Chapter 9 : Support vector Machines

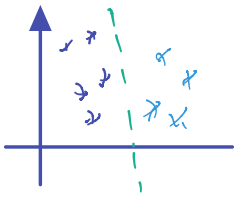
Maximal Margin Classifier - separable cases only

hyperplane : subspace of $p-1$ dim in p dim
flat and affine (doesn't need to go through the origin)

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0$$

divides space in half, if not "on the plane", then above (>0) or below (<0)

Assume the data $x_1 = \begin{pmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1n} \end{pmatrix}$, x_n is separable by a hyperplane



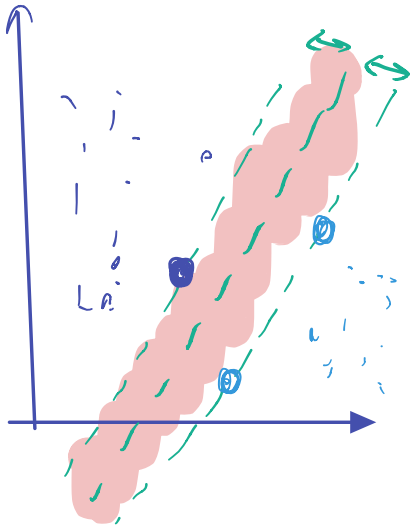
e.g $\beta_0 + \beta_1 x_{i1} + \ldots \beta_p x_{ip} > 0$ if $y_i = 1$
< 0 if $y_i = 0$

Classify based on sign e.g. $\text{sgn}(f(x^*)) = y^*$
distance e.g $|f(x^*)|$ tells us about the confidence

The best hyperplane is the one that maximizes the distance of the observations from the hyperplane

→ maximal margin hyperplane

- This is also the width of the largest "slab" you can insert between the data
- This only depends on a few points exactly on the margin, not the rest → you could move the rest and the margin wouldn't move



only depends on these three points → the support vectors

Problem:

maximize M

s.t. $\sum \beta_j^2 = 1$

$y_i(\beta_0 + \beta_1 x_{i1} + \dots \beta_p x_{ip}) > M \quad \forall i$ → correct side of the hyperplane + padding

↓ this is not really a constraint, but simplifies life:

$y_i(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip})$ will be the distance from the hyperplane to the i^{th} observation

So this means each point is at least a distance M from the hyperplane, on the correct side

Support Vector Classifiers – Non-separable cases

Separating hyperplanes are very sensitive to individual observations

→ 0% error on training

→ one observation can move the plane by a lot

→ overfitting

We might want • more robust to individual obs

• better classification to most of the training obs.

e.g. misclassify a few to do better on the rest

"soft margin classifier"

↳ margin can be violated by some obs.

Problem:

width of margin

$$\text{maximize } M$$

$$\text{s.t.} \quad \sum \beta_j^2 = 1$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots \beta_p x_{ip}) \geq M(1 - \varepsilon_i)$$

$$\varepsilon_i \geq 0, \quad \sum \varepsilon_i \leq C$$

slack variable

allowing obs. to be on the wrong side

$\varepsilon_i = 0$ if on correct side of the margin

$\varepsilon_i > 0$ violates the margin

$\varepsilon > 1$ if wrong side of hyperplane

C: budget of how wrong observation can be classified

if C grows, margin grows

if $C = 0$, separating hyperplane

C fit w/ CV

C bigger: more bias, less variance

C smaller: more variance

Observations on the margin and on the wrong side affect the classifier, the rest do not

↓

support vectors

(big C: many support vectors)

For non-linear boundaries, you can fit only poly

$$\beta_0 + \sum \beta_{j_1} x_{ij} + \sum \beta_{j_2} x_{ij}^2 + \dots \sum \beta_{jp} x_{ij}^p) \quad \text{etc.}$$

Use SVM instead: → more computationally efficient

Support vector Machines

use kernels for non-linear boundaries

it can be shown that $f(x)$ for support vector classifiers is of the form

$$f(x) = \beta_0 + \sum \alpha_i \langle x, x_i \rangle$$

where $\alpha_i \neq 0$ iff x_i is a support vector

for an SVM, you use a kernel for the inner product

e.g. $f(x) = \beta_0 + \sum \alpha_i K(x, x_i)$

$K(x_i, x_{i'}) = \sum x_{ij} x_{i'j}$ → just the same linear

$$K(x_i, x_{i'}) = \left(1 + \sum x_{ij} \cdot x_{i'j}\right)^d \rightarrow \text{polynomial}$$

$$k(x_i, x_{i'}) = \exp\left(-\gamma \sum (x_{ij} - x_{i'j})^2\right) \quad \text{radial} \quad \gamma > 0$$

More than two classes:

1-vs-1: train $\binom{k}{2}$ SVM, each compares a pair of classes

for a test observation, we count up how many each class got in total across the $\binom{k}{2}$, and put it in the most frequent one

1-vs-all: train k SVM, each compares one class w/ the remaining $k-1$

let $\beta_{0k}, \beta_{1k}, \ldots \beta_{pk}$ the params from fitting an SVM comparing the kth class to the rest

we assign the test observation to the class that has the highest $\beta_{0k} + \beta_{1k} x^*_1 + \ldots \beta_{pk} x^*_p$, as this has best confidence

Relationship w/ logistic regression:

we can rewrite the SVM problem as

L^2 / ridge penalty

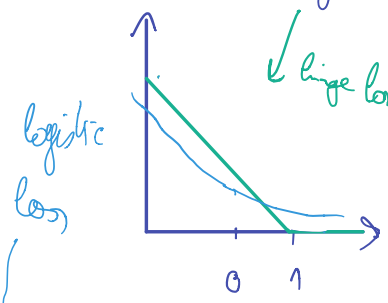
$$\text{minimize} \sum_i \max\left[0, 1 - y_i \cdot f(x_i)\right] + \lambda \sum_j \beta_j^2$$

hinge loss

tuning param

small $\lambda \approx$ small C

when λ small, fewer violations



hinge loss

- in this form, margin is at value 1, with width $\varepsilon \beta_j^2$
- observations not on the margin are not affected b.c hinge loss gives 0 error for them no matter how far they are from the margin, because $y_i(\beta_0 + \dots \beta_p x_{ip}) \geq 1$

in contrast, this is not the for logistic regression - the loss is not 0, but very small

- SVM and logistic regression give very similar results
- when classes are separated, SVM tends to be better
- when lots of overlap, log. reg. is often better

Chapter 10 - Unsupervised learning

PCA - discussed in chapter 6 for PCR

directions along which data is most variable

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \dots \phi_{p1} X_p \qquad \sum \phi_{j1}^2 = 1$$

$\phi_{11} \dots \phi_{p1}$ → loadings of 1st pc

how to construct ϕ?

- center X to have mean 0 and scale it
- maximize ϕ $\left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\}$ s.t. $\sum \phi_{j1}^2 = 1$

e.g. maximize variance of Z_1 (since Z_1 is mean 0)

If $z_{i1} = \phi_{11} x_{i1} + \phi_{21} x_{i2} + \dots + \phi_{p1} x_{ip}$

then maximize $\frac{1}{n} \sum_i z_{i1}^2$ s.t. $\sum \phi_{j1}^2 = 1$, with $\frac{1}{n} \sum z_{i1} = 0$

can be solved by eigen decomposition

since $\frac{1}{n} \sum_i x_{ij} = 0$)

$\phi_{11}, \dots \phi_{p1}$ → defines the direction where data varies the most

So just maximize sample variance of scores

$z_{11}, z_{21}, \dots z_{n1}$ → score of PC

if we project the n points $x_1, \dots x_n$ onto this direction, the values are the scores $z_{11}, \dots z_{n1}$

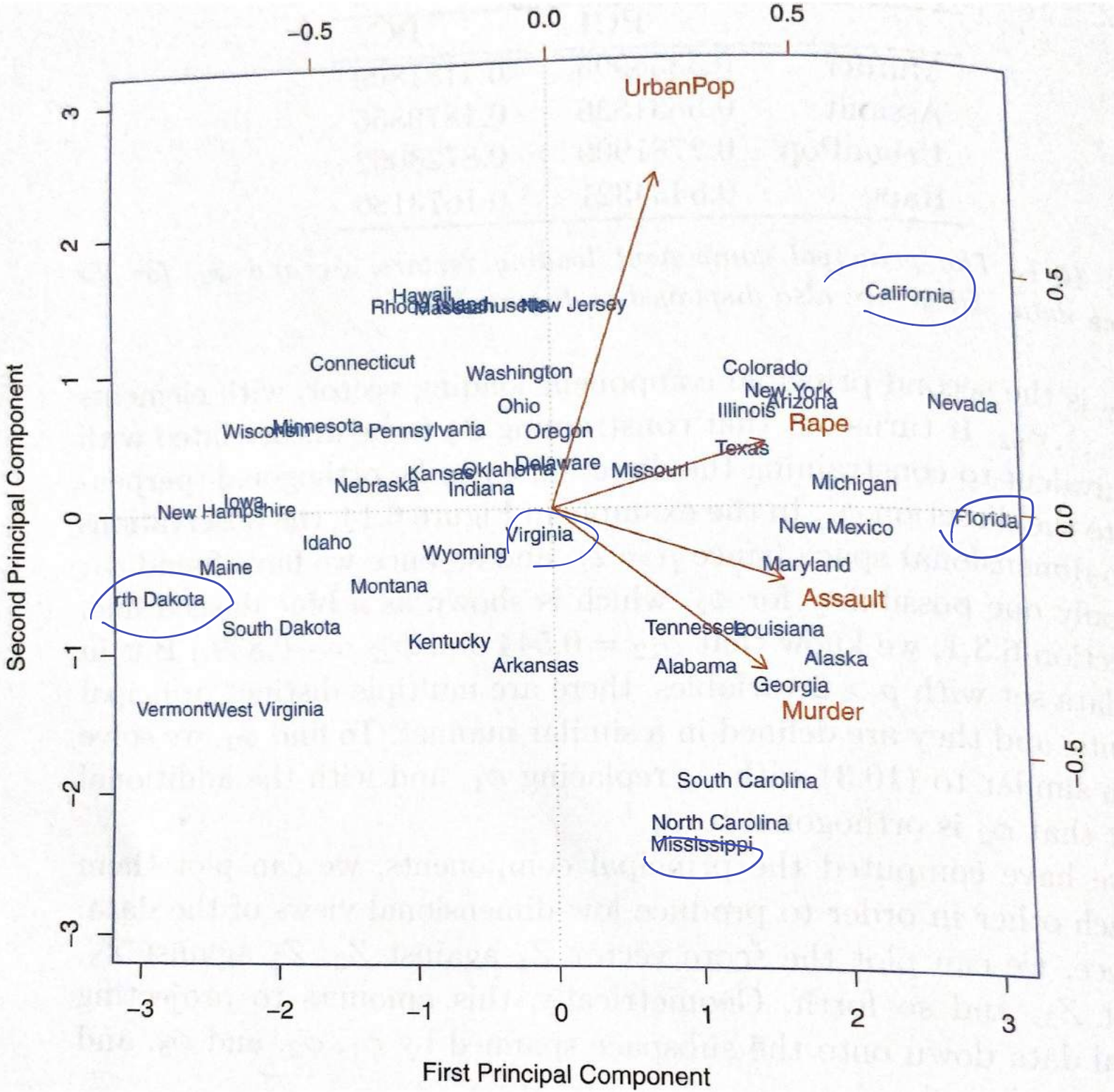
For next component z_2, w/ $z_{i2} = \phi_{12} x_{i1} + \dots \phi_{p2} x_{ip}$,

we need $\phi_2 \perp \phi_1$

We get the components by taking the eigenvectors of $X^T X$,

and the variances of the components are the eigenvalues

There are at most $\min(n-1, p)$ PCs


UrbanPop
Rape
Assault
Murder
California
Florida
Nevada
Colorado
New York
Arizona
Illinois
Texas
Michigan
New Mexico
Maryland
Alaska
Georgia
Louisiana
Tennessee
Alabama
South Carolina
North Carolina
Mississippi
New Jersey
Connecticut
Washington
Ohio
Oregon
Pennsylvania
Delaware
Missouri
Indiana
Nebraska
Kansas
Oklahoma
Virginia
Wyoming
Idaho
Maine
Montana
South Dakota
Kentucky
Arkansas
Vermont
West Virginia
New Hampshire
Iowa
Hawaii
Wisconsin
Minnesota
Pennsylvania
−0.5
0.0
0.5
−3
−2
−1
0
1
2
3
First Principal Component
Second Principal Component

Biplot: 4 variables above projected onto P.component axes

variables close to each other are correlated

e.g Rape, Assault and Murder

Florida: High crime rate

North Dakota: low crime rate

California: High crime rate + urbanization

Mississippi: low urbanization

Virginia: avg levels of both

Alternative interpretation: low dimensional hyperplanes that are closest to the observations (in p dimensions)

1st PC: line that is closest to observations

1st + 2nd PC: Plane that is closest

1st + 2nd + 3rd 3rd hyperplane that is closest

we can get back x_{ij} from the pcs:

$$x_{ij} \approx \sum_{m=1}^{M} z_{im} \phi_{jm} \quad \text{for } M \text{ pcs}$$

where if $M = \min(n-1, p)$, this is exact: $x_{ij} = \sum_{m=1}^{M} z_{im} \phi_{jm}$

(I think this is b.c ϕ are orthogonal to each other)

Scaling

- Important to scale variables to have std 1
- Or means variables in same units if you can

All of the method relies on size of variance in a direction, so if a variable is inflated, that will change everything

Proportion of variance explained (PVE)

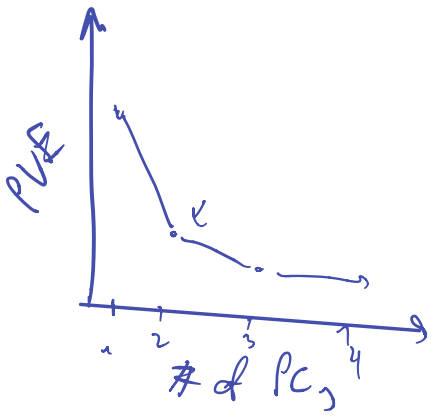
Total variance in data $= \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2$

variance explained by mth pc: $= \frac{1}{n} \sum_{i=1}^{n} \hat{z}_{im}^2 = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{jm} x_{ij} \right)^2$

PVE: $$\frac{\sum_{i=1}^{n} \left(\sum_{j=1}^{l} \phi_{jm} x_{ij} \right)^2}{\sum_{j=1}^{p} \sum_{i=1}^{n} x_{ij}^2}$$

But how many PCs to use?

Scree plot:



look for an "elbow"

ad hoc

Clustering methods

cluster obs. on the basis of features to find groups in the observations

or reverse: cluster features on the basis of observations to group features

K-means clustering:

- K clusters $C_1, \dots C_K$
- minimize variation within clusters

$$\min_{C_1, \dots C_K} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

Algo:

1. Randomly assign a label $1, \dots K$ to each observation
2. Iterate until assignments change:
   - compute cluster centroid
   - Assign each observation to the closest cluster centroid

This is a local optimum, because

$$\frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j} (x_{ij} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2$$

and step 2 decreases this every step

Important to run for different starting points, beware local optima

Hierarchical clustering

no need to specify K before the algo

look by eye where to cut the dendogram

works better on hierarchical data

Algo: 1. Begin w/ n obs + a measure for all $\binom{n}{2}$ dissimilarities

2. For $i = n, n-1, \ldots 2$:

(a) Examine all pairwise cluster dissimilarities and find least dissimilar clusters. Fuse them

(b) Compute the new dissimilarities

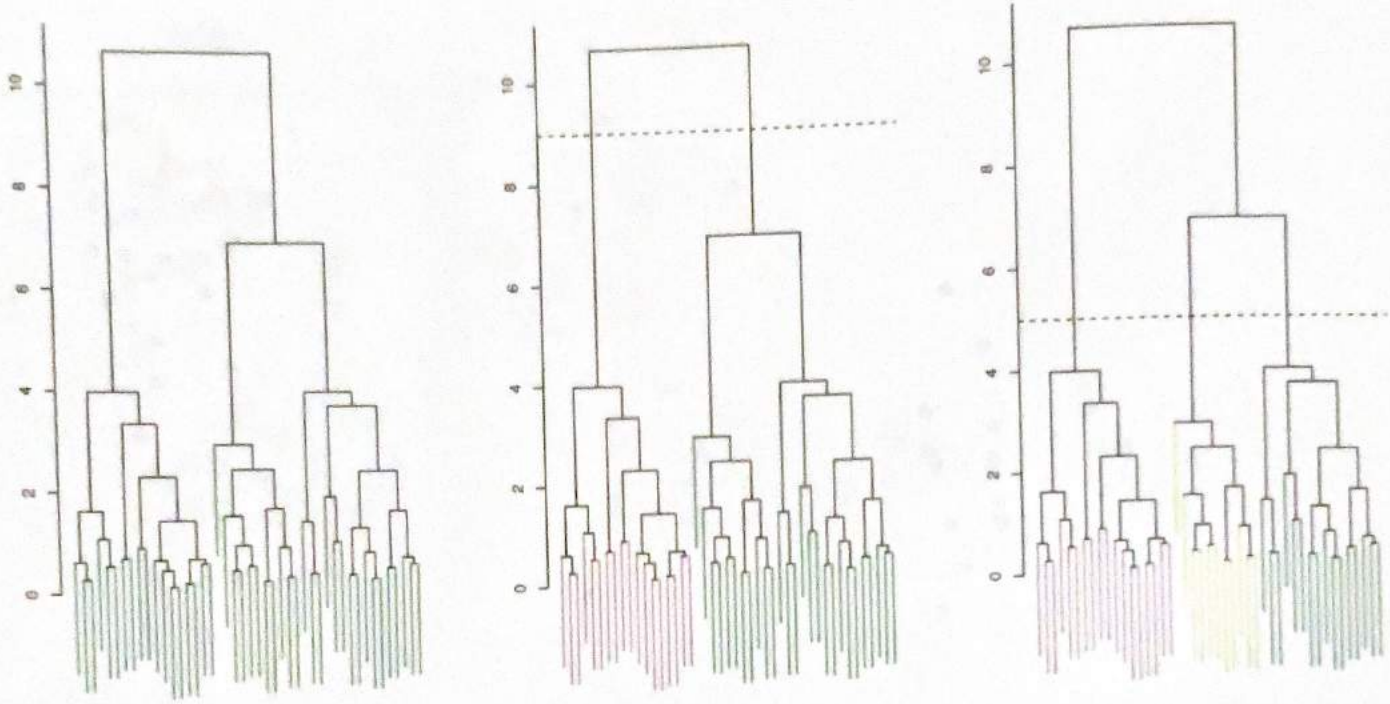


FIGURE 10.9. Left: *dendrogram obtained from hierarchically clustering the data from Figure 10.8 with complete linkage and Euclidean distance.* Center: *the dendrogram from the left-hand panel, cut at a height of nine (indicated by the dashed line). This cut results in two distinct clusters, shown in different colors.* Right: *the dendrogram from the left-hand panel, now cut at a height of five. This cut results in three distinct clusters, shown in different colors. Note that the colors were not used in clustering, but are simply used for display purposes in this figure.*

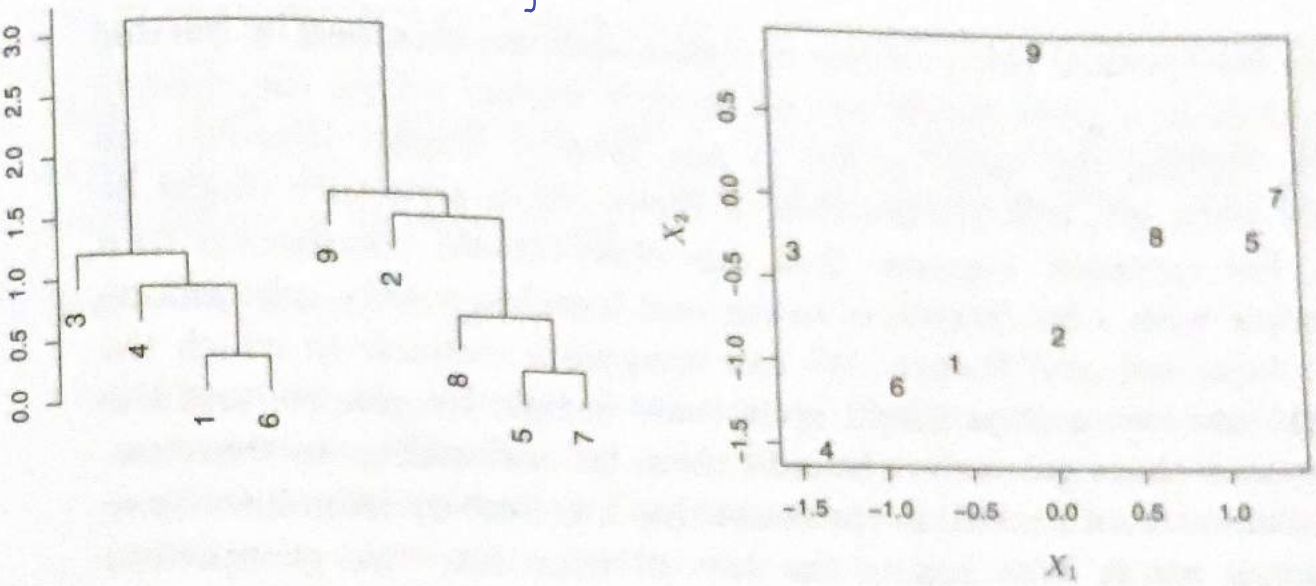


FIGURE 10.10. *An illustration of how to properly interpret a dendrogram with nine observations in two-dimensional space.* Left: *a dendrogram generated using Euclidean distance and complete linkage. Observations* 5 *and* 7 *are quite similar to each other, as are observations* 1 *and* 6. *However, observation* 9 *is* no more similar to *observation* 2 *than it is to observations* 8, 5, *and* 7, *even though observations* 9 *and* 2 *are close together in terms of horizontal distance. This is because observations* 2, 8, 5, *and* 7 *all fuse with observation* 9 *at the same height, approximately* 1.8. Right: *the raw data used to generate the dendrogram can be used to confirm that indeed, observation* 9 *is no more similar to observation* 2 *than it is to observations* 8, 5, *and* 7.

Linkage : dissimilarity between two group

- Can be euclidean distance (table below) of two observations
- Or correlation between the features for two observations

e.g useful for retail shopping carts :

→ data is 1 if bought product, 0 o/w

→ Euclidean would put everyone buying 1 product close to each other

Might need scaling too to std = 1

: rocks we bought one after other computers

Euclidean Means:

Linkage	*Description*
Complete	Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities.
Single	Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.
Average	Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *average* of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*.

Correlation-based measure

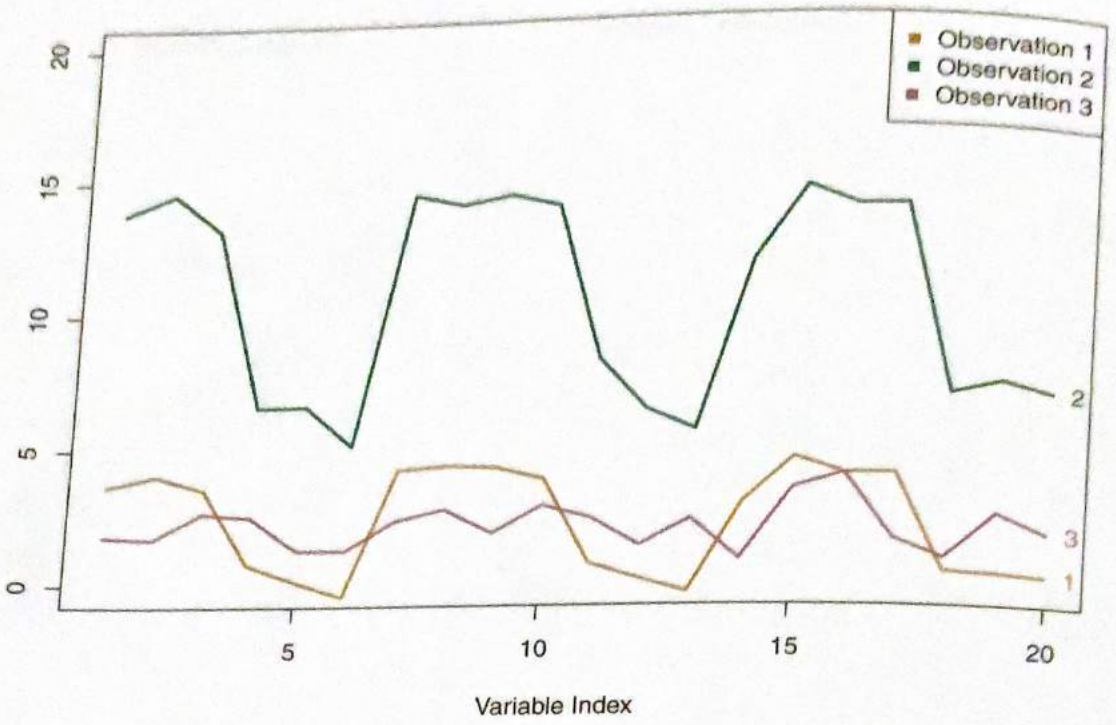


FIGURE 10.13. *Three observations with measurements on 20 variables are shown. Observations 1 and 3 have similar values for each variable and so there is a small Euclidean distance between them. But they are very weakly correlated, so they have a large correlation-based distance. On the other hand, observations 1 and 2 have quite different values for each variable, and so there is a large Euclidean distance between them. But they are highly correlated, so there is a small correlation-based distance between them.*

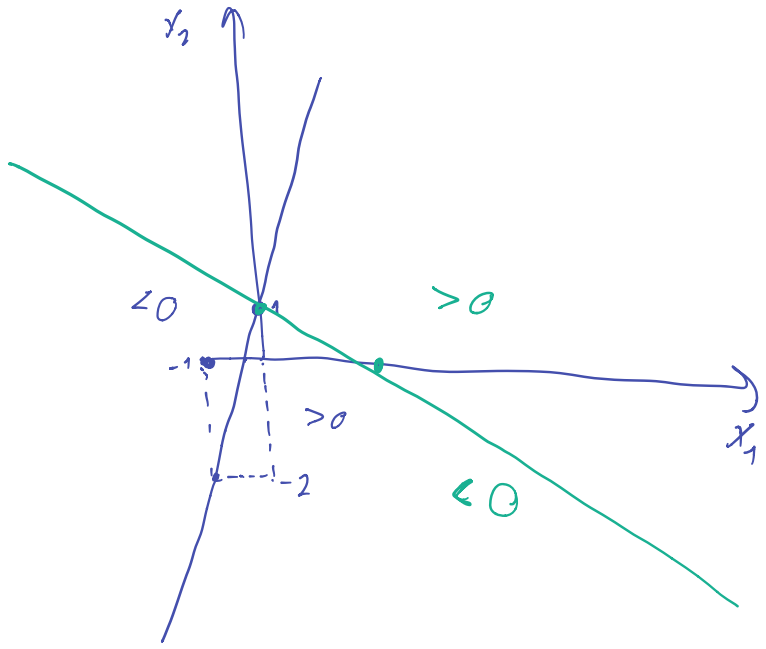
Practical issues for clustering:

- Should I center and scale the variables?
- For hierarchical, which dissimilarity measure?
- Which type of linkage?
- Where should we cut dendogram?
- For k-means, what k?

- No consensus on how to validate clusters
- Not robust to perturbations

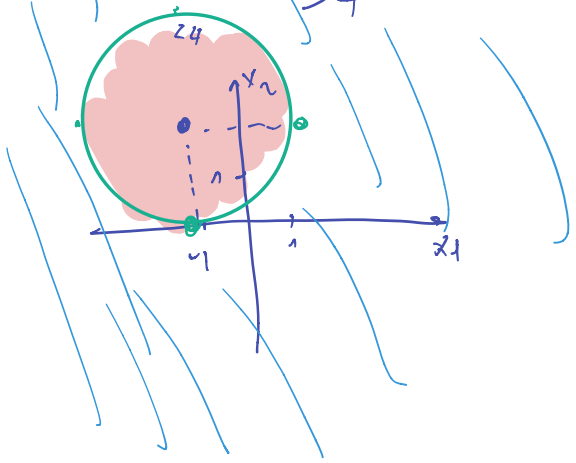
Exercises - chapter 9

1) $1 + 3x_1 - x_2 = 0$ $\qquad -2 + x_1 + 2x_2 = 0$



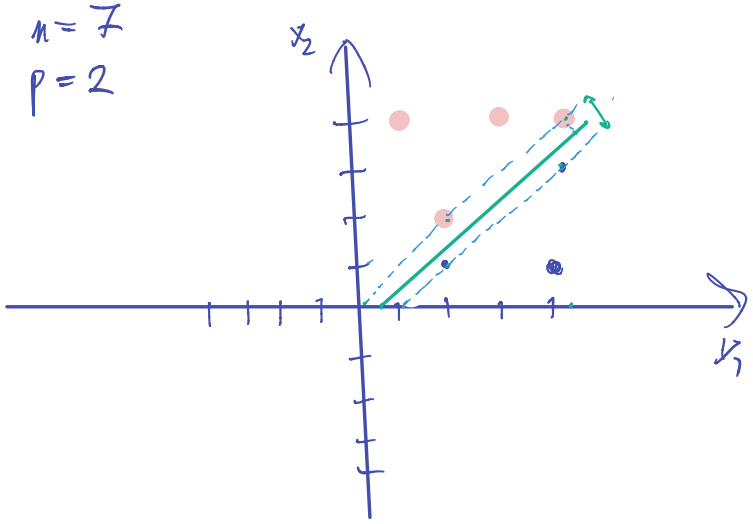
2) $(1+x_1)^2 + (2-x_2)^2 = 4$

a) $\qquad >4$ $\qquad$ c)



3, $n = 7$

$p = 2$



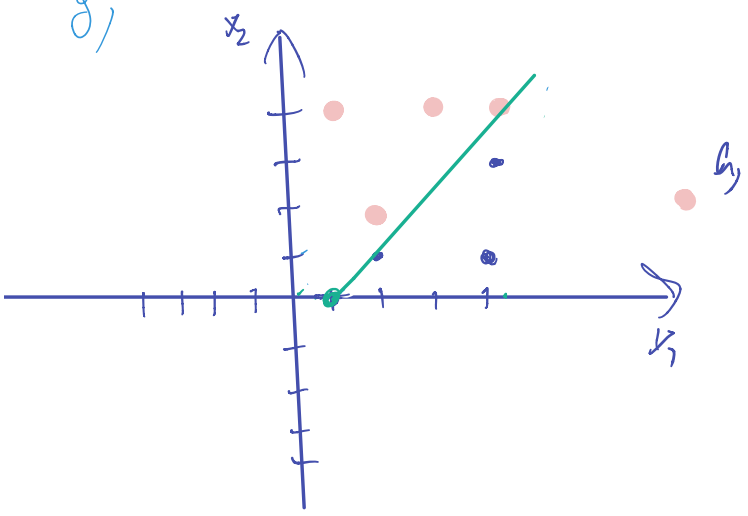
$x_1 - x_2 = 1/2$

$1/2 + x_2 - x_1 = 0$

$d >$

$d <$

8)



Equation of the straight $(3, 5/4)$ and $(4, 15/4)$

$$x_2 - \frac{5}{4} x_1 + 5/4 = 0$$

Exercises chapter 10

a) $\frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2$

$$LHS = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_j (x_{ij}^2 + x_{i'j}^2 - 2 x_{ij} x_{i'j})$$

$$= \frac{1}{|C_k|} \sum_{i \in C_k} \sum_{j=1}^{p} \left(2|C_k| x_{ij}^2 - 2 x_{ij} \cdot \sum_{i' \in C_k} x_{i'j} \right)$$

$$= 2 \sum_{i \in C_k} \sum_j \left(x_{ij}^2 - x_{ij} \frac{\sum_{i \in C_k} x_{ij}}{|C_k|} \right)$$

$$= 2 \sum_{i \in C_k} \sum_{j=1}^{p} \left(x_{ij}^2 - x_{ij} \frac{\bar{x}_{kj}}{|C_k|} \right)$$

$$= 2 \sum_{i \in C_k} \sum_{j=1}^{p} \left(x_{ij}^2 + \bar{x}_{kj} - 2 x_{ij} \frac{\bar{x}_{kj}}{|C_k|} \right)$$

$$= 2 \sum_i \sum_j (x_{ij} - \bar{x}_{kj})^2$$

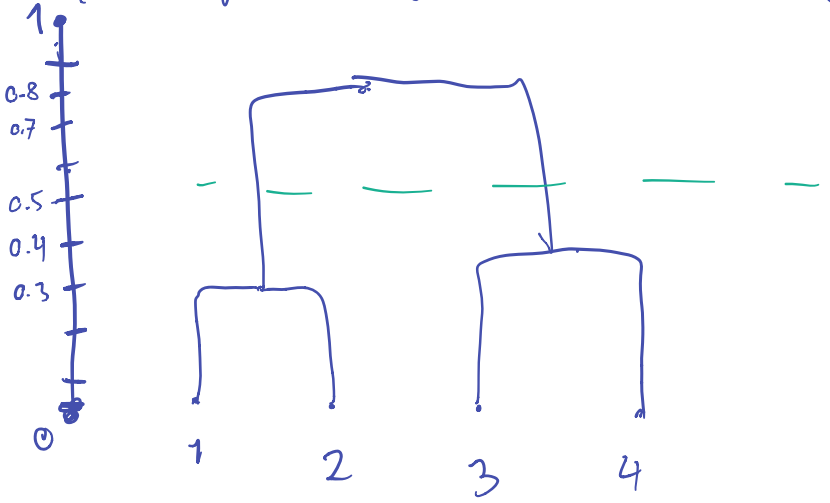
$$\frac{\sum_i x_{ij} \bar{x}_{kj}}{|C_k|} = \bar{x}_{kj}^2$$

2/

$$\begin{bmatrix} - & 0.3 & 0.4 & 0.7 \\ 0.3 & - & 0.5 & 0.8 \\ 0.4 & 0.5 & - & 0.45 \\ 0.7 & 0.8 & 0.45 & - \end{bmatrix}$$

Dendogram:

a/ complete linkage



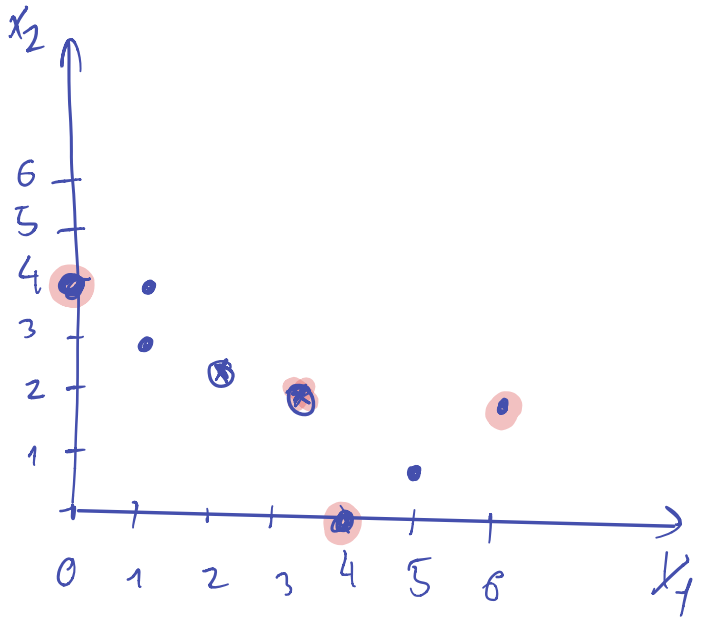
c/ (1,2) , (3,4)

b/ single linkage

d/ (1,2,3) (4)

2 3 4

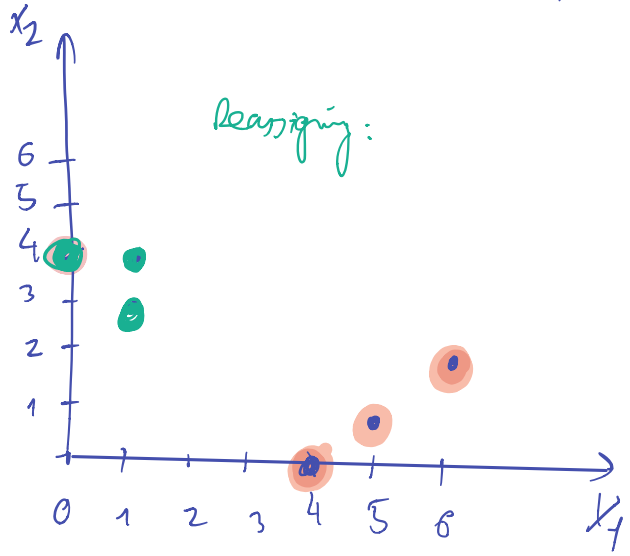
3)



Red: (0, 4) (4, 0) (6, 2)

Blue: (1, 4) (1, 3) (5, 1)

centroid: $(\frac{10}{3}, 2)$ $(\frac{7}{3}, \frac{8}{3})$



4) a) right lever

b) same height

5)