Clustering and Disjoint Principal Component Analysis (CDPCA) in Financial Markets

Syed Muhammad Zeeshan Bukhari (123155)
Silvia Mastracci (123177)
Oleksandr Solovei (126784)

November 19, 2024

Introduction

Problem Statement

- High dimensionality of financial market data
- Complex correlations between market sectors
- Missing data in financial time series adds another layer of complexity
- Challenge in identifying distinct market patterns

Objectives

- Apply CDPCA to financial market data
- Introduce a new element by integrating missing data handling into the CDPCA framework.
- Address the challenge of missing data using different techniques
- Identify disjoint components in market sectors

Motivation

Why CDPCA for Financial Markets?

- PCA: Components can be hard to interpret due to mixed loadings
- K-means alone: Misses underlying market structure
- Need: Clear sector-based patterns for investment decisions

Key Advantages

- Disjoint components: Each asset belongs to exactly one component
- Simultaneous clustering: Groups similar assets together
- Enhanced interpretability: Clear sector-based patterns

CDPCA Model

Mathematical Framework

$$\mathbf{X} = \mathbf{U}\hat{\mathbf{Y}}\mathbf{A}' + \mathbf{E}$$

where:

- $\mathbf{X}$: Matrix of asset returns $(I \times J)$
- $\mathbf{U}$: Asset cluster membership $(I \times P)$
- $\hat{\mathbf{Y}}$: Cluster centroids $(P \times Q)$
- $\mathbf{A}$: Component loading matrix transposed $(Q \times J)$
- $\mathbf{E}$: Error matrix $(I \times J)$

CDPCA Algorithm

Alternating Least Squares (ALS) Steps

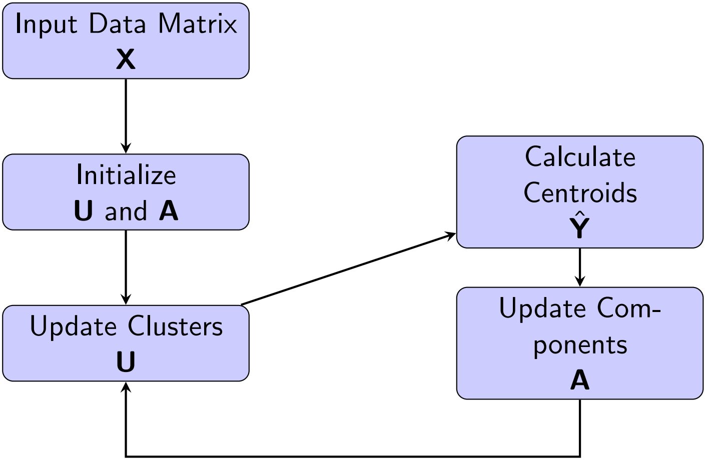
1. Update asset clusters ($\mathbf{U}$)
2. Calculate cluster centroids ($\hat{\mathbf{Y}}$)
3. Update component loadings ($\mathbf{A}$)
4. Repeat until convergence

Optimization Problem

Maximize between-cluster variance:

$$\max_{\mathbf{U},\hat{\mathbf{Y}},\mathbf{A}} \|\mathbf{U}\hat{\mathbf{Y}}\mathbf{A}'\|^2$$

CDPCA Visualization: Process Flow



Introducing Missing Data Challenge

- **Challenge: Missing Data in Time Series**
  - In financial and stock market data, missing values are common due to various reasons, such as:
    - Stock market holidays
    - Errors in data collection or reporting
    - Partial data availability across different time series
  - These missing values can significantly impact the results of principal component analysis (PCA), and even more so for CDPCA.

Data Overview

Data Source and Structure

- Source: Yahoo Finance (quantmod R)
- Period: 2022-01-01 to 2023-12-31 (252 trading days/year)
- Daily market data for 9 stocks

Variables and Properties

Price Open, High, Low, Close, Adjusted Close (adjusted for corporate actions)

Volume Daily trading volume (number of shares traded)

- Data standardized, missing values removed
- Dimensions: Rows (trading days), Columns (6 variables)

Data Structure & Coverage

Dataset Overview

- **Observations**: 4,509 total
- **Period**: 501 trading days
- **Variables**: 6 per stock
- **Missing Values**: None after cleaning

Variable Statistics

- All variables standardized
- **Volume Range**: -0.91 to 6.79
- **Price Range**: -1.33 to 2.57
- High correlations within price variables

Market Sectors

Technology (NASDAQ)

- AAPL: Apple Inc
- MSFT: Microsoft Corp
- GOOGL: Alphabet Inc

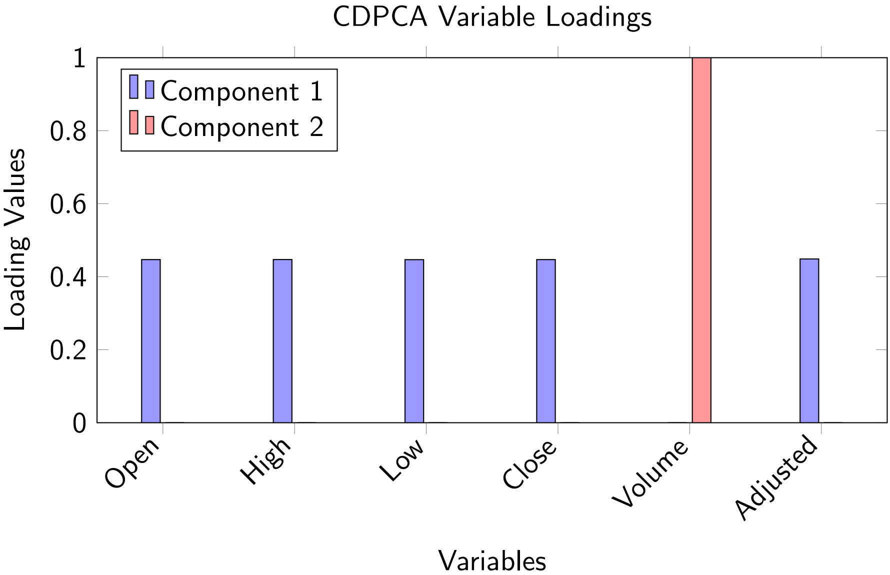
Finance (NYSE)

- JPM: JPMorgan Chase
- V: Visa Inc
- MA: Mastercard Inc

Consumer (NYSE)

- KO: Coca-Cola Co
- PG: Procter & Gamble
- WMT: Walmart Inc

Variable Structure



Handling Missing Data in CDPCA

- Imputation: Replacing missing values with statistical estimates (mean, median, or regression-based methods).
- Model-based Methods: Using algorithms that can handle missing data directly (e.g., Expectation-Maximization, Bayesian approaches).
- Data Filtering: Removing rows or columns with excessive missing values (but potentially losing information).

Mean Imputation

Original Data

Stock prices for 5 days:

$$X = [100, 102, \text{NA}, 103, 101]$$

Step-by-step Imputation

1. Sum non-missing values:
   $100 + 102 + 103 + 101 = 406$
2. Calculate mean using formula $\frac{\sum_{i=1}^{n} x_i}{n}$:
   $\bar{x} = \frac{406}{4} = 101.5$
3. Replace NA:
   $X_{\text{imputed}} = [100, 102, \mathbf{101.5}, 103, 101]$

Median Imputation

Original Data

Stock prices for 5 days:

$$X = [100, 102, \text{NA}, 103, 101]$$

Step-by-step Imputation

1. Arrange non-missing values in ascending order: $[100, 101, 102, 103]$
2. Identify the median:
   For $n = 4$, the median is the average of the middle two values:
   $\text{Median} = \frac{101+102}{2} = 101.5$
3. Replace NA:
   $X_{\text{imputed}} = [100, 102, \mathbf{101.5}, 103, 101]$

K-Nearest Neighbors (KNN) Imputation

Original Data

Stock prices for 5 days:

$$X = [100, 102, \text{NA}, 103, 101]$$

Step-by-step Imputation

1. Choose $k = 2$ nearest neighbors based on the most similar observations in the dataset:
   - Consider values close in time or other similar variables.
   - Here, neighbors are 102 and 103 (values adjacent to the missing value in this example).
2. Calculate the mean of k nearest neighbors:
   $\hat{x} = \frac{102+103}{2} = 102.5$
3. Replace NA:
   $X_{\text{imputed}} = [100, 102, \mathbf{102.5}, 103, 101]$

Expectation-Maximization (EM) Imputation

Original Data

Stock prices for 5 days:

$$X = [100, 102, \text{NA}, 103, 101]$$

Step-by-step Imputation

1. Assign an initial guess for NA (e.g., the mean of observed values): $X_{\text{init}} = [100, 102, 101.5, 103, 101]$.
2. Expectation: Use the observed data and current estimates to compute statistical parameters (mean μ, variance σ^2). Example: $\mu = 101.9$, $\sigma^2 = 1.56$.
3. Maximization: Update the missing value based on these parameters. Replace NA with the expected value conditioned on μ and σ^2: $X_{\text{updated}} = [100, 102, \mathbf{101.9}, 103, 101]$.
4. Repeat Expectation and Maximization steps until convergence.

Data Filtering

Original Data

Stock prices for 6 days:

$$X = [100, 102, \text{NA}, 103, \text{NA}, 101]$$

Approach: Filter Out Missing Data

1. Identify Missing Values: Locate the positions of missing data: $X_{\text{NA}} = [\text{Index } 3, \text{Index } 5]$.
2. Remove Rows with Missing Values: Exclude any rows (or days) with NA values: $X_{\text{filtered}} = [100, 102, 103, 101]$.
3. Resulting Dataset: Filtered data only contains complete records:

$$X_{\text{filtered}} = [100, 102, 103, 101]$$

Method Comparison

Key Findings

- Data Filtering: Highest cluster deviance but information loss
- EM: Best balance between deviance and error
- Mean/Median: Similar performance, simple implementation
- KNN: Good for preserving local structure

Recommendations

- Use EM for comprehensive analysis
- Consider KNN for pattern preservation
- Data Filtering when *quality* > *quantity*

Experimental Setup

Dataset Configuration

- Stocks: 9 (3 sectors $\times$ 3 stocks)
- Variables per stock: 6
- Time period: 2022-2023
- Missing rate: 5%

CDPCA Parameters

- Number of clusters (P): 3
- Number of components (Q): 2
- Tolerance: 10^{-5}
- Maximum iterations: 100

Implementation Details

R Implementation

- Libraries: quantmod, Amelia, VIM
- Data Structure: 501 trading days $\times$ 6 variables
- Missing Data: 5% randomly introduced

Key Functions

- `CDpca()`: Core CDPCA algorithm
- `get_stock_data()`: Data acquisition
- `amelia()`: EM imputation
- `kNN()`: KNN imputation

Implementation Example

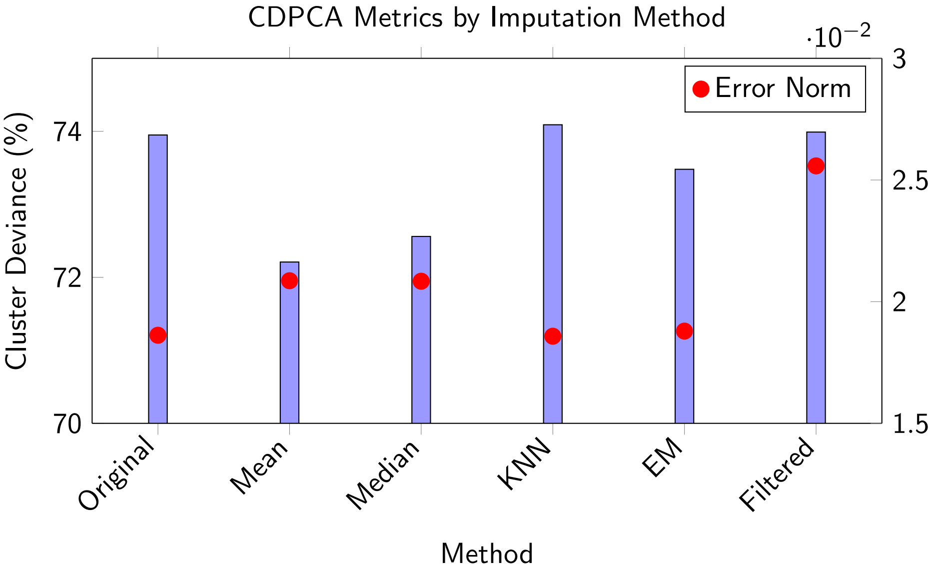
R Code for Missing Data Handling

```
# Create missing data pattern
missing_count <- floor(n * p * 0.05)
missing_indices <- sample(1:(n*p), missing_count)
X_missing[missing_indices] <- NA

# Apply EM imputation
X_em <- amelia(X_missing, m=1)$imputations[[1]]

# Run CDPCA
cdpc_results <- CDpca(X_em, P=3, Q=2,
                      tol=1e-5, maxit=100)
```

Imputation Results



Detailed Method Comparison

Method	Deviance	Error	Time	Memory
Original	73.95%	0.01862	-	-
Mean	72.21%	0.02086	Fast	Low
Median	72.56%	0.02084	Fast	Low
KNN	74.09%	0.01858	Medium	Medium
EM	73.48%	0.01879	Slow	High
Filtered	73.99%	0.02558	Fast	Low

Summary: CDPCA and Missing Data

CDPCA Contributions

- Combines imputation and PCA.
- Handles different data missing patterns.
- Improves latent data representation.

Why CDPCA?

- Reduces bias in imputation.
- Enhances interpretability.
- Efficient for high-dimensional data.