



Numerische Mathematik I | Hauptkomponentenanalyse

Robert Beinert | Technische Universität Berlin



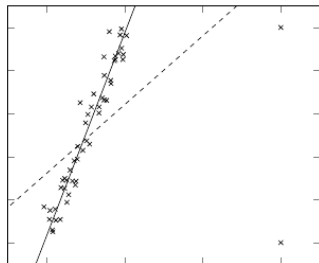
Hauptkomponentenanalyse

- Die Hauptkomponentenanalyse (Principle Component Analysis) ermöglicht die Dimensionsreduzierung von Daten.
- **Gegeben:** Punkte $\mathbf{x}_n \in \mathbb{R}^D$, $n = 0, \dots, N-1$.
- **Gesucht:** Affiner Teilraum

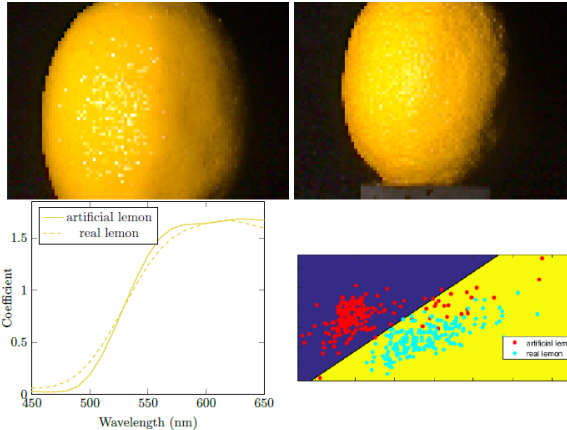
$$H_d := \{\mathbf{A}\mathbf{t} + \mathbf{b} : \mathbf{t} \in \mathbb{R}^d\}$$

mit $\mathbf{A} \in \mathbb{R}^{D \times d}$, $\mathbf{A}^\top \mathbf{A} = \mathbf{I}_d$ und $\mathbf{b} \in \mathbb{R}^D$, sodass

$$(\mathbf{A}, \mathbf{b}) := \operatorname{argmin}_{\mathbf{A}, \mathbf{b}} \sum_{n=0}^{N-1} \|\operatorname{proj}_{H_d}(\mathbf{x}_n) - \mathbf{x}_n\|_2^2.$$



Analyse von hyperspektralen Daten

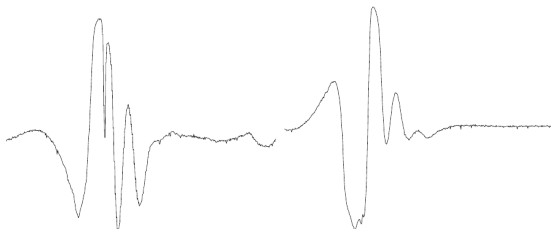




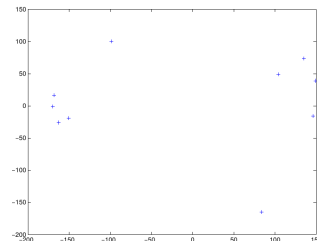
Analyse von Herzrhythmusstörungen

K1	K2	K3	K4	K5	K6	K7	K8	K9	K10
46.7638	109.6551	112.1864	100.0416	136.3707	144.3016	147.3712	161.7344	127.3121	87.9512
134.7171	82.7988	104.7244	87.5778	110.2042	91.4568	117.2545	106.7980	99.0934	82.5778
115.7308	72.8282	63.7577	55.7361	327.1906	269.7880	324.6902	341.0829	298.0034	63.1357
376.2790	481.3505	438.6366	413.6692	272.3134	373.6575	298.1814	347.9774	312.1234	399.3571
572.3792	502.2387	480.7142	472.6041	528.8898	481.6783	539.8783	525.4575	506.8756	480.1002
544.7938	514.4898	557.3757	552.4195	522.7821	397.3818	527.4377	462.0548	506.3419	527.3344
374.5493	396.9842	399.6761	409.8757	376.5966	430.2774	356.7877	371.4003	348.3973	429.9641
216.6788	232.8553	247.8097	282.3835	232.4479	336.6525	231.1340	251.3432	241.2378	227.3835
114.1481	123.9211	130.6727	178.4171	197.9071	264.5890	140.7754	185.1989	137.1735	189.5137

Parameter von 10 Patienten



Gesunder (links) und pathologischer (rechts) Herzschlag



Reduktion auf zwei Dimensionen



Projektion auf affinen Teilraum

- Die Projektion auf $H_d := \{\mathbf{A}\mathbf{t} + \mathbf{b} : \mathbf{t} \in \mathbb{R}^d\}$ ist definiert durch

$$\text{proj}_{H_d}(\mathbf{x}) = \underset{\mathbf{y} \in H_d}{\text{argmin}} \|\mathbf{y} - \mathbf{x}\|_2^2.$$

- Zur Bestimmung des Minimierers setzen wir $\mathbf{y} = \mathbf{A}\mathbf{t} + \mathbf{b}$ ein und erhalten

$$\mathbf{t} = \underset{\mathbf{t} \in \mathbb{R}^d}{\text{argmin}} \|\mathbf{A}\mathbf{t} + \mathbf{b} - \mathbf{x}\|_2^2.$$

- Ableiten nach \mathbf{t} und Nullsetzen liefert

$$\mathbf{A}^\top(\mathbf{A}\mathbf{t} + \mathbf{b} - \mathbf{x}) = \mathbf{0} \quad \text{oder} \quad \mathbf{t} = \mathbf{A}^\top(\mathbf{x} - \mathbf{b}).$$

- Die Projektion ist also

$$\text{proj}_{H_d}(\mathbf{x}) = \mathbf{A}\mathbf{A}^\top(\mathbf{x} - \mathbf{b}) + \mathbf{b}.$$



Bestimmung des affinen Teilraumes

- Wir setzen die Projektion ein und erhalten

$$\begin{aligned}(\mathbf{A}, \mathbf{b}) &:= \operatorname{argmin}_{\mathbf{A}, \mathbf{b}} \sum_{n=0}^{N-1} \|\operatorname{proj}_{H_d}(\mathbf{x}_n) - \mathbf{x}_n\|_2^2 \\&= \operatorname{argmin}_{\mathbf{A}, \mathbf{b}} \sum_{n=0}^{N-1} \|\mathbf{A}\mathbf{A}^\top(\mathbf{x}_n - \mathbf{b}) + \mathbf{b} - \mathbf{x}_n\|_2^2 \\&= \operatorname{argmin}_{\mathbf{A}, \mathbf{b}} \sum_{n=0}^{N-1} \|(\mathbf{I}_D - \mathbf{A}\mathbf{A}^\top)(\mathbf{x}_n - \mathbf{b})\|_2^2.\end{aligned}$$

- Ableiten nach \mathbf{b} und Nullsetzen liefert

$$\sum_{n=0}^{N-1} (\mathbf{I}_D - \mathbf{A}\mathbf{A}^\top)(\mathbf{I}_D - \mathbf{A}\mathbf{A}^\top)(\mathbf{b} - \mathbf{x}_n) = \sum_{n=0}^{N-1} (\mathbf{I}_D - \mathbf{A}\mathbf{A}^\top)(\mathbf{b} - \mathbf{x}_n) = \mathbf{0}.$$



Bestimmung des affinen Teilraumes

- Weiteres Umstellen liefert

$$N(I_D - \mathbf{A}\mathbf{A}^T)\mathbf{b} = (I_D - \mathbf{A}\mathbf{A}^T) \sum_{n=0}^{N-1} \mathbf{x}_n.$$

- Eine partikuläre Lösung ist offensichtlich der arithmetische Mittelwert

$$\mathbf{b} = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{x}_n.$$

- Der Kern von $(I_D - \mathbf{A}\mathbf{A}^T)$ ist gerade der Unterraum der von \mathbf{A} aufgespannt wird, also

$$\ker(I_D - \mathbf{A}\mathbf{A}^T) = \{\mathbf{A}\mathbf{t} : \mathbf{t} \in \mathbb{R}^d\}.$$

- Wir können uns für \mathbf{b} auf den Mittelwert beschränken.



Bestimmung des affinen Teilraumes

- Es bleibt die Minimierung nach \mathbf{A} :

$$\mathbf{A} = \underset{\mathbf{A}}{\operatorname{argmin}} \sum_{n=0}^{N-1} \|(\mathbf{I}_D - \mathbf{A}\mathbf{A}^T)(\mathbf{x}_n - \mathbf{b})\|_2^2.$$

- Mit $\mathbf{y}_n := \mathbf{x}_n - \mathbf{b}$ und $\mathbf{A} = [\mathbf{a}_0 | \dots | \mathbf{a}_{d-1}]$ können wir die Zielfunktion umstellen nach

$$\begin{aligned} \sum_{n=0}^{N-1} \|(\mathbf{I}_D - \mathbf{A}\mathbf{A}^T)\mathbf{y}_n\|_2^2 &= \sum_{n=0}^{N-1} \mathbf{y}_n^T \mathbf{y}_n - \sum_{n=0}^{N-1} \mathbf{y}_n^T \mathbf{A} \mathbf{A}^T \mathbf{y}_n = \sum_{n=0}^{N-1} \|\mathbf{y}_n\|_2^2 - \sum_{n=0}^{N-1} \mathbf{y}_n^T \left(\sum_{k=0}^{d-1} \mathbf{a}_k \mathbf{a}_k^T \right) \mathbf{y}_n \\ &= \sum_{n=0}^{N-1} \|\mathbf{y}_n\|_2^2 - \sum_{n=0}^{N-1} \sum_{k=0}^{d-1} \mathbf{y}_n^T \mathbf{a}_k \mathbf{a}_k^T \mathbf{y}_n = \sum_{n=0}^{N-1} \|\mathbf{y}_n\|_2^2 - \sum_{k=0}^{d-1} \mathbf{a}_k^T \left(\sum_{n=0}^{N-1} \mathbf{y}_n \mathbf{y}_n^T \right) \mathbf{a}_k. \end{aligned}$$



Bestimmung des affinen Teilraumes

- Die Lösung des Minimierungsproblems

$$\mathbf{A} = \underset{\mathbf{A}}{\operatorname{argmin}} \sum_{n=0}^{N-1} \|\mathbf{y}_n\|_2^2 - \sum_{k=0}^{d-1} \mathbf{a}_k^\top \underbrace{\left(\sum_{n=0}^{N-1} \mathbf{y}_n \mathbf{y}_n^\top \right)}_{=\mathbf{Y}\mathbf{Y}^\top} \mathbf{a}_k.$$

besteht spaltenweise aus den Eigenvektoren zu den d größten Eigenwerten von $\mathbf{S} = \mathbf{Y}\mathbf{Y}^\top$ mit $\mathbf{Y} := [\mathbf{y}_0 | \dots | \mathbf{y}_{N-1}]$.

- Es ist numerisch stabiler anstelle der Eigenwertzerlegung $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^*$ die Singulärwertzerlegung $\mathbf{Y} = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^*$ zu berechnen.
- Die Eigenwerte in der Diagonalmatrix $\mathbf{\Lambda}$ sind die quadrierten Singulärwerte in $\mathbf{\Sigma}$.
- Die Eigenvektoren sind die linken Singulärvektoren.



Dimensionsreduktion

- \mathbf{S} ist bis auf einen Faktor die empirische Kovarianz-Matrix

$$\left(\sum_{n=0}^{N-1} (x_{n,k} - b_k)(x_{n-\ell} - b_\ell) \right)_{k,\ell=0}^{D-1}.$$

- Die Hauptkomponenten sind also die d Richtungen mit der größten Varianz.
- Anstelle der Daten \mathbf{x}_n untersucht man anschließend die reduzierten Daten

$$\mathbf{A}^T(\mathbf{x}_n - \mathbf{b}) \in \mathbb{R}^d$$

(Koeffizienten bezüglich der Basis in \mathbf{A}).

Klassifizierung (K -Means-Algorithmus)

- Die reduzierten Daten ermöglichen eine einfachere Klassifizierung.
- **Gegeben:** Punkte $\mathbf{x}_n \in \mathbb{R}^d$, $n = 0, \dots, N - 1$.
- **Gesucht:** Klassenmittelpunkte/Prototypen $\mathbf{r}_k \in \mathbb{R}^d$, $k = 0, \dots, K - 1$, sodass

$$(\mathbf{r}_1, \dots, \mathbf{r}_{K-1}) = \underset{(\mathbf{r}_1, \dots, \mathbf{r}_{K-1})}{\operatorname{argmin}} \sum_{n=0}^{N-1} \min_{0 \leq k \leq K-1} \|\mathbf{r}_k - \mathbf{x}_n\|_2^2.$$

- Anschließend identifizieren wir den Punkt \mathbf{x} mit der Klasse

$$k = \underset{k \in \{0, \dots, K-1\}}{\operatorname{argmin}} \|\mathbf{r}_k - \mathbf{x}\|_2^2.$$



Klassifizierung (K -Means-Algorithmus)

- Minimierungsproblem:

$$(\mathbf{r}_1, \dots, \mathbf{r}_{K-1}) = \underset{(\mathbf{r}_1, \dots, \mathbf{r}_{K-1})}{\operatorname{argmin}} \sum_{n=0}^{N-1} \min_{0 \leq k \leq K-1} \|\mathbf{r}_k - \mathbf{x}_n\|_2^2.$$

- K -Means-Algorithmus:

1. **Initialisierung:** Klassenmittelpunkte $\mathbf{r}_1, \dots, \mathbf{r}_{K-1}$.

2. **Wiederhole:**

- 2.1 Für $n = 0, \dots, N - 1$, ordne \mathbf{x}_n der Klasse C_k zu mit

$$k = \underset{k \in \{0, \dots, K-1\}}{\operatorname{argmin}} \|\mathbf{r}_k - \mathbf{x}_n\|_2^2.$$

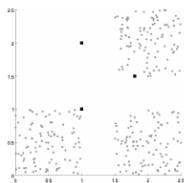
- 2.2 Für $k = 0, \dots, K - 1$, bestimme neue Prototypen durch

$$\mathbf{r}_k = \frac{1}{|C_k|} \sum_{\mathbf{x}_n \in C_k} \mathbf{x}_n.$$

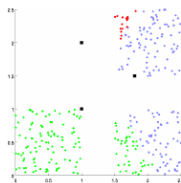
Entferne hierbei leere Klassen und verringere K entsprechend.



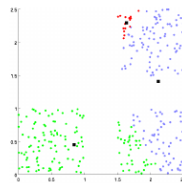
Klassifizierung (K -Means-Algorithmus)



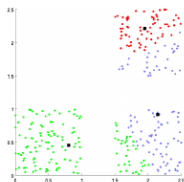
(a) Initialisation.



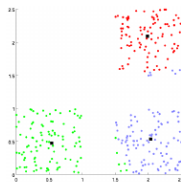
(b) Step 1



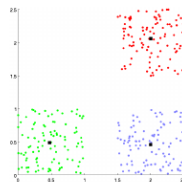
(c) Step 2



(d) 2. Iteration



(e) 3. Iteration



(f) 4. Iteration



Umsetzung in PYTHON

- Wir werden den MNIST-Datensatz untersuchen, welcher aus Bildern von handschriftlichen Ziffern besteht.

- Die Bilder können mit den Funktionen

- `imgs = np.fromfile('train-images.idx3-ubyte', dtype=np.uint8)`
- `imgs = np.reshape(imgs[16:], [-1, 28, 28])`

geladen werden. Die zugehörigen Labels mit

- `labs = np.fromfile('train-labels.idx3-ubyte', dtype=np.uint8)`
- `labs = labs[8:]`

- Die Singulärwertzerlegung kann bestimmt werden mit

`numpy.linalg.svd`