KNEWTON DATA CHALLENGE

JOHN WANG

1. Problem Formulation

Let there be n individuals taking an exam each year, each of whom take k of the K total questions available. We require that L of the K questions be offered to at least one student, and also that $0 < k < K$ be satisfied. In this paper, I will examine strategies for finding a ranking of students given a previous year's results as training data.

2. Question Difficulty

In order to do this, we want to formulate some measure of the difficulty of each question j. This motivates our examination of r_j, the probability that a student will get question j correct. To estimate r_j, one could naively use the sample mean from the training data for each question:

$$r_j \approx \frac{1}{n_j} \sum_{i=1}^{n} x_{ij} \tag{1}$$

Where x_{ij} denotes whether or not individual i answered question j correctly and n_j is the number of times question j was asked in the training data. This scheme works as long as the questions were assigned uniformly at random. However, as soon as there exists dependence among the questions, then this technique no longer works. If some set of questions q_1 are assigned to a group of students s_1 with higher probability than other questions and s_1 has a higher intelligence level than the average student, then r_j for $j \in q_1$ will be biased upwards.

Therefore, it is necessary to introduce a new variable θ_i which captures the intelligence level of student i. The probability that student i answers question j correctly will now have an additive factor of θ_i and will be given by $r_j + \theta_i$. If $r_j + \theta_i \geq 1$, then student i always answers j correctly, and if $r_j + \theta_i \leq 0$, then student i never answers question j correctly. Let $\gamma_{ij} = 1$ if student i was assigned question j and $\gamma_{ij} = 0$ otherwise. We will try to minimize the distance to the probability prediction:

$$\sum_{i=1}^{K} |x_{ij} - (r_j + \theta_i)|\gamma_{ij} \tag{2}$$

Using this distance function, we will attempt to estimate r_j for all j and θ_i for all i with the following algorithm. We start by initializing $\theta_i = 0$ for all i and $r_j = \frac{1}{n_j} \sum_{i=1}^{n} x_{ij}$ for all j. Now will will loop for T iterations. Suppose we are in iteration $t > 0$ and $t \leq T$. We will first update all intelligence values θ_i for all i by estimating $\sum_{j=1}^{K} |x_{ij} - (r_j + \theta_i)|\gamma_{ij}$. We will also estimate this with θ_i replaced with $\theta_i + \Delta_t$ and $\theta_i - \Delta_t$. The algorithm then replaces θ_i with $\theta_i - \Delta_t, \theta_i, \theta_i + \Delta_t$ depending on which one gives the smallest distance. We do this for all i and do the same for r_j by calculating $\sum_{i=1}^{n} |x_{ij} - (r_j + \theta_i)|\gamma_{ij}$ for $r_j - \Delta_t, r_j, r_j + \Delta_t$. We do this for all j and finish the iteration.

To calculate the distance offset Δ_t at each iteration t, we will use the function $\Delta_t = \frac{1}{2}e^{-t}$. This allows our distance to decrease at each iteration in an exponential manner. Intuitively this allows the algorithm to explore the sample space at the beginning and narrow down the search in later iterations.

To prove that the resulting r_j and θ_i will be good approximations of their actual values, we will show that our distance to the optimal result is bounded. Let r_j^* be the optimal value of r_j and θ_i^* be the optimal value of θ_i for all j and i. The distance using these values is given by:

$$E\left[\sum_{j=1}^{K} |x_{ij} - (r_i^* + \theta_i^*)|\gamma_{ij}\right] = round(k(r_i^* + \theta_i^*)) - k(r_i^* + \theta_i^*) \tag{3}$$

Here, $round(.)$ denotes the function that rounds . to the nearest integer. The expression follows because there will be k questions for which $\gamma_{ij} = 1$ for each student i, and since there can only be an integer number of $x_{ij} = 1$.