Dimensionality reduction is a common task in data analysis

Dimensionality reduction is a common task in data analysis

If 3 variables all give the same information, why not just have 1?

Dimensionality reduction is a common task in data analysis

If 3 variables all give the same information, why not just have 1?

There are two related methods for reducing dimensionality:

Dimensionality reduction is a common task in data analysis

If 3 variables all give the same information, why not just have 1?

There are two related methods for reducing dimensionality:

- Principle Components Analysis (PCA)

Dimensionality reduction is a common task in data analysis

If 3 variables all give the same information, why not just have 1?

There are two related methods for reducing dimensionality:

- Principle Components Analysis (PCA)

- Factor Analysis

PCA is one way to reduce dimensionality. Let M be an $N \times J$ matrix of data

PCA is one way to reduce dimensionality. Let M be an $N \times J$ matrix of data

Decompose M as follows:

$$\underbrace{M}_{N\times J} = \underbrace{\boldsymbol{\theta}}_{N\times J} \times \underbrace{\Lambda}_{J\times J}$$

PCA is one way to reduce dimensionality. Let M be an $N \times J$ matrix of data

Decompose M as follows:

$$\underbrace{M}_{N \times J} = \underbrace{\boldsymbol{\theta}}_{N \times J} \times \underbrace{\Lambda}_{J \times J}$$

- Λ are stacked eigenvectors

PCA is one way to reduce dimensionality. Let M be an $N \times J$ matrix of data

Decompose M as follows:

$$\underbrace{M}_{N \times J} = \underbrace{\boldsymbol{\theta}}_{N \times J} \times \underbrace{\Lambda}_{J \times J}$$

- Λ are stacked eigenvectors
- $\boldsymbol{\theta}$ is an orthogonalized transformation of M (columns of $\boldsymbol{\theta}$ are independent)

PCA is one way to reduce dimensionality. Let M be an $N \times J$ matrix of data

Decompose M as follows:

$$\underbrace{M}_{N \times J} = \underbrace{\boldsymbol{\theta}}_{N \times J} \times \underbrace{\Lambda}_{J \times J}$$

- Λ are stacked eigenvectors
- $\boldsymbol{\theta}$ is an orthogonalized transformation of M (columns of $\boldsymbol{\theta}$ are independent)
- Λ indicates the rotation angle to get from $\boldsymbol{\theta}$ back to M

PCA is one way to reduce dimensionality. Let M be an $N \times J$ matrix of data

Decompose M as follows:

$$\underbrace{M}_{N\times J} = \underbrace{\boldsymbol{\theta}}_{N\times J} \times \underbrace{\Lambda}_{J\times J}$$

- Λ are stacked eigenvectors
- $\boldsymbol{\theta}$ is an orthogonalized transformation of M (columns of $\boldsymbol{\theta}$ are independent)
- Λ indicates the rotation angle to get from $\boldsymbol{\theta}$ back to M
- If M were orthogonal to begin with, $\Lambda = I$ and $M = \boldsymbol{\theta}$

Nothing on the previous slide helps us with dimensionality reduction per se

Nothing on the previous slide helps us with dimensionality reduction per se

- Reduce dimensionality by choosing the largest-magnitude eigenvectors

Nothing on the previous slide helps us with dimensionality reduction per se

- Reduce dimensionality by choosing the largest-magnitude eigenvectors
- These represent the dimensions $\boldsymbol{\theta}$ with the greatest variance

Nothing on the previous slide helps us with dimensionality reduction per se

- Reduce dimensionality by choosing the largest-magnitude eigenvectors
- These represent the dimensions $\boldsymbol{\theta}$ with the greatest variance
- We say that we "select the first K principal components of M"

Mathematically, we “reduce” (i.e. “approximate”) M by choosing a subset of $\boldsymbol{\theta}$ and Λ

Mathematically, we "reduce" (i.e. "approximate") M by choosing a subset of $\boldsymbol{\theta}$ and Λ

$$\underbrace{\widetilde{M}}_{N \times J} = \underbrace{\boldsymbol{\theta}_k}_{N \times K} \times \underbrace{\Lambda_k^{'}}_{K \times J}$$

Mathematically, we “reduce” (i.e. “approximate”) M by choosing a subset of $\boldsymbol{\theta}$ and Λ

$$\underbrace{\widetilde{M}}_{N\times J} = \underbrace{\boldsymbol{\theta}_k}_{N\times K} \times \underbrace{\Lambda_k^{'}}_{K\times J}$$

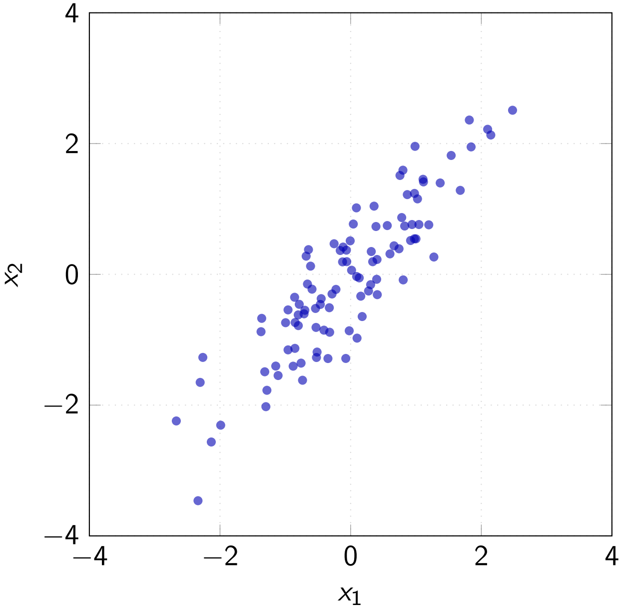
$$M = \boldsymbol{\theta}_k \Lambda_k^{'} + \boldsymbol{\varepsilon}$$

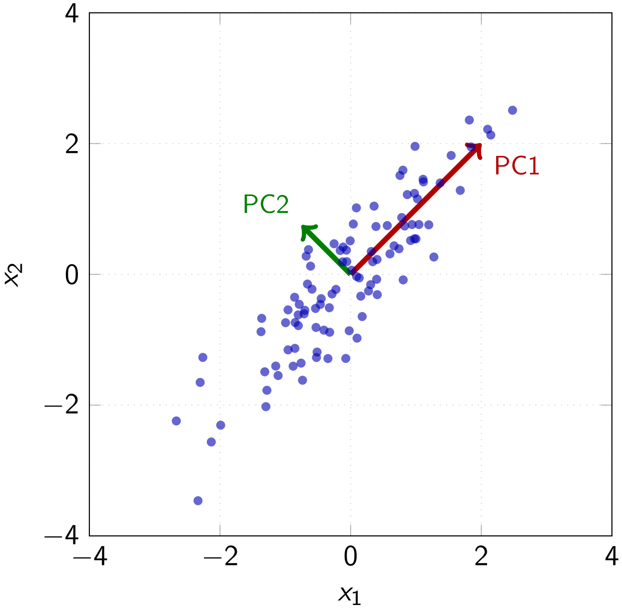
Mathematically, we "reduce" (i.e. "approximate") M by choosing a subset of $\boldsymbol{\theta}$ and Λ

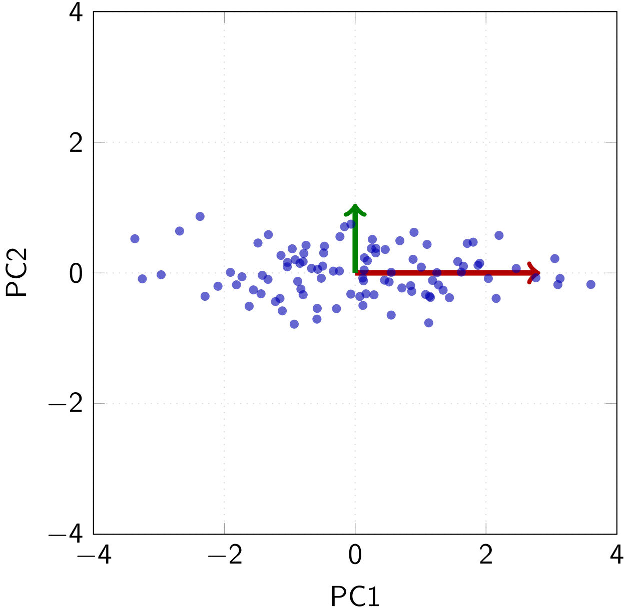
$$\underbrace{\widetilde{M}}_{N \times J} = \underbrace{\boldsymbol{\theta}_k}_{N \times K} \times \underbrace{\Lambda_k^{'}}_{K \times J}$$

$$M = \boldsymbol{\theta}_k \Lambda_k^{'} + \boldsymbol{\varepsilon}$$

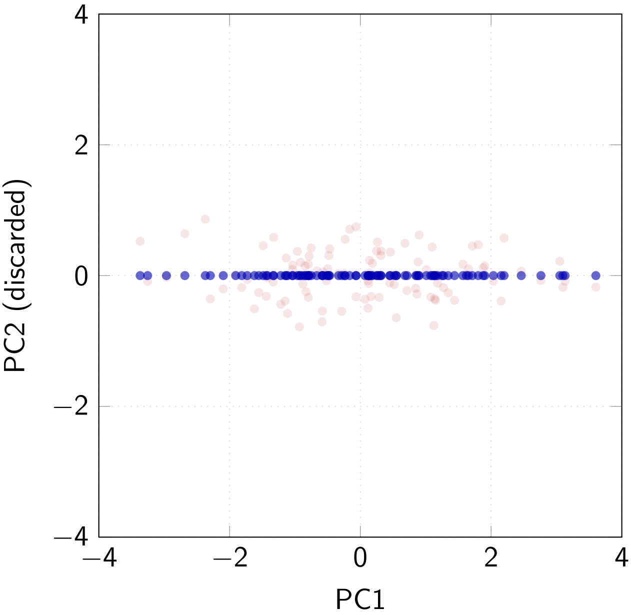
where $\boldsymbol{\varepsilon} \equiv M - \widetilde{M}$ is a $N \times J$ matrix


4
2
0
−2
−4
−4
−2
0
2
4
x_1
x_2


PC1
PC2
x_1
x_2
−4
−2
0
2
4



PC1
PC2
−4
−2
0
2
4
−4
−2
0
2
4



Use $PC1$ for all further analysis