CEFET/RJ

Dimensionality Reduction & Curvature Analysis

High-dimensional datasets often contain redundant features that complicate analysis and reduce model efficiency. Principal Component Analysis (PCA) transforms data into fewer orthogonal components while preserving maximum variance. Curvature analysis provides an objective, automated approach to selecting the optimal number of components—eliminating guesswork from the dimensionality reduction process.

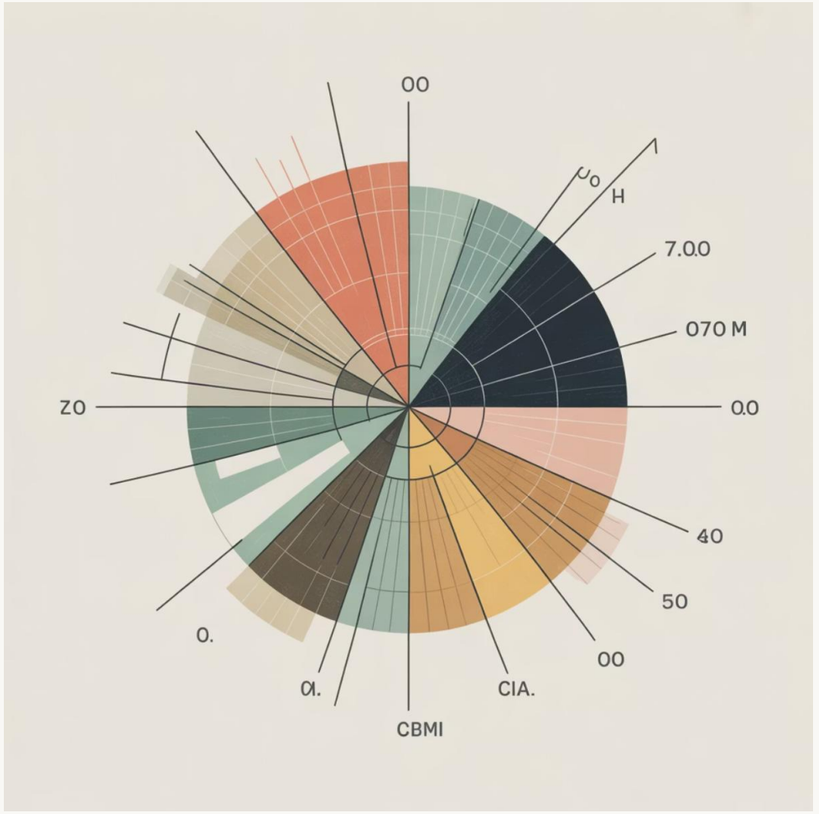
Eduardo Ogasawara
eduardo.ogasawara@cefet-rj.br
https://eic.cefet-rj.br/~eogasawara

PCA Fundamentals: Theory & Concepts

How PCA Works

PCA projects high-dimensional data onto orthogonal principal components ordered by the variance they capture. The first component captures the maximum variance, the second captures the next highest variance orthogonal to the first, and so on.

This transformation removes redundancy by eliminating correlations between features, creating a compact representation that retains the most important information.



Orthogonal Projection
Data mapped to uncorrelated axes

Maximum Variance
Components ranked by information captured

Redundancy Removal
Correlated features consolidated

Efficiency Gains
Faster models with fewer features

References: Han, J., Kamber, M., & Pei, J. – *Data Mining: Concepts and Techniques* (3rd Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – *Data Mining: Practical Machine Learning Tools and Techniques* (4th Ed.)

Implementing PCA with DAL Toolbox

The DAL Toolbox makes PCA implementation straightforward with just three simple steps: create a PCA object, fit it to your data, and transform your dataset. Here's a practical example using the classic Iris dataset:

```
# Create PCA transformation object
mypca <- dt_pca("Species")
# Fit PCA model to training data
mypca <- fit(mypca, datasets::iris)
# Transform dataset to principal components
iris_pca <- transform(mypca, datasets::iris)
# View transformed data
head(iris_pca)
```

This concise syntax handles standardization, eigenvalue decomposition, and transformation automatically. The result is a reduced-dimension dataset ready for downstream analysis or modeling.

01

Initialize

Create PCA object specifying target variable

02

Fit

Learn principal components from training data

03

Transform

Project data onto new component space

Complete working example: https://github.com/cefet-rj-dal/daltoolbox/blob/main/examples/transf/dal_pca.md

The Component Selection Challenge

How many components should you keep?

Too Many Components

Retaining unnecessary dimensions defeats the purpose—complexity and noise remain in your model

Too Few Components

Aggressive reduction risks losing critical information patterns that drive model performance

The Goldilocks Zone

Curvature analysis objectively identifies the optimal balance point automatically

Arbitrary rules like "keep 95% variance" or fixed component counts ignore the unique structure of each dataset. Curvature methods adapt to your data's specific characteristics, finding the natural inflection point where additional components add minimal value.

References: Han, J., Kamber, M., & Pei, J. – *Data Mining: Concepts and Techniques* (3rd Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – *Data Mining: Practical Machine Learning Tools and Techniques* (4th Ed.)

Curvature Analysis: Finding the Elbow

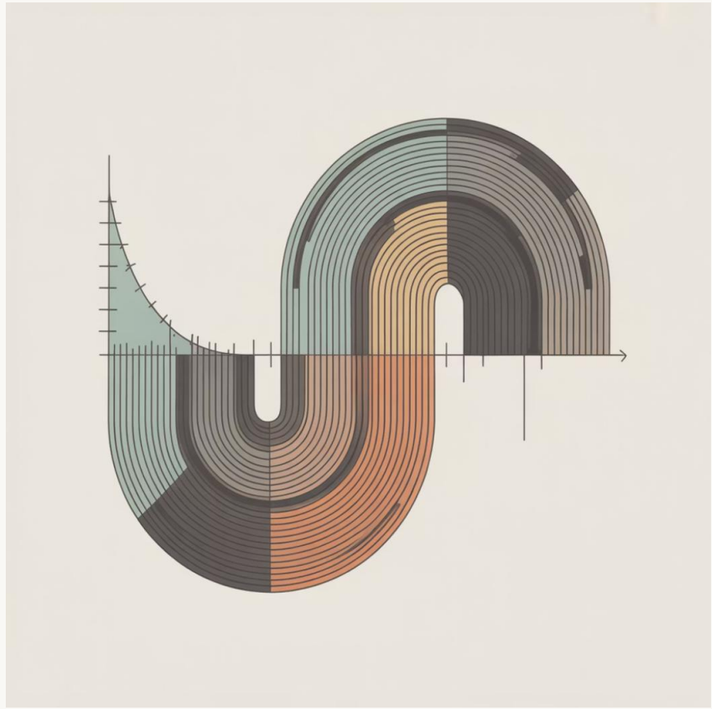
Mathematical Approach to Parameter Selection

Curvature analysis mathematically identifies inflection points—the "elbow" or "knee"—in curves where the rate of change shifts dramatically. This provides an objective criterion for parameter selection that adapts to your data.

Two curvature methods serve different purposes:

- **Curvature minimum:** Used for increasing curves like cumulative variance—finds where gains plateau
- **Curvature maximum:** Applied to decreasing curves like model error—locates where improvements taper off

Both approaches eliminate subjective threshold choices, automating what traditionally required manual inspection and domain expertise.



Objective Detection

Mathematical curvature calculation removes subjective judgment from parameter selection

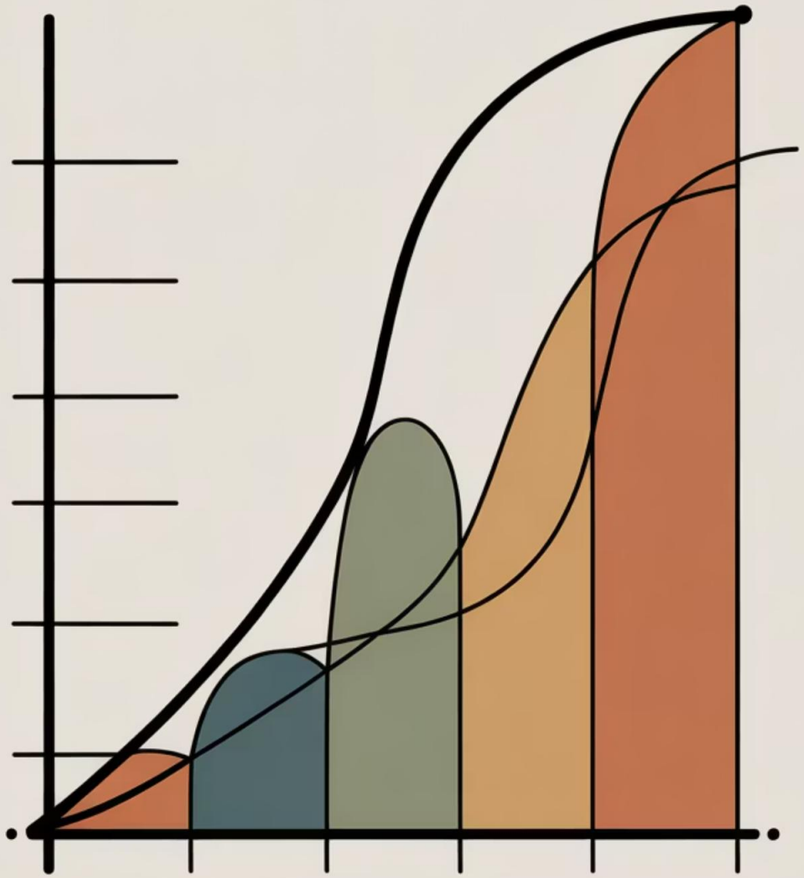
Adaptive Method

Works across different data distributions and variance patterns without preset thresholds

Reproducible Results

Same data produces identical component recommendations every time

References: Han, J., Kamber, M., & Pei, J. – *Data Mining: Concepts and Techniques* (3rd Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – *Data Mining: Practical Machine Learning Tools and Techniques* (4th Ed.)



Practice: Curvature Minimum

Curvature minimum identifies the elbow point in *increasing* curves. For PCA, we apply it to cumulative explained variance to find where additional components provide diminishing returns. Here's how to implement it with the Iris dataset:

```
# Perform PCA on Iris features
pca <- prcomp(datasets::iris[,1:4], center=TRUE, scale.=TRUE)
# Calculate cumulative proportion of variance
y <- cumsum(pca$sdev^2/sum(pca$sdev^2))
# Fit curvature minimum detector
km <- fit_curvature_min()
# Find optimal number of components
res <- transform(km, y)
res$x
# Returns recommended component count
```

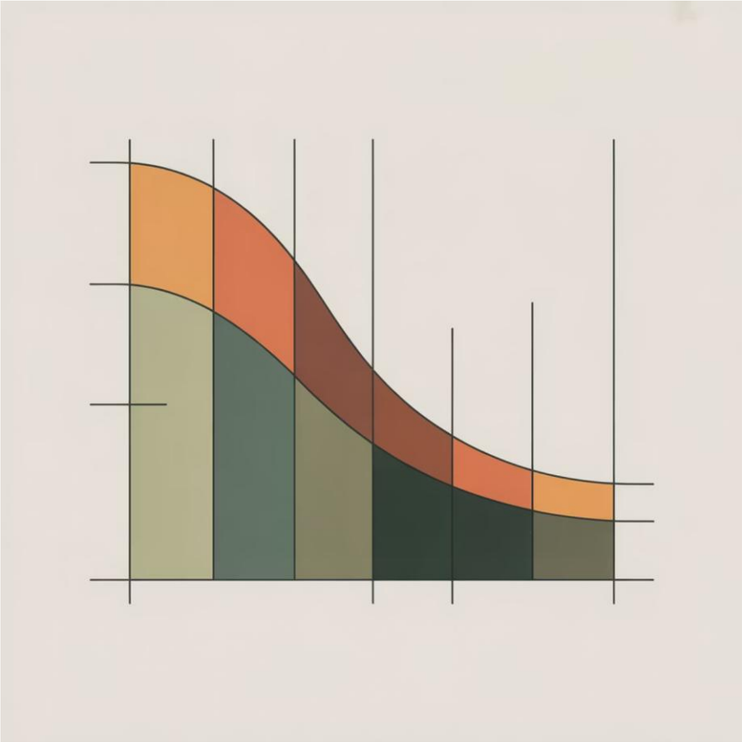
The algorithm analyzes the second derivative of the variance curve to pinpoint where the rate of variance gain slows significantly. This component count balances information retention with dimensionality reduction.

Explore the full implementation: https://github.com/cefet-rj-dal/daltoolbox/blob/main/examples/transf/curvature_minimum.md

Practice: Curvature Maximum

Curvature maximum detects inflection points in *decreasing* curves, ideal for scenarios like model error reduction or loss functions. The method identifies where improvements begin to plateau. Here's a demonstration with a logarithmic decay curve:

```
# Create example decreasing curve
x <- seq(1, 10, by=.5)
y <- -log(x)
# Fit curvature maximum detector
km <- fit_curvature_max()
# Identify elbow point
res <- transform(km, y)
res$x
# Returns x-value at maximum curvature
```



When to Use

- Training error curves
- Cross-validation loss
- Information decay
- Convergence analysis

The maximum curvature point indicates where continuing the process yields minimal additional benefit—perfect for early stopping decisions.

Complete code example: https://github.com/cefet-rj-dal/daltoolbox/blob/main/examples/transf/curvature_maximum.md

Key Takeaways

PCA Reduces Dimensions

Transform high-dimensional data into compact representations that preserve maximum variance and eliminate redundancy

Curvature Automates Selection

Mathematical curvature analysis objectively identifies optimal component counts without arbitrary thresholds

DAL Toolbox Integration

Seamless R implementation combines PCA transformation with curvature-based parameter optimization in one toolkit

Reproducible Decisions

Replace subjective parameter choices with data-driven, reproducible methods that adapt to each dataset's unique structure

---

Ready to implement? Visit the DAL Toolbox repository for complete examples, documentation, and additional dimensionality reduction techniques.