PCA for Feature Reduction

Group 3:

Preston Buscay, Johannes Djupesland, Tore Gude, Cameron Louis Penne, Nicole Quattrini

Overview

- Introduction
- Quick return to PCA
- Significance Testing
- Example
- Pros/Cons

Introduction

Why do we need feature reduction?

- Improve model performance
- Reduce computational cost
- Guiding feature engineering

What is feature reduction by PCA?

- Filtering method: only look at X
- Reduce dimensionality by PCA
- Analyze which features contribute to selected PCs

PCA

Data Standardisation

- Normalisation of the features to have where mean = 0 and st.dev = 1

Covariance Matrix

- Identification of covariance within the X matrix

Eigen-decomposition

- Based on covariance matrix: linear transformation into eigenvectors & eigenvalues
- Identification of potential components that capture the most variance.

Select PCs

- Quantification of the variance explained by new features

Significance test

- Determine which variables are important for each of the principal components
- Based on an uncertainty estimates computed from the PCA loadings
- PCA loadings are computed from:
  - Full dataset
  - Cross-validated dataset

Source: F. Westad, M. Hersleth, P. Lea, and H. Martens, 'Variable selection in PCA in sensory descriptive and consumer data', *Food Quality and Preference*, vol. 14, no. 5, pp. 463–472, Jul. 2003, doi: 10.1016/S0950-3293(03)00015-6.

Significance test

$$s(p_{ak}) = \sqrt{\left(\sum_{m=1}^{M} (p_{ak} - p_{ak(-m)})^2\right) \frac{M-1}{M}}$$

- a: index of principal component
- k: index of variable (spectral band)
- M: number of cross-validation segments
- p_{ak}: loadings (full dataset)
- $p_{ak(-m)}$: loadings (cross-validation segment)

Significance test

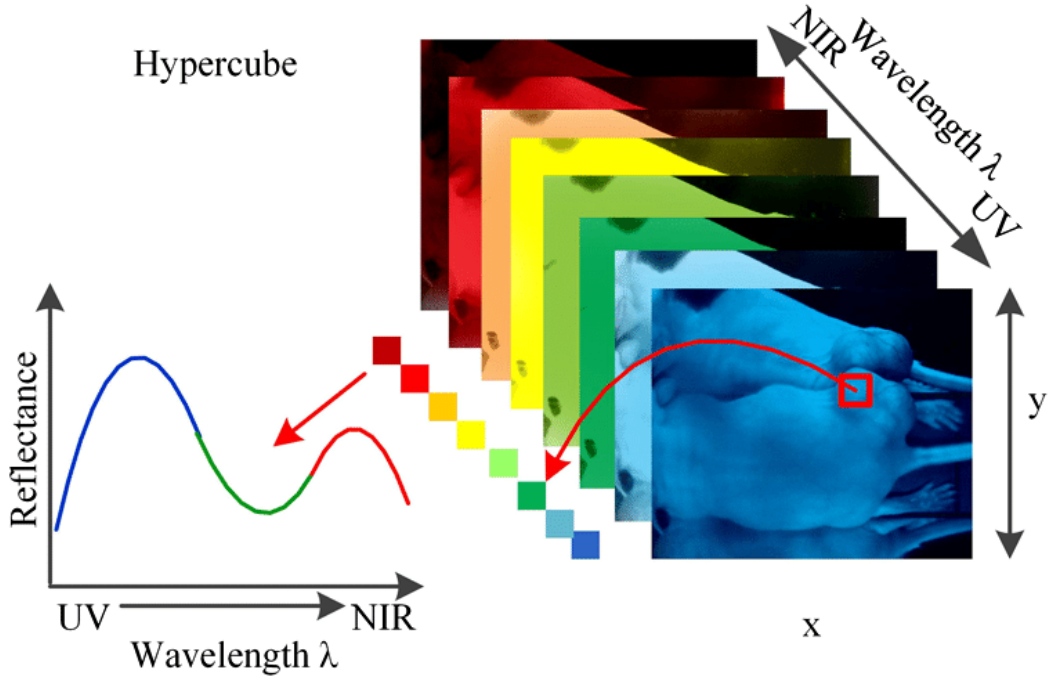
$$s(p_{ak}) = \sqrt{\left(\sum_{m=1}^{M} (p_{ak} - p_{ak(-m)})^2\right) \frac{M-1}{M}}$$

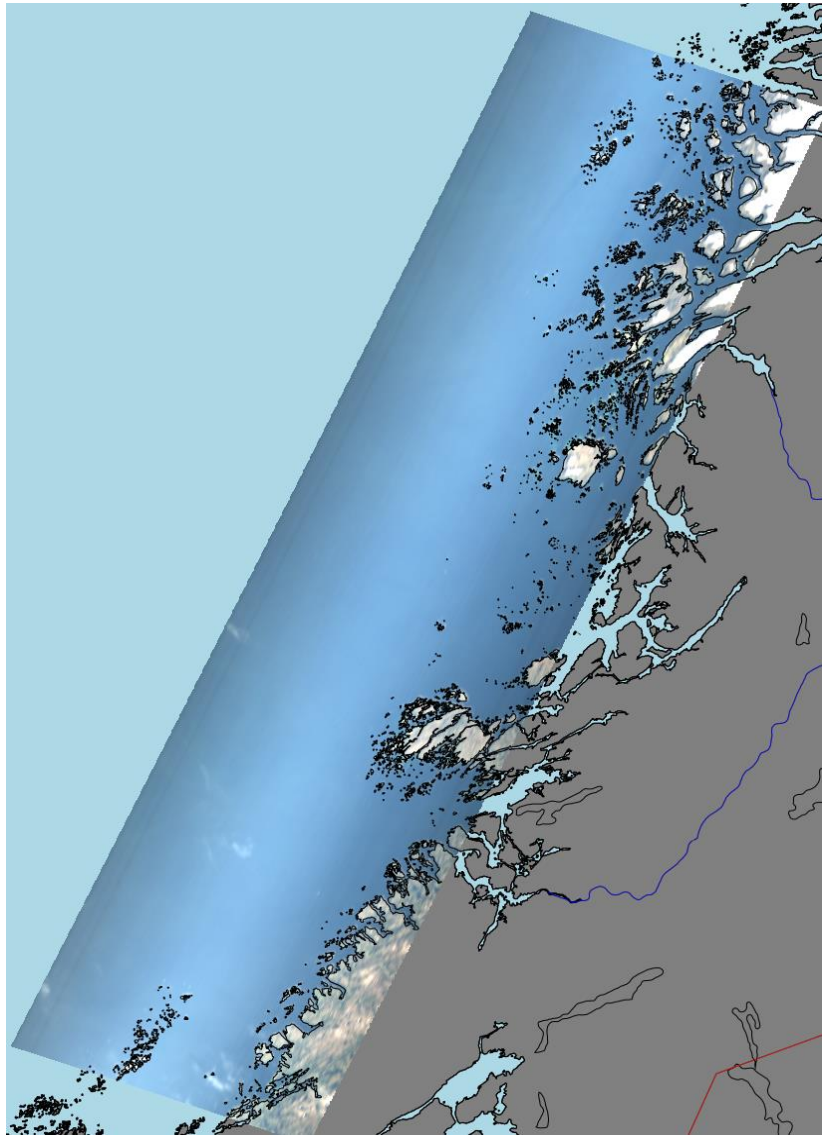
- a: index of principal component
- k: index of variable (spectral band)
- M: number of cross-validation segments
- p_{ak}: loadings (full dataset)
- $p_{ak(-m)}$: loadings (cross-validation segment)

Compute difference, square, and sum across all cross-validation folds

Example - Data

- 300k+ samples: pixels
- 115 variables: bands 400nm to 800nm





Example – PCA

$$s(p_{ak}) = \sqrt{\left(\sum_{m=1}^{M} (p_{ak} - p_{ak(-m)})^2\right) \frac{M-1}{M}}$$

```
# Run PCA on full dataset
n_components = 10 # number of components
pca = PCA(n_components=n_components)
pca.fit(X_scaled)
loadings = pca.components_.T * np.sqrt(pca.explained_variance_)
p_ak = loadings.T
del pca
```

Example – Running PCA

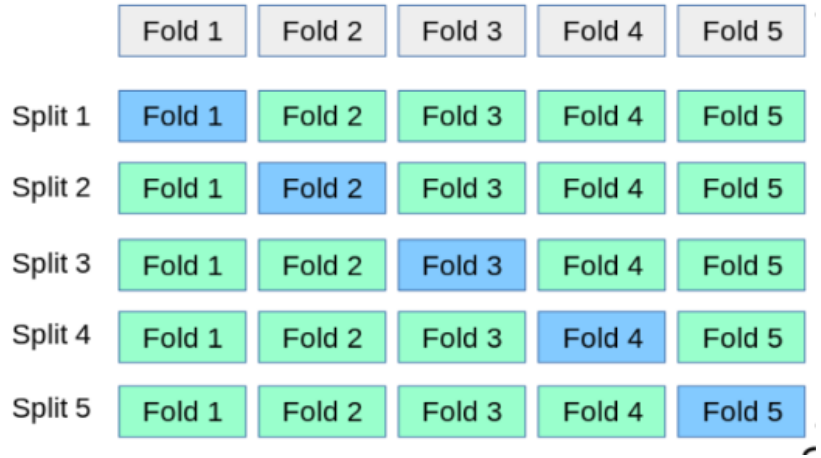
$$s(p_{ak}) = \sqrt{\left(\sum_{m=1}^{M} (p_{ak} - p_{ak(-m)})^2\right) \frac{M-1}{M}}$$

```
M = 25 # number of folds
kf = KFold(n_splits=M, shuffle=True)

# Run PCA on each cross-validation segment
p_akm = np.zeros((M, n_components, n_variables))
for m, (train_index, test_index) in enumerate(kf.split(X)):
    pca = PCA(n_components=n_components)
    pca.fit(X_scaled[train_index])
    loadings = pca.components_.T * np.sqrt(pca.explained_variance_)

    C , _ = orthogonal_procrustes(loadings.T, p_ak)
    loadings = loadings.T @ C

    p_akm[m,:,:] = loadings
    del pca
```

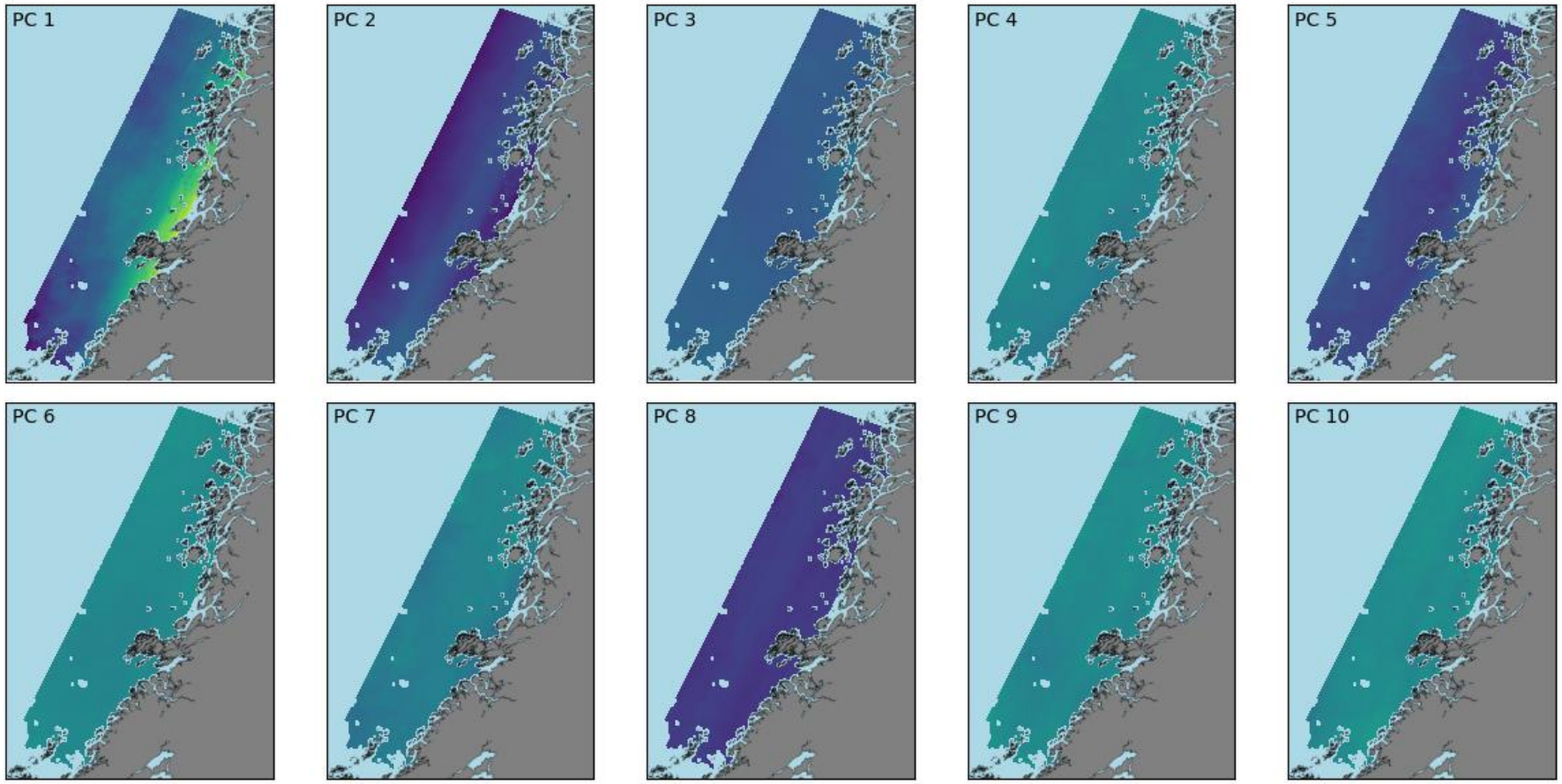


Example – Uncertainty Variance

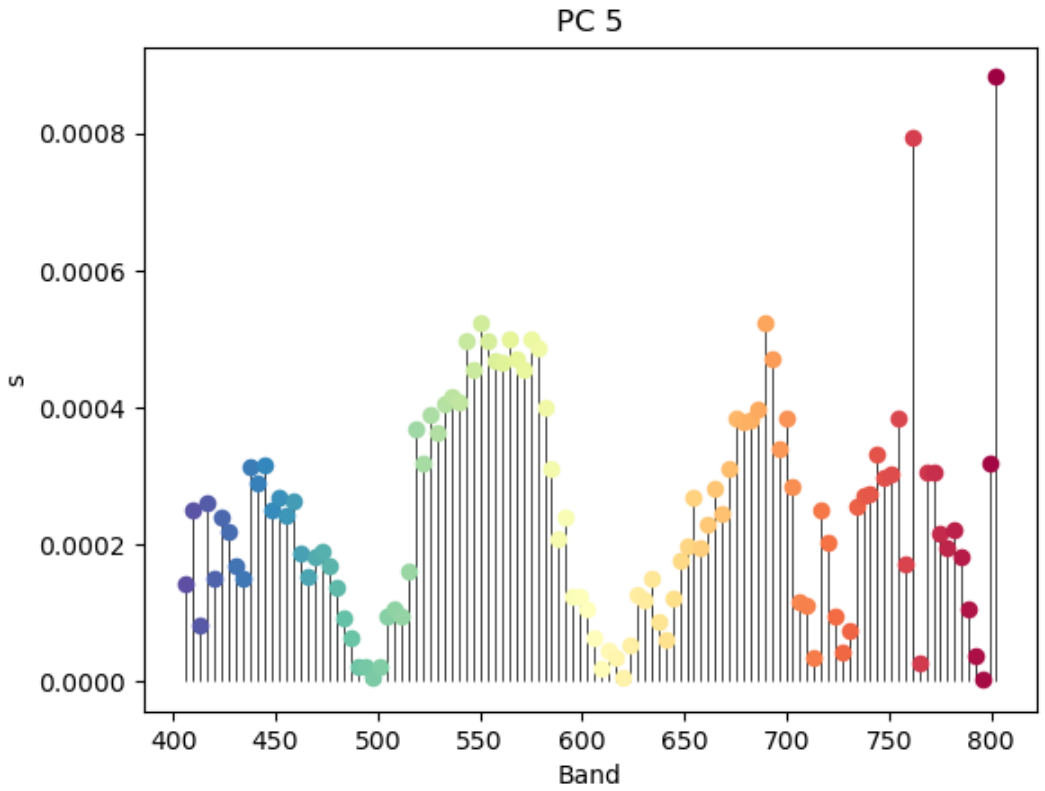
$$s(p_{ak}) = \sqrt{\left(\sum_{m=1}^{M} (p_{ak} - p_{ak(-m)})^2\right) \frac{M-1}{M}}$$

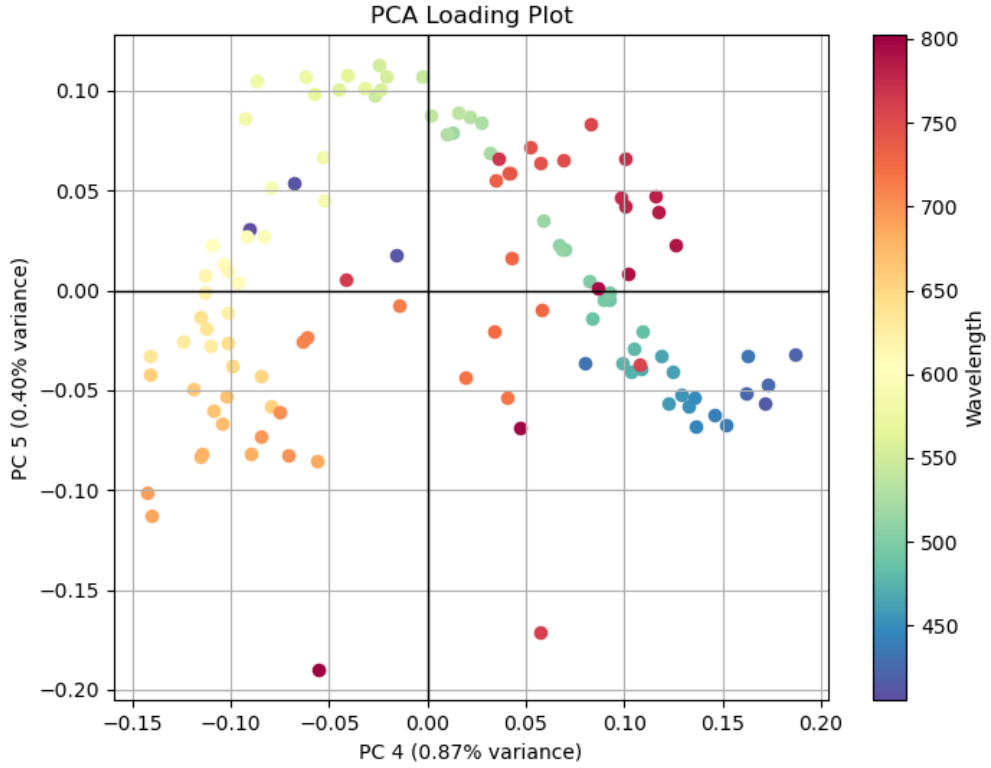
```
# Compute uncertainty values
s = np.zeros((n_components, n_variables))
for a in range(0, n_components):
    for v in range(0, n_variables):
        s[a,v] = np.sqrt(np.sum((p_ak[a,v] - p_akm[:,a,v])**2) * ((M-1)/M))
```

Example – Principal Components



Example – Uncertainty Variance





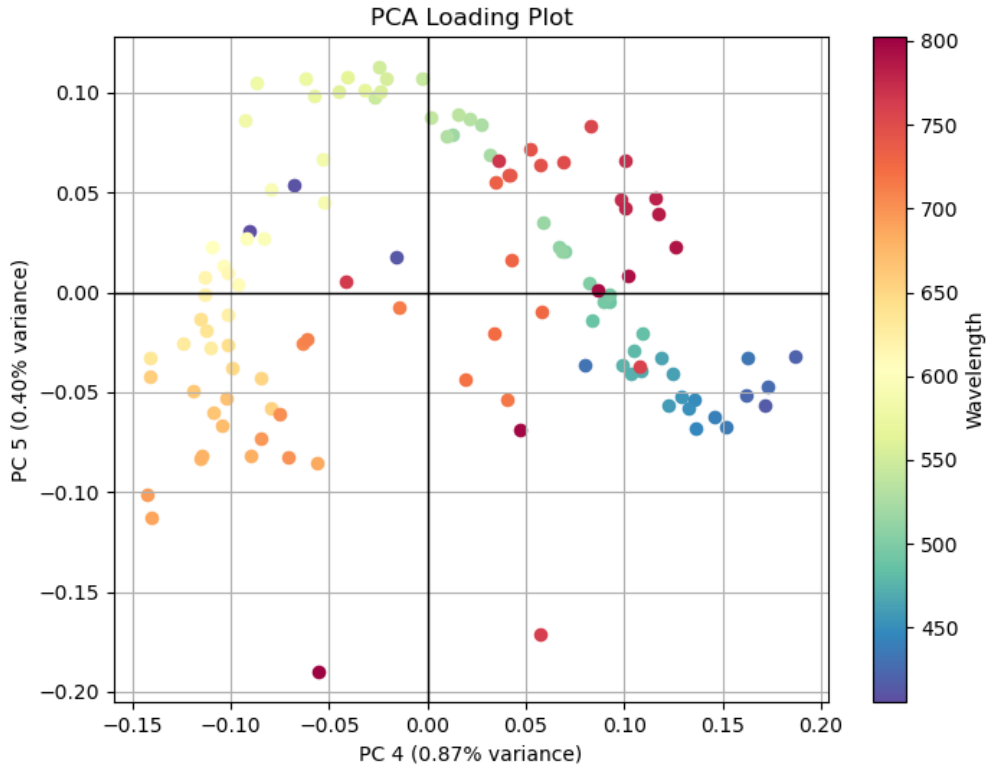
Example – Significance Test

- Modified t-test
- P_ak / s(p_ak) = 0

```
# t-test
t_values = p_ak / s

# Compute p-values based on t-distribution
df = M - 1  # Degrees of freedom
p_values = 2 * (1 - t.cdf(np.abs(t_values), df))

# Determine significance (e.g., p < 0.05)
significant = p_values < 0.05
```



Pros & Cons

Pros

- Reduces dimensionality
- Reduces number of dimensions in a training set
- De-noise data

Cons

- PCA is not robust against outliers
- Assumes linear relationship