# Principal component analysis as a game with Nash Equilibrium


**Team KENA:**
Nina Konovalova
Albina Klepach
Elvira Plumite
Ekaterina Radionova

# PLAN

1. Problem Statement
2. Recent methods
3. PCA as finding the Nash equilibrium
4. Algorithm
5. Our hypothesis
6. Setting up an experiment
7. Experiments with different data (synthetic data, MNIST, IRIS, digits)
8. Future work
9. References

# PROBLEM STATEMENT

The **principle components (PC)** of data are the vectors that align with the directions of maximum variance.

Principal components analysis (PCA) has **two main purposes**:

- Interpretable features
- Data compression

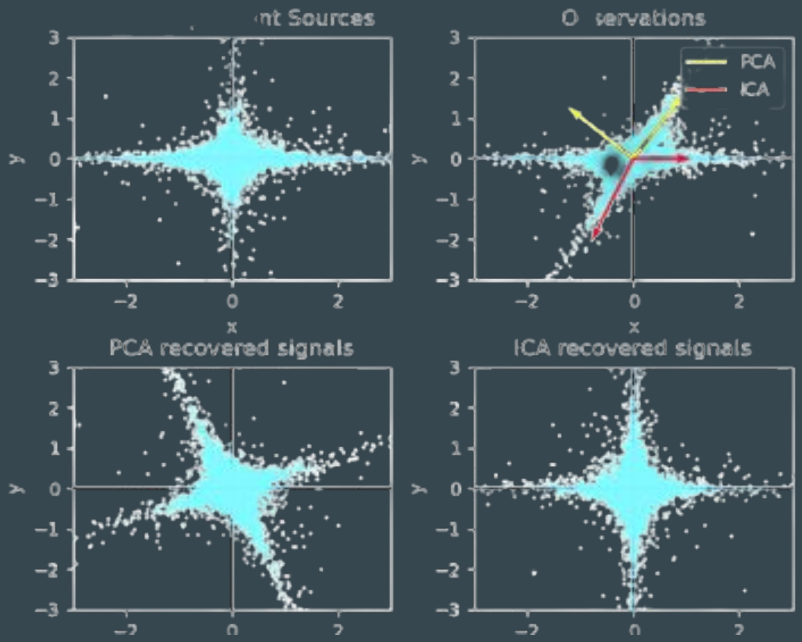
**PCA solution** of dataset X:

- Eigenvectors of $X^TX$ or right singular vectors of X.

# RECENT METHODS

$X - n\, x\, d$

- Full SVD
- Power method
- Oja’s algorithm



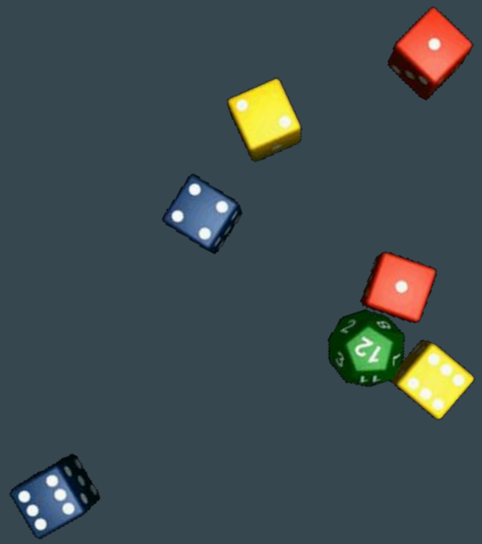
# PCA AS EIGENGAME

Considering **PCA as a game**, where

- **Player** is an eigenvector

- **Player's objective** is to maximize their own local utility function

- **Our purpose** is to reach Nash equilibrium



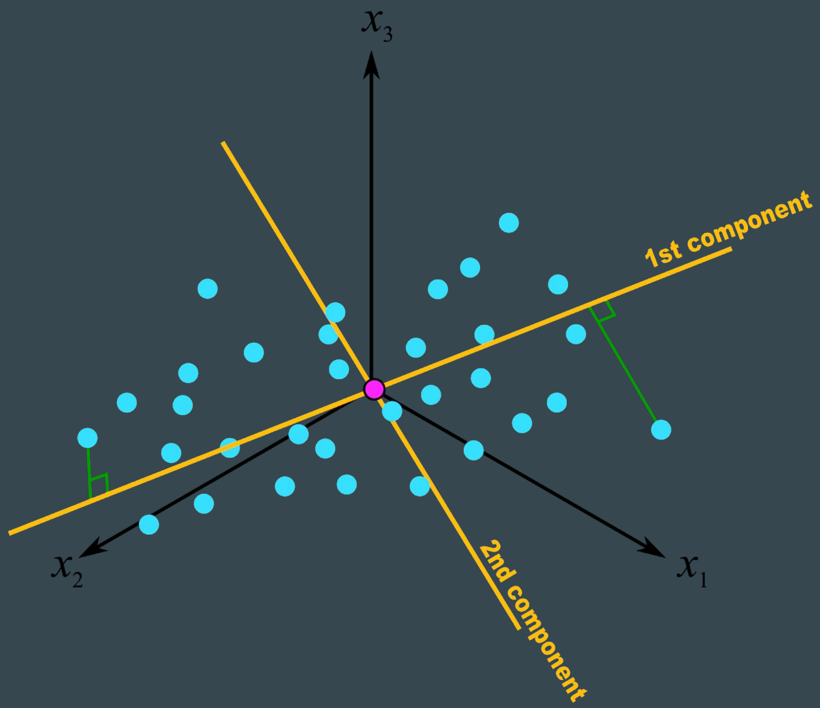
# PCA AS EIGENGAME

$$X^T X = M$$

$$V^T M V = V^T V \Lambda = \Lambda$$

$$R(\hat{V}) = \hat{V}^T M \hat{V}$$

$$\max_{\hat{V}^T \hat{V} = I} \sum_i R_{ii} - \sum_{i \neq j} R_{ij}^2$$



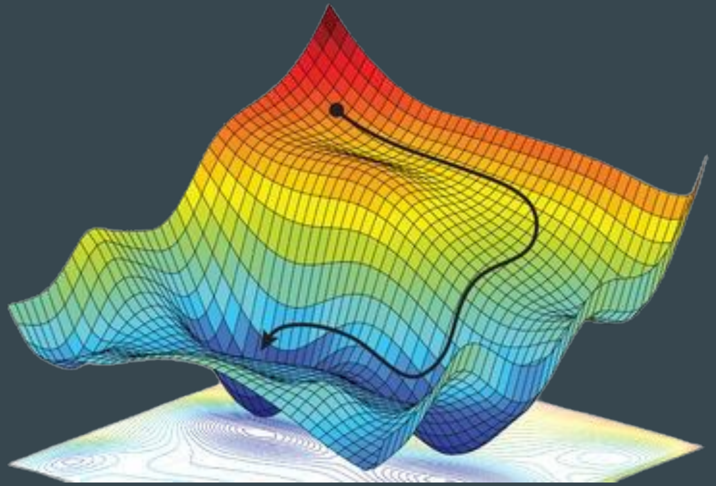
# PCA AS EIGENGAME

**Utility functions:**

$$\max_{\hat{v}_1^T \hat{v}_1 = 1} \langle \hat{v}_1, M \hat{v}_1 \rangle$$

$$\max_{\hat{v}_2^T \hat{v}_2 = 1, \hat{v}_1^T \hat{v}_2 = 0} \langle \hat{v}_2, M \hat{v}_2 \rangle - \frac{\langle \hat{v}_2, M \hat{v}_1 \rangle^2}{\langle \hat{v}_2, M \hat{v}_2 \rangle}$$

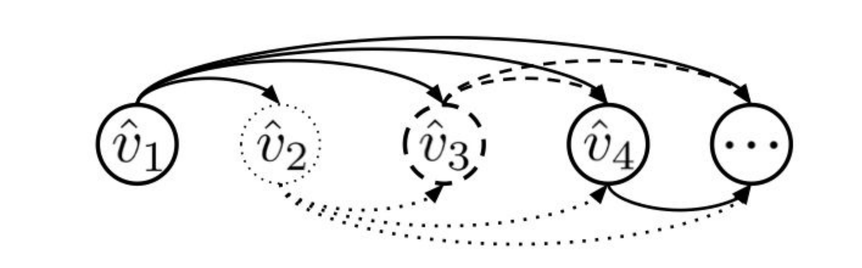
$$u_i(\hat{v}_i | \hat{v}_{j<i}) = \hat{v}_i^T M \hat{v}_i - \sum_{j<i} \frac{\left(\hat{v}_i{}^T M \hat{v}_j\right)^2}{\hat{v}_j{}^T M \hat{v}_j} = ||X \hat{v}_i||^2 - \sum_{j<i} \frac{\langle X \hat{v}_i, X \hat{v}_j \rangle^2}{\langle X \hat{v}_j, X \hat{v}_j \rangle}$$



# PCA AS EIGENGAME

**<u>Nash equilibrium</u>** - strategy profile when no player can do better by unilaterally changing their strategy.

**<u>Theorem:</u>** Assume that the *top-k* eigenvalues of $X^T X$ are distinct. Then the *top-k* eigenvectors form the unique strict-Nash equilibrium of the proposed game.



# ALGORITHM EIGENGAME

**Utility gradient**

$$\nabla_{\hat{v}_i} u_i(\hat{v}_i | \hat{v}_{j<i}) = 2M \left[ \hat{v}_i - \sum_{j<i} \frac{\hat{v}_i^T M \hat{v}_j}{\hat{v}_j^T M \hat{v}_j} \hat{v}_j \right] =$$

$$2X^T \left[ X\hat{v}_i - \sum_{j<i} \frac{\langle X\hat{v}_i, X\hat{v}_j \rangle}{\langle X\hat{v}_j, X\hat{v}_j \rangle} X\hat{v}_j \right]$$

**Algorithm 1** EigenGame$^R$-Sequential

Given: matrix $X \in \mathbb{R}^{n \times d}$, maximum error tolerance $\rho_i$, initial vector $\hat{v}_i^0 \in \mathcal{S}^{d-1}$, learned approximate parents $\hat{v}_{j<i}$, and step size $\alpha$.

$\hat{v}_i \leftarrow \hat{v}_i^0$

$t_i = \lceil \frac{5}{4} \min(||\nabla_{\hat{v}_i^0} u_i||/2, \rho_i)^{-2} \rceil$

**for** $t = 1 : t_i$ **do**

  `rewards` $\leftarrow X\hat{v}_i$

  `penalties` $\leftarrow \sum_{j<i} \frac{\langle X\hat{v}_i, X\hat{v}_j \rangle}{\langle X\hat{v}_j, X\hat{v}_j \rangle} X\hat{v}_j$

  $\nabla_{\hat{v}_i} \leftarrow 2X^\top \left[ \texttt{rewards} - \texttt{penalties} \right]$

  $\nabla^R_{\hat{v}_i} \leftarrow \nabla_{\hat{v}_i} - \langle \nabla_{\hat{v}_i}, \hat{v}_i \rangle \hat{v}_i$

  $\hat{v}_i' \leftarrow \hat{v}_i + \alpha \nabla^R_{\hat{v}_i}$

  $\hat{v}_i \leftarrow \frac{\hat{v}_i'}{||\hat{v}_i'||}$

**end for**

return $\hat{v}_i$

# DATA

**Synthetic data:**

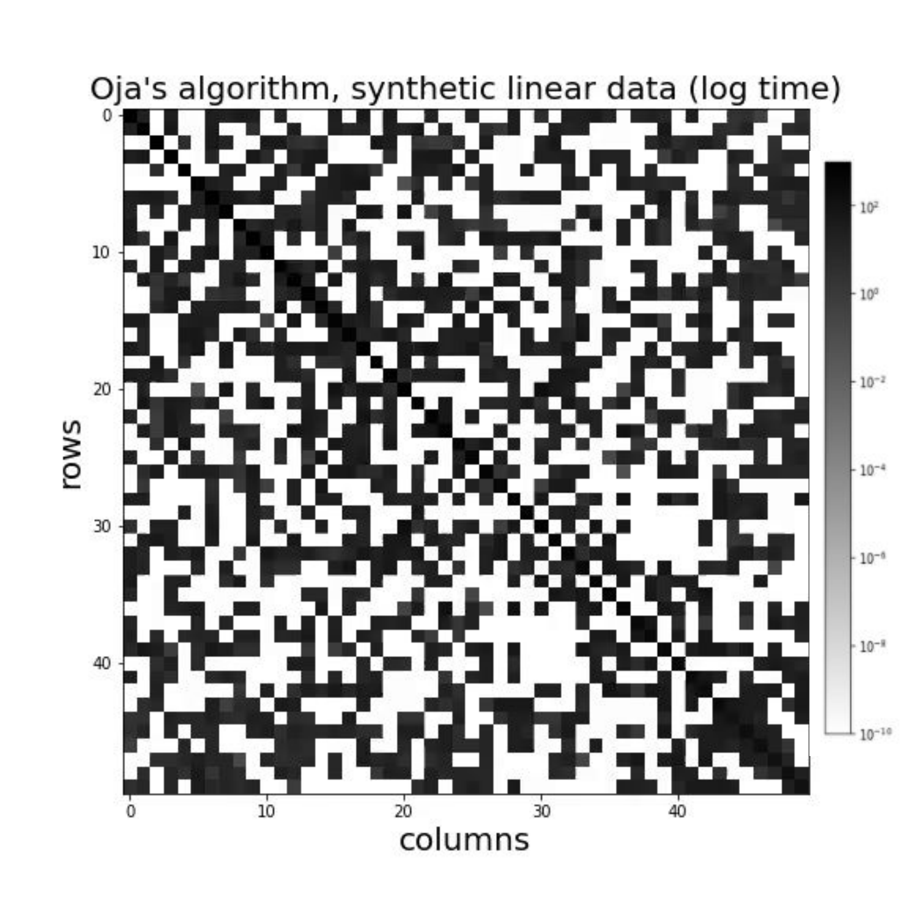
- *V* is initialized randomly so $M \in \mathbb{R}^{50\times 50}$ is constructed as a diagonal matrix without loss of generality
- Linear spectrum ranges from 1 to 1000 with equal spacing
- Exponential spectrum ranges from $10^3$ to $10^0$ with equal spacing on the exponents

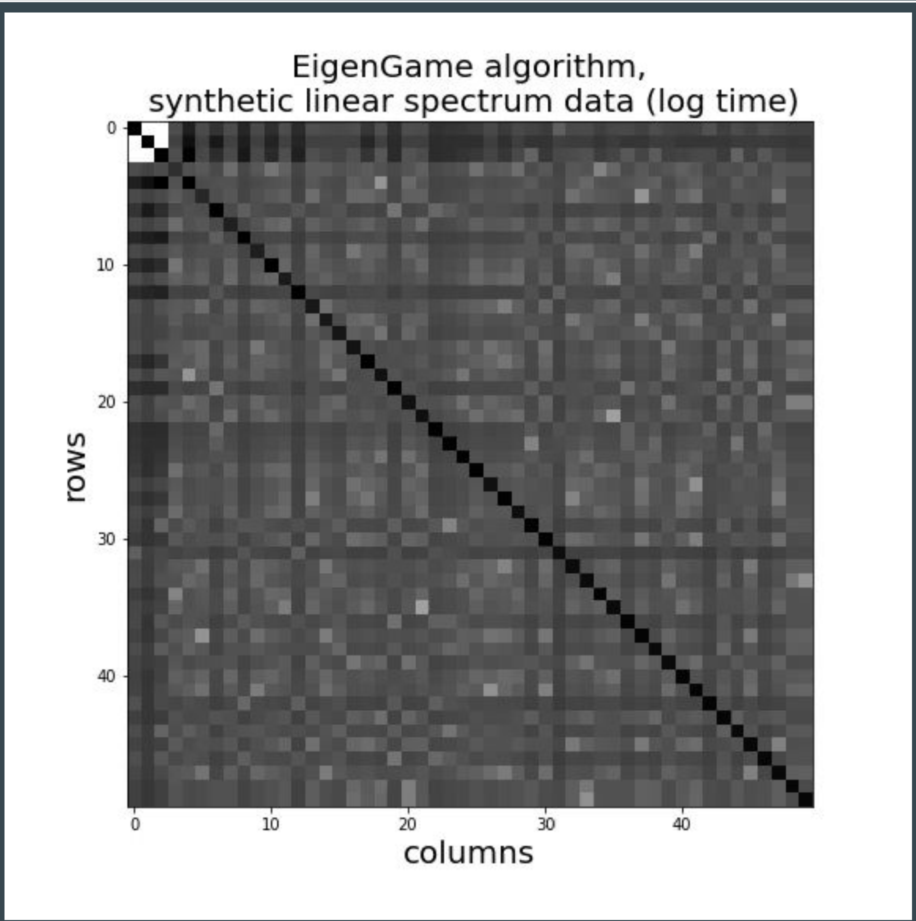
**MNIST dataset**

**IRIS dataset**

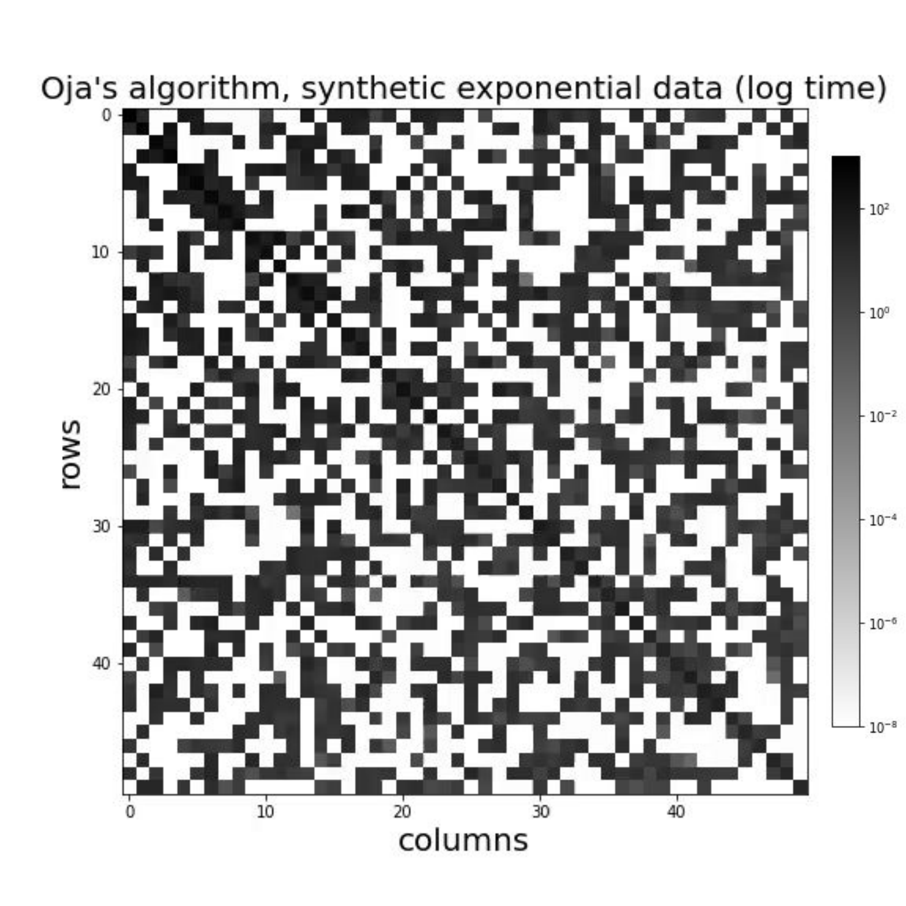
**Digits**

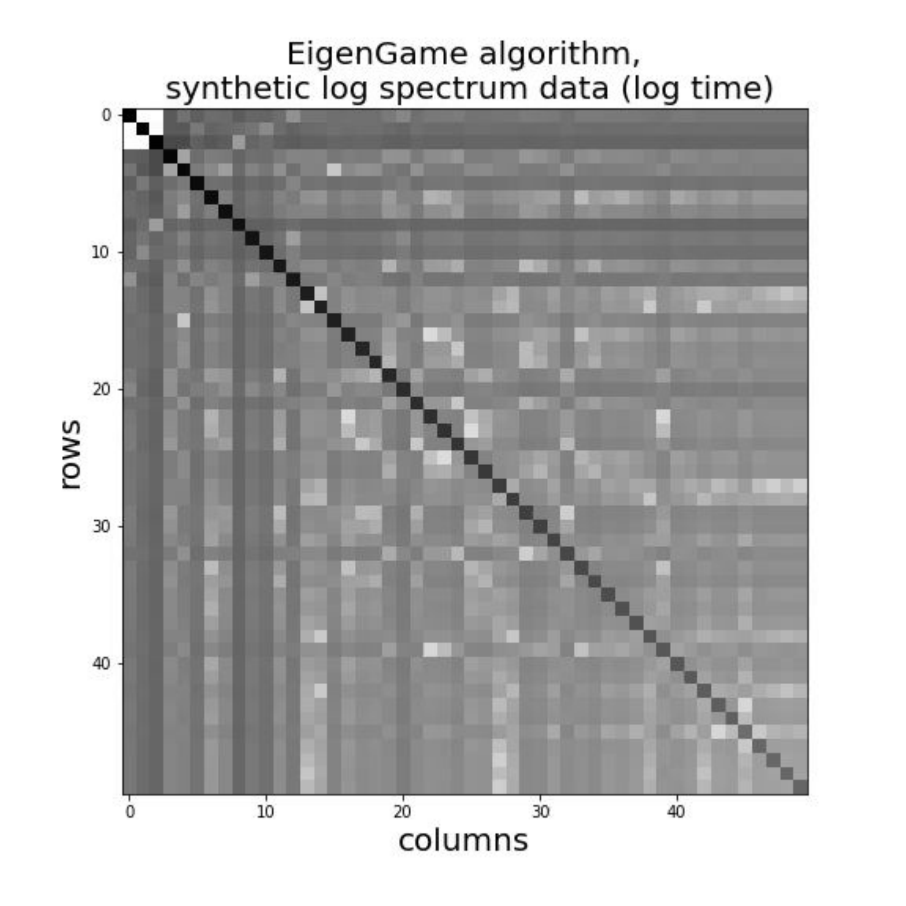
# EXPERIMENTS



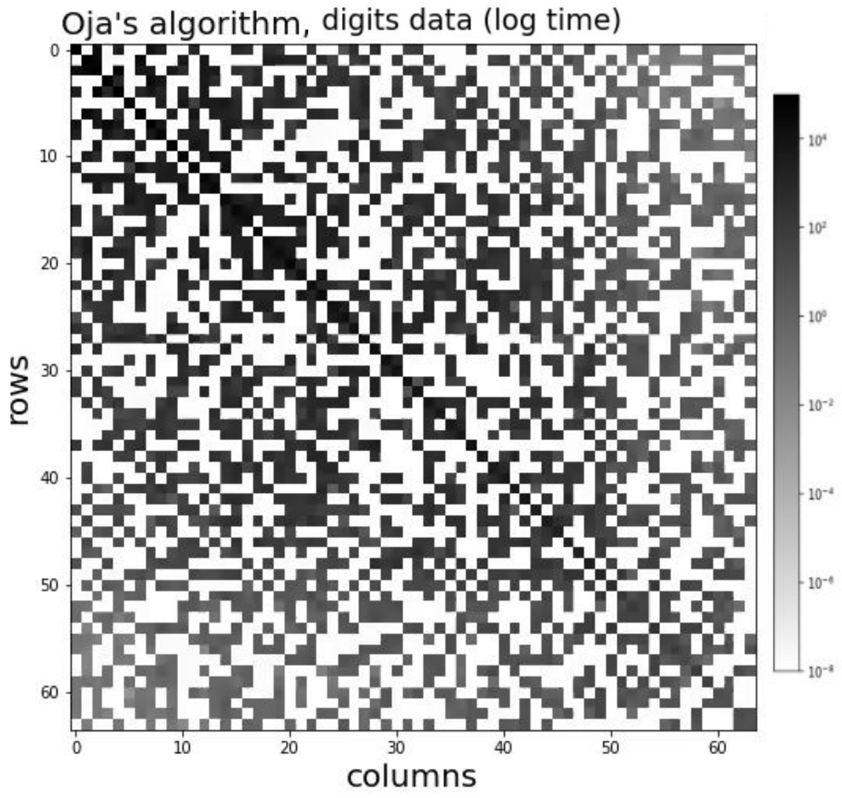



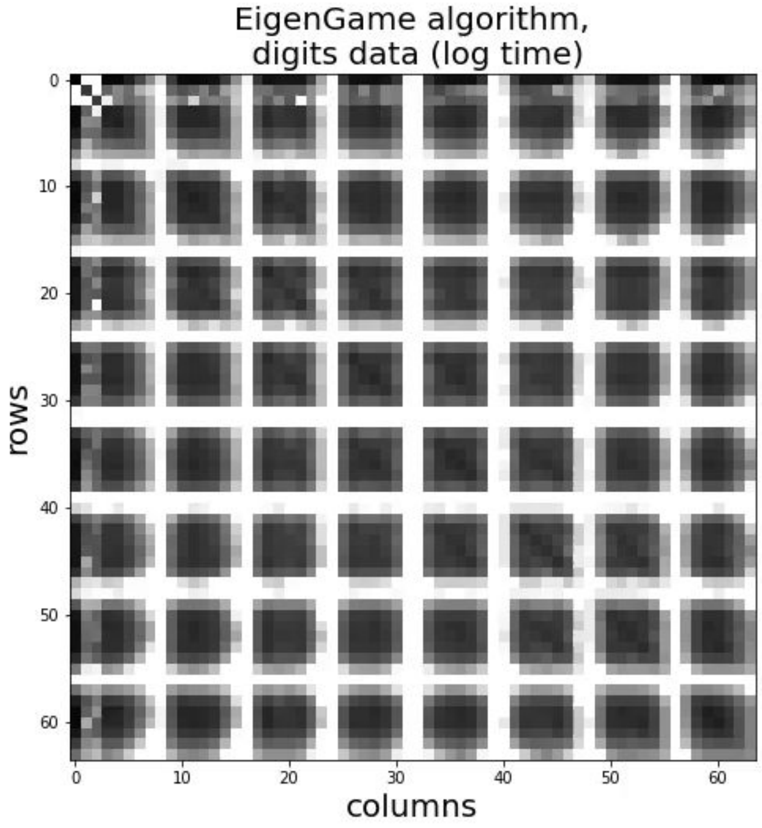
# EXPERIMENTS



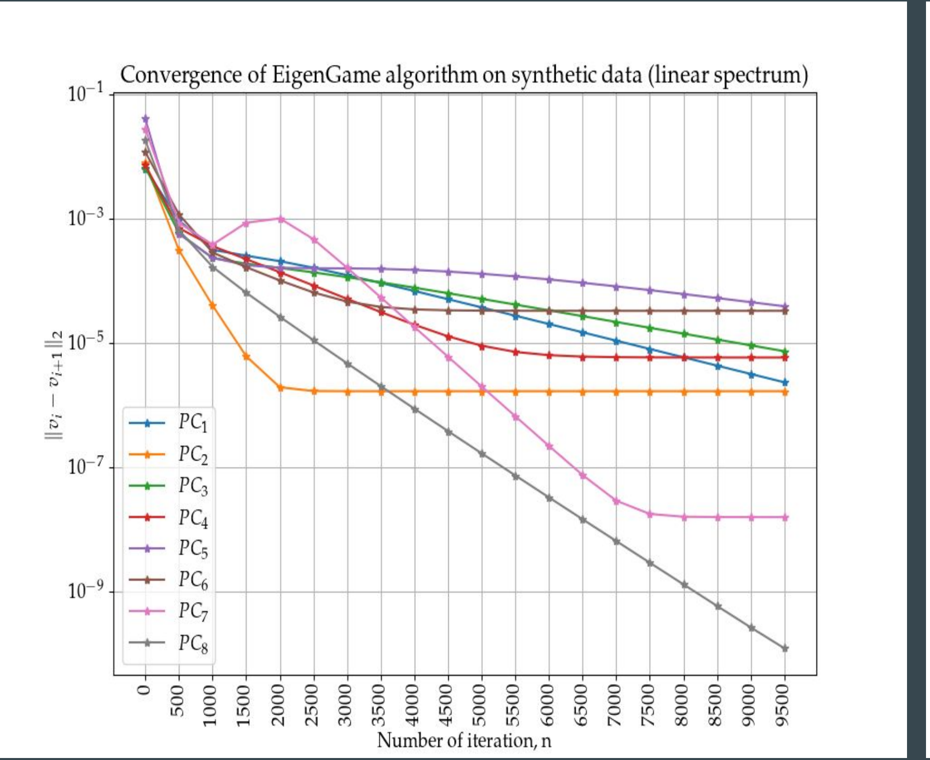



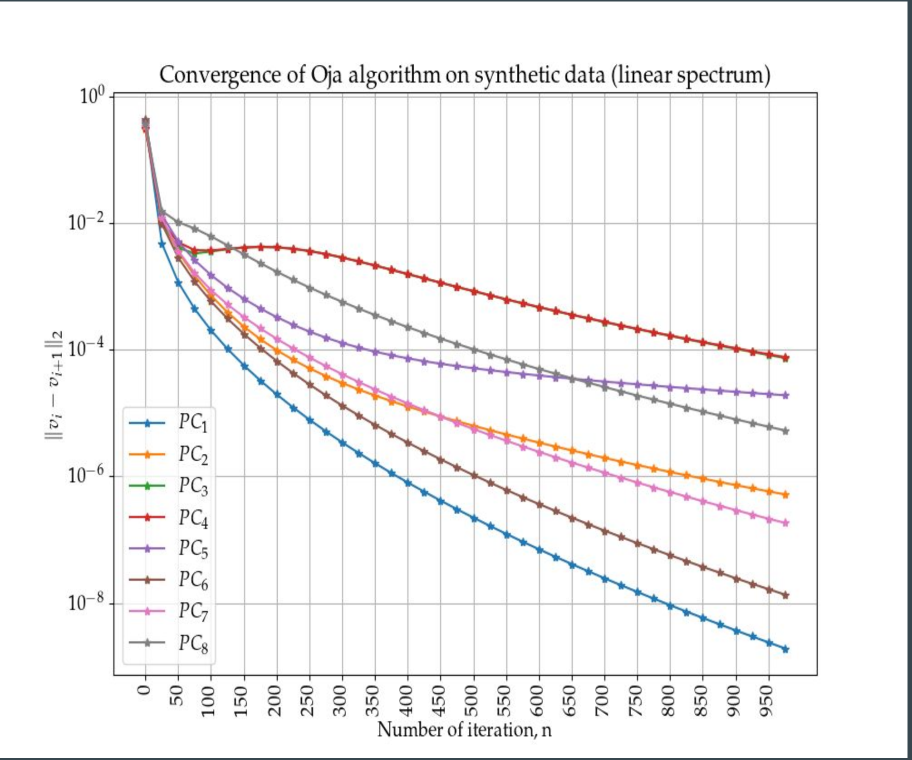
# EXPERIMENTS



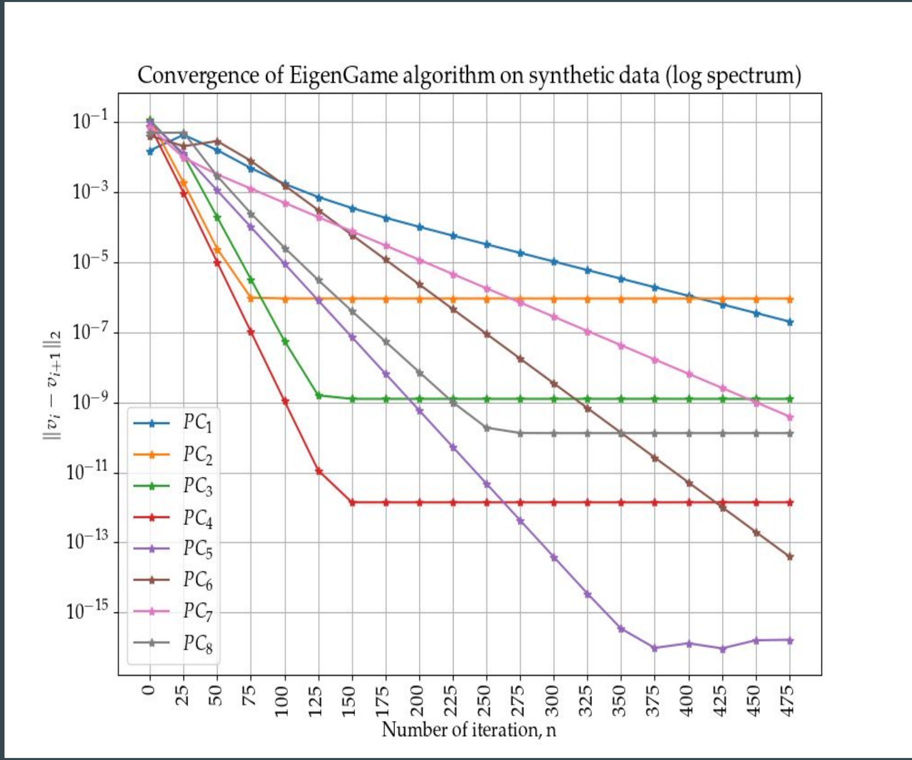



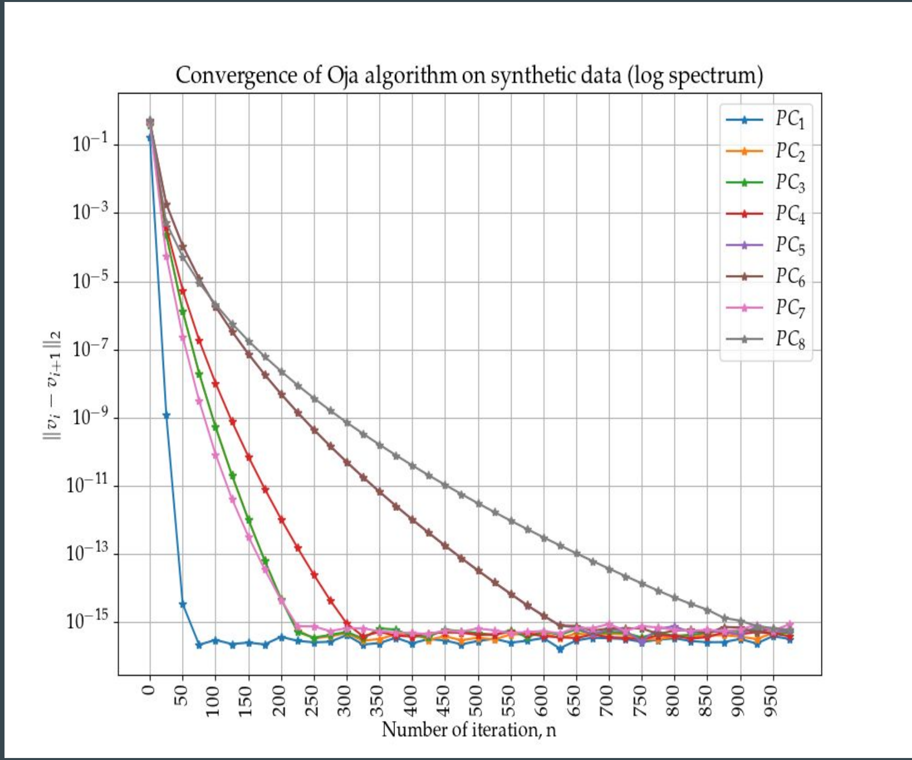
# EXPERIMENTS



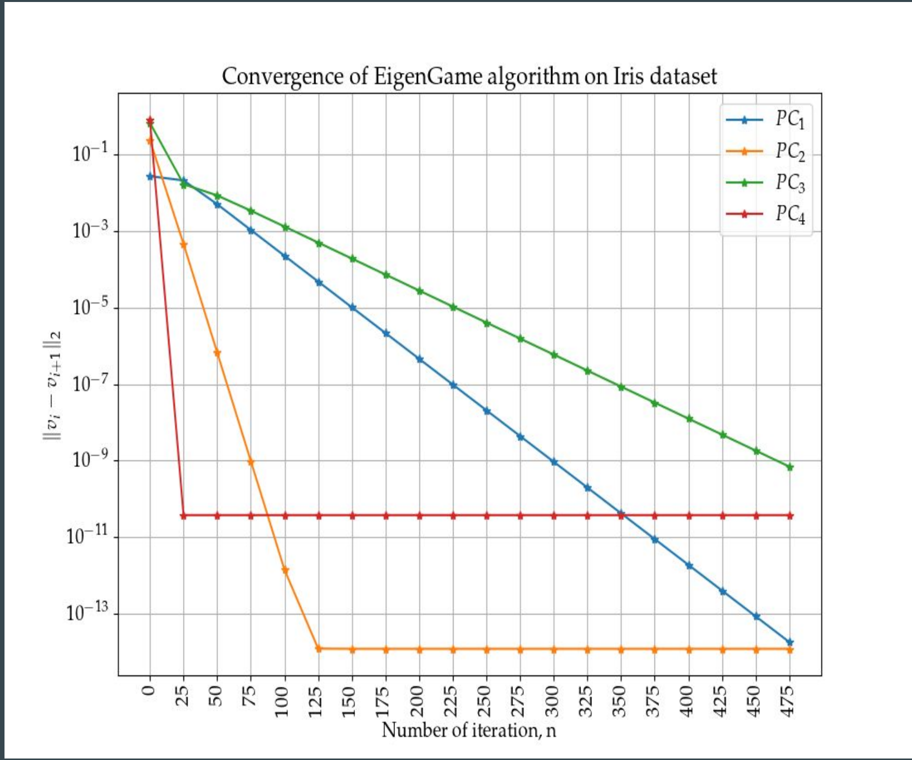



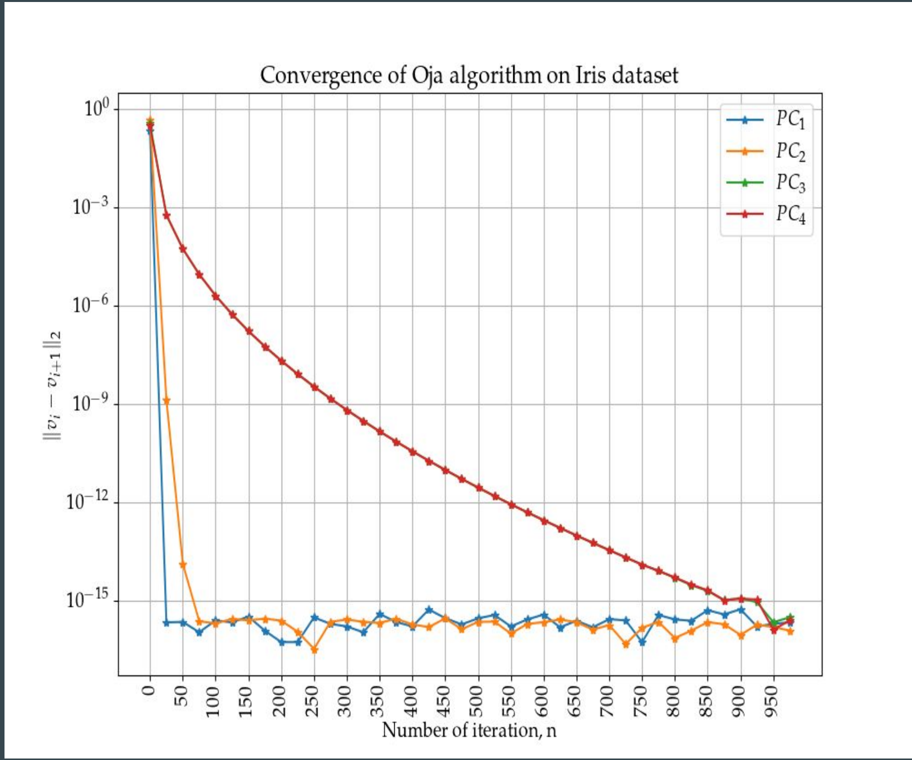
# EXPERIMENTS



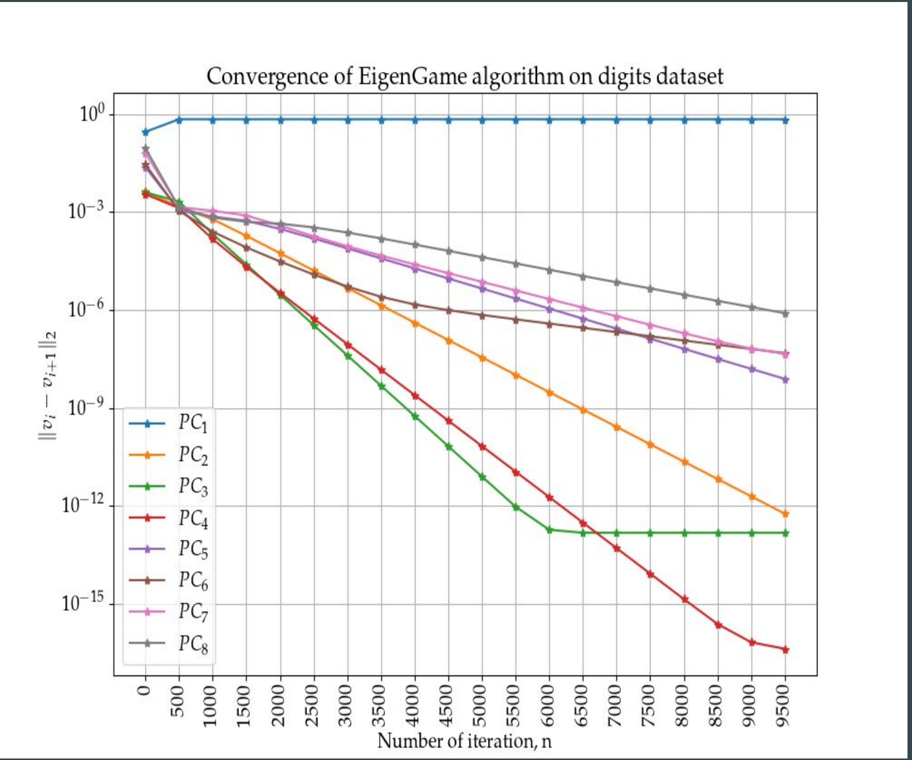



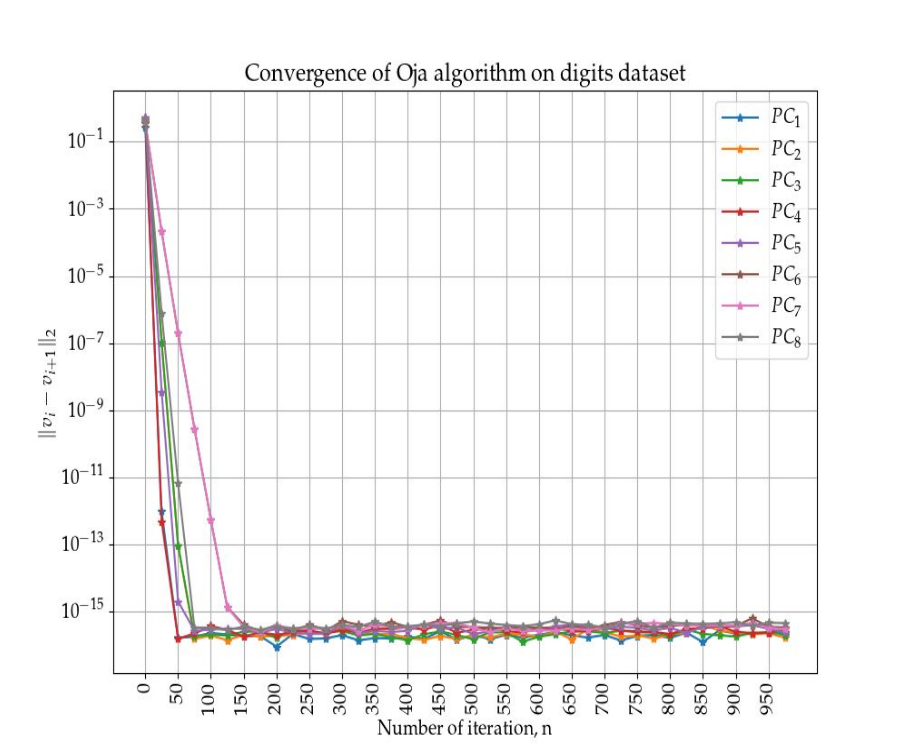
# EXPERIMENTS



# EXPERIMENTS



# RESULTS

- Realization of EigenGame algorithm
- Realization of Oja’s algorithm
- Diagonal convergence
- Checking algorithm on different datasets (synthetic data, MNIST, IRIS, digits)
- Visualizations of algorithms

# FURTHER DEVELOPMENT

1. **Parallelization:** faster computation

2. **Deep Variants:** $X\hat{v}_i \longrightarrow f_i(X|weights)$

3. **Scale:** improve efficiency

4. **Core ML:** accelerating training

# Thank you for your attention!

# REFERENCES


1. EigenGame: PCA as a Nash Equilibrium
2. A Stochastic PCA and SVD Algorithm with an Exponential Convergence Rat
3. First Efficient Convergence for Streaming k-PCA: a Global, Gap-Free, and Near-Optimal Rate
4. Oja's rule: Derivation, Properties
5. Nash Equilibrium
6. Principal component analysis
7. AdaOja: Adaptive Learning Rates for Streaming PCA