STAT215A Midterm review

Rebecca Barter

November 19, 2014

1 Causal Inference

Definition (**Causal effect**). A Causal effect is defined as the difference between two potential outcomes, but when only one of the two potential outcomes is observed.

Causal inferences are made from *observational studies, natural experiments*, and *randomized controlled experiments*. When using observational data to make causal inferences, the key problem is *confounding*, however, in medicine and social science, causal inferences are most solid when based on randomized controlled experiments, where investigators assign subjects at random to a *treatment group* or *control group*. Up to random error, the two groups are the same with respect to all relevcant factors other than treatment. Differences between the treatment group and the control group are therefore due to treatment.
In an observational study, it is the subjects who assign themselves to different groups. However, in such circumstances, there may be some hidden confounding factor which, for example makes people smoke and also makes them sick.

Definition (**Confounding**). Confounding means a difference between the treatment and control groups - other than the treatment - which affects the response being studied.

1.1 Neyman-Rubin model

The Neyman-Rubin model is an approach to the analysis of cause and effect base don the framework of potential outcomes. The causal aspect of the model lies in the mechanism by which treatment is assigned. For observational studies, one relies on the assumption that the *assignment of treatment* can be treated as if it were random.
The basic setup of the Neyman model is simple. Let Y_{i1} denote the potential outcome for unit i if the unit receives treatment, and let Y_{i0} denote the potential outcome for unit i in the control regime. The treatment effect for observation i is defined by $\tau_i = Y_{i1} - Y_{i0}$. Causal inference is a missing data problem because Y_{i0} and Y_{i1} are never *both* observed.
Some assumptions have to be made to make progress, the most compelling being that of a randomized experiment. Let T_i be a treatment indicator (1 is when i is

in the treatment group and 0 otherwise). The *observed outcome* for observation i is then:

$$Y_i = T_i Y_{i1} + (1 - T_i) Y_{i0}$$

Note that in contrast to the usual regression assumptions, the potential outcomes, Y_{i0} and Y_{i1} are *fixed quantities* and not random variables, and that Y_i is only random because of treatment assignment.
In principle, if assignment to treatment is randomized, causal inference is straightforward because the two groups are drawn from the same population by construction, and treatment assignment is independent of all baseline variables. The distributions of both observed and unobserved variables between treatment and control groups are equal. Treatment assignment is independent of Y_0 and Y_1 (i.e. $Y_{i0}, Y_{i1} \perp Y_i$). In other words, the distributions of both of the *potential outcomes* (Y_0, Y_1) are the same for treated $(T = 1)$ and control $(T = 0)$. Hence for $j = 0, 1$,

$$E(Y_{ij}|T_i = 1) = E(Y_{ij}|T_i = 0)$$

where the expectation is taken over the distribution of treatment assignments. This equation states that the distributions of potential outcomes in treatment and control groups are the same in expectation. But for treatment observations, one observes Y_{i1} and for control observations, Y_{i0}. Treatment status filters which of the two potential outcomes we observe but does not change them.
The average treatment effect (ATE) is then defined to be

$$\begin{aligned} \tau &= E(Y_{i1}|T_i = 1) - E(Y_{i0}|T_i = 0) \\ &= E(Y_i|T_i = 1) - E(Y_i|T_i = 0) \end{aligned}$$

1.2 Compliance

In an experimental setup, a common problem is compliance. For example, a person assigned to treatment may refuse it, or a person assigned to control may find some way to receive treatment nevertheless. When there are compliance issues, the above equation for τ describes the *intention to treat* (ITT) estimated. The intention to treat analysis is when the experimental comparison is between the whole treatment group (for example all those invited to be screened for breast cancer, whether or not they accepted the screening) and the whole control group (which will consist of people who would both have accepted and not accepted, so restricting the comparison to just those women in the treatment group who accepted screening with the whole control group, is biased agains screening).

1.3 Observational data

In an observational setting, treatment and non treatment groups are almost never balanced because the two groups are not ordinarily drawn from the same population. Thus a common quantity of interest is the *average treatment effect*

for the treated (ATT):

$$\tau|(T=1) = E(Y_{i1}|T=1) - E(Y_{i0}|T_i=1)$$

where the expectation is taken over the distribution of treatment assignments. This cannot be directly estimated since Y_{i0} is not observed for the treated.

2 Taylor expansion

Chances are, I'll need to use a Taylor expansion at some point. The Taylor expansion of $f(x)$ about a is given by

$$f(x) \approx f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + ...$$

3 Bias/Variance trade-off

The bias/variance tradeoff comes from the fact that

$$MSE = Bias^2 + Variance$$

which can be shown as follows

$$\begin{aligned}
MSE(\hat{\theta}) &= E\left[\hat{\theta} - \theta\right]^2 \\
&= E\left[\hat{\theta} - E(\hat{\theta}) + E(\hat{\theta}) - \theta\right]^2 && \text{(adding and subtracting } E(\hat{\theta})) \\
&= E\left[\hat{\theta} - E(\hat{\theta})\right]^2 + 2E\left[(\hat{\theta} - E(\hat{\theta}))(E(\hat{\theta}) - \theta)\right] + E\left[E(\hat{\theta}) - \theta\right]^2 \\
&= Var(\hat{\theta}) + 2(E(\hat{\theta}) - \theta))E\left[\hat{\theta} - E(\hat{\theta})\right] + \left(E(\hat{\theta} - \theta)\right)^2 && (E(\hat{\theta}) \text{ and } \theta \text{ are constant}) \\
&= Var(\hat{\theta}) + Bias(\hat{\theta})^2
\end{aligned}$$

where the last equality follows from: $E\left[\hat{\theta} - E(\hat{\theta})\right] = E(\hat{\theta}) - E(\hat{\theta}) = 0$.

4 Kernel density estimation

Given a 1-dimensional kernel function (e.g. the standard Gaussian density function), $K(\cdot)$, such that

$$\int K(t)dt = 1 \quad \text{and} \quad \int tK(t)dt = 0$$

we have define the rescaled kernel density function with bandwidth h as

$$K_h(t) = \frac{1}{h}K\left(\frac{t}{h}\right)$$

Then a kernel density function of a dataset $x_1, x_2, ..., x_n$ is given by

$$g_{n,h}(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x_i - x)$$

A Kernel density plot can be simply descriptive. For example, varying h gives different views of the data: larger h's give the bigger picture of the data, while smaller h's give the details.
If the random variables $X_1, ..., X_n$ share the same distribution with density function f, then we can consider the following kernel density estimator for f.

$$\hat{f}_{n,h}(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(X_i - x)$$

4.1 Bias of kernel density estimator

Note that

$$\begin{aligned} E\left(\hat{f}(x)\right) &= \frac{1}{n}\sum_{i=1}^{n} E\left[\frac{1}{h}K\left(\frac{X_i - x}{h}\right)\right] \\ &= \int_{-\infty}^{\infty} \frac{1}{h}K\left(\frac{z-x}{h}\right) f(z)dz \end{aligned}$$

using the change of variables

$$u = \frac{z-x}{h}$$

we get

$$E\left(\hat{f}(x)\right) = \int_{-\infty}^{\infty} K(u)f(x+hu)du$$

This integral is not analytically solvable, so we approximate it using a Taylor expansion of $f(x+hu)$ in the argument hu, which is valid as $h \to 0$. For a vth order kernel, we take the expansion out to the vth term

$$f(x+hu) = f(x) + f^{(1)}(x)hu + \frac{1}{2}f^{(2)}(x)h^2u^2 + ... + \frac{1}{v!}f^{(v)}(x)h^v u^v + o(h^v)$$

Integrating term by term, and using $\int_{-\infty}^{\infty} K(u)du = 1$, we have that

$$bias = E(\hat{f}_h(x)) - f(x) = \frac{f''(x)}{2}\int_{-\infty}^{\infty} K(u)u^2h^2du + o(h^2)$$

note that the term with the first derivative dropped out since

$$\int_{-\infty}^{\infty} uK(u)du = 0$$

Note that the bias will go to zero if the bandwidth h shrinks to zero.

4.2 Variance of kernel density estimator

Using Taylor expansion once again, we get

$$\begin{aligned}
Var\left(\hat{f}(x)\right) &= \frac{1}{n} Var\left(\frac{1}{h} K\left(\frac{x-X}{h}\right)\right) \\
&= \frac{1}{n}\left[E\left[\frac{1}{h^2} K^2\left(\frac{x-X}{h}\right)\right] - \left(E\left[\frac{1}{h} K\left(\frac{x-X}{h}\right)\right]\right)^2\right] \\
&= \frac{1}{n}\left[\int_{-\infty}^{\infty} \frac{1}{h^2} K^2\left(\frac{x-z}{h}\right) f(z) dz - \left(f(x) + o(h^2)\right)^2\right] \\
&= \frac{1}{n}\left[\int_{-\infty}^{\infty} \frac{1}{h} K^2(u) f(x-hu) du - f^2(x) + o(h^2)\right] \\
&= \frac{1}{n}\left[\int_{-\infty}^{\infty} \frac{1}{h} K^2(u)\left(f(x) + huf'(1)(x) + \frac{h^2u^2}{2} f''(x)\right) du - f^2(x) + o(h^2)\right] \\
&\geq \frac{f(x)}{nh} \int_{-\infty}^{\infty} K^2(u) du + \text{smaller order terms}
\end{aligned}$$

4.3 MSE of kernel density estimator

Given the results above, we can formulate the point-wise bias-variance trade-off as follows

$$\begin{aligned}
MSE(x) &= E(\hat{f}(x) - f(x))^2 \\
&= bias^2 + Var \\
&\approx \left[\frac{(f''(x))^2}{4}\left(\int_{-\infty}^{\infty} K(u) u^2 du\right)^2\right] h^4 + \frac{f(x)}{nh} \int_{-\infty}^{\infty} K^2(u) du
\end{aligned}$$

By setting the derivative with respect to h, the above point-wise risk (MSE) is optimized by a bandwidth (depending on x naturally) that satisfies

$$\frac{dMSE(x)}{dh} = \left[(f''(x))^2\left(\int_{-\infty}^{\infty} K(u) u^2 du\right)^2\right] h^3 - \frac{1}{nh^2} \int_{-\infty}^{\infty} K^2(u) du = 0$$

implying that

$$h^* = \left[\frac{1}{n} \frac{\int_{-\infty}^{\infty} K^2(u) du}{\int_{-\infty}^{\infty} K(u) u^2 du} \frac{f(x)}{(f''(x))^2}\right]^{\frac{1}{5}}$$

5 The delta method

The delta method gives

$$Var(g(X)) \approx \left[g'(\mu_X)\right]^2 Var(X), \qquad \text{where } \mu_X = E(X)$$

6 Kernel smoothing

6.1 Nadaraya-Warson kernel smoother

It can be informative to add a smoothing line to a plot. There are various types of kernel smoothers we could use, such as the **Nadaraya-Watson kernel smoother** which, given data pairs (x_i, y_I) and bandwidth h is defined by

$$g_h(x) = \frac{\sum_{i=1}^n K_h(x_i - x) y_i}{\sum_{i=1}^n K_h(x_i - x)}$$

For any fixed x, this is the (local) constant minimizer of a weighted LS over θ:

$$\sum_{i=1}^n (y_i - \theta)^2 w_i(x), \qquad \text{where } w_i(x) = K_h(x_i - x)$$

6.2 Local polynomial smoother

We could do a local polynomial fit, instead of a constant fit, by minimizing

$$\sum_{i=1}^n \left(y_i - \sum_{j=1}^p \beta_j (x_i - x)^j \right)^2 w_i(x)$$

$p = 0$ gives the Nadaraya-Watson kernel smoother and $p = 1$ gives a local linear smoother.

6.3 LOESS

The Locally Weighted Scatter plot Smoothing (Lowess or Loess) robustifies the locally weighted least squares above ad uses the tricube kernel weight

$$w(x) = \frac{70}{81}(1 - |t|^2)^2 \mathbb{1}_{[-1,1]}(x)$$

There is some extra stuff in lecture 6 on these methods!!

7 Principal component analysis

Let $G = X^T X$ be the sample covariance matrix, which is symmetric. The eigenvalue decomposition of G is given by

$$G = UDU^T$$

where U is a unitary/orthogonal matrix ($U^T U = I$) and D is a diagonal matrix with non-negative diagonal entries that are the eigenvalues of G. Let the jth column vector of U be U_j and of D be d_j, then

$$GU_j = d_j U_j$$

The geometric interpretation of U is a rotation and $\sqrt{D}$ is a rescaling. Then after the rotation applied to X, we get

$$(Z_1, ..., Z_p) = XU = (X_1, ..., X_p)(U_1, ..., U_p)$$

It follows that

$$Z_j = (X_1, ..., X_p)U_j = u_{1j}X_1 + ... + u_{pj}X_p$$

where $(_{1j}, ..., u_{jp})^T = U_j$ and $\sum_k u_{kj}^2 = 1$ because U is orthonormal. $Z_1, ..., Z_p$ are called **principal components**. Further, because

$$D = U^T GU = (XU)^T(XU) = Z^T Z$$

we have

$$Var(Z_j) = d_j \qquad \text{and} \qquad Cov(Z_j, Z_i) = 0 \text{ if } i \neq j$$

so the proportion of variance explained by the jth principal component is the same as the jth eigenvalue divided by the sum of the eigenvalues. Moreover, the principal components are independent.
An important property of the eigenvalues is that they add up to the trace of the covariance matrix G or the variance of the sum of the original predictors:

$$\sum_j d_j = trace(G) = \sum_j Var(X_j)$$

thus in a sense, PCA keeps all the variation in the original data.

An equivalent way to define principal components is to first find the linear combination (loading) with norm 1 of the columns of X such that its variance is maximized. Such a loading forms the first principle component vector. Similarly we can define the second principal component vector to be the vector orthogonal to the first PC vector, has norm 1 and maximizes the variance of the resulting linear combination of the columns of X, and so on.

7.1 The variance maximizing property of eigenvectors

Claim: the first eigenvector U_1 maximizes

$$||Xu||^2 \qquad \text{subject to} \qquad ||u||^2 = 1$$

Proof: Note that $G = X^T X = UDU^T$, where U is orthonormal. Let $v = U^T u$. Then we want to maximize

$$||Xu||^2 = u^T X^T Xu = u^T UDU^T u = v^T Dv \qquad \text{subject to} \qquad ||v||^2 = ||U^T u||^2 = 1$$

because U is a rotation.

Note that $v^T Dv = \sum_j v_j^2 d_j$, which is maximized if we put all the mass on the largest eigenvalue d_1 or if we take $v = (1, 0, 0, ..., 0)^T$ which corresponds to $u = U^T v = U_1$, the first eigenvector.
Remark that if X is centered, then $||Xu||^2$ is also a variance.

7.2 Centering and normalization before eigen decomposition

In order to avoid one predictor having an undue influence on the principal components, it is common to first standardize the predictors to have mean 0 and variance 1 before PCA. Hence PCA has the following steps:

- Center the predictors to mean 0, standard deviation 1
- Form the sample covariance $G = X^T X$
- Carry out an eigenvalue decomposition of G to get eigenvalues $d_1 \geq ... \geq d_p \geq 0$ and the corresponding eigenvector U_j
- Keep all the large principal components to account for most of the variation in the data. For example, starting with 20 predictors, it might be the case that the first 4 components account for 90% of the total variation, i.e. the sum of the first 4 eigenvalues is about 90% of the total sum of all eigenvalues.

8 Clustering

The main methods of clustering include K-means, Hierarchical clustering and spectral clustering (PCA and K-means)

8.1 K-means

Suppose that there are n points (objects), $x_1, ..., x_n$ in $\mathbb{R}^p$ to be clustered. Then, given the number of clusters, k_c, K-means aims to find a partition of the index set $\{1, ..., n\}$ into $I_1, ..., I_{k_c}$ and its associated cluster centers, $C_1, ..., C_{k_c}$ which minimizes the following goal objective function for a distance metric d

$$\sum_{i=1}^{k_c} \sum_{x_j : j \in I_i} d(x_j, C_i)$$

for example, if $d = L^2$, this corresponds to finding a partition of the index set into $I_1, ..., I_{k_c}$ with associated centers $C_1, ..., C_{k_c}$ which minimizes

$$\sum_{i=1}^{k_c} \sum_{x_j : j \in I_i} (x_j - C_i)^2$$

i.e. we want to find the partition and centers such that each point in any given partition is as close as possible to the center of that partition.

8.1.1 Initial step

First, select k_c initial cluster/group centers (e.g. randomly select k_c points without replacement from the n points). Denote these centers as

$$C_1^0, ..., C_{k_c}^0$$

Then a partition P^0 can be created of the n points into

$$I_1^0, ..., I_{k_c}^0$$

which are non-overlapping sets of indices of the n points. The assignment rule is that a point i belongs to I_k^0 if and only if i is closer to C_k^0 than all other centers in d distance. (The clusters are delimited in space by convex polyhedral divisions formed by the median planes of the segments joining all the center pairs).

8.1.2 The algorithm

k_c new centers are determined

$$C_1^1, ..., C_{k_c}^1$$

by using the centers of gravity of the clusters

$$I_1^0, ..., I_{k_c}^0$$

That is, for each cluster, take the new center to be the point such that the sum of its distances to all of the other points in the cluster is minimized. If d is the Euclidean L^2 norm, the center of gravity is the mean of the points in a cluster. If d is the L^1 norm, the center of gravity is the median point.

The process is terminated if two succeeding iterations lead to the same partition or when a stopping criterion is satisfied (the within-groups variance decrease is less than a threshold; or a pre-set maximum number of iterations is achieved).

8.2 PAM

The PAM function in R's cluster library is a modified K-means algorithm. It attempts to minimize the global sum of squares as in K-means, but subject the the constraint that *the centers $C_1, ..., C_{k_c}$ have to be at the data points.*

One way to minimize this objective function is to modify the K-means or the LLoyd-Max algorithm by finding the means of the groups first and then replacing them by the closes data points to these means. This gives the PAM algorithm or the K-medoids algorithm.

8.3 Hierarchical clustering

For a hierarchical clustering algorithm, we need:

- a distance or dissimilarity measure between any two points
- a rule to calculate the distance/dissimilarities between disjoint clusters of objects. This between cluster distance can generally be calculated directly from the distances of the various elements involved in the clustering.

For example, if H is a set of points and the point z is not in H, then

$$d(H, z) = \min_{x \in H} d(x, z)$$

This is the distance used in single-linkage hierarchical clustering.

Hierarchical algorithms are based on common sense rather than formalized theory. WLOG, we call all points either the objects to be clustered or the clusters of objects generated by the algorithm.

8.3.1 The algorithm

- Initially (step 1), there are n points to cluster.
- Find two points that are closest together and aggregate them into a new "point".
- Return to step 1 with $n - 1$ points to cluster
- Again, find the two closest "points" and aggregate them
- Calculate the new distances and repeat the process until there is only one unclustered point remaining.

Because of the equivalence of minimum spanning tree and single-linkage hierarchical clustering, the computational complexity of both is $O(n^2 \log n)$.

The family of clusters built by such an ascending algorithm forms a so-called hierarchy. This family has the property of having the whole set and also every one of the objects separately. By cutting the tree of a hierarchical clustering tree, we get a partition in which the number of clusters increases as the cit-point approaches the initial points. A hierarchy allows us to obtain a set of n nested partition containing I to n groups/clusters.

8.4 Silhouette

We can use Silhouette to decide on the number of clusters. Having clustered the data points into k groups, given any data point i, let a_i be the average distance or dissimilarity of i with all other points in the same cluster, and let b_i be the smallest average distance of i to other clusters. In particular

- a_i measures how well i fits into its cluster
- b_i is the smallest average distance of i to other clusters

Define

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

which is a number between -1 and 1.

We have that s_i is close to 1 if point i is in a tight cluster far away from other clusters, and s_i is close to -1 if it is in a loose cluster and close to other clusters. We can select the number of clusters by determining the value of k which maximizes the average $\frac{1}{n}\sum_{i=1}^{n} s_i$.

8.5 Minimum spanning tree (MST)

Given n nodes (n points) with pairwise edge values (distances or dissimilarities), a complete graph can be constructed by connecting all pairs and assigning weights equal to these distance edge values. However, this complete graph might not be informative, and subgraph might be more desirable.

Definition (**Minimum spanning tree)**. A minimum spanning tree is a tree (a graph without cycles) that connects all the nodes such that the total edge value is minimized.

For simplicity, we assume that all edge values are distinct, so that there is a unique minimum spanning tree. There exist a number of different algorithms to construct the minimum spanning tree such as

8.5.1 Kruskal's algorithm

Arrange the $\frac{n(n-1)}{2}$ edges in order of increasing edge weight. Stating with the first two edges (the two edges with the smallest weights), take the edges one by one (in order of increasing weight) and keep the edges that do not form a cycle with the edges already chosen. Stop when there are $n-1$ edges.

Here we offer a heuristic proof. The sum of the edges is obviously minimum. Moreover, since there are no cycles within a tree, a tree with $n-1$ edges has to have n nodes. Thus the tree connects all of the nodes.

The concept of a hierarchy is closely linked to the concept of ultra-metric distance. An ultra metric distance is a distance with the stronger triangle inequality property that

$$d(x, y) \leq \max(d(x, z), d(y, z))$$

For a single-linkage hierarchy, the corresponding ultra metric distance is in some sense closest to the original distance. It is call the maximum lower ultra metric distance (or the subdominant ultra metric distance). Under this metric, the single-linkage method is equivalent to finding the minimum spanning tree on a graph.

8.6 Cross validation

Cross-validation can be used to decide the number of K-means clusters as follows

- Divide the data into V batches
- For a given K, the number of clusters, remove the batches one by one and use the other $V-1$ batches to cluster by K-means.
- Calculate the K-means objective function value (for example, the sum of the squared distances to the centroids for k-means) on the removed batch
- Add up the V objective function values. Minimize this sum over different k's.

8.7 Spectral clustering

Spectral clustering involves performing K-means following PCA. If $A \in \mathbb{R}^{n \times n}$ is the (symmetric) adjacency matrix, where

$$A_{ij} = A_{ji} = \begin{cases} 1 & \text{if } (i,j) \text{ is an edge} \\ 0 & \text{otherwise} \end{cases}$$

and $D \in \mathbb{R}^{n \times n}$ is the (diagonal) degree matrix, where

$$D_{ii} = \sum_j A_{ij}$$

Then spectral clustering deals with the eigenvectors of the normalized symmetric Laplacian matrix

$$L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

Other matrices used include

$$\begin{aligned} D^{-1}A &\quad \text{(Normalized random walk Laplacian)} \\ D - A &\quad \text{(Unnormalized Laplacian)} \\ A &\quad \text{(Adjacency matrix)} \end{aligned}$$

8.7.1 The algorithm

To find K clusters:

- Compute the $n \times K$ matrix V consisting of the top K eigenvectors of L
- Cluster the rows of V into K clusters (for example, using K-means)

Each row of V represents a node in the graph.

Note that spectral clustering has a computational advantage since it requires eigenvector decomposition which is very fast. However, an empirical observation

is that the performance of spectral clustering improves greatly through regularization.

The regularized algorithm involves

- Add a constant matrix to the adjacency matrix A

$$A_\tau = A + \frac{\tau}{n}\mathbf{1}\mathbf{1}^T, \qquad \tau > 0$$

- Construct the laplacian, L_τ from A_τ
- Compute the $n \times K$ matrix V_τ consisting of the top K eigenvectors of L_τ
- Cluster the rows of V_τ into K clusters

8.7.2 Stochastic block model (SBM)

The adjacency matrix could be generated using a stochastic block model. Given a set of n nodes, the edges (i, j) are drawn independently with probability P_{ij}, where

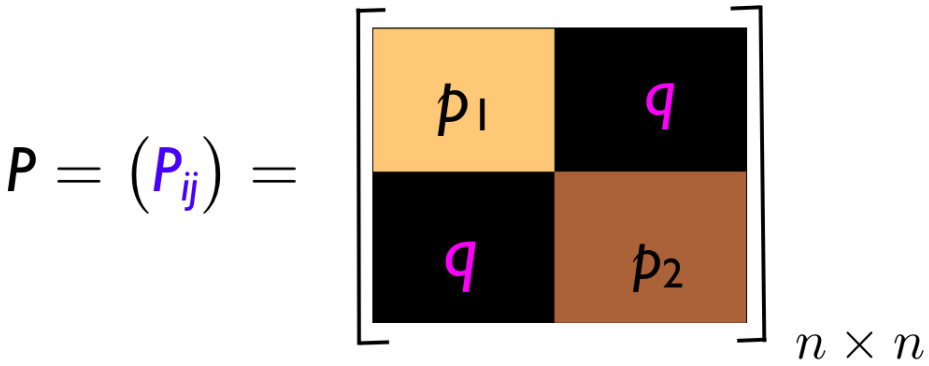

Figure 1: Stochastic block model with two blocks

So we end up with a "block" adjacency matrix

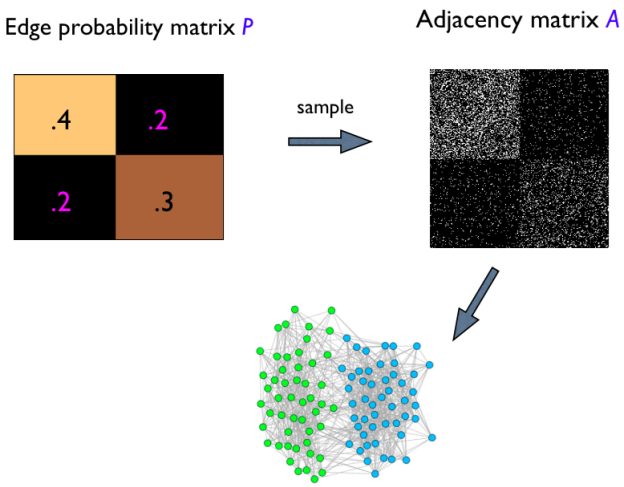

Figure 2: Adjacency matrix and clustering from a stochastic block model with two blocks

There is more in the spectral clustering pdf, but I'll come back to this

9 EM algorithm

Given a statistical model consisting of a set X of observed data, a set of unobserved latent data or missing values Z, and a vector of unknown parameters θ along with a likelihood function $L(\theta; X, Z) = p(X, Z|\theta)$, the maximum likelihood estimate of the unknown parameters is determined by the marginal likelihood of the observed data

$$L(\theta; X) = p(X|\theta) = \sum_z p(X, Z|\theta)$$

However, this quantity is often intractable. The EM algorithm seems to find the MLE of the marginal likelihood by iteratively applying the following two steps.

- **Expectation step (E step)**: Calculate the expected value of the log likelihood function, with respect to the conditional distribution of Z given X under the current estimate of the parameters $\theta^{(t)}$

$$Q(\theta)|\theta^{(t)}) = E_{Z|X,\theta^{(t)}}[\log L(\theta; X, Z)]$$

- **Maximization step (M step)**: Find the parameter that maximizes this quantity:

$$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}} Q(\theta|\theta^{(t)})$$

The details for the derivation are in Borman's tutorial.

9.1 Example

The Expectation-Maximisation (EM) algorithm can be used to perform clustering. Fitting a mixture of gaussians to data via maximum likelihood results in "clusters" as descriptive characteristics of data.

Suppose $X_1, ..., X_n$ are iid from a 1-dimensional two-compoennt Gaussian mixture

$$f_\theta(x) = \pi g(x|\mu, \sigma^2) + (1 - \pi)g(x|\nu, \tau^2)$$

where g is a normal density with the parameters specified, and

$$\theta = (\pi, \mu, \sigma^2, \nu, \tau^2)$$

we might ask, what is a possible random mechanism that generated the data from the above model?.

Definition (**Model identifiability**). A parametric statistical model is identifiable on its parameter domain if and only if no two distinctive parameter values correspond to the same distribution.

Data contains information about only the distribution, but without identifiability, parameter estimation is ill-posed. So the question remains, when is our model identifiable? Or for what parameter domain is the model identifiable?
In our example, the populated model can be obtained via a hidden state variable or a latent variable in two steps:

First, we generate $Z \sim Bernoulli(\pi)$, the indicator to which component the data point X belongs. Given Z, generate the data point according to the corresponding Gaussian distribution. X is called the incomplete data.

EM will estimate all five parameters by iterating between

1. "Imputing" Z given the current parameter values (E-step)
2. Using both X and the imputed Z to estimate the parameters (M-step)

9.2 Performing EM estimation for the two-component Gaussian mixture

To set up the problem, let $x = (x_1, ..., x_n)$ be a sample of n independent observations from a mixture of two multivariate normal distributions of dimension d, and let $z = (z_1, ..., z_n)$ be the latent variables that determine the component from which the observation originates.

$$X_i|(Z_i = 1) \sim N_d(\mu_1, \Sigma_1) \quad \text{and} \quad X_i|(Z_i = 2) \sim N_d(\mu_2, \Sigma_2)$$

where

$$P(Z_i = 1) = \tau_1 \quad \text{and} \quad P(Z_i = 2) = \tau_2 = 1 - \tau_1$$

The aim is to estimate the unknown parameters representing the mixing value between the gaussians and the means and covariances of each:

$$\theta = (\tau, \mu_1, \mu_2, \Sigma_1, \Sigma_2)$$

where the likelihood function is

$$L(\theta; x, z) = P(x, z|\theta) = \prod_{i=1}^{n} P(Z_i = z_i) f(x_i|\mu_{z_i}, \Sigma_{z_i})$$

or

$$L(\theta; x, z) = P(x, z|\theta) = \prod_{i=1}^{n} \sum_{j=1}^{2} \mathbb{1}_{(z_i=j)} \tau_j f(x_i; \mu_j, \Sigma_j)$$

This may be written in exponential family form:

$$L(\theta; x, z) = \exp\left\{ \sum_{i=1}^{n} \sum_{j=1}^{2} \mathbb{1}_{z_i=j} \left[\log \tau_j - \frac{1}{2} \log |\Sigma_j| - \frac{1}{2}(x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j) - \frac{d}{2} \log(2\pi) \right] \right\}$$

9.2.1 E-step

given our current estimate of the parameters $\theta^{(t)}$, the conditional distribution of the Z_i is determined by Bayes' theorem to be

$$T_{j,i}^{(t)} = P(Z_i = j | X_i - x_i; \theta^{(t)}) = \frac{\tau_j^{(t)} f(x_i; \mu_j^{(t)}, \Sigma_j^{(t)})}{\tau_1^{(t)} f(x_i; \mu_1^{(t)}, \Sigma_1^{(t)}) + \tau_2^{(t)} f(x_i; \mu_2^{(t)}, \Sigma_2^{(t)})}$$

These are called the membership probabilities and are normally considered the output of the E step (although this is not the Q function of below)

$$\begin{aligned}
Q(\theta|\theta^{(t)}) &= E[\log L(\theta; x, Z)] \\
&= E\left[\log \prod_{i=1}^{n} L(\theta; x_i, z_i) \right] \\
&= E\left[\sum_{i=1}^{n} \log L(\theta; x_i, z_i) \right] \\
&= \sum_{i=1}^{n} E[\log L(\theta; x_i, z_i)] \\
&= \sum_{i=1}^{n} \sum_{j=1}^{2} T_{j,i}^{(t)} \left[\log \tau_j - \frac{1}{2} \log |\Sigma_j| - \frac{1}{2}(x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j) - \frac{d}{2} \log(2\pi) \right]
\end{aligned}$$

9.2.2 M-step

Here, we simply maximize $Q(\theta|\theta^{(t)})$ independently for each parameter. Due to laziness. here we'll just do $\tau^{(t+1)}$.

$$\begin{aligned}\tau^{(t+1)} &= \underset{\tau}{\text{argmax}} Q(\theta||\theta^{(t)}) \\ &= \underset{\tau}{\text{argmax}} \left\{ \left[\sum_{i=1}^{n} T_{1,i}^{(t)}\right] \log \tau_1 + \left[\sum_{i=1}^{n} T_{2,i}^{(t)}\right] \log \tau_2 \right\}\end{aligned}$$

which has the same form as the MLE for the binomial distribution, so

$$\tau_j^{(t+1)} = \frac{\sum_{i=1}^{n} T_{j,i}^{(t)}}{\sum_{i=1}^{n} T_{1,i}^{(t)} + T_{2,i}^{(t)}}$$

and similarly for the other variables.

9.3 Some theoretical results

Under regularity conditions, the EM converges to a stationary point of the likelihood equation. Furthermore, when the number of components is not known, we can estimate the mixing probabilityies at only $O\left(n^{-\frac{1}{4}}\right)$, which is much slower than the usual $O\left(n^{-\frac{1}{2}}\right)$

9.4 Assessment of clustering results

How do we know whether what we finds meaningful, especially when there are so may parameters to tune? A few ideas to consider:

- **Stability** of clusters across different bootstrapped samples or subsamples (if IID) or perturbed samples in a sensible way
- **Stability** across reasonable distances, methods and tuning parameters
- **Similarities** of the elements in one cluster and dissimilarities across clusters
- Subject matter validates the clusters found
- Judge clustering results in the iterative learning framework. Ask question regarding the impact of clustering analysis in downstream steps and in light of new data (does prediction make sense based on the clustering?)

10 Prediction

Much of science, social science, and engineering is about finding relationships between predictor variables and response variables. If the response variables are observable, it is called *regression* (when the response is continuous) or *classification* (when the response is discrete. And they are both called *supervised learning*.
For finite data it is possible that a wrong statistical model with estimated parameters gives a better prediction than a correct model with estimated parameters.

10.1 A simple example where the wrong model gives a better prediction

Assume $X_1, ..., X_n$ are IID $N(\theta, \sigma^2)$. The goal is to predict X_{n+1}, which follows the same distribution as the previous X_i's and is independent of them. Suppose we have two models

$$M_0 : \theta = 0 \quad \text{and} \quad M_1 : \theta \neq 0$$

we get two corresponding predictors: 0 and $\bar{X}_n$. It is easy to calculate that prediction error for 0 is $\theta^2 + \sigma^2$ which is smaller than the prediction error $\frac{\sigma^2}{n} + \sigma^2$ for $\bar{X}_n$ provided that $|\theta| < \frac{\sigma}{\sqrt{n}}$ Why is this true??.
The biased estimator based on M_0 outperforms the MLE ($\bar{X}_n$), which is often the case in high-dimensional estimation via regularzation.

10.2 Prediction and evaluation

Much of recent machine learning research has been on prediction in the batch doe. That is, based on a set of training data, a prediction scheme is developed and then this scheme is repeatedly applied to future data.

The batch-mode statistical prediction problem: Given training data that are units of predictor variables and response variable, or $(x_i, y_i), i = 1, ..., n$ where $x_i \in \mathbb{R}^p$, a vector of p real values, and y_i is a scalar, one finds a prediction rule $f(x_i)$, mapping the predictor vector to a response. So that for a new predictor x, we can predict the response variable by $f(x)$.

The fundamental assumptions of predictions are that the available situations are similar, with respct to the aspects of the situation to be prediction on, to new situations that prediction is needed. For example, the data to be predicted for is similar to the training data. This implies that the prediction rules learned from the training data are applicable to new predictor variables.

10.3 Evaluation of a prediction rule

To evaluate how good the prediction rule it, we could compare the predicted values with the observed values. Sometimes one prediction point decides, and other times we want to see an average performance. Examples of things to consider are

- A measure of prediction error: a loss function between a prediction of a response and a realization of a response. L^2 is a convenient choice, but not always the appropriate one.
- An estimated prediction error (in an average sense)
- Uncertainty in the estimated prediction error
- Computational cost: memory space, access time and CPU time
- Interpretability.

We can use cross validation to estimate the prediction error. When the training data is limited, one often re-uses, when appropriate, the same data units for training and testing via CV. Note however, that when the number of data units is small relative to the complexity of the prediction rule, CV can be far away from the real prediction error and can't be trusted. CV error is NOT prediction error!.

There are several versions of cross-validation:

1. V-fold CV: divide the training data into V equally sized groups. Cycle through the V groups by using the V-1 groups for training and the left out group for testing. Add up the errors from the V operations and average
2. Leave one out: this is a special case, where each group consists only of one data unit. This is similar to jacknife for estimating bias and later variance
3. Random split CV: randomly select $(1-\alpha)\times 100\%$ data units for training and use the raining data units for testing. Repeat this process M times and average the resulted prediction errors. Common: $\alpha = 0.1$, $M = 100$.

Note that **CV ESTIMATES the prediction error** and as an estimator, it could have large variance. When n is small relative to the complexity of the rules fitted and because the summands in CV are often positively correlated because they share a lot of data in common. *Positively correlated summands lead to higher variance than when they are independent*
The fundamental assumption for CV is that the data units are exchangeable or there is symmetry.

10.4 One-way CV

One-way CV is an online mode of prediction where the accumulated prediction error is the performance metric. As we accumulate new data, the model/rule is updated and a prediction is cast and compared with the next response. The prediction errors are accumulated/added up. This is one-way CV because the future observation is not involved in its own prediction.

Suppose the data units form some natural order (e.g. a time series). Begin an initial predictor, which could take into account subject knowledge or information from previous data. At the t use all of the data up to this time to form a predictor, then use the observation at time $t+1$ to evaluate the prediction error. Add all the prediction errors and average.
This accumulated prediction error can be used to compare different prediction rules and select among competing models.

11 Matrix algebra

11.1 Matrix inverse

The inverse of a 2×2 matrix is

$$A^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{det(A)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

11.2 Positive definite matrices

An $n \times n$ orthogonal (unitary) matrix R has $RR^T = I$ and $R^T R = I$. Geometrically, R is a rotation which preserves angles and distances. G is **non-negative definite** if

1. G is symmetric, and
2. $x^T GX \geq 0$ for any n vector x

Theorem 11.1. *The matrix G is non-negative definite if and only if there is a diagonal matrix D whose elements are non-negative, and an orthogonal matrix T such that $G = RDR^T$. The matrix G is positive definite if and only if the diagonal entries of D are all positive.*

The columns of R are the eigenvectors of G, and the diagonal elements of D are the eigenvalues.

It follows from this theorem that a non-negative definite G has a non-negative definite square-root $G^{\frac{1}{2}} = RD^{\frac{1}{2}}R^T$. A positive definite G has a positive definite inverse $G^{-1} = RD^{-1}R^T$

11.3 Matrix derivative

If a and b are $K \times 1$ vectors, then

$$\frac{\partial a^T b}{\partial b} = \frac{\partial b^T a}{\partial b} = a$$

If A is any symmetric matrix, then

$$\frac{\partial b^T A b}{\partial b} = 2Ab = 2b^T A$$

If $X^T X$ is a $K \times K$ matrix, and β is a $K \times 1$ vector, then

$$\frac{\partial 2\beta^T X^T y}{\partial \beta} = \frac{\partial 2\beta^T (X^T y)}{\partial \beta} = 2X^T y$$

and

$$\frac{\partial \beta^T X^T X \beta}{\partial \beta} = \frac{\partial \beta^T A \beta}{\partial \beta} = 2X^T X \beta$$

11.4 Singular value decomposition

The singular value decomposition of an $m \times n$ matrix M is a factorization of the form

$$M = U \Sigma V^T$$

where

- U is an $m \times m$ unitary matrix
- Σ is an $m \times n$ rectangular diagonal matrix
- V is an $n \times n$ unitary matrix

12 Least squares

Least squares (LS) is a curve fitting method (a polynomial fit when $p = 1$). Given data $(x_i, y_i), i = 1, ..., n$, we fit a line to the at a cloud through LS with x as the predictor and y as the response. The goal is to find parameters (a, b) such that

$$\sum_{i=1}^{n} (y_i - a - bx_i)^2$$

is minimized. The LS line is not the line of symmetry line unless all data points fall exactly on a line.

12.1 Regression fallacy

If the x_i's are mid-term scores and the y_i's are the final exam scores, they almost certainly won't fall on a straight line. For example, a low performance group (those with scores below average) will improve on average at the final, and the high performance group (those with scores above the avert) will get worse on average for the final.
Regression fallacy in this situation says that the improvement of the low performance group is not due to variability of data, but something else, such as student efforts.

12.2 Multiple regression

Suppose we have the multiple regression model

$$Y_{n\times 1} = X_{n\times p}\beta_{p\times 1} + \epsilon$$

The assumptions are

- The data on Y are the observed values of $X\beta + \epsilon$
- The ϵ_i are independent and identically distributed with mean 0 and variance σ^2
- If X is random, then ϵ is independent of X

In particular, things that we don't need to assume are that

- The columns of X don't have to be orthogonal to each other
- The random errors don't need to be normally distributed

Theorem 12.1. *The OLS estimator of β is given by*

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

Proof. The OLS estimate is given by

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} ||Y - X\beta||_2^2$$

$$\begin{aligned} ||Y - X\beta||_2^2 &= (Y - X\beta)^T (Y - X\beta) \\ &= Y^T Y - \beta^T X^T Y - Y^T X\beta + \beta^T X^T X\beta \\ &= Y^T Y - 2\beta^T X^T Y + \beta^T X^T X\beta \end{aligned}$$

so

$$\frac{\partial}{\partial \beta} ||Y - X\beta||_2^2 = -2X^T Y + 2X^T X\beta$$

setting this to zero, if $X^T X$ is full rank, implies

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

□

The residuals are defined as

$$e = Y - X\hat{\beta}$$

Theorem 12.2. *The following are true:*

- $e \perp X$
- *As a function of the* $p \times 1$ *vector* γ, $||Y - X\gamma||_2^2$ *is minimized when* $\gamma = \hat{\beta}$

Theorem 12.3. *OLS is conditionally unbiased, that is,* $E(\hat{\beta}|X) = \beta$

Proof.

$$\begin{aligned}
\hat{\beta} &= (X^T X) X^T Y \\
&= (X^T X)^{-1} X^T (X\beta + \epsilon) \\
&= (X^T X)^{-1} X^T X\beta + (X^T X)^{-1} X^T \epsilon \\
&= \beta + (X^T X)^{-1} X^T \epsilon
\end{aligned}$$

so

$$E(\hat{\beta}|X) = \beta$$

□

Theorem 12.4. $cov(\hat{\beta}|X) = \sigma^2 (X^T X)^{-1}$

Proof.

$$\begin{aligned}
cov(\hat{\beta}|X) &= cov((X^T X)^{-1} X^T Y|X) \\
&= (X^T X)^{-1} X^T cov(\epsilon|X) X (X^T X)^{-1} \\
&= \sigma^2 (X^T X)^{-1}
\end{aligned}$$

□

Usually, however, σ^2 is unknown and has to be estimated from the data. If we knew the ϵ_i we could estimate σ^2 (remember $var(\epsilon_i) = \sigma^2$) as

$$\frac{1}{n} \sum_{i=1}^{n} \epsilon_i^2$$

but we don't know the ϵ_i's. So the next thing we might try could be

$$\frac{1}{n} \sum_{i=1}^{n} e_i^2$$

however, this is a little too small. Generally, the e_i's are smaller than the ϵ_i's because β was chosen to make the sum of the e_i^2 as small as possible. Moreover, this estimator is unbiased:

$$\begin{aligned}
E\left[||e||^2|X\right] &= E\left[||(I-H)Y||^2|X\right] \\
&= E\left[Y^T(I-H)Y|X\right] \\
&= E\left[trace\left(Y^T(I-H)Y\right)\Big|X\right] \\
&= E\left[trace\left((I-H)YY^T\right)\Big|X\right] \\
&= trace\left(E\left[(I-H)YY^T\Big|X\right]\right) \\
&= trace\left((I-H)E\left[YY^T\Big|X\right]\right) \\
&= trace\left((I-H)E\left[\epsilon\epsilon^T\Big|X\right]\right) \\
&= trace\left((I_{n\times n}-H)\sigma^2\right) \\
&= \sigma^2(n-p)
\end{aligned}$$

Thus the usual fix is to divide by the *degrees of freedom* $n-p$ rather than n:

$$\hat{\sigma}^2 = \frac{1}{n-p}\sum_{i=1}^{n} e_i^2$$

Now $\hat{\sigma}^2$ is conditionally unbiased (theorem below), and **this is the reason we need $n > p$ not just $n \geq p$.**

Recall that $\hat{\beta} = (X^TX)^{-1}X^TY$ and $\text{cov}(\hat{\beta}|X) = \sigma^2(X^TX)^{-1}$. However, as mentioned above, we don't know σ^2. Thus we can plug in $\hat{\sigma}^2$, which is almost the mean square of the residuals.

Define the **hat matrix** as

$$H = X(X^TX)^{-1}X^T$$

and the fitted values

$$\hat{Y} = X\hat{\beta}$$

The fitted values are connected to the hat matrix by the equation

$$\hat{Y} = X(X^TX)^{-1}X^TY = HY$$

Here are some fun facts:

- $e = (I-H)Y$
- H is symmetric, so is $I-H$
- H is idempotent (so $H^2 = H$ and so is H

- X is invariant under H, that is $HX = X$
- $e = Y - HX \perp X$
- $(I - H)X = 0$
- $(I - H)H = H(I - H) = 0$

Thus H is a projection matrix that projects Y into the column space of X since

$$HY = \hat{Y} = X\hat{\beta}$$

Note that LS is very sensitive to outliers. Statistical leverage, $h_i = [H]_{ii}$, measures the "outlearns" of data points. For data point i, the ith diagonal element of H is its leverage score.

12.3 Residual vector

Recall that

$$e = Y - X\hat{\beta}$$

so that

$$Y = X\hat{\beta} + e$$

however, note that this is a matrix identity, not a linear regression model. Moreover, since $X \perp e$, we have

$$||X\hat{\beta}||_2^2 + ||e||_2^2 = ||Y||_2^2$$

12.4 Pseudo-inverse of X^TX via SVD

When X is not full rank we can still take a basis of span(X) and project, but the solution to the LS problem is not unique anymore in terms of fitting a linear combination of columns of X. Often, in modern settings when we have $p > n$, X is never full-rank, and thus not invertible. We can, however, get a pseudo-inverse of X^TX by using singular value decomposition (SVD).

Recall that a square matrix U is unitary if and only if $U^{-1} = U^T$. Note that

$$X = USV^T$$

where

- U is an $n \times n$ left-eigenvector matrix which is unitary and consists of the eigenvectors of XX^T
- S is an $n \times p$ rectangular diagonal matrix whose non-zero eigenvalues are the square root of the eigenvalues of X^TX or XX^T
- V is a $p \times p$ right-eigenvector matrix, which is unitary and consists of the eigenvectors of X^TX.

Now the pseudo-inverse of X is given by

$$X^- = V\ I/S\ U^T$$

where I/S is diag(reciprocal of non-zero elements of S and setting the rest to zero).

We can use X^- to find a particular solution to the LS problem, and the resultant fitted value is the projection of Y onto the space spanned by the columns of X and it is unique.

12.5 Sub-sampling rows of X using leverage scores to reduce compassion time of LS when $n >> p$

Sample rows with probability $p_i = h_{ii} = ||U_i||^{\frac{2}{p}}$, where U_i is the ith row of the $n \times p$ matrix that consists of the first p eigenvectors (we claim that $||U_i||^2$ is the leverage score for i).

To see why the leverage score measures the influence of the data point i, we can compare the LS estimators with and without the ith observation.

$$\hat{\beta} - \hat{\beta}_{(i)} = (X^T X)^{-1} x_i^T \frac{e_i}{1 - h_{ii}}$$

Taylor expansion based MSE approximation reveals the need to control for bias and variance.

12.6 Regression model with fixed predictors

For $i = 1, ..., n$, and a fixed design matrix X

$$Y_i = x_i^T \beta + \epsilon_i$$

where ϵ_i is a random variable with mean zero and x_i is a fixed value vector. Note that before we specify any assumptions, this model could be, for example

- the **linear regression model**, where x_i and ϵ_i are independent
- the **Neyman-Rubin model** ($Y_i = T_i a_i + (1 - T_i) b_i$) in which case both x_i and ϵ_i depend on T_i, the randomization variable, and the pre-experiment variable Z_i is correlated with the potential outcome Y_i.

This expression is effectively a **causal model**: we fix x_i and there is a vector β whose inner product with x_i gives the expected response of y_i, and y_i is composed of taking an error term from a distribution and adding it to the expected response.

12.7 Randomization experiments for effect estimation or AB testing: LS is often used

To answer questions such as "is the new painkiller better than the existing one?", we can use the Neyman-Rubin causal model. Given a group of subjects $i = 1, ..., n$, their potential outcomes are a_i under A and b_i under B, however we only get to observe one of them. We randomize the group into the treatment (A) group or the control (B) group with probability p. This is represented by the treatment variable $T_i \sim Bernoulli(p)$. The observed response for subject i is

$$y_i = T_i a_i + (1 - T_i) b_i$$

We are interested in testing

$$H_0 : \bar{a} = \bar{b}$$

we can use the sample averages from the A and B groups to form a t-test to test H_0.

It turns out that the linear regression model and the Neyman-Rubin model have the same form, but linear regression assumes independence of x_i and ϵ_i, whereas the Neyman-Rubin model has a stochastic assumption (that x_i and ϵ_i both depend on T_i). That is, these are two different models, which can be written in such a way that they look the same. What differs are the *assumptions*.

From now on we follow the standard model

$$Y_i = x_i^T \beta + \epsilon_i$$

such that $E\hat{\beta}_{OLS} = \beta$ iff $E(\epsilon_i | X) = 0$.

12.8 Neyman-Scott problem

In the setup of the Neyman-Scott problem, suppose that (X_i, Y_i) are independent $N(\mu_i, \sigma^2)$, for $i = 1, 2, ..., n$. Then we have $p = n+1$ parameters to estimate $(\mu_1, ..., \mu_n, \sigma^2)$, so this is in the reals of high-dimensional problems as $n \approx p$. It turns out that the MLE for σ^2 is inconsistent

$$MLE(\sigma^2) \to \frac{1}{2}\sigma^2$$

A Bayesian fix is to put a normal prior on μ, and to integrate μ. and use a profile likelihood to estimate σ^2.

13 Weighted least squares

Now consider a Gaussian linear model with heteroscedasticity. Then

$$Y = X\beta + \epsilon$$

where

- $\epsilon_i \sim i.i.d.N(0, \sigma_i^2)$
- ϵ is independent of X

i.e. the variance is not constant across sub populations. If the σ_i^2 are known, then the WLS method is valid without the Gaussian assumption, where the WLS estimate is given by

$$\hat{\beta}_{WLS} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\frac{Y_i - x_i^T \beta}{\sigma_i} \right)^2$$

which downweights noisy observations.

14 Generalized least squares

Suppose now that there may exist some nonzero covariance between the ϵ_i's. In this case, we consider a generalized least squares model

$$Y = X\beta + \epsilon$$

where

- $\epsilon \sim N(0, G)$, where G is a positive definite, symmetric covariance matrix.
- ϵ is independent to X

If G is known, the GLS estimate of β is

$$\hat{\beta}_{GLS} = \underset{\beta}{\operatorname{argmin}} (Y - X\beta)^T G^{-1} (Y - X\beta)$$

We can use SVD of G to turn $(Y - X\beta)G^{-1}(Y - X\beta)$ into a WLS objective function via transformation of both X and Y. We can write $G^{-1} = G^{-\frac{1}{2}} G^{-\frac{1}{2}}$ since the SVD of G (positive definite) tells us that $G = RDR^T$ where R is orthogonal and D is diagonal, so $G^{-1} = RD^{-1}R$, and $G^{\frac{1}{2}} = RD^{\frac{1}{2}}R^T$ and $G^{-\frac{1}{2}} = RD^{-\frac{1}{2}}R^T$.

$$(Y - X\beta)^T G^{-1} (Y - X\beta) = (G^{-\frac{1}{2}} Y - G^{-\frac{1}{2}} X\beta)^T (G^{-\frac{1}{2}} Y - G^{-\frac{1}{2}} X\beta)$$

so

$$\hat{\beta}_{GLS} = \underset{\beta}{\operatorname{argmin}} ||\tilde{Y} - \tilde{X}\beta||_2^2$$

where $\tilde{Y} = G^{-\frac{1}{2}} Y$ and $\tilde{X} = G^{-\frac{1}{2}} X$

14.1 Poisson model example

If $Y_i \sim Poisson$, then $EY_i = var(Y_i)$ so we could use the estimate

$$\sigma_i^2 = |x_i^T \beta|$$

or some scalar multiple $\sigma_i^2 = |x_i^T \beta| \sigma^2$ if $Y_i = x_i^T \beta + N(0, |x_i^T \beta| \sigma^2)$

14.2 Confidence interval

We could write a 95% confidence interval for β_j, $j = 1, 2, ..., p$ as

$$\hat{\beta}_j \pm 1.96\sqrt{(X^T X)^{-1}_{jj}\hat{\sigma}^2}$$

where

$$\hat{\sigma}^2 = \frac{||e||^2}{n-p}$$

14.3 Interpretation of linear model coefficients

When comparing the relative importance of a predictor, it is important not to simply consider the model coefficients alone. We must normalize the coefficients by the standard error of the estimator, i.e. compare the

$$\frac{\hat{\beta}_j}{\hat{SE}(\hat{\beta}_j)}$$

If for example x_1 and x_2 are highly correlated, then it is hard to say which predictor is more important than the other. We could instead estimate $\beta_1 + \beta_2$, assuming $var(X_1) = var(X_2) = 1$. In this case we could estimate α in

$$y = \alpha\tilde{x} + \epsilon$$

in summary

- Always normalize. The normalization that provides the right scale for interpretation is
$$\frac{\hat{\beta}_j}{\hat{SE}(\hat{\beta}_j)}$$
- some $\hat{\beta}_j$'s cannot be considered separately from other correlated $\hat{\beta}_j$'s

to deal with the second point, we could cluster the x_j's first and interpret the group effect by using least squares with super predictors. We could, for example, take the centers of the clusters as our predictors, or from each cluster, take the column of X that is closest to the center of the cluster.

Ideally, however, to avoid the situation in which we have correlated variables, and for ease of interpretation, we want the non-diagonal terms of our covariance matrix for $\hat{\beta}$ to be zero, which is equivalent to

$$X^T X = I_{p\times p}$$

15 Leverage scores and stability

Recall that the Hat matrix (projection matrix) is defined by

$$H = X(X^T X)^{-1} X^T$$

and that

$$\hat{Y} = HY$$

the ith leverage score, h_{ii}, is the ith diagonal entry of H. We claim that (obvious from the formula for $cov(\hat{\beta})$)

$$var(\hat{Y}_i) = h_{ii}\sigma^2$$

this implies that if $h_{ii} \approx 0$, then the variance of $\hat{Y}_i$ is small, which hints at the consistency of $\hat{\beta}_{OLS}$.

Theorem 15.1.

$$0 \leq h_{ii} \leq 1$$

Proof. since $H^2 = H$, we have that

$$h_{ii} = h_{ii}^2 + \sum_{i \neq j} h_{ij}^2 \geq 0$$

so

$$h_{ii}^2 \leq h_{ii}$$

implying that

$$0 \leq h_{ii} \leq 1$$

□

The hat matrix plays an important role. For example, Chebyshev's inequality tells us that

$$\begin{aligned} P(|\hat{Y}_i - E\hat{Y}_i| > \epsilon) &\leq \frac{var(\hat{Y}_i)}{\epsilon^2} \\ &= h_{ii}\frac{\sigma^2}{\epsilon^2} \end{aligned}$$

Theorem 15.2 (Central limit theorem)**.** *Suppose that $X_1, ..., X_n \sim IID$ random variables with $E|X_i|^3 < \infty$, $EX_i = 0$ and $EX_i^2 = 1$ then*

$$\sqrt{n}\bar{X}_n \to N(0, 1)$$

15.1 M-estimation: least absolute deviation versus least squares

An M-estimate of β is given by

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \rho(y_i - x_i^T \beta)$$

where ρ is some function such as

- Least squares: $\rho(x) = x^2$
- Least absolute deviation (a generalization of the median): $\rho(x) = |x|$

15.1.1 Computing the LAD estimate

Using the fact that $\frac{|X|}{|X|} = |X|$, we can iteratively compute the LAD estimate in the following way. Suppose the initial estimate is the OLS estimate. Given a current estimate $\hat{\beta}^{(t)}$, calculate a new LAD estimate by

$$\begin{aligned}\hat{\beta}^{(t+1)} &= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \frac{|y_i - x_i^T \beta|^2}{|y_i - x_i^T \beta^{(y)}|} \\ &= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} w_i(t) |y_i - x_i^T \beta|^2\end{aligned}$$

To reduce the influence of unwanted "outliers", the L^1 loss could be used instead of the L^2 loss. i.e Use LAD over OLS if this is your goal!

16 Residual plots to check normality of errors

We often standardize the residuals and then plot them against different predictors and the fitted values. Look for trends and patterns which are suggestive of transformation to predictors to get better fitting. Large standardized residuals might cluster in predictor space indicating where the current model doesn't work.

17 Multiple testing

In many cases, multiple tests are carried out. Suppose we want to apply t-tests to test m tests of the form $H_0 : \beta = 0$ at level α. If X is orthogonal, the probability that at least one test is rejected is very high even for small m's, **even if all null hypotheses are true**

17.1 Bonferroni correction

The Bonferroni correction involves carrying out each test at level $\frac{\alpha}{p}$ to control the famislywise type-1 error rate at α via a union bound. Suppose that we are simultaneously testing two hypotheses (both at the α-level of $\alpha = 5\%$):

$$H_0^{(1)} : \beta_1 = 0$$

$$H_0^{(2)} : \beta_2 = 0$$

Then the family-wise error rate (the probability of rejecting at lest one null-hypothesis when the null hypothesis is in fact true) is

$$\begin{aligned} P_0(|t_1| > 1.96 \text{ or } |t_2| > 1.96) &= P_0(\{|t_1| > 1.96\} \cup \{|t_2| > 1.96\}) \\ &\leq P_0(|t_1| > 1.96) + P_0(|t_2| > 1.96) \\ &\leq 2\alpha \end{aligned}$$

so as we increase the number of hypotheses, we increase the possibility that we will witness a rare event and therefore increase the probability of rejecting the null hypothesis when it is true. Thus, since we want out family-wise error to be at most α, we should test at the $\frac{\alpha}{m}$-level instead of α.

17.2 False discovery rate (FDR)

More recently, FDR has been a popular approach

$$\text{FDR} = \frac{\text{number of false discoveries}}{\text{number of discoveries}}$$

where "discovery" means "rejection of a null hypothesis". The general procedure involves

1. Order observed p-values $p_1, ..., p_m$
2. Find the max k such that $p_k < \frac{\alpha k}{m}$
3. Reject all hypotheses j with $p_j < p_k$. If there is no such k, reject nothing

18 Logistic regression

A logistic regression model is described by

$$P(Y_i = 1|x_i) = \frac{e^{x_i^T \beta}}{1 + e^{x_i^T \beta}}$$

which assumes that Y_i is a Bernoulli random variable and that the log odds ratio is linear in β, that is,

$$\log\left(\frac{P(Y_i = 1|x_i)}{P(Y_i = 0|x_i)}\right) = x_i^T \beta$$

18.1 The likelihood function

We could estimate $\hat{\beta}$ using the maximum likelihood estimate (MLE). Given the assumption that the $Y_i|x_i$'s are independent, we would have

$$\mathcal{L}(Y_1, ..., Y_n|X) = \mathcal{L}(Y_1|X)...\mathcal{L}(Y_n|X)$$

Note that for the O-ring example, the Y_i's are not independent if we don't take into account the temperature.

It turns out that independence is very important for inference but less important for prediction.

Suppose that the $Y_1, ..., Y_n$'s are conditionally independent given the $X_1, ..., X_n$'s. Then for the logistic regression model, the likelihood is given by

$$L(y_1, ..., y_n|X) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i}$$

where $\pi_i = P(y_i = 1|x_i) = \frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}}$, in which case the log likelihood is given by

$$\begin{aligned}
\ell(y_1, ..., y_n|X) &= \log \prod_{i=1}^{n} L(y_1, ..., y_n|X) \\
&= \sum_{i=1}^{n} (y_i \log \pi_i + (1-y_i)\log(1-\pi_i)) \\
&= \sum_{i=1}^{n} \left(y_i \log \frac{\pi_i}{1-\pi_i} + \log(1-\pi_i) \right) \\
&= \sum_{i=1}^{n} \left(y_i x_i^T \beta - \log\left(1 + e^{x_i^T\beta}\right) \right)
\end{aligned}$$

where the last inequality follows from the fact that $1 - \pi_i = \frac{1}{1+e^{x_i^T\beta}}$. We can use this log-likelihood function to get the MLE and throw away all the stochastic assumptions by using

$$\hat{y}_i = \frac{e^{x_i^T\hat{\beta}_{MLE}}}{1 + e^{x_i^T\hat{\beta}_{MLE}}}$$

as a prediction rule. For example, we could use a cutoff of 0.5 so that

$$\hat{y}_i \leq 0.5 \Rightarrow \tilde{y}_i = 0$$

$$\hat{y}_i > 0.5 \Rightarrow \tilde{y}_i = 1$$

where $\tilde{y}_i$ is our predicted binary response.

18.2 Maximum likelihood estimation

How can we find the MLE, since this log likelihood function has no closed form solution for its maximizer. We can use **Newton-Rhapson (iteratively weighted least squares).**

Suppose, that $p = 1$. Define the **score function** or (U-statistic) to be

$$U(\beta) = \frac{d\ell(\beta)}{d\beta} = \sum_{i=1}^{n} \left(y_i x_i - \frac{x_i e^{x_i \beta}}{1 + e^{x_i \beta}} \right)$$

We will estimate the MLE using iterated linear approximations based on each current estimate. In particular after setting the first estimate to be $\beta^{(0)} = \beta_{OLS}$, the mth estimate of β is found by solving

$$U^{(m)}(\beta) = 0$$

where $U^{(m)}(\cdot)$ is the linear approximation of $U(\cdot)$ at $\beta = \beta^{(m-1)}$. We can estimate $U^{(m)}(\beta)$ by

$$U^{(m)}(\beta) \approx U\left(\beta^{(m-1)}\right) + \frac{dU}{d\beta}\Big|_{\beta=\beta^{(m-1)}} \left(\beta - \beta^{(m-1)}\right)$$

which is similar to a first order Taylor expansion of $U(\cdot)$ about $\beta^{(m-1)}$. Setting $U^{(m)}(\beta)$ to zero, and solving for β to get $\beta^{(m)}$, we get

$$\beta^{(m)} = \beta^{(m-1)} + \left[-\frac{dU}{d\beta}\Big|_{\beta=\beta^{(m-1)}} \right]^{-1} U\left(\beta^{(m-1)}\right)$$

Note that

$$\frac{dU}{d\beta} = \frac{D^2\ell(\beta)}{d\beta^2}$$

and we define the **observed Fisher information** matrix to be

$$I = -\frac{d^2\ell}{d\beta^2}$$

If we have $p \geq 1$, then the estimate of β at the mth iteration, $\beta^{(m)}$ can be found by calculating

$$\beta^{(m)} = \beta^{(m-1)} + \left[-\frac{\partial^2\ell}{\partial\beta_i \partial\beta_j}\Big|_{\beta=\beta^{(m-1)}} \right]^{-1} U\left(\beta^{(m-1)}\right)$$

where $U(\beta) = \left(\frac{\partial\ell}{\partial\beta_1}, ..., \frac{\partial\ell}{\partial\beta_p}\right)$. In this case, the observed Fisher information matrix is defined by

$$I_{ij} = -\frac{\partial^2\ell}{\partial\beta_i\beta_j}$$

18.3 Comparing the least squares and maximum likelihood estimates

For the logistic regression model (challenger data), the least squares estimate can be found by finding the argmin of

$$LS(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

where y_i is a binary response variable indicating O-ring failure or not, and x_i is the temperature. On the other hand, the log-likelihood for this problem is given by

$$-\ell(\beta_0, \beta_1) = -\sum_{i=1}^{n} (Y_i \log \pi_i - (1 - Y_i) \log(1 - \pi_i))$$

where $\pi_i = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}}$.

19 The link function

Suppose that $Y_i \sim Bernoulli(\pi_i)$. The key idea behind the link function is to link π_i to x_i. Note that $E(Y_i) = \pi_i$.

For the least squares model outlined above, $E(Y_i) = x_i^T \beta$, implying that the relationship between π_i and x_i is given by $\pi_i = x_i^T \beta$, and we will thus define a function g by

$$\pi_i = x_i^T \beta = g^{-1}(x_i^T \beta)$$

in this case, g is simply the identity function $g(a) = a$.

For the logistic regression model above,

$$E(Y_i) = \pi_i = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}} = g^{-1}(x_i^T \beta)$$

so in this case,

$$g(a) = \log\left(\frac{a}{1 - a}\right)$$

The function g that links π_i and x_i is called the **link function**. More specifically, the link function is the function g such that

$$g(\pi_i) = x_i^T \beta$$

Note that any CDF can be used as the inverse of a link function. In particular, if we take the inverse of the link function to be $\Phi(\cdot)$, then we get the profit model for binary responses, where

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{t}{2}} dt$$

20 Model checking and validation

There are many ways to check how well a model fits the data

20.1 Residuals

We could examine the standardized residuals given by

$$\frac{y_i - \hat{y}_i}{\hat{SE}(\hat{y}_i)}$$

however, this is not informative if the y_i's are 0's and 1's ($y_i \sim Bernoulli(\pi_i)$). We could instead aggregate similar y_i's into groups or group replicates, and define the new observations to be the number of original observations which fall into the bins generated by the aggregation. These new observations would be approximately Binomially distributed ($y_i \sim Binomial(\pi_i, n_i)$, and we assume that in any bin the π_i's are similar. For the aggregated observations we have

$$var(y_i) = \pi_i(1 - \pi_i)$$

which might be estimated using

$$\hat{\pi}_i = \frac{e^{x_i^T \hat{\beta}_{MLE}}}{1 + e^{x_i^T \hat{\beta}_{MLE}}}$$

if π_i is not too different from $\frac{1}{2}$ and n_i is not too small, then approximately

$$y_i \sim N(n_i\pi_i, n_i\pi_i(1 - \pi_i))$$

20.2 Prediction

Classification error can be used to check how well a model is performing

$$\text{classification error } = \frac{\#\{\text{false positive}\} + \#\{\text{false negative}\}}{\#\{\text{test samples}\}}$$

20.3 Likelihood ratio test (F-test in Gaussian linear regression)

For a saturated model, we estimate

$$\tilde{\pi}_i = \frac{y_i}{n_i}$$

where $y_i \sim Binomial(\pi_i, n_i)$. For an unsaturated model (e.g. logistic regression) used to predict $\hat{\pi}_i$, a good model will have

$$\hat{\pi}_i^{(MLE)} \approx \tilde{\pi}_i$$

The likelihood ratio estimate is an aggregated measure of the collective closeness of $(\tilde{\pi}_i)_{i=1}^m$ and $(\hat{\pi}_i)_{i=1}^m$. In particular, the likelihood ratio is defined by

$$R^* = \frac{\text{likelihood(data, } \{\tilde{\pi}_i\})}{\text{likelihood(data, } \{\pi_i\left(\hat{\beta}^{(MLE)}\right)\})}$$

and we define a quantity called the **deviance** by

$$\begin{aligned} \text{Deviance } &= 2\log R^* \\ &\approx \sum_{i=1}^{m}\left[\frac{y_i - \hat{y}_i}{\hat{SE}(\hat{y}_I)}\right]^2 \end{aligned}$$

which is the sum of the residuals. Note that asymptotically

$$Deviance \sim \chi^2_{m-p}$$

where m is equal to the number of groups into which we have aggregated the data.

21 The bootstrap

We can use the bootstrap method to estimate properties of $\bar{X}$. Consider the observed data as a "little population". Next, simulate n draws, made at random *with replacement* to get a bootstrap sample $X_1^*, ..., X_n^*$. From this bootstrap sample, we can generate a bootstrap estimator $\bar{X}^* = \frac{1}{n}\sum_{i=1}^n X_i^*$. We generate M bootstrap replicates in order to obtain M bootstrap estimators $\bar{X}^*_{(1)}, ..., \bar{X}^*_{(M)}$.

The idea is, that if the sample size is large, the distribution of $\bar{X}^* - \bar{X}$ will be a good approximation to the distribution of $\bar{X} - \mu$. In particular, we should have that

$$P\left(\frac{\bar{X} - \mu}{\sigma_X} < t\right) \approx P\left(\frac{\bar{X}^* - \bar{X}}{\sigma_X} < t\right)$$

and the process of sampling from the population should be comparable to the process of taking a bootstrap sample from our overall sample.

since the bootstrap estimator $\bar{X}^*$ is generated by sampling from the real *sample* whose mean is $\bar{X}$ (not μ), $\bar{X}^*$ is not a new estimator for the parameter μ, but rather is something generated to better understand the behavior of the estimator $\bar{X}$, that we started with. For example, $\bar{X}^*_{ave} = \frac{1}{M}\sum_{k=1}^M \bar{X}^*_{(k)}$ and we can estimate the bias of $\bar{X}$ by

$$bias(\bar{X}) = \bar{X} - \bar{X}^*_{ave}$$

and the SE of $\bar{X}$ by

$$SE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}[\bar{X}^*_{(i)} - \bar{X}^*_{ave}]}$$

The bootstrap SE says how good the original $\bar{X}$ was, as an estimate for μ.

21.1 Standard regression model: sampling the residuals

There are many ways to apply bootstrapping to a regression problem. Suppose that we have the model

$$Y = X\beta + \epsilon$$

with $\epsilon_i \sim N(0, \sigma^2)$. Suppose we're interested in obtaining properties of $\hat{\beta}_{OLS}$ but don't know any closed formulas. We could resample the residuals. In this case our "little population" is the residuals $e_1, ..., e_n$ and we could draw from this little population n times at random to get bootstrap sample $\epsilon^*_1, ..., \epsilon^*_n$. For each bootstrap sample, we then regenerate the Y_i's to get the Y_i^*'s:

$$Y^* = X\hat{\beta}_{OLS} + \epsilon^*$$

For the kth bootstrap model, we can calculate the kth bootstrap estimate of $\hat{\beta}_{OLS}$ using

$$\hat{\beta}^*_{(k)} = (X^T X)^{-1} X^T Y^*_{(k)}$$

The idea is that if the sample size is large, then the distribution of $\hat{\beta}^*_{(k)} - \hat{\beta}_{OLS}$ is a good approximation to the distribution of $\hat{\beta}_{OLS} - \beta$ and in particular

$$\sqrt{n}(\hat{\beta}_{OLS} - \beta) \stackrel{\mathcal{L}}{=} \sqrt{n}(\hat{\beta}^* - \hat{\beta}_{OLS})$$

Note that the CLT implies that

$$|\mathcal{L}(\sqrt{n}(\hat{\beta} - \beta)) - N(0, \sigma^2(X^T X)^{-1})| < o(\frac{1}{\sqrt{n}})$$

However, note that when ϵ has heavy tails, CLT and bootstrap methods don't work well.

21.2 Logistic regression: sampling the observations

We could sample from our little population $(x_1, y_1), ..., (x_n, y_n)$ to get the bootstrap sample $(x^*_1, y^*_1), ..., (x^*_n, y^*_n)$, from which we could obtain $\hat{\beta}^*_{MLE}$.

22 Generalized linear models

22.1 Exponential family

The exponential family is the one-dimensional density family indexed by parameter θ described by

$$f(y,\theta) = \exp\{a(y)b(\theta) + c(\theta) + d(y)\}$$

Note that if $a(Y) = Y$ (canonical parametrization), then

$$E(Y) = -\frac{x'(\theta)}{(b'(\theta))^3}$$

and

$$Var(Y) = \frac{b''(\theta)c'(\theta) - c''(\theta)b'(\theta)}{(b'(\theta))^3}$$

To prove these facts, note that for all θ,

$$\int f(y,\theta)dy = 1$$

which implies tat

$$\frac{d}{d\theta}\int f(y,\theta)dy = \frac{d^2}{d\theta^2}\int f(y,\theta)dy = 0$$

switching integration and differentiation gives

$$\begin{aligned}
0 &= \int \frac{d}{d\theta} f(y,\theta)dy = \int \frac{d}{d\theta}[\exp\{a(y)b(\theta) + c(\theta) + d(y)\}]dy \\
&\Rightarrow \int (b'(\theta)y + c'(\theta))f(y,\theta)dy = 0 \\
&\Rightarrow b'(\theta)E(Y) + c'(\theta) = 0 \\
&\Rightarrow E(Y) = -\frac{x'(\theta)}{b'(\theta)}
\end{aligned}$$

22.2 GLMs

Note that for GLMs we need

- The **distribution of** Y_i: the Y_i's are independent and from $f(y,\theta_i)$ which is an exponential family density
- The **link function**, g (strictly monotone), which describes the relationship

  $$g(\mu_I) = x_i^T\beta$$

  where $\mu_i = \mu(\theta_i) = EY_i$.

Define $\mu_i = E(Y_i)$. The fundamental assumption for a generalized linear model is that there exists a transformation (link function), g, of μ_i of the form

$$g(\mu_i) = x_i^T \beta$$

For example, for the standard linear regression model, where Y_i is normally distirbuted

$$Y_i = x_i^T \beta + \epsilon_i$$

the link function is simply the identity function $g(\mu_i) = \mu_i$ because $\mu_i = E(Y_i) = x_i^T \beta$.

For the logistic regression model, where Y_i has a Bernoulli distribution, the link function is of the form

$$g(\mu_i) = \log\left(\frac{\mu_i}{1-\mu_I}\right)$$

22.3 Maximum likelihood for GLMs/Iteratively reweighed least squares

We will show that the MLE of β for GLMs is an IRWLS estimate. Recall that

$$E(Y_i) = \mu_i = -c'(\theta_i)/b'(\theta_i)$$

and also that

$$g(E(Y_i)) = g(\mu_i) = x_i^T \beta$$

Define

$$\eta_i = x_i^T \beta$$

in which case

$$g(\mu_i) = \eta_i$$

If the observations are independent and the x_i's are fixed, the likelihood function is given by

$$\begin{aligned}\ell(\beta) &= \sum_{i=1}^{n} \ell_i(\beta) \\ &= \sum_{i=1}^{n} [a(Y_i)b(\theta_i) + c(\theta_i) + d(Y_i)]\end{aligned}$$

we are interested in finding the value of β that maximizes $\ell(\beta)$, and we can do this using Newton-Rhapson. Recall the score function $U(\beta) = \Delta\ell(\beta)$.
Using the chain rule for differentiation, we have

$$[U(\beta)]_j = \frac{\partial \ell(\beta)}{\partial \beta_j} = \sum_{i=1}^{n} \frac{d\ell}{d\theta_i}\frac{d\theta_i}{d\mu_i}\frac{d\mu_i}{d\eta_i}\frac{d\eta_i}{d\beta_j}$$

where for the canonical parametrization, we have

$$\begin{aligned}\frac{d\ell}{d\theta_i} &= a(Y_i)b'(\theta_i) + c'(\theta_i) \\ &= Y_i b'(\theta_i) + c'(\theta_i) \\ &= b'(\theta_i)\left(Y_i + \frac{c'(\theta_i)}{b'(\theta_i)}\right) \\ &= b'(\theta_i)(Y_i - \mu_i)\end{aligned}$$

Next,

$$\begin{aligned}\frac{d\theta_i}{d\mu_i} &= \frac{1}{d\mu_i/d\theta_i} \\ &= -\left(\frac{b'(\theta_i)c''(\theta_i) - b''(\theta_i)c'(\theta_i)}{(b'(\theta_i))^2}\right)^{-1} \\ &= (b'(\theta_i)Var(Y_i))^{-1}\end{aligned}$$

and finally,

$$\frac{d\eta_i}{d\beta_j} = x_{ij}$$

Thus our score function simplifies to

$$[U(\beta)]_j = \sum_{i=1}^{n} \frac{Y_i - \mu_i}{Var(Y_i)} \frac{d\mu_i}{d\eta_i} x_{ij}$$

The covariance matrix of $[U(\beta)]_j$ is the **Fisher information matrix**, and is given by

$$\mathcal{J} = E(-H(\ell)) = E(U(\beta)U(\beta)^T)$$

where $H(\ell)$ is the Hessian matrix of the likelihood function. So

$$\begin{aligned}\mathcal{J}_{jk} &= E\left[\left(\sum_i \frac{Y_i - \mu_i}{Var(Y_i)} \frac{d\mu_i}{d\eta_i} x_{ij}\right)\left(\sum_l \frac{Y_l - \mu_l}{Var(Y_l)} \frac{d\mu_l}{d\eta_l} x_{lk}\right)\right] \\ &= E\left[\sum_i \frac{(Y_i - \mu_i)^2}{(Var(Y_i))^2} \left(\frac{d\mu_i}{d\eta_i}\right)^2 x_{ij}x_{ik}\right] \\ &= \sum_i \frac{x_{ij}x_{ik}}{Var(Y_i)} \left(\frac{d\mu_i}{d\eta_i}\right)^2\end{aligned}$$

which implies that

$$\mathcal{J} = X^T W X$$

where the weight matrix has entries

$$W_{ii} = \frac{1}{Var(Y_i)} \left(\frac{d\mu_i}{d\eta_i}\right)^2$$

22.3.1 Netwon-Rhapson/Fisher-Scoring iteration

The Fisher-scoring iteration method for estimating β is thus

$$\beta^{(t)} = \beta^{(t-1)} + \left[\mathcal{J}^{(t-1)}\right]^{-1} U(\beta^{(t-1)})$$

Multiplying both sides by $\mathcal{J}^{(t-1)}$, we get

$$\mathcal{J}^{(t-1)}\beta^{(t)} = \mathcal{J}^{(t-1)}\beta^{(t-1)} + U(\beta^{(t-1)})$$

which is equivalent to

$$\sum_{k=1}^{p}\sum_{i=1}^{n} \frac{x_{ij}x_{ik}}{Var(Y_i)} \left(\frac{d\mu_i}{d\eta_i}\right)^2 \beta_k^{(t)} = \sum_{k=1}^{p}\sum_{i=1}^{n} \frac{x_{ij}x_{ik}}{Var(Y_i)} \left(\frac{d\mu_i}{d\eta_i}\right)^2 \beta_k^{(t-1)} + \sum_{i=1}^{n} \frac{Y_i - \mu_i}{Var(Y_i)} \frac{d\mu_i}{d\eta_i} x_{ij}$$

where μ_i and $\partial\eta_i/\partial\mu_i$ are evaluated at $\beta^{(t-1)}$. This can be written in matrix form as

$$X^T W X \beta^{(t)} = X^T W Z$$

where Z has elements

$$Z_i = \sum_{k=1}^{p} x_{ik}\beta_k^{(t-1)} + (Y_i - \mu_i)\frac{d\eta_i}{d\mu_i}$$

Note that this is very similar to the normal setting for a linear model obtained by least squares, but it has to be solved iteratively since Z and W depend on β. This, for generalized linear models, MLEs are obtained by an iterative weighted lest squares procedure.