Dimensionality Reduction with Principal Component Analysis

Name: Trinh Tran

Matriculation: 1105425

Instructor: Damir Dobric

Course: Software Engineering

Date: 2019 - 03 - 12

Layout

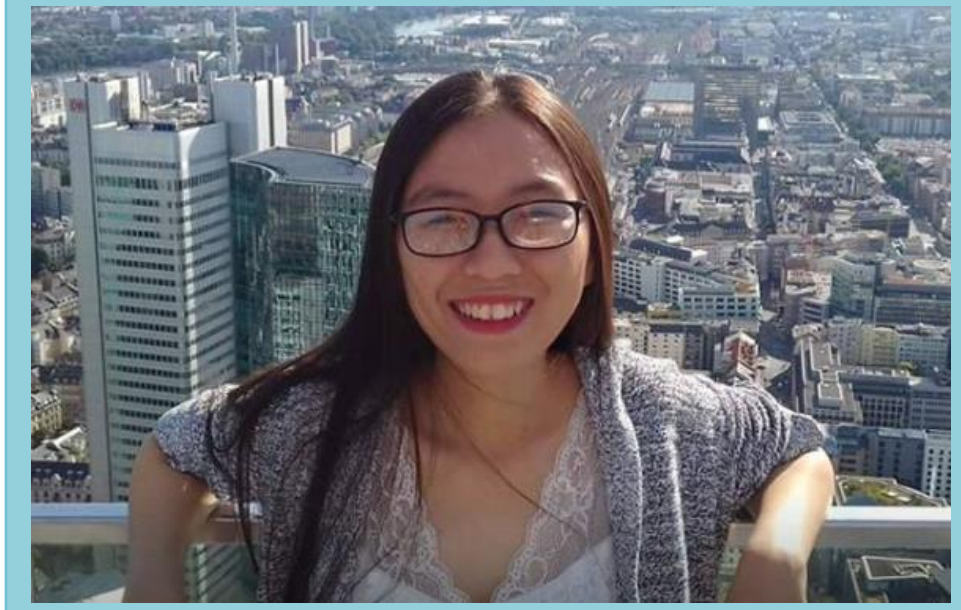
- Introduction
  - Which reduce?
  - Why reduce?
- Methods
  - How reduce ?
  - PCA
- Implementation
  - How reduce ?
  - .NET Core
- Test & Discussion
  - Unit Test
  - Integration Test
  - Operational Test

Introduction

Methods

Implementation

Test & Discussion



“An image is worth a thousand words”

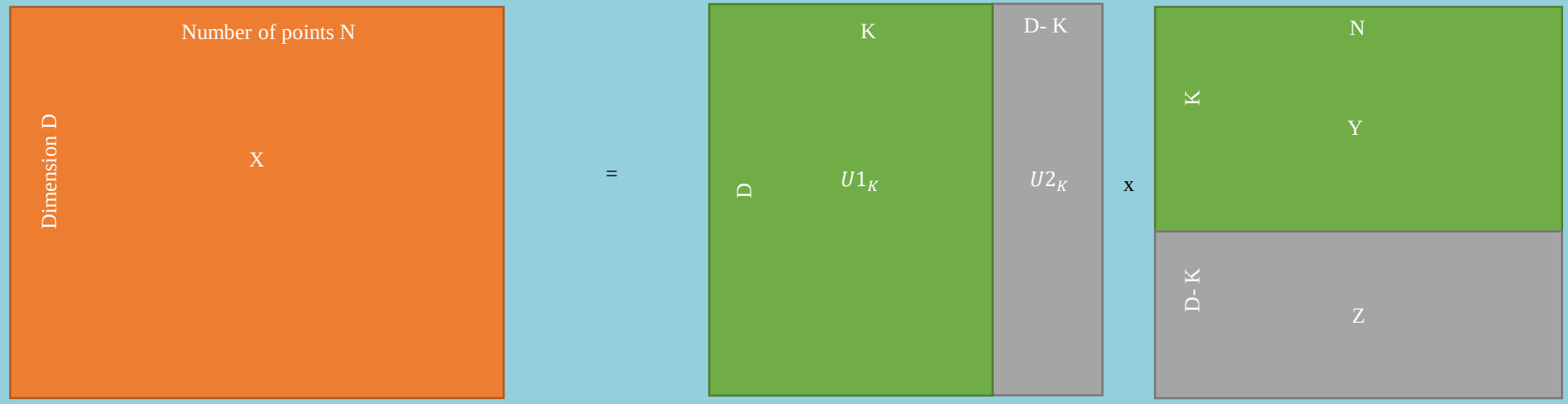
➔ But it’s too much!

Introduction

Methods

Implementation

Test & Discussion



$$\boldsymbol{X = U_1 Y + U_2 Z}$$

Main idea of PCA:

- System U could be divided into two sub-spaces :

  More important $\boldsymbol{U_1}$ and less important $\boldsymbol{U_2}$

  ➔ Keep $\boldsymbol{U_1}$ and remove $\boldsymbol{U_2}$

1. Find the mean vector of the input data:

$$\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_N$$

2. Transform to the zero-corrected data:

$$\widehat{x_N} = x_N - \bar{x}$$

3. Find covariance matrix:

$$S = \frac{1}{N}\hat{X}\hat{X}^T$$

4. Compute eigenvalues and eigenvectors of S

5. Sort the eigenvalues descending and choose only K most important elements.

6. Project the input data to the new subspace:

$$Y = U_K^T \hat{X}$$

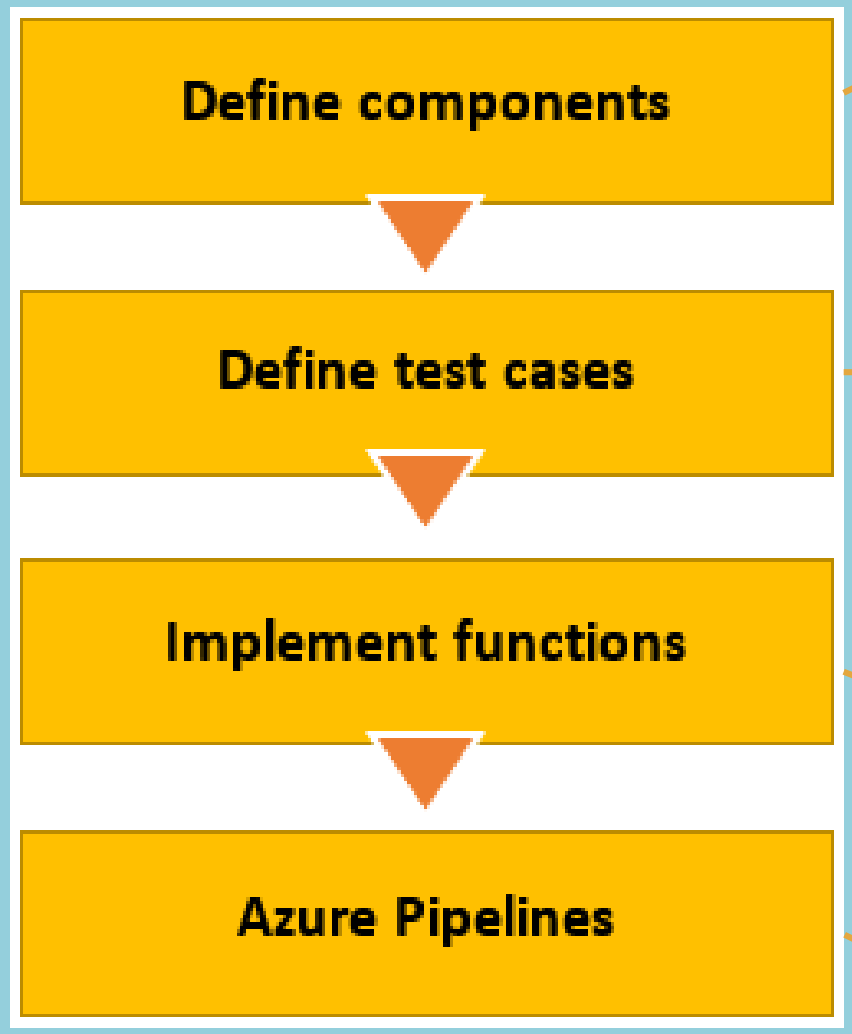
Introduction

Methods

Implementation

Test & Discussion

Test Driven Development



1. Main component – PCA
2. Linear Algebra Operations (Abstract)

➔ PCA calls LAO

Based on the LAO (abstract) – define test cases

➔ Test calls LAO

Based on the LAO (abstract) – implement functions IF

➔ IF plug to LAO

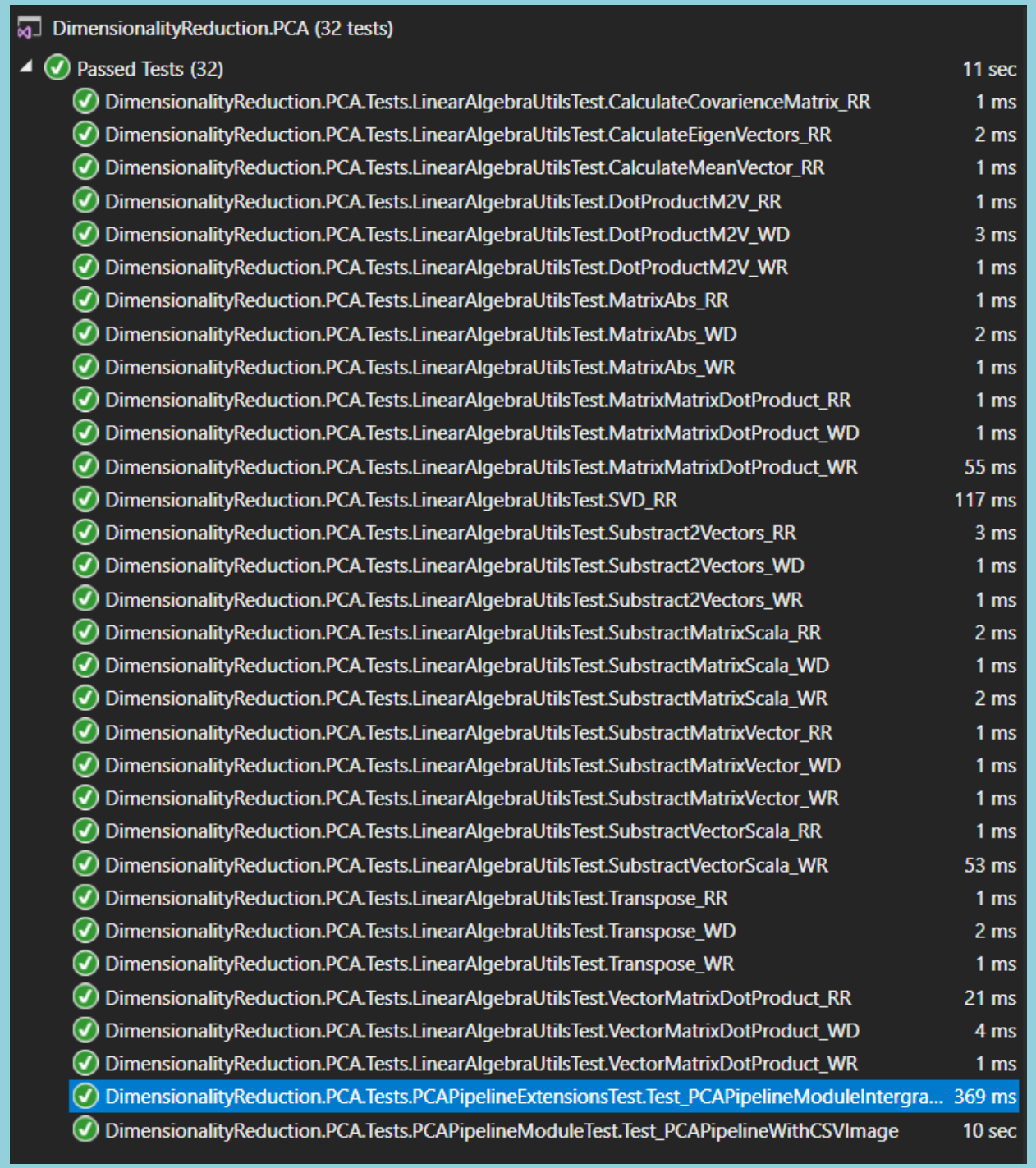
Build + Run Test + Package

Introduction

Methods

Implementation

Test & Discussion



- Linear Functions such as covariance matrix computation, matrix vector dot product, etc.
- Each function should be tested once with a true expectation, a wrong expectation and a wrong dimension.
- In summary, there are 32 test cases developed and run successfully with XUnit.

Introduction

Methods

Implementation

Test & Discussion

Apply PCA - Reduce dimension to compress an image.

From left to right: input image, loss 1%, loss 3%, loss 5%, loss 10%.

More loss – Less components involved – Less quality output

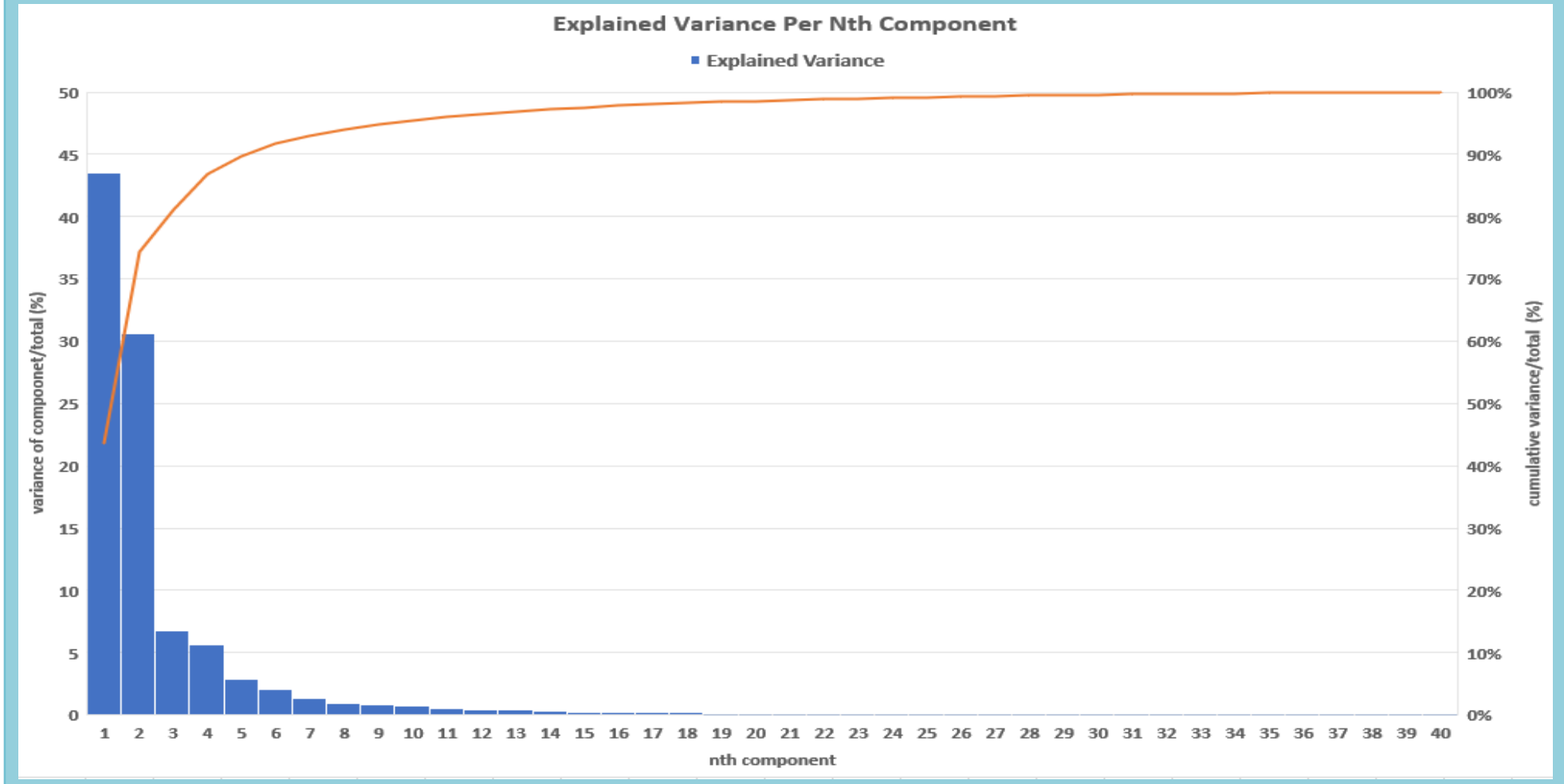
– Lower dimensional data set - More efficient in processing

Introduction

Methods

Implementation

Test & Discussion



- No. components ~ their importance.
- The blue column is the importance of nth component (descending).
- The red line is the level of output quality when cumulating from 1st component to nth component.

Thank you for your time!

Name: Trinh Tran

Email: phrong@stud.fra-uas.de

LinkedIn: https://www.linkedin.com/in/trinh-nguyen-phuong-tran-2a0553142/

Date: 2019 - 03 - 12