High-Performance Dimensionality Reduction in Large Language Models

Nahid.Emad@uvsq.fr

Maison de la Simulation, Li-Parad

University of Paris Saclay, France

First workshop on

Distributed and Parallel Programming for Extreme-scale AI

June 16-17th, 2025 | Mines-Paris, PSL University, France

Dimensionality Reduction in AI

Reducing the cost of LLMs through dimensionality reduction techniques that simplify data representation while retaining relevant information

Distilling big/complex/raw data to produce unified/focused/usable ones

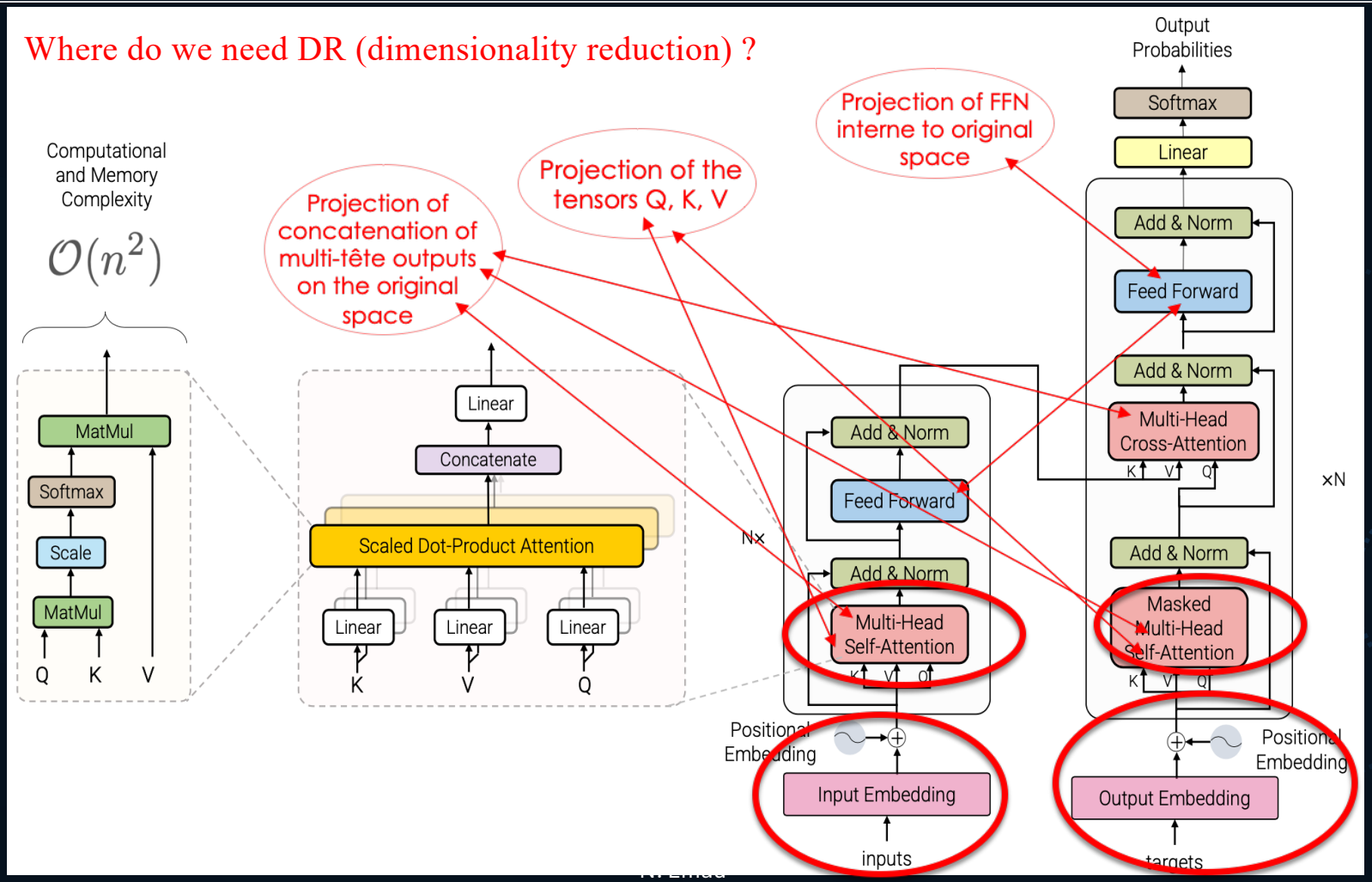
- Direct impact on applications as clustering, data visualization, anomaly detection, data compression, data preprocessing, …
- Optimization of LLMs like Transformer by applying them to almost every layer

Processing reduced data provides accelerated & efficient LLMs

Challenges : Robust distributed dimensionality reduction

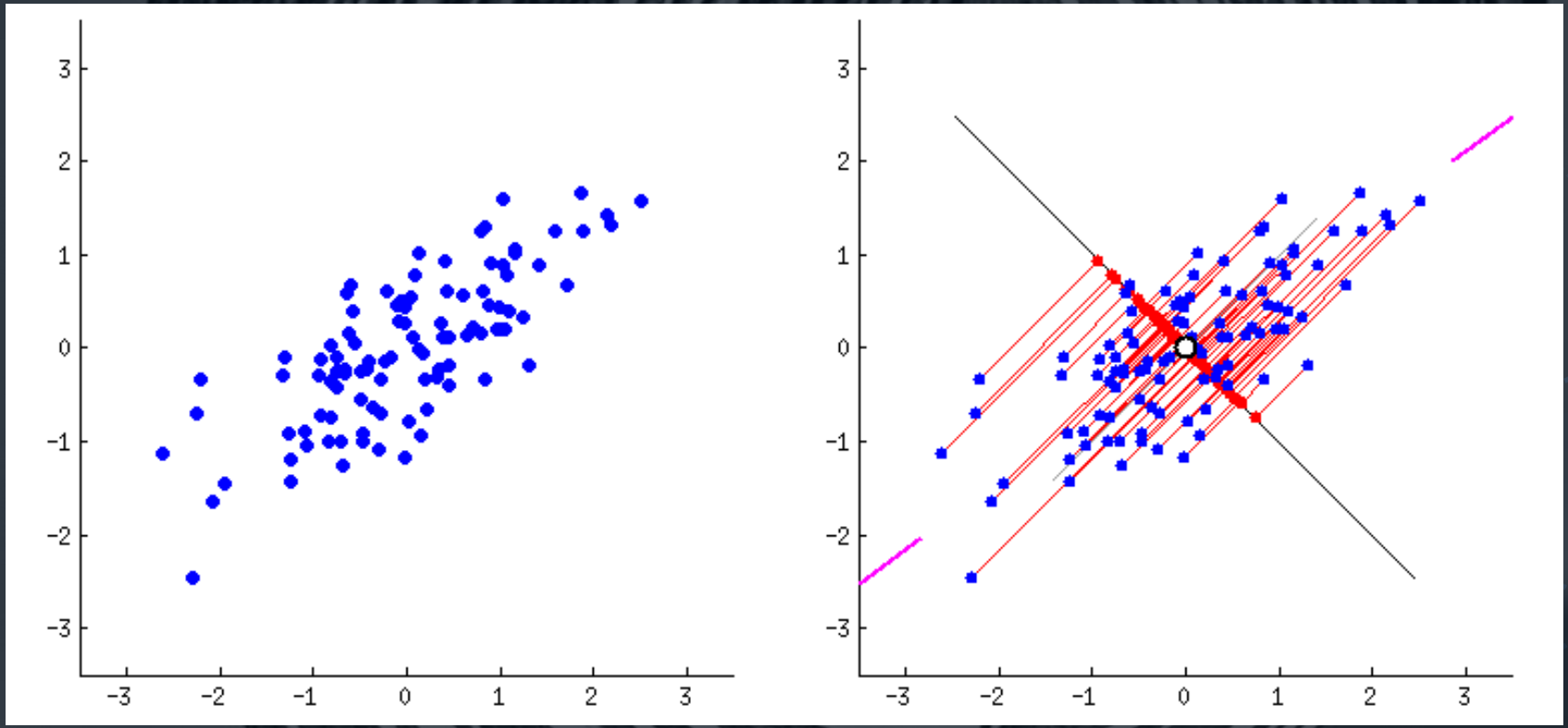
DR FOR TRANSFORMER-BASED LLMS

Efficient Transformers: A Survey. Yi Tay and al., ACM Computing Surveys, 2022,vol. 55.



PROJECTION ON SMALLER OR LARGER SPACES

1. Projection on smaller subspace



An Example application of PCA to a financial data set in **WEKA**, a Java data mining software

https://youtu.be/OdlNM96sHio?si=9I3OO4dZgRE8pMxJ

2. Projection on larger space

OUTLINE

- A brief overview of DR techniques
- Unite and Conquer methods in high-performance linear algebra
- Multiple implicit restarted Arnoldi method (MIRAMns)
- Concluding remarks

A BRIEF OVERVIEW OF DR TECHNIQUES

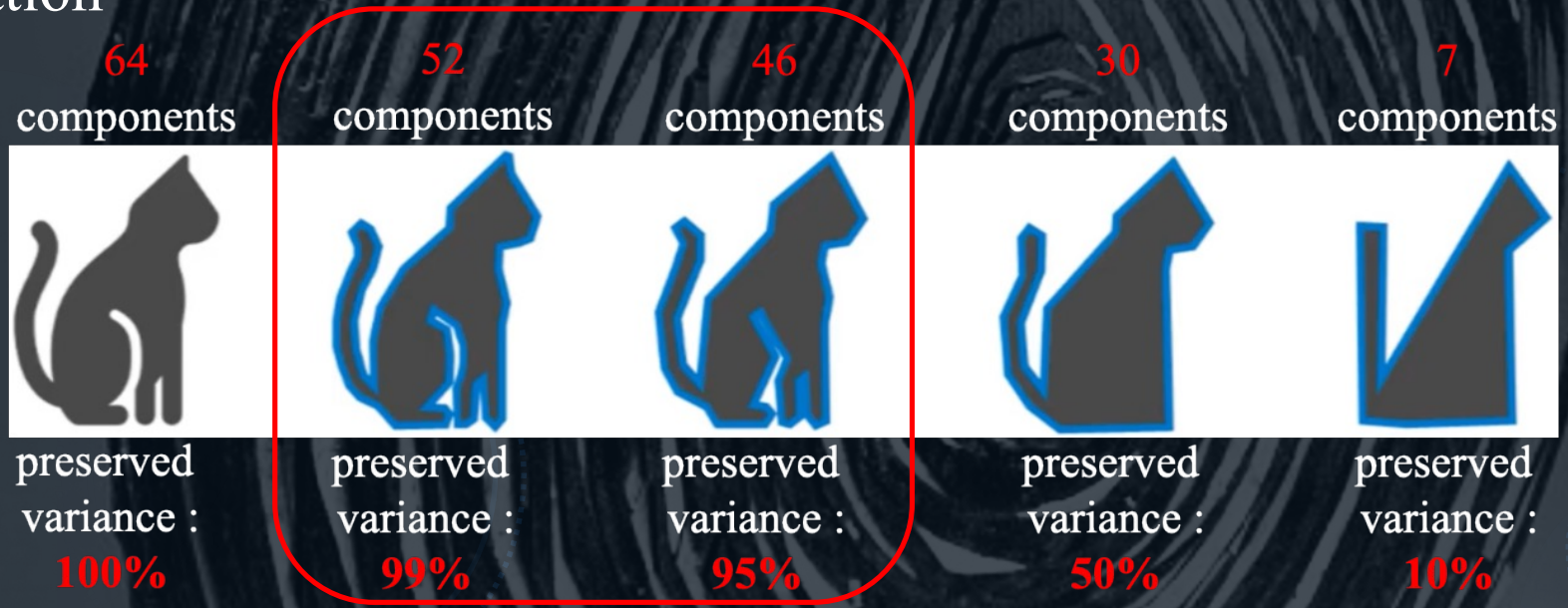
Curse of dimensionality : Problems like increase of data size and parameters, points too far apart making it difficult to estimate data distributions, overfitting risk of the model, loss of the meaning of proximity notions, ...

Major challenge : Learn from a simplified data representation of the original dataset to gain more insights from the original dataset.

Methodology : Apply DR techniques to map the raw information into a new feature space where data analysis methods can be used.

DIMENSIONALITY REUCTION

Reducing data by simplifying its representation while keeping relevant information

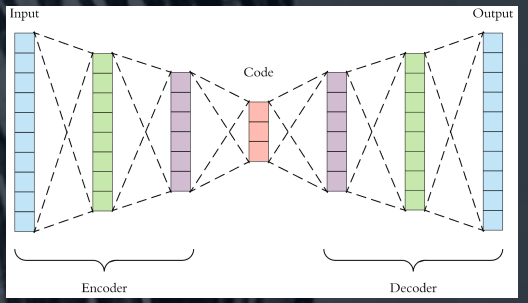


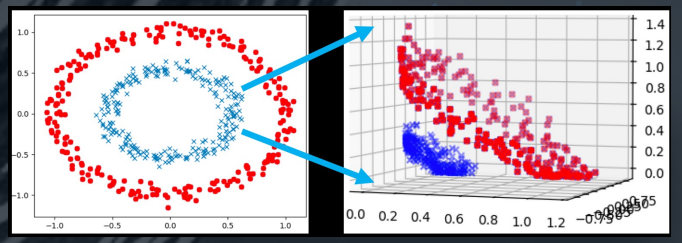
Guillaume Saint-Cirgue : https://www.youtube.com/c/MachineLearnia

DR TECHNIQUES BASED ON **NON-LINEAR** TRANSFORMATION

- ✓ Autoencoder : Encoding (DR) + decoding for reconstructing data. *In case of linear activations or a single hidden layer of sigmoid, the ideal solution for an autoencoder is heavily linked to PCA.*
- ✓ Pruning and quantization to compress models after training a mobile application (integration in on-ship SRAM)
- ✓ Locally Linear Embedding : Build a weight matrix representing the local reconstruction of each data point from its nearest neighbors, then, computes the eigenvectors of the global weight matrix (PCA)
- ✓ t-distributed Stochastic Neighbor Embedding (PCA + t-SNE),
- ✓ Kernel PCA: Transforms the dataset into a higher dimensional feature space + PCA
- ✓ …

Some are **often linked** to spectral calculation





DR TECHNIQUES BASED ON **LINEAR** TRANSFORMATION

- ✓ Principal/ Independent Component Analysis (PCA, ICA),
- ✓ Linear Discriminant Analysis (known #variables - classification),
- ✓ Low-Rank approximations : Singular Value and CUR Decompositions (importance of dimensions)
- ✓ Random Projections (data.Random matrix),
- ✓ Partial Least Squares (PLS),
- ✓ t-distributed Stochastic Neighbor Embedding (PCA + t-SNE),
- ✓ …

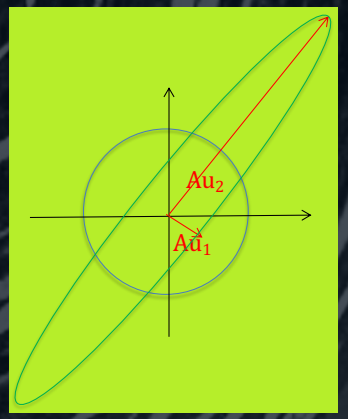
Methods **closely related** to the Spectral Calculation

SPECTRAL COMPUTATION WITH LARGE SPARSE MATRICES

For $A \in \mathbb{C}^{nxn}$, compute a small number of (less)dominant eigenpairs $\Lambda_k = (\lambda_1, \ldots, \lambda_k) \in \mathbb{C}^k$ and $U_k = (u_1, \ldots, u_k)$: $Au_i = \lambda_i u_i$ for $i \in [1, k]$ (P_n)

The vectors $u_1, \ldots, u_k$ are the privileged axes, according to which the matrix application behaves like a dilation, multiplying the vectors by a constant. This dilation ratio is called eigenvalue, the vectors to which it applies are called eigenvectors, united in an eigenspace.

The calculation of the eigenelements of the matrices is commonly called, spectral calculation.



$$A\,[u_1, u_2] = \begin{pmatrix} 0.5 & 0 \\ 0 & 3 \end{pmatrix} [u_1, u_2]$$

ITERATIVE PROJECTION METHODS LARGE SPARSE EIGENPROBLEM

Random choice of an initial subspace $\mathbb{K}_0$,

For $i = 0, 1, \ldots$ until convergence do:

1. Projection of the problem onto $\mathbb{K}_i$
2. Solve the projected problem in the subspace
3. Use 2 to compute approximated solutions in original space
4. If no convergence, with a better subspace go to 1

IRAM
(Implicitly Restarted Arnoldi Method)

Improvement of $\mathbb{K}_m(A, v)$ by that of v

A very commonly used subspace is Krylov subspace :

$$\mathbb{K}_m(A, v) = span\,(v, Av, \ldots, A^{m-1}v) \text{ with } (k \leq m \ll n)$$

UNITE AND CONQUER METHODS IN HP LINEAR ALGEBRA

Suppose we have ℓ iterative methods to solve the same given problem. The unite and conquer approach consists of making collaborate these ℓ methods to accelerate the convergence of the whole system.

Unite and Conquer method	Co-methods components of UC method				
	Bi-Lanczos	+	Bi-Lanczos	+	…
	ERAM	+	ERAM	+	
	ERAM	+	GMRES	+	…
	Projection method	+	Convergence accelerator	+	…
	…	+	…	+	…

Before with hybrid methods :

Y. Saad, *Chebyshev acceleration techniques for solving nonsymmetric eigenvalue; 1984*, C. Brezinski Hybrid procedures for solving linear systems; 1994), Code coupling (in simulation), ...

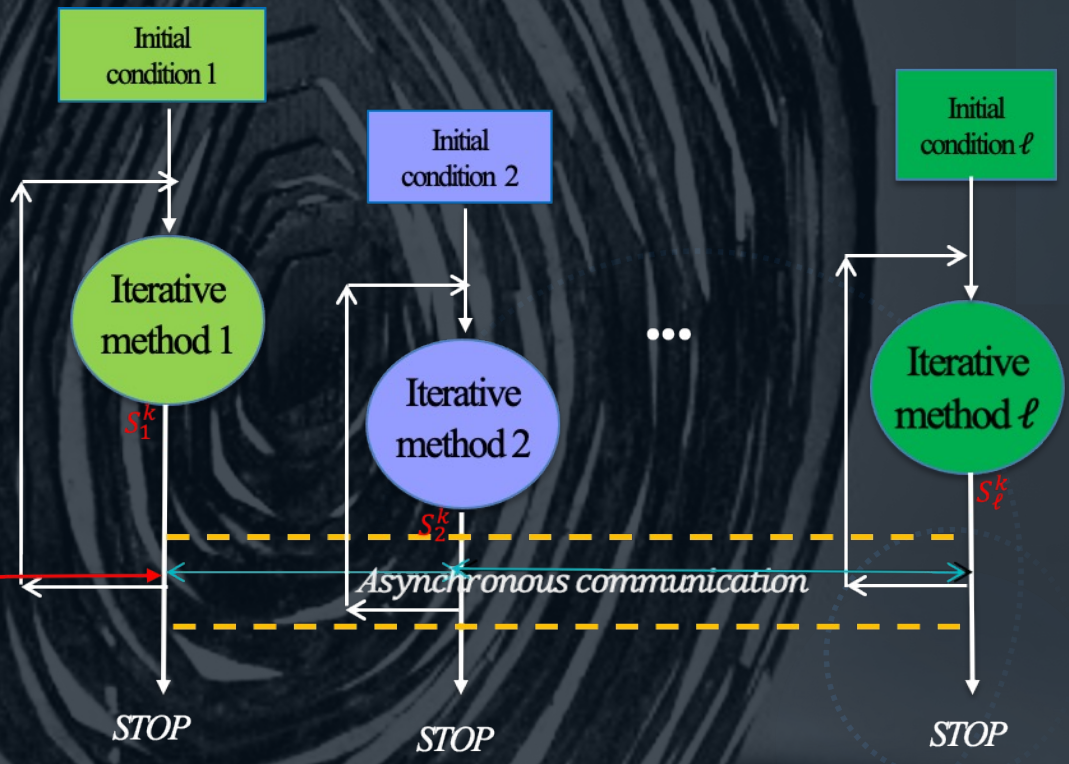
UNITE AND CONQUER METHODS IN HP LINEAR ALGEBRA

Characteristics of UC methods

- Multi level parallelism (coarse grain and fine grain)
- Overlapping of comp/comm (asyn comm)
- Fault tolerance
- Great potential to dynamic load balancing
- Many parameters, many reuse software components
- Need well suited «standard» programming tools

The key part of the method: restarting strategy

$f(S_1^k, \ldots, S_\ell^k)$

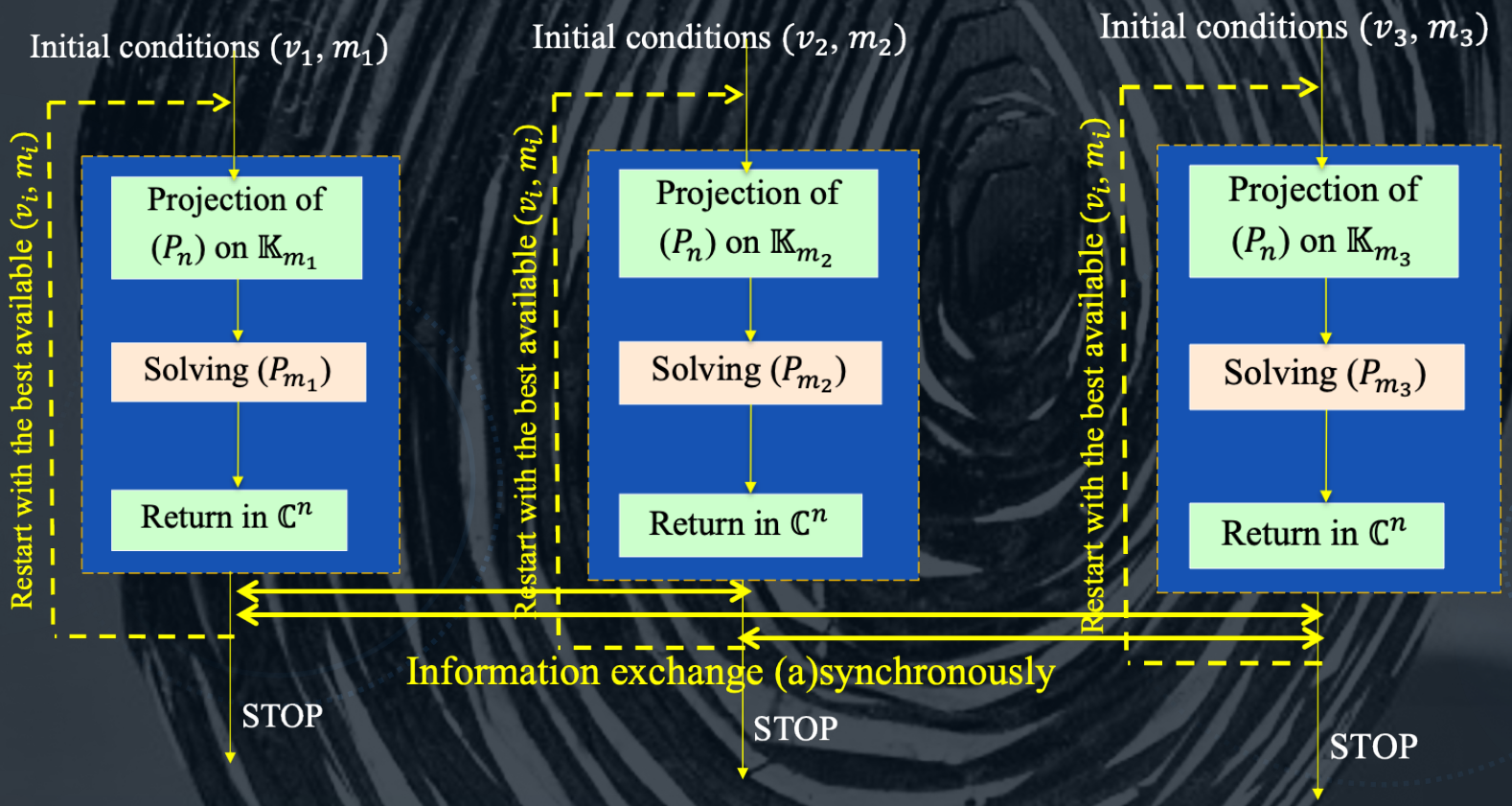


Well suited to high-scale computing systems

N. Emad, S. Petiton. ***Unite and Conquer Approach for High Scale Numerical Computing****, Journal of Computational Science, ISSN-1877-7503, 2016*

A Specific Case of UC Methods: Multiple-Methods

IRAM (Implicitly Restarted Arnoldi Method)



How to Further Improve Subspace Quality ?

Improving $\mathbb{K}_m(A, v) = span\,(v, Av, \ldots, A^{m-1}v)$ using both m and v

Different subspaces

$\mathbb{K}_{m_i}(A, v_i) = span\,(v_i, Av, \ldots, A^{m_i-1}v_i)$,

with $m_i \neq m_j$ and $v_i \neq v_j$ (for $i, j \in [1, \ldots, \ell]$ and $i \neq j$)

Nested subspaces ($m_1 < m_2 < \ldots < m_\ell$)

$\mathbb{K}_m(A, v) =$

$span\,(v, Av, \ldots, A^{m_1-1}v, A^{m_1}v, \ldots, A^{m_2-1}v, \ldots, A^{m_{\ell-1}}v, \ldots, A^{m_\ell}v)$,

with $\mathbb{K}_{m_1} \subset \mathbb{K}_{m_2} \subset \ldots \subset \mathbb{K}_{m_\ell}$.

Performance of Unite and Conquer methods

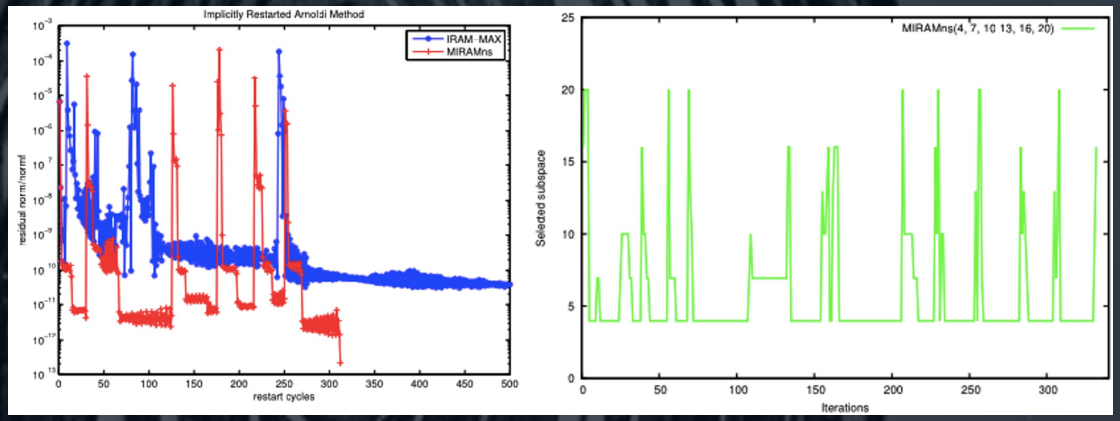
Good convergence even for difficult input

Crucial finding : the best subspace in which we search for principal axes is not always the largest.

Track for researching the number of clusters in unsupervised learning

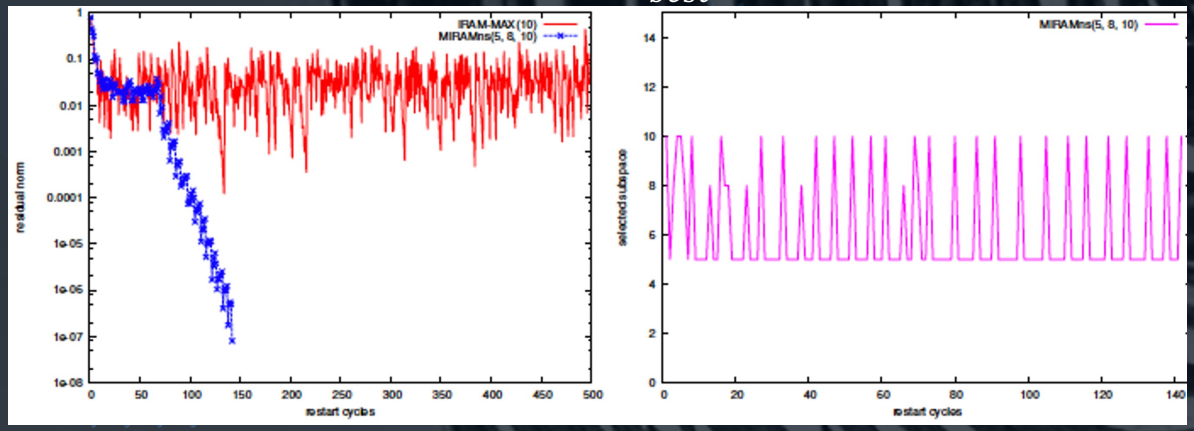
Iterations/collaboration concept in AI algorithms like EL or MixExpert

RoadNet-PA matrix, 8 subspaces

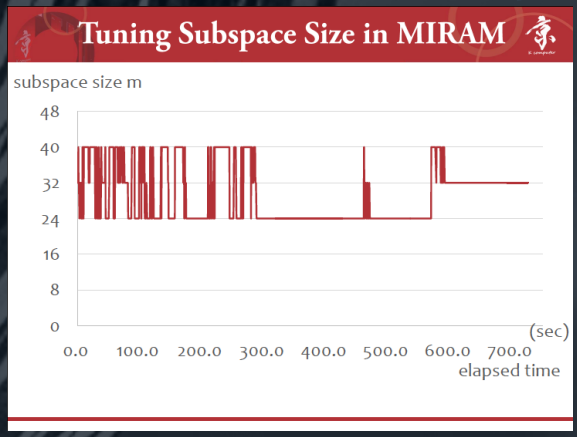


Convergence of MIRAMns vs IRAM and Evolution of m_{best} in MIRAMns along restart cycles

Autotuned IRAM and Evolution of m_{best} within ARPACK



Bfw782a, $k = 2$, $tol = 10^{-8}$ and a random initial guess



Schenk/nlpkkt240, n=27993600

APPLICATION OF DDR WITH MIRAMNS IN NN

	Date Fruit Red Dim: $34 \rightarrow 5$		Fetal Health Red Dim: $21 \rightarrow 5$		Radar Red Dim: $175 \rightarrow 12$	
	No Emb.	Emb.	No Emb.	Emb.	No Emb.	Emb.
Number of parameters	939,015	233,991	1,063,427	233,475	1,143,815	233,991
Training Exec. time (s)	2.085	1.354	3.7349	2.371	1112.81	716.668
Train accuracy (%)	97.63	93.04	95.94	91.88	98.13	97.95
Evaluation accuracy (%)	94.44	89.99	93.66	90.84	98.79	98.36

Performance with and without dimension reduction for different data

- A quarter reduction in network size (at least) while minimizing loss of precision.
- The performance improvement is an increasing function of the size of the input (implying reduction in large model training time).

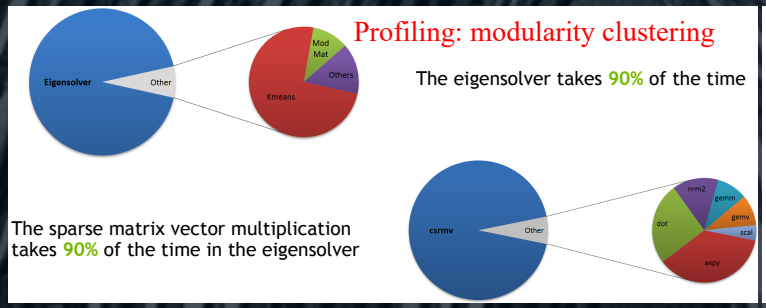
Work in progress with Mines-Paris & Huawei (Chong Li, Quentin Petit, …)

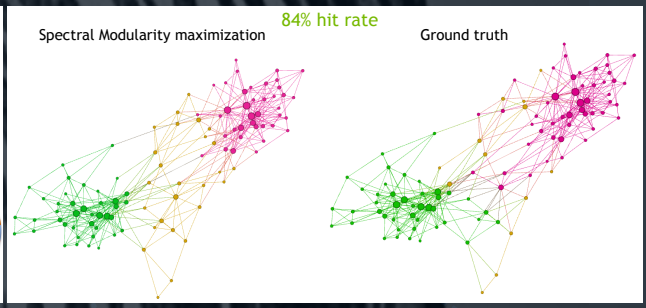
CLUSTERING AND ANOMALY DETECTION WITH UC METHODS

Clustering

NVIDIA (nvGraph library)

MIRAMns, Mlnzos_ns





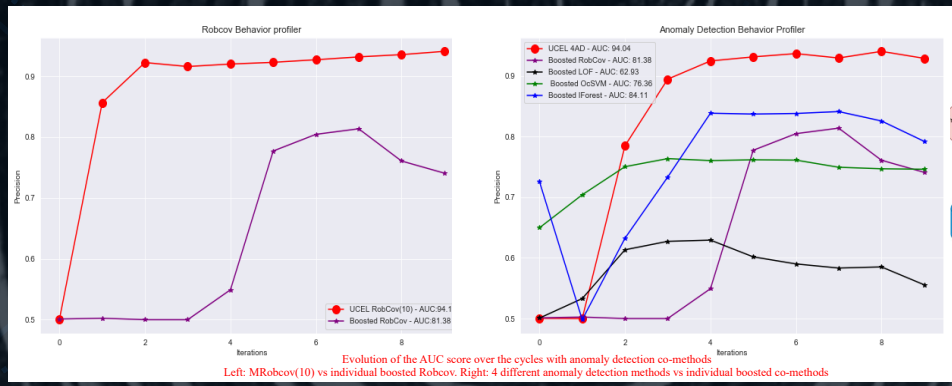
A. Fender, N. Emad, S. Petiton, M. Naumov, *Parallel Modularity Clustering*, Procedia Computer Science, Volume 108, 2017, Pages 1793-1802

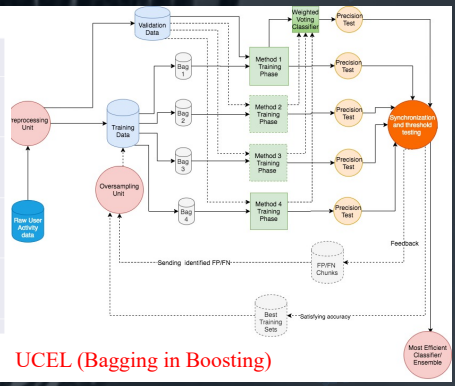
Anomaly detection

UCEL (ATOS)

UC2B Generalization of UCEL(NUMERYX)

Z. Ziani (ANOTHER BRAIN)





A. Diop, et al. *A Unite and Conquer Based Ensemble Learning Method for User Behavior Modeling*, IEEE IPCCC, 2020

Concluding Remarks

- Nonlinear approaches often use linear approaches
- Need to improve spectral computation techniques
- The extensibility of the UC approach opens multiple avenues for new techniques
  - Excellent convergence properties even for difficult input
  - Big finding : a key to choose the best search subspace to mine for dominant eeigenespace
    - This best subspace for searching principal axes is not always the largest.
- Using the concept of iterations/collaboration in AI algorithms like EL or MoE

Concluding Remarks

Trade-off between reducing costs and maintaining the quality of results

Two important challenges in dimensionality reduction :

A. What is the optimal number of principal axes to search ?

B. What is the optimal size of the search space for A ?