Magic Method

We are given a group of n students , each of which wrote some subset of m tests. We wish to create an overall ranking of these students, taking into account that not all students wrote all of the tests. Thus if a student did not write a given test, we should try to predict how the student would have performed at the given test had they written it. This should take into account how the student performed in other tests, and adjust the prediction according to how difficult those tests were.

The method which we utilise attempts to create an idealised score s_i for student i which captures the student's ability. The students are then ranked according to this score s_i. In addition, for each test $j \in \{1, \ldots, m\}$, we wish to determine a difficulty rating d_j. The goal is to choose these values in such a way that if the actual score that student i attained on test j is a_{ij}, then $s_i d_j \approx a_{ij}$.

Given an allocation of scores s_i to the students, and an allocation of difficulty levels d_j to the tests, we define a "badness" value given by

$$B(s_1, \ldots, s_n, d_1, \ldots d_m) = \sum_{i=1}^{n} \sum_{j \in T_i} w_j (s_i d_j - a_{ij})^2. \tag{1}$$

Here $T_i \subseteq \{1, 2, \ldots, m\}$ is the subset of tests written by student i, and w_j is a weighting that captures how important test j is. The better an approximation $s_i d_j$ is for the actual score a_{ij}, the smaller this "badness" score will be. We thus seek to minimise this quantity.

We note that we obtain the same result for the "badness" by scaling all of the student scores by a factor λ, and simultaneously scaling all of the test difficulty levels by a factor $1/\lambda$. We thus additionally impose the normalising constraint

$$\sum_{j=1}^{m} d_j = 1. \tag{2}$$

We can minimise Equation 1 subject to the constraint in Equation 2 by using the method of *Lagrange Multipliers*. We obtain the system of equations

$$\sum_{j \in T_i} w_j d_j (s_i d_j - a_{ij}) = 0 \tag{3}$$

$$\sum_{\substack{1 \le i \le n \\ j \in T_i}} w_j s_i (s_i d_j - a_{ij}) = \lambda. \tag{4}$$

Here 3 represents one equation for each value of $i \in \{1, \ldots, n\}$, and 4 represents one equation for each value of $j \in \{1, \ldots, m\}$.

It is difficult to solve this system of equations explicitly. We note, however, that if we already an allocation of difficulties to the tests (d_j), then Equation 3 allows us to calculate an allocation of scores to the students:

$$s_i^\star = \frac{\sum_{j \in T_i} w_j d_j a_{ij}}{\sum_{j \in T_j} w_j {d_j}^2} \tag{5}$$

Similarly, if we already have an allocation of scores for the students, then Equation 4 allows us to calculate an allocation of difficulties to the tests:

$$d_j^\star = \frac{\lambda + \sum_{j \in T_i} w_j s_i a_{ij}}{\sum_{j \in T_i} w_j s_i^2} \tag{6}$$

Here λ should be chosen so that ${d_j}^\star$ satisfies the normalising constraint

$$\sum_{j=1}^{m} d_j^\star = 1.$$

We note that for a given selection (s_i) of student scores, the values $(d_j^\star)$ are those that minimise the badness score, and for a given selection (d_j) of difficulties, the values $(s_i^\star)$ are those that minimise the badness score. Thus if we have an initial estimate (s_i) for the student scores, and an initial estimate (d_j) for the test difficulties, then it follows that

$$B(s_1, \ldots, s_n, d_1^\star, \ldots, d_m^\star) \leq B(s_1, \ldots, s_n, d_1, \ldots, d_m)$$

and

$$B(s_1^\star, \ldots, s_n^\star, d_1, \ldots, d_m) \leq B(s_1, \ldots, s_n, d_1, \ldots, d_m).$$

It follows that replacing either the scores (s_i) with $(s_i^\star)$ or the difficulties (d_j) with $(d_j^\star)$ will give us a better estimate of the correct values for the scores and the difficulties.

We thus iteratively apply the following algorithm until the badness score changes by less than some tolerance ε between two different iterations:

1. Start with an estimate (s_i) for the student scores and (d_j) for the test difficulties.

2. Replace the student scores with the values $(s_i^\star)$ obtained using Equation 5 applied to the current estimate (s_i) and (d_j).

3. Replace the test difficulties with the values $(d_j^\star)$ obtained using Equation 6 applied to the updated estimate $(s_i^\star)$ and (d_j) from Step 2.