Data-Science 1

principal component analysis

Karel de Grote Hogeschool

17/05/2023

Inhoud

- inleiding
- PCA
- in Python

Inleiding

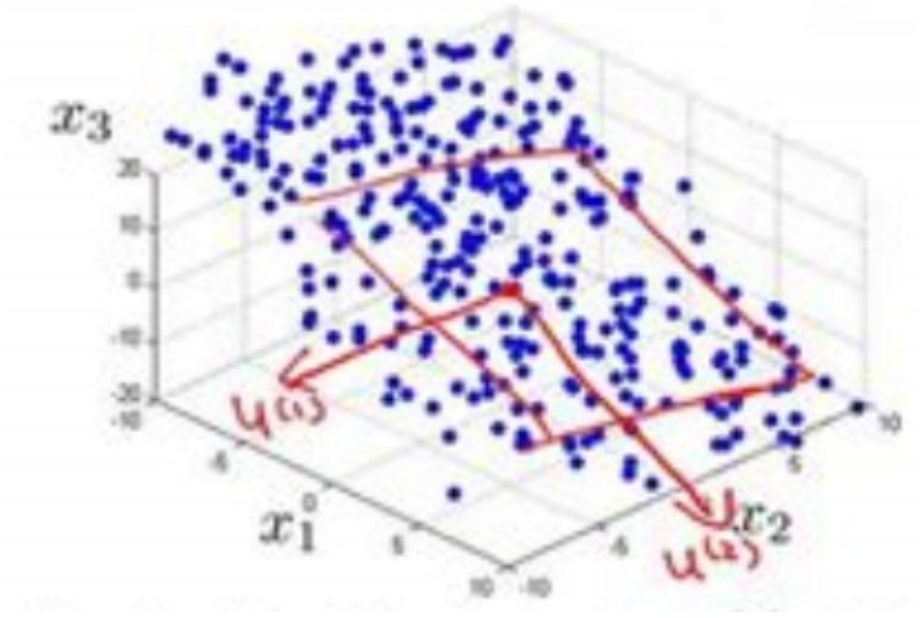
PCA voorbeeld 1

- Simpsons: 3 kolommen bepalen de simpsons
  - haarlengte, gewicht, leeftijd
- probleem: plot de simpsons in een scatterplot
  - slechts 2 dimensies
  - als je een dimensie laat vallen, verlies je ook informatie
- oplossing: zoek ‘camera’ positie zodat de simpsons zo ver mogelijk uit elkaar liggen

PCA voorbeeld 2

- stel dat je een dataset hebt met heel veel kolommen
  - MNIST databank: 70000 beeldjes met elk 784 (28x28) pixels
- probleem: algoritmes hebben enorm veel tijd nodig om hier analyses op uit te voeren
- oplossing:
  - probeer het aantal kolommen te verminderen
  - behoud daarbij zoveel mogelijk informatie

PCA: voorbeeld 3



PCA

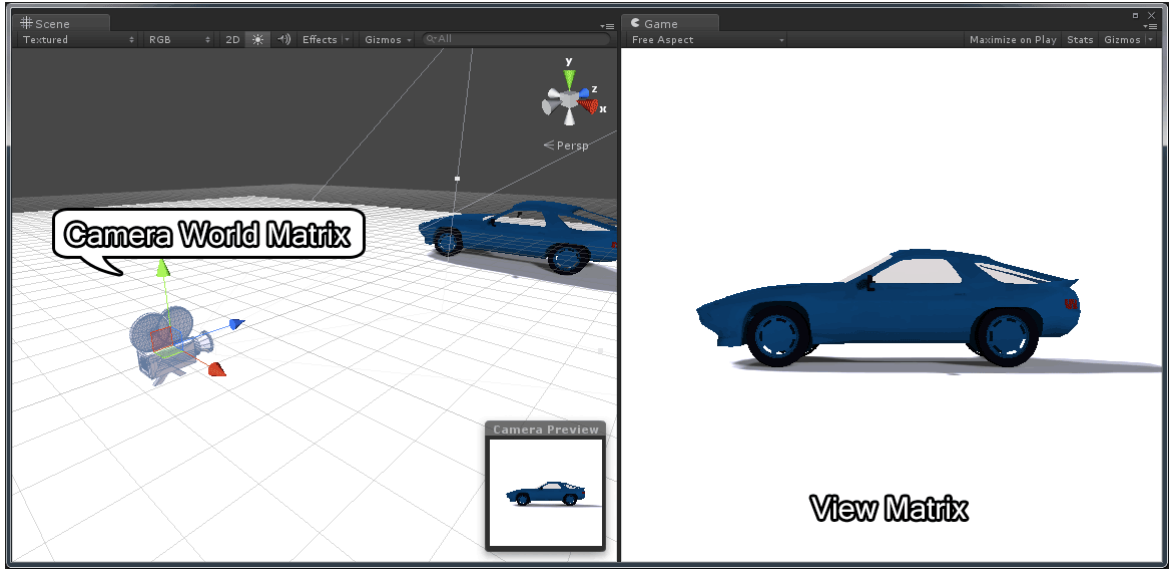
2

Vertrekpunt

- tabel met minstens 2 variabelen
- alle variabelen hebben ratio meetniveau
- er zijn correlaties tussen de variabelen (de correlaties zijn niet heel laag): hoe meer correlaties, hoe beter
- aangezien PCA dikwijls als pre-processing stap gebruikt wordt, moet de data ook voldoen aan de eisen van de volgende stappen

PCA in de praktijk

- we zoeken een nieuw assenstelsel zodat alle punten zo ver mogelijk uit elkaar liggen
- vergelijk: wereld-coördinaten en camera-coördinaten:



$$\begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} w_x \\ w_y \\ w_z \end{pmatrix}$$

$$c_x = a_{11} \cdot w_x + a_{12} \cdot w_y + a_{13} \cdot w_z$$
$$c_y = a_{21} \cdot w_x + a_{22} \cdot w_y + a_{23} \cdot w_z$$
$$c_z = a_{31} \cdot w_x + a_{32} \cdot w_y + a_{33} \cdot w_z$$

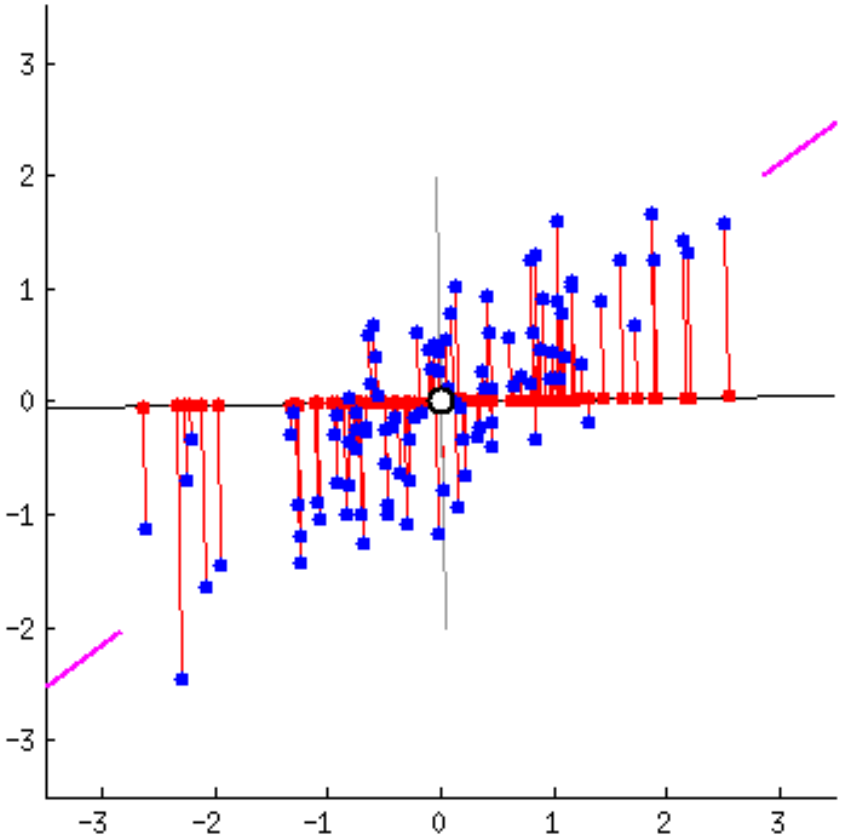
PCA

- de PCA zoekt dus een matrix die punten transformeert
- de punten zijn daarbij eerst verplaatst zodat het gemiddelde in de oorsprong ligt en meestal ook geschaald zodat alle standaardafwijkingen 1 zijn: Z-scores
- er gaat geen informatie verloren door deze transformatie

Principal components

- de principal components worden zodanig gekozen dat de eerste de meeste "variantie verklaren"
  - werkwijze: https://www.youtube.com/watch?v=FgakZw6K1QQ
- men laat in de praktijk de waarden van de laatste principal components vallen
  - nu gaat er wel informatie verloren, maar minimaal
  - we doen dus een reductie in het aantal dimensies
- voordelen
  - andere algoritmen kunnen nu sneller worden uitgevoerd
  - als er clusters zijn doordat punten dichter op elkaar liggen, dan zijn die nog steeds zichtbaar in de getransformeerde ruimte
  - visualisatie is mogelijk door bv de eerste 2 principal components te tonen

Principal components



In Python

Python

- zie Python code

Oefeningen

4

Oefeningen

- Protein consumption
- Goblets
- CPU’s