Mutlivariate Statistics

Emanuel Nussli

June 2021

1 Introduction and Visualization

Fisher's z-test

Assume (X, Y) bivariate normally distributed with true correlation ρ. Let $z = 1/2 \log(\frac{1+r}{1-r})$ and $\xi = 1/2 \log(\frac{1+\rho}{1-\rho})$ such that for large n we have $\sqrt{n-3}(z-\xi) \sim \mathcal{N}(0,1)$

Different Plots

- Glyph plot: scatter plot where the plotting symbols vary according to a thrid variable (up to 5-10 variables by using colors, combinations from a data set that is not too large)
- star plot: length of spokes correspond to relative values of variables (scaled by $\frac{x_i - \min(x)}{\max(x) - \min(x)}$) and used for $n < 50, q < 10$
- Chernoff faces with the same idea as star plots but play into the human ability to differentiate faces
- parallel coordinates plot: all variables on vertical parallel coordinate axes so that the max and min of all variables are aligned. Values of one observation are connected.
- Conditioning plot: X, Y, Z where Z is divided into a number of intervals (if numerical) (might be overlapping). The panel above then is a scatterplot of Y versus X. Useful for big datasets but only 3 or 4 variables.

Categorical Data

- Mosaic Plot: area proportional to frequencies. Pearson chi-squared statistic: $X^2 = \sum_{i=1}^{I} \sum_{j=1}^{J} \frac{(n_{ij} - \mathbb{E}_{ij})^2}{\mathbb{E}_{ij}} = \sum_{i=1}^{I} \sum_{j=1}^{J} R_{ij}^2$ where $\mathbb{E}_{ij} = n \times \hat{\pi}_{ij}$ where R_{ij} is the Pearson residual which measures the contribution to misfit of each cell. We have $X^2 \sim \chi^2_{(I-1)(J-1)}$ asymptotically. The colors correspond to p-values for H_0 of $R_{ij} = 0$ being below 0.05 and 0.0001 respectively assuming $\mathcal{N}(0,1)$ but $Var(R_{ij}) < 1$

Multivariate Outliers

Squared Mahalanobis distance in $\mathbb{R}$: $D_M^2 = \frac{(x-\mu)^2}{\sigma^2}$ and in $\mathbb{R}^d \ \forall \ d \geq 2 : D_M^2 = (x-\mu)^T \sum^{-1} (x-\mu)$ while $D_M(x)$ is the distance of x from mean μ in standard deviations in direction of x. We have $D_M^2(x) \sim \chi^2(q)$ as the MD is just a sum of squares of q independent **normal random variables** has a chi-squared distribution with q degrees of freedom.

Masking Effect

Use robust estimators in order to prevent huge outliers from masking themselves by heavily influencing the estimate.

Practical Consideration

Even if data is not mutivariate normal, points with a large Mahalanobis distance are still potentially outliers. Automatic outlier detection is very dependent on model parameters but can be good for screening large amounts of data $\implies$ rather look at QQ-plots if possible. Generally: run analysis with and without the outliers and look if some change occurs and try to answer if the outliers is a data-processing error pr due to the DGP.

PC-Out

Automatically find outliers, based on robust PC, good in high dimensions, conservative (favors false positives over false negatives).

PCA

- first PC is in direction with maximum variance (or in direction that minimizes the orthogonal SSR).
- second PC is othogonal to the first one and directed in a way to maximize the variance of the points when projected onto the hyperplane (or SSR is minimal to the hyperplane)

$Y = a^t X (= X^T a)$ such that $Var(Y)$ is maximized and with a normalized linear combination, e.g $\sum_{j=1}^{q} a_j^2 = 1$ and $\mathbb{E}(X_j)_{j=1}^{q} = 0$

$$y_k = Xa_k \tag{1}$$
$$\mathbf{Y} = \mathbf{X}A \tag{2}$$

with $\dim(\mathbf{Y}) = n \times q$ being q pricipal components. $\dim(\mathbf{X}) = n \times q$ and $\dim(A) = q \times q$ with the vectors $a_k = (a_{1k}, ..., a_{qk})^T$ being called loadings and the values y_{ik} with $y_k = (y_{1k}, ..., y_{nk})^T$ being called the scores. Spectral Decomposition of sample covariance matrix S:

$$S = ADA^T \tag{3}$$

with $A, D \in \mathbb{R}^{q \times q}$ and $D = \text{diag}(\lambda_i)_{i=1}^{q}$ with λ_i being the eigenvalues of S and the columns of A being the eigenvectors and A being orthogonal s.t $A^T A = AA^T I$ and $Sa_k = \lambda_k a_k$

Algorithm:

1. Calculate Spectral Decomposition (princomp) of S and get eigenvalues and eigenvectors
2. Sort eigenvalues and corresponding eigenvectors s.t $\lambda_1 \geq ... \geq \lambda_q$
3. the k-th sample PC_k is given by $y_k = Xa_k$

while $\text{Var}(PC_k) = \lambda_k$ and all sample PCs being mean-zero and all y_k having pairwise zero sample correlation.

Dimension Reduction

$$Y = A^T X \iff Y^T = X^T A \iff X = AY \tag{4}$$

where we reduce the dimension if we only keep the k-largest principal components such that $X = AY = A_1 Y^{(1)} + A_2 Y^{(2)} \approx A_1 Y^{(1)}$ with $Y^{(1)} = (Y_1, ..., Y_k)^T$ such that $\dim(Y^{(1)} = k \leq q = \dim(X)$

Remarks

- sign of PC loadings is arbitrary as $S = ADA^T = (-A)D(-A)^T$
- if measurements are correlated, the **first** PC is often some kind of average of the measurements (look at first row, e.g y_1 and the weights of the variables) and the other PCs give important information about the remaining pattern. Often, it is the difference in the two variables.
- scaled data if (often) preferred (if on very different scales but makes all the variables equally important)
- using the correlation matrix is equal to using scaled variables that have mean 0 and variance 1.
- rules for how many PCs: at least 0.8 cum. variance, keep PCs with above average variance (if scaled data or corr. matrix: eigenvalue$_k \geq 1$, look at elbow plot to keep only the PCs before the drop
- new observations:
  1. new observation $x = (x_1, ..., x_q)^T$
  2. subtract mean $x - \hat{\mu}$
  3. calculate scalar product of loading of PC$_k$ with $\tilde{x}$ to get the coordinate in direction of PC$_k$ by $y_k = a_k^T \tilde{x}$ or $Y = A^T \tilde{x}$
- robust PCA as PCA is sensitive to outliers as the covariance and correlation matrix are sensitive to outliers which can be resolved via *covmat*() in *princomp*
- SVD is numerically more stable than the specatral decomposition
- does not rely on distributional assumptions, only considers linear combinations, reduction can only be achieved if variables are even correlated, PCA is **not scale invariant**, prcomp is better for large data than princomp

Use of PCA

- dimension reduction for graphical representations of the data
- as input for regression analysis to overcome collinearity
- find a one dimensional index the separates units best
- image compression, stock return, modeling of yield curve, NLP (bag-of-words model)

MDS

Input is the distance matrix D and the data matrix X is not needed and MDS can identify points up to shift rotation and reflection. Algorithm:

1. let $\mathbf{B} = \mathbf{X}\mathbf{X}^\mathbf{T}$ which we can solely calculate based on the distances D by $b_{ij} = -\frac{1}{2}(d_{ij}^2 - d_{i.}^2 - d_{.j}^2 + d_{..}^2)$ where $d_{i.}^2 = n^{-1}\sum_{j=1}^n d_{ij}^2$. We first center points to avoid shift invariance ($\sum_{i=1}^n x_{ik} = 0 \implies \sum_{iorj}^n b_{ij} = 0$)
2. we then calculate $B = V\Lambda V^T$ using the spectral decomposition (works as B is symmetric and positive semidefinite with $\dim(B) = \times n$) with $\Lambda = \text{diag}(\lambda_i)_{i=1}^n$ and V has normalized eigenvectors as columns. Drop the last $n - q$ eigenvalues and write $\Lambda = \text{diag}(\lambda_i)_{i=1}^q$
3. $\mathbf{X} = V_1\Lambda^{1/2}$ to get the points only from the distances between all points
4. we do not keep all non-zero eigenvalues but only $\tilde{q}$ of them to reduce the dimension: $\tilde{\mathbf{X}} = \tilde{V}_1\tilde{\Lambda}_1^{1/2}$ where we choose $\tilde{q}$ such the explained variance is at least about 0.8 or look at scree plot.
5. coordinates of unit i: values of i for the q variables

Remarks

- classical MDS minimizes $\sum_{i,j=1}^n (d_{ij} - \tilde{d}_{ij})^2$ where the first term is the euclidian distances and the second term the corresponding distances in the low-dimensional world.
- if d_{ij} are non-euclidian, we do not have $B = \mathbf{X}\mathbf{X}^T$ and B can be non-positive definite and some of its eigenvalues may be smaller than 0. If there are only few negative eigenvalues that are close to 0, then using the first few positive eigenvalues might still give a good representation and for choosing $\tilde{q}$, one can look at the cumulative proportion of absolute or squared eigenvalues.
- Sammon mapping: minimizes the stress function $\sum_{i,j=1}^n \frac{(d_{ij} - \tilde{d}_{ij})^2}{d_{ij}}$ which puts more emphasis on preserving small distances.
- relatively fast

Non-metric MDS

Preserve ranking of dissimilarities (that are on an ordinal scale) Algorithm:

1. finds a monotone transformation $f(\cdot)$
2. minimizes $S^2 = \frac{\sum_{i<j}(f(d_{ij}) - \tilde{d_{ij}})^2}{\sum_{i<j}\tilde{d_{ij}^2}}$ where the $\tilde{d}$ are euclidian distances. S^2 is called the stress criterion. The denominator is here to make sure we do not just get 0sas shrinking $f()$ gives lower values of S^2

Non-metric MDS has problems with converging to local optima by using isotonic (monotonic) regressions for point one and iterating between the two points. Trying different starting values is thus important but the procedure is time-consuming and yields a **non-unique solution**. The dimension is chosen by plotting S^2 versus $\tilde{q}$ and using the elbow method.

- S=20%: poor
- S=10%: fair
- S=5%: good
- S=0: perfect

Scaling the Data?

If not scaled:

- variable with largest range has most weight
- distances depend on scales of variables

Scale if:

- variables are on different units
- you want every variable to have equal weight

Variants of scaling:

$$x'_{ij} = \frac{x_{ij} - \hat{\mu}_j}{\hat{\sigma}_j} \tag{5}$$

$$x'_{ij} = \frac{x_{ij} - \min(x_{ij})}{\max(x_{ij}) - \min(x_{ij})} \tag{6}$$

Distances and Dissimilarities

Categorical data:

- binary
- nominal (red, blue, green) w.o ordering
- ordinal (1st,2nd,...)

A **distance** staisfies

$$d(i,j) \geq 0 \tag{7}$$

$$d(i,j) = 0 \iff i = j \tag{8}$$

$$d(i,j) = d(j,i) \tag{9}$$

$$d(i,j) \leq d(i,h) + d(h,j) \tag{10}$$

with the Minkowski distance being defined as

$$((x_{i1} - x_{j1})^p + ... + (x_{iq} - x_{jq})^p)^{1/p} \tag{11}$$

A **dissimilarity** satisfies all the above with the exception of the triangle inequality and they are thus more general. **Proximity** encompasses both similarity and dissimilarity. **Distances:**

- binary data: SMD (simple matching distance) $d(i,j) = \frac{b+c}{a+b+c+d}$ which is the proportional of variable sin which units disagree
- binary data: JD (jaccard distance) $d(i,j) = \frac{b+c}{a+b+c}$ which is the same as above but ignoring $(0,0)$ and is used where $0,1$ are not equally important
- nominal data: SMD $d(i,j) = \frac{m}{q}$ with m being the number of variables for which units mismatch
- ordinal data: normalized ranks with rank of variables $k = 1, ..., q$ given by $r_{ik} \in \{1, ..., M_k\}$ and normalized by $z_{ik} = \frac{r_{ik}}{M_{ik}}$ and then treating z_{ik} as a cont. variable.
- mixed data: Gower diss. $d(i,j)?q^{-1} \sum_{k=1}^{q} d_{ij}^{(k)}$ with $d_{ij}^{(k)} \in (0,1)$ where for binary and nominal data one uses a distance measure from above and for continous data one uses $d_{ij}^{(k)} = \frac{|x_{ik} - x_{jk}|}{R_k}$ with R_k being the range of variable k for all units and for ordinal data, one uses normalized ranks and then proceeds as with cont. data

ISOMAP (non-linear manifold learning)

Manifold learning: find structure and dimension of an unknown lower-dimensional manifold. Geodesic distance: shortest path on the manifold.
Algorithm:

1. find neighbors of each point (either $k \in \mathbb{N}$ or all $x \in \epsilon$) and form a weighted graph based on the neighbors where neighbors are connected and the weights of the edges are the euclidian distances
2. compute graph distances d_{ij}^G which is the summed up weights of the shortest path between all points (Floyd's algorithm)
3. Run classical MDS using D

Remarks

- Isometry: true geodesic distances on the manifold and the euclidian distances on the low-dimensional representation are equal
- the choice of the dimension (often 2 or 3) can be done using the elbow method (residual variance vs. dimension)
- can choose k, ϵ via cross validation. Start with the smallest neighborhood s.t the graph is connected and successively increase K/ϵ and check how the results change.
- ISOMAP is bad in large data sets, bad with non-convex (particularily ones with holes), strongly curved manifolds and noisy data (results in short circuits).

t-SNE

t-distributed stochastic neighborembedding for visualizing high-dimensional data. It captures well local structures and according to the inventors also global structures.
Algorithm:

1. define similarity of points x_i and x_j in original data using conditional gaussian probabilities where the similarity $p_{ij} = \frac{p_{j|i}+p_{i|j}}{2n}$
2. define similarity for y_i and y_j in the lower dimension by q_{ij} using the t-distribution. (Used to avoid crowing and optimization problems)
3. measure the mismatch between the similarities using the Kullback-Leibler divergence: $KL(P \mid\mid Q) = \sum_{i,j} p_{ij} \log(\frac{p_{ij}}{q_{ij}})$
4. the low-dimensional points are chosen such that KL is minimized, which is done via gradient descent

Tuning parameter is the perplexity parameter: smooth measure of effective number of neighbors. The σ_i^2 are chosen such that this number is approximately constant which controls trade-off between global and local structure where one should try values between 5 and 50.

Remarks

- no guarantee for global optimum
- cannot map new data to the low-dimensional space
- cluster sizes can be meaningless as well as distances between clusters (careful about overinterpretation)
- Gradient descent can be done via Barnes-Hut algorithm which approximates gradient. The larger θ, the faster but the worse the approximation. $\theta = 0$ yields the exact calculation.

Clustering (unsupervised learning)

To summarize data or the obtain new insights. With k clusters and n observations, there are k^n possible combinations.

Hierarchical Clustering

- agglomerative: start with individual observations
- divisive: start with the whole group (computationally bad)

Algorithm:

1. start with n clusters
2. find the nearest pair of distinct clusters C_i and C_j and let $C_i = C_i \cup C_j$ and decrease the number of clusters by one
3. repeat until there is only one cluster
   $\implies$ connected observations cannot be separated anymore and results depend on the distance measure

Distances between Clusters

- single linkage: $d_{AB} = \min_{i\in A, j\in B} d_{ij}$: for finding strechted out clusters
- complete linkage: $d_{AB} = \max_{i\in A, j\in B} d_{ij}$: for finding compact but not nec. well separated clusters
- average linkage: $d_{AB} = \frac{1}{n_A n_B} \sum_{i\in A} \sum_{j\in B} d_{ij}$: compromise between the first two
- Ward's method: $d_{AB} = WGSS_{AB} - (WGSS_A + WGSS_B)$ where $WGSS_C = \sum_{i\in C} \|x_i - \bar{x}_C\|^2$: join two clusters whose merger leads to the smallest increase in within-groups sum of squares, produces equal-sized and convex and compact clusters

Number of Clusters via Dendrogram

Find the largest vertical drop in the dendrogram but there is no strict rule.

Silhouette Plot

$S(i) = \frac{b(i)-a(i)}{\max(a(i),b(i))} \in [-1,1]$ indicates how well each observation i fits to its cluster. $a(i)$ is the average diss. between i and the other points in the cluster while $b(i)$ is the lowest average dissimilarity between i and all points from **any** other cluster. Big S means good clustering while negative S means wrong cluster while $S(i) > 0.5$ is acceptable.

Interpretation of Clusters

1. look at position of cluster centers or representatives (better to look at raw data than scaled data if data was scaled for interpretability)
2. Apply a dimension reduction technique such as PCA or t-SNE and plot the reduced dimensional data and label/color the points according to the cluster they belong to

Partitioning methods

k-means

Exact solution is computationally infeasible (minimize $WGSS = \sum_{m=1}^{k} \sum_{i:C(i)=m} \|x_i - \bar{x}_m\|^2$) but algorithm:

1. start with k centers for k clusters (fixed in advance)
2. assign each data point to the nearest center (euclidian distance)
3. recompute the center with the new center being the average over all data points in that cluster
4. repeat until the centers do not change anymore

$\implies$ no guarantee for global optima and dependent on starting values. Run mutiple times with random starting values to avoid local optima. Choice of number of clusters can be done via plotting $WGSS$ against the number of clusters and use the elbow technique.

Partitioning Around Medoids

Cluster centers are observations (k-medoid) while the medoid is the object of a cluster whose average distance/diss. to all the objects in the cluster is minimal. More robust, deals with any distance or dissimilarity, finds representative objects per cluster, comp. more expensive and dependent on starting values. PAM is the most common method in the class of **k-medoids**.

Scale

Results from all the clustering algorithms depend on the scale which is why one might want to scale (or not). Similar as in PCA or MDS.

Model-Based Clustering (GMM)

Based on a statistical model, not based on heuristic arguments concerning distances/dissimilarities.

Gaussian Mixture Models GMM

let $f(x; p, \theta) = \sum_{j=1}^{k} p_j g_j(x; \theta_j)$
s.t we have k different Gaussian distributions while p_j is a prior of a point belonging to that cluster with $\sum_{j=1}^{k} p_j = 1$ and $g_j(x; \theta_j)$ being the Gaussian pdf of x given x lies in cluster j

Algorithm:

1. find k, p_j and θ_j using the expectation-maximization algorithm
2. assign x_i to cluster j such that $\mathbb{P}(x \in cluster_j \mid X = x) = \frac{p_j g_j(x_i;\theta_j)}{f(x_i;p,\theta)}$ is maximized (Bayes Theorem)

That means we have a mean vector and a covariance matrix **per cluster**. If one did not restict the covariance matrix of different clusters, one would have to estimate a lot of parameters. We thus impose restriction on $\sum_j$ the volume, shape and orientation of the covariance matrices. GMMs are fitted via a difficult maximum likelihood problem. Model selection via $BIC = -2\log -likelihood + p\log(n)$ where $p :=$ # parameters. Lower is better. $MClust$ calculates the negative BIC: higher is better.

Density-Based Clustering (DBSCAN)

Informal assumption: Clusters are areas of high density that are separated from another by areas of low density. Can find cluster of **arbitrary shape** and is a **robust technique**. Tuning parameters are the neighborhood ϵ and the minimum number of points in neighborhood $MinPts$. Let's define:

- neighborhood: all points with a distance less than ϵ from that point
- Core point: it's neighborhood contains at least $MinPts$
- Border point: not a core point but in the neighborhood of a core point
- Noise point: Neither a core nor a border point

Algorithm:

1. label all points according to the labels from above
2. Group core points: add an edge between core points that are within ϵ of each other and assign all connected core points to the same cluster
3. assign each border point to one of the clusters of its associated core points (non-unique).

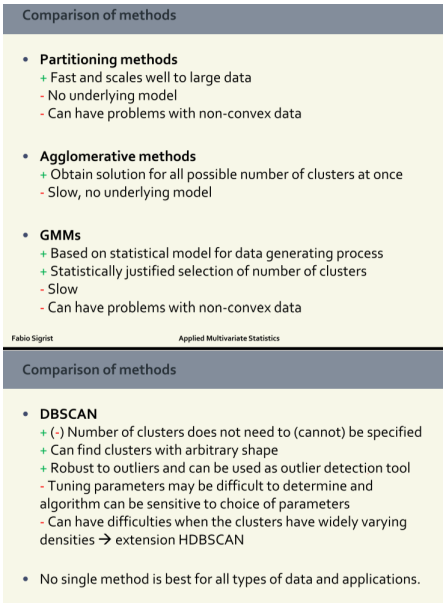
Is implemented a little differently to reduce comp. costs. Works for **all distance functions** and can be used to detect outliers.

Choice of Tuning Parameters for DBSCAN

k-NN distance: distance from a point to its k-nearest neighbor.

1. pick an initial k where usually $k \geq d + 1$ where d is the dimension of the data. For $d = 2, k = 4$ often works well
2. compute k-NN distance for each point
3. sort in increasing order and plot them
4. look for shape increase and choose this distance as ϵ and k as $MinPts$. Change k if there is no elbow

Comparison



Curse of Dimensionality

Under assumptions:

$$\lim_{q \to \infty} \frac{\max(Dist) - \min(Dist)}{\min(Dist)} = 0 \tag{12}$$

Factor Analysis

The variables of interest are not measureable directly - they are called **latent variables** or **common factors**. One measures variables that are indicators of the latent variables, and we call those **manifest variables** or **observed**. Factor analysis wants to find the latent variables, study the relationsship of the latent and observed variables and often wants to do dimension reduction (we only do exploratory factor analysis).

$$\mathbf{X} = \mathbf{\Lambda f} + \mathbf{u} \tag{13}$$

where $\mathbf{X}, \mathbf{u} \in \mathbb{R}^q$ and $\mathbf{f} \in \mathbb{R}^k$ and $\dim(\mathbf{\Lambda}) = q \times k$ being a deterministic matrix containing the loadings. The other components are random. We have the following assumptions:

- $\mathbb{E}(u) = 0$ and $\mathrm{Cov}(u) = \mathrm{diag}(\Psi)$
- $\mathrm{Cov}(f_l, u_j) = 0$: gnauer ahluege
- $\mathbb{E}(X) = 0$ and $\mathbb{E}(f) = 0, \mathrm{Cov}(f) = I$ meaning we have standardized and uncorrelated latent factors

so the values of the observed variables are independent, given the latent factors. We have

$$\mathrm{Cov}(X) = \mathrm{Cov}(\Lambda f + u) = \Lambda\Lambda^T + \Psi \tag{14}$$

$$\sigma_j^2 = \mathrm{Var}(X_j) = \sum_{l=1}^{k} \lambda_{jl}^2 + \psi_j \tag{15}$$

where the sum is called the communality, i.e the variance that results from the latent factors and ψ_j is the specific or unique variance that does not result from the latent factors. We especially have:

$$X = \Lambda f + u \iff \Sigma = \Lambda\Lambda^T + \Psi \tag{16}$$

Scale Invariance

Let $Y = (c_1 X_1, ..., c_q X_q)$ be a scaled version of $\mathbf{X}$. We then say that X, Y admit the same factor model as it does not matter if we scale the data first and then apply the factor model or if we apply the factor model on $\mathbf{X}$ and then scale the loadings and the specific variances. We assume that we use scaled variables $(\frac{X_1}{\sigma_1}, ..., \frac{X_q}{\sigma_q})$ which is the same as using the correlation matrix instead of $\sum$.

Estimation: Principal Factor Analysis

First estimate the sample covariance matrix $\Sigma 5$. We then want to find Ψ via

1. Get an initial estimate for Ψ via $\sigma_j \overset{scaled}{=} 1 = \sum_{l=1}^{k} \lambda_{jl}^2 + \psi_j$ while one can obtain an estimate for the communality via the square of the multiple correlation coefficient of the j-th variable with all the others **or** the largest correlation coefficient between the j-th variable and one of the others
2. using Ψ_1, determine Λ_1 via $\sum - \Psi_1 = \Lambda_1 \Lambda_1^T$ via the spectral decomposition: $\sum - \Psi_1 = A\Omega A^T$ while only keeping the k largest eigenvalues and eigenvectors A_1, Ω_1. Set $\Lambda = A_1 \Omega_1^{1/2}$
3. obtain Ψ_2 as $\Psi_2 = \sum - \Lambda_1 \Lambda_1^T$
4. repeat until convergence $\implies$ do all with correlation matrix

Estimation: Maximum Likelihood

Assume $X = (X_1, ..., X_q)^T \sim \mathcal{N}(\mu, \sum)$ and

$$(\hat{\Psi}, \hat{\Lambda}) = \mathrm{argmax}_{\Psi, \Lambda} \left(-\frac{n}{2} (\log(|2\pi(\Lambda\Lambda^T + \Psi)|) + \mathrm{trace}((\Lambda\Lambda^T + \Psi)^{-1} S)) \right) \tag{17}$$

which is done iteratively (factanal in R). One can test whether the number of factors is sufficient using the MLE approach.

Heywood Cases

It can happen that $\hat{\psi}_j < 0$ or $\hat{\psi}_j > 1$ which makes no sense as a variance must be positive and $1 = h_j^2 + \psi_j$ meaning ψ_j cannot exceed 1 $\implies$ collect more data, use a different model, try a different estimation technique.

Number of Factors

H_0 : k-factor model holds true
H_A : Σ is more general meaning k factors are not sufficient
One starts with a small value of k and increases successively until the test does not reject but its possible that the test always rejects (indication that the model fit is bad or that sample size is large)

Interpretation of Factor Loadings

let $MM^T = I$ and transform $f^\star = M^T f$ and $\Lambda^\star = \Lambda M$. We can then show

$$X^\star = \Lambda^\star f^\star + u = (\Lambda M)(M^T f) + u = \Lambda F + u = X \tag{18}$$

$$\Sigma^\star = \Lambda^\star \Lambda^{\star T} + \Psi = (\Lambda M)(\Lambda M)^T + \Psi = \Sigma \tag{19}$$

which means that factor models are not unique. Estimation procedures impose restrictions on the model parametrization in order to make the factor model unique. And after obtaining a solution, one can change the loadings using rotations to facilitate interpretability. One can rotate the model to find a rotation that allows an interpretation one likes (bad) but the rotation does not change the model (pro).

Rotations

- orthogonal rotation: the factors are restricted to be uncorrelated. We have that the factor loadings are the covariances between factors and observed variables, i.e $\text{Cov}(X, f) = \Lambda$. One example is the **varimax** rotation (most popular) which makes each factor have few large and many small loadings.
- oblique rotation: the factors may be correlated. Uncorrelated factors are maybe unrealistic which is why one might obtain better interpretable factors once dropping this assumption. One example is the **Promax** rotation which wants low correlation between factors.

Factor Scores

Many very similar methods with small differences. The factors are not observed so we now want to predict them and we call the predicted values the factor scores. Thompsons's method assumes that the X, f are jointly Gaussian and via MLE we get:

$$\hat{f}_i = (\hat{f}_{i1}, ..., \hat{f}_{ik})^T = \hat{\Lambda}^T \hat{\Sigma}^{-1} x_i \tag{20}$$

Remarks

- Does not work if variables are almost uncorrelated: PCA has nothing to explain meaning $\psi_j = 1 \; \forall \; j \in J$
- PCA and Factor Analysis give similar results if the specific variances are small. If they are zero, then PCA and FA are the same (up tp scaling)
- FA is scale **invariant**
- considering $k + 1$ factors instead of k factors may change the first k factors

Classification

$Y \in \{1, ..., k\}$ such there are k classes or groups and the predictors can be quantitative or categorical.

LDA & QDA

The unconditional distribution of X is a Gaussian mixture with density

$$\sum_{j=1}^{k} p_j g_j(x; \theta_j) \tag{21}$$

$$g_j(x; \theta_j) = \frac{1}{(2\pi)^{p/2} |\Sigma_j|^{1/2}} exp\left(-\frac{1}{2} (x - \mu_j)^T \Sigma_j^{-1} (x - \mu_j) \right) \tag{22}$$

Use Bayes Th. to get $\mathbb{P}(Y = j \mid X = x)$ using a prior probability $p_j = \mathbb{P}(Y = j)$ and a conditional distribution $X \mid Y = j \sim \mathcal{N}(\mu_j \Sigma_j)$. One gets the posterior

$$\mathbb{P}(Y = j \mid X = x) = \frac{p_j g_j(x; \theta_j)}{\sum_{j=1}^{k} p_j g_j(x; \theta_j)} = \frac{\mathbb{P}(\{Y = j\} \cap \{X = x\})}{\mathbb{P}(X = x)} \tag{23}$$

We maximize the logarithm of this expression to find class j.

- QDA assigns $X = x$ to the class j for which $\delta_j(x) = \log(p_j) - 0.5\log(|\Sigma_j|) - 0.5(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j)$ is maximized.$X = x$ for which $\delta_j(x) = \delta_i(x)$ are called decision boundaries. The term $\frac{p}{2}log(2\pi)$ is dropped as it is the same for all j.
- LDA assumes equal covariance matrices for the different classes which yields $\delta_j(x) = \log(p_j) + x^T\Sigma^{-1}\mu_j - 0.5\mu_j^T\Sigma^{-1}\mu_j$ as the common term $-0.5x^T\Sigma^{-1}x$ has been dropped as it does not depend on j.

Parameter Estimation

p_j is the fraction of observations that belong to class j, μ_j is the sample mean of all observations belonging to j and Σ_j is the sample covariance matrix S_j of all observations belonging to j

Naive Bayes

Special Case of QDA meaning

$$X \mid Y = j \sim \mathcal{N}(\mu_j, \text{diag}(\sigma_{1j}^2, ..., \sigma_{pj}^2)) \tag{24}$$

meaning the Xs are independent, conditional on $Y = j$.

Fisher‘s Disciminant Analysis

Find $U = a^TX$ which maximizes the ratio of between-class (BCV) variance to the within-class (WCV) variance (rayleigh quotient).
Algorithm:

1. Find linear combination a^TX which satisfies the above statement. More formally: $\frac{BCV}{WCV} = \frac{a^TBa}{a^TWa}$ with $B = \sum_{j=1}^k \frac{n_j}{n}(\bar{x_j}-\bar{x})(\bar{x_j}-\bar{x})^T$ being the between-group sample covariance matrix and $W = n^{-1}\sum_{j=1}^k \sum_{i:C(i)=j}(x_i - \bar{x}_j)(x_i-\bar{x}_j)^T$. The vector a that maximizes that expression is given by the eigenvector of W^{-1} corresponding to the largest eigenvalue. For two groups, we have $a = W^{-1}(\bar{x}_1 - \bar{x}_2)$. k is the number of groups.
2. compute average score $a^T\bar{x}_j$ for each group $j = 1, ..., k$
3. compute the score a^Tx_{new} for a new observation
4. classify the x_{new} into the group j for which $|a^Tx_{new} - a^T\bar{x}_j| < |a^{tT}x_{new} - a^T\bar{x}'_j| \; \forall \; j' \neq j$

Remarks

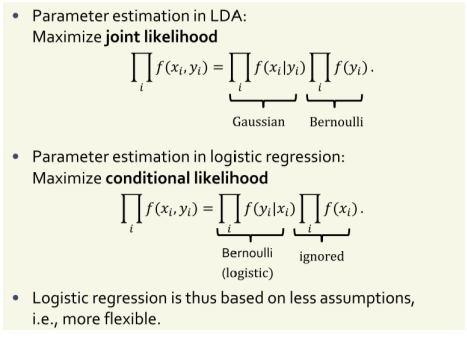
- for two groups from two mutivariate Gaussians with equal covariance matrices, Fisher‘s Discriminant Analysis is equivalent to LDA (with equal prior probabilities)

Logistic Regression

$\mathbb{P}(Y = 1 \mid X = x) = F_\beta(x)$ where $F_\beta = \frac{1}{1+exp(-(x^T\beta))} = \frac{exp(x^T\beta)}{1+exp(x^T\beta)}$ which is called the logistic function or the inverse logit function. That is derived from the log-odds:

$$\log\left(\frac{\mathbb{P}(Y=1 \mid X=x)}{\mathbb{P}(Y=0 \mid X=x)}\right) = x^T\beta = \log\left(\frac{\mathbb{P}(Y=1 \mid X=x)}{1-\mathbb{P}(Y=1 \mid X=x)}\right) \tag{25}$$

For LDA, we have the log-odds given by $\log\left(\frac{p_0}{p_1}\right) - 0.5(\mu_0+\mu_1)^T\Sigma^{-1}(\mu_1-\mu_0) + x^T\Sigma^{-1}(\mu_1-\mu_0) = \alpha_0 + x^T\alpha$ as we have for the logreg.



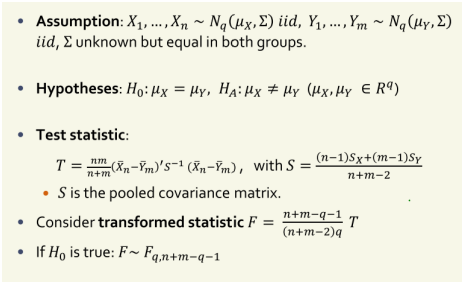
Evaluation

ROC-Curve

Plot true positive rate against false positive rate for various thresholds while threshold is the value one chooses for $\hat{\mathbb{P}}(Y = 1 \mid X = x) > \delta$ and the AUC is the area under the curve which should be maximized. False/ True is always in reference to the underlying mechanisms ($truth = 0$) or ($truth = 1$) and positive/negative is in reference to the prediction.

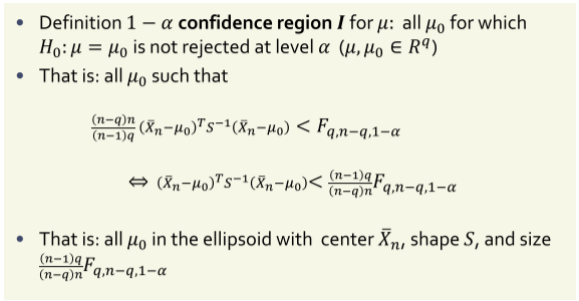
Extending Univariate Methods

- Univariate one-sample test with $H_0 : \mu = \mu_0$ gives $T = \sqrt{n}\frac{\bar{X}_n - \mu_0}{s} \overset{H_0}{\sim} t_{n-1}$ with $p = \mathbb{P}(|T| \geq |t|) = 2\mathbb{P}(Z \geq |z|)$ as the t-dist. is symmetric or calculate critical value $t_{n-1,1-\alpha/2}$
- Hotelling's one-sample T^2-test with same H_0 but multivariate gives $T = n(\bar{X}_n - \mu_0)^T S^{-1}(\bar{X}_n - \mu_0)$ and $F = \frac{n-q}{(n-1)q}T$ where $\mu \in \mathbb{R}^q$ and under $H_0 : F \sim F(q, n-q)$ where in general $F_{m,n} = \frac{\chi^2_m/m}{\chi^2_n/n}$ where the χ^2 distributions are independent. Reject if $\mathbb{P}(F > f)$ or critical value $F_{q,n-q,1-\alpha}$. One could work with Hotelling's T-distribution but converting to a F-distribution is easier.
- univariate two-sample t-test: the groups have different means (X has n observations Y has m observations) μ_x, μ_y but the same variance which is unknown as before. We test $H_0 : \mu_x = \mu_Y$ vs. $H_A : \mu_X \neq \mu_Y$ which yields $T = \frac{\bar{X}_n - \bar{Y}_m}{\hat{\sigma}_{\bar{X}_n - \bar{Y}_m}} = \sqrt{\frac{nm}{n+m}}\frac{\bar{X}_n - \bar{Y}_m}{s_p}$ with s_p being the pooled std given by $s_p^2 = \frac{(n-1)s_x^2 + (m-1)s_Y^2}{n+m-2}$ which follows (under the Null) as t-distribution with $n + m - 2$ degrees of freedom.
- Hotelling's two-sample T^2-test: Same as before but mutivariate:



Confidence Regions

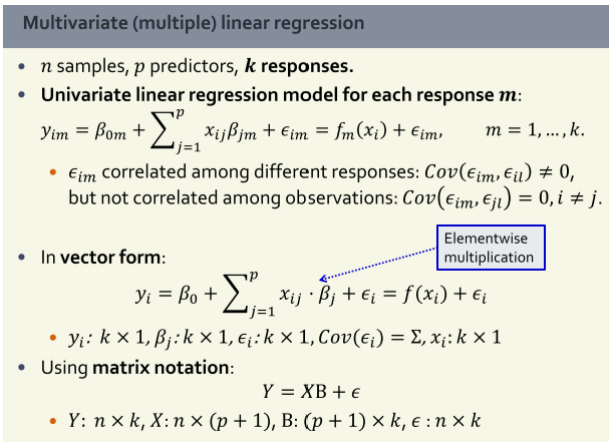
- Confidence interval: all μ_0 for which $H_0 : \mu = \mu_0$ is not rejected at level α : $|\sqrt{n}\frac{\bar{X}_n - \mu_0}{s}| < t_{n-1m1-\alpha/2}$ which we alter to get $I = \bar{X}_n \pm \frac{s}{\sqrt{n}} t_{n-1,1-\alpha/2}$ such that we have $\mathbb{P}(\mu \in I) = 1 - \alpha$.
- the same goes for multivariate data:



MANOVA

- ANOVA: are the means in x groups the same by comparing within-group to between-group variation and assumes normality (for small sample sizes). p-values can be computed via F-test.
- MANOVA: equal mean vectors possible? Are the mutivariate means in x groups the same? Also compares within to between and there are various test statistics. Also assumes normality in small sample sizes and p-values can be computed using the Wilks-test.

Multivariate (mutiple) Regression



Multiple Testing

$$FWER = \mathbb{P}(V \geq 1) = \mathbb{P}(\text{getting at least one false positive}) \overset{FDA}{<} 5\% \tag{26}$$

Probability of seeing a significant result purely by chance is α for one test but for m independent test it is $1-(1-\alpha)^m$ which can be much larger than α. One can do mutivariate tests or correct for multiple testing.

- Bonferroni method: $p_i \leq \frac{\alpha}{m}$ with very low power (very conservative) but $FWER \leq \alpha$
- Holm-Bonferroni:
  1. sort all p-values in increasing order
  2. for $i = 1, 2, ...,$ if $(m+1-i)p_{(i)} \leq \alpha \iff p_{(i)} \leq \frac{\alpha}{m+1-i}$ reject $H_{0(i)}$ and carry on with $i+1$. If at same point $H_{0(i)}$ cannot be rejected, stop the procedure. $FWER \leq \alpha$ but has never worse power than Bonferroni method but still conservative.
- $FDR = \mathbb{E}(\text{false positives} : \text{total amount of significant results})$ with $FDR = 0.1$ being accpetable for screening. We can control FDR instead of FWER to be less conservative
- Benjamin-Hochberg method to control FDR:
  1. sort p-values in increasing order
  2. Choose the largest i such that $\frac{m}{i}p_{(i)} \leq q \iff p_{(i)} \leq \frac{i}{m}q$ where q is the FDR. Reject all the hypothesis before i but not the ones coming after.

FWER controlling for confirmatory analysis, FDR for exploratory and screening stuff.

Trees and Random Forest

Regression Trees

- **Missclassification rate** in Yes class: How many of the yes-class were falsely classified.
- restrict to recursive binary splits which is a greedy algorithm

Regression tree:

$$f(x) = \sum_{m=1}^{M} c_m 1_{(x \in R_m)} \tag{27}$$

where $P = \{R_1, ..., R_M\}$ is a binary partition of $\mathbb{R}^p$ and each cell R_m is a rectangle of the form $(u_1, l_1] \times ... \times (u_p, l_p] \subset \mathbb{R}^p$ with $-\infty \leq l_j \leq u_j \leq \infty$.

Algorithm:

1. Use a greedy algorithm to grow a large tree: repeat the splitting step of each the resulting regions but only split one region (the one for which splitting yields the lowest RSS). **Remarks**: Search over s on a discretized set (e.g quantiles or midpoints between observations) and stop splitting when some minimal node size (observations per node) is reached. Note that T is the number of end-nodes.
2. apply cost complexity pruning to obtain the best subtree as a function of α where the criterion is given by $C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|$ where N_m is the number of observations in one partition and $Q_m(T) = N_m^{-1} \sum_{i:x_i \in R_m} (y_i - \hat{c}_m)^2$ is an impurity measure which is given by the MSE for continous variables.

3. Use Cross-Validation to obtain α (Test and Training set and evaluate the MSE on the test set as a function of α and pick α such that average out of sample error on all test sets is minimized)
4. return the subtree from 2. corresponding to the chosen α

Classification Trees

Two important measures:

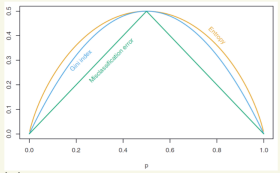
$$\hat{p}_{mk} = N_m^{-1} \sum_{i:x_i \in R_m} 1_{(y_i = k)} \tag{28}$$

$$k(m) = \text{argmax}_k \, \hat{p}_{mk} \tag{29}$$

There is no MSE so one uses

- **Misclassification error (MCE):** $\frac{1}{N_m} \sum_{i:x_i \in R_m} 1_{(y_i \neq k(m))} = 1 - \hat{p}_{mk(m)}$
- **Gini index:** $\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$
- **Cross-entropy** or **deviance:** $-\sum_{k=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk})$

- Often, one uses the Gini index for partition building and the misclassification error for pruning.
- Illustration of impurity measure as a function of p for two classes:



Random Forests

Build B identically distributed, deep trees having low bias but high variance and take the average of those trees to make the variance smaller and the bias remains the same for the individual trees.

Algorithm:

1. for $b = 1$ to B:
   (a) draw a bootstrap sample $\mathbf{Z}^\star$ of size N from the training data
   (b) grow a random forest tree T_b to the bootstrapped data by executing the following steps until n_{min} is reached
      i. select $m \approx \sqrt{p} \approx \frac{p}{3}$ at random from the p variables
      ii. pick the best split-variable among the m
      iii. split the node into two daughter nodes
2. Output the ensemble of trees $\{T_b\}_{b=1}^B$
3. For Regression: $\hat{f}_{rf}^B(x) = B^{-1} \sum_{i=1}^B T_b(x)$ and for Classification: $\hat{C}_{rf}^B = k(m)$

- The trees are not independent but $\text{Var}\left(\frac{1}{B} \sum_{i=1}^B T_i(x)\right) = \rho\sigma^2 + \sigma^2 \frac{1-\rho}{B}$ where ρ decreases as m decreases.
- **OOB error**: Use the observations not used due to bootstrapping and evaluate them on all the trees and take an average over all of them.
- variable importance via permutations:
  - calculate the OOB error e_b for all the trees
  - randomly permute the values of variable i in the OOB sample and calculate the OOB error p_b again
  - average the decrease in accuracy $d_b = p_b - e_b$ over all trees
  - $s_d^2 = (B-1)^{-1} \sum_{t=1}^B (d_t - \bar{d})^2$ and $v_i = \frac{\bar{d}}{s_d}$ is the variable importance
- variable importance via impurity: record the total amount of decrease in the impurity measure (RSS, Gini) due to splits over variable i and average over all trees

Analysis of Repeated Measures Data

Observations are not independent as we observe multiple observations for the same unit (rep. measures, longitudinal, panel data)

Fixed Effects Model

$$y_{it} = \beta_{0i} + \beta_1 x_t + \epsilon_{it} \tag{30}$$

with the disadvantages of having to estimate a lot of intercepts which is bad if we have litte n but large I.

Random Effects Model

$$y_{it} = (\beta_0 + u_i) + \beta_1 x_t + \epsilon_{it} \tag{31}$$

where ϵ_{it} and u_i are independent and parameters $\beta_0, \beta_1, \sigma, \sigma_1$ and which is called mixed effects model as there are random and fixed effects. It has the advantage that it has **information pooling** as the specific intercepts are drawn towards β_0 as they are cconstrained by $u_i \sim \mathcal{N}(0, \sigma_u^2)$ which allows for a more efficient estimation of β_1 which is very good for little n.
We can easily show:

- $\text{Var}(y_{it}) = \sigma^2 + \sigma_u^2$
- $\text{Cov}(y_{it}, y_{it'}) = \sigma_u^2 \forall\ t \neq t'$
- $\text{Cov}(y_{it}, y_{lt\prime}) = 0$

which means (cond. on x) that for the same unit, samples are correlated but not between units (cond. on x) and that this correlation is constant over time.

Random Intercept and Random Slope Model

$$y_{it} = (\beta_0 + u_{i1}) + (\beta_1 + u_{i2}) x_t + \epsilon_{it} \tag{32}$$

with $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and $u_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ and u_i and ϵ_{it} being independent and parameters $\beta_0, \beta_1, \sigma, \Sigma$
We can show easily

- $\text{Var}(y_{it}) = \sigma^2 + \sigma_1^2 + 2\sigma_{12} x_t + \sigma^2 x_t^2$
- $\text{Cov}(y_{it}, y_{it'}) = \sigma_1^2 + \sigma_{12}(x_t + x_{t'}) + \sigma_2^2 x_t x_{t'}$
- $\text{Cov}(y_{it}, y_{lt'}) = 0 \forall\ l \neq i$

which means that the unit-wide correlations can change (more complex).

Estimation and Model Checking

- ML: variances are biased but tests between two models with differing fixed and random effects are possible
- REML (restricted): variance unbiased but only test between two models with the same **fixed effects**
- get p-values etc. via asymptotic theory

Predict Random Effects

Not estimated as part of the model but we can estimate them using the estimated model via BLUP which minimize the MSE and are a linear combination of the responses y_{it}

Model Checking

Residual analysis via Tukey-Anscombe (r vs. $\hat{y}$) and QQ plots of residuals and additionally QQ-plot for predicted random effects.