DIMENSIONALITY REDUCTION: PCA AND t-SNE
LECTURE: UNSUPERVISED LEARNING AND EVOLUTIONARY COMPUTATION USING R

Jakob Bossek

MALEO Group, Department of Computer Science, Paderborn University, Germany

16th Dec, 2024

Learning Goals

- Why handling high-dimensional data is a pain
- Principal Component Analysis
- t-distributed Stochastic Neighbour Embedding

Motivation: visualisation

Recall our setting: we have a data set $\mathcal{X}$ where each $x \in \mathcal{X}$ is a p-dimensional observation.

- For $p = 1$: visualisation is no problem at all
  Histogram, boxplot etc.
- For $p = 2$: visualisation is no problem at all
  Scatter-plots etc.
- For $p = 3$: visualisation in OK, but gets harder
  3D scatter-plot etc.
- For $p = 4$: puh!
  Scatter-plots with aesthetics for further variables, pair-wise scatter-plots etc.
- ...
- What about $p = 20$? ☹

Motivation: multivariate analysis

With a high number of observations ...

- Visualisation is hard
  For p features we would need

$$\binom{p}{2} = \frac{p(p-1)}{2} = \mathcal{O}(n^2)$$

  pairwise scatter-plots / correlations!

- ... further analysis is more difficult (and time-consuming)[1]
  Identification of outliers, interpretation of clustering results etc.

[1] Note that for many multi-variate methods the dimensionality p plays a rule in the running time bounds (see lecture notes).

Dimensionality reduction

Goal

Reduce the dimensionality of $\mathcal{X}$. I.e., replace the original p variables with q variables such that ideally

$$q \ll p$$

- Remove redundancy in data
- Identify correlated variables
- Identify hidden patterns/characteristics in/of the data set

Any ideas?

Towards a concept

Let's collect ideas

- Plain simple approach: select only one or two variables
  Not very useful!
- Use the q variables with the highest variance
  Better! But it does not account for inter-dependencies.
- If two variables are strongly correlated, keep just one
  Interesting!
- Use the mean $\frac{1}{p}\sum_{i=1}^{p} X_i$ as a *surrogate*[2]
  Ok, but not all variables are equally important.

So-called *Principal Component Analysis* (PCA) kind of combines all these ideas!

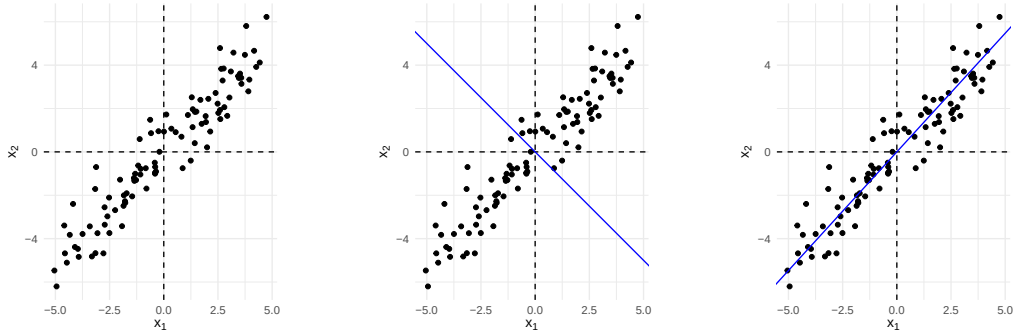
[2] Kind of a (combined) replacement for the original variables.

Principal Component Analysis

Basic idea

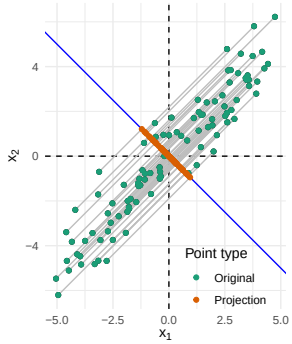
PCA = **P**rincipal **C**omponent **A**nalysis aims to identify the principal "directions" in which the data varies most!

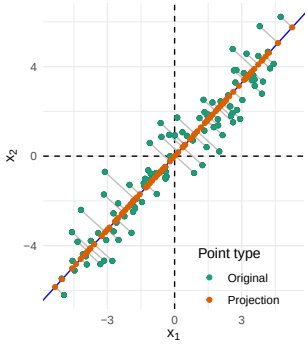
Assumption: direction with largest variation is the most important! Direction with second-largest variation is the second-most important etc.



Geometric interpretation

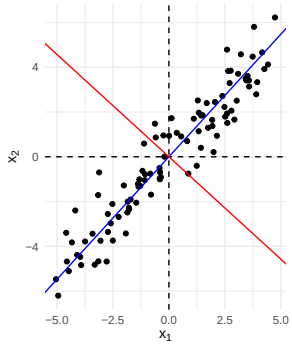
Minimize the sum of orthogonal projections:

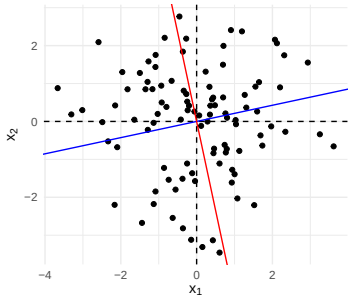




Geometric interpretation

PCA makes sense if there are many correlations (i.e., high redundancy) in the data:





Linear algebra basics

Assumption: data is centered, i.e., $E(X_i) = 0, 1 \leq i \leq p$

Focus on the structure of the data.

Definition (Covariance matrix)

The *covariance matrix* $\mathrm{Cov}(X)$ of a random vector $X = (X_1, \ldots, X_p)^T$ contains the pairwise covariance values. I.e., for all $1 \leq i, j \leq p$:

$$\mathrm{Cov}(X)_{ij} = \mathrm{Cov}(X_i, X_j) = E\left[(X_i - E(X_i)) \cdot (X_j - E(X_j))\right].$$

In matrix notation using our assumption:

$$\mathrm{Cov}(X) = E(XX^T) - \underbrace{E(X)E(X)^T}_{=0} = E(XX^T).$$

Principal Component Analysis (PCA) (Johnson and Wichern 2013)

Key idea: replace original (correlated) variables $X_1, \ldots, X_p$ with p new variables $Y_1, \ldots, Y_p$ – the so-called *Principal Components* (PCs) – where

$$Y_i = \sum_{j=1}^{p} \gamma_{ij} X_j = \gamma_{i1} X_1 + \gamma_{i2} X_2 + \ldots + \gamma_{ip} X_p. \tag{1}$$

- The new variables are *linear combinations of the original variables*
- They have decreasing variance, i.e.,

$$\mathrm{Var}(Y_1) \geq \mathrm{Var}(Y_2) \geq \ldots \geq \mathrm{Var}(Y_p)$$

  I.e., they encode a decreasing ammount of information.
- $\mathrm{Cov}(Y_i, Y_j) = 0$ for $i \neq j \rightsquigarrow$ uncorrelated

  I.e., each new variable reveals new information.
- Replacing p with p not very helpful

  Select first q PCs that account for the majority of the variance.

Construction of the PCs

Iterative process: 1st PC

For the first PC

$$Y_1 = \gamma_1^T X = \sum_{j=1}^{p} \gamma_{1j} X_j = \gamma_{11} \cdot X_1 + \gamma_{12} \cdot X_2 + \ldots + \gamma_{1p} \cdot X_p$$

we maximize the variance explained:

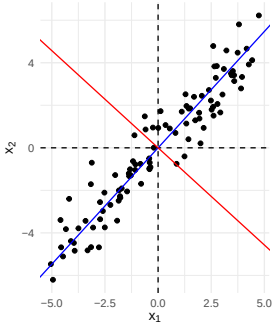
$$\underbrace{\max_{\gamma_1} \mathrm{Var}(\gamma_1^T X)}_{\text{Maximize variance}} \text{ s.t. } \underbrace{\gamma_1^T \cdot \gamma_1 \stackrel{!}{=} 1}_{\text{Normalization}}.$$

- Why is the normalization necessary?
  Otherwise we could grow the variance without limits by increasing the components of the weights $\gamma_1 = (\gamma_{11}, \ldots, \gamma_{1p})$!

Construction of the PCs

Iterative process: i^{th} PC

Recall, that PCs need to be orthogonal (i.e., uncorrelated) to each other. I.e., the second PC needs to be orthogonal to the first PC and so on.



Construction of the PCs

Iterative process: i^{th} PC

For PC number $1 < i \leq p$

$$Y_i = \sum_{j=1}^{p} \gamma_{ij} X_j = \gamma_{i1} \cdot X_1 + \gamma_{i2} \cdot X_2 + \ldots + \gamma_{ip} \cdot X_p$$

we maximize the variance explained under the additional constraint that it must be uncorrelated to all previous PCs $1 \leq j < i$

$$\underbrace{\max_{\gamma_i} \text{Var}(\gamma_i^T X)}_{\text{Maximize variation}} \text{ s.t. } \underbrace{\gamma_i^T \cdot \gamma_i \stackrel{!}{=} 1}_{\text{Normalization}} \text{ and } \underbrace{\gamma_i^T \cdot \gamma_j = 0 \quad \forall j < i}_{\substack{\text{Require being uncorrelated} \\ \text{to all previous PCs}}}.$$

where the condition $\gamma_i^T \cdot \gamma_j = 0$ is equivalent to $\text{Cov}(Y_i, Y_j) = 0$.

Construction of the PCs

Let's write the variance of the i-th PC differently: For the random vector X and weight vector for the PC γ_i we obtain:

$$\begin{aligned}
\mathrm{Var}(\gamma_i^T X) &= E((\gamma_i^T X)^2) \\
&= E((\gamma_i^T X)(\gamma_i^T X)) \\
&= E((\gamma_i^T X)(X^T \gamma_i)) \\
&= E(\gamma_i^T (XX^T)\gamma_i) \\
&= \gamma_i^T \underbrace{E(XX^T)}_{=\mathrm{Cov}(X)} \gamma_i = \gamma_i^T \Sigma \gamma_i
\end{aligned}$$

Here, we used

$$\gamma_i^T X = \sum_{j=1}^{p} \gamma_{ij} X_j = \sum_{j=1}^{p} X_j \gamma_{ij} = X^T \gamma_i.$$

Solving the optimization problem

Now we can solve the *constrained* optimization problem (for the first PC)[3]

$$\underbrace{\max_{\gamma_1} \mathrm{Var}(\gamma_1^T X)}_{\text{Maximize variance}} \text{ s.t. } \underbrace{\gamma_1^T \cdot \gamma_1 \stackrel{!}{=} 1}_{\text{Normalization}}$$

$$\equiv \quad \max_{\gamma_1} \gamma_1^T \Sigma \gamma_1 \text{ s.t. } \gamma_1^T \cdot \gamma_1 - 1 \stackrel{!}{=} 0$$

by using *Lagrange-multipliers*. I.e., we define a variable λ and solve the *unconstrained* problem

$$\max_{\gamma_1, \lambda} L(\gamma_1, \lambda) = \gamma_1^T \Sigma \gamma_1 - \lambda(\gamma_1^T \cdot \gamma_1 - 1)$$

[3] Works analogously for the other PCs.

Solving the optimization problem

We maximize

$$\max_{\gamma_1, \lambda} L(\gamma_1, \lambda) = \gamma_1^T \Sigma \gamma_1 - \lambda(\gamma_1^T \cdot \gamma_1 - 1)$$

analytically by calculating the partial deriviatives of $L(\gamma_1, \lambda)$ with respect to γ_1 and λ:

$$(I) \quad \frac{\partial L}{\partial \gamma_1} = 2\Sigma\gamma_1 - 2\lambda\gamma_1 \stackrel{!}{=} 0 \qquad (II) \quad \frac{\partial L}{\partial \lambda} = \gamma_1^T \gamma_1 - 1 \stackrel{!}{=} 0$$

This holds exactly if

$$\Sigma\gamma_1 = \lambda\gamma_1 \text{ and } \gamma_1^T \gamma = 1$$

I.e., γ_1 is the (normalized) eigenvector to the largest eigenvalue λ of the covariance matrix Σ.

Calculation via matrix algebra

I.e., PCA $\equiv$ determining eigenvalues and eigenvectors!

Theorem (Eigenvalue decomposition (EVD))

Every symmetric matrix $\Sigma \sim (p, p)$ can be decomposed into

$$\Sigma = A \cdot D \cdot A^T$$

where

- *$D = diag(\lambda_1, \ldots, \lambda_p)$ contains the (non-negative) eigenvalues of Σ in decending order*
- *$A \sim (p, p)$ is an orthogonal matrix[4] and contains the respective eigenvectors in the columns. I.e., the the i-th column is γ_i.*

[4] This means $A^T A = A A^T = I_p$, i.e., the column-vectors are pairwise orthogonal.

Calculation via matrix algebra

Consequence: calculate the EVD of the covariance matrix $\Sigma = A \cdot D \cdot A^T$ where

$$A = [\gamma_1 \dots \gamma_p] = \begin{bmatrix} \gamma_{11} & \gamma_{21} & \cdots & \gamma_{p1} \\ \gamma_{12} & \gamma_{22} & \cdots & \gamma_{p2} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{1p} & \gamma_{2p} & \cdots & \gamma_{pp} \end{bmatrix} \sim (p, p).$$

Then, the principal components are given by

$$Y = A^T X \Leftrightarrow \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_p \end{bmatrix} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1p} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{p1} & \gamma_{p2} & \cdots & \gamma_{pp} \end{bmatrix} \cdot \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^p \gamma_{1i} X_i \\ \sum_{i=1}^p \gamma_{2i} X_i \\ \vdots \\ \sum_{i=1}^p \gamma_{pi} X_i \end{bmatrix}.$$

PCs are uncorrelated

Now we convince ourselves that the principal components are indeed uncorrelated. To this end we calculate the covariance matrix of Y yielding:

$$
\begin{aligned}
\mathrm{Cov}(Y) &= E[A^T X (A^T X)^T] && \text{(by definition)} \\
&= E[A^T X (X^T \underbrace{(A^T)^T}_{=A})] && \text{(properties of transposition)} \\
&= A^T \cdot \underbrace{E[XX^T]}_{=\Sigma} \cdot A && \text{(by linearity)} \\
&= A^T \Sigma A \\
&= D && \text{(by EVD)} \\
&= \mathrm{diag}(\lambda_1, \ldots, \lambda_p). && \text{uncorrelated!}
\end{aligned}
$$

PCA: recipe for given data

Given feature vectors $x_1, \ldots, x_N \in \mathbb{R}^p$ and data matrix $X \sim (N, p)$

1. Center data, i.e., subtract mean values
2. Optional: standardize data to account for different scales
3. Calculate the empirical covariance/correlation matrix

$$\Sigma = \frac{1}{N} \sum_{i=1}^{p} x_i \cdot x_i^T = \frac{1}{N} X^T X \sim (p, p)$$

4. Calculate the eigenvectors and eigenvalues:

$$\Sigma \gamma_i = \lambda_i \gamma_i$$

5. Sort eigenvalues (and respective) vectors in decreasing order yielding rotation matrix A (columns are from left to right the sorted eigenvectors)
6. Project data to PC-space via

$$Y = X \cdot A.$$

Example

We have $N = 5$ data points on $p = 3$ numeric features $X = (X_1, X_2, X_3)^T$. I.e.

USA state	X_1 (**Murder**)	X_2 (**Rape**)	X_2 (**Robbery**)
ME	2.0	14.8	28
NH	2.2	21.5	24
VT	2.0	21.8	22
MA	3.6	29.7	193
RI	3.5	21.4	119

The covariance matrix is

$$\Sigma = \text{Cov}(X) = \begin{pmatrix} 0.668 & 2.962 & 59.34 \\ 2.962 & 27.913 & 314.61 \\ 59.335 & 314.615 & 5863.70 \end{pmatrix}$$

Example

Calculation of the eigenvalues solving[5]

$$\det(\Sigma - \lambda I_3) = 0$$

and subsequently solving the linear equality systems to get eigenvectors yields:

$$\lambda_1 = 5881.2125 \geq \lambda_2 = 11.0054 \geq \lambda_3 = 0.0631$$

and

$$\gamma_1 = \begin{pmatrix} -0.01010 \\ -0.05367 \\ -0.99851 \end{pmatrix}, \gamma_2 = \begin{pmatrix} 0.02077 \\ -0.99835 \\ 0.05346 \end{pmatrix}, \gamma_3 = \begin{pmatrix} 0.9997 \\ 0.0202 \\ -0.0112 \end{pmatrix}.$$

[5] See math foundations! ☺

Example

With this we obtain the *rotation matrix*

$$A = \begin{bmatrix} \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix} = \begin{pmatrix} -0.01010 & 0.02077 & 0.9997 \\ -0.05367 & -0.99835 & 0.0202 \\ -0.99851 & 0.05346 & -0.0112 \end{pmatrix}$$

and the EVD

$$\Sigma = \underbrace{\begin{pmatrix} -0.01010 & 0.02077 & 0.9997 \\ -0.05367 & -0.99835 & 0.0202 \\ -0.99851 & 0.05346 & -0.0112 \end{pmatrix}}_{=A} \cdot \underbrace{\begin{pmatrix} 5881.21 & 0 & 0 \\ 0 & 11.01 & 0 \\ 0 & 0 & 0.06 \end{pmatrix}}_{=D=\mathrm{Cov}(Y)} \cdot \underbrace{\begin{pmatrix} -0.01010 & -0.05367 & -0.99851 \\ 0.02077 & -0.99835 & 0.05346 \\ 0.9997 & 0.0202 & -0.0112 \end{pmatrix}}_{=A^T}.$$

On the variation explained

We know due to the EVD:

1. Variance of the PCs is decreasing:

$$\text{Var}(Y_1) \geq \text{Var}(Y_2) \geq \ldots \geq \text{Var}(Y_p)$$

2. Variance corresponds to eigenvalues, i.e.:

$$\text{Var}(Y_i) = \lambda_i \, \forall 1 \leq i \leq p$$

3. The total variance of the PCs is equal to the total variance of the original variables:

$$\sum_{i=1}^{p} \text{Var}(Y_i) = \text{tr}(D) = \text{tr}(A^T \cdot \Sigma \cdot A)$$

$$= \text{tr}(\underbrace{A^T \cdot A}_{=I_p} \cdot \Sigma) = \text{tr}(\Sigma) = \sum_{i=1}^{p} \text{Var}(X_i).$$

On the variation explained

- The first PC accounts for a fraction of

$$P_1 = \frac{\mathrm{Var}(Y_1)}{\sum_{j=1}^{p} \mathrm{Var}(Y_j)} = \frac{\lambda_1}{\mathrm{tr}(\Sigma)}$$

  of the total variation.
- In general the i^{th} PC accounts for a proportion of

$$P_i = \frac{\mathrm{Var}(Y_i)}{\sum_{j=1}^{p} \mathrm{Var}(Y_j)} = \frac{\lambda_i}{\mathrm{tr}(\Sigma)}$$

  of the total variation.
- The first $1 \leq k \leq p$ PCs in sum explain

$$P^{(k)} = \frac{\sum_{i=1}^{k} \mathrm{Var}(Y_i)}{\sum_{j=1}^{p} \mathrm{Var}(Y_j)} = \frac{\sum_{i=1}^{k} \lambda_i}{\mathrm{tr}(\Sigma)}$$

  of the total variation.

Scaling issues

- If variation of original variables differ strongly, variables with high variance will dominate first PCs
- I.e., extracting PCs from covariance matrix only if the are roughly on the same scale
  This is rarely the case in practise!
- Solution: standardize to unit variance by using the correlation matrix instead

$$R = D^{-1/2} \cdot \Sigma \cdot D^{-1/2} \text{ with } D^{-1/2} = \text{diag}(1/s_1, \ldots, 1/s_p)$$

with $s_i = \sqrt{s_i^2}$ being the sample standard variation.
Now variables are "equally important" and scale-independent.

How many PCs are enough?

Complete variation (usually) captured only if *all* PCs are used

Unless there are perfect linear relationships in the data.

Heuristics based on average PC variation

Define a *threshold*: desired amount of explained variation (usually 70% to 90%)

- Exclude all PCs with variance below the average variance

$$\frac{1}{p}\sum_{i=1}^{p}\lambda_i$$

- If correlation matrix S is used: $\mathrm{tr}(S) = p$ and average distance equals 1

  Exclude all PCs Y_i with $\lambda_i < 1$.

How many PCs are enough?

Scree-plot[6]

Define a *threshold*: desired amount of explained variation (usually 70% to 90%)

- Line-plot of variations λ_k against the PC number k and search for "elbow"/"knee" in the

  Rationale: steep downwards trend before, flattend after the "elbow"

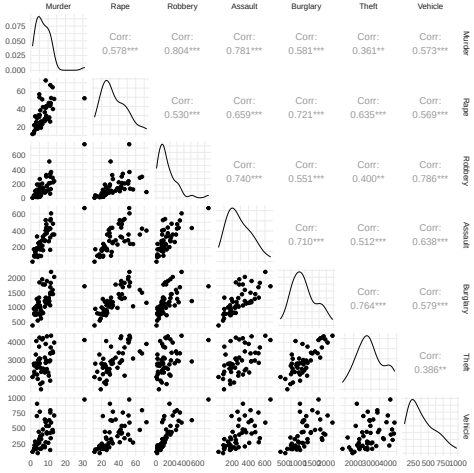
- Alternative: line-plot or bar-plot of

$$P^{(k)} = \frac{\sum_{i=1}^{k} \mathrm{Var}(Y_i)}{\sum_{j=1}^{p} \mathrm{Var}(Y_j)} = \frac{\sum_{i=1}^{k} \lambda_i}{\mathrm{tr}(\Sigma)}$$

  against k and search for lowerst k, where $P^{(k)}$ is larger than the chosen threshold.

[6] Approach similar to elbow-plot in k-means clustering.

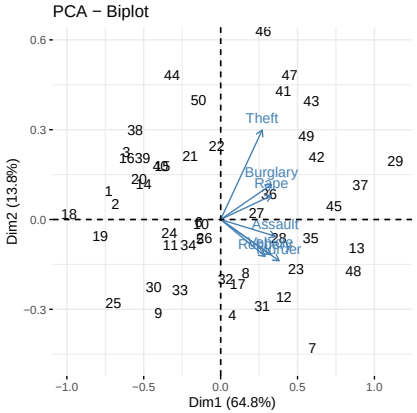
PCA example: US Crime Data



PCA: visualisation of results

A (very) common method is the so-called *bi-plot* (K.R.Gabriel 1971). Given two PCs it displays

1. **Points** to represent the scores of the observations on the PCs
   I.e., the transformed observations.
2. **Vectors** to represent the loadings of the PCs
   I.e., the cooefficients/weights of the PCs.



Biplot: interpretation of points

We can interpret the *relative position of points*:

- Points close together have similar scores for the displayed PCs
  I.e., they also are similar with respect to the original variables.
- Points / point groups far apart have dissimilar scores
  I.e., they are likely dissimilar with respect to original variables.
- Large distance from origin indicates large interaction effect with at least one variable (vector).

Biplot: interpretation of vectors

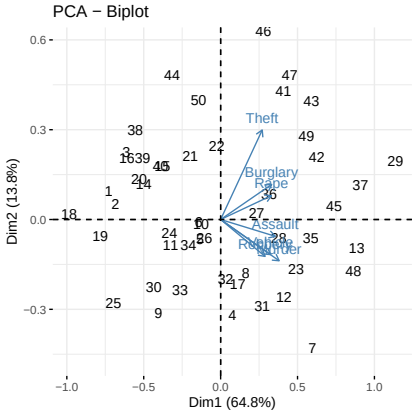
Following Rositter (Rositter 2017) we can interpret the vectors in the biplot as follows:

- *Orientation* of the vector to the PC-axis: the more parallel the vector is to a PC, the more it contributes solely to this PC.
- *Angle(s) between vector(s)*: indicates variable correlations (similar *response pattern*)
  Small angle $\rightsquigarrow$ high positive correlation,
  right angles $\rightsquigarrow$ no correlation,
  opposite angles $\rightsquigarrow$ negative correlation
- *Vector length*: indicator for variable variability in the two displayed PCs
  The longer, the higher the variablility. Short vectors are represented better by other PCs.

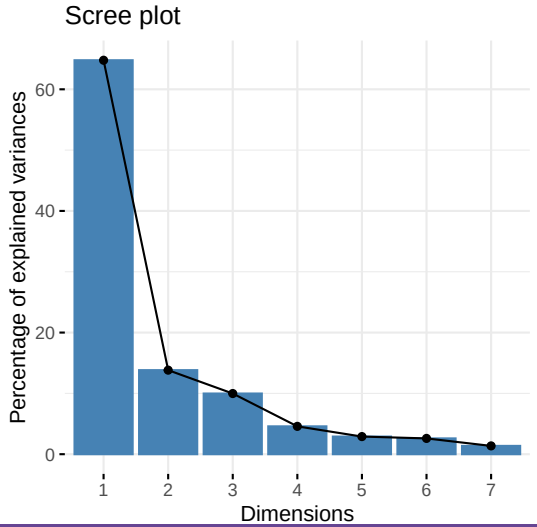
PCA: visualisation of results

A (very) common method is the so-called *bi-plot* (K.R.Gabriel 1971). Given two PCs it displays

1. **Points** to represent the scores of the observations on the PCs
   I.e., the transformed observations.
2. **Vectors** to represent the loadings of the PCs
   I.e., the cooefficients/weights of the PCs.



PCA example: US Crime Data scree-plot



PCA: final remarks

- Replaces p original variables with p "synthetic" variables maximizing explained variation
  Uncorrelated linear combinations of the originals.
- Scree-plots help to find the right number of PCs
- The biplot is a powerful tool to visualize two PCs
  Interpretation of both scores and loadings.
- Basically relies on linear algebra: Eigenvalue decomposition calculation
  $\rightsquigarrow$ Computationally efficient (see lecture notes for details).

t-Distributed Stochastic Neighbor Embedding

Drawbacks of PCA

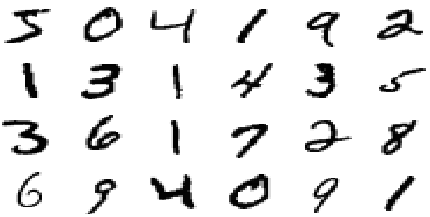
Major weakness

- PCA mainly aims at preserving the structure of dissimilar points
  - Maximize explained variance!
    Based on squared distances since $\mathrm{Var}(X) = E((X - E(X))^2)$.
  - Distance of dissimilar points is large
    $\rightsquigarrow$ huge impact on variance.
  - Distance of similar points is small
    $\rightsquigarrow$ neglebible influence on variance
- **Consequence**: PCs are incapable of preserving local structure, since focus is on global structure
  $\rightsquigarrow$ often no convincing results for high dimensional data sets.

An example where PCA fails

The MNIST data-set (Lecun et al. 1998)[7]

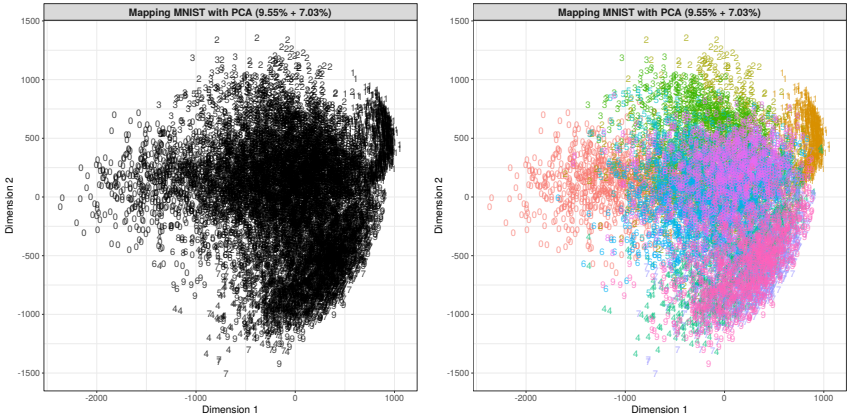
- Data base of handwritten digits stored as images with resolution 28×28 pixels
  I.e., $p = 28^2 = 784$
- Labelled data (10 classes)
  60 000 training examples and 10 000 test examples.



[7] http://yann.lecun.com/exdb/mnist/

An example where PCA fails

PCA applied to 10 000 digits from MNIST



t-Distributed Stochastic Neighbor Embedding

t-SNE: **t**-Distributed **S**tochastic **N**eighbor **E**mbedding

Main characteristics

- Stochastic algorithm
- Preserve local proximity of similar observations in low-dimensional space
  Address major weakness of PCA.
- and keep dissimilar observations far apart from each other in low-dimensional space
  Keep good behavior of PCA.

t-Distributed Stochastic Neighbor Embedding

t-SNE in a nutshell

1. Compute pairwise distances $d(x_i, x_j) = ||x_i - x_j||^2, 1 \leq i, j \leq N$ in original (high-dim.) space[8] and represent the distances as joint probabilities p_{ij}
2. Randomly place points $y_1, \ldots, y_N$ in low-dimensional target space
3. Calculate $d(y_i, y_j) = ||y_i - y_j||^2, 1 \leq i, j \leq N$ an likewise represent as joint probabilities q_{ij}
4. Minimize mismatch between p_{ij} and q_{ij} for all pairs by optimizing cost function

[8] Here, we use Euclidean distance, but other measures are perfectly possible.

t-Distributed Stochastic Neighbor Embedding

Step 1:Joint probabilities for original space

We first define the *conditional probabilities*

$$p_{j|i} = \frac{\exp\left(-||x_i - x_j||^2/\sigma_i^2\right)}{\sum_{k=1,k\neq i}^{N} \exp\left(-||x_i - x_k||^2/\sigma_i^2\right)} \text{ with } p_{j|j} = 0.$$

With the words of Van der Maaten (Maaten and Hinton 2008): "The similarity of datapoint x_j to data point x_i is the conditional probability $p_{j|i}$, that x_i would pick x_j as its neighbor if neighbors were picked in propotion to their probability density under a Gaussian centered at x_i."
and then set the probability p_{ij} for i (j) to select j (i) as its neighbor to

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}.$$

The variance/bandwidth σ_i is set based on the density

Smaller values of σ_i in dense regions, higher in less dense regions.

t-Distributed Stochastic Neighbor Embedding

Step 2: Probabilities in target space

We first define the *conditional probabilities*

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k=1, k \neq i}^{N} (1 + ||y_i - y_k||^2)^{-1}} = q_{ji}$$

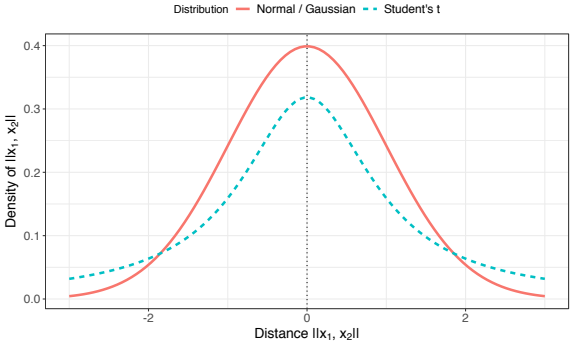
- q_{ij} is based on a *Student's t-distribution with one degree of freedom (Cauchy distribution)*

  $\rightsquigarrow$ ensures that distant observations will can be set far apart in target space.

- Low-dimensional representation will be (almost) invariant o changes in very distant observations

t-Distributed Stochastic Neighbor Embedding

Gaussian distribution vs. t-distribution



Observation: t-distribution is heavy-tailed, i.e., the likelihood of exteme deviations from the mean is higher.

t-Distributed Stochastic Neighbor Embedding

Optimization step

Recall: if $y_1, \ldots, y_N$ model $x_1, \ldots, x_N$ nicely for each pair $1 \leq i, j \leq N$

$$p_{ij} - q_{ij} \approx 0$$

would hold!

t-SNE measures the deviation by the sum of the so-called *Kullback-Leibler divergences*

$$C = \sum_{i=1}^{N} \sum_{j=1}^{N} p_{ij} \cdot \log\left(\frac{p_{ij}}{q_{ij}}\right) \to \text{min!}$$

t-Distributed Stochastic Neighbor Embedding

Algorithm t-Distributed Stochastic Neighbor Embedding (t-SNE)

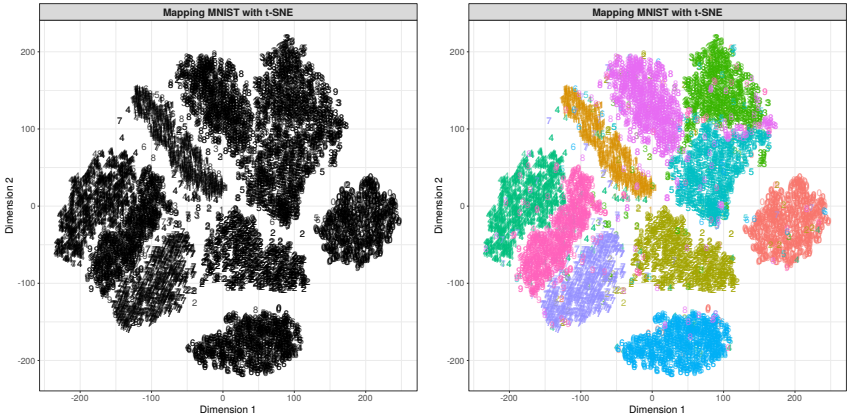
Require: Data set $\mathcal{X} = \{x_1, \ldots, x_N\}$, $x_i \in \mathbb{R}^p$, target dimension q, perplexity parameter σ, maximum number of iterations $T_{\max}$, learning rate η and momentum $\alpha(\cdot)$

1: Compute all pairwise distances $d(x_i, x_j), 1 \leq i, j \leq N$
2: Calculate joint probabilities p_{ij}
3: Sample initial targets $\mathcal{Y}^{(0)} = \{y_1^{(0)}, \ldots, y_N^{(0)}\}$ using $\mathcal{N}(0, 10^{-4} \cdot I_q)$
4: **for** $t \leftarrow 1$ to $T_{\max}$ **do**
5: Compute $d(y_i, y_j)$ and q_{ij}
6: Calculate gradients $\frac{\partial C}{\partial y_i}$
7: Set $y_i^{(t)} = y_i^{(t-1)} + \eta \cdot \frac{\partial C}{\partial y_i} + \alpha(t) \cdot \left(y_i^{(t-1)} - y_i^{(t-2)}\right)$
8: **return** $\mathcal{Y}^{(T_{\max})}$

Remark: learning rate η and momentum $\alpha(\cdot)$ are parameters of the gradient descent search

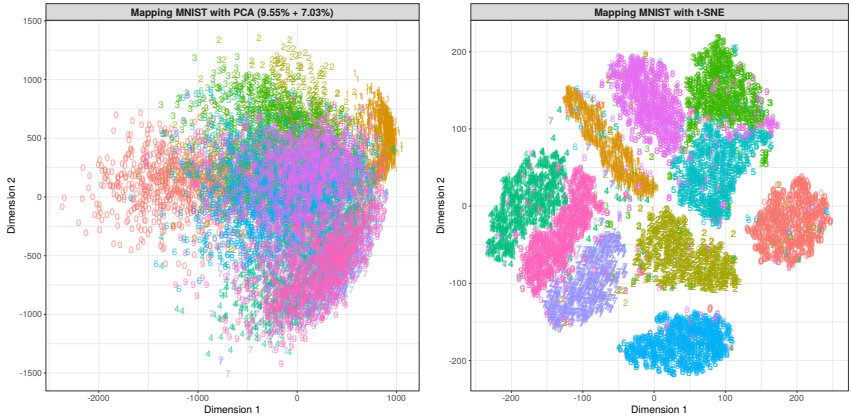
t-Distributed Stochastic Neighbor Embedding

t-SNE applied to 10 000 digits from MNIST



t-Distributed Stochastic Neighbor Embedding

PCA versus t-SNE on 10 000 samples from MNIST



t-Distributed Stochastic Neighbor Embedding

Remarks

- Very promising results for a variety of high-dimensional applications
- Computationally demanding tue to numerous distance recalculations
- Stochastic due to initial sample
  $\rightsquigarrow$ multiple runs recommended
- Visit t-SNE's webpage[9] for
  - an overview of further enhancements
  - a link to a Google Techtalk on t-SNE by one of its main authors
  - implementations etc. (package `snedata` in R)

[9] `http://lvdmaaten.github.io/tsne/`

What we learned today

- More than 3 dimensions are difficult to visualise
- PCA is a means to reduce the variables to linear combinations of the original variables (aims for explaining as much variance as possible)
- *t*-SNE is another powerful methods aiming for preserving the local structure

References I

Johnson, R.A. and D.W. Wichern (2013). *Applied Multivariate Statistical Analysis: Pearson New International Edition*. Pearson Education. ISBN: 9781292037578.

K.R.Gabriel (1971). "The biplot graphic display of matrices with application to principal component analysis 1.". In: *Biometrika* 58.3, pp. 453–467.

Rositter, D. G. (2017). *Tutorial: An example of statistical data analysis using the R environment for statistical computing*. URL: http://www.css.cornell.edu/faculty/dgr2/_static/files/R_PDF/corregr.pdf.

Lecun, Y. et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324. DOI: 10.1109/5.726791.

Maaten, Laurens van der and Geoffrey Hinton (2008). "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9, pp. 2579–2605. URL: http://www.jmlr.org/papers/v9/vandermaaten08a.html.

Hinton, Geoffrey and Sam Roweis (2003). "Stochastic Neighbor Embedding". In: *Advances in neural information processing systems* 15. Ed. by S Thrun S Becker and KEditors Obermayer, pp. 833–840.