Creative Data Mining

Lecture 03: Intro to RStudio and Clustering

7 March 2016

Danielle Griego, griegod@ethz.ch

Matthias Standfest, standfest@arch.ethz.ch

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

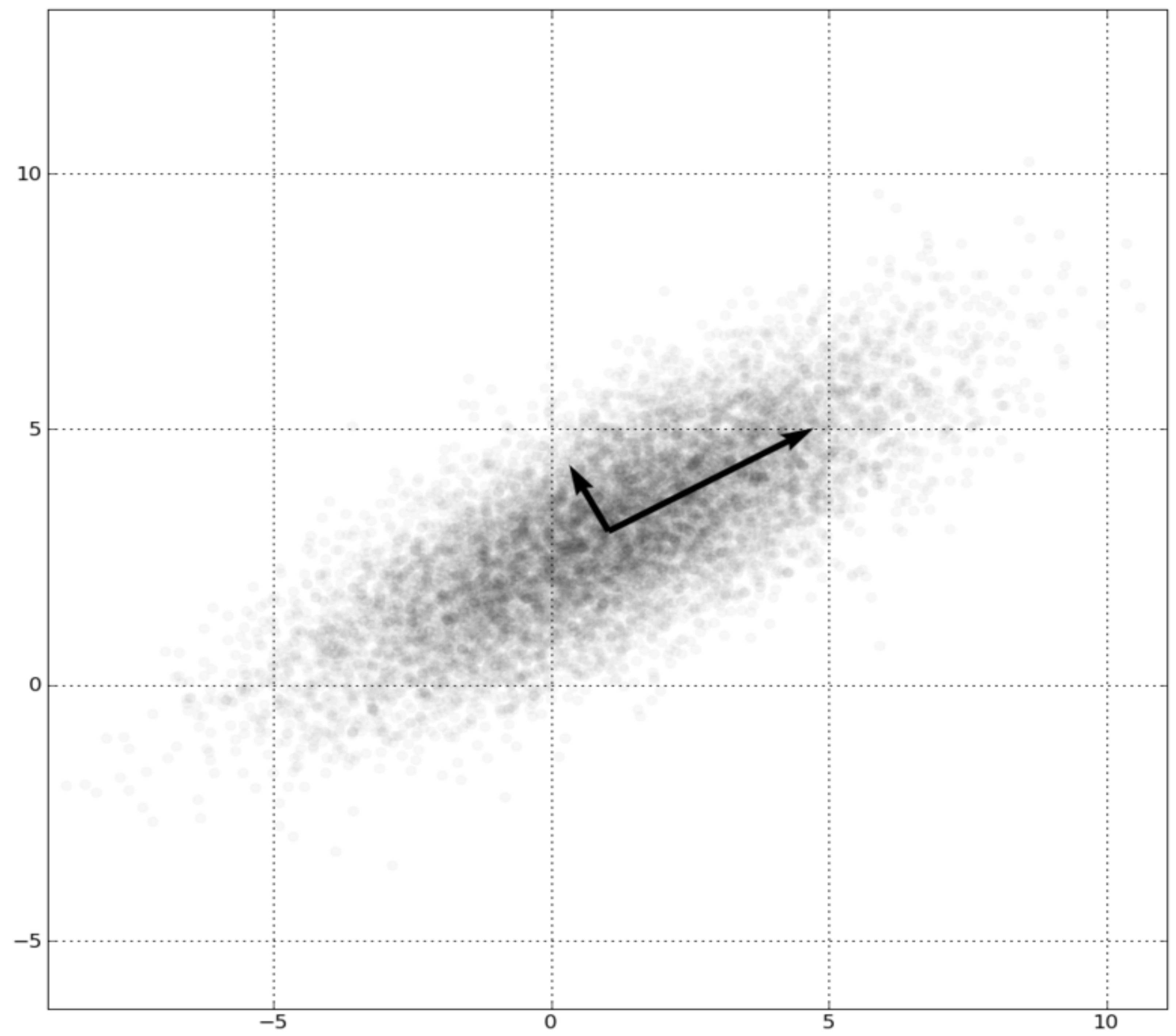
DARCH

Chair of Information Architecture

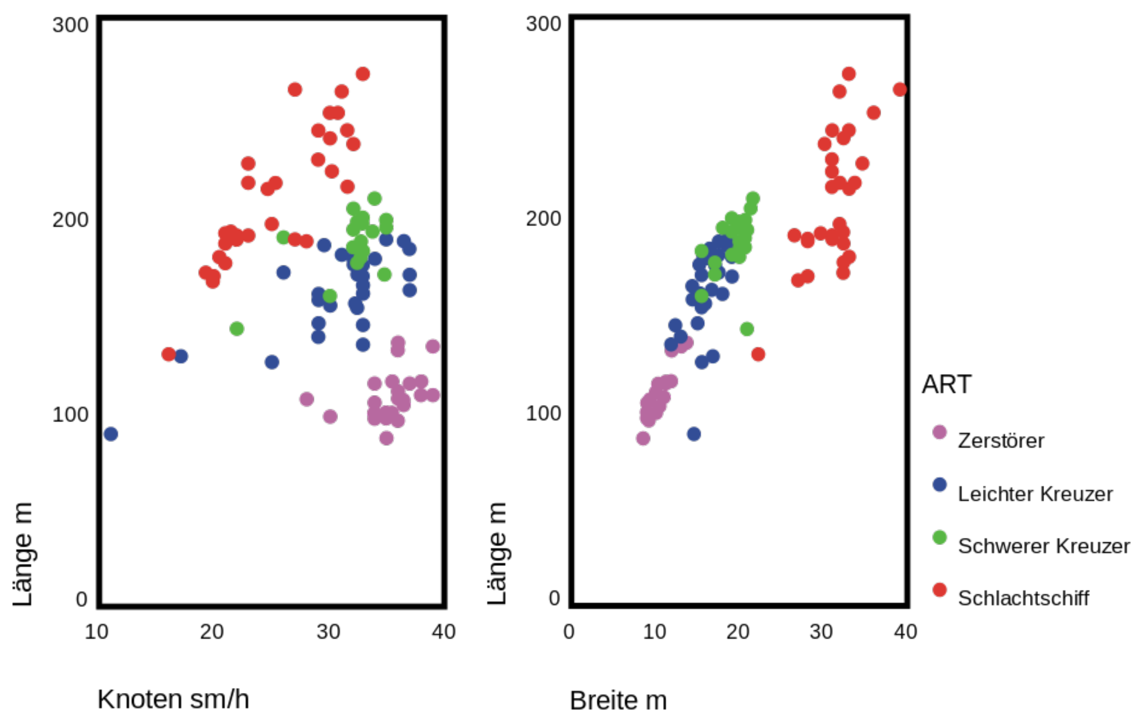
THE MATH UNDERNEATH: INTRODUCTION TO CLUSTERING

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich
DARCH
iA Chair of Information Architecture

PRIMARY COMPONENT ANALYSIS (PCA): MANY APPROACHES, WE USE SINGULAR VALUE DECOMPOSITION (SVD)

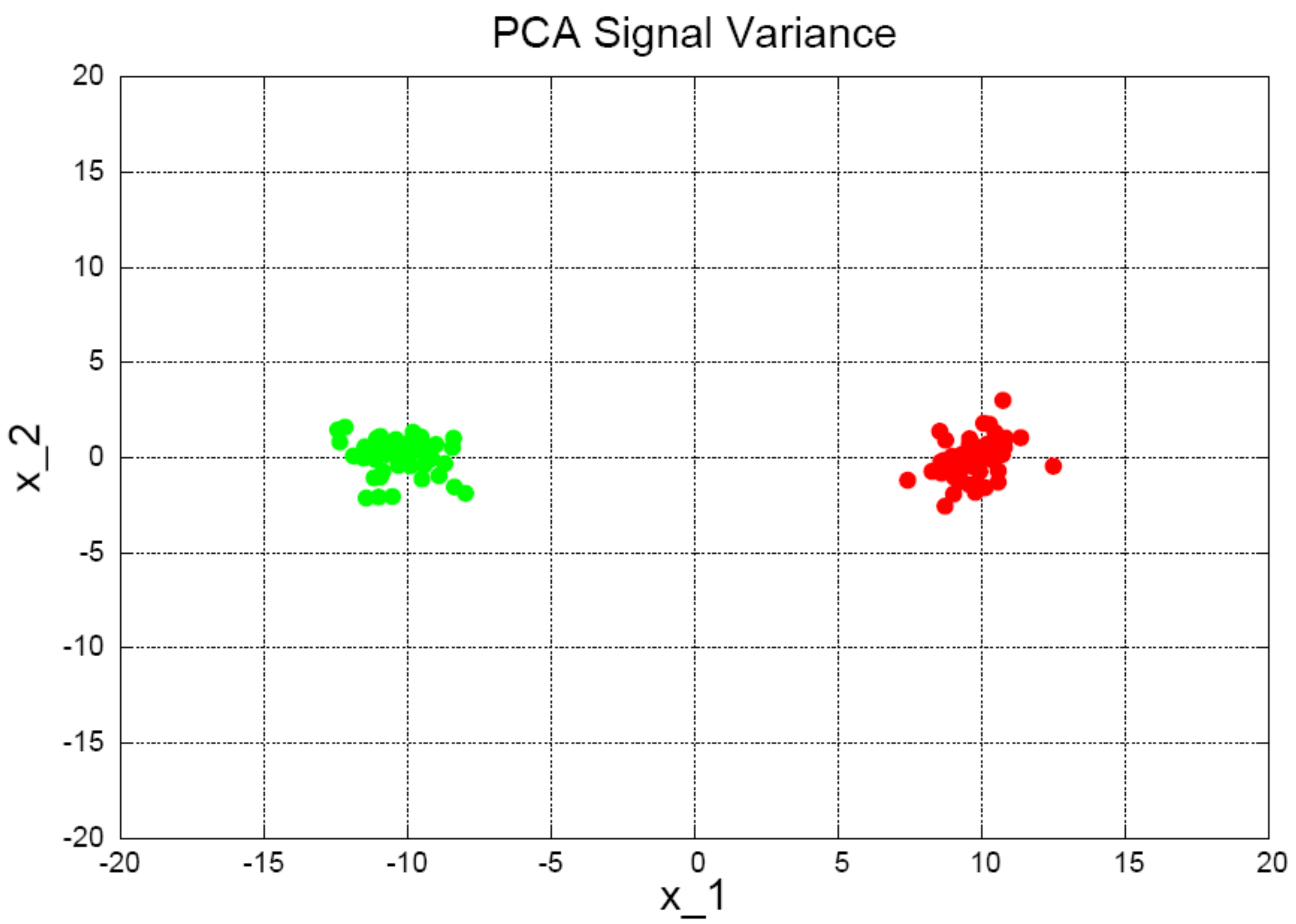


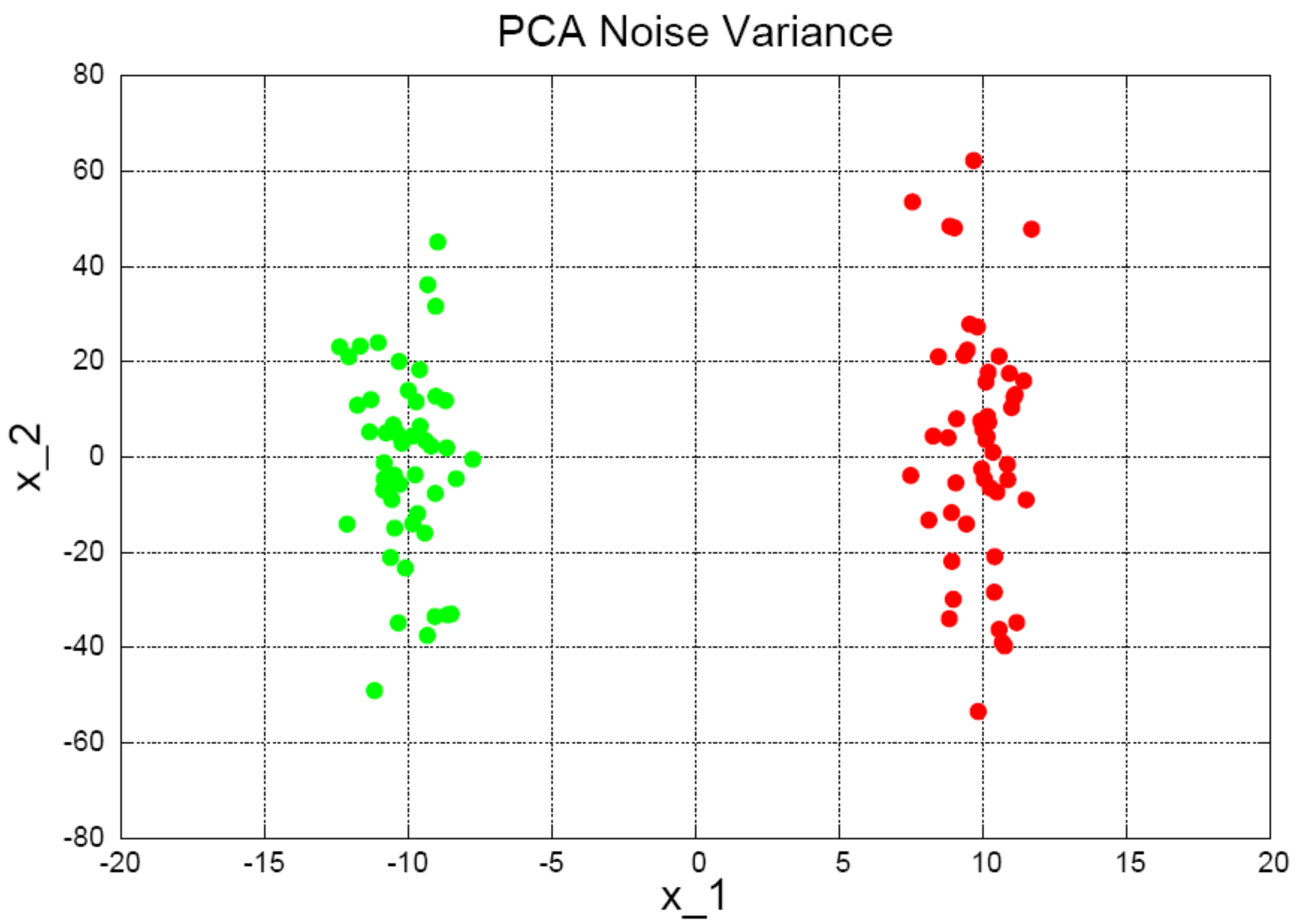
FIRST COMPONENT OF WAR SHIPS:



FACTOR	A	B	C
LENGTH	0,862	0,481	−0,159
WIDTH	0,977	0,083	0,198
SPEED	−0,679	0,730	0,082

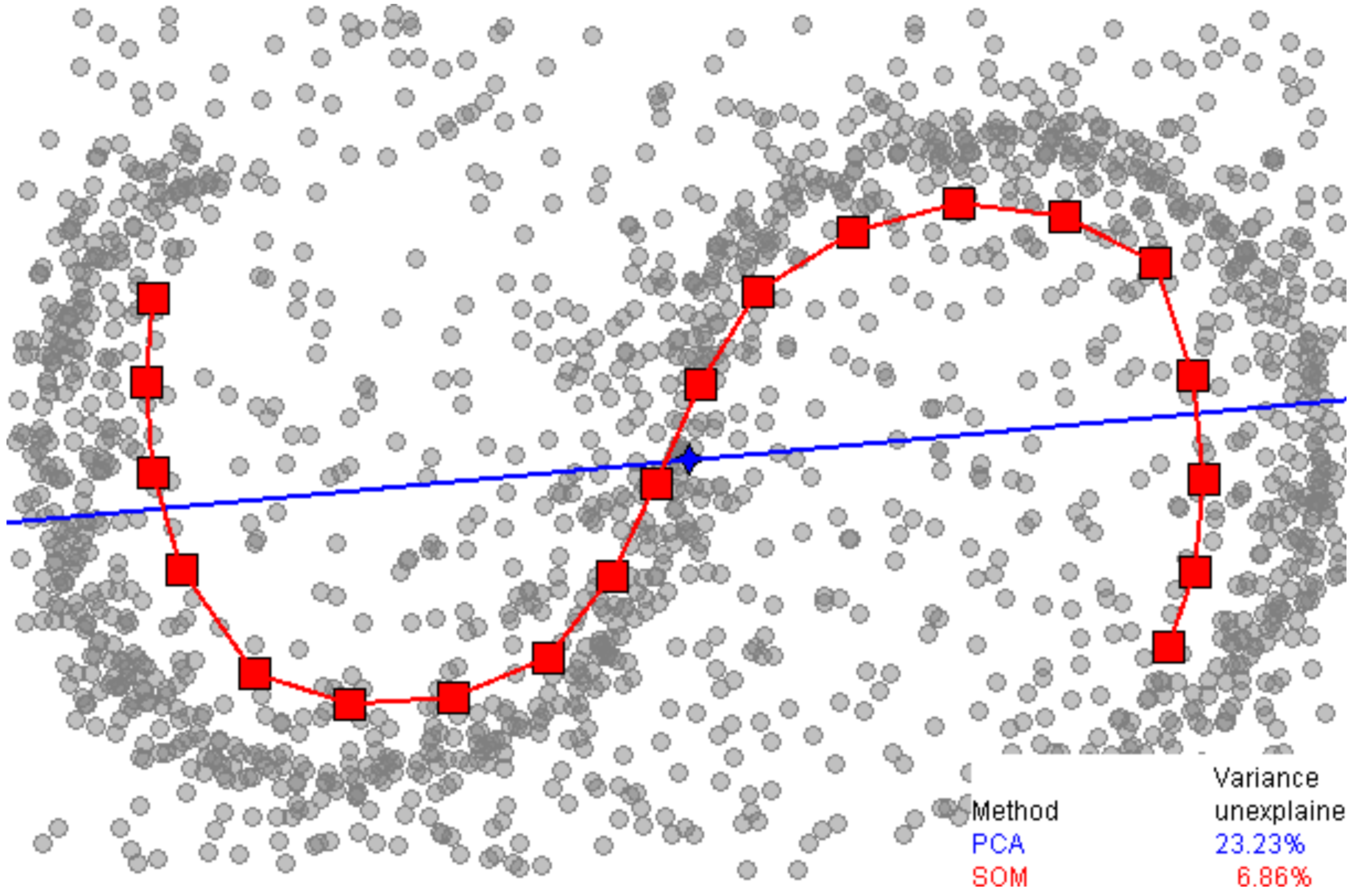
$$Y_A = 0.862 * LENGTH + 0.977 * WIDTH - 0.679 * SPEED$$


PCA Signal Variance
x_2
x_1
20
15
10
5
0
-5
-10
-15
-20
-20
-15
-10
-5
0
5
10
15
20

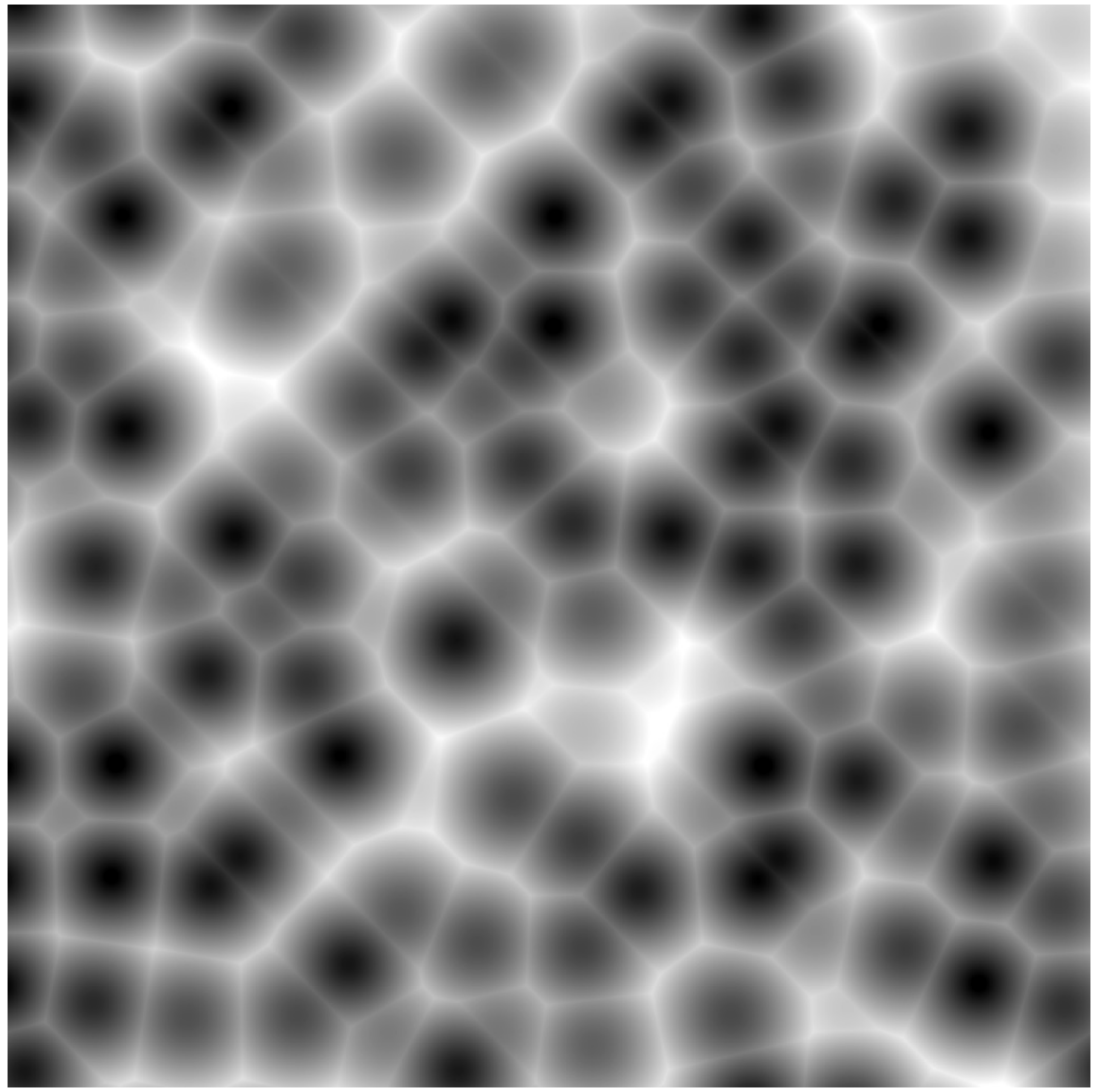

PCA Noise Variance
x_2
x_1
80
60
40
20
0
-20
-40
-60
-80
-20
-15
-10
-5
0
5
10
15
20

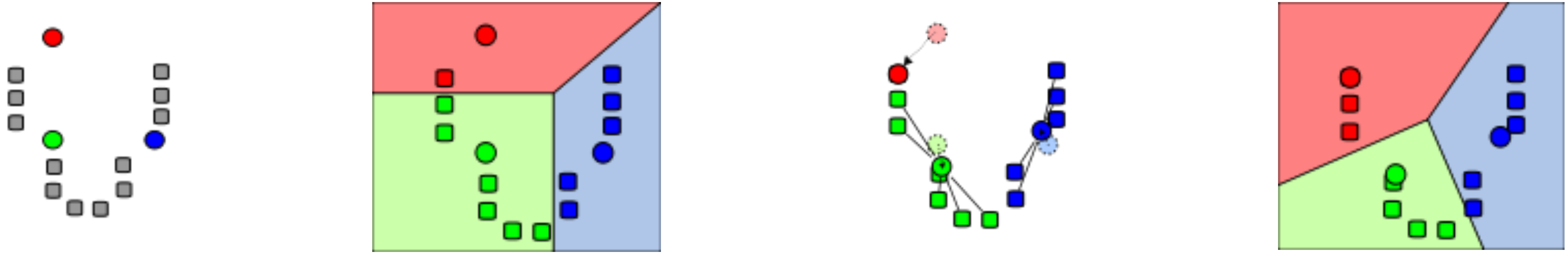
FOR FITTING
STRAIGHT LINES
WITH HIGHEST
POSSIBLE VARIANCE.

PROBLEM: LINEAR.



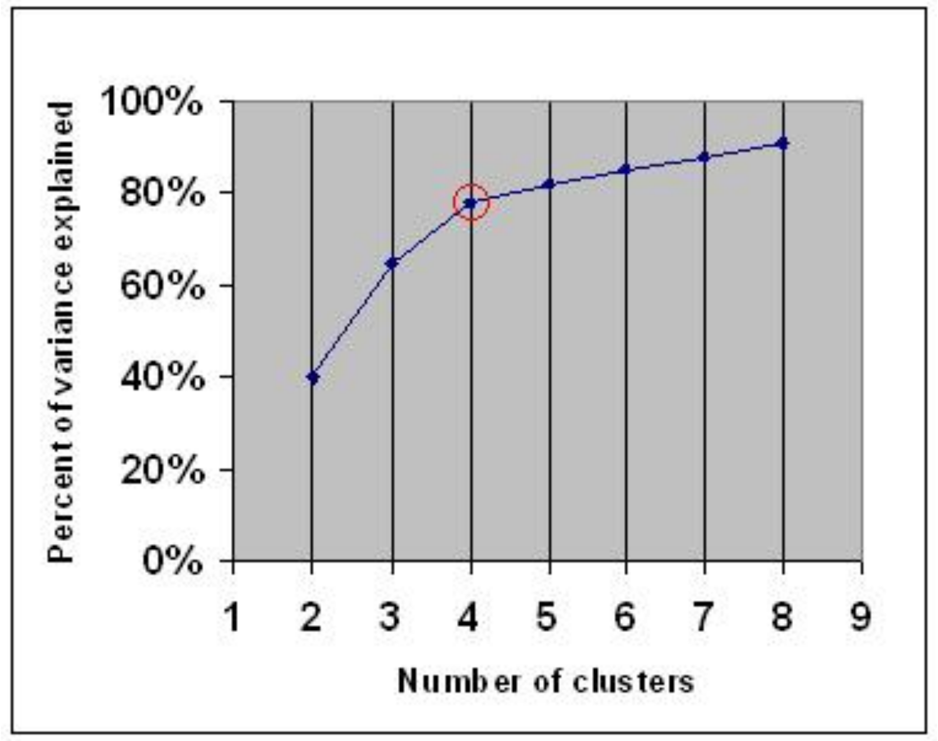
KMEANS: VORONOI CELLS FOR EACH CLUSTER



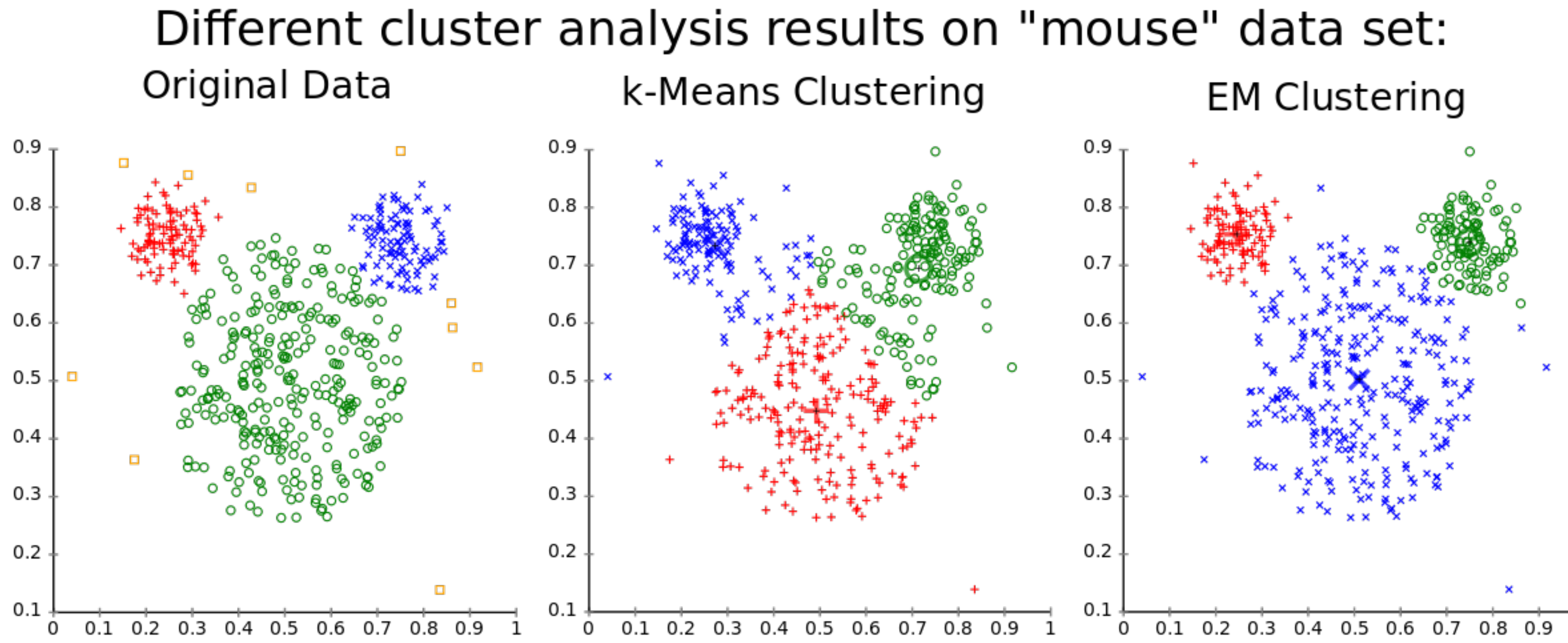


KMEANS:
NUMBER OF CLUSTERS NEEDS

TO BE ESTIMATED. ELBOW CURVE.



KMEANS PROBLEM: SAME SIZED CLUSTERS



FIRST K-MEANS CLUSTERING

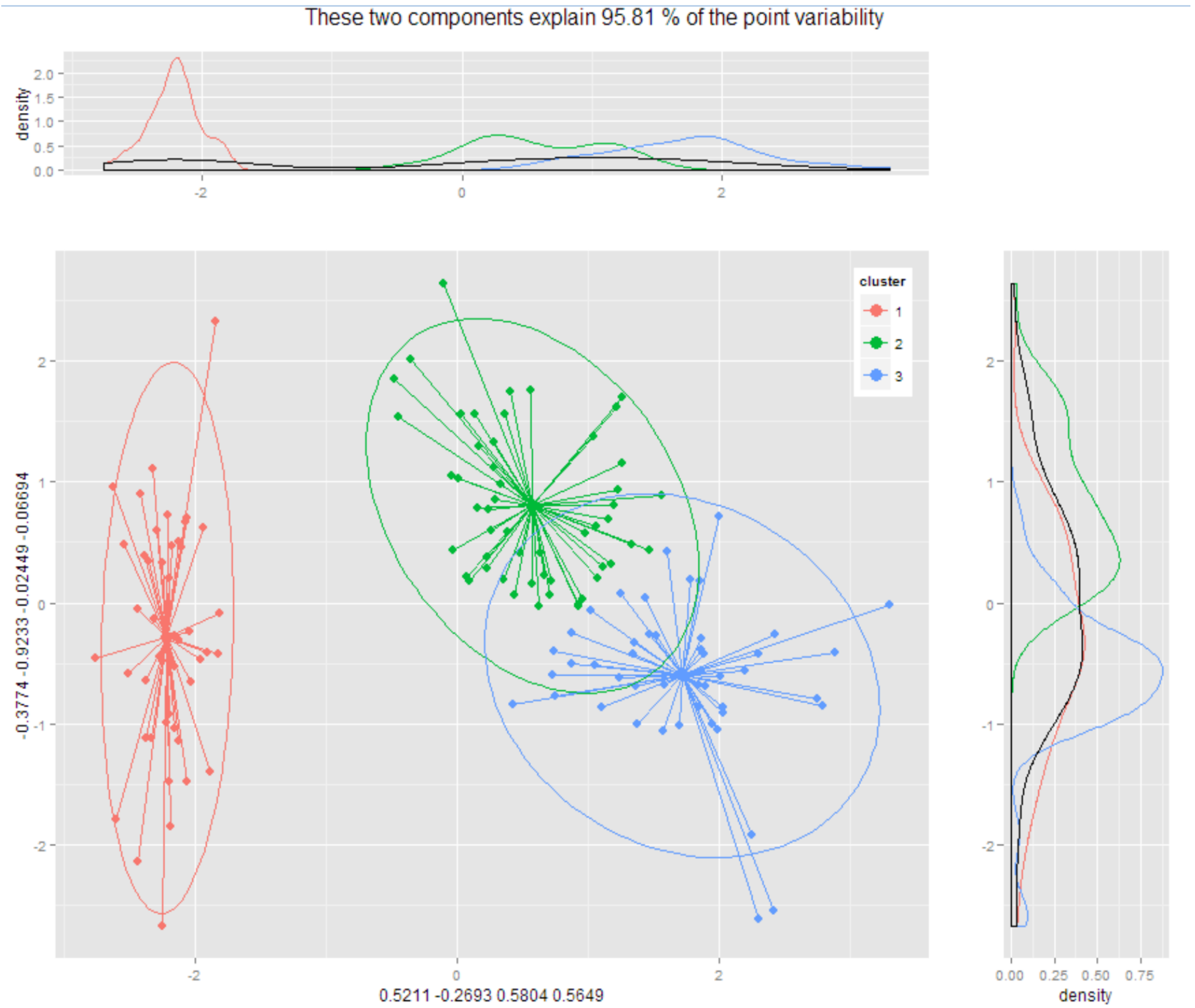
lecture_2_05.R

```r
#
# CREATIVE DATA MINING - FS 2016
# LECTURE 2 - R101
# Matthias Standfest
# Danielle Griego

## NOW, LET'S JUMP RIGHT IN AND MAKE YOUR FIRST K-MEANS ANALYSIS!

# 1. set your working directory
setwd("~/Dropbox/00_Work/01_Teaching/Creative Data Mining/001_FS15-DataMining/lecture_2/RSCRIPTS_FS16/")

# 2. get data
iris2 <- iris[,1:4]

# 3. Prepare data and inspect data
mydata <- na.omit(iris2) # listwise deletion of missing
summary(mydata) # look at the data
plot(mydata$Sepal.Length)

help(scale) # scale is generic function whose default method centers and/or scales the columns of a numeric matrix
mydata_s <- scale(mydata) # standardize variable
summary(mydata_s)
#plot(mydata_s$Sepal.Length) # note that this will not work, error will say that "$ operator is invalid for atomic vectors"
plot(mydata_s[,1]) # so need to look at lecture_2_04.R for other column references

# 4. Determine number of clusters
maxCluster <- round(sqrt(nrow(mydata_s)/2)*2) # max cluster estimation
set.seed(1) #set seed of random variable so we have the same results each time (to be able to compare)
library(NbClust) #import a library
result<-NbClust(mydata, diss=NULL, distance = "euclidean", min.nc=2, max.nc=maxCluster,
              method = "complete", index = "kl")
(CLUSTERESTIMATION <- result$Best.nc[1])

# 5. K-Means cluster analysis
set.seed(1)
fit <- kmeans(mydata, CLUSTERESTIMATION)

# 6.1. make plot using package cluster
library(cluster)
clusplot(mydata,fit$cluster)

# 6.2. make plot using package ade4
library(ade4)
pca <-prcomp(mydata, scale.=T, retx=T)  # principal components analysis
plot.mydata <- cbind(pca$x[,1], pca$x[,2]) # first and second PC
s.class(plot.mydata, factor(fit$cluster))
```

DIFFERENT VISUALISATION
lecture_2_06.R

```r
#
# CREATIVE DATA MINING - FS 2016
# LECTURE 2 - R101
# Matthias Standfest
# Danielle Griego

## NOW LET'S IMPROVE THE VISUALIZATION FROM THE PREVIOUS ANALYSIS

# 1. set your working directory
setwd("~/Dropbox/00_Work/01_Teaching/Creative Data Mining/001_FS15-DataMining/lecture_2/RSCRIPTS_FS16/")

# 2. get data
iris2 <- iris[,1:4]

# 3. prepare data
mydata <- na.omit(iris2) # listwise deletion of missing
summary(mydata)
mydata <- scale(mydata) # standardize variables
summary(mydata)

# 4. Determine number of clusters
maxCluster <- round(sqrt(nrow(mydata)/2)*2) # max cluster estimation
set.seed(1) #set seed of random variable so we have the same results each time (to be able to compare)
library(NbClust) #import a library
result<-NbClust(mydata, diss=NULL, distance = "euclidean", min.nc=2, max.nc=maxCluster,
                method = "complete", index = "kl")
(CLUSTERESTIMATION <- result$Best.nc[1])

# 5. K-Means cluster analysis
set.seed(1)
fit <- kmeans(mydata, CLUSTERESTIMATION)

# 6. plot clustering using GGPLOT2
# Cluster Plot against 1st 2 principal components
# vary parameters for most readable graph
# ggplot solution for clusplot(mydata, fit$cluster, color=FALSE, shade=TRUE,labels=2, lines=0)
pca     <-prcomp(mydata, scale.=T, retx=T)  # principal components analysis
# gg: data frame of PC1 and PC2 scores with corresponding cluster
gg <- data.frame(cluster=factor(fit$cluster), x=pca$x[,1], y=pca$x[,2])
# calculate cluster centroid locations
centroids <- aggregate(cbind(x,y)~cluster,data=gg,mean)
# merge centroid locations into ggplot dataframe
gg <- merge(gg,centroids,by="cluster",suffixes=c("",".centroid"))
# calculate 95% confidence ellipses
library(ellipse)
conf.rgn  <- do.call(rbind,lapply(1:3,function(i)
  cbind(cluster=i,ellipse(cov(gg[gg$cluster==i,2:3]),centre=as.matrix(centroids[i,2:3])))))
conf.rgn  <- data.frame(conf.rgn)
conf.rgn$cluster <- factor(conf.rgn$cluster)
# plot cluster map
library(ggplot2)

cumulativeVariability <- (cumsum((pca$sdev)^2) / sum(pca$sdev^2))[2] #cumulativeVariability might return a characte
class(cumulativeVariability)
#cumulativeVariability <- as.numeric((cumsum((pca$sdev)^2) / sum(pca$sdev^2))[2]) # so might need to use as.numberic

labelx <- paste(formatC(pca$rotation[,1], width=5), collapse = ' ')
labely <- paste(formatC(pca$rotation[,2], width=5), collapse = ' ')
clusterscatter <- ggplot(gg, aes(x,y, color=cluster))+
  geom_point(size=3) +
  geom_point(data=centroids, size=4) +
  geom_segment(aes(x=x.centroid, y=y.centroid, xend=x, yend=y))+
  geom_path(data=conf.rgn)+
  ylab(labely)+
  xlab(labelx)+
  theme(legend.position=c(1,1),legend.justification=c(1,1))
plot(clusterscatter) # plot nices clusters with ellipses using ggplot2

# 7. add distribution clusterwise to the primary component axes
# http://www.r-bloggers.com/ggplot2-cheatsheet-for-visualizing-distributions/ #good link!
plot_top <- ggplot(gg, aes(x=x, col=cluster)) +
  geom_density(alpha=.5) +
  geom_density(color="black") +
  theme(legend.position = "none", axis.title.x = element_blank())
plot_right <- ggplot(gg, aes(x=y, col=cluster)) +
  geom_density(alpha=.5) +
  geom_density(color="black") +
  coord_flip() +
  theme(legend.position = "none", axis.title.y = element_blank())
#placeholder plot - prints nothing at all
empty <- ggplot()+geom_point(aes(1,1), colour="white") +
  theme(
    plot.background = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    axis.text.x = element_blank(),
    axis.text.y = element_blank(),
    axis.ticks = element_blank()
  )
#arrange the plots together, with appropriate height and width for each row and column
library(gridExtra)
library(grid)
chart1 <- grid.arrange(plot_top, empty, clusterscatter, plot_right, ncol=2, nrow=2, widths=c(4, 1), heights=c(1, 4),
                       top=paste("These two components explain", formatC(cumulativeVariability*100, width=4), "% of the point variability"))
```

DIFFERENT VISUALISATION
lecture_2_06.R

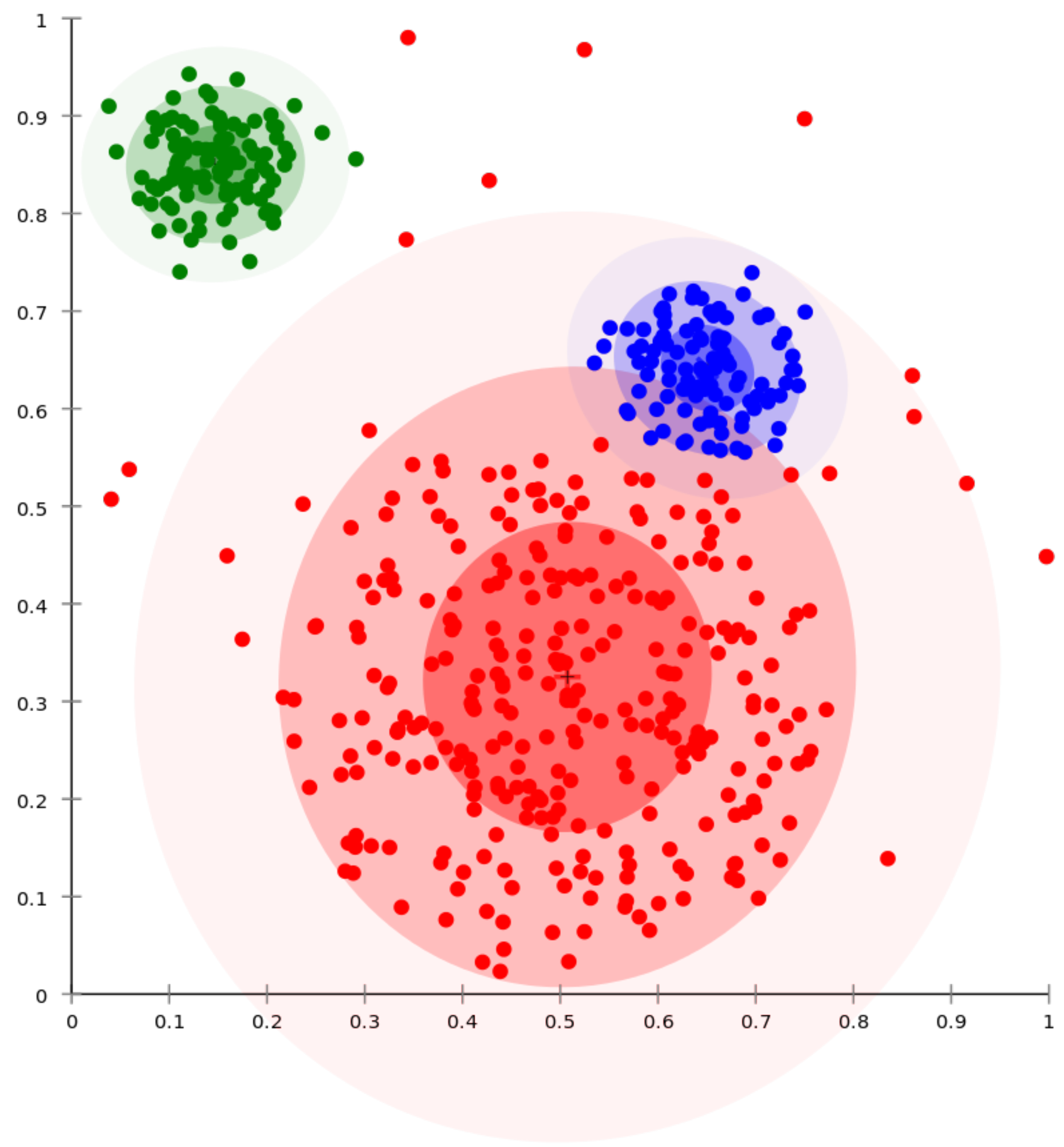


codingStyle – StyleTheCode() *#TODO(you):restyle*

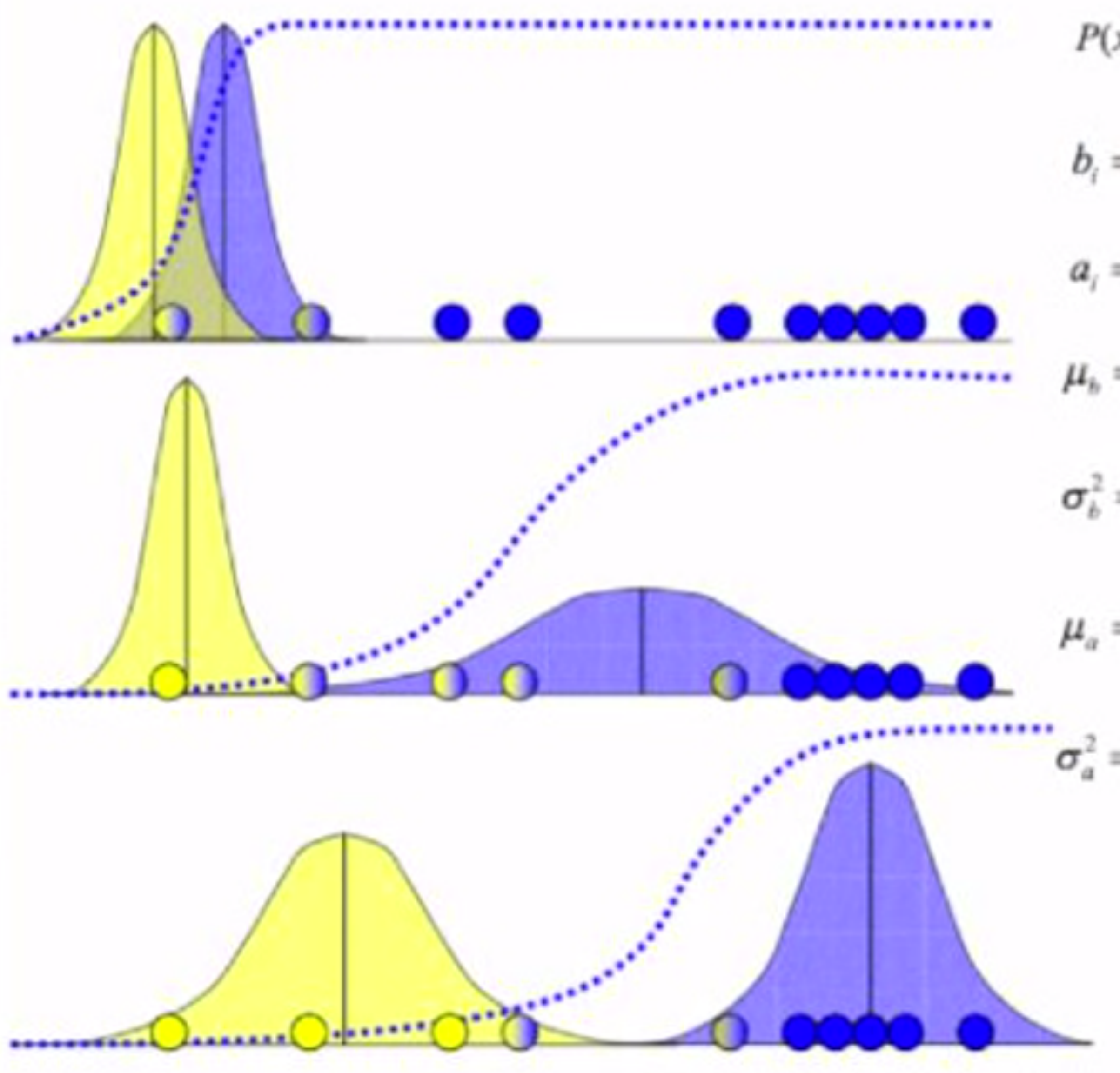
- https://google.github.io/styleguide/Rguide.xml
- kApple, vectorBanana, matrixCherry, scalarDurian
- GatherElderBerries(100)
- https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/00Index.html
- str(faithful) *# mtcars, rock, trees, LifeCycleSavings*
- *#vectorized instead of for, boolean tables as filters instead of if clause*

OTHER CLUSTERING TECHNIQUES

EXPECTATION MAXIMIZATION ALGORITHM WORKS WITH DISTRIBUTION



EM: 1-d example



$$P(x_i \mid b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{(x_i-\mu_b)^2}{2\sigma_b^2}\right)$$

$$b_i = P(b \mid x_i) = \frac{P(x_i \mid b)P(b)}{P(x_i \mid b)P(b) + P(x_i \mid a)P(a)}$$

$$a_i = P(a \mid x_i) = 1 - b_i$$

$$\mu_b = \frac{b_1x_1 + b_2x_2 + \ldots + b_nx_{n_b}}{b_1 + b_2 + \ldots + b_n}$$

$$\sigma_b^2 = \frac{b_1(x_1-\mu_1)^2 + \ldots + b_n(x_n-\mu_n)^2}{b_1 + b_2 + \ldots + b_n}$$

$$\mu_a = \frac{a_1x_1 + a_2x_2 + \ldots + a_nx_{n_b}}{a_1 + a_2 + \ldots + a_n}$$

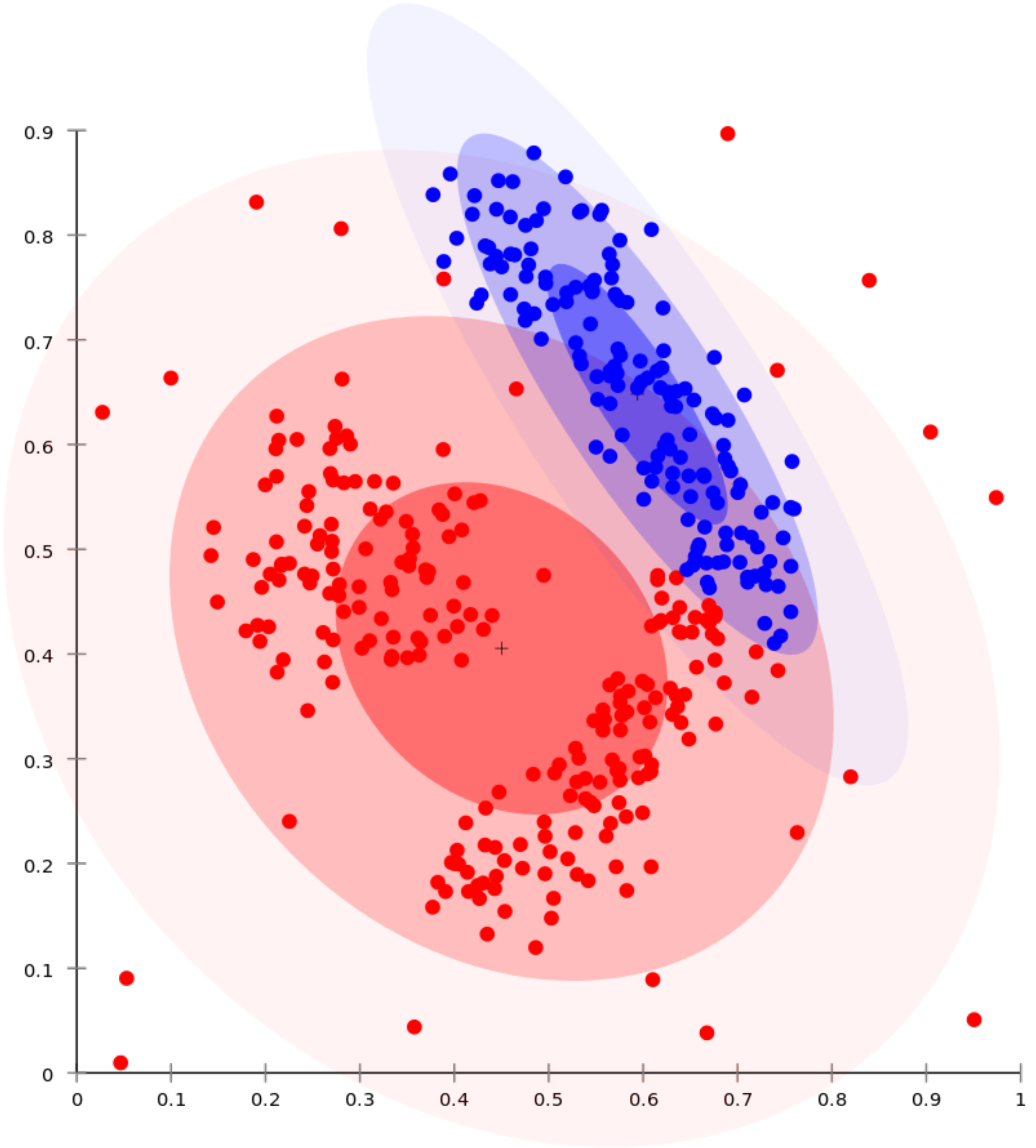
$$\sigma_a^2 = \frac{a_1(x_1-\mu_1)^2 + \ldots + a_n(x_n-\mu_n)^2}{a_1 + a_2 + \ldots + a_n}$$

could also estimate priors:

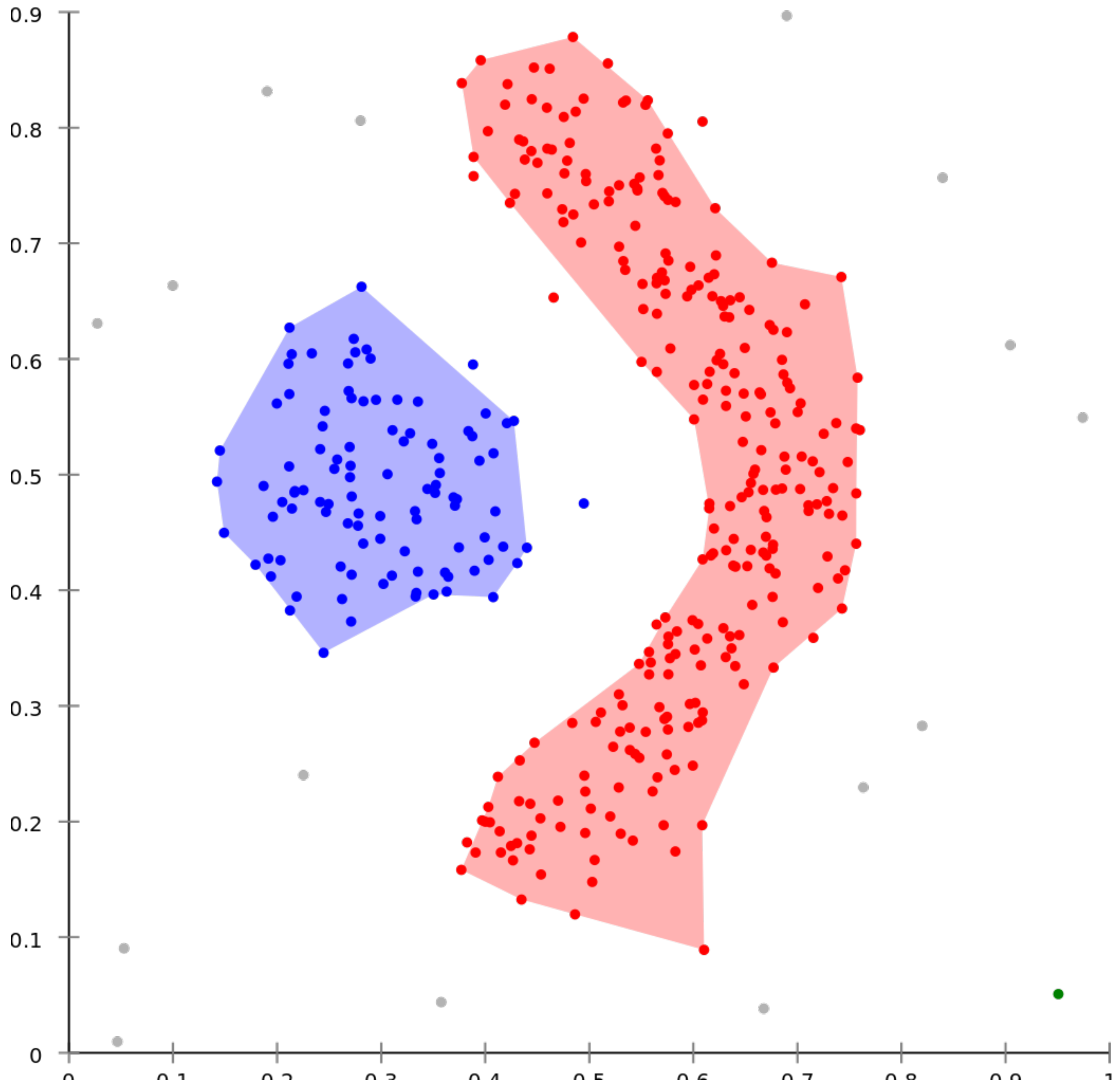
$P(b) = (b_1 + b_2 + \ldots b_n) / n$

$P(a) = 1 - P(b)$

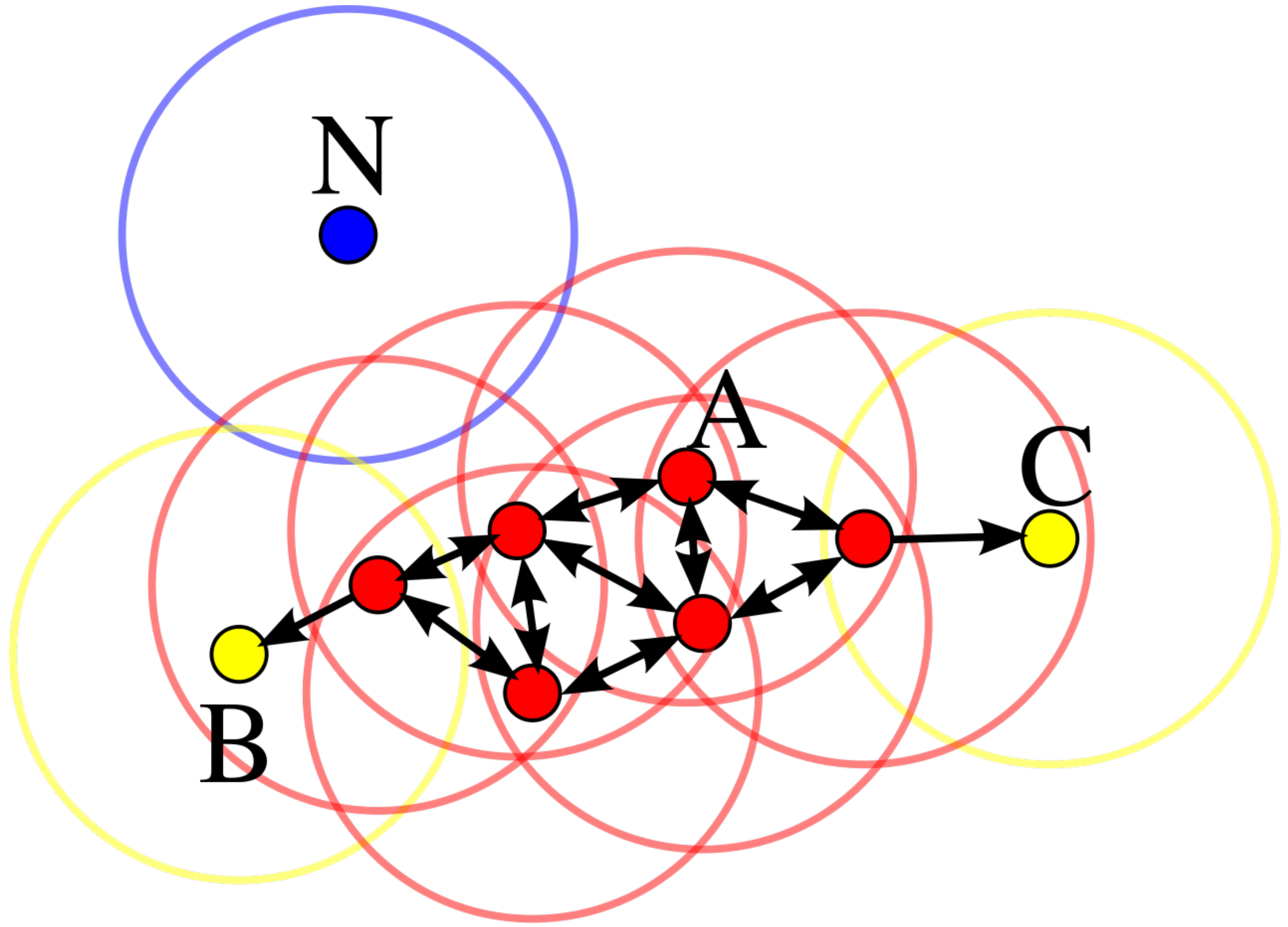
EM ALGO.: PROBLEM DENSITY CLUSTERS



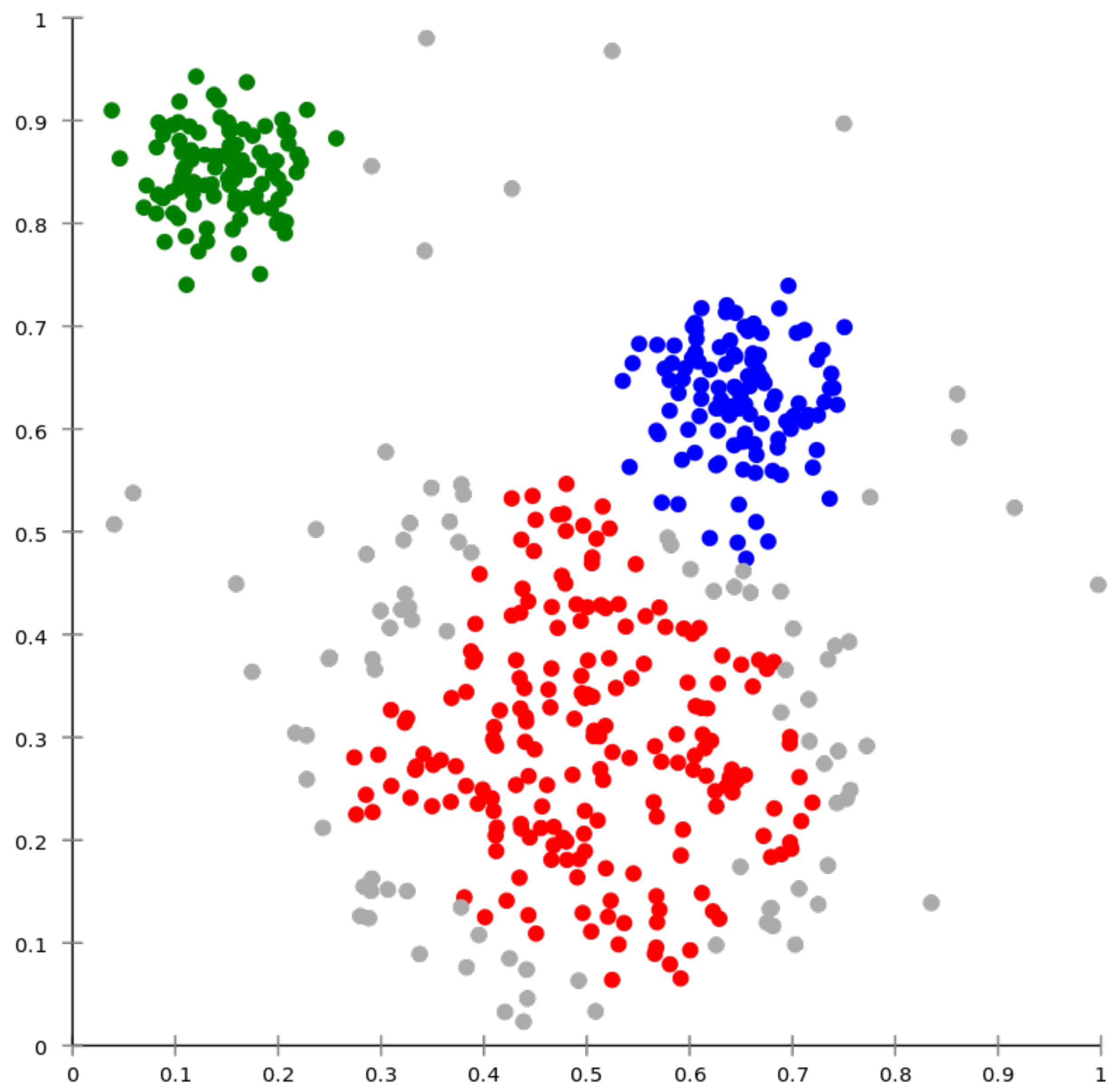
DBSCAN
DENSITY-BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE



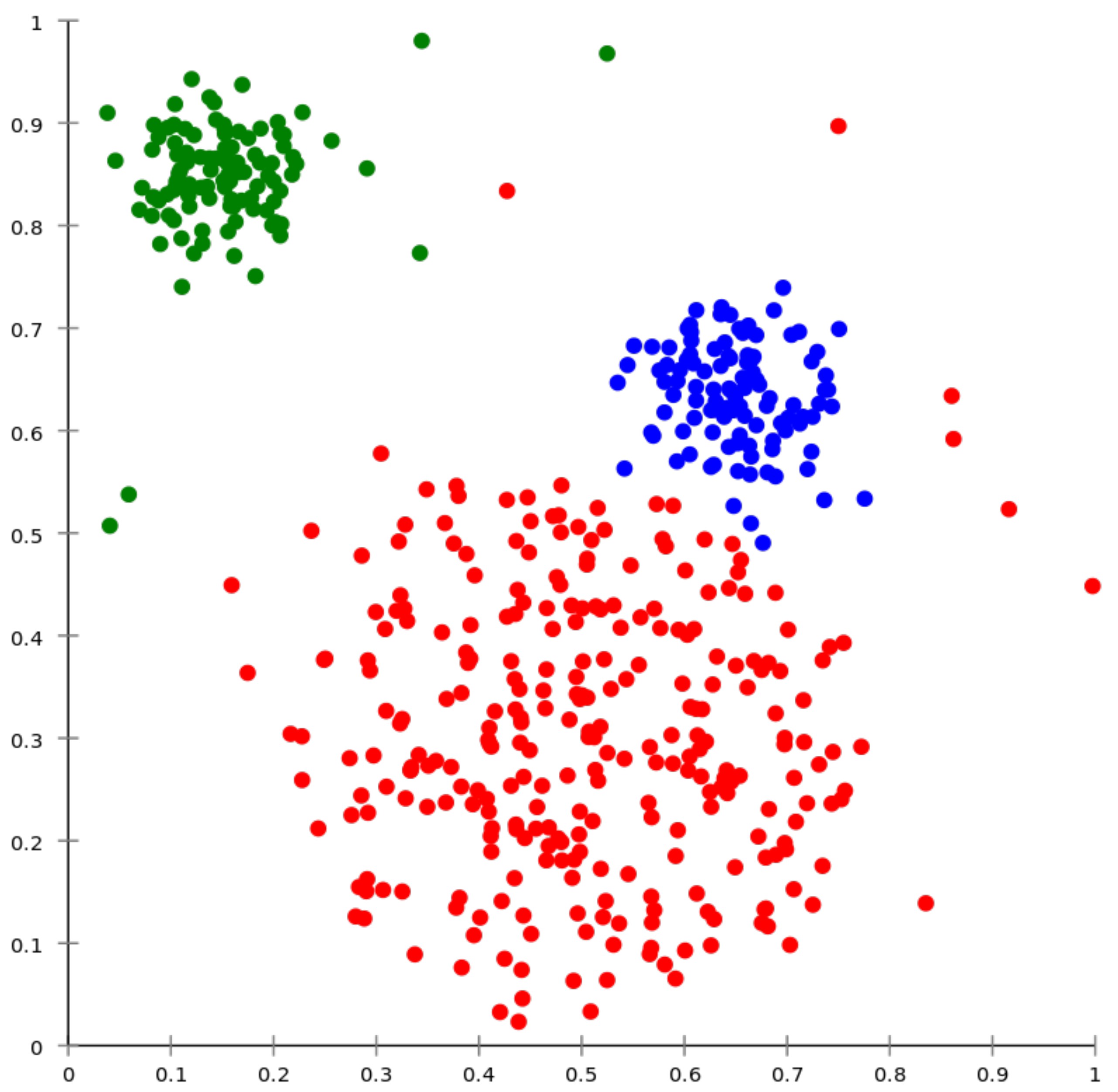
IN THIS DIAGRAM, MINPTS = 3. A AND THE OTHER RED POINTS ARE CORE POINTS, BECAUSE AT LEAST THREE POINTS SURROUND IT IN AN *R* RADIUS. BECAUSE THEY ARE ALL REACHABLE FROM ONE ANOTHER, THEY FORM A SINGLE CLUSTER. POINTS B AND C ARE NOT CORE POINTS, BUT ARE REACHABLE FROM A (VIA OTHER CORE POINTS) AND THUS BELONG TO THE CLUSTER AS WELL. POINT N IS A NOISE POINT THAT IS NEITHER A CORE POINT NOR DENSITY-REACHABLE.



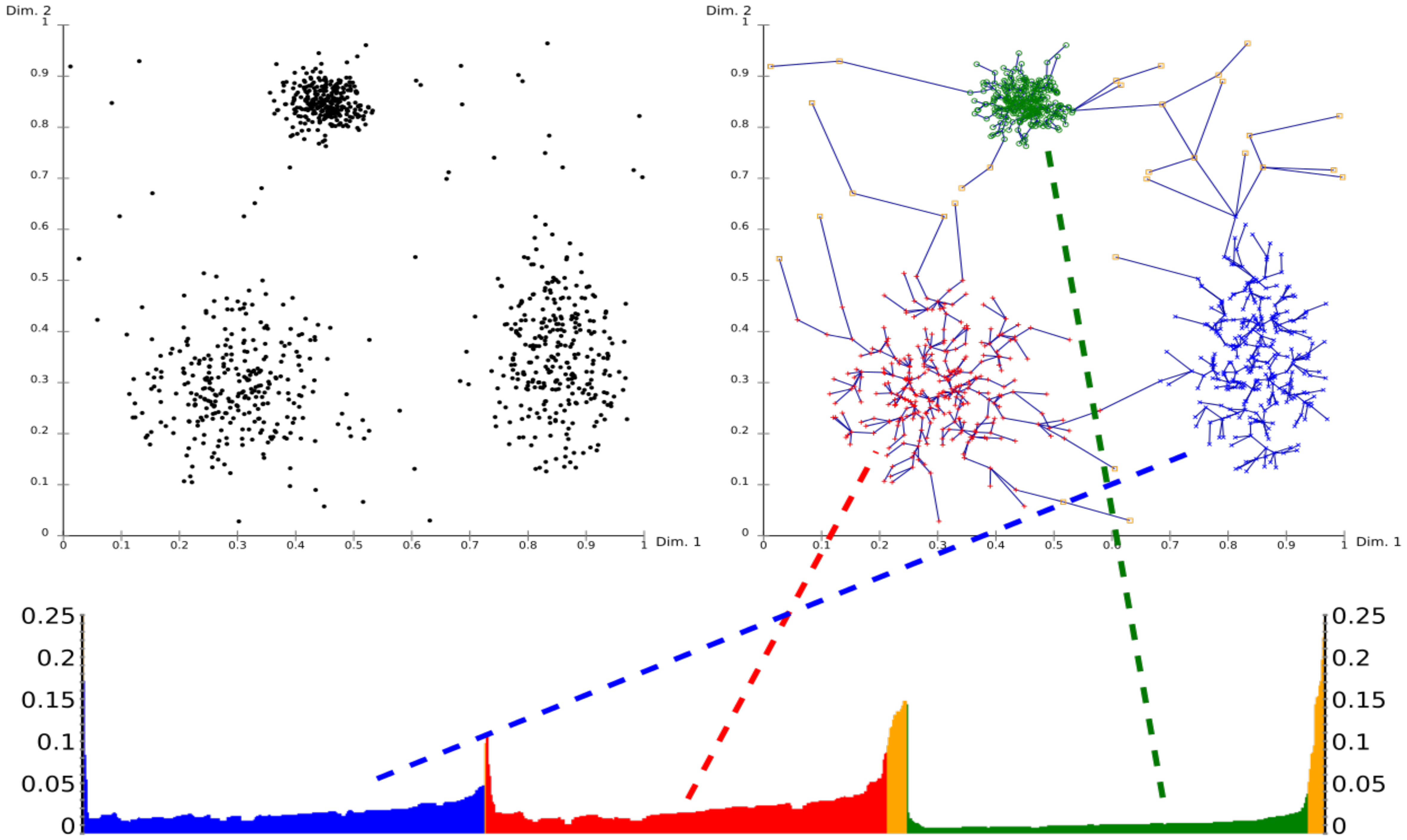
DBSCAN
PROBLEMS WITH VARYING DENSITY



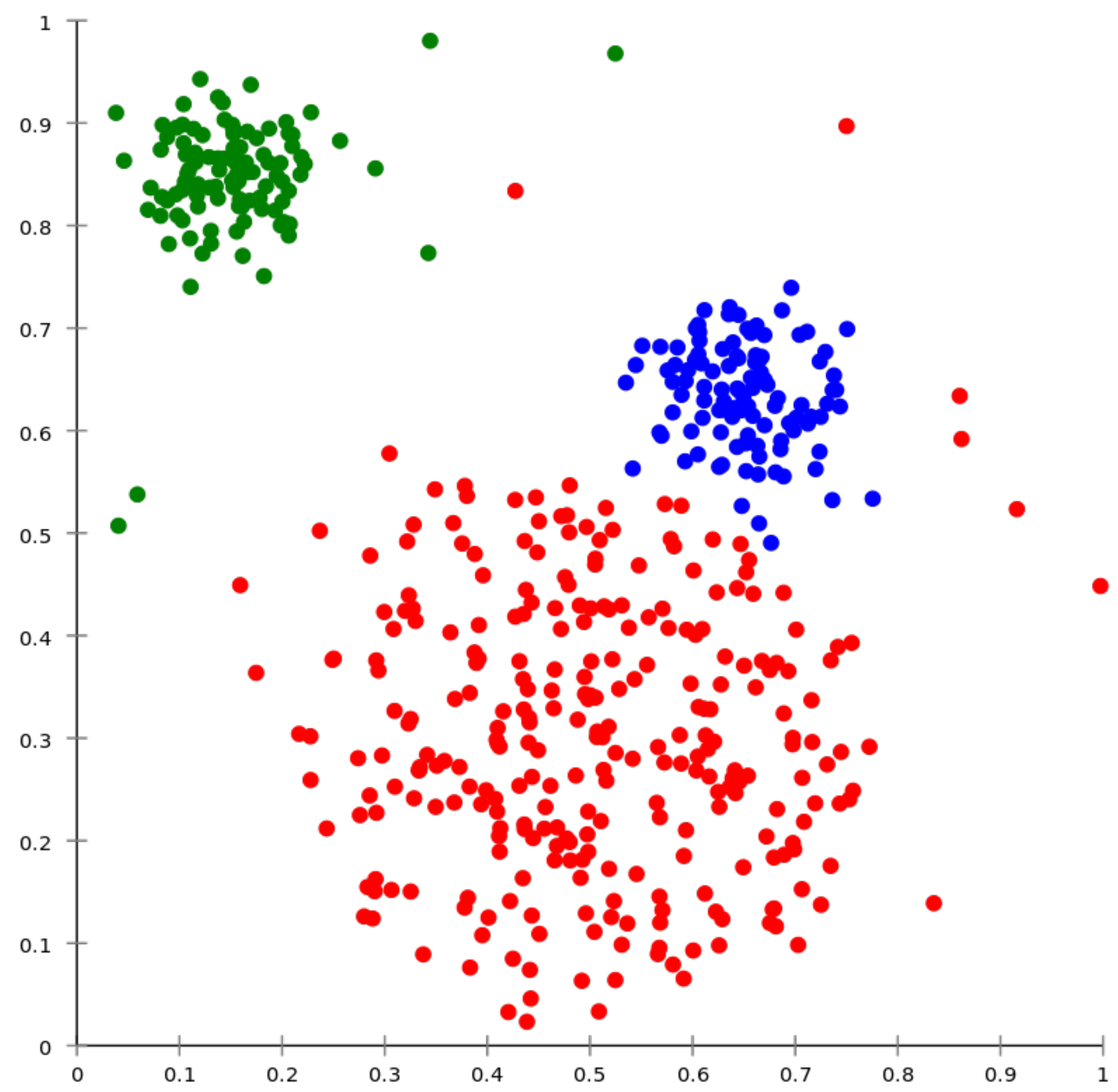
OPTICS
ORDERING POINTS TO IDENTIFY THE CLUSTERING STRUCTURE



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich
DARCH
iA Chair of Information Architecture

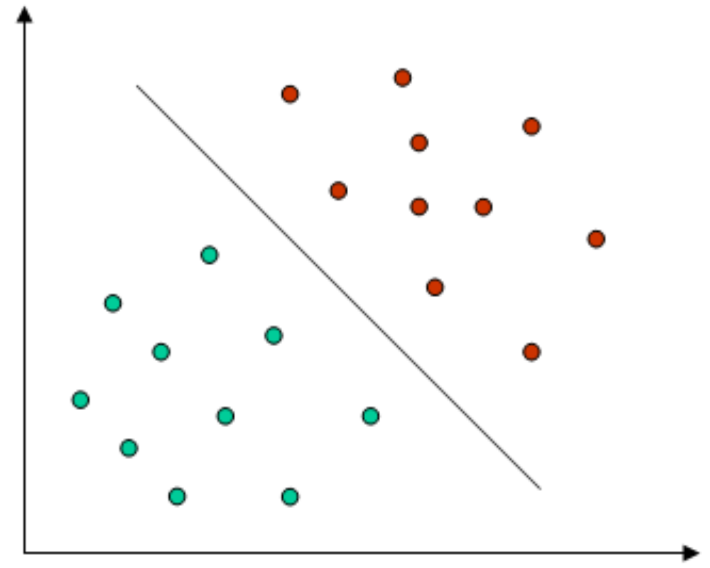

Dim. 2
Dim. 1
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Dim. 2
Dim. 1
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0.25
0.2
0.15
0.1
0.05
0

OPTICS
PROBLEM WITH LINEARITY OF PARTITION

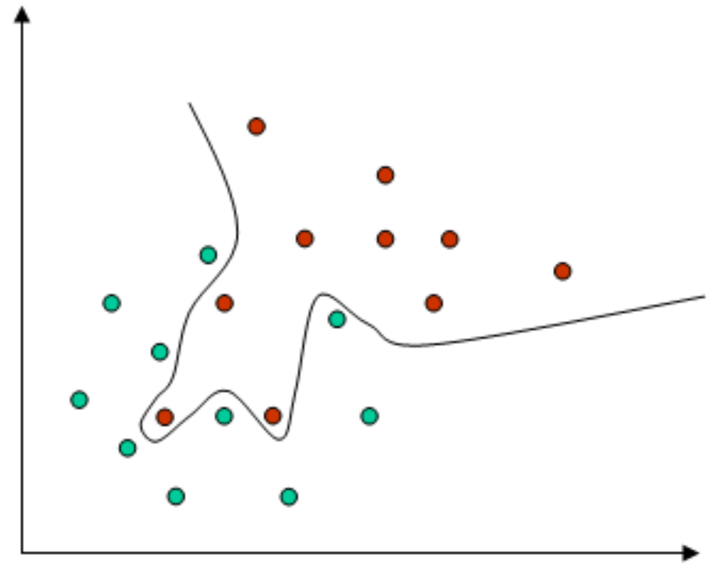


SUPPORT VECTOR MACHINE (SVM)

MAXIMIZE BORDERS

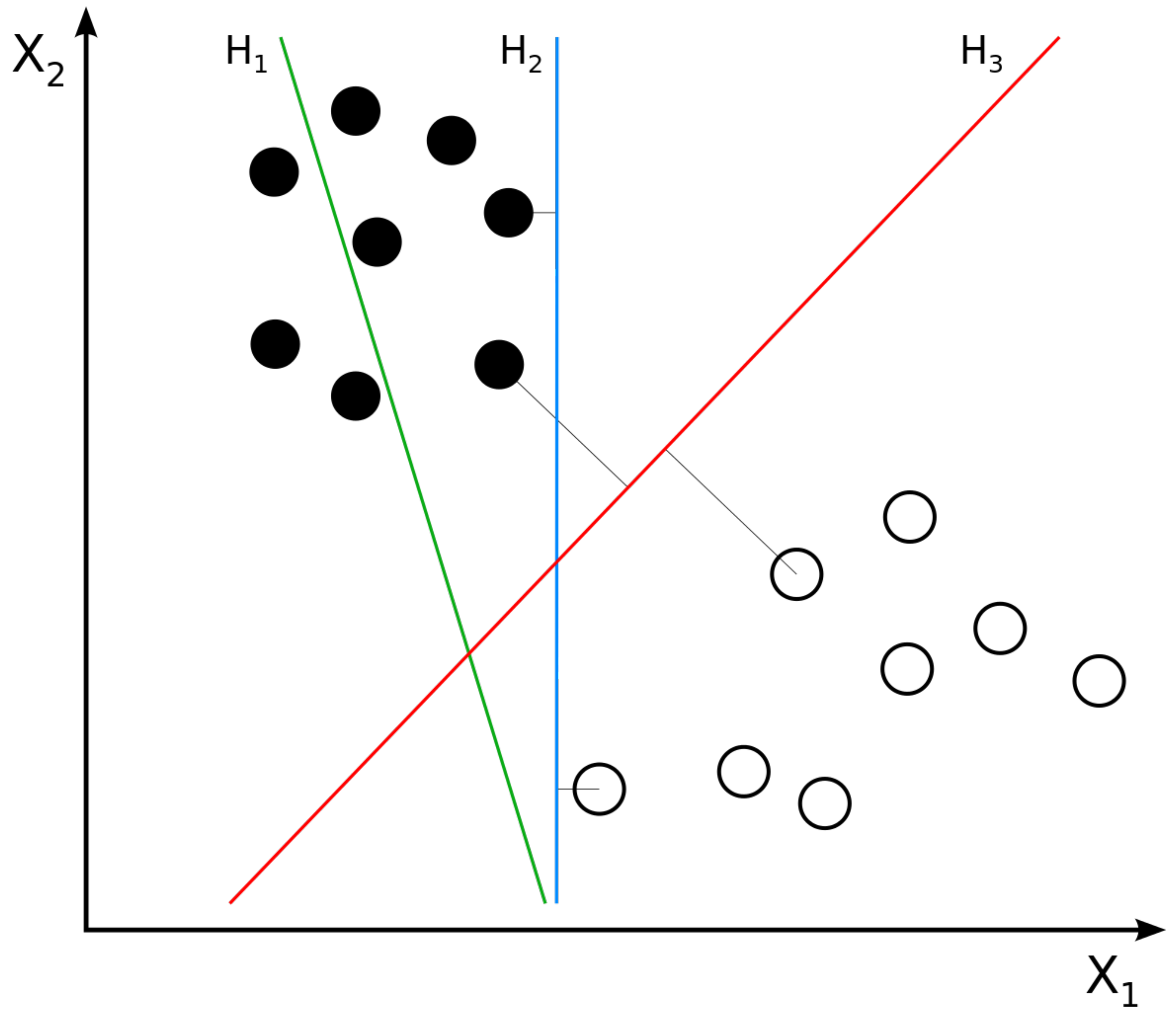


linear trennbar

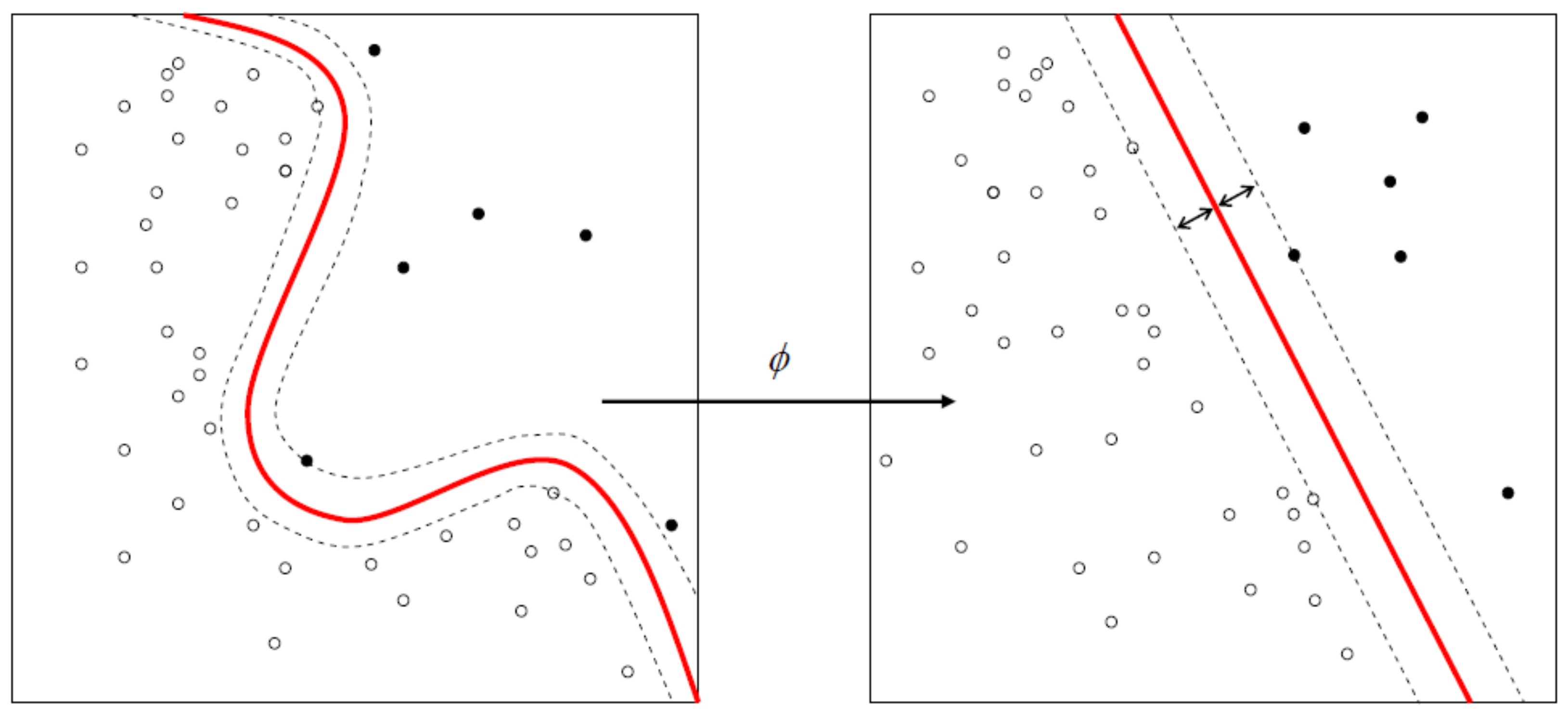


nicht linear trennbar

MAXIMIZE DISTANCE TO SEPARATING HYPERPLANES

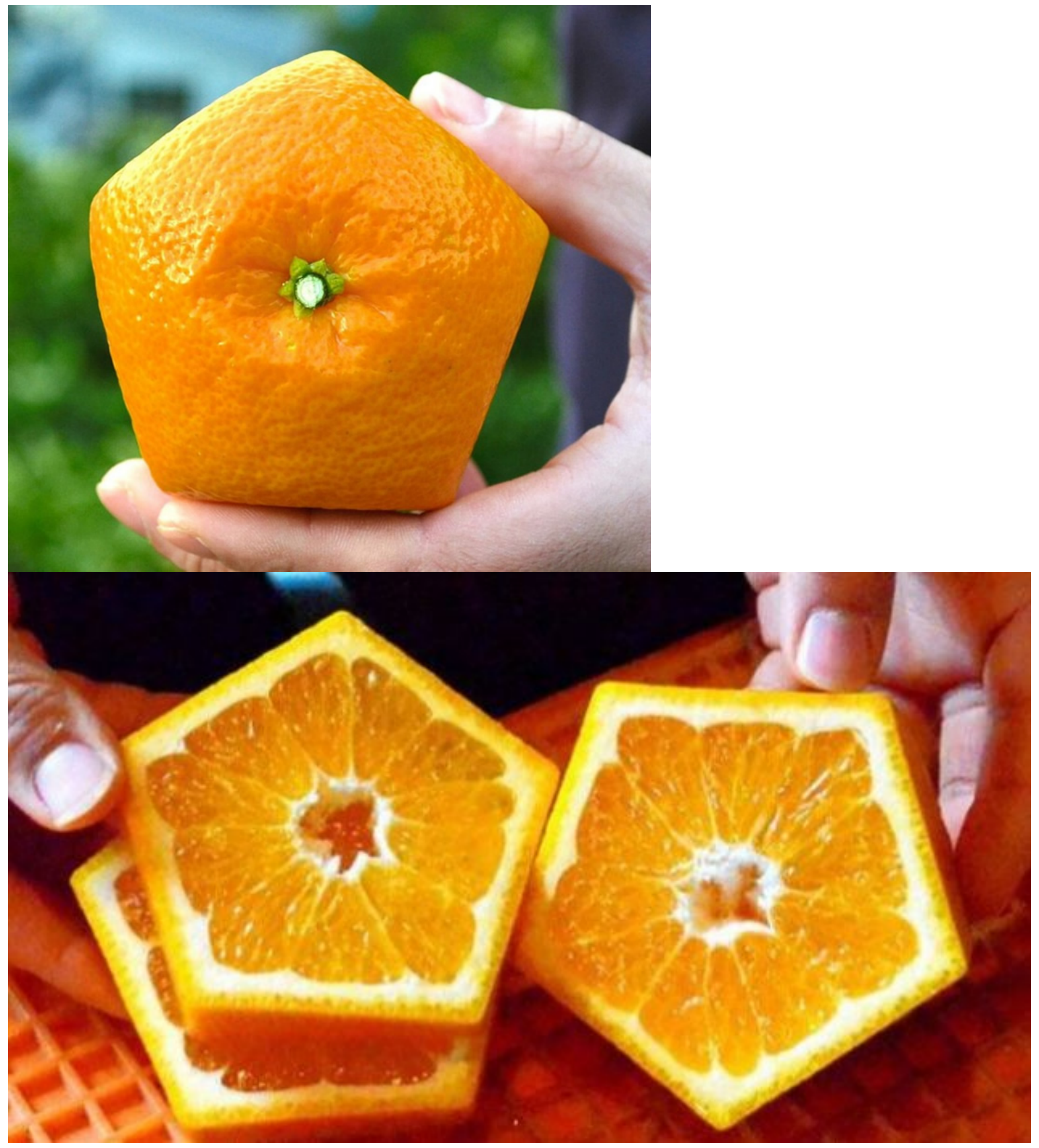


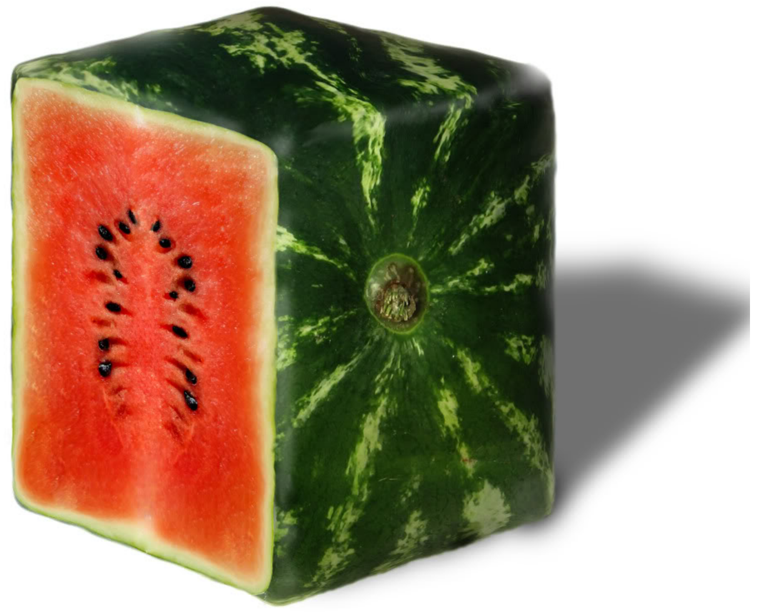
KERNEL TRICK
MAP TO HIGHER DIMENSIONS AND BACK



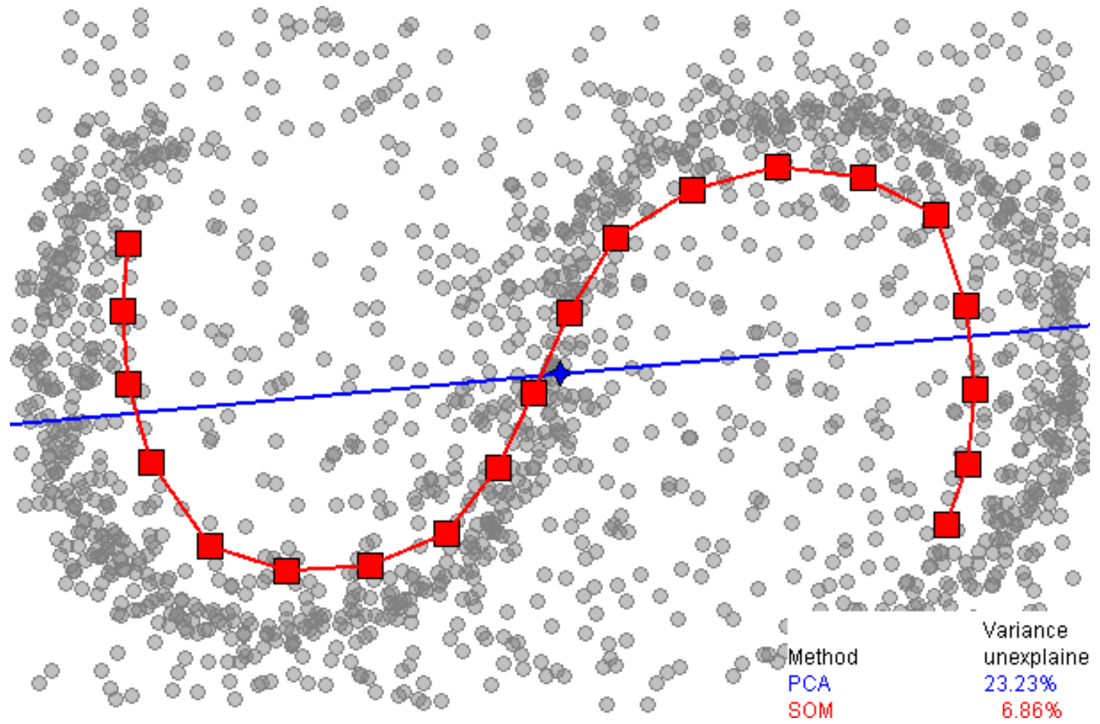
SVM
„PROBLEM“: SUPERVISED APPORACH

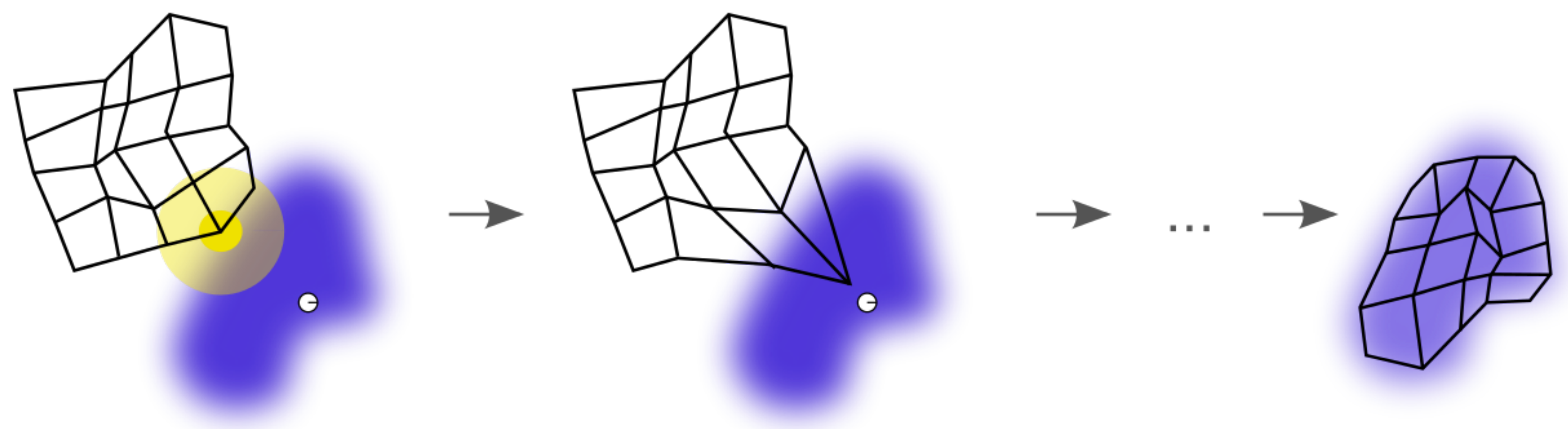
FIT A STRUCTURE OR FIT THE FEATURES





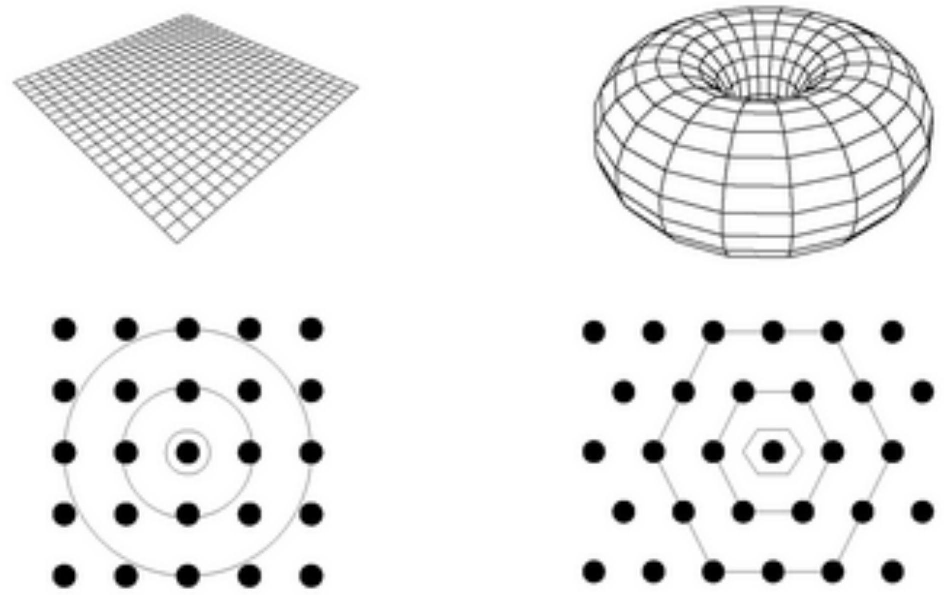
SELF-ORGANIZING MAP (SOM)
NEURAL NETWORK APPORACH



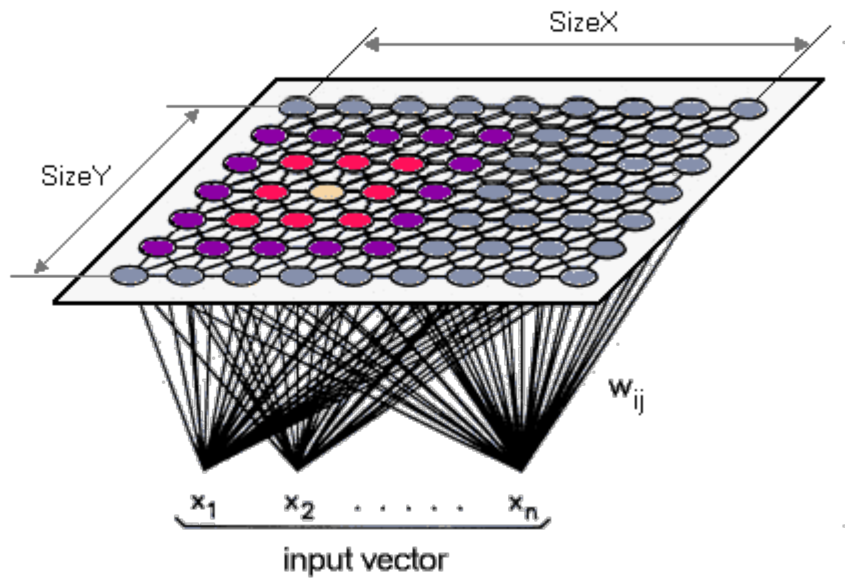


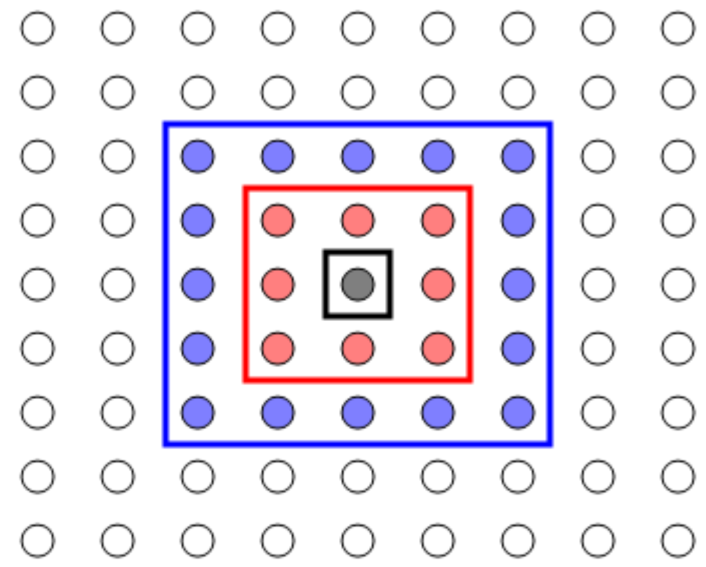
SELF-ORGANIZING MAP (SOM)

FIXED TOPOLOGY
ONLY FEATURES
ARE CHANGING

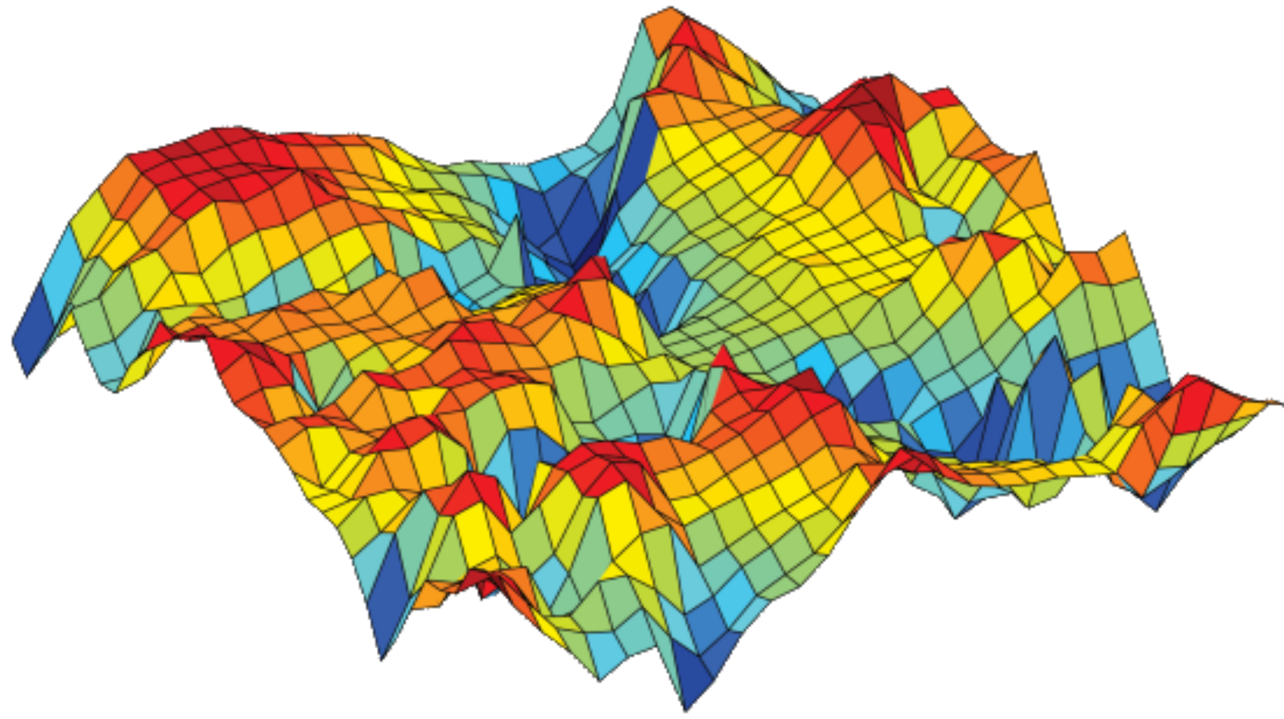


SELF-ORGANIZING MAP (SOM)
ITERATIVE TRAINING





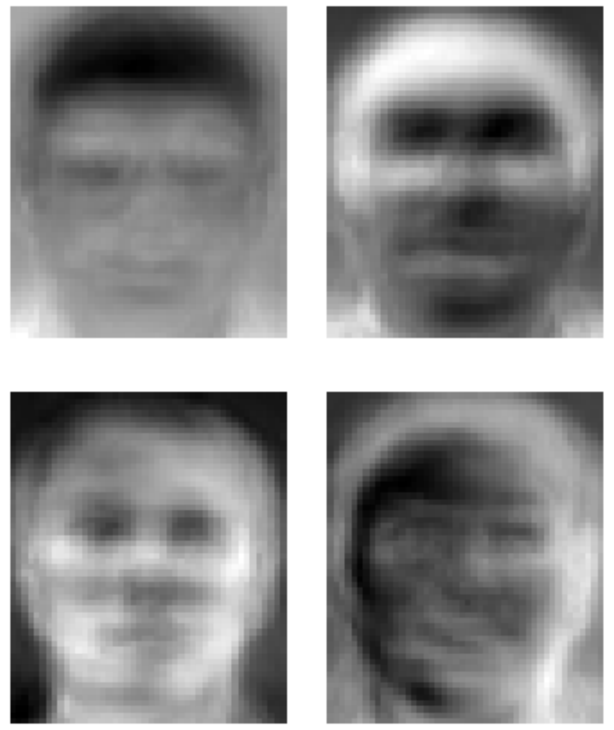
SELF-ORGANIZING MAP (SOM)
DISTANCES AS TOPOLOGY, BUT REGULAR GRID IN 2D



SELF-ORGANIZING MAP (SOM)
ADVANTAGE: READABILITY

ONTOLOGIC DISCUSSION OF SOM. INTRINSIC VS. CONTEXTUAL MEANING

EIGENFACES
KIND OF PCA VIA EIGENVECTORS



EIGENFACES
A FEW FACES ARE ENOUGH TO IDENTIFY ALL OTHERS, WITHOUT PARAMETERS

Schedule

Creative Data Mining
Intuitively Analysing Design Ideas

The goal of this course is to introduce various data mining techniques for design and urban planning applications. Students will learn how to select relevant data sources and collect their own data using a “sensor backpack”. Various methods will be applied to a common project to evaluate the predominant influencing factors of the urban environment on our perceptual experiences. A select neighborhood in the city will be used as a case study. Final results will be presented in the last class.

The course will start with an initial overview to data mining and the relevant mathematics as well as an introduction to the programming tool (RStudio). Then students will learn how to use and interpret results from a machine-learning tool to cluster self-made design sketches, which automatically generate qualitative collages. Finally, students will collect data using a “sensor backpack” with environmental sensors such as noise, temperature, illuminance, and air particulates. Students will also generate the data for perceptual quality in this neighborhood through time-stamped and geo-referenced surveys and biofeedback wristbands. Students will be given a work-flow to collect, process, analyze and interpret this data which may be used in their final projects.

Where
HIT H 12

Supervision

Danielle Griego griego@arch.ethz.ch
Matthias Standfest standfest@arch.ethz.ch

22.02.2016	**Course Introduction** Introduce data-mining techniques and case study
29.02.2016	**Introduction to the Environment** Introduction to R Studio and clustering
07.03.2016	**From analog to digital analysis** Use hand-drawn sketched to auto-generated collages
14.03.2016	**Seminar week (No lecture)**
21.03.2016	**Analysis and interpretation I** Evaluate auto-generated collages
28.03.2016	**Holiday (No lecture)**
04.04.2016	**Time-series data analysis and Urban Planning** Introduction to time-series analysis
11.04.2016	**Data collection with sensor backpack** Collect data and introduce workflows
18.04.2016	**Holiday (No lecture)**
25.04.2016	**Analysis and interpretation II** Evaluate sensor backpack data
02.05.2016	**Q&A Feedback Workshop** Finalise semester projects
09.05.2016	**Final iA critique** Combined critique with the other iA courses (14:00 - 16:00)

Requirement Former knowledge of any digital tool or coding language is most welcome but NOT required. You only need to provide a reasonable amount of motivation and of course a notebook.

* Total 60 h = 2 ECTS

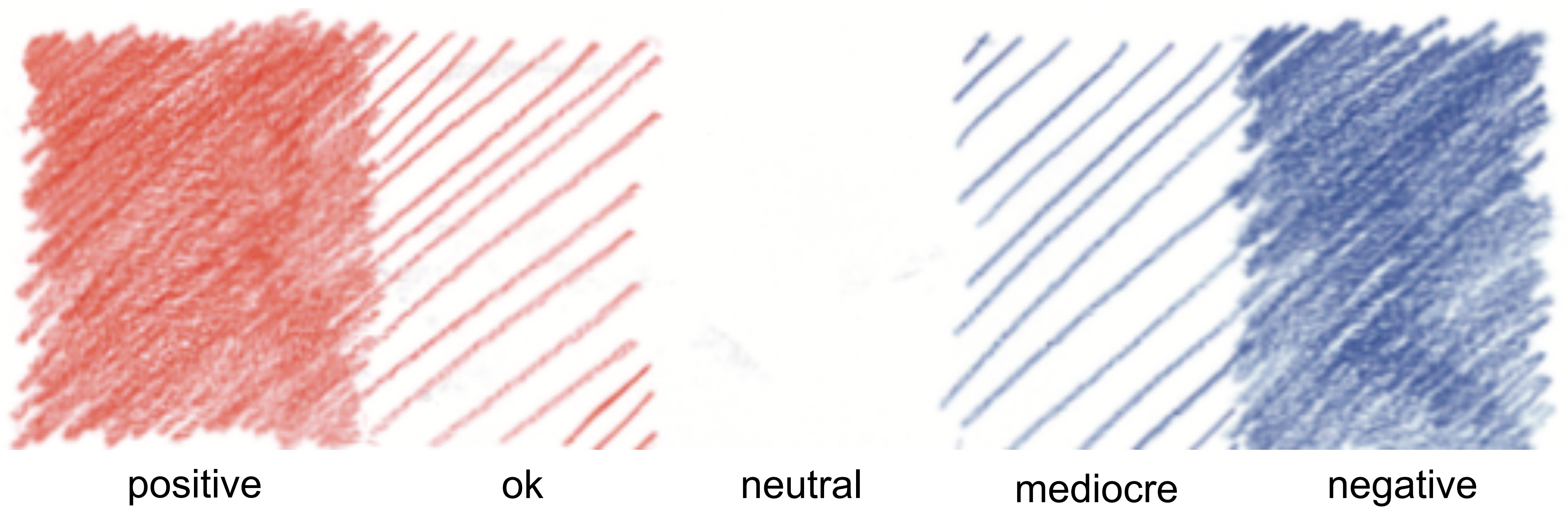
Exercises 40% (documentations)
Final Presentation 40% (Final project)
Attendance 20%

The most recent outline will be found on www.ia.arch.ethz.ch

Homework:

Part 1:

1. Color the segment maps of Alt-Wiedikon we provided today according to your perception of the urban space from a 2D plan. Hand in the hardcopies by Friday 18 March before 5pm.
2. Keep the following in mind:
    1. Use the shaded diagram below as a guide,
    2. Keep this task simple and work intuitively
    3. Be consistent for all 9 plans
    4. Reference Google satellite image to better understand the actual urban layout.



Homework:

Part 2:

1. Review the R-tutorials lecture_2_05 through lecture_2_06
2. Use a different built in dataset such as (mtcars, rock, trees, LifeCycleSavings) and visualize the clustering analysis using the improved visualization from lecture_2_06 tutorial
3. Submit a pdf of the final image by Monday 21 March