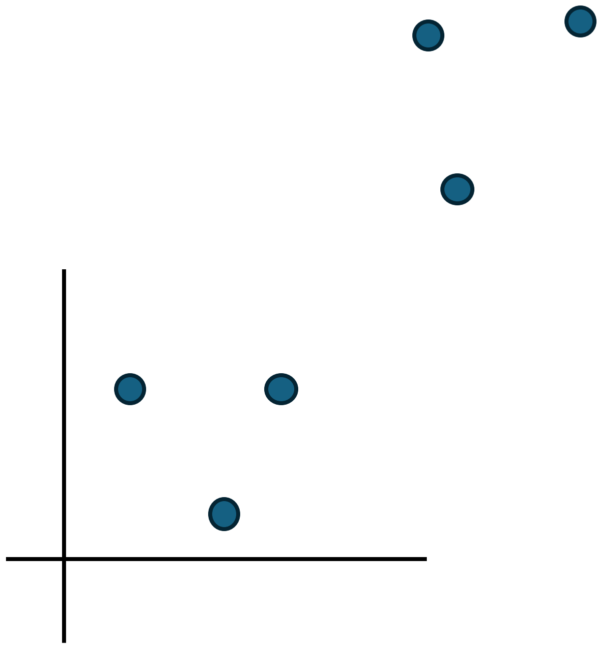
Pros and Cons of NIPALS

Aafan Ahmad Toor, Aleksei Berdiuzhenko, Nicole Quattrini, Aria Alinejad

Creds to: [mda-b202 - PCA NIPALS algorithm (youtube.com)]

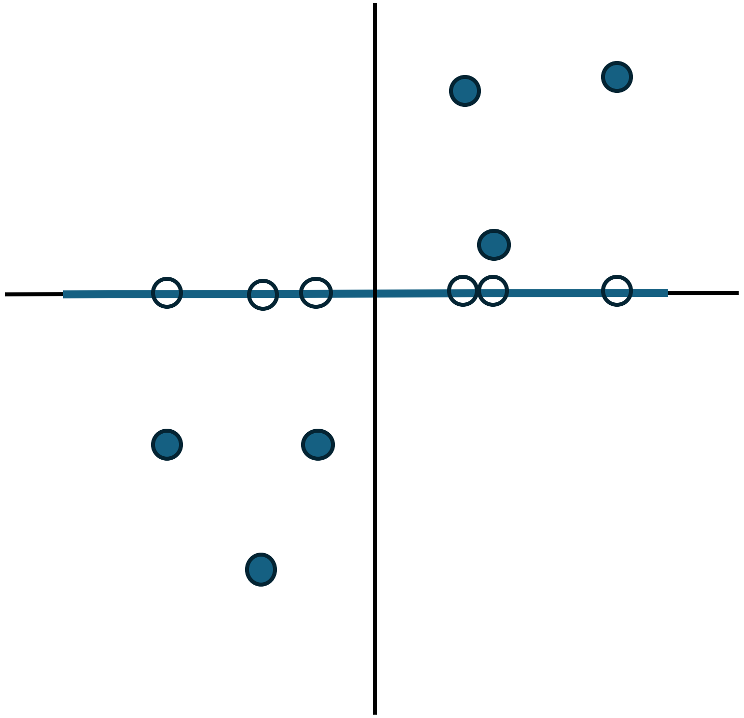
NIPALS explained



X_1	X_2	t
14	12	
15	14	
20	22	
12	11	
22	25	
83	84	

NIPALS explained

P0: [1.00, 0.00]



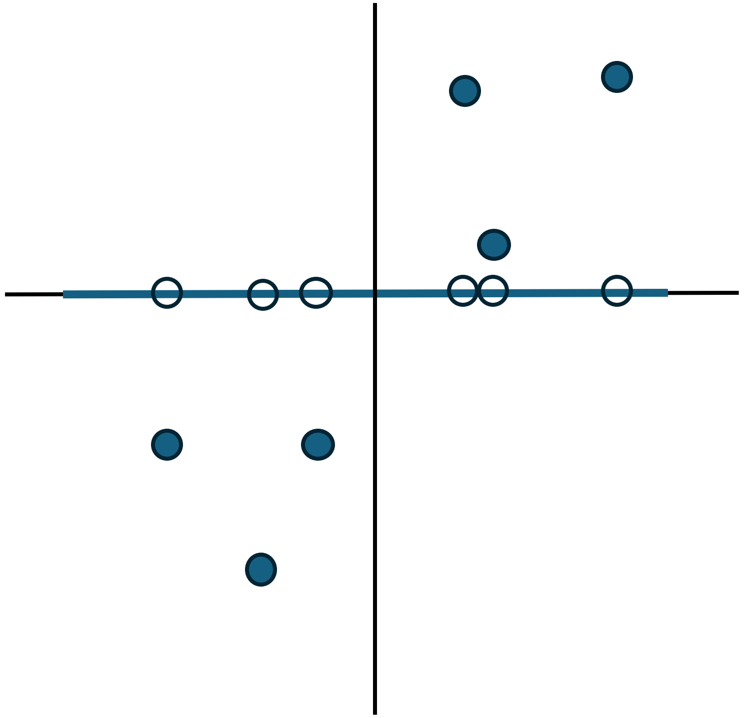
X_1	X_2	t
-2.6	-4.8	-2.6
-1.6	-2.8	-1.6
3.4	5.2	3.4
-4.6	-5.8	-4.6
5.4	8.2	5.4
0	0	0

NIPALS explained

P0: [1.00, 0.00]

P1: [49.33, 72.39], |P1| = 87.6
P1: [0.56, 0.82], |P1|= 1

$$Cov(x,y) = \frac{1}{n-1}\sum(x_i - m_x)(y_i - m_y)$$

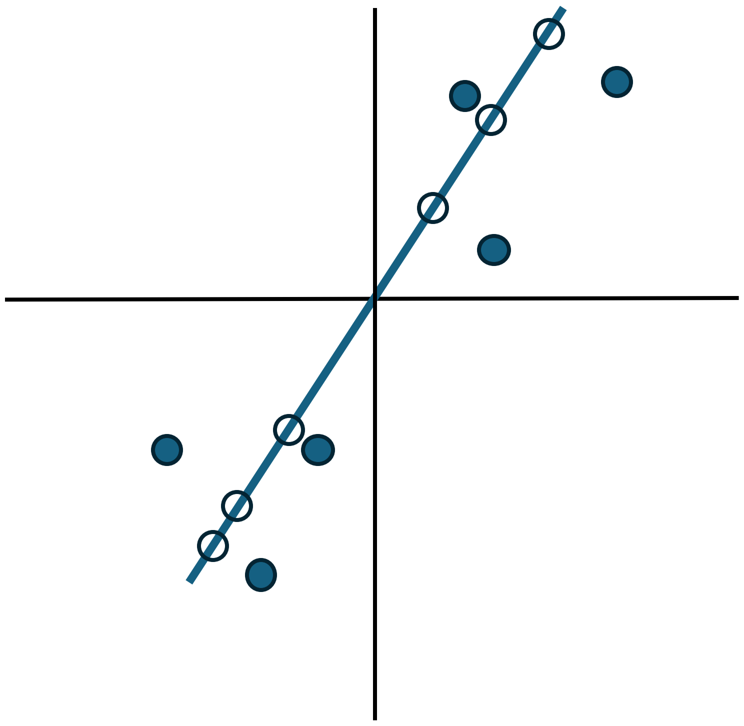


X_1	X_2	t
-2.6	-4.8	-2.6
-1.6	-2.8	-1.6
3.4	5.2	3.4
-4.6	-5.8	-4.6
5.4	8.2	5.4
0	0	0

P1: [0.56, 0.82]

NIPALS explained

$$Cov(x,y) = \frac{1}{n-1}\sum(x_i - m_x)(y_i - m_y)$$



X_1	X_2	t
-2.6	-4.8	
-1.6	-2.8	
3.4	5.2	
-4.6	-5.8	
5.4	8.2	
0	0	

Algorithm

Mean center X, E_0 = X

Set t-vector to column in X (the scores)

p is the loadings

1. Project X on to t to find the loadings (e.g. [1,0] to start)
   - $p = \frac{E_{i-1}^T t}{t^T t}$
2. Normalize loading p to length 1
   - $p = \frac{p}{\sqrt{p^T p}}$
3. Project X on to p to find corresponding score vector (Find new scores with the new p)
   - $t = \frac{E_{i-1} p}{p^T p}$
4. Check for convergence (stopping criterion), if not go to (1)
5. Remove estimated PC from E_(i-1), then run all again for next PC
   - $E_i = E_i - tp^T$

NIPALS vs SVD

- Singular Value Decomposition (SVD)
  - A factorization of a real or complex matrix into a rotation, followed by a rescaling followed by another rotation*

$$M = U \quad \Sigma \quad V^T = \begin{bmatrix} u_1, & u_2, & \cdots, u_r \end{bmatrix} \begin{bmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & & \\ & & \ddots & \\ 0 & & & \sigma_r \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_r^T \end{bmatrix}$$

*https://en.wikipedia.org/wiki/Singular_value_decomposition

NIPALS vs. SVD

- NIPALS is a variation of SVD that allows **missing values** in data
  - NIPALS updates one variable at a time in an **iterative** way, which allows it to bypass missing values. However, SVD requires a complete matrix.
- NIPALS is numerically more accurate but slower than SVD
  - SVD performs **full eigen-decomposition**, which makes it computationally inefficient
- SVD requires that number of columns must be **larger** than number of rows, however NIPALS works with dataset of any size.

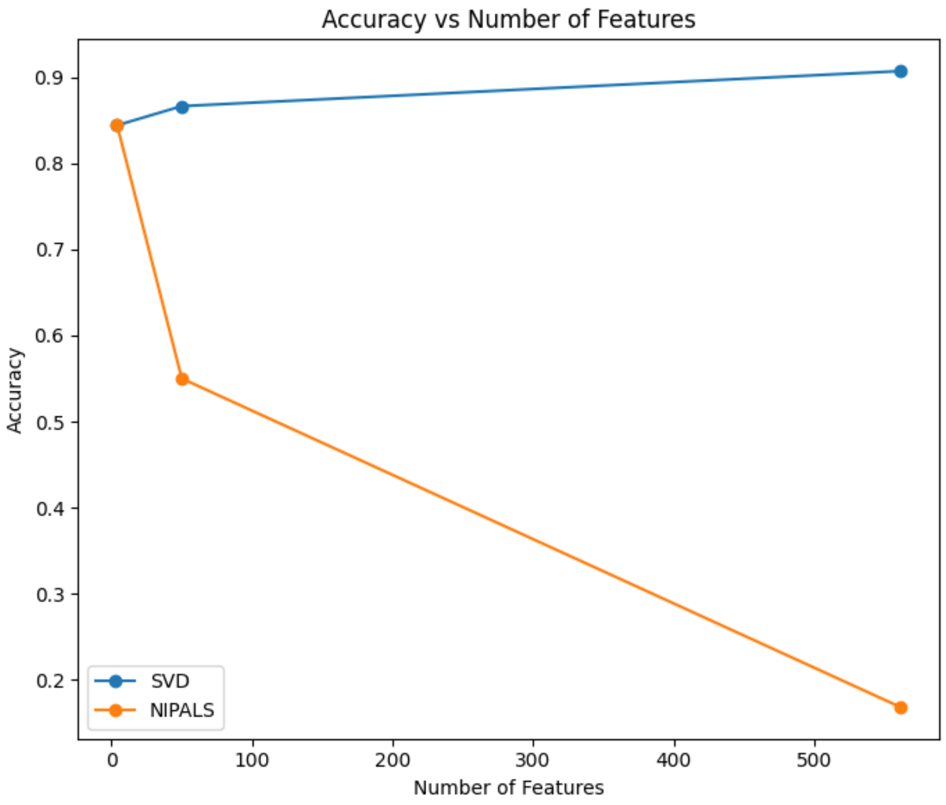
Code for illustration – NIPALS vs SVD

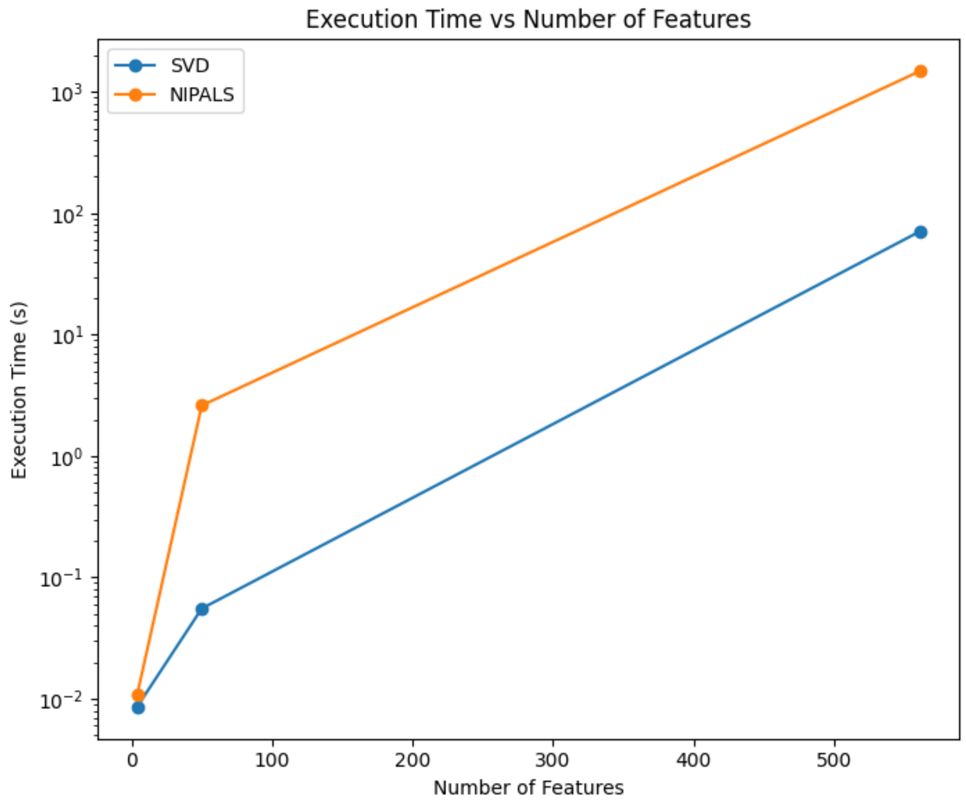
- [Google collab Notebook]

3 Datasets:

1. **Big dataset:** Human Activity Recognition Using Smartphones : 561 feature
2. **Medium dataset**: Synthetic: 50 features
3. **Iris dataset:** 4 features

At first glance NIPALS Looks quite poor (and it is often mentioned that SVD is commonly used for PCA), Here execution time and accuracy of classification better for SVD

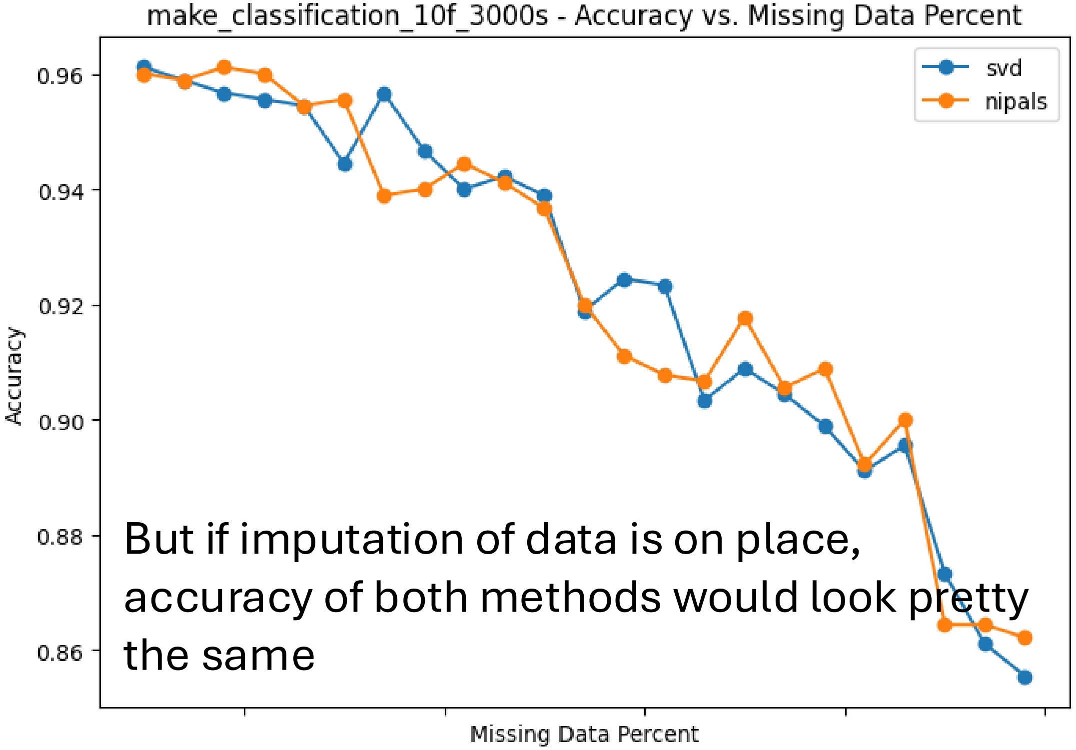




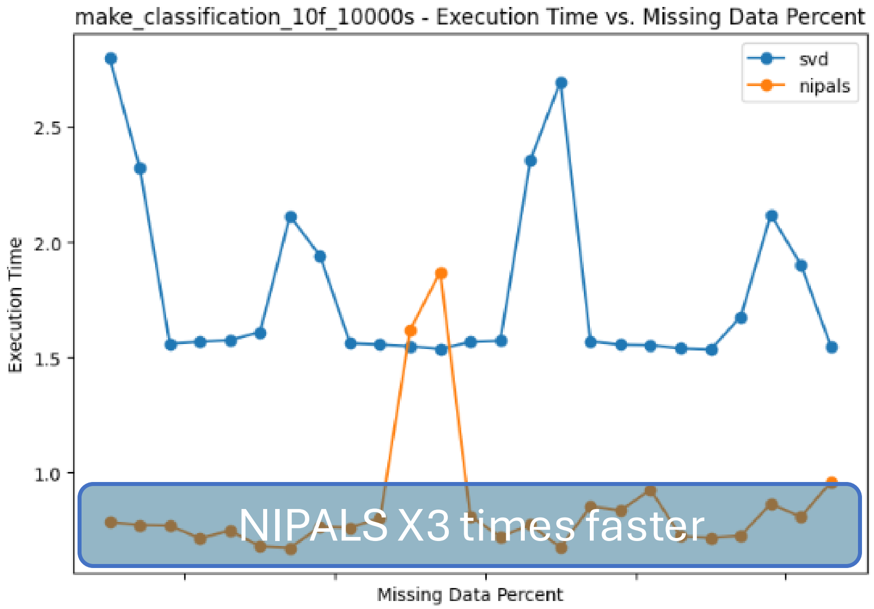
NIPALS for missing data / NIPALS speed

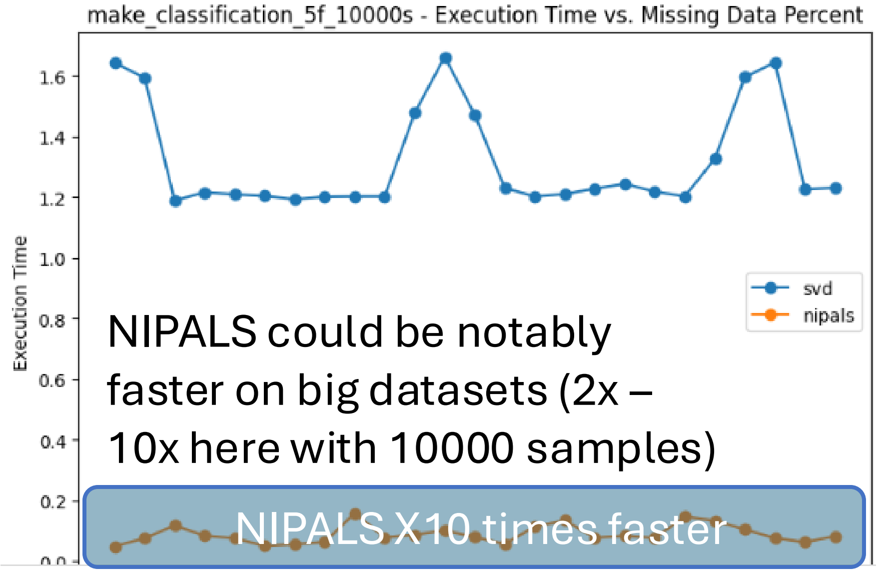
- [Google collab Notebook]

- **SVD** **will not work** with missing data (imputation strategies are needed
- **NIPALS** ***would work*** with missing data **without** imputation strategy being on place



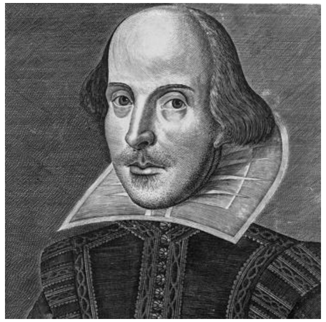
However, imputation could distort and bias data





NIPALS: *“To Use Or Not To Use?*

That is the question.”



Use NIPALS	Use SVD
Data matrix with larger number of variables to sample ratio	Smaller dataset
Data contains missing values	Data matrix more balanced (in number of observations and variables)
Large dataset – iterative methodology of NIPALS more efficient	Greater accuracy