

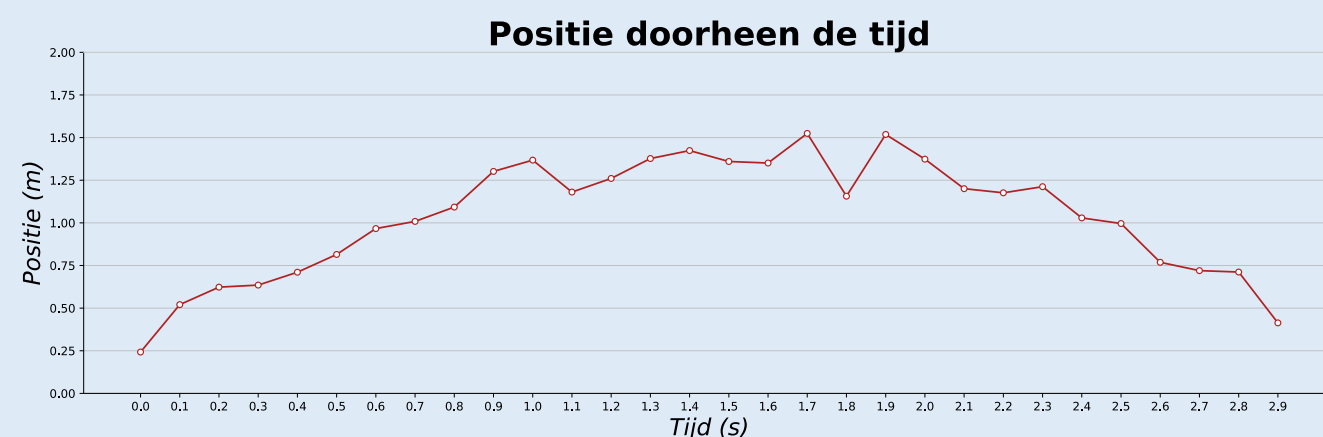
# Tensor Time Series Clustering

## De Dataset

### AMIE: Automatic Monitoring of Indoor Exercises

- Metingen van 3 fysieke oefeningen:
  - Squats
  - Forward lunges
  - Side lunges
- Sensoren op hand, knie, voet, etc.
- Positie van sensoren opdelen in X-, Y- en Z-as

- **Tijdsreeksen** als datapunten: Posities bij de uitvoering gemeten doorheen de tijd



Voorbeeld van een tijdsreeks

### Tijdsreeksen vergelijken

- **Dynamic Time Warping (DTW):** berekent afstand tussen 2 tijdsreeksen
  - Gelijkaardige data krijgen een kleine afstand
  - Kan niet-gesynchroniseerde tijdsreeksen vergelijken
  - $O(n^2)$  tijd
- **Afstandstensor:** elk element is een DTW afstand tussen 2 datapunten  
→ Alle elementen berekenen kostelijk!

## Benadering van een Tensor

### Candecomp/Parafac (CP) decompositie

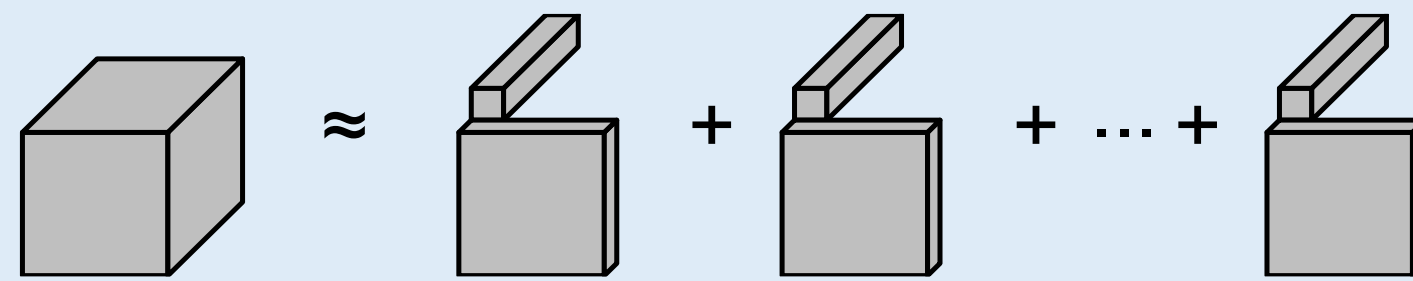
- Som van uitwendige producten van vectoren
- Alternating Least Squares algoritme
- Heeft de volledige tensor nodig → zeer kostelijk

### Adaptive Cross Approximation for Tensors (ACA-T) decompositie

- Som van uitwendige producten van matrices/vectoren
- **Adaptief** = Gericht vectoren in Tensor kiezen
- Heeft maar een deel van de tensor nodig → zeer efficiënt

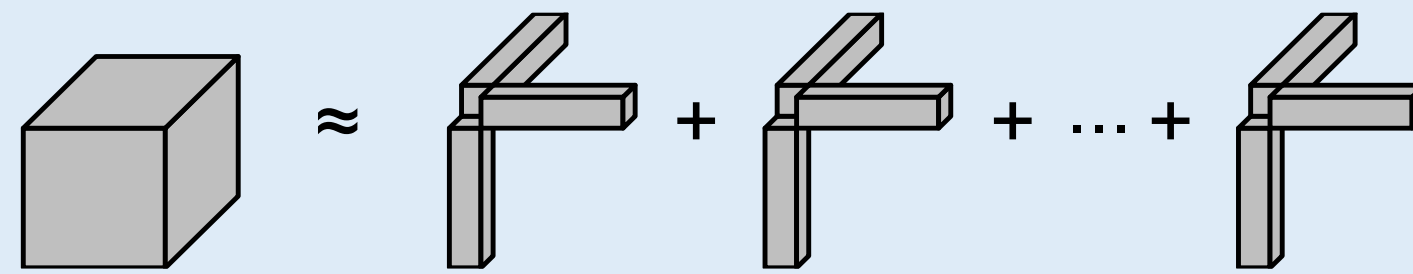
#### Matrix Methode (Matrix ACA-T):

- Kostelijk

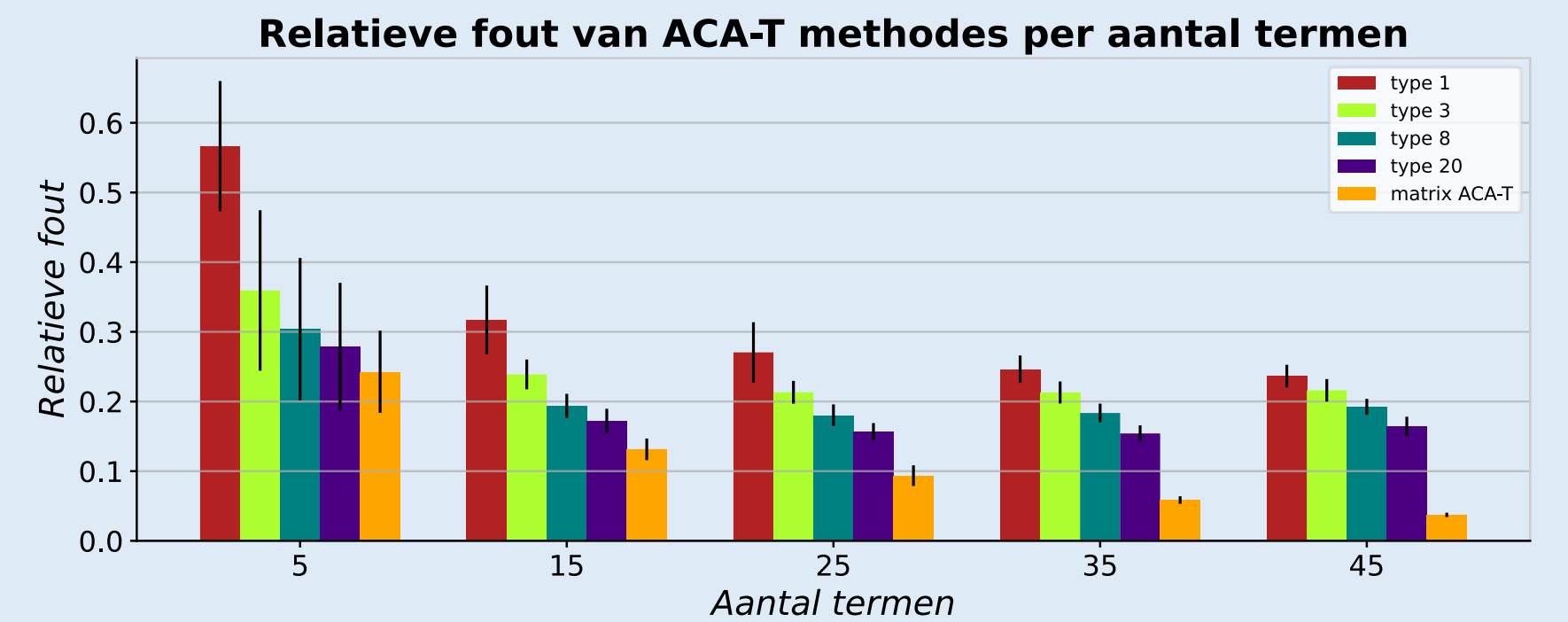


#### Vectoren Methode (Vector ACA-T type 1):

- Hogere relatieve fout bij eenzelfde aantal termen

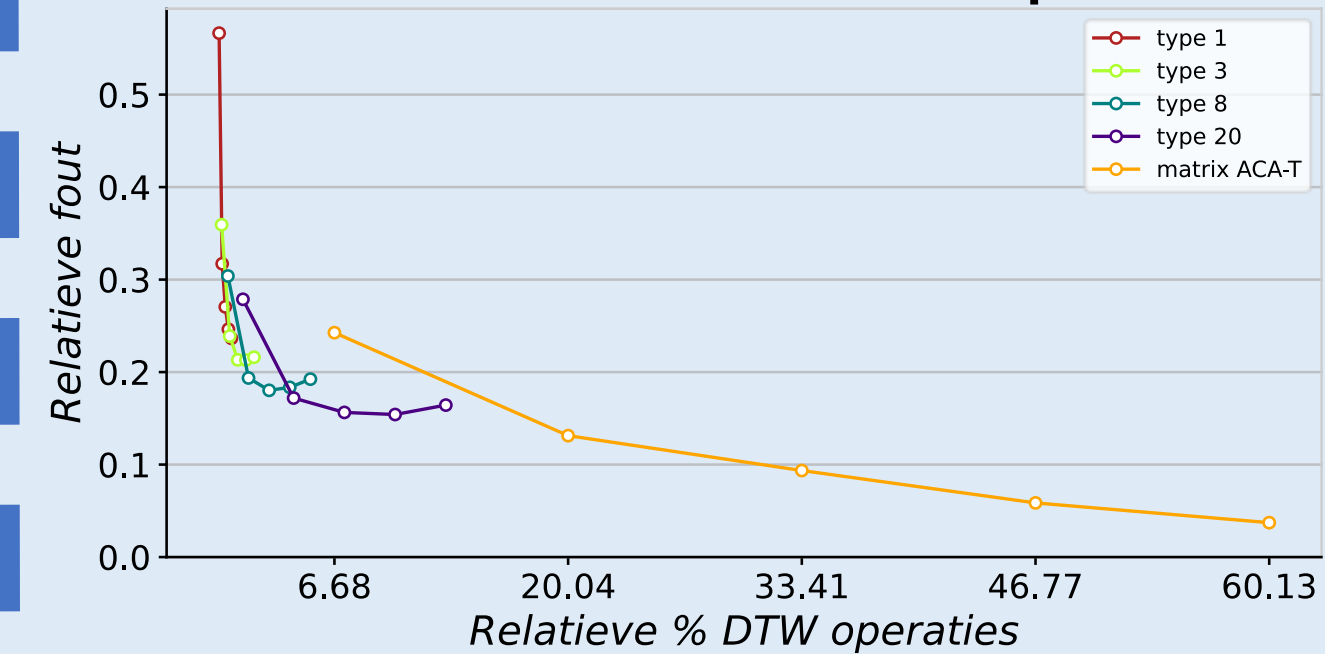


## Resultaten: Tensor Benadering



Bovenstaande figuur toont de relatieve fout van enkele types van de uitgebreide vectoren methode voor verschillende decompositie groottes. De balk is het gemiddelde en de zwarte streep de standaard afwijking (n = 50).  
**Opmerkelijk:** Hogere types liggen dichterbij de matrix methode bij lage rang, en dichterbij de vectoren methode (type 1) bij hoge rang.

#### Relatieve fout van ACA-T methodes versus hun relatieve % DTW operaties



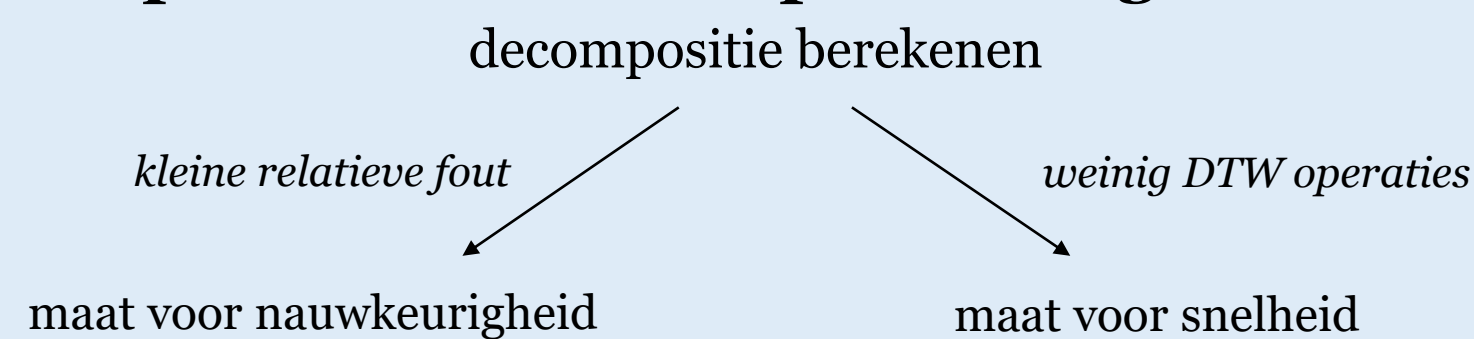
- De y-as bevat opnieuw het gemiddelde van de relatieve fout zoals in bovenstaande figuur (zonder standaard afwijking).
- Op de x-as staat het percentage DTW operaties nodig om deze relatieve fout te verkrijgen. 100% staat voor de volledige tensor berekenen.
- We willen een methode in de linker onderhoek wat een goede balans tussen rekentijd en nauwkeurigheid aangeeft.

## Doel van het Onderzoek

### Tijdsreeksen clusteren

- 1) Tensor decompositie berekenen
  - *CP decompositie: benadert tensor zeer goed*
  - *Adaptieve methodes: steunen op het ACA algoritme*
- 2) Rijen/kolommen/tubes uit decompositie als feature vectoren kiezen
  - *Dimensie bepaalt wat we clusteren*
  - *Bij AMIE-dataset: clusteren op sensoren of personen*
- 3) K-means clustering algoritme gebruiken met deze feature vectoren

### DTW operaties van decompositie laag houden



### Adaptive Cross Approximation (ACA) uitbreiden

- Decompositie algoritme voor matrices
- **Adaptief:** Focust op grootste en dus belangrijkste elementen
- **Cross:** Iteratief een rij en kolom aan de decompositie toevoegen
- Kleinere elementen niet nodig  
→ *Ideaal voor een decompositie van de afstandstensor*  
→ *ACA methodes voor tensoren ontwikkelen*

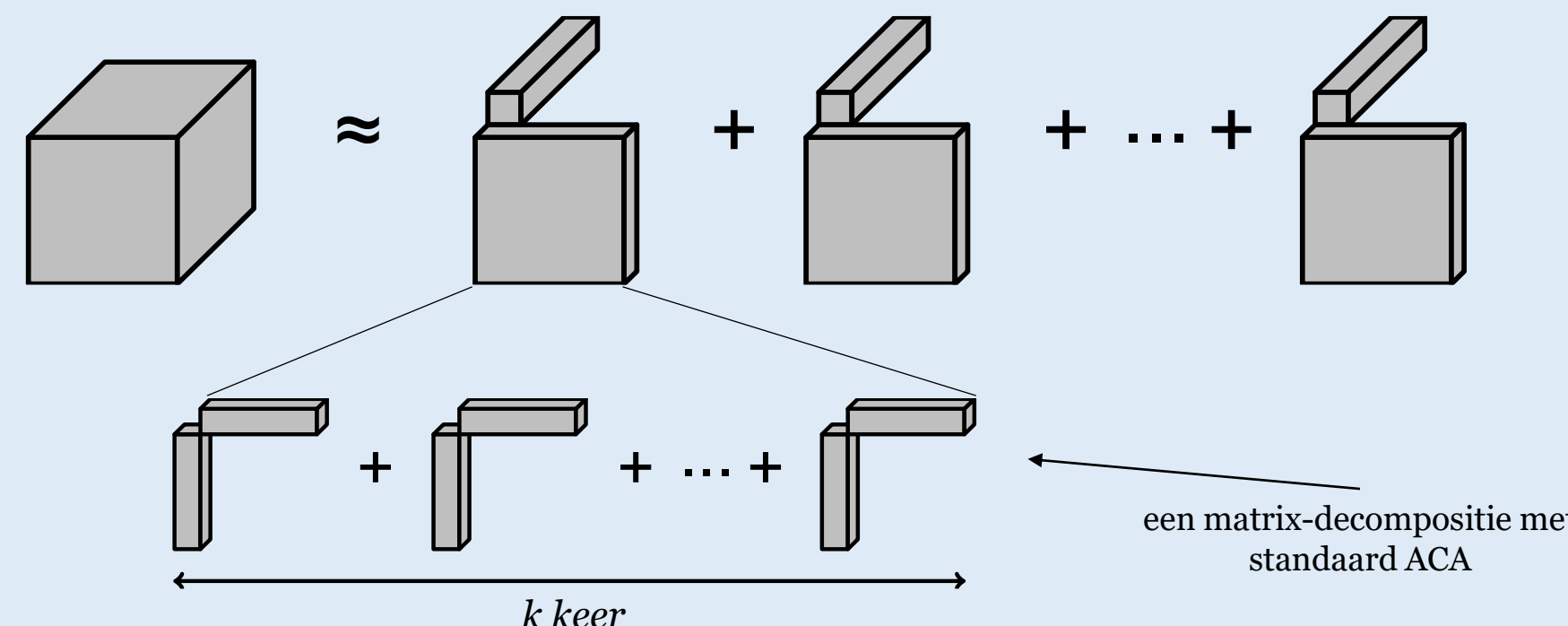
## Ons Onderzoek

### Hypothese: De uitgebreide vectoren methode zal voor een kleinere relatieve fout zorgen maar het aantal DTW operaties laag houden.

#### De uitgebreide vectoren methode

- Inspiratie halen bij matrix ACA-T:
- Matrix-slices volledig berekenen te duur  
→ **Het idee:** elke matrix-slice benaderen met ACA

- Notatie: **type k** = er zijn k termen in elke matrix decompositie
- Elke term bestaat uit:
  - **Tube:** een vector mode-3
  - **Matrix-decompositie:** som van termen met 2 vectoren mode-1 en 2



## Resultaten: Clusteren

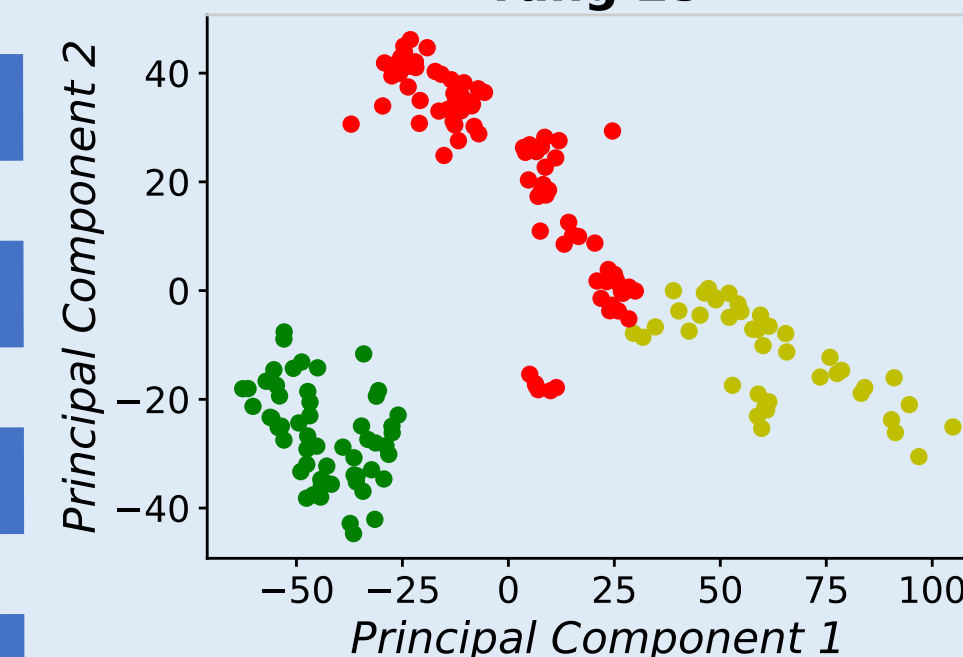
(verdere verwerking van resultaten nodig)

ID	Person	Exercise	Cluster
1	Person8	squat	1
2	Person8	squat	1
3	Person8	squat	1
4	Person8	squat	1
5	Person8	squat	1
6	Person8	squat	1
7	Person8	lunge	0
8	Person8	lunge	0
9	Person8	lunge	0
10	Person8	lunge	0
...	...	...	...
186	Person5	sidelunge	2

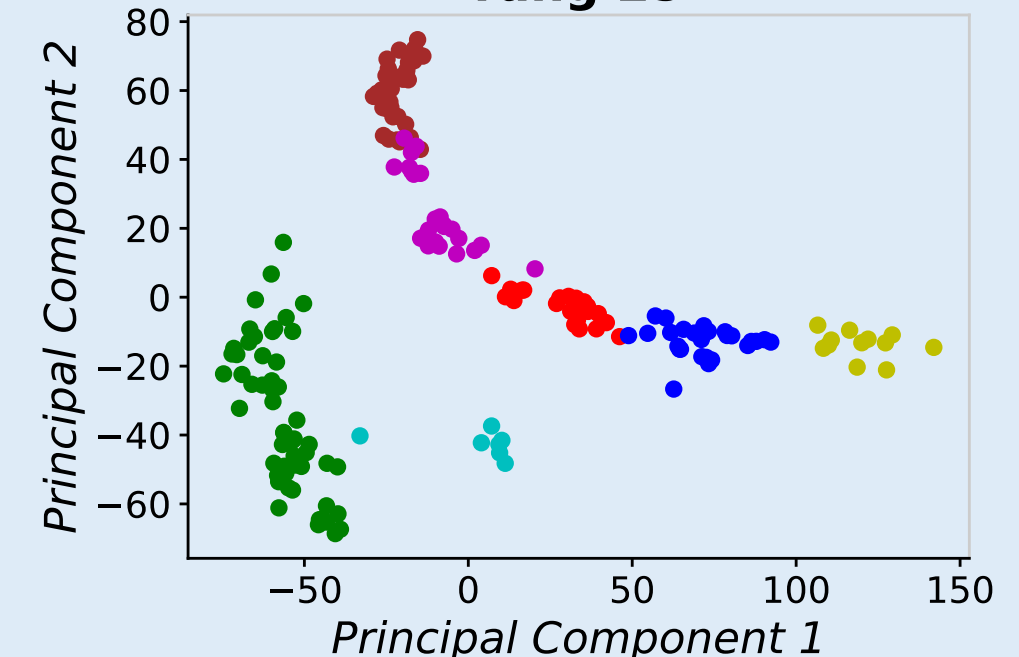
ID	Person	Exercise	Cluster
1	Person8	squat	2
2	Person8	squat	2
3	Person8	squat	4
4	Person8	squat	4
5	Person8	squat	4
6	Person8	squat	4
7	Person8	lunge	0
8	Person8	lunge	0
9	Person8	lunge	0
10	Person8	lunge	0
...	...	...	...
186	Person5	sidelunge	1

Tabellen met clustering via het K-means clustering algoritme weergegeven met 3 clusters links en 7 clusters rechts. De decompositie werd voor beide berekend met de uitgebreide vectoren methode type 3 rang 25.

#### 3 Clusters van Vector ACA-T type 3 rang 25



#### 7 Clusters van Vector ACA-T type 3 rang 25



Visualisatie van de resulterende clusters door middel van *Principal Component Analysis*. De rijen van de decompositie werden als feature-vectoren gebruikt.