single pass over the data is possible

- **Purpose**

➢ To compute a **significantly smaller** sketch matrix B such that $A \approx B$ or $A^T A \approx B^T B$.

  - This lets future computations be performed on B rather than on A without losing much in precision.

➢ Here introduce a new matrix sketching technique: Frequent Directions

2. Frequent Directions

Source paper:

Ghashami, M., Liberty, E., Phillips, J. M., & Woodruff, D. P. (2016). Frequent directions: Simple and deterministic matrix sketching. *SIAM Journal on Computing*, *45*(5), 1762-1792.

2.1. Item frequency approximation

- Frequent directions is an extension of a well-known algorithm for approximating item frequencies (aka `FrequentItems`) in streams.

- Goal: for $A = [a_1; \ldots; a_n]$, $a_i \in R^d$
  1. To approximate for all j their frequency $f_j = |\{a_i \in A \,|\, a_i = a_j\}|$
     - f_j : the number of times a_j appears in the stream.
     - Suppose $j = [d]$ $(i.e. j = \{1, 2, \ldots, d\})$ j<=d

       → $O(d)$ of space for all exact f_j counters

       → $O(d^2)$ of space for all a_j's
  2. Interested in using less space while producing approximate frequencies $\widehat{f_j}$
     - i.e., only store the top ℓ $(< d)$ a_j's and $\widehat{f_j}$'s

2.2. Algorithm Simulation of **FrequentItems**

- Suppose for $A = [a_1; \dots; a_n]$, there's j<=10 distinct $a_{(j)}$ appearing in the stream, say, $a_{(1)}, a_{(2)}, \dots, a_{(10)}$
  - i.e., for $a_i \in A \rightarrow a_i \in a_{(j)},\ 1 \leq j \leq 10,\ for\ \forall i$

- Initialization: set $\ell = 5 < d$
  - Only record 5 itmes (save space)

Items	Counters
$a_{(1)}$	0
$a_{(2)}$	0
$a_{(3)}$	0
$a_{(4)}$	0
$a_{(5)}$	0

At least one of the items is mapped to a counter of value zero.

2.2. Algorithm Simulation of **`FrequentItems`**

- Start: read a row of a_i at a time (i=1,…,n)
- Case 1:
  - If a_1 belongs to an item on the table, say, $a_1 \in a_{(3)}$ $\rightarrow \widehat{f_{(3)}} \mathrel{+}= 1$

a_1

Items	Counters
$a_{(1)}$	0
$a_{(2)}$	0
$a_{(3)}$	1
$a_{(4)}$	0
$a_{(5)}$	0

Counter is incremented

2.2. Algorithm Simulation of **FrequentItems**

- Start: read a row of a_i at a time (i=1,…,n)
- Case 2:
  - If a_2 doesn't belong to any item on the table, say, $a_1 \in a_{(8)} \rightarrow \widehat{f_{(8)}} \mathrel{+}= 1$

a_2

Items	Counters
$a_{(1)}$	0
$a_{(2)}$	0
$a_{(3)}$	1
$a_{(4)}$	0
$a_{(8)}$	1

it replaces one of the items mapping to zero value with the new item

2.2. Algorithm Simulation of `FrequentItems`

- The above process is continued until the invariant is violated
  - ie., all counters of $\ell = 5$ items ≥ 1.
- Case 3:

At this point, all counts are decreased by the same amount until at least one item maps to a zero value.

Items	Counters	
$a_{(1)}$	4	-2
$a_{(5)}$	0	-2
$a_{(3)}$	1	-2
$a_{(10)}$	0	-2
$a_{(8)}$	2	-2

2.2. Algorithm Simulation of `FrequentItems`

- The final values in the map give approximate frequencies $\widehat{f_j}$; unmapped j implies $\widehat{f_j} = 0$
  - Error bound: $0 \le f_j - \widehat{f_j} \le \frac{n}{\ell}$ $for\ \forall j \in [d]$

- Connection to matrix sketching- Frequent Directions

1. Items $a_{(j)}, for\ j \in [\ell]$ ⟺ Eigenvectors v_j

2. $\widehat{f_j}$ ⟺ Eigenvalues σ_j^2

by defining $\widehat{f_j} = \|Bv_j\|^2 (= \sigma_j^2)$

2.3. Algorithm of FrequentDirections

Algorithm 1 FrequentDirections

Input: $\ell, A \in R^{n \times d}$
$B \leftarrow 0^{\ell \times d}$
for $i \in 1, \ldots, n$ **do**
  $B_\ell \leftarrow a_i$ # ith row of A replaces (all-zeros) ℓth row of B
  $[U, \Sigma, V] \leftarrow \mathsf{svd}(B)$
  $\delta \leftarrow \sigma_\ell^2$
  $B \leftarrow \sqrt{\Sigma^2 - \delta I_\ell} \cdot V^T$ # The last row of B is again zero
end for
return B

- SVD running time ~ $O(d\ell^2)$. (dominate running time of Algorithm 1)
- Total running time ~ $O(d\ell^2)$

2.3. Algorithm of `FrequentDirections`

- **Error bound analysis**

THEOREM 3.1. *Let $B \in \mathbb{R}^{\ell \times d}$ be the sketch of* FREQUENTDIRECTIONS *on an input matrix $A \in \mathbb{R}^{n \times d}$. Then for any unit vector $x \in \mathbb{R}^d$ it holds that*

$$0 \leq \|Ax\|^2 - \|Bx\|^2 \leq \|A - A_k\|_F^2 / (\ell - k).$$

Or equivalently

$$\|A^T A - B^T B\|_2 \leq \|A - A_k\|_F^2 / (\ell - k) \quad \text{and} \quad A^T A \succeq B^T B.$$

This holds for all $k < \ell$ including $k = 0$ where we define A_0 as the $n \times d$ all-zeros matrix. Note that setting $\ell = \lceil 1/\varepsilon + k \rceil$ yields an error of $\varepsilon \|A - A_k\|_F^2$ using $O(d\ell) = O(dk + d/\varepsilon)$ space.

  - $A_k = U_k \Sigma_k V_k^T$, rank-k approximation of A

2.3. Algorithm of FrequentDirections

Algorithm 2 FAST-FREQUENTDIRECTIONS

Input: $\ell, A \in R^{n \times d}$

$B \leftarrow$ all-zeros matrix $\in R^{2\ell \times d}$

for $i \in 1, \ldots, n$ **do**

  Insert a_i into a zero valued row of B

  if B has no zero valued rows **then**

    $[U, \Sigma, V] \leftarrow \mathsf{svd}(B)$

    $\delta \leftarrow \sigma_\ell^2$

    $B \leftarrow \sqrt{\max(\Sigma^2 - I_\ell \delta, 0)} \cdot V^T$ # The last $\ell + 1$ rows of B are zero valued.

  end if

end for

return B

- SVD of B is computed only $n/(\ell + 1)$ times
- Total running time: $O(nd\ell^2) \rightarrow O\left((\frac{n}{\ell})d\ell^2\right) = O(nd\ell)$

3. A Practice

Practice writing algorithm of

1. Frequent Directions (Algorithm)
2. Rank-k Leverage scores
3. Rank-k Projection scores

3.1. Introduction

- Purpose

➢Replicate the algorithm 1 from Sharan, V., et al (2018)

Algorithm 1: Algorithm to approximate anomaly scores using Frequent Directions

Input: Choice of k, sketch size ℓ for Frequent Directions [26]

First Pass:

Use Frequent Directions to compute a sketch $\tilde{\mathbf{A}} \in \mathbb{R}^{\ell \times d}$

SVD:

Compute the top k right singular vectors of $\tilde{\mathbf{A}}^T \tilde{\mathbf{A}}$ or SVD of $\tilde{A} = UVD^T$

Second Pass: *As each row* $a_{(i)}$ *streams in,*

Use estimated right singular vectors to compute leverage scores and projection distances

- "Frequent Direction" use algorithm 1 from Ghashami, M., et al (2016)

- Evaluation

➢Compare estimated AD scores (L_k, T_k) derived from A and sketch B($\tilde{A}$)

3.2. Data

- Data source:
  - p53 mutants
  - Here I only use a subset of p53 mutants dataset

Data	Size (n × d)
p53 mutants	16772 × 5408
Subset used: A	5681 × 5408

➢ We first have to decide parameters k and ℓ.

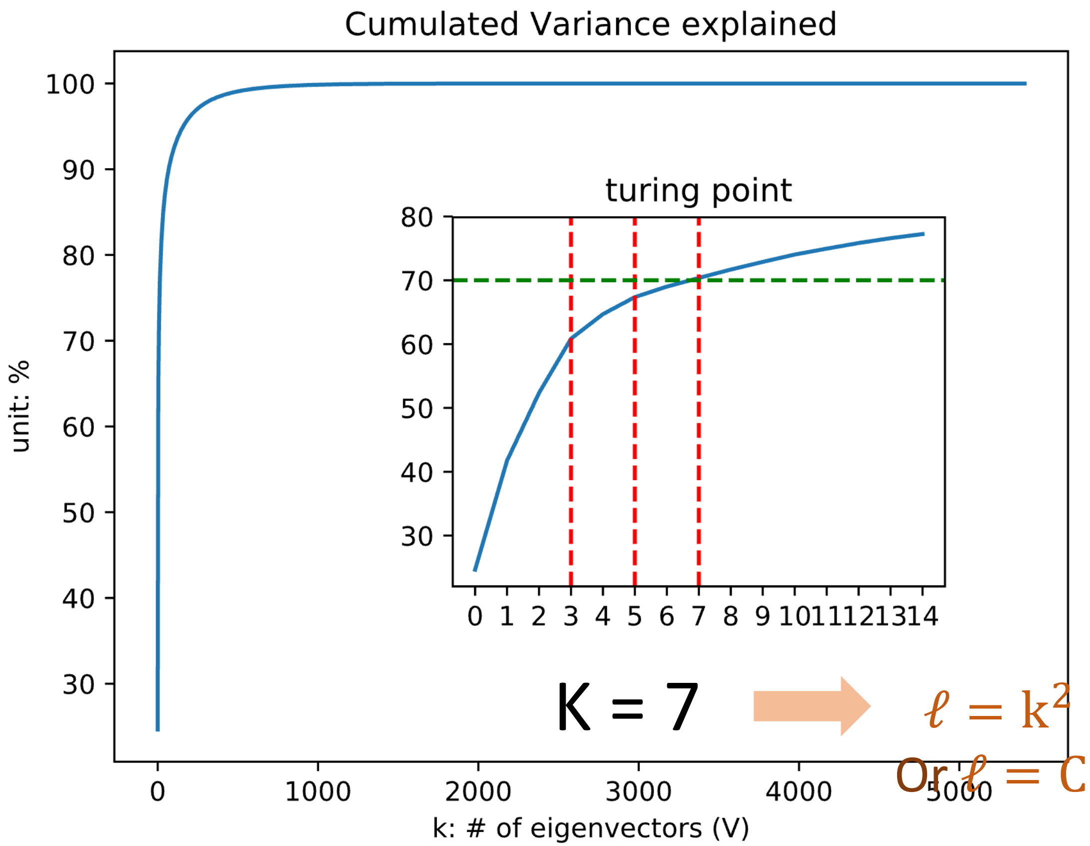
1. k: # of eigenvevtors $v_i \in R^d$
2. ℓ : # of rows of B to sketch matrix A ($\ell \ll n$)

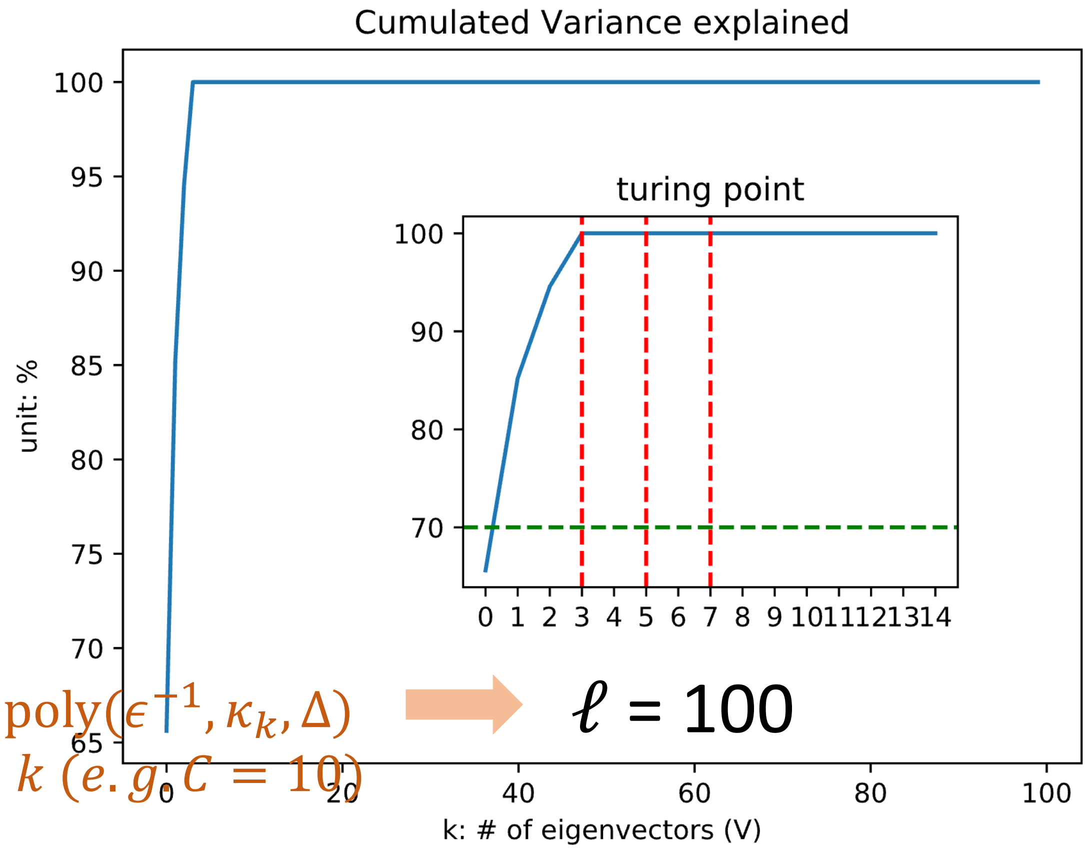
3.2. Data

- Data exploration: SVD for deciding k (# of eigenvectors)

Origin: $A \in R^{n \times d}$

Sketch: B ($\tilde{A}$) $\in R^{l \times d}$



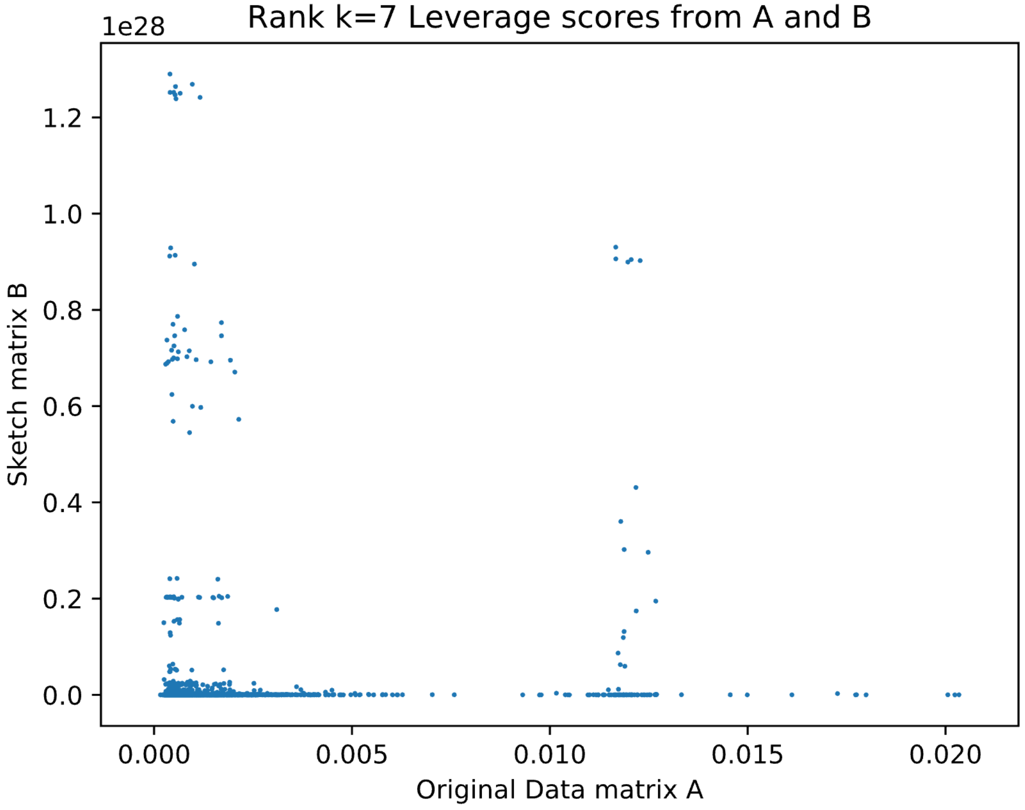


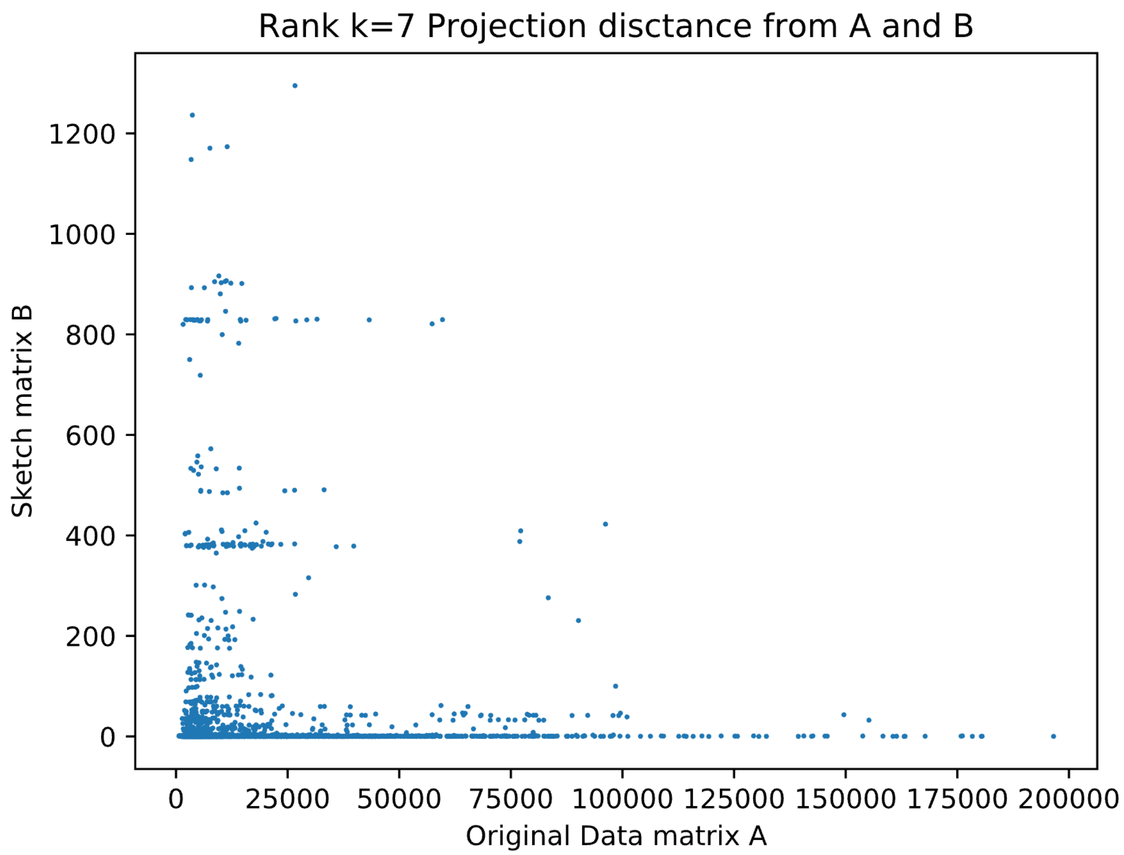
K = 7 → $\ell = \mathrm{k}^2 \cdot \mathrm{poly}(\epsilon^{-1}, \kappa_k, \Delta)$ → ℓ = 100

Or $\ell = C \cdot k$ $(e.g.\ C = 10)$

3.4. Results

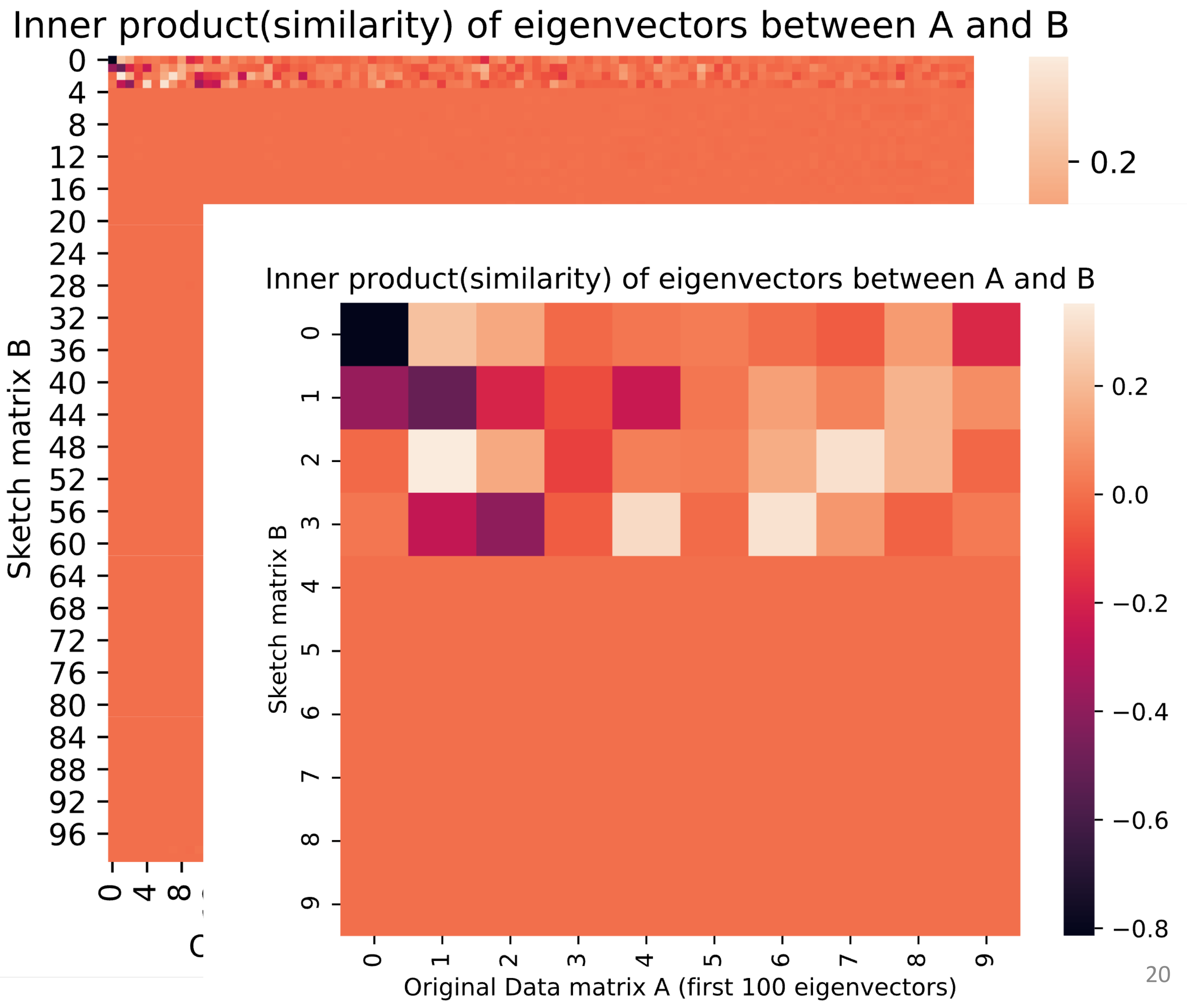
- Comparison of estimated AD scores (L_7, T_7) derived from A and sketch B given ℓ=100





3.4. Results

- Comparison of first 100 eigenvectors derived from A and sketch B
- By similarity matrix $V_B^T V_A$



3.5. Conclusion

- Check if k and ℓ reasonable?
- Do they follow the theorem below:

- Our main results say that given $\mu > 0$ and a $\boxed{\text{(k, Δ)-separated}}$ matrix A $\in R^{n\times d}$ with top singular value σ_1, any sketch $\tilde{A} \in R^{l\times d}$ satisfying

$$\|A^T A - \tilde{A}^T \tilde{A}\| \leq \boxed{\mu\sigma_1^2}$$

or a sketch $\tilde{A} \in R^{n\times l}$ satisfying

$$\|AA^T - \tilde{A}\tilde{A}^T\| \leq \mu\sigma_1^2$$

can be used to approximate rank *k* leverage scores and the projection distance from the principal *k*-dimensional subspace.

老師回饋

- 下週任務：
  - 試著用osPCA分析Data(e.g., p53 mutants) 比較他們的ground truth (Sharan, V., Gopalan, P., & Wieder, U. (2018)) 與osPCA結果差異
- Paper至少一篇讀清楚
- 找paper的Github(Github上搜尋)
- osPCA未來研究
  1. osPCA error bound 證明（老師有想法但未證，未來可研究）
  2. 擴充至>= 2 eigenvectors →計算 row data projection到k個軸（e.g.k=2）的weights(e.g. w1, w2)=> AD scores for each eigenvector,算所有AD score加權整合，如果落到ball(正常範圍)外，則視為異常。