

Machine Learning and Data Mining
Summer 2015
Exercise Sheet 10

Presentation of Solutions to the Exercise Sheet on the 08.07.2015

Exercise 10-1 PCA

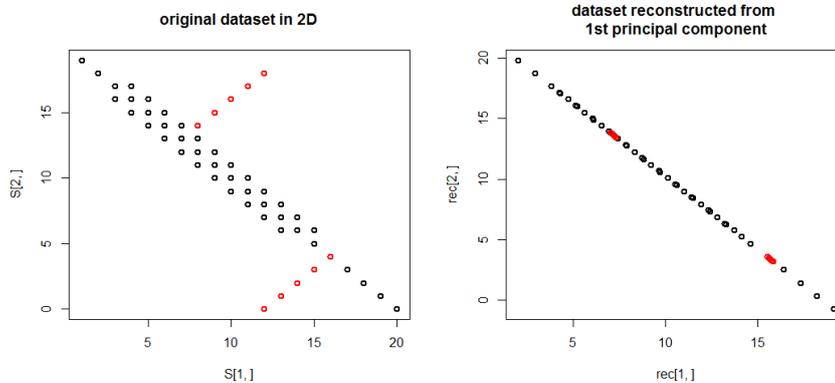
- a) Please describe what a PCA aims for and under what circumstances it is most helpful.
- b) Which possibly negativ consequences might arise when applying PCA to a dataset of unknown structure?

Possible Solution:

a) direkt aus Skript zu beantworten:

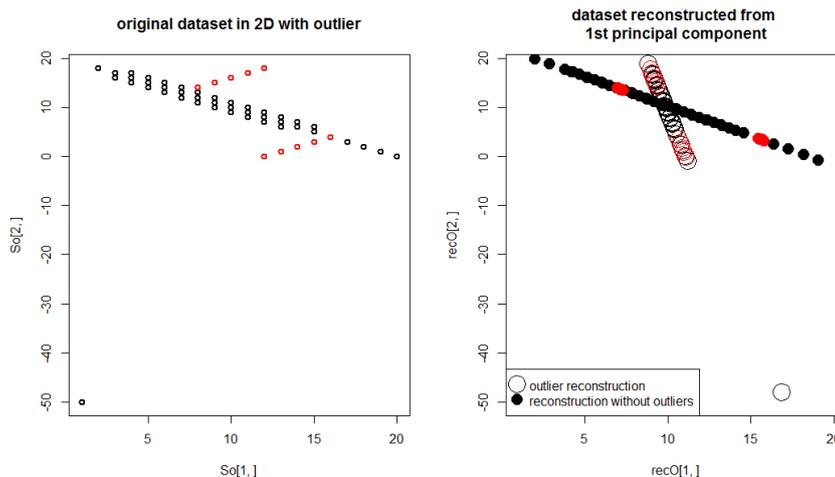
- In der Hauptkomponentenanalyse versucht man, versteckte Variablen zu schätzen (einfaches Bsp: *Umsatz* wird geschätzt durch die Variablen *Einnahmen* und *Ausgaben*).
- Dimensionsreduktion: korrelierte Daten auf neue Zufallsvariable abbilden $\mathbf{x} \in \mathbb{R}^M \mapsto \mathbf{U}^t \mathbf{x} \in \mathbb{R}^r$; Ziel ist die verringerte Komplexität im Vergleich zu Berechnungen im Originalraum – andere Anwendungen (LDA, RCA, ...) haben darüber hinaus die Minimierung von Rauscheinflüssen oder die Modellierung mehrerer zugrundeliegenden Verteilungen zum Ziel.
- Man nimmt an, dass sich die beobachteten Variablen als Linearkombinationen der versteckten Variablen schreiben lassen: $\mathbf{x} = \mu + \mathbf{U}\mathbf{w} + \epsilon$. Sollte dies klar nicht der Fall sein, so ist die PCA ein verfehelter Ansatz.

b) skalierte Daten können das Ergebnis der PCA verzerren. \Rightarrow daher skalieren/normieren
 Relevante Strukturen können verloren gehen, wie die 2 roten Linien im Bild



Lösung: subspace clustering / correlation clustering

Ein weiteres Problem entsteht durch Outlier. Diese können eine PCA-Transformation stark verzerren.



Exercise 10-2 PCA

Consider the $\mathbf{X} \in \mathbb{R}^{M \times N}$ matrix containing six data points $\mathbf{x}_i \in \mathbb{R}^2$. Note that in contrast to the conventional

representation, the patterns are held in columns here.

dim 1	1	2	3	5	6	7
dim 2	0	0	0	6	6	6

Conduct a PCA on the given data. Please state the eigenvectors, eigenvalues, covariance matrix and visualize the data before and after the PCA.

Possible Solution:

Die Matrix entspricht Punktpaaren, wobei die Spalten jeweils die x-/y-Koordinate eines Punktes bilden.

- Mittelwertbereinigung / zentrieren:
Zeilenweise Mittelwert bilden und von den Daten subtrahieren:

$$\hat{\mu} = \frac{1}{N} \sum_i \mathbf{x}_i = \begin{pmatrix} 4 \\ 3 \end{pmatrix}$$
$$\tilde{\mathbf{X}} = (\mathbf{x}_i - \hat{\mu})_i = \begin{array}{|c|c|c|c|c|c|} \hline -3 & -2 & -1 & 1 & 2 & 3 \\ \hline -3 & -3 & -3 & 3 & 3 & 3 \\ \hline \end{array}$$

Dies ist nötig um zu verhindern, dass die erste Hauptkomponente lediglich eine Art Mittelwertschätzer wird.

- Kovarianzmatrix $E[(X - E(X)) \cdot (X - E(X))^T]$ berechnen.

$$\text{cov}(X) \approx \hat{\Sigma} = \frac{1}{N} \sum_i \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^T = \frac{1}{N} \tilde{\mathbf{X}} \tilde{\mathbf{X}}^T = \begin{bmatrix} \frac{14}{3} & 6 \\ 6 & 9 \end{bmatrix} = \begin{bmatrix} 4,7 & 6 \\ 6 & 9 \end{bmatrix}$$

- Eigenwerte und Eigenvektoren berechnen: Gesucht ist $\hat{\Sigma} = \hat{\mathbf{U}} \hat{\mathbf{D}} \hat{\mathbf{U}}^T$ mit absteigend sortierten Eigenwerten $\hat{\lambda}_j$ in $\hat{\mathbf{D}}$.

$$\begin{aligned} \det(\hat{\Sigma} - \lambda I) &= \det \begin{bmatrix} \frac{14}{3} - \lambda & 6 \\ 6 & 9 - \lambda \end{bmatrix} = \left(\frac{14}{3} - \lambda\right) \cdot (9 - \lambda) - 36 = \\ &= 14 \cdot 3 - 36 - \frac{14 + 27}{3} \lambda + \lambda^2 = \lambda^2 - \frac{41}{3} \lambda + 6 \stackrel{!}{=} 0 \\ \lambda_{1,2} &= \frac{\frac{41}{3} \pm \sqrt{\left(\frac{41}{3}\right)^2 - 4 \cdot 6}}{2} = 13,21 \text{ und } 0,45 \end{aligned}$$

Die Eigenvektoren können nun durch Einsetzen der Eigenwerte bestimmt werden:

Solving the first equation with respect to y results in $y = 1.42x$. Plugging this into the second equation gives a trivial result. Hence the linear equation system has infinitely many solutions, namely all different lengths of the solution vector. To get a unique solution, we choose the one with unit length.

$$\begin{bmatrix} \frac{14}{3} & 6 \\ 6 & 9 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \lambda_{1,2} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$
$$\stackrel{\lambda_1}{\Rightarrow} \begin{array}{l} \frac{14}{3}x + 6y = \lambda_1 x \\ 6x + 9y = \lambda_1 y \end{array} \Rightarrow \text{erster Eigenvektor: } \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0,57 \\ 0,82 \end{bmatrix}$$

etc. ... Sollte etwas derartiges in der Klausur drankommen, dann nur mit schön aufgehenden Zahlen!

$$\text{Eigenwerte: } \text{diag}(\hat{\mathbf{D}}) = \begin{bmatrix} 13 \\ 0,45 \end{bmatrix}$$

$$\text{Eigenvektoren: } \hat{\mathbf{U}} = \begin{bmatrix} 0,57 & 0,82 \\ 0,82 & -0,57 \end{bmatrix}$$

Possible Solution:

- Nun können wir den zweiten Eigenvektor entfernen und eine Transformationsmatrix \mathbf{U} bilden (i.d. Literatur oft \mathbf{A} oder Φ):

$$\mathbf{U} = \begin{bmatrix} 0,57 & 0 \\ 0,82 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0,57 \\ 0,82 \end{bmatrix}$$

- Daten transformieren: $\mathbf{Y} = \mathbf{A} \cdot \tilde{\mathbf{X}}$ wobei \mathbf{A} die Transformationsmatrix \mathbf{U}^T ist: $\mathbf{X} \mapsto \mathbf{U}^T \cdot \tilde{\mathbf{X}}$. Bei uns erhalten wir hier eine neue Daten-“matrix”:

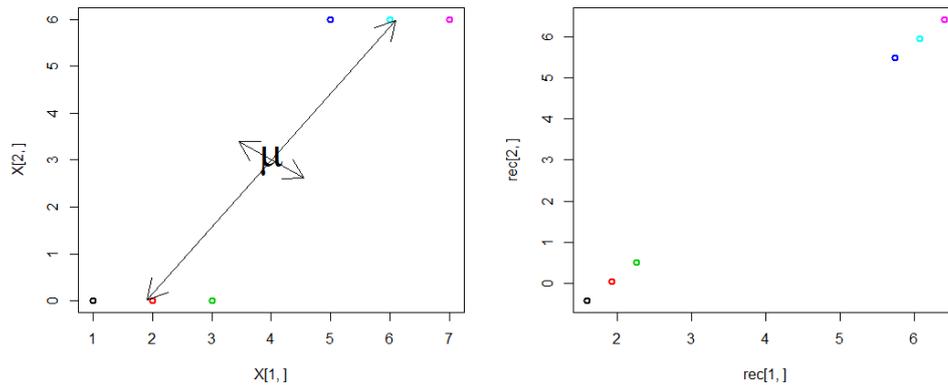
$$\mathbf{Y} = \begin{bmatrix} -4,18 & -3,60 & -3,03 & 3,03 & 3,60 & 4,18 \end{bmatrix}$$

Wir können nun versuchen, die Originalmatrix als $\hat{\mathbf{Z}} = \mu \cdot \mathbf{1} + \mathbf{U} \cdot \mathbf{Y} = \mu \cdot \mathbf{1} + \mathbf{U} \cdot \mathbf{U}^T \cdot \tilde{\mathbf{X}}$ zu rekonstruieren:

$$\hat{\mathbf{Z}} = \begin{bmatrix} 1,60 & 1,93 & 2,26 & 5,74 & 6,07 & 6,40 \\ -0,42 & 0,05 & 0,52 & 5,48 & 5,95 & 6,42 \end{bmatrix}$$

Dadurch, dass wir eine Hauptkomponente verworfen haben, ist die Rekonstruktion natürlich nicht vollkommen.

- Visualisierung:



Exercise 10-3 Eigenfaces

The term *eigenfaces* describes the eigenvectors of a normalized covariance matrix.

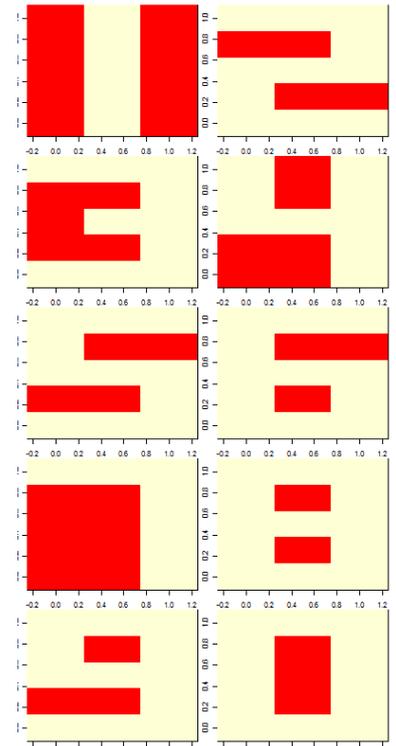
- Find the eigenfaces of the number dataset from the `numberMatrix.RTable`.
- How many principal components are required to reconstruct the dataset? Are the eigenfaces sufficient for this purpose? Are all patterns reconstructable with equal quality?

Possible Solution:

Eigenfaces sind klar: für $\hat{\Sigma} = \hat{\mathbf{U}}\hat{\mathbf{D}}\hat{\mathbf{U}}^T$ sind die Eigenfaces die Eigenvektoren in den Spalten von $\hat{\mathbf{U}}$.

a) Eigenwerte: $\text{diag}(\hat{\mathbf{D}}) =$

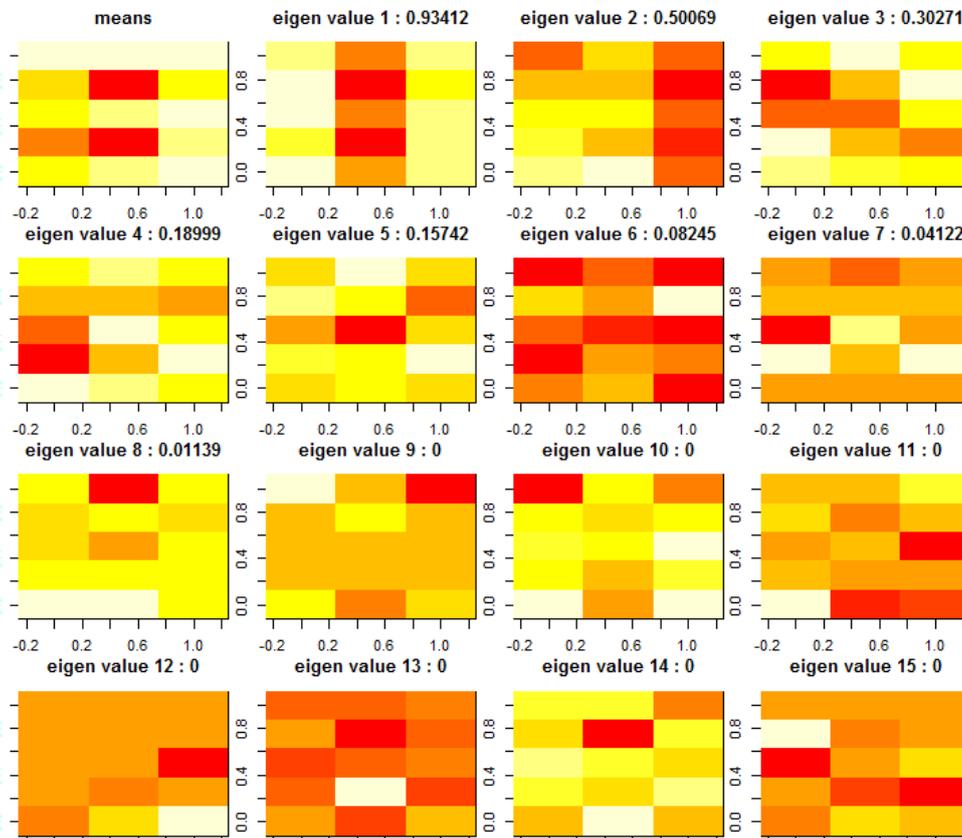
$$\begin{bmatrix} 0.9341151 \\ 0.500686 \\ 0.3027136 \\ 0.1899948 \\ 0.1574221 \\ 0.08244985 \\ 0.04122402 \\ 0.01139459 \\ 3.315557e-17 \\ 1.111085e-17 \\ 1.488616e-18 \\ -1.770126e-18 \\ -4.033219e-18 \\ -3.158965e-17 \\ -3.86529e-17 \end{bmatrix}$$



Wir erhalten also eine Reihe relevanter Eigenwerte; dann, mit dem 8. Wert einen Bruch zu Eigenwerten = 0. Die Eigenvektoren sind entsprechend aussagearm in den hinteren Dimensionen.

Wir betrachten die Eigenfaces. Da wir später auch noch den Mittelwert-

vektor benötigen, kommt der noch mit in die folgende Visualisierung:



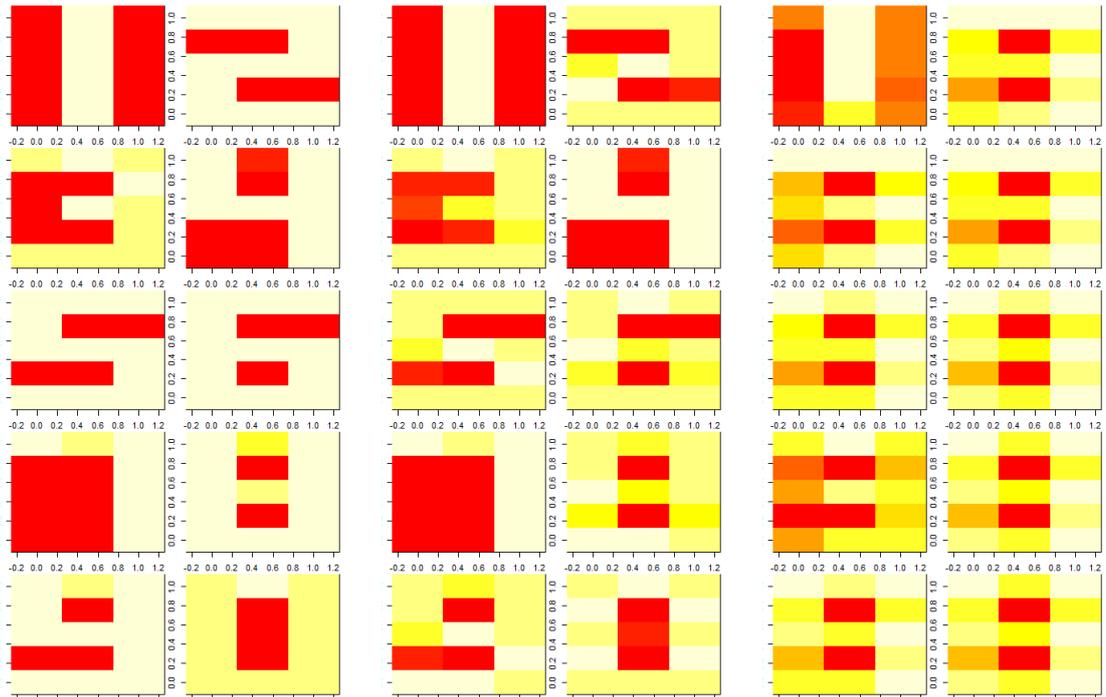
Possible Solution:

b) Wir rekonstruieren die Eingabematrix als $\hat{\mathbf{Z}} = \mu \cdot \mathbf{1} + \mathbf{U} \cdot \mathbf{Y} = \mu \cdot \mathbf{1} + \mathbf{U} \cdot \mathbf{U}^T \cdot \tilde{\mathbf{X}}$, wobei \mathbf{U} die ersten r Eigenvektoren des PCA-Ergebnisses enthält.

Natürlich brauchen wir nicht nur die Eigenfaces zur Rekonstruktion, sondern auch den Mittelwert $\mu = (0.9, 0.9, 0.9, 0.6, 0.1, 0.7, 0.7, 0.8, 0.9, 0.4, 0.1, 0.8, 0.7, 0.8, 0.9)^T$.

Erst bei einer Reduktion zu 7 Hauptkomponenten erhalten wir einen mittleren Absolutfehler $\sum_{i,j} |\hat{z}_{i,j} - x_{i,j}|$ von 1,88. Der mittlere DFFS (*distance from feature space*) = $\sum_{j=1}^M (\hat{\mathbf{z}}_j - \mathbf{x}_j)^2$ beträgt 0,0114. Mit sinkender Dimensionalität sinkt auch die Qualität der Rekonstruktion.

Die 1 scheint besonders gut rekonstruierbar zu sein, da sie sich von den anderen Zahlen stark abhebt – die anderen Zahlen sind recht stark untereinander korreliert und werden daher in niederdimensionalen Rekonstruktionen Opfer von korreliert rekonstruierten Variablen.



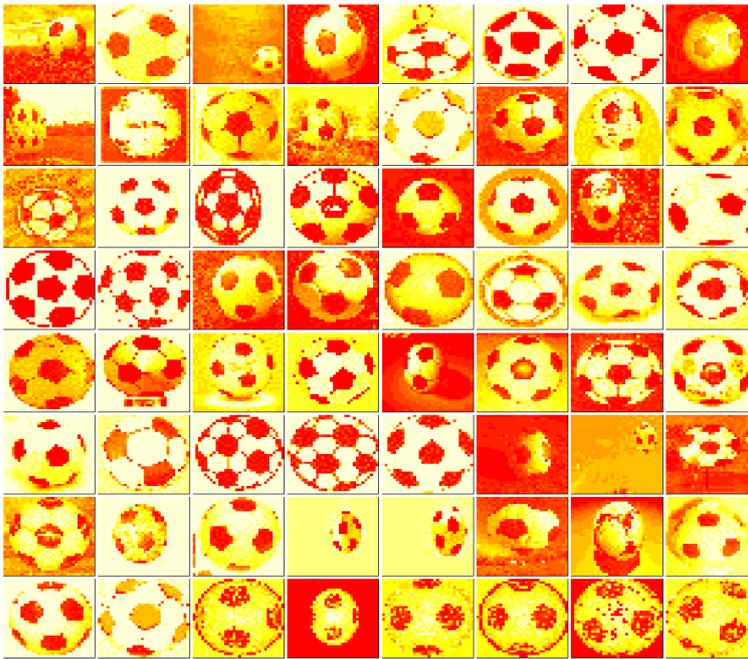
Exercise 10-4 Soccer Ball PCA

The Caltech 101 dataset consists of more than 9000 images which have each been assigned to one out of 102 classes. We consider down-scaled images ((32 × 32)-thumbnails) taken from the classes `soccer ball` and `faces easy`.

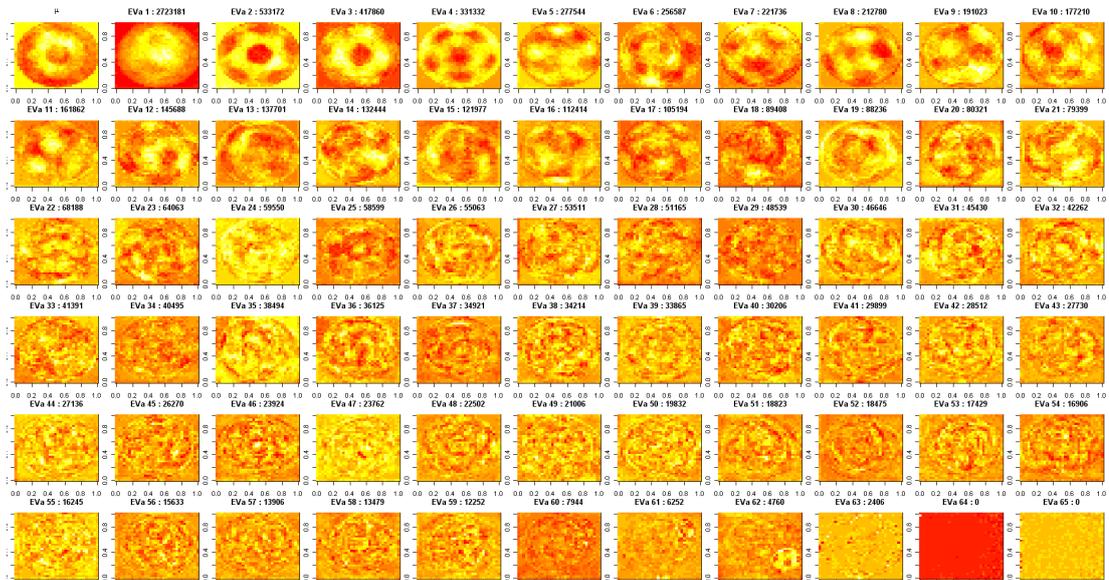
- Conduct a PCA on the 64 `soccer ball` images. Can the images be reconstructed losslessly with only part of the principal components?
- Now consider the `faces easy` dataset, consisting of 435 image. Can this dataset be adequately reconstructed using the principal components from part a)?
- Now consider the dataset of part a) and the principal components of the dataset of part b). Does reconstructing the soccer balls from the faces' components work?

Possible Solution:

Der Datensatz \mathbf{X} ist eine 1024×64 Matrix bestehend aus hübschen 32×32 Bildern:



a) μ ist der mittlere Pixelwert: $\hat{\mu} = \frac{1}{N} \sum_i \mathbf{x}_i$, $\tilde{\mathbf{X}}$ enthält die zentrierten Daten $\tilde{\mathbf{X}} = (\mathbf{x}_i - \hat{\mu})_i$. Wir bestimmen wieder die Eigenfaces als die Spalten in $\hat{\mathbf{U}}$ von $\hat{\Sigma} = \hat{\mathbf{U}}\hat{\mathbf{U}}^T$.



Der Datensatz \mathbf{X} wird nun auf r Dimensionen reduziert: $\mathbf{Y} = \mathbf{U}^T \cdot \tilde{\mathbf{X}}$. Rekonstruktion: $\hat{\mathbf{Z}} = \mu \cdot \mathbf{1} + \mathbf{U} \cdot \mathbf{Y} = \mu \cdot \mathbf{1} + \mathbf{U} \cdot \mathbf{U}^T \cdot \tilde{\mathbf{X}}$, mit \mathbf{U} als den ersten r Eigenvektoren des PCA-Ergebnisses.

Der absolute Fehler $\sum_{i,j} |\hat{\mathbf{z}}_{i,j} - \mathbf{x}_{i,j}|$ und der DFFS (*distance from feature space*) = $\sum_{j=1}^M (\hat{\mathbf{z}}_j - \mathbf{x}_j)^2$ sind bis einschließlich 63 Hauptkomponenten nahe 0.

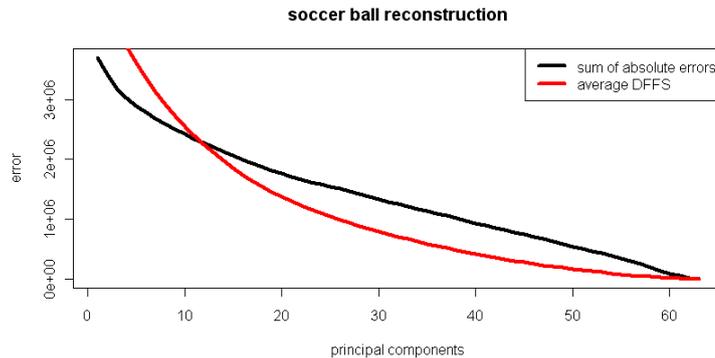
Warum genau 63? Direkt korreliert mit der Größe des Datensatzes abzüglich der bereits vorhandenen Information aus μ : $N - 1 = 63$. Also: im Gegensatz zum recht variationsarmen Nummerndatensatz ($N = 10$, aber 8 Hauptkomponenten reichten aus) kommt aus diesem Datensatz *keine* direkte Reduktion der Komplexität zustande: \mathbf{U} ist in $\mathbb{R}^{1024 \times 63}$ plus μ (1024 Dimensionen); damit sind wir trotz reduziertem $\mathbf{Y} \in \mathbb{R}^{63 \times 64}$ wieder beim Ausgangspunkt 1024×64 (plus den Overhead von \mathbf{Y}).

Possible Solution:

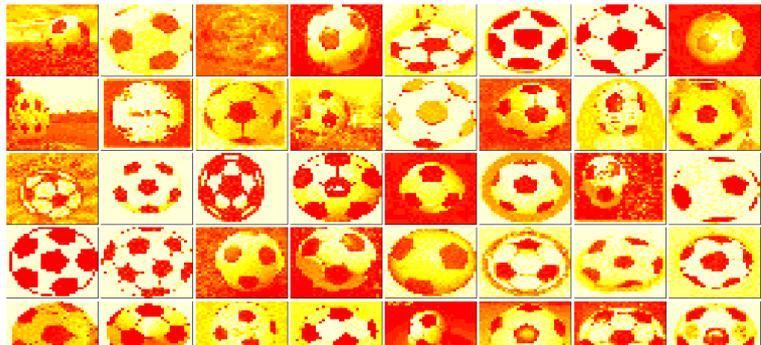
Mit weniger als 63 Dimensionen recht hohe Fehler; jeweils mehrere Tausend. Allerdings gilt es hier abzuwägen was tatsächlich als starker Fehler zu werten ist: beide Fehler sollte man noch durch die Anzahl der Bilder mal der Anzahl der Pixels teilen:

	absoluter Fehler	abs. Fehler / $(32^2 \cdot 64)$	mittlerer DFFS	DFFS / $(32^2 \cdot 64)$
dim = 63	0	0	0	0
dim = 62	14628	0,22	2406	2,35
dim = 50	540623	8,25	164512	160,66

Dieses Verhalten setzt sich natürlich für sinkende Dimensionen fort:

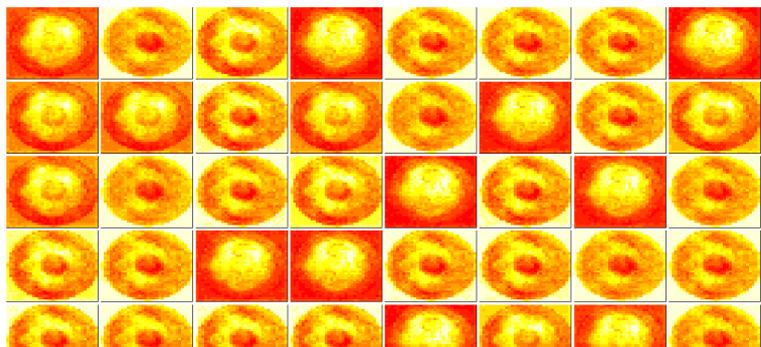


Die Visualisierung der $\hat{\mathbf{Z}} = \mu \cdot \mathbf{1} + \mathbf{U} \cdot \mathbf{Y} = \mu \cdot \mathbf{1} + \mathbf{U} \cdot \mathbf{U}^T \cdot \tilde{\mathbf{X}}$, läuft daher erwartbar ab.



61 Dimensionen:

Sollen die Daten aus einer Hauptkomponente rekonstruiert werden, läuft alles auf eine Abbildung des Mittelwertes ([fast] unverändert oder negiert) hinaus.



1 Dimension:

b) Wir verwenden das Modell (μ_S, \mathbf{U}_S) von oben um den Datensatz $\mathbf{X}_F \in \mathbb{R}^{1024 \times 435}$ zu reduzieren und zu rekonstruieren:

$$\begin{aligned} \hat{\mathbf{Z}}_F &= \mu_S \cdot \mathbf{1} + \mathbf{U}_S \cdot \mathbf{Y}_F = \mu_S \cdot \mathbf{1} + \mathbf{U}_S \cdot \mathbf{U}_S^T \cdot \tilde{\mathbf{X}}_F = \\ &= \mu_S \cdot \mathbf{1} + \mathbf{U}_S \cdot \mathbf{U}_S^T \cdot (\mathbf{X}_F - \mu_S \cdot \mathbf{1}) \end{aligned}$$

b.w.

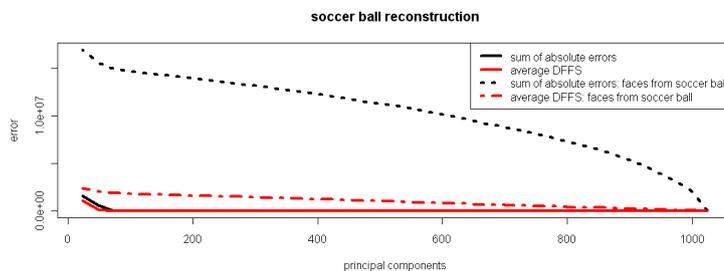
Possible Solution:



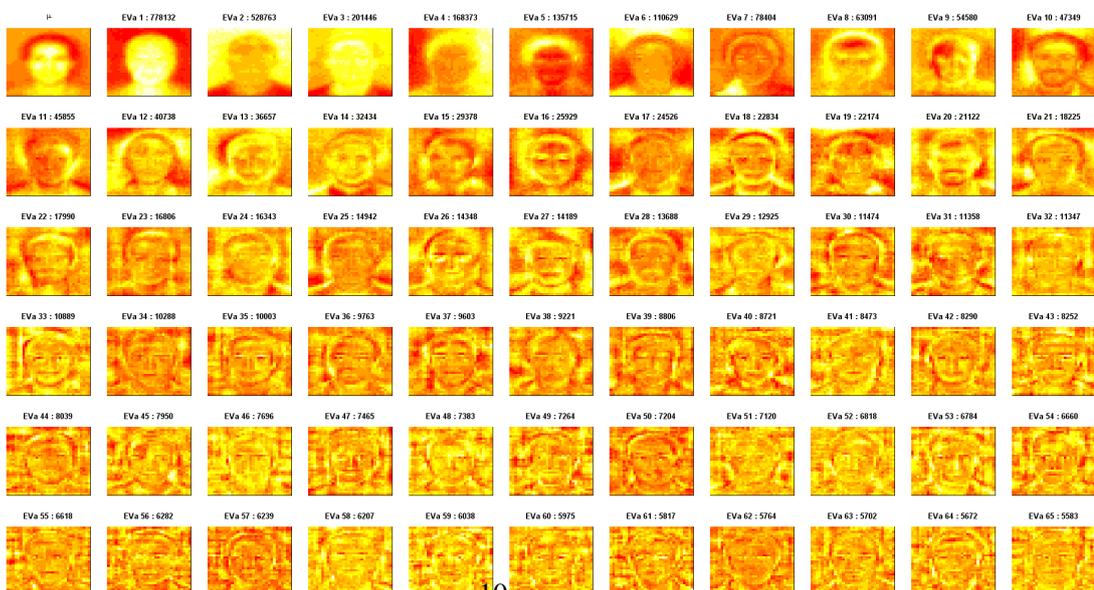
Der Gesichts-Datensatz lässt sich nicht mit den 63 oben ausgewählten Hauptkomponenten rekonstruieren:

Klar; der Rang der Matrix U reicht nicht, um den wesentlich größeren Datensatz zu rekonstruieren. Und offenbar sehen Fußbälle eben doch nicht aus wie Gesichter. Nimmt man wieder ganz \hat{U}_S , also mit $r = 1024$, funktioniert die Rekonstruktion wieder; allerdings ist das wenig sinnvoll, da wir ja lediglich die Identitätsmatrix einmal auf X_F multiplizieren.

Hier der Verlauf von Rekonstruktionen mit mehr als 63 Dimensionen:

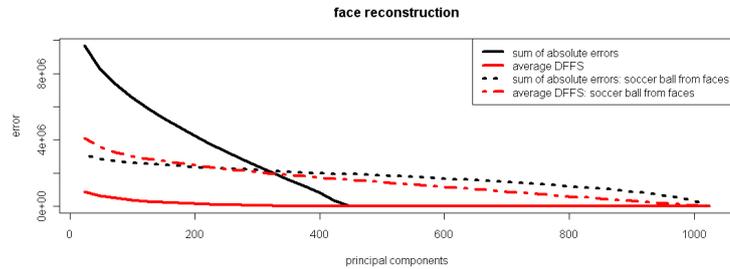


c) Die Zerlegung von X_F führt natürlich auch zu sehr schönen Eigenfaces:



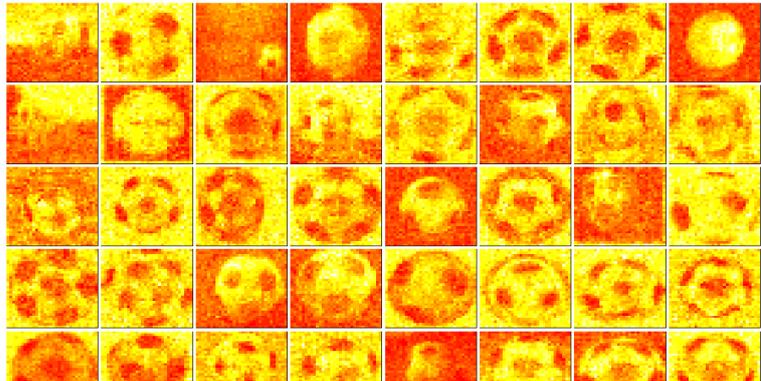
Possible Solution:

Deren Rekonstruktion ist aber etwas störrischer: erst bei 434 Hauptkomponenten haben wir Null Fehler. Gleiches Prinzip wie vorhin. Die Freiheitsgrade des Datensatzes $N = 435$ müssen wieder im Modell ($\mu_F \in \mathbb{R}^{1024}$, $U_F \in \mathbb{R}^{1024 \times 334}$) enthalten sein, da es keine perfekt korrelierten Bilder in X_F gibt.



Der Fußballdatensatz kann wieder nur mit der vollen Matrix $\hat{U}_F \in \mathbb{R}^{1024 \times 1024}$ rekonstruiert werden. Die Rekonstruktionsgüte nimmt vergleichbar stark ab mit der Anzahl der Hauptkomponenten wie in Teilaufgabe b).

300 Dimensionen:



63 Dimensionen:

