

# Some organizational issues

# HISPOS

- Remember to register before 27 May
  - Withdrawal one week before the first exam latest
  - Informatik, Medieninformatik, CuK, VC, Cybersicherheit, Promotion Vorb. contact `studium@cs.uni-saarland.de` if problems
  - Wirtschaftsinformatiker, Erasmuss, Juniorstudenten, some others: no HISPOS

# Sample solutions

- Sample solutions to homeworks are available
- Require username and password outside university network

# Programming assignment

- First programming assignment is available
- Data requires username and password outside uni network (same as sample solutions)
- We provide some examples, try it out before tomorrow's tutorial
  - Go to tutorial if you have any issues

# The assignment

D5: Databases and Information Systems  
Data Mining and Matrices, SS 2015  
Programming assignment #1: SVD and pre-processing  
Due: **14 June 2015** at 23:29 CEST



You must hand in machine-typed report in PDF format and a script file (e.g. \*.R file). The report must explain your approach to the problem, the results you obtained, and your interpretation of the results. Naturally, the report must also answer to any direct question presented in the problems. You can, and in many cases should, add plots and other illustrations to your answers. The script file must show every step you have taken to solve these tasks, that is to say, if we run the script file we must get the same results you reported and see the same figures you presented. You can discuss these problems with other students, and you are encouraged to discuss with the tutor, but everybody must hand in their own answers and own code. Return your answers by email to [dmm15@mpi-inf.mpg.de](mailto:dmm15@mpi-inf.mpg.de). Remember to write your name and matriculation number to every answer sheet!

## Task 1: Normalization

Download the data and utility files from [http://resources.mpi-inf.mpg.de/departments/d5/teaching/ss15/dmm/assignments/assignment\\_1.zip](http://resources.mpi-inf.mpg.de/departments/d5/teaching/ss15/dmm/assignments/assignment_1.zip). That package contains file `assignment1.R`. You can fill your answers to that file and return it as a part of your solution.

Follow the steps in the file to load the `worldclim` data. This data contains information about the bioclimatic conditions (minimum, maximum, and average temperature in degrees Celcius and average precipitation in millimetres) per months in Europe. Spend some time exploring the data (you do not need to report these explorations, though). Compute the SVD of the data, and plot the first two left singular vectors to a map of Europe. This is also explained in the provided file. Can you interpret the results?

Play around with different color schemes and markers. Can you make the results more interpretable that way? Does the color scheme effect the interpretation?

Normalize the data to z-scores. Given the type of data we have, do you think this normalization is sensible?

Compute the SVD of the normalized data, and plot again the first two left singular vectors. Have they changed? Has your interpretation changed? Why?

## Task 2: Selecting the rank

In this task, we use the normalized `worldclim` data that you did in the previous task. Compute the SVD of the data. Try the following rank selection methods to decide what would be a good rank for the truncated SVD:

```
## Plot the first column of U. The color indicates the value, with
## red being low, green being middle, and blue being high
## Try
##?color.scale
## for more information about the color scale.
## If you don't have working rworldmap, replace 'points' with 'plot'
points(coord[,1], coord[,2], col=color.scale(U[,1], c(1, 0, 0), c(0, 1,
0), c(0, 0, 1), color.spec="rgb"))

## Alternative plot with different color scheme and filled circles
points(coord[,1], coord[,2], col=color.scale(U[,1], c(0,1), 0.8, 1,
color.spec="hsv"), cex=.6, pch=19)

## A color legend to explain the colors
color.legend(xLim[1]+1, yLim[1]-5, xLim[2]-1, yLim[1]-3, c(round(min(U[,
1]), 4), round(mean(U[,1]), 4), round(max(U[,1]), 4)),
color.scale(sort(U[,1]), c(0,1), 0.8, 1, color.spec="hsv"),
gradient="x")

## Plot the second column
plot(map, xlim=xLim, ylim=yLim, asp=1)
points(coord[,1], coord[,2], col=color.scale(U[,2], c(0,1), 0.8, 1,
color.spec="hsv"))
color.legend(xLim[1]+1, yLim[1]-5, xLim[2]-1, yLim[1]-3, c(round(min(U[,
2]), 4), round(mean(U[,2]), 4), round(max(U[,2]), 4)),
color.scale(sort(U[,2]), c(0,1), 0.8, 1, color.spec="hsv"),
gradient="x")
```

## YOUR PART STARTS HERE

# Few words on assignment reports

- Return a PDF report and R script
  - Report contains your answers & discussion, explanation what you did, and necessary visualizations
    - Research report
  - R script contains contains all code necessary to repeat your analysis & findings
    - Use the provided R file

# Chapter 1

# **SVD, PCA & Pre- processing**

Part 2: Pre-processing and selecting the rank



# Pre-processing

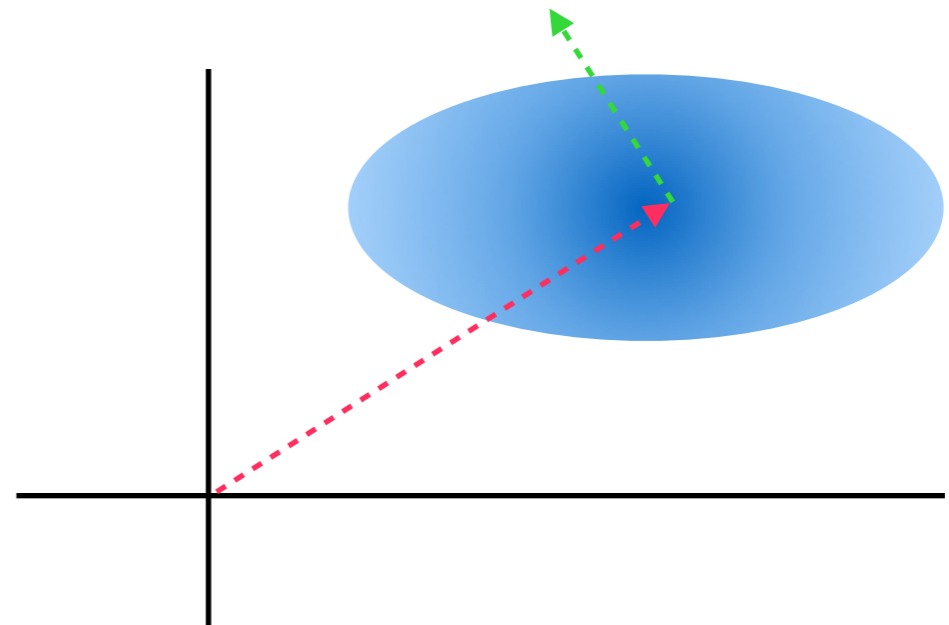


# Why pre-process?

- Consider matrix of weather data
  - Monthly temperatures in degrees Celcius
    - Typical range [-20, +25]
  - Monthly precipitation in millimeters
    - Typical range [0, 100]
- Precipitation seems much more important
  - But what if the temperatures where in degrees Kelvin?
    - The range is now [250, 300]

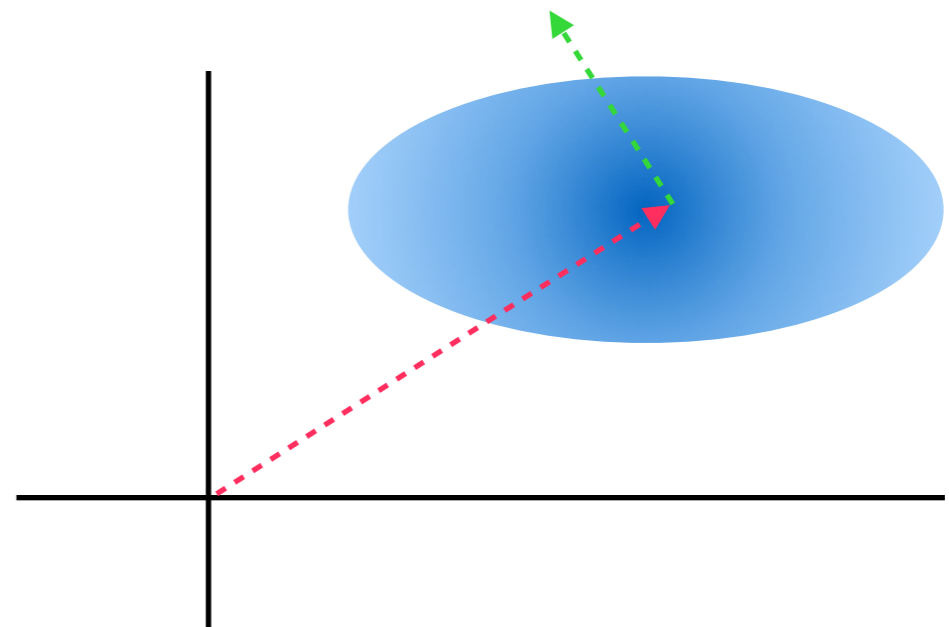
# Why pre-process

- If  $\mathbf{A}$  is nonnegative, the first singular vector just shows where the average of  $\mathbf{A}$  is
- The remaining vectors still have to be orthogonal to the first



# Why pre-process

- If  $\mathbf{A}$  is centered to the origin, the singular vectors show the directions of variance in  $\mathbf{A}$
- We will use this feature later...



# The z-scores

- The **z-scores** are attributes whose values are transformed by
  - centering them to 0 by removing the (column) mean from each value
  - normalizing the magnitudes by dividing every value with the (column) standard deviation

$$X' = \frac{X - \mu}{\sigma}$$

# When z-scores?

- Attribute values are approximately normally distributed, c.f.  $X' = \frac{X - \mu}{\sigma}$
- All attributes are equally important
- Data does not have any important structure that is destroyed
  - Non-negativity, sparsity, integer values, ...

# Other normalizations

- Large values can be reduced in importance by
  - taking logarithms (from positive values)
  - taking cubic roots
- Sparsity can be preserved by only considering non-zero values
- **The effects of normalization must always be considered**

# How many factors?

- Assume we want to compute rank- $k$  truncated SVD to analyze some data
- But how to select the  $k$ ?
  - Too big, and we have to handle unimportant factors
  - Too small, and we lose important structure
- So we need a way to select a good  $k$

# Guttman–Kaiser criterion and captured energy

- **Method 1:** select  $k$  s.t. for all  $i > k$ ,  $\sigma_i < 1$ 
  - Motivation: components with singular values  $< 1$  are uninteresting

- **Method 2:** select smallest  $k$  s.t.

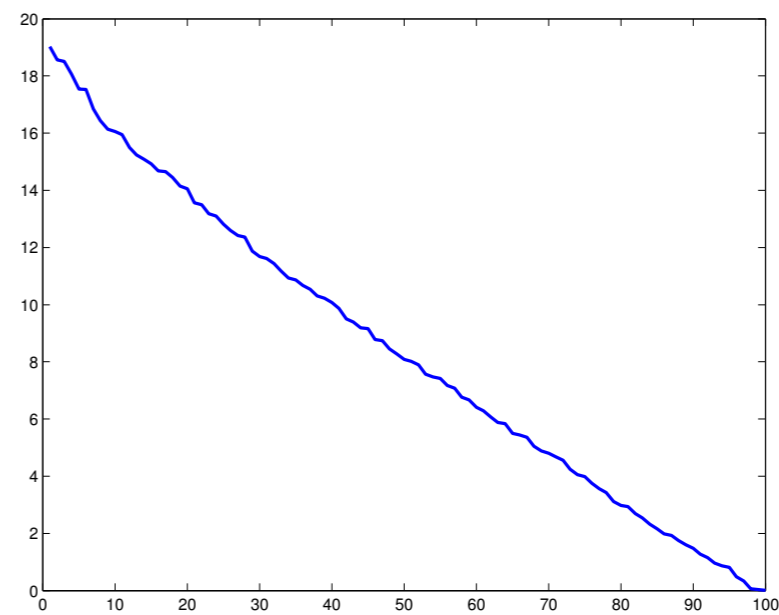
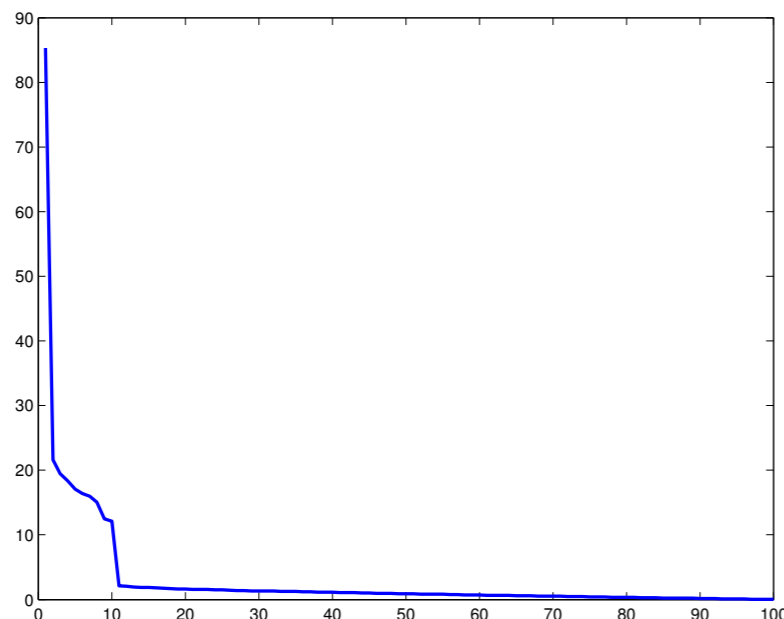
$$\sum_{i=1}^k \sigma_i^2 \geq 0.9 \sum_{i=1}^{\min\{n,m\}} \sigma_i^2$$

- Motivation: this explains 90% of the Frobenius norm (a.k.a. energy)
- Both methods are based on arbitrary thresholds



# Cattell's Scree test

- The **scree plot** has the singular values plotted in decreasing order
- In scree test, the rank is selected s.t. in the plot
  - there is a clear drop in the magnitudes; or
  - the singular values start to even out




# Entropy-based method

- The **relative contribution** of  $\sigma_k$  is  $r_k = \sigma_k^2 / \sum_i \sigma_i^2$

- The **entropy**  $E$  of singular values is

$$E = -\frac{1}{\log(\min\{n,m\})} \sum_{i=1}^{\min\{n,m\}} r_i \log r_i$$

$0 \cdot \infty = 0$



- Set the rank to the smallest  $k$  s.t.  $\sum_{i=1}^k r_i \geq E$
- Intuition: low entropy = the mass of the singular values is packed to the begin

# Random flip of signs

- Consider a random matrix  $\mathbf{A}'$  created by multiplying every element of  $\mathbf{A}$  by 1 or  $-1$  u.a.r.
  - The Frobenius norm doesn't change, but the spectral norm does change
  - How much the spectral norm changes depends on the amount of “structure” in  $\mathbf{A}$
- Idea: use this to select  $k$  that isolates the structure from the noise

# Using random flips

- The **residual matrix**  $\mathbf{A}_{-k}$  is

$$\mathbf{A}_{-k} = \mathbf{A} - \mathbf{A}_k = \mathbf{U}_{-k} \mathbf{\Sigma}_{-k} \mathbf{V}_{-k}^T$$

- $\mathbf{U}_{-k}$  ( $\mathbf{V}_{-k}$ ) contains the last  $n - k$  ( $m - k$ ) left (right) singular vectors
- Let  $\mathbf{A}_{-k}$  be the residual of  $\mathbf{A}$  and  $\mathbf{A}'_{-k}$  that of  $\mathbf{A}'$
- Select  $k$  s.t.  $|\|\mathbf{A}_{-k}\|_2 - \|\mathbf{A}'_{-k}\|_2| / \|\mathbf{A}_{-k}\|_F$  is small
  - On average, over multiple random matrices

# Issues with the methods

- Require computing the full SVD first or otherwise computationally heavy

Guttman–Kaiser

scree

entropy-based

random flips

- Require subjective evaluation

scree

random flips

- Based on arbitrary thresholds

Guttman–Kaiser

entropy-based

# Summary

- Pre-processing can make all the difference
  - Often overlooked
- Selecting the rank is non-trivial
  - Guttman–Kaiser and scree test are often used in other fields