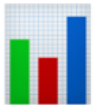


Multivariate Statistics Summary

WARNING: There are some important parts missing here (e.g. estimation of Factors in FA) which he does definetely ask in the exams. This is not complete

Chapter 1 Introduction

Eigenvector & -value

A... matrix

λ... eigenvalue

v... eigenvector

$$A\mathbf{v} = \lambda\mathbf{v}$$

The eigenvectors of **A** are the vectors that **A** merely elongates or shrinks, and the amount that they elongate/shrink by is the eigenvalue.

How to solve:

I... identity matrix (1s in diagonal)

$$\det(A - \lambda I) = 0$$

Spectral Decomposition

Spectral Decomposition is the representation of a matrix in terms of its eigenvalues and eigenvectors. Only possible for diagionalizable matrices.

If we again have matrix A, spectral decomposition express A as:

$$A = Q\Lambda Q^{-1}$$

Q... matrix with columns as eigenvectors of A

Λ... (Uppercase lambda) diagonal matrix containing eigenvalues of A in its diagional

Spectral Decomposition is useful for certain operations (e.g. matrix powers) as it simplifies the calculations. Also for PCA (more on that later).

Expectation and Covariance Basic Properties

X... matrix of Random Variables

A, B, C... data matrices (not random); equivalent to scalar

- expectation is linear → $E(AXB + C) = AE(X)B + C$
- Covariance
  - $Cov(x, y) = E[(x - \mu_x)(y - \mu_y)] = E(xy^\top) - \mu_x\mu_y^\top$
  - $Cov(Ax, By) = ACov(x, y)B^\top$
  - $Cov(x, x) = Cov(x) = \Sigma = K_{xx}$ → called Variance-Covariance matrix
  - $Cov(Ax) = ACov(x)A^\top$

Multivariate Normal Distribution

A Vector of random variables, each which is normally distributed, with specified relationships (correlations) between them.

Density Function:

$$f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n|\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top\Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)$$

Mahalanobis Distance

measure of distance between point x and mean, consider covariance structure of distribution. It adjusts for correlation and different scales of the data

$$d_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^\top\Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})}$$

this is used in the density function of multivariate normal distribution

Chapter 2 Cluster Analysis

Similarity = Homogeneity ↔ Dissimilarity = Heterogeneity

- homogeneity: within-cluster-sum-of-squares
- heterogeneity: single-linkage, complete-linkage, between-clusters-sum-of-squares

Distance Measures

Distances can be computed for every observation pair, which results in symmetric distance matrix D. Diagonal elements are zero.

Euclidean Distance

$$d(i,j) = \sqrt{\sum_{k=1}^{p}(x_{ik} - x_{jk})^2}$$

Manhattan Distance (City-Block)

$$d(i,j) = \sum_{k=1}^{p}|x_{ik} - x_{jk}|$$

less robust than euclidean

Partitioning vs. Hierarchical

Partition

- observation is assigned to exactly one cluster → Clusters are non-overlapping
- Usually number of clusters (k) is fixed beforehand

Hierarchy

- each level of hierarchy consists of a partition
- starting from bottom, with n clusters, combining clusters for each hierarchy level → agglomerative clustering
- starting from Top with one single cluster and step by step splitting clusters → Divisive clustering
- can be visualized with a dendrogram:
  - vertical axis (height): similarity measure at which clusters are paired

- for each step the pair of clusters with smallest values of the measures will be merged

Fuzzy

- each observation is proportionally distributed among all clusters
- membership coefficient $0 \leq u_{ij} \leq 1$ for each observation and each cluster

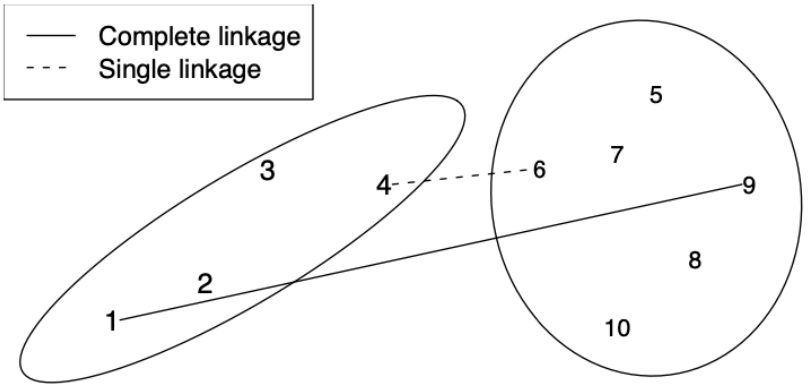
Hierarchical Clustering

Complete Linkage

- similarity between clusters is defined by pair of objects from different clusters which have biggest distance
- $\max_{i \in C_k, j \in C_l} d(i,j)$

Single Linkage

- similarity is given by closest observations from two clusters
- $\min_{i \in C_k, j \in C_l} d(i,j)$
- tends to be unbalanced → big clusters are quickly combined → tends to produce many small groups and few large groups
- suitable to detect outliers



Average Linkage

- similarity is defined as average of all pairwise distances

- $\frac{1}{n_k n_l} \sum_{i \in C_k} \sum_{j \in C_l} d(i,j)$

Centroid Method

- first compute arithmetic means of observations of each cluster
- similarity is then given by euclidean distance of cluster centers (centroids)

Wards method

- similarity is defined as increase of in-cluster variance when merging two clusters
- $\frac{||\bar{x}(C_k) - \bar{x}(C_l)||^2}{1/n_k + 1/n_l}$

Partitioning

K-means

- works with Euclidean distance
- typically assumes spherical shaped clusters (due to euclidean distances)

Objective Function:

$$W(C) = \sum_{k=1}^{K} n_k \sum_{\mathbf{i} \in C_k} \|\mathbf{x}_{i.} - \bar{\boldsymbol{x}}_k\|^2$$

1. Cluster centers are initialized. Usually randomly selecting k observations
2. minimize objective function by assigning each observation to closest cluster center
3. replace cluster centers with arithmetic means of observations per cluster

- could end in local optimum
- multiple restarts are recommended

Model-based Clustering

- can also be used to obtain cluster partition
- but also gives probability for class membership

- assumed that clusters stem from p-dimesional normal distributions
- parameters of normal distributions have to be estimated
- easiest case: $\Sigma_k = \sigma^2 \boldsymbol{I}$ → only one variance has to be estimated
- less restricted case: $\Sigma_k = \sigma_k^2 \boldsymbol{I}$ → different sizes and shapes of spheres for clusters

Fuzzy Clustering

- proportional assignment of observation to all clusters
  - membership coefficient $u_{ik} \in [0, 1]$
- sum of proportions is 1
- best-know algorithm is **fuzzy k-means**
  - objective function, which needs to be minimized:

$$\sum_{j=1}^{k} \sum_{i=1}^{n} u_{ij}^2 \|\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_j\|^2 \longrightarrow \min,$$

  - $\tilde{x}$... weighted cluster center

Evaluation of Classification

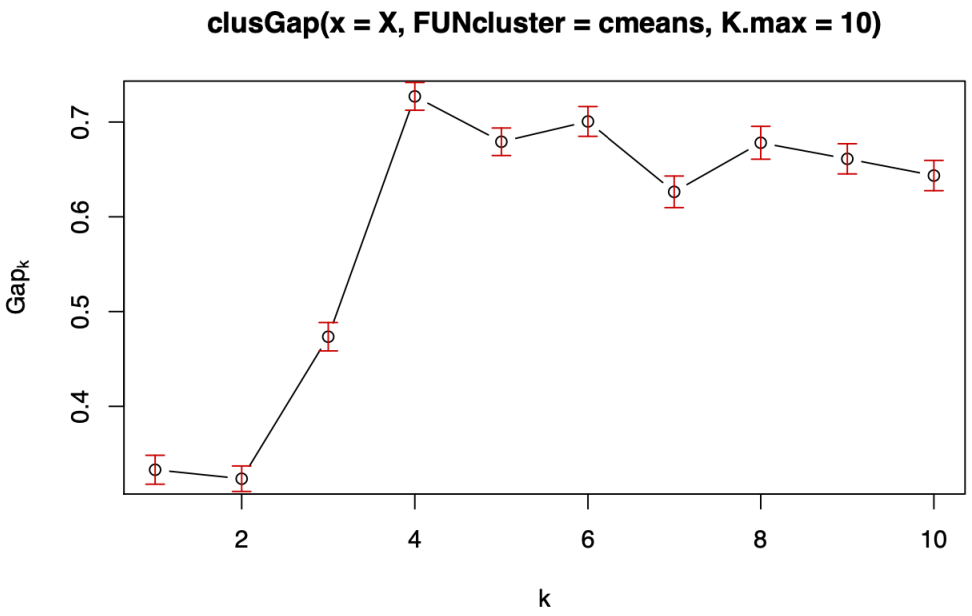
- heterogeneity can be measured by **Between Cluster Sum of Squares**
  - $B_K = \sum_{k=1}^{K} ||\bar{x}_k - \bar{x}||^2$
  - with $\bar{x}_k$ as center of cluster k and $\bar{x}$ as overall mean of cluster centers
- homogeneity can be defined by **Within Cluster Sum of Squares**
  - $W_K = \sum_{k=1}^{K} \sum_{i \in C_k} ||x_i - \bar{x}_k||^2$
- B_K should be large, W_K should be small
- but both of these measures depend on number K of clusters, and this should be considered in a validity measure
- two important measures:
  - **Calinski Harabasz index**
    - $CH_K = \frac{B_K/(K-1)}{W_K/(n-K)}$
  - **Hartigan index**

- $H_K = \ln \frac{B_K}{W_K}$

“Practically, one considers a range of values for the possible number of clusters and computes the validity measure(s) for each cluster solution. The largest value of the index determines the optimal number of clusters.”

other important validity measures:

- **Average Ailhouette Width**
  - measures clustering validity by separation and cohesion of clusters
  - **Calculation**: Compares average intra-cluster distance (cohesion) with nearest inter-cluster distance (separation)
    - $d_{i,C_k} = \frac{1}{n_k - 1} \sum_{i,j \in C_k, i \neq j} d^2(i,j)$
    - $d_{i,C_l} = \frac{1}{n_l} \sum_{j \in C_l} d^2(i,j)$ → smallest of these is $d_{i,C} = min_l d_{i,C_l}$
    - $s_i = \frac{d_{i,C} - d_{i,C_k}}{max(d_{i,C_k}, d_{i,C})}$
  - **Score Range**: Values range from -1 (poor clustering) to +1 (ideal clustering)
  - **Purpose**: Higher scores indicate better-defined clusters; aids in selecting optimal clustering models
  - average silhouette width is taking average over all individual silhouette widths and serves as measure of overall quality of classification
  - silhouette plot: bars of silhouette widths of each point, sorted within each cluster
    - wide bars → coherent clusters
- **Gap Statistic**
  - $GAP_n(K) = E_n^* log(\tilde{W}_K) - log(\tilde{W}_K)$
  - with $\tilde{W}_K$ being the pooled within-cluster sum of squares
    - $\tilde{W}_K =$
  - E_n^* is expectation under a sample size n from appropriate reference distribution
  - comparison of within sum of squares of our data and within sum of squares of uniform data is taken
  - we search for smallest k so that gap statistic yields a local maximum
  - but since we estimate, we need to factor error into it
    - therefore usually use smallest K so that gap statistic is less than one standard error away from first local maximum



- so here optimal K = 4

Chapter 3 Multiple Linear Regression

Multiple Linear Regression → predict single response variable from multiple explanatory variables

Multivariate Regression → predict multiple response variables from multiple explanatory variables

Σ in this context is the variance-covariance matrix of the residuals (errors)

Multiple Linear Regression

General form:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon$$

Can also be written i matrix form for n observations:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

- y... vector of observed values for dependent variable
- X... matrix of predictors (first column of 1s for intercept, has a size of n x (p + 1))
- beta... vector of coefficients

- epsilon... vector of errors
- Assumptions:
  - $\mathrm{E}(\epsilon) = 0$
  - $Cov(\epsilon) = \mathrm{E}(\epsilon\epsilon^\top) = \sigma^2 \boldsymbol{I}_n$
  - σ^2 is the variance of the error term and is the same for all components. the different error terms are uncorrelated

Least-squares estimator

- minimizes the sum of squared residuals
  - $S(\beta) = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_p x_{ip})^2 = (y - X\beta)^\top (y - X\beta)$
- reordering and deriving by beta yields
  - $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$
- with beta hat then the fitted values y hat can be calculated
  - $\hat{y} = X\hat{\beta} = X(X^\top X)^{-1} X^\top y$
- the Hat-matrix H (important later) is defined as
  - $H = X(X^\top X)^{-1} X^\top$

Multivariate linear regression

- now instead of one response variable we have m response variables
- Assumptions:
  - $E(\epsilon) = 0$
  - $Cov(\epsilon) = \Sigma$ → error terms for different responses an be correlated
- general matrix form
  - $\mathbf{Y} = \mathbf{XB} + \mathbf{E}$
  - dimensions of matrices
    - Y: n x m
    - X: n x (p + 1)
    - B: (p + 1) x m
    - E: n x m
- Assumptions:

    - $E(\epsilon_j) = 0$
    - $Cov(\epsilon_j, \epsilon_k) = \sigma_{jk} I_n$
- least squares estimator
    - $S(B) = \sum_{i=1}^{n} \sum_{j=1}^{m} (y_{ij} - \beta_{0j} - \beta_{1j} x_{i1} - \cdots - \beta_{pj} x_{ip})^2$
    - which leads to:
    - $\hat{\mathbf{B}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}$
- the fitted values are obtained by
    - $\hat{Y} = X\hat{B}$

- unbiased estimator for Σ is residual matrix
    - $S_R = \hat{\mathbf{\Sigma}} = \frac{1}{n-p-1} (\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}})^\top (\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}})$
- we could also consider a reduced model, and test wether this reduced model is adequate
    - H_0 → reduced model is sufficient
    - H_1 → full model provides significant better fit
- we can use likelihood-ratio test statistic
    - $\Lambda = (\frac{|S_R|}{|S_{R^*}|})^{n/2}$
    - this follows a wilks lambda distribution
    - for large n this can be approximated by chi-squared distribution
        - df: m(p - q)

Chapter 4 Robust Statistics

- goal of robust statistics: less dependence on model assumptions; we fit only majority of the data, where the requirements are fulfilled
- good robust estimators downweight outliers
    - but also provide diagnostics to reveal outliers

Basic concepts

Influence function

- used to assess sensitivity of an estimator to the changes of one observation in the data
- EIF → empirical influence function
- EIF is ideally smooth, no local spikes, no steps
- $IF(x, T, G) = \lim_{\epsilon \to 0} \frac{T[(1-\epsilon)G + \epsilon\delta_x] - T(G)}{\epsilon}$
  - IF... influence function
  - T... estimator
  - G... distribution

Maxbias Curve

- shows how much the estimate can be biased by a fraction of contaminated data

$$\text{maxbias}(m, T, X) = \sup_{\check{X}} ||T(\check{X}) - T(X)||$$

- $\check{X}$... dataset in which m out of n observations are replaced with arbitrary values

Breakdown Point

- every estimator has a point where maxbias tends to infinity → breakdown point
- basically percentage of data that are outliers, before estimator yields useless values

$$\epsilon_n^*(T, X) = \min\{\frac{m}{n}; \text{maxbias}(m, T, X) = \infty\}$$

- least squares has breakdown point of 0 (it is not robust at all)
- mean → 0, median → 0.5
- maximum breakdown point of regression estimators is 0.5
- higher breakdown point → better; but should not lose efficiency

Statistical efficiency

- every estimator also has a corresponding variance
- e.g. least squares estimator has minimum variance for unbiased estimators of linear model

- therefore any other regressione estimators will have higher uncertainty
- estimators with high efficiency will require less data for same certainty

Robust regression

outliers in response → **vertical outliers**

outliers in explanatory variables → **leverage points**

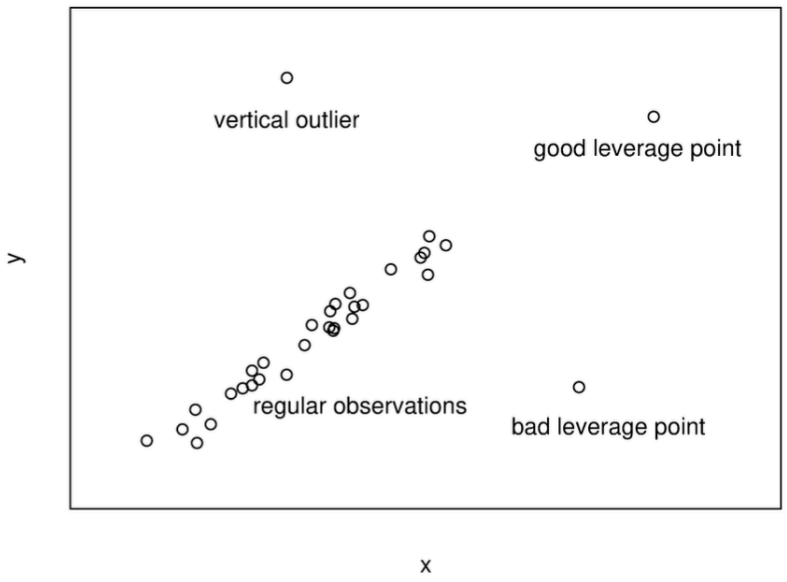


Figure 4.9: Different types of outliers, illustrated in simple linear regression.

M-estimator

- broad class of estimators
- defined as solutions to optimization problem
- $\hat{\theta}$... (theta) parameter to be estimated
- general form:

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} \rho(y_i, \theta)$$

- example for LS-estimator:
  - r... residual

    - $\hat{\beta}_{LS} = \arg\min_\beta \sum_{i=1}^n r_i(\beta)^2$
- this could be rewritten as
    - $\hat{\beta} = \arg\min_\beta \sum_{i=1}^n \rho(r_i(\beta))$
    - this is the definition of m-estimators
- to not depend on scale a better definition would be:
    - $\hat{\beta} = \arg\min_\beta \sum_{i=1}^n \rho(\frac{r_i(\beta)}{\hat{\sigma}})$
- now differentiated after beta we get the m-estimating equation
    - $\sum_{i=1}^n \psi(\frac{r_i(\beta)}{\hat{\sigma}})x_i = 0$
    - where $\psi = \rho'$
- these can now be extended to use a weight function in place of psi
    - $\sum_{i=1}^n w_i(y_i - x_i^\top \beta)x_i = 0$
- for an estimator to be robust, observations with large residuals should get small weight

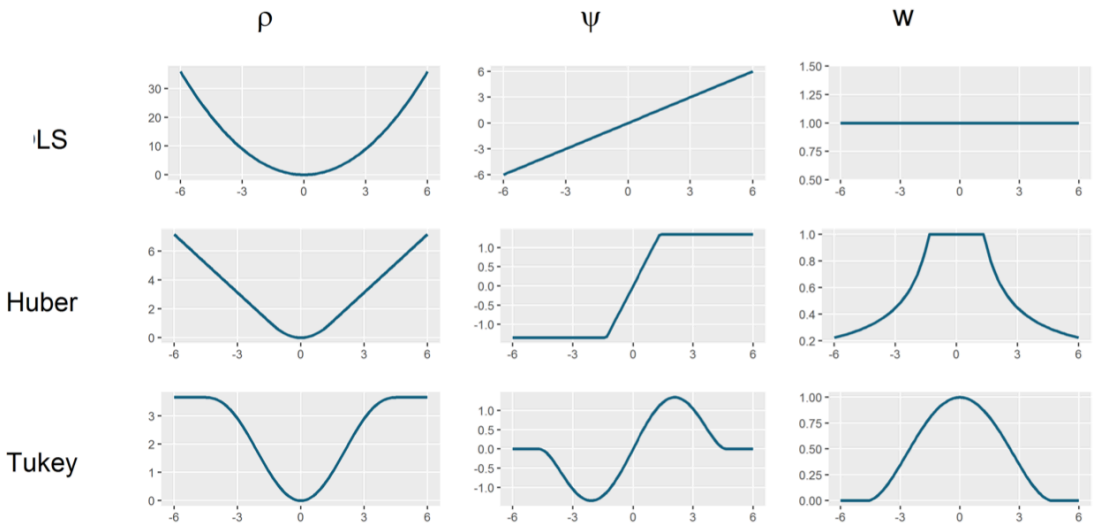


Figure 4.2: Different options for the function ρ, ψ and the weights w.

- Tukey is good here, because the rho function is bounded
    - a large x_i can therefore not dominate the sum
    - therefore good to use rho function with scale involved
- leverage point: good if not influencing regression line, bad if influencing
- with these functions we could do **iterative reweighted least squares (IRWLS)**
    - we need robust initial estimate β_0

- then iterate and improve estimate each time

Regression estimators based on robust residual scale

Least median squares

$$\hat{\sigma}(r) = \text{med}_i |r_i|$$

Least trimmed squares

- |r|_i → ordered absolute values of residuals
- then take $h \in [n/2, n]$; smaller h → higher breakdown point, but less efficient
- $\hat{\sigma}(r) = \sqrt{\frac{1}{h} \sum_{i=1}^{h} |r|_i^2}$
- basically: largest residuals get omitted from mean calculation

M-estimator of scale / S-estimator

solution to equation of this form

$$\frac{1}{n} \sum_{i=1}^{n} \rho(\frac{r_i}{\hat{\sigma}}) = b$$

- rho is bounded; good choice is huber or tukey biweight
- b is a tuning constant that determines robustness-efficiency tradeoff

Robust location and covariance

Affine equivariance

- we want location and covariance estimates to respond in a mathematically convenient form to certain transformations of the data
- if we have a transformation of our observations given by
  - $Ax_j + b$
- and t is our location estimator
- then we want to have
  - $t(Ax_i + b) = A * t(x_i) + b$
- same for covariance estimator C
  - $C(Ax_i + b) = A * C(x_i) * A^\top$

- estimators that fulfill both of this are called **affine equivariant**
- these estimators transform properly under changes of the origin, scale, or rotations

MCD estimator

- Minimum Covariance Determinant
- mcd is affine equivariant and has high breakdown point
- works similar to LTS estimator
- we search for h with lowest determinant of empirical covariance matrix
- location **t** is then **mean** of **h observations**
- covariance **C** is given by covariance matrix, but multiplied by constant to be consistent for normal distribution

Robust regression diagnostics

- fitted values can be expressed as
  - $\hat{y} = Hy$ → hat matrix "puts the hat" on y
  - the diagonal elements h_{ii} of H represent the leverage of the i-th observation
    - high leverage → large influence on fitted values
  - high values of h_{ii} indicate outliers
  - rule of thumb: $h_{ii} > 2\frac{p+1}{n}$ → leverage point
- we can show that h_{ii} elements are proportional to non-robust mahalanobis distances
  - these can clearly be spoiled by outliers
- also: h_{ii} values only consider x-values, therefore cannot differentiate between good and bad leverage points
- robust alternative: **mahalanobis distance based on robust estimators (e.g. MCD)**

Robust multivariate regression

we want to robustly estimate B and Σ, using multivariate S-estimators

refresher: B... regression coefficients, Sigma... covariance matrix of error terms

- we want to have an estimator C of Sigma
- we have to minimze
  - $\hat{\sigma}(r_{1.}^{\top} C^{-1} r_{1.}, \ldots, r_{n.}^{\top} C^{-1} r_{n.})$
  - using an m-estimator of scale $\hat{\sigma}$
  - under Constraint $|C| = 1$

Chapter 5 PCA

- goal: describe complex relationships in data in a simpler form
- data is represented by linear combinations of specific components, while aiming to preserve as much information as possible
- dimensionality is reduced to number of components

Definition

- x... p-dimensional random vector
- mu... expectation, Sigma... covariance matrix
- we can now take
  - $\Gamma = (\gamma_1, \ldots, \gamma_p)$... p x p matrix with fixed values (non-random)
  - γ_i are unitary vectors (sum of components = 1)
  - γ_i are orthogonal to each other
- now take linear transformation
  - $z = \Gamma^{\top}(x - \mu)$
    - or equivalent as components
    - $z_i = \gamma_i^{\top}(x - \mu)$
- then we have new random variable z (of dimension p)
  - with $Var(z_i) = \gamma_i^{\top} \Sigma \gamma_i$

- now we want to select to components of Gamma to have maximum variances

1. Principal Component

- we want to maximize $Var(z_1)$ and have $\gamma_1^\top \gamma_1 = 1$
- as lagrangrian problem that is:
  - $\phi_1 = \gamma_1^\top \Sigma \gamma_1 - a_1(\gamma_1^\top \gamma_1 - 1)$
- we partially derive after γ_1 and get
  - $\Sigma \gamma_1 = a_1 \gamma_1$
- this is an eigenvector/eigenvalue problem
  - γ_1 is the eigenvector to covariance matrix
  - a_1 is the eigenvalue
- but now which eigenvector of covariance matrix should we take to maximise variance?
- we can use previous result and insert into Variance equation
  - $Var(z_i) = \gamma_i^\top \Sigma \gamma_i = \gamma_i^\top (a_1 \gamma_i) = a_1 \gamma_i^\top \gamma_i = a_1$
- therefore we can take the eigenvector with the largest eigenvalue
- Component z_1 is now the first principal component and γ_1 is the direction of it

2. Principal Component

- now we want next component which maximizes variance, but is also uncorrelated to first PC
- $Cov(z_1, z_2) = \gamma_1^\top \Sigma \gamma_2 = \gamma_2^\top \Sigma \gamma_1 = \gamma_2^\top a_1 \gamma_1 = a_1 \gamma_2^\top \gamma_1 = 0$
- since $a_1 \neq 0$ we know that γ_1 and γ_2 need to be orthogonal
- we can now formulate a new Lagrange problem
  - $\phi_2 = \gamma_2^\top \Sigma \gamma_2 - a_2(\gamma_2^\top \gamma_2 - 1) - b\gamma_2^\top \gamma_1$
- partial derivative after γ_2 and equal to zero leads to $b = 0$
- therefore we can reduce to
  - $\Sigma \gamma_2 = a_2 \gamma_2$
- this can be repeated for all other PCs

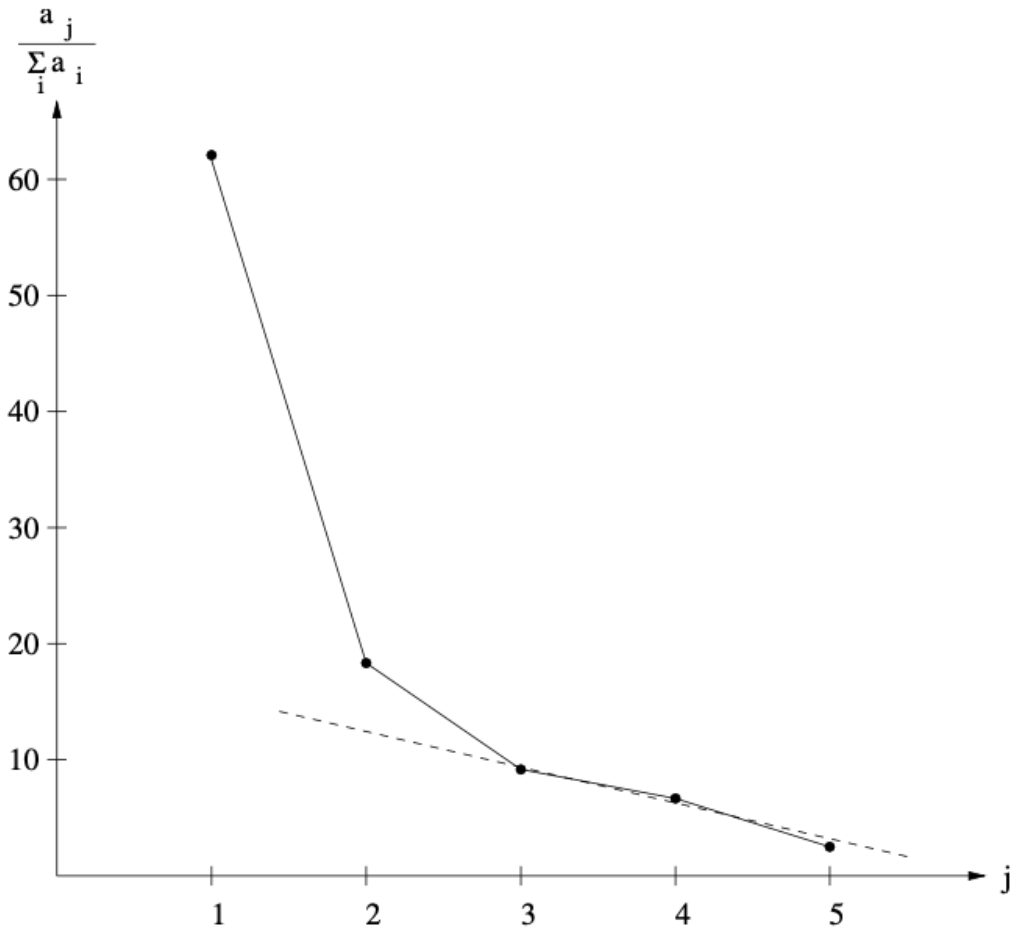
- we can now collect these Eigenvector in matrix $\Gamma = (\gamma_1, \dots, \gamma_p)$
- and have eigenvalues in diagonals of matrix $A = \text{Diag}(a_1, \dots, a_p)$
- PC solution can be expressed as
  - $\Sigma\Gamma = \Gamma A$
  - or also $\Sigma = \Gamma A \Gamma^{-1}$ (spectral decomposition)
- this equation from earlier is **Principal Component transformation**
  - $z = \Gamma^\top (x - \mu)$
  - i-th element of z is the i-th principal component
- also: $X = Z\hat{\Gamma}^\top$
- note: this transformation is not scale-invariant, i.e. PCs depend on scale of data
  - therefore its usually a good idea to scale and center (standardize) the data
- the Gamma matrix relates x and z, this is also called **loadings matrix**
  - element γ_{ij} shows influence of x_i on z_j
- estimate of Sigma
  - $\hat{\Sigma} = S = \frac{1}{n-1} \sum_{i=1}^n (x_{i.} - \bar{x})(x_{i.} - \bar{x})^\top$
- estimate of mu: just use mean
  - $\hat{\mu} = \bar{x} = \frac{1}{n} \sum_{i=1}^n x_{i.}$

PCs based on data

- we need to estimate expectation vector and covariance matrix
  - can be done by sample mean and sample covariance matrix
- sample PCs are computed by
  - $Z = (X - \mathbf{1}\bar{x}^\top)\hat{\Gamma}$
  - where $\mathbf{1}$ is a column vector with 1s in it
- Z is of same dimension as data matrix X, and basically presents data in orthogonally rotated coordinate system
- the elements of Z are called **scores**

Number of PCs

- last few PCs with small variance are usually not too important, if we are interested in dimensionality reduction
- last few PCs might also only contain noise, which is unwanted anyway
- Variance can be counted by adding diagonal elements of A matrix (trace)
- we could do a statistical test to determine the last p-k PCs that are not significant
- we could also select k PCs which contained a pre-defined percentage of overall variance, e.g. 90%
- we could also exlude all PCs which have lower variance (eigenvalue) than the average
    - if data is standardized, the sum of eigenvalues is p, and average therefore 1
- or we could do a scree plot
    - exlcude PCs which follow about a linear trend



Singular Value Decomposition

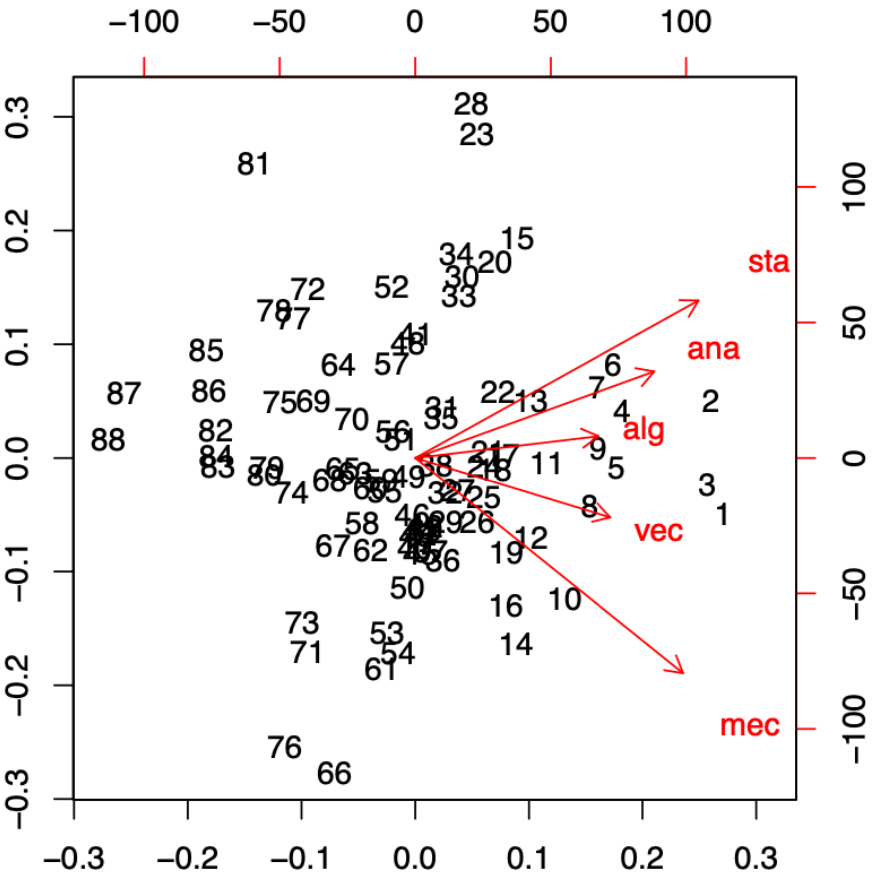
- SVD is alternative algorithm to get to PCs
- does not use covariance matrix, but directly data matrix X
- good, because if X is flat (n < p) covariance matrix will always have be singular (0s in some diagonals)
- SVD factorizes matrix into three Components
  - $X = UDV^\top$
    - U... p x p matrix, with orthogonal eigenvectors of $XX^\top$ as columns
    - D... n x p matrix with positive diagonal elements d_{ii} (called singular values of X) and the rest is zeroes; these are the roots of the eigenvalues to eigenvectors of both U and V
      - so d_{ii}^2 is the eigenvector to both $u_i u_i v_i$
    - V... n x n matrix, with orthogonal eigenvectors of $X^\top X$ as columns
- steps for PCA using SVD:
  1. **center data** (center columns of X)
  2. compute SVD components
  3. V are the principal components
  4. squared values of D are the explained variance
     a. proportional Variance of i-th PC: $\frac{d_{ii}^2}{\sum_j d_{jj}^2}$
  5. projected data (scores) are given by $Z = UD$

- SVD is numerically stable, since covariance matrix is not computed
- efficiently computable for sparse matrices

Biplots

- Biplot: show joint presentation of variables and observations
- helps visualize relationship between observations and variables
- biplot for PCA
  - first two PCs can be visualized in 2-dimensional biplot
  - the Axis are the PCs 1 and 2

- observations are the points in the plot
- variables are drawn as vectors, originating from origin
  - length and direction of vector shows the contribution to PCs
  - angle between vectors shows the correlation between variables
    - spitzer winkel → positive correlation
    - right angle → no correlation
    - stumper winkel → negative correlation



Diagnostics for PCA

- possible to detect outliers using PCA
- we have **score** and **orthogonal distances**
- **Score distance (SD):** Measures how extreme an observation is in the principal component space.

- **Orthogonal distance (OD):** Measures how far an observation is from the principal component subspace in the original space.
- big OD, small SD: vertical outlier
- big OD, big SD: bad leverage point
- small OD, big SD: good leverage point

Chapter 6 Factor Analysis

- FA wants to capture “latent variables”, which are underlying the data but not directly observable → these are called factors
  - e.g. factor intelligence is measured indirectly through attentiveness, knowledge,...
- it is assumed observations are linear combinations of these underlying factors + error
- similar to PCA dimensionality reduction is also goal
- but factors should be interpretable

Model definition

- **center and scale first**
- responses y can be explained by factors and error
  - k... number of factors
- $y = \Lambda f + e$
  - Λ... Loadings
  - e... error/unique factor
- Assumptions
  - $E(f) = 0$
  - $E(e) = 0, Cov(e_i, e_j) = 0$
  - $Cov(f, e) = 0$
  - $Var(f_i) = 1$
- therefore covariance matrix of error is diagonal

    - $Cov(e) = \Psi = Diag(\psi_{11}, \dots, \psi_{pp})$
- correlation matrix of the data can be reformulated as:
    - $\rho = Cor(x) = Cov(y) = Cov(\Lambda f + e) = \Lambda \Phi \Lambda^\top + \Psi$
- Φ is k x k matrix with correlations between factors
    - factors are uncorrelated
    - therefore $Cov(f) = \Phi = \boldsymbol{I}$
- so we get
    - $\rho = \Lambda \Lambda^\top + \Psi$
    - or $\rho_{reduced} = \Lambda \Lambda^\top = \rho - \Psi$
    - the diagonal elements $\kappa_i^2 = 1 - \psi_{ii}$ of $\Lambda \Lambda^\top$ are called communalitites
    - the communalitites describe the proportion of variance explained by factors

Uniqueness

- model can be reformulated with an additional k x k matrix inserted in model
- would still yield a valid model
- → but then the factor loadings cannot be uniquely determined
- is not a big problem, as factors will be rotated later anyways for better interpretation
- to obtain unique solution for now, we can constrain that $\Lambda^\top \Psi^{-1} \Lambda$ or $\Lambda^\top \Lambda$ is diagonal

Parameter estimation

- we can just estimate correlation matrix $\hat{\rho}$ from our data
- with that we can then we can also estimate loadings $\hat{\Lambda}$ and error variances $\hat{\Psi}$
    - they need to fulfill $\hat{\Lambda} \hat{\Psi}^{-1} \hat{\Lambda} = Diag$ or $\hat{\Lambda}^\top \hat{\Lambda} = Diag$
    - because of $\hat{\rho} = \hat{\Lambda} \hat{\Lambda}^\top + \hat{\Psi}$
- there is an upper bound for number k of factors, for which it is better to estimate factor model compared to correlation matrix
- → main job is to estimate Lambda and Psi
    - communalities can be estimated using e.g. the highest correlation coefficients
        - over-estimating: having uniquness in variance; under-estimating: having variance in the uniqueness

    - loadings can be estimated next using the estimated communalities, by reformulating factor model with spectral decomposition

Factor rotation

- because we constrained that $\Lambda^\top \Psi^{-1} \Lambda$ or $\Lambda^\top \Lambda$ have to be diagonal we have unique solution
- but this solution is not necessarily interpretable
- a rotation changes loadings, and therefore interpretation
- the goal is to rotate in a way, that loadings matrix is simple and interpretable
    - → loadings matrix should contain many small values and a few large ones (near -1 or 1)
    - these large values indicate strong contribution of a variable to a factor
- orthogonal rotation: minimum criterion
    - if we rotate so that the points are close to an “axis” in the rotated space
    - we consider the sum of all squared products of coordinates as criterion for simplicity
        - $\sum_{s<j=1}^{k} \sum_{i=1}^{p} (\lambda_{is} \lambda_{sj})^2 \to \min$
    - we want to minimize this
    - an orthogonal rotation like this does not change communalities
- oblique rotation: basically changing angle between axis
    - also has some criterions (quartimin, oblimin)
- orthogonal rotation may lead to better interpretability, but might also do nothing
- oblique rotation can always deliver good result

Estimation of factor scores

for both methods we consider loadings Λ and uniqueness Ψ as known

Weighted least squares

- goal is to minimize residuals between observed variables (y) and predicted values ($\Lambda f + e$)
- we have the inverted uniqueness matrix used as a weight

- this means that variables with smaller unique variances are weighted stronger, as they might be more reliable indicators of the factors

- we regress y on Lambda, with regression coefficients f
- $\hat{f} = (\Lambda^\top \Psi^{-1} \Lambda)^{-1} \Lambda^\top \Psi^{-1} y$
- or
- $\hat{F} = Y \Psi^{-1} \Lambda (\Lambda^\top \Psi^{-1} \Lambda)^{-1}$

Regression method

- goal is to maximize relationship between factors and observed variables
- we regress f on y
- $\hat{F} = Y R^{-1} \Lambda$

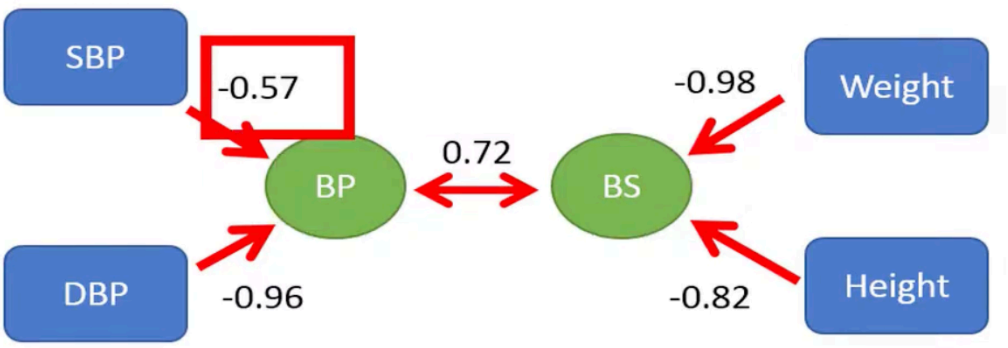
Chapter 7 Correlation Analysis

Multiple correlation analysis

- measure of dependency of feature y on p-dimensional feature **x**
- we assume both x and y are random variables with joint distribution
- the **multiple correlation coefficient (R)** shows correlation between y and the best linear prediction function
- for multiple linear regression the error is random, and the coefficients are chosen, so that the mean squared error (MSE) is minimal
  - so correlation between y and the best predictor function for y
  - $Corr(y, \beta_0 + \beta^\top x) = \rho_{y,x} = \sqrt{\frac{\sigma_{xy}^\top \Sigma_{xx}^{-1} \sigma_{xy}}{\sigma_{yy}}} = \sqrt{\frac{\beta^\top X^\top y}{y^\top y}}$
- **Coefficient of Determination (R^2)** the square of multiple correlation coefficient
  - it indicates how well feature y is explained by properties **x**

Canonical correlation analysis

- linear dependence between two groups of variables
- not one single correlation coefficient, but a subspace which describes dependence between groups
- best to center and scale data first → then coefficients for linear combinations can be compared
- we have random variables **x** (p-dimensional) and **y** (q-dimensional)
- $\phi = a^\top x$ → linear combination of x
- and $\eta = b^\top y$ → linear combination of y
- ϕ and η are called canonical variables, basically weighted aggregates of sets of variables
- canonical correlation coefficients ρ_k
  - are correlation coefficients between ϕ_k and η_k
  - ρ_k measures strenth of correlation between k-th canonical variables
  - canonical variables within each set are uncorrelated (orthogonal)
- Solution:
  - we first compute cross-covariance matrix $\Sigma_{xy} = Cov(x, y)$
  - and also compute Σ_{xx} and Σ_{yy}
  - now lets take
    - $\Sigma_x = \Sigma_{xx}^{-1}\Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}$ and $\Sigma_y = \Sigma_{yy}^{-1}\Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy}$
    - and then compute eigenvectors and eigenvalues of those
    - eigenvectors → weights of linear combinations
    - eigenvalues → squares of correlation coefficients
    - example from youtube:



    - BP & BS → canonical variables

- 0.72 → canonical correlation coefficient
- -0.57, -0.96, -0.98, -0.82 → weights (a and b) for linear combinations

Chapter 8 Discriminant Analysis

- we want to classify objects into groups
- and also have ability to put new objects to previously determined groups
- → we want a discriminant function that allows best possible separation

- x... observations
- π_1, π_2... populations 1 and 2
- R_1, R_2... space of observations of π_1 and π_2
- $\Omega = R_1 \cup R_2$
- f_1, f_2... probability functions, that observation is in group 1 or 2
- prior probabilities: probability that observation x belongs to group 1 or 2
- calculation of misclassification:

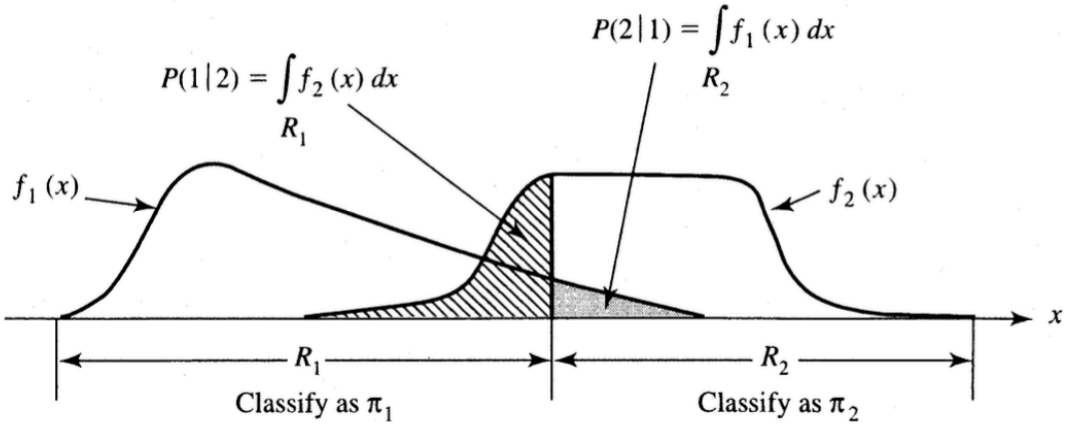


Figure 8.1: Probabilities of misclassification

- often there are also costs associated with wrong classification
- we can define expected costs of misclassification function
  - $ECM = c(2|1)P(2|1)p_1 + c(1|2)P(1|2)p_2$

    - reshape to
    - $ECM = \int_{R_1} [c(1|2)p_2 f_2(x) - c(2|1)p_1 f_1(x)]\, dx + c(2|1)p_1$
- the classification rule should keep ECM as low as possible
- R_1 is defined for x for which this applies:
    - $\frac{f_1(x)}{f_2(x)} \geq (\frac{c(1|2)}{c(2|1)})(\frac{p_2}{p_1})$
    - or
    - $c(1|2)p_2 f_2(x) \leq c(2|1)p_1 f_1(x)$
- R_2 is defined for x for which applies:
    - $\frac{f_1(x)}{f_2(x)} < (\frac{c(1|2)}{c(2|1)})(\frac{p_2}{p_1})$

- we could also define different criteria for classification rule
- for example R_1 and R_2 could be chosen so that Total probability of misclassification (TPM) is minimal; but this essentially leads to the same solution

Two-group case

- limited to multivariate normally distributed populations
- → f_1 and f_2 are density functions of multivariate normal distributions

Special case $\Sigma_1 = \Sigma_2 = \Sigma$

- usually the means and the covariance is not given
- therefore they have to be estimated from sample
- an observation x_0 is assigned to π_1 if
    - $(\mu_1 - \mu_2)^\top \Sigma^{-1} x_0 - \frac{1}{2}(\mu_1 - \mu_2)^\top \Sigma^{-1}(\mu_1 + \mu_2) \geq \ln\left(\frac{c(1|2)}{c(2|1)} \frac{p_2}{p_1}\right)$
- since we assume that $\Sigma_1 = \Sigma_2$, we can estimate one pooled covariance
    - $S_{pooled} = \frac{1}{n_1 + n_2 - 2} \sum_{l=1}^{2} \sum_{j=1}^{n_l} (x_{lj} - \bar{x}_l)(x_{lj} - \bar{x}_l)^\top$
    - this is an unbiased estimate of Sigma
- decision rule then simplifies to:
    - $(\bar{x}_1 - \bar{x}_2) S_{pooled}^{-1} x_0 - \frac{1}{2}(\bar{x}_1 - \bar{x}_2)^\top S_{pooled}^{-1}(\bar{x}_1 + \bar{x}_2) \geq ln(\frac{c(1|2)}{c(2|1)} \frac{p_2}{p_1})$

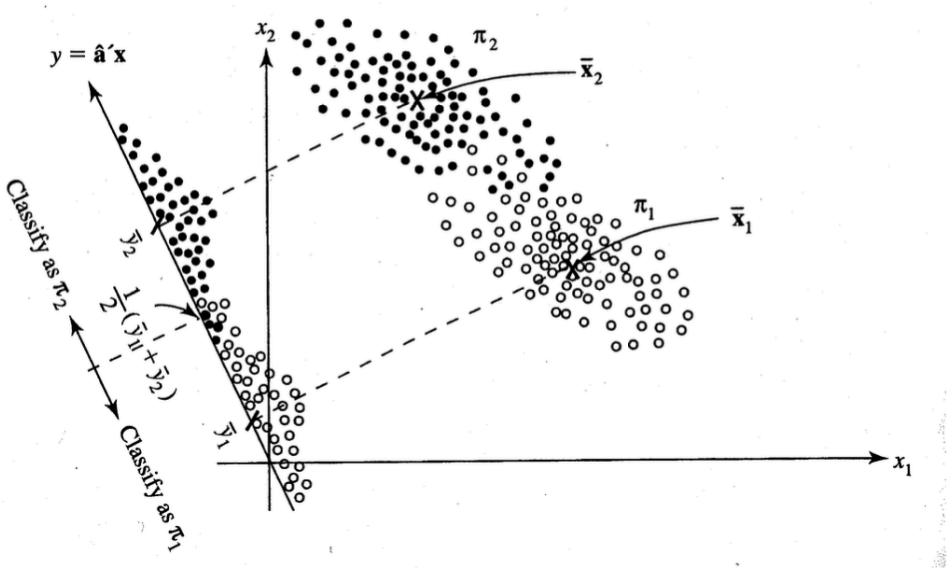
- if the right part of the decision rule is 0 ($\ln(1) = 0$), it simplifies to:
  - we have parameter y
    - $\bar{y}_1 = (\bar{x}_1 - \bar{x}_2)S^{-1}_{pooled}\bar{x}_1 = \hat{a}^\top \bar{x}_1$
    - $\bar{y}_2 = (\bar{x}_1 - \bar{x}_2)S^{-1}_{pooled}\bar{x}_2 = \hat{a}^\top \bar{x}_2$
  - and the midpoint $\hat{m} = \frac{1}{2}(\bar{x}_1 - \bar{x}_2)^\top S^{-1}_{pooled}(\bar{x}_1 + \bar{x}_2) = \frac{1}{2}(\bar{y}_1 + \bar{y}_2)$
  - if y value of observation is greater than or equals midpoint → group 1
  - y value is below midpoint → group 2
  - note: coefficient $\hat{a}$ is usually standardized to have length of 1
    - $\hat{a}^* = \frac{\hat{a}}{\sqrt{\hat{a}^\top \hat{a}}}$

Special case $\Sigma_1 \neq \Sigma_2$

- the classification rule ends up being
  - assign x_0 to π_1 if:
  - $-\frac{1}{2}(\Sigma_1^{-1} - \Sigma_2^{-1})x_0 + (\mu_1^\top \Sigma_1^{-1} - \mu_2^\top \Sigma_2^{-1})x_0 - k \geq ln(\frac{c(1|2)}{c(2|1)}\frac{p_2}{p_1})$
    - with $k = \frac{1}{2}\ln(\frac{|\Sigma_1|}{|\Sigma_2|}) + \frac{1}{2}(\mu_1^\top \Sigma_1^{-1}\mu_1 - \frac{1}{2}\mu_2^\top \Sigma_2^{-1}\mu_2)$
  - otherwise assign x_0 to π_2
- this rule is quadratic in x → qda
- unknown parameters μ, Σ need to be estimated

Fishers linear discriminant function

- his idea was to find a linear combination/direction $a \in \mathbb{R}^p$, from which values y could be derived
- a should be chosen so it **maximizes difference** between **arithmetic means** of groups y_1 and y_2
- visual explanation:
  - a projects values onto a line which best possible separates into two groups



- difference is expressed in standard deviation, and used as criterion for separation
  - $\frac{|\bar{y}_1-\bar{y}_2|}{s_y} \rightarrow \max$
- we want to find a which allows maximum separation of the sample means $\bar{y}_1$ and $\bar{y}_2$
- variance of the y-values is pooled
  - $s_y^2 = \frac{\sum_{j=1}^{n_1}(y_{1j}-\bar{y}_1)^2+\sum_{j=1}^{n_2}(y_{2j}-\bar{y}_2)^2}{n_1+n_2-2}$
- Solution: it turns out that
  - $\hat{y} = \hat{a}^\top x = (\bar{x}_1 - \bar{x}_2)S_{pooled}^{-1}x$
  - achieves this goal of maximizing difference
  - with midpoint $\hat{m} = \frac{1}{2}(\bar{x}_1 - \bar{x}_2)^\top S_{pooled}^{-1}(\bar{x}_1 + \bar{x}_2)$
  - if $\hat{y} \geq \hat{m}$ → assign to group 1
- we also have to assume same covariances because we used the pooled estimate

Multiple populations

ECM rule

- we get ECM

$$ECM = \sum_{i=1}^{g} p_i \left(\sum_{k=1, k\neq i}^{g} P(k|i)c(k|i) \right)$$

    - an optimal classification rule **minimizes** this expression
- the ECM is then minimized by assigning **x** to π_k for which the following expression is minimal
    - $\sum_{i=1,k\neq i}^{g} p_i f_i(x) c(k|i)$
- assumption from now on: costs for misclassification are the same
- this can be simplified to
    - observation x is assigned to π_k if
        - $p_k f_k(x) > p_i f_i(x) \qquad \text{for all } i \neq k$
    - these rules can only be applied if prior probabilities, misclassification costs and density functions are known

Classification under normal distribution

- we calculate the **quadratic discriminant values** $d_i^Q(x)$ for each group
    - $d_i^Q(x) = -\frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(x-\mu_i)^\top \Sigma_i^{-1}(x-\mu_i) + \ln p_i$
- x is assigned to group with highest quadratic discriminant value
- if the covariances are the same for all groups, quadratic discriminant values can be simplified to be linear in x (linear discriminant values)
    - $d_i(x) = \mu_i^\top \Sigma^{-1} x - \frac{1}{2}\mu_i^\top \Sigma^{-1}\mu_i + \ln p_i$
- pooled covariance matrix is estimated
- $S_{pooled} = \frac{1}{(\sum_{i=1}^{g} n_i) - g} \sum_{i=1}^{g}\sum_{j=1}^{n_i}(x_{ij}-\bar{x}_i)(x_{ij}-\bar{x}_i)^\top$

Fisher rule

refresher: g... number of groups, p... dimensions of data

- we again want to project data into a smaller (K - 1) dimensional space
- but this time multiple discriminant axes are needed, instead of only 1 for two-group case
- we again maximize between group scatter and minimizing within-group scatter
- assumption: Covariance of all groups are the same
- before we tried to maximize standardized differences of group means
    - group means written as random variables are written as expectation $\mu_{i,y}$

- when we also now consider the prior probabilities we get weighted mean
  - $\bar{\mu}_y = p_1\mu_{1,y} + p_2\mu_{2,y} = a^\top(p_1\mu_1 + p_2\mu_2)$

- we now have to maximize
  - $\frac{\sum_{i=1}^{g} p_i(\mu_{i,y} - \bar{\mu}_y)^2}{\sigma_y^2}$
  - with $\sigma_y^2 = a^\top \Sigma a$
  - also
- if group covariances are not equal, it would be best to use a pooled covariance
  - $W = \sum_{i=1}^{g} p_i \Sigma_i$
  - W... variation within groups
- Variation between Groups:
  - $B = \sum_{i=1}^{g} p_i(\mu_i - \bar{\mu})(\mu_i - \bar{\mu})^\top$
- maximisation problem can then be express as
  - $\frac{a^\top B a}{a^\top W a}$

- the solution to the maximization is given by the eigenvectors $a_1, \ldots, a_l$ of $W^{-1}B$, scaled so that $a_j^\top W a_j = 1$
- we have $l \leq \min(g-1, p)$
  - due to ranks of W and B
- now we can define fisher discriminant functions $y_j = a_j^\top x$ for every j = 1...l
  - this is the projection of x in direction j
- we have as classification rule the fisher discrimant values
  - $d_i^F(x) = \sum_{j=1}^{l}(y_j - \mu_{i,y_j})^2 - 2\log p_i$
  - measures the deviation of x to the i-th group mean, adjusted with prior probability
  - x is assigned to group with smallest discriminant value