JYU DISSERTATIONS 165

Yariv Aizenbud

Random Projections for Matrix Decomposition and Manifold Learning

UNIVERSITY OF JYVÄSKYLÄ
FACULTY OF INFORMATION
TECHNOLOGY

JYU DISSERTATIONS 165

Yariv Aizenbud

Random Projections for Matrix Decomposition and Manifold Learning

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston Agora-rakennuksen Beeta-salissa
joulukuun 13. päivänä 2019 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in building Agora, Beeta hall, on December 13, 2019 at 12 o'clock noon.

JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2019

Editors
Timo Männikkö
Faculty of Information Technology, University of Jyväskylä
Ville Korkiakangas
Open Science Centre, University of Jyväskylä

Permanent link to this publication: http://urn.fi/URN:ISBN:978-951-39-7965-2

ISBN 978-951-39-7965-2 (PDF)
URN:ISBN:978-951-39-7965-2
ISSN 2489-9003

ABSTRACT

Aizenbud, Yariv
Random Projections for Matrix Decomposition and Manifold Learning
Jyväskylä: University of Jyväskylä, 2019, 26 p. (+included articles)
(JYU Dissertations
ISSN 2489-9003; 2489-9003; 165)
ISBN 978-951-39-7965-2 (PDF)

The thesis focuses on solving problems that are related to the behavior of random variables in high-dimensional spaces. The main motivation comes from the understanding that many of the scientific challenges involve large amounts of high-dimensional data. It is known that there are always a small number of "hidden" parameters that encode the "interesting" part of the data. The question is, how do we identify and extract these parameters? This thesis is focused on two different aspects of data analysis: Numerical linear algebra and manifold learning.

Numerical linear algebra is a major component for data analysis. It includes matrix factorization algorithms such as SVD and LU. SVD is considered to be the single most important algorithm in numerical linear algebra. However, due to the computational complexity of classical SVD algorithms, they cannot be applied in practice to huge datasets. One possible solution to this problem is to use low-rank methods. The idea of low-rank methods is the fact that in many cases there are dependencies and redundancies within the data. Therefore, the data can be well approximated and processed by utilizing its low-rank property which results in a faster processing of smaller data. In this thesis, Low-rank SVD and LU approximation algorithms are presented. They create a trade-off between accuracy and computational time. We improve on the state-of-the-art algorithms for Low-rank SVD and LU approximation. Since matrix factorization algorithms play a central central role in almost any modern computation, this part of the thesis provides general tools for many of the modern big data, and data analysis challenges.

Understanding high-dimensional data via manifold learning. Many data analysis problems are formulated in the language of manifold learning. A typical assumption is that the data is on (or near) some unknown manifold embedded in high dimensions, and the goal is to "understand" the structure of this manifold. The thesis presents two result on this subject. First, a connection between two of the most classical methods in manifold learning, PCA and least squares, is presented. Secondly, a method for regression over manifold is presented. It allows to interpolate functions defined on manifolds given only the values of the function in several sampled points, without knowing the manifold on which the function is defined. The ability to solve regression problems over manifolds, can enable us to gain new insights from complex sampled data.

Keywords: Matrix decompositions, Random projections, SVD, LU, manifold learning, Regression over manifolds

TIIVISTELMÄ (ABSTRACT IN FINNISH)

Aizenbud, Yariv
Satunnaisprojektiot matriisin tekijöihin jakamisessa ja monisto-oppimisessa
Jyväskylä: University of Jyväskylä, 2019, 26 s. (+artikkelit)
(JYU Dissertations
ISSN 2489-9003; 2489-9003; 165)
ISBN 978-951-39-7965-2 (PDF)

Väitöskirja keskittyy ratkaisemaan ongelmia, jotka liittyvät satunnaisten muuttujien käytökseen korkeaulottuvuuksissa avaruuksissa. Päämotivaatio tulee siitä ymmärryksestä, että monet tieteelliset haasteet sisältävät suuria määriä korkeaulottuvuuksellista dataa. Samalla tiedetään, että pieni määrä ″piilossa pysyviä″ parametreja määrittää datan ″mielenkiintoisia″ osia. Kysymys kuuluukin, kuinka pystymme tunnistamaan ja uuttamaan nämä parametrit. Tämä väitöskirja keskittyy data-analyysin kahteen alueeseen: Numeeriseen lineaariseen algebraan ja ″manifold learning″ eli ″monisto-oppimisen″ termillä tunnettuun lähestymistapaan.

Tässä väitöskirjassa esitellään Low-rank SVD ja LU approksimaatioalgoritmit. Ne muodostavat kompromissin tarkkuuden ja laskenta-ajan välillä. Nämä metodit parantavat nykyaikaisia algoritmeja Low-rank SVD ja LU approksimaatioille. Uudet tulokset poistavat rajoitteita, jotka liittyvät ratkaistavissa olevien haasteiden kokoon ja tarkkuuteen. Koska matriisin tekijöihin jakaminen on keskiössä melkein kaikessa nykyaikaisessa laskennassa, tämä väitöskirjan osio tarjoaa yleisesti hyödynnettäviä työkaluja moneen nykyaikaiseen suuren datan ja data-analyysin ongelmanratkaisuun.

Väitöskirjan tulokset todistavat satunnaisprojektioiden tehokkuuden erilaisissa matriisin tekijöihin jakamisen metodeissa, sekä antavat uuden näkökulman korkeiden ulottuvuuksien data-analyysille. Nämä kaksi aluetta ovat useiden tutkimusongelmien ytimessä. Tällä tutkimuksen alueella on yhä useita vastaamattomia kysymyksiä, joiden tutkimuksessa satunnaisprojektiot voivat olla avuksi.

Avainsanat: Matriisien tekijöihin jakaminen, Satunnaisprojektiot,SVD, LU, monisto-oppiminen, regressio monistoja hyödyntäen

Author

Yariv Aizenbud
Faculty of Information Technology
University of Jyväskylä
Finland

Supervisors

Professor Pekka Neittaanmäki
Faculty of Information Technology
University of Jyväskylä
Finland

Professor Amir Averbuch
Faculty of Information Technology
University of Jyväskylä
Finland

Reviewers

Professor Larisa Beilina
Chalmers University of Technology and University of Gothenburg
Sweden

Dr. Dan Kushnir
Nokia Bell Labs
Murray Hill, NJ
USA

Opponent

Professor Keijo Ruotsalainen
University of Oulu
Finland

ACKNOWLEDGEMENTS

I would like to thank my supervisors, Prof. Amir Averbuch and Prof. Pekka Neittaanmäki, for all their guidance, their help, and for making this research possible. I am grateful for the opportunity of working with Amir and Pekka.

I would like to thank all my colleagues with whom I worked on the research in this thesis. In particular, many thanks are reserved to the coauthors of the papers in this thesis, namely, Dr. Gil Shabat, Dr. Yaniv Shmueli, Dr. Barak Sober and Prof. David Levin. I wish to express my great appreciation to Prof. Yoel Shkolinsky for fruitful discussions that had an important impact on the nature of my research. I am certain that without all these people this research would not have been so enjoyable, enriching and rewarding.

LIST OF FIGURES

CONTENTS

LIST OF INCLUDED ARTICLES

PI Gil Shabat, Yaniv Shmueli, Yariv Aizenbud, Amir Averbuch. Randomized LU Decomposition. *Applied and Computational Harmonic Analysis*, 44(2), 246-272, 2018.

PII Yariv Aizenbud and Amir Averbuch. Matrix Decompositions Using sub-Gaussian Random Matrices. *Information and Inference: A Journal of the IMA*, 8.3, 445-469, 2018.

PIII Yariv Aizenbud and Barak Sober. Approximating the Span of Principal Components via Iterative Least-Squares. *arXiv preprint arXiv:1907.12159*, 2019.

PIV Barak Sober, Yariv Aizenbud and David Levin. Approximation of Functions over Manifolds: A Moving Least-Squares Approach. *arXiv preprint arXiv:1711.00765*, 2019.

In [PI] the author developed the fast-randomized LU algorithm (Algorithm 4.4), including its computational complexity calculations and the theoretical guarantees. Additionally the author was an integral part in the CPU and GPU code development, experiment design, and the writing of Section 5.

In [PII] the author developed the presented algorithm, proved the theoretical guarantees together with Prof. Averbuch. the author also developed the code (now publicly available), and experiment design.

In [PIII] the author proved the equivalence of the iterative algorithm (Algorithm 3) to the Subspace iterations algorithm (Theorem 1).

In [PIV] the author, jointly with Dr. Sober, developed the regression algorithm (Algorithms 1 and 2) based on tools developed by Dr. Sober.

1 INTRODUCTION

In the modern world, many of the problems that concern scientists both in academia and industry have to do with how to manipulate large amounts of data. A simple familiar example is provided by the ranking of web pages when searching the Internet. There are billions of pages and each page has many parameters that determine its content. The problem is to understand which parameters or combinations of parameters are more relevant for ranking and which should be considered as “noise". It is known for many years that there is always a small number of parameters that encode the important part of the data. The question is how do we extract them? In many cases, an equally important question is how to extract them efficiently.

It is useful and illuminating to think that many of the phenomena in nature that we do not understand are related to characterizing and understanding random processes. Modeling a phenomenon via the properties of random variables is a challenging task, but in many cases, such modeling can lead to important new results. Maybe the simplest example is that averaging of many samples of a signal with additive random noise with zero mean will be close to the clean signal. This observation and its generalizations can help us denoise multidimensional data with a low computational cost. Another example is the use of random matrices. Random matrices can be used as a powerful tool in analyzing high dimensional big data problems. While processing raw data from high dimensional problems is difficult and in many cases seems like an impossible task, the usage of random matrices can reformulate the task to become feasible. Random matrix theory is an active field of research that deals with the structure of matrices with random entries. A fascinating new aspect of the advancements in this area is that we can learn a lot about questions related to the “behavior" of data by the way our data “interacts" with a random matrix. This idea leads to new and astonishing results in many areas. Random matrices are applicable in numerous areas such as thermodynamics, quantum chaos and structural biology.

The theses focuses on two areas, which at first sight looks very different, but in each of which, deep understanding of random behavior can result in novel theories and algorithms superior to existing approaches:

1. Random projections for matrix factorization (see Section 1.1).
   In this section we present two results:
   (a) Randomized LU decomposition (based on [PI])
   (b) Matrix Decompositions Using sub-Gaussian Random Matrices (based on [PII])
2. Manifold Learning (see Section 1.2).
   In this section we present two results:
   (a) Approximating the Span of Principal Components via Iterative Least-Squares (based on [PIII])
   (b) Approximation of Functions over Manifolds: A Moving Least-Squares Approach (based on [PIV])

1.1 Random projections for matrix factorization

Recently, developments that are based on random matrix theory resulted in new algorithms for some of the classical building blocks of numerical linear algebra (a recent review is given in [8, 32]). These algorithms provide an approximate answer for several basic problems in numerical linear algebra while allowing to adjust the trade-off between speed and precision for matrix decomposition related problems.

More formally, the idea of randomized decomposition algorithms is that given a matrix A of size $m \times n$ (assume $m \geq n$) and a random matrix G of size $n \times k$, the product AG is computed to obtain a smaller matrix that potentially captures most of the range of A. In most of these applications, k is set to be much smaller than n to obtain a compact approximation for A. The core of this idea stands on the Johnson-Lindenstrauss Lemma (JL) [10]. The JL Lemma shows that there is a random distribution of linear dimensionality reduction operators that preserves, with bounded error and high probability, the norm of a set of vectors. For example, Gaussian random matrices satisfy this property.

Matrix factorizations play an important role in many applications. There are several ways to decompose a matrix that provide different insights into the matrix/operator structure. We will focus on the Singular Value Decomposition and the LU decomposition.

The **Singular Value Decomposition (SVD)** factorizes a real or complex matrix. It generalizes the eigendecomposition of a positive semidefinite normal matrix i.e., a symmetric diagonalizable matrix with positive eigenvalues, to any $m \times n$ matrix. Formally, the singular value decomposition of an $m \times n$ matrix A is a decomposition of the form

$$A = U\Sigma V^*$$

where $U \in M^{m\times m}$ and $V \in M^{n\times n}$ are real or complex unitary matrices and $\Sigma \in M^{m\times n}$ is a rectangular diagonal matrix with non-negative real numbers on the diagonal. The values on the diagonal of Σ are called singular values.

Intuitively, SVD enables us to represent a matrix as a composition of a rotation, scaling (different scaling in each axis) and another rotation. The singular values tell us how much scaling is done in each axis. Large singular values tell us that there are directions (vectors) which, after applying A, get much larger. The largest singular value of A is exactly the operator norm (L_2-norm) of A. Small singular values tell us about directions/vectors that after application of A almost or completely vanish. In many cases, small singular values are considered as "noise" and are discarded.

As mentioned above, SVD has a strong connection to eigendecomposition. Indeed, from a theoretical point of view, if we consider the positive semi-definite matrix AA^*, then by using the SVD decomposition we can write

$$AA^* = U\Sigma V^* V\Sigma U^* = U\Sigma^2 U^*$$

which is exactly the eigendecomposition of AA^*. Another, perspective of SVD provides us the Principal Component Analysis (PCA). The PCA of a matrix provides the principal components which are the directions in which the "data" has the largest variance. These directions are exactly the rows of U (or rows of V, depending if the "data" is written as rows or columns of the matrix A) in the SVD.

Efficient computation of the SVD or eigendecomposition is a problem of great importance. Unfortunately, there is no closed-form formula for computing these decompositions (for matrices larger than 5×5). Finding the eigenvalues of a matrix of size $n \times n$ is equivalent to finding the roots of a polynomial of degree n. Therefore, having a closed-form algorithm to find the eigenvalues of a general matrix will result in an algebraic expression for the roots of a polynomial of any given degree, which contradicts Abel-Ruffini's Theorem [9].

Most of the known approaches for solving the eigenproblem are generalizations of the Power Iterations [28, 30]. The basic idea behind the Power Iteration algorithm is to apply repeatedly A on some initial vector u, where in each iteration the output is normalized. The limit of the iterations is the most dominant eigenvector - see Algorithm 1. For a more rigorous and comprehensive treatment see [28]. The random projection approach is similar in nature, but a careful analysis shows that in many cases, a single application of the matrix on a set of random vectors will provide enough information to recover the leading singular values.

Lower–Upper (LU) decomposition factors a matrix $A \in M^{m\times n}$ as the product of a lower triangular matrix and an upper triangular matrix. The product sometimes includes permutation matrices as well. Explicitly,

$$A = LU \quad \text{or} \quad PA = LU \quad \text{or} \quad PAQ = LU,$$

where $L \in M^{m\times m}$ is a lower triangular matrix with ones on the diagonal, $U \in M^{m\times n}$ is an upper triangular matrix, and P and Q are permutation matrices. LU decomposition can be viewed as the matrix form of Gaussian elimination. LU decomposition can also be considered as an extension of the Cholesky decomposition for non-symmetric positive-definite matrices.

Algorithm 1 Power Iterations

```
Input: $A \in \mathbb{R}^{p \times p}, u \in \mathbb{R}^p$, where $u$ is the first guess for the leading eigenvector (can be chosen at random).
Output: $v \in \mathbb{R}^p$ - the leading eigenvector of $A$.
$v = u$
repeat
    $v_{prev} = v$
    $z = Av$
    $v = \frac{z}{\|z\|}$
until $\|v - v_{\text{prev}}\| < \epsilon$
```

There are numerous applications for the LU decomposition. In many cases, when machines solve systems of linear equations, and LU decomposition is used. Many problems require to solve a system of linear equations (and thus compute an LU decomposition) such as in optimization, circuit designing, and many more. LU decomposition is also a critical component when inverting a matrix or computing its determinant.

1.2 Data analysis and manifold learning in high dimensions

Manifold learning, high-dimensional data analysis, and regression over manifolds have been very active research fields for the past two decades. Big high-dimensional datasets are common in many areas nowadays. The trend of bigger and bigger datasets continues due to increasing availability of technology and continuous technological advances. Typically, in large modern datasets, both the number of observations as well as the number of features that are used are large. The number of features in a dataset is considered the dimension of the data, since each observation corresponds to a vector whose coordinates are determined by these features. Analyzing these directly poses many challenges for machine learning and data analysis methods, which are referred to as the “curse of dimensionality”. The main common theme of “curse of dimensionality” problems is the relation between the high dimension of a dataset and the volume taken by its data points in the space defined by data features. As the dimension of the dataset increases, the data points occupy an increasingly smaller portion of the space. As a result, the high dimensional representation of the data becomes too sparse to directly obtain practical useful information from it.

Principle component Analysis (PCA) is a classic, yet key method in data analysis. PCA has two equivalent interpretations: first, it finds the directions in which the variance of the data is maximal, these directions are the ones that “carry" the information in the data. The other directions, where the variance is small, are considered as noise. The second interpretation is that PCA finds linear space that approximate the data, and thus it can be thought of as linear dimensionality reduction technique. PCA is also connected to the SVD described in Section 1.1. The generality and simplicity of the PCA also contribute to its fame.

Unfortunately, in most cases, PCA does not answer all the questions in data analysis. In many cases the data in not on a linear subspace, but rather on some low-dimensional manifold. With the exception of deep neural nets, whose theoretical behavior is not well understood, the vast majority of algorithms in this field are based on embedding the data into a low-dimensional space, either linear as in PCA, or non-linear as in LLE, Isomap, Laplacian eigenmaps, diffusion maps. These embeddings necessarily introduce distortions and errors, even if the data were sampled from a manifold without noise, let alone in the noisy case. In most cases no theory allows to analyze these distortions, and so there are no bounds for the error introduced by the embedding.

One possible approach to "Learning a manifold", which does not require dimensionality reduction, is based on the Moving Least-Squares (MLS) framework. The MLS approximation was originally designed for the purpose of smoothing and interpolating scattered data, sampled from some multivariate function [14, 16, 21, 22]. Then, it evolved to deal with surfaces (i.e., $n-1$ dimensional manifolds in $\mathbb{R}^n$), which can be viewed as a function locally rather than globally [3, 17]. This has been generalized lately in [27] to the Manifold Moving Least-Squares (MMLS), which deals with manifolds of an arbitrary dimension d embedded in $\mathbb{R}^n$.

2 CONTRIBUTION

2.1 Contributions of the thesis to matrix factorization

The work in [PI], introduces a new algorithm for a fast randomized LU decomposition that provides a low rank approximation for the LU decomposition (see Algorithm 2. Formally, given an $m \times n$ matrix A and an integer r, the presented LU decomposition algorithm generates matrices P, Q, L and U such that

$$\|PAQ - LU\|_2 \leq C(m,n,r)\sigma_{r+1} \tag{1}$$

with a negligible failure probability. $C(m,n,r)$ depends on the size of the matrix A and on r and also on the desired success probability. P and Q are orthogonal permutation matrices, L is an $m \times r$ lower triangular matrix, U is a $r \times n$ upper triangular matrix and σ_{r+1} is the $(r+1)$th singular value of A. This algorithm, as in randomized SVD algorithms [8, 20, 33], projects the original (huge) matrix into a randomized low-dimensional space and then calculates the decomposition in this low-dimensional space. This approach results in a fast decomposition with proven accuracy bounds. The randomized LU has the following advantages over other randomized decompositions for low-rank approximation. First, since it is based on LU decomposition, it is faster, uses less memory since computations are done in-place, suitable to process sparse data and is fully parallelizable. See Figure 1 for running time comparison with other randomized decomposition methods. Second, the algorithm can run on a GPU with no GPU-CPU memory transfer making it 10 times faster than the same algorithm that runs on a powerful eight-core CPU. See Figure 2 for performance comparison between 8-core CPU and a GPU . Third, the approximation error of the algorithm is close to the error obtained using the randomized SVD. The analysis done in this chapter, in addition to proving the performance bounds for the presented algorithm, also improves the existing bounds for the randomized SVD decomposition. The algorithm was later improved in [2].

The work in [PII], is concerned with the Singular Value Decomposition (SVD). Formally, the SVD of an $m \times n$ matrix A is a decomposition of the form $A = U\Sigma V^*$

Algorithm 2 Randomized LU Decomposition

Input: A matrix of size $m \times n$ to decompose, k desired rank, $l \geq k$ number of columns to use.

Output: Matrices P, Q, L, U such that $\|PAQ - LU\| \leq \mathcal{O}(\sigma_{k+1}(A))$ where P and Q are orthogonal permutation matrices, L and U are the lower and upper triangular matrices, respectively.

```
1: Create a matrix $G$ of size $n \times l$ whose entries are i.i.d. Gaussian random variables with zero mean and unit standard deviation.
2: $Y \leftarrow AG$.
3: Apply rank revealing LU decomposition to $Y$ such that $PYQ_y = L_y U_y$.
4: Truncate $L_y$ and $U_y$ by choosing the first $k$ columns and the first $k$ rows, respectively, such that $L_y \leftarrow L_y(:, 1:k)$ and $U_y \leftarrow U_y(1:k, :)$.
5: $B \leftarrow L_y^{\dagger} PA$.
6: Apply LU decomposition to $B$ with column pivoting $BQ = L_b U_b$.
7: $L \leftarrow L_y L_b$.
8: $U \leftarrow U_b$.
```

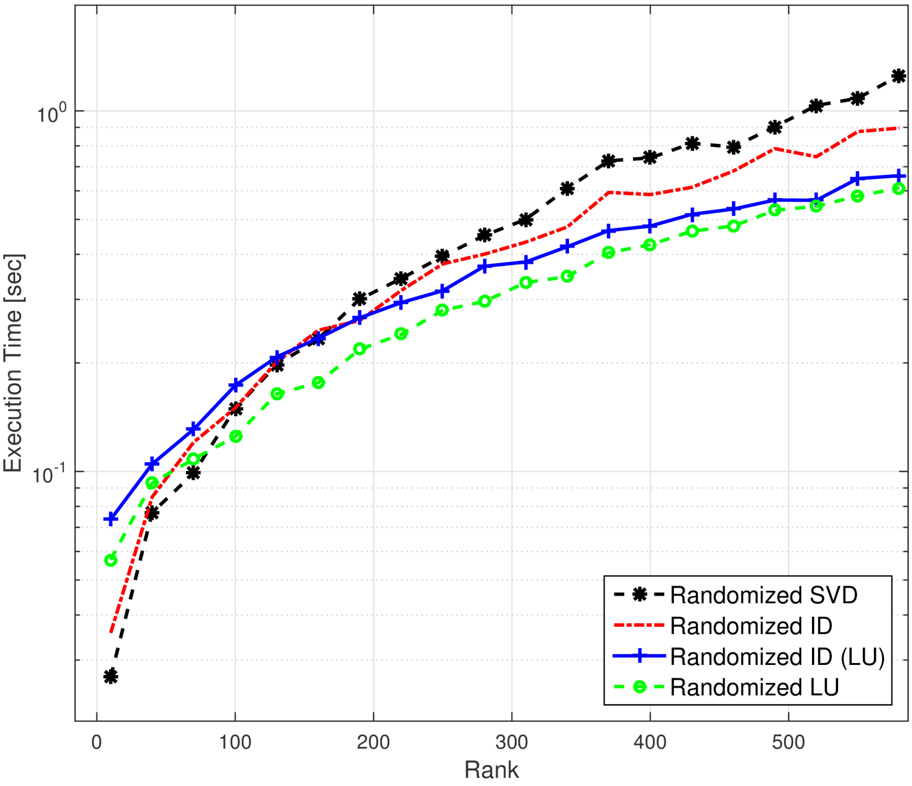

FIGURE 1 The execution times of different algorithms: Randomized SVD [20], Randomized ID [20] (QR and LU [PI]) and Randomized LU [PI] running on a single core CPU.

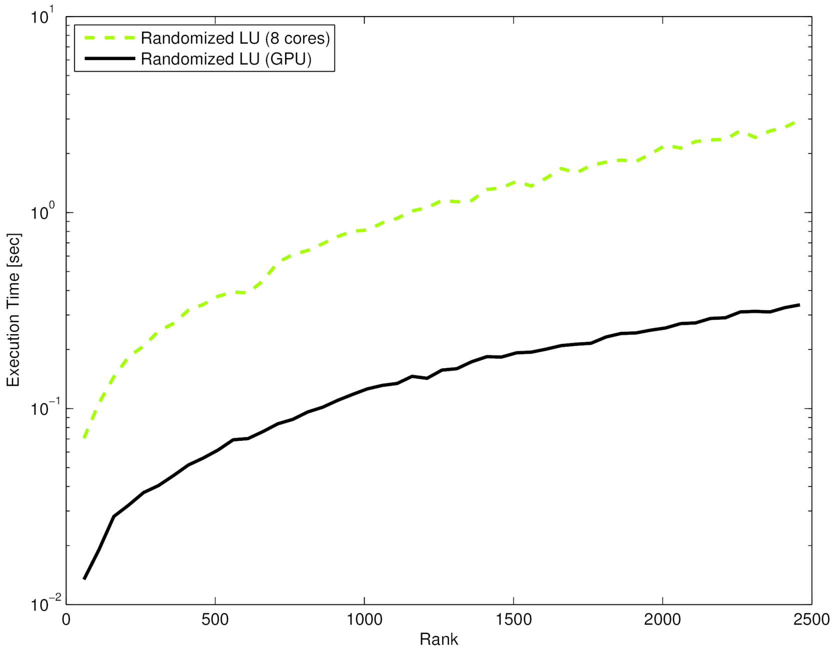

FIGURE 2 The execution times from running the randomized LU algorithm on different computational platforms: CPU with 8 cores and GPU.

where $U \in M^{m\times m}$ and $V \in M^{n\times n}$ are real or complex unitary matrices and $\Sigma \in M^{m\times n}$ is a rectangular diagonal matrix with non-negative real numbers on the diagonal. The values on the diagonal of Σ are called singular values. Efficient computation of the SVD is a problem of great importance. Unfortunately, there is no closed-form formula for computing these decompositions (for matrices larger than 5×5).

Most of the classical approaches for calculating the SVD are generalizations of the Power Iterations [28, 30]. The recently developed random projection approach is similar in nature, but a careful analysis shows that in many cases, a single application of the matrix on a set of random vectors (with JL distribution) will provide enough information to recover the leading singular values with high accuracy.

The work in [PII] presents an algorithm for low-rank SVD approximation that is based on sub-Gaussian random projections (see Algorithm 3). Formally, given an $m \times n$ matrix A and an integer r, the sub-Gaussian randomized SVD results in matrices U, Σ and V, such that U and V are orthogonal matrices, Σ is a diagonal matrix with real positive values on the diagonal. U, Σ and V satisfies

$$\|A - U\Sigma V^T\|_2 \leq C(m,n,r)\sigma_{r+1}$$

with a very high success probability. $C(m,n,r)$ depends on the size of the matrix A and on r, but also on the desired success probability. σ_{r+1} is the $(r+1)$th singular value of A. The sub-Gaussian randomized SVD, in many cases, for the same accuracy, is faster than the state-of-the-art low-rank approximation algorithms. The proof that guarantees the high accuracy with high probability of the presented algorithm relies on the fact that the used distribution of the random projections has metric conserving properties. Thus, as a by-product, we show that an i.i.d. sub-Gaussian matrix with a large probability of having null entries is metric conserving. This result is used in the SVD approximation algorithm, as well as to improve the performance of the previously proposed approximated LU decomposition algorithm.

Algorithm 3 Sub-Gaussian-based Randomized SVD

Input: Matrix A of size $m \times n$ to decompose, r desired rank, k_1, k_2, l number of columns to use.

Output: Matrices U, Σ, V such that $\|A - U\Sigma V^*\|_2 \le \mathcal{O}_\sigma(\sigma_{r+1}(A))$, where U and V are matrices with orthonormal columns and Σ is a diagonal matrix.

```
Create a random sub-Gaussian matrix $\Omega_1$ of size $k_1 \times n$.
Create a random Gaussian matrix $\Omega_1'$ of size $l \times k_1$.
Compute $B = A\Omega_1^*\Omega_1'^*$ ($B \in M_{m\times l}$).
Compute the QR decomposition: $B = QR$, $Q \in M_{m\times l}$ with orthonormal
    columns, $R \in M_{l\times l}$ is a full rank upper triangular matrix.
Create a random sub-Gaussian matrix $\Omega_2$ of size $k_2 \times m$.
Compute $\Omega_2 Q$, $\Omega_2 A$ and $(\Omega_2 Q)^\dagger$.
Compute the SVD of $(\Omega_2 Q)^\dagger \Omega_2 A = \tilde{U}_1 \Sigma_1 V_1^*$.
$U_1 \leftarrow Q\tilde{U}_1$.
$U \leftarrow U_1(:, 1:r)$.
$\Sigma \leftarrow \Sigma_1(1:r, 1:r)$.
$V \leftarrow V_1(:, 1:r)$.
```

2.2 Contributions of the thesis to data analysis

The work in [PIII], explores the relation between two of the essential building blocks of data analysis: The PCA and the Linear least squeres. Although PCA is normally addressed as a statistical tool aiming at finding orthogonal directions on which the variance is maximized, its first introduction by Pearson at 1901 was done through defining a non-linear least-squares minimization problem of fitting a plane to scattered data points. Thus, it seems natural that PCA and linear least-squares regression are somewhat related, as they both aim at fitting planes to data points. This Chapter presents a new way to find the space spanned by the leading PCs of a given matrix through an iterative least-squares procedure. Apparently, the iterative least-squares algorithm in [PIII] coincides with the well known Subspace Iterations algorithm. As a consequence each iteration of Subspace (or Power) Iterations, can be interpreted as a solution to a least-squares problem. In other words, solving a least-squares problem is equivalent to multiplying a basis with the sample covariance matrix. Even-though, computationally, it is not more efficient than the alternatives, this approach leads to a new and useful geometrical interpretation of the omnipresent PCA (e.g., see [PIV, 27]).

In the last three decades, there has been a rapid development of mathematical frameworks aiming to deal with complexity challenges, originating from high dimensional data. In many works, there exists an underlying assumption that the high dimensional domain of a sample set (i.e., point cloud) has a lower intrinsic dimension (e.g., [1, 5, 7, 11, 12, 13, 23, 24, 29]). In other words, the data points $\{r_i\}_{i=1}^N \subset \mathbb{R}^n$ are samples of a lower dimensional manifold $\mathcal{M}^d$, where d is the intrinsic dimension of $\mathcal{M}$ and $d \ll n$. Therefore, a natural way of reducing the effective number of parameters (in case of a parametric estimation) as well as computation complexity, would be to harvest this geometric relationship among the points. The framework of dimension reduction proposes to embed the data into a lower dimension Euclidean domain while maintaining some sort of local distances (for a survey see [15]). Then, the lower dimensional representations can be utilized to perform function approximation over the data. However, such methods inherently introduce distortion to the input data, as, for example, the curvature information is lost after performing such an embedding. In addition, performing out-of-sample extensions, in most of these methods, will require the re-computation of the embedding. Another effective framework, dealing with such problems is the Support Vector Machine based methods [26, 25], which in some sense are another way of performing non-linear dimension reduction prior to performing regression.

A somewhat different approach, designed to deal with a more general definition of low dimensionality, is the Geometric Multi-Resolution Analysis (GMRA), introduced in a series of papers [4, 6, 18, 19]. The GMRA utilizes a local affine representation of the data, in order to store the data in a multi-resolution dictionary. Thus, it does not project the data onto a lower dimensional Euclidean domain, but creates a tree-like representation of the original data based upon partitioning and performing local Singular Value Decomposition. This approach, leads to a faithful, locally sparse, representation of the input data in case the original tree was built from clean samples. Subsequently, these representations can be used to approximate functions over the original input data (e.g., [31]). However, this approach does not aim at yielding smooth or even continuous approximations.

The work in [PIV], presents a new method for performing regression over manifolds. The method is utilizing only noisy function values at locations sampled from the manifold with noise. Explicitly, Let $\mathcal{M}$ be a d dimensional manifold and $\psi : \mathcal{M} \to \mathbb{R}^n$ some function. Then, suppose we have a set of noisy samples of ψ, and we are interested in evaluating ψ in some new point $r \in \mathcal{M}$. Convergence results, under some noise module, are presented. To produce the approximation, the method does not require any knowledge regarding the manifold other than its dimension d. The resulting approximant is shown to be a function defined over a neighborhood of a manifold, approximating the originally sampled manifold. In other words, given a new point, located near the manifold, the approximation can be evaluated directly on that point. Also, the proposed algorithm has linear time complexity with respect to the ambient-space's dimension. Thus, we can avoid the computational complexity, commonly encountered in high dimensional approximations, without having to perform non-linear dimension reduction, which inevitably introduces distortions to the geometry of the data.

3 CONCLUSION

The Thesis has contributions in two areas: It shows the power that random projections have in different matrix decomposition methods, and gives a different perspective for high dimensional data analysis. These two areas stand in the heart of many research problems, and there are still many open questions that need to be answered, and in many cases looking at these questions with the "eye of random projections" can help us find the solutions.

Matrix decomposition methods are a basic building block in almost any computational task. The datasets in the modern world are getting larger, and of higher dimension. Although there is a continues progress in the development of matrix decomposition algorithms there are still several significant open problems, the solution of which, together with these existing techniques, will have great significance in diverse scientific fields:

1. How do we know when it is useful to apply such randomized algorithms instead of the classical ones? What is the expected error (not asymptotically)? What are the optimal (or near optimal) choices of parameters? After running an algorithm, suppose we are not satisfied with the accuracy of the results, can we improve them at some reasonable cost? Constructive solutions will make these new class of algorithms widely applicable.
2. In many important applications matrices have special structures that can be utilized in order to improve general approaches. For example, in the Filter Diagonalization Method (FDM) used in nuclear magnetic resonance (NMR) applications, a generalized eigenvalue problem needs to be solved, where one of the matrices in known to be nearly unitary. This information can help us solve the problem more efficiently. In some quantum chemistry applications, it is required to compute singular values of a large matrix. Its entries are sampled from a smooth function. Again, this special structure can help us solve the problem more efficiently.

Manifold Learning, high dimensional data analysis and regression over manifolds are very active fields of research in the last two decades. During this time numerous approaches have been proposed. In addition to deep neural nets, whose theoretical behaviour is not well understood, the vast majority of the solutions are based on embedding the data into a low-dimensional linear space where linear (PCA, ICA, etc.) or non-linear (LLE, ISOMAP, Laplacian eigenmaps, diffusion maps, etc) embedding is used. In the case of regression, after the embedding, the manifold regression becomes a regular regression. On one hand, these embeddings necessarily introduces distortions and errors even if the data were sampled from a manifold without noise, let alone in the noisy case. These distortions are difficult to analyze and thus we are unable to bound the error of the resulting dimensionality reduction methodology compared with the (not necessarily existing) “ground truth”. On the other hand, today, these are the leading methods for analyzing high dimensional data. These methods allow us to derive conclusions from datasets which otherwise where incomprehensible.

The next step of this research research is to develop a new framework that “learns” the manifold directly in the ambient space, while keeping the computations dependent only linearly on the ambient space dimension. This framework can lead to many new insights of the latent activities in the ambient space. It paves a way to reconstruct a manifold without (asymptotically) introducing errors, to denoise sampled datasets, to solve regression problems with statistical guarantees, to be able to follow geodesic lines (and thus creates a “natural” smooth transition between data points) and many more. All these problems are key in many AI and big data problems. This new approach can also be used as a preliminary step for existing dimensionality reduction algorithms that improve their performances. The ideas used in the manifold regression algorithm [PIV] are a first step in this direction, and further research looks very promising. The connection between this framework and the behaviour of random variables in high dimensions is two-fold: First, the data points are scattered on the manifold in a random fashion, and our goal is to achieve the best possible approximation relying on as few samples as possible. Second, the analysis has to utilize understanding of the noise behaviour which is a high dimensional random process. The understanding of the behaviour of these processes is essential for this framework to achieve breakthrough in the areas mentioned above.

YHTEENVETO (SUMMARY IN FINNISH)

Väitöskirja keskittyy ratkaisemaan ongelmia, jotka liittyvät satunnaisten muuttujien käytökseen korkeaulottuvuuksissa avaruuksissa. Päämotivaatio tulee siitä ymmärryksestä, että monet tieteelliset haasteet sisältävät suuria määriä korkeaulottuvuuksellista dataa. Samalla tiedetään, että pieni määrä ”piilossa pysyviä” parametreja määrittää datan ”mielenkiintoisia” osia. Kysymys kuuluukin, kuinka pystymme tunnistamaan ja uuttamaan nämä parametrit. Tämä väitöskirja keskittyy data-analyysin kahteen alueeseen: Numeeriseen lineaariseen algebraan ja ”manifold learning” eli ”monisto-oppimisen” termillä tunnettuun lähestymistapaan.

Numeerinen lineaarinen algebra on tärkeä osakomponentti data-analyysissa. Se sisältää matriisin tekijöihin jakamisen algoritmit, kuten SVD (Singular value decomposition), eli Pääakselihajotelma, ja LU (lower–upper decomposition), eli LU-hajotelma. SVD koetaan tärkeimmäksi yksittäiseksi algoritmiksi lineaarisessa algebrassa. Kuitenkin, johtuen klassisten SVD-algoritmien laskennallisesta monimutkaisuudesta, niitä ei pystytä hyödyntämään käytännönläheisesti suurissa datajoukoissa. Yksi mahdollinen ratkaisu tämän ongelman ohittamiseksi on käyttää low-rank metodeja. Ajatus onnistuneiden low-rank metodien takana perustuu siihen, että useissa tapauksissa data sisältää riippuvuuksia ja päällekkäisyyksiä. Siten, data pystytään hyvin approksimoimaan ja prosessoimaan hyödyntämällä sen low-rank ominaisuutta, joka johtaa nopeampaan pienemmän datan prosessoimiseen.

Tässä väitöskirjassa esitellään Low-rank SVD ja LU approksimaatioalgoritmit. Ne muodostavat kompromissin tarkkuuden ja laskenta-ajan välillä. Nämä metodit parantavat nykyaikaisia algoritmeja Low-rank SVD ja LU approksimaatioille. Uudet tulokset poistavat rajoitteita, jotka liittyvät ratkaistavissa olevien haasteiden kokoon ja tarkkuuteen. Koska matriisin tekijöihin jakaminen on keskiössä melkein kaikessa nykyaikaisessa laskennassa, tämä väitöskirjan osio tarjoaa yleisesti hyödynnettäviä työkaluja moneen nykyaikaiseen suuren datan ja data-analyysin ongelmanratkaisuun.

Korkeaulottuvuuksisen datan ymmärtämisessä ”Manifold learning” –metodia käyttäen, tyypillinen olettamus on, että data on jollain (tai on lähellä) matriisin tuntemattomalla korkeiden ulottuvuuksien sisällä olevalla monistolla. Päämääränä on ”ymmärtää” moniston rakenne. Väitöskirja esittää kaksi vastausta tähän aiheeseen. Ensin esitetään yhteys kahden klassisimman monisto-oppimisen, pääkomponenttianalyysin (PCA, Principal component analysis) ja Pienimmän neliösumman menetelmän, välillä. Yhteys tuo esille uutta ja parempaa ymmärrystä mainituista metodeista. Seuraavaksi esitetään moniston regression metodi. Se sallii useiden monistolla määritettyjen funktioiden regression vaatien ainoastaan funktion arvot useissa esimerkkipisteissä, ilman vaadittua tuntemusta funktion määrittävästä monistosta. Kyky ratkaista regressio-ongelmia monistojen kautta sallii uuden ymmärryksen kompleksisesta näytedatasta.

REFERENCES

[1] Y. Aizenbud, A. Bermanis, and A. Averbuch, *PCA-based out-of-sample extension for dimensionality reduction*, arXiv preprint arXiv:1511.00831, (2015).

[2] Y. Aizenbud, G. Shabat, and A. Averbuch, *Randomized LU decomposition using sparse projections*, Computers & Mathematics with Applications, 72 (2016), pp. 2525 – 2534.

[3] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, *Computing and rendering point set surfaces*, IEEE Transactions on Visualization and Computer Graphics, 9 (2003), pp. 3–15.

[4] W. K. Allard, G. Chen, and M. Maggioni, *Multi-scale geometric methods for data sets ii: Geometric multi-resolution analysis*, Applied and Computational Harmonic Analysis, 32 (2012), pp. 435–462.

[5] M. Belkin and P. Niyogi, *Laplacian eigenmaps for dimensionality reduction and data representation*, Neural computation, 15 (2003), pp. 1373–1396.

[6] G. Chen, A. V. Little, and M. Maggioni, *Multi-resolution geometric analysis for data in high dimensions*, in Excursions in Harmonic Analysis, Volume 1, Springer, 2013, pp. 259–285.

[7] R. R. Coifman and S. Lafon, *Diffusion maps*, Applied and computational harmonic analysis, 21 (2006), pp. 5–30.

[8] N. Halko, P.-G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM review, 53 (2011), pp. 217–288.

[9] N. Jacobson, *Basic algebra I*, Courier Corporation, 2012.

[10] W. B. Johnson and J. Lindenstrauss, *Extensions of Lipschitz mappings into a Hilbert space*, Contemporary Mathematics, 26 (1984), p. 1.

[11] I. Jolliffe, *Principal component analysis*, Wiley Online Library, 2002.

[12] P. W. Jones, M. Maggioni, and R. Schul, *Manifold parametrizations by eigenfunctions of the laplacian and heat kernels*, Proceedings of the National Academy of Sciences, 105 (2008), pp. 1803–1808.

[13] T. Kohonen, *Self-organizing maps*, vol. 30, Springer Science & Business Media, 2001.

[14] P. Lancaster and K. Salkauskas, *Surfaces generated by moving least squares methods*, Mathematics of computation, 37 (1981), pp. 141–158.

[15] J. A. Lee and M. Verleysen, *Nonlinear dimensionality reduction*, Springer Science & Business Media, 2007.

[16] D. Levin, *The approximation power of moving least-squares*, Mathematics of Computation of the American Mathematical Society, 67 (1998), pp. 1517–1531.

[17] ——, *Mesh-independent surface interpolation*, in Geometric modeling for scientific visualization, Springer, 2004, pp. 37–49.

[18] A. V. Little, M. Maggioni, and L. Rosasco, *Multiscale geometric methods for data sets i: Multiscale svd, noise and curvature*, Applied and Computational Harmonic Analysis, 43 (2017), pp. 504–567.

[19] M. Maggioni, S. Minsker, and N. Strawn, *Geometric multi-resolution analysis for dictionary learning*, in Wavelets and Sparsity XVI, vol. 9597, International Society for Optics and Photonics, 2015, p. 95971C.

[20] P. Martinsson, V. Rokhlin, and M. Tygert, *A randomized algorithm for the decomposition of matrices*, Applied and Computational Harmonic Analysis, 30 (2011), pp. 47–68.

[21] D. H. McLain, *Drawing contours from arbitrary data points*, The Computer Journal, 17 (1974), pp. 318–324.

[22] A. Nealen, *An as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation*, URL: http://www. nealen. com/projects, 130 (2004), p. 150.

[23] S. T. Roweis and L. K. Saul, *Nonlinear dimensionality reduction by locally linear embedding*, Science, 290 (2000), pp. 2323–2326.

[24] L. K. Saul and S. T. Roweis, *Think globally, fit locally: unsupervised learning of low dimensional manifolds*, The Journal of Machine Learning Research, 4 (2003), pp. 119–155.

[25] B. Schölkopf, A. Smola, and K.-R. Müller, *Nonlinear component analysis as a kernel eigenvalue problem*, Neural computation, 10 (1998), pp. 1299–1319.

[26] A. J. Smola and B. Schölkopf, *A tutorial on support vector regression*, Statistics and computing, 14 (2004), pp. 199–222.

[27] B. Sober and D. Levin, *Manifolds' projective approximation using the moving least-squares (mmls)*, arXiv preprint arXiv:1606.07104, (2016).

[28] G. W. Stewart, *Matrix Algorithms: Volume II: Eigensystems*, SIAM, 2001.

[29] J. B. Tenenbaum, V. De Silva, and J. C. Langford, *A global geometric framework for nonlinear dimensionality reduction*, Science, 290 (2000), pp. 2319–2323.

[30] L. N. Trefethen and D. Bau III, *Numerical linear algebra*, vol. 50, SIAM, 1997.

[31] Y. Wang, G. Chen, and M. Maggioni, *High-dimensional data modeling techniques for detection of chemical plumes and anomalies in hyperspectral images and movies*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 9 (2016), pp. 4316–4324.

[32] D. P. Woodruff et al., *Sketching as a tool for numerical linear algebra*, Foundations and Trends in Theoretical Computer Science, 10 (2014), pp. 1–157.

[33] F. Woolfe, E. Liberty, V. Rokhlin, and M. Tygert, *A fast randomized algorithm for the approximation of matrices*, Applied and Computational Harmonic Analysis, 25 (2008), pp. 335–366.

ORIGINAL PAPERS

PI

RANDOMIZED LU DECOMPOSITION

by

Gil Shabat, Yaniv Shmueli, Yariv Aizenbud, Amir Averbuch

2018 Applied and Computational Harmonic Analysis, 44(2), 246-272

https://doi.org/10.1016/j.acha.2016.04.006

https://arxiv.org/pdf/1310.7202.pdf

PII

MATRIX DECOMPOSITIONS USING SUB-GAUSSIAN RANDOM MATRICES

by

Yariv Aizenbud and Amir Averbuch 2018

Information and Inference: A Journal of the IMA, 8.3, 445-469

https://doi.org/10.1093/imaiai/iay017

https://arxiv.org/pdf/1602.03360.pdf

PIII

APPROXIMATING THE SPAN OF PRINCIPAL COMPONENTS VIA ITERATIVE LEAST-SQUARES

by

Yariv Aizenbud and Barak Sober 2019

Submitted to Proceedings of the National Academy of Sciences

https://arxiv.org/pdf/1907.12159.pdf

PIV

APPROXIMATION OF FUNCTIONS OVER MANIFOLDS: A MOVING LEAST-SQUARES APPROACH

by

Barak Sober, Yariv Aizenbud and David Levin 2019

Submitted to Foundations of Computational Mathematics

https://arxiv.org/pdf/1711.00765.pdf