XI. Exact Methods

Lecture 11

J. Bradley–Terry Model for Paired Preferences

1. Compare items pairwise
2. Try to compare them, using all results at the same time.
3. Use more information than just pairs
4. Model probability for item \( j \) beating \( k \) as
   \[
   \exp(\beta_j - \beta_k)/(1 + \exp(\beta_j - \beta_k))
   \]
5. Adding a constant to each of the \( \beta \) keeps probabilities the same
   a. Hence one of the items must be taken as baseline
6. Note that \( P [j \text{ beats } k] = 1 - P [k \text{ beats } j] \), as it should.
7. \( \beta_k \) represents strength of item \( k \)
8. Can add intercept for “home team advantage”,
9. Model fit is same as for quasi-symmetry model

A. 5.4.2–5.4.3

6. Computation
   a. Model:
   \[
   X_{ij} \sim \mathcal{P}(\lambda_{ij})
   \]
   b. \( \log(\lambda_{ij}) = \alpha_i + \beta_j + \gamma_{ij} \)
   c. \( \beta_j = 0, \gamma_{ij} = \gamma_{ji} = 0 \forall i, j \).
2. \( H_0 : \gamma_{ij} = 0 \forall i, j \) vs. \( H_A : \gamma_{ij} \neq 0 \) for some \( i, j \).
3. Test statistics:
   a. Score statistic is Pearson \( \chi^2 \):
   \[
   T = \sum_i (X_{ij} - X_i X_j/X_{..})^2/(X_i X_j/X_{..})
   \]
   b. LR statistic
   c. Fisher’s statistic \( 1/P [X] \)
   4. Remove effect of unknown parameters:
      a. Remove \( \alpha_i \) by conditioning on \( X_i \)

   - That is, note that for any common \( \pi_0 \)
     value, \( P_{\pi_0} \left[ \sup_{\pi \in \{0,1\}} P_\pi [Z \geq 2] \right] \leq \sup_{\pi \in \{0,1\}} P_\pi [Z \geq 2] \).
   - Convexity condition: If test rejects for \( (x_1, x_2) \),
     then test rejects for more extreme \( (x_1, x_2 + 1) \) and \( (x_1 - 1, x_2) \).
   - Extend to \( p \)-values: \( p(x_1, x_2) \geq p(x_1, x_2 + 1) \)
     and \( p(x_1, x_2) \geq p(x_1 - 1, x_2) \).
   - Implies that one need only look along boundary of alternative hypothesis for maximizer.
   - That is, \( p \)-value is the same whether we test
     \( H_0 : \delta = \delta_0 \) vs. \( H_A : \delta > \delta_0 \) or \( H_0 : \delta < \delta_0 \) vs.
     \( H_A : \delta > \delta_0 \).
   - Heuristically, because rejection region probabilities become less
   - Need to check for convexity: \( Z \) statistics, Fisher’s exact test all work.
   - Can also phrase question in terms of relative risk \( \pi_2/\pi_1 \).
3. Result is hypergeometric distribution from \( 2 \times 2 \) table
   - Hence \( P [X = x | X_i = x_i] = \prod_i p_{ij} \) for hypergeometric probabilities \( p_{ij} \).

5. Difference not on log scale
   a. \( H_0 : \pi_2 = \pi_1 + \delta \) vs. \( H_A : \pi_2 > \pi_1 + \delta \)
   b. Different from previous if \( \delta \neq 0 \).
   c. Important practical problem: non-inferiority trial
   d. No parallel conditional analysis
      i. Not a canonical exponential family

\[
(1 - \pi_1)^{n_1 - x_1} (1 - \delta - \pi_1)^{n_2 - x_2} \pi_1^{x_1} (\delta + \pi_1)^{x_2} \Gamma(1 + n_1) \Gamma(1 + n_2)
\]
\[
(1 + n_1 - x_1) \Gamma(1 + x_1) (1 + n_2 - x_2) \Gamma(1 + x_2)
\]
\[\times \Gamma(\pi_1(1 - \pi_1)/n_1 + \pi_2(2 - \pi_2)/n_2)/n_1 \]
\[\times \Gamma(\pi_1/\pi_2)/n_2 \]

   - Called \( \delta \)-projected \( Z \) statistic
   - \( \pi_1 \) maximizing likelihood is given by a cubic equation
   - Remove nuisance parameter effect using suprema

B. Logistic Regression (Hirji, Mehta, and Patel, 1987)

2. \( X_j \sim \text{Bin}(\exp(x_j\theta)/(1 + \exp(x_j\theta)), n_j) \)
   i. Suppose \( n_j = 1 \)
   d. \( T = Z^T X \)
   e. \( \text{probabilities } \exp(t^T \theta - \sum_j n_j \log(1 + \exp(z_j\theta))c(t) \text{ for } \}
   f. \( i. c(t) \text{ the number of } x \text{ vectors with } Z^T x = t \).
   c. \( T = (U, V) \text{, } \theta = (\omega, \tau) \)
ii. $P [V = v | U = u] = c(u, bv) \exp(v\tau) / \sum_v c(u, bv) \exp(v\tau)$

So we need algorithm to generate list of $v$ consistent with $u$, and to calculate $c(u, v)$ for these $u$

h. Let

i. $\Omega_i$ be sample space using observations 1, \ldots, $i$, satisfying conditioning statement

ii. $c_i$ be counts of $X_1, \ldots, X_i$ ensembles giving $t$

i. Note that

ii. $\Omega_1 = \{0, z_1\}$

iii. $\Omega_i = \Omega_{i-1} \cup (z_i + \Omega_{i-1})$

- After removing duplicates

iv. $c_i(t) = c_{i-1}(t) + c_{i-1}(t - z_i)$

j. Collects all possible $t$

i. Excessive: we only need vectors consistent with conditioning event.

ii. Algorithm more efficient if we can eliminate from $\Omega_i$ many entries that can never satisfy conditioning event

iii. Easiest condition to implement: dump those if component gets too large or small.