XI. Exact Methods

A: 5.4.2–5.4.3

Lecture 11

4. Note that \( P_{ij} = P[j \text{ beats } k] = 1 - P[k \text{ beats } j] \), as it should.

7. \( \beta_k \) represents strength of item \( k \)

8. Can add intercept for "home team advantage".

Note that \( \pi_{ij} = \pi_{ji} \).

A. Bradley–Terry Model for Paired Preferences

1. Suppose \( H_0 : \pi_{ij} = \pi_{ji} \).

2. Test reject for \((x_1, x_2)\), then test reject for more extreme \((x_1, x_2 + 1)\) and \((x_1 - 1, x_2)\).

3. Extend to \( p\)-values: \( p(x_1, x_2) \geq p(x_1, x_2 + 1) \) and \( p(x_1, x_2) \geq p(x_1 - 1, x_2) \)

4. Implies that one need only look along boundary of alternative hypothesis for maximizer.

5. That is, \( p\)-value is the same whether we test \( H_0 : \delta = \delta_0 \) vs \( H_A : \delta > \delta_0 \) or \( H_0 : \delta \leq \delta_0 \) vs \( H_A : \delta > \delta_0 \)

6. Heuristically, because rejection region probabilities become less

7. Need to check for convexity: \( Z \) statistics, Fisher’s exact test all work.

8. Can also phrase question in terms of relative risk \( \pi_2/\pi_1 \).

9. Reduce \( I \times J \) independent Poisson variables to \( J \) independent multinomials, each with \( I \) bins.

10. \( X_i \) are exactly ancillary

11. Little loss due to discreteness

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

1. \( X_j \sim \operatorname{Bin}(\exp(z_\theta)/1 + \exp(z_\theta), n_j) \)

2. Suppose \( n_1 = 1 \)

3. \( T = Z^\top X \)

4. \( \exp(\alpha^\top \theta - \sum_j n_j \log(1 + \exp(z_j \theta)))c(t) \), for \( c(t) \) the number of \( x \) vectors with \( Z^\top x = t \).

5. Conditional probabilities

6. \( T = (U, V), \theta = (\omega, \tau) \)
ii. \( P[V = v | U = u] = \frac{c(u, bv) \exp(v\tau)}{\sum_v c(u, bv) \exp(v\tau)} \)

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 \( t \).
   iii. Note that
       i. \( \Omega_1 = \{0, z_1\} \)
       ii. \( c_1(0) = c_1(z_1) = 1 \).
       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.