Problem 2 – Modeling Optimal Teams

Shamay G Samuel, Donnie Sahyouni, Zhongqi Chen

November 12, 2017

Contents

1 Nontechnical Summary

The preparation for a math modeling competition is beginning at a large on-line university. There are exactly 300 students who are participating, and 100 teams of 3 students will need to be formed by the competition organizer. Each student has submitted their scores of Mathematics (M), Computer programming (C), Writing skills (W), and Personality (P). When grouped together, each team member's score in the four sections will contribute to the overall performance of their team, which can be positive or negative. Our goal is to build a mathematical model which will allow us to predict the performance of a team given the three members, to decide on the best team that can be put together, and to strategically select a third person given two people in the team.

Our first step was to determine four functions for evaluating the overall team scores in the four sections (M, C, W, and P) given the individual scores of each group members. These functions would take into account both the natures of different attributes and the three facts presented by the problem. After finding these functions, we created a probabilistic model, or the `score predictor` that predicted the distribution of a team's result given the team scores in each individual section. The overall idea that we follow is that scores in Mathematics and Computer programming serve as the base for a team's performance, and scores in Writing and Personalities will serve as multipliers for a team's performance.

We therefore constructed an algorithm that models the above probabilistic model. The algorithm was then run against the first dataset provided; and the result of the test, predictions for the best team and proposals for sample searches for a third team member were obtained.

In conclusion, we developed a mathematical model that provides a reasonable estimate for a team's performance given the attributes of each team member. With this model, the on-line contest organizer will no longer struggle in assigning the teams and will surely optimize its assignment decision to create a fun and fair competition!

2 Introduction

We attempt to find efficient and accurate methods to predict a team's outcome in the math modeling competition by considering the attribute scores (which are deterministic and given) of each individual team member and how each attribute contributes to the expected outcome.

1. M scores range from 0 to 10, with 10 being the highest.
2. C scores range from 0 to 10, with 10 being the highest.
3. W scores range from 0 to 10, with 10 being the highest.
4. P scores range from 0 to 29, indicating one of 30 personality types.

We don't see a correlation between M and C scores, but it is apparent that $M + C$ is to some degree negatively correlated with W. That is, those who are above average in mathematics and computer programming are typically not above average in writing. P seems to be independent of W, M, and C.

M, C, P, and W seem to affect the team score in the following ways:

1. Greater M and C scores generally correlate with better team result, but a single person seems to be able to "carry" the team, but other members also contribute with no negative contributions. That is, a single team member with talent in mathematics or computer programming is able to handle all the responsibilities in their respective skill area, but other teammates can make marginal contributions.
2. High W act as a multiplier for M and C in the final score, and thus, contribute a lot more to the score than M and C. And because multiple people are expected to contribute to writing the report, the lowest W score should bring down the team W score.
3. Teams with personality types that are very close together tend to do better than teams with very distinct personalities. (*Note:* P values being "close" is in terms of modulo distance. I.e. $2 = dist(2, 4) = dist(1, 29) = 2$)

We found that team M, C, and W scores were easily modeled by special cases of generalized means[1]. We modeled personality with use of clever geometric intuition, and we expressed team scores by a normal distribution whose mean μ is a function of M, C, P, and W and whose variance is constant.

[1] The class of functions $M(x_1, x_2 \cdots x_n) = (\frac{1}{n}\sum_{i=1}^{n} x_i^p)^{\frac{1}{p}}$

3 Developing the Predictor Distribution

In the following sections we describe our methods for converting a team's set of math (M), computer (C), personality (P), and writing (W) scores to a deterministic mean score function and then to a stochastic distribution of actual scored outcomes. This "predictor function" takes as input each team member's M, C, P, and W scores (12 values) and outputs the distribution of that team's performance at a modeling competition.

3.1 Modeling M, C, P, and W

We seek to define team scores for M, C, P, and W as to simplify the process in determining how M, C, P, and W affect the overall team score.

Assumption 1: Team math and computer science scores are modeled by the cubic mean of the individual math and science scores.

That is, for given M_1, M_2, M_3:

$$C(M_1, M_2, M_3) = \sqrt[3]{\frac{M_1^3 + M_2^3 + M_3^3}{3}}$$

Where C is the cubic mean function [2].

The cubic mean allows for a single person to "carry" the team in the sense that large values of M_i impact the mean more than small M_i.

To illustrate:

Define $M_s(a, b, c)$ to be the team math score with individual math scores a, b, and c.

Further define $C_s(a, b, c)$ to be the team computer science score with individual computer science scores a, b, and c.

Our model predicts the following relationships, though some of them might not be intuitive:

1. $M_s(10, 1, 1) > M_s(5, 5, 5)$ and $C_s(10, 1, 1) > C_s(5, 5, 5)$
2. $M_s(10, 7, 1) > M_s(10, 5, 5)$ and $C_s(10, 7, 1) > C_s(10, 5, 5)$
3. $M_s(10, 5, 5) > M_s(10, 1, 1)$ and $C_s(10, 5, 5) > C_s(10, 1, 1)$

It is assumed that the member with a high math/computer score can "carry" the respective responsibilities of the modeling competition and that the math/computer quality of other teammates matters little, as illustrated in example 1. We call this property "best score bias." Example 2 illustrates the assumption that math/computer talent spread across multiple team members is worth less then talent concentrated among few team members. But it is also necessary to consider the marginal contribution of team members who do not have the best math/computer score, and example 3 illustrates this.

The cubic mean strikes a balance between the quadratic mean, which fails to model the ability of a single individual to carry the team, and the quartic mean, which discounts the ability of weaker team members to make a positive contribution.

Assumption 2: Team personality scores are modeled by the function

$$F(P_1, P_2, P_3) = \frac{1}{3}\sqrt{3 + 2\cos\frac{(P_1 - P_2)\pi}{15} + 2\cos\frac{(P_2 - P_3)\pi}{15} + 2\cos\frac{(P_3 - P_1)\pi}{15}}$$

where $P_1,\ P_2,\ P_3 \in \{0, 1, \cdots, 29\}$ represent the respective personality score of each member in the team.

We represent all personality scores by a triacontagon (30-gon) centered at the origin in a Cartesian plane. We assume the score of 0 to be at the 1st vertex of the polygon, or the point $(0, 1)$. Subsequent scores rotate in the counterclockwise direction and a score n is represented by the $(n+1)th$ vertex of the 30-gon. It is easy to observe that the coordinate of the nth score is

$$P(n) = (\sin\frac{n\pi}{15}, \cos\frac{n\pi}{15})$$

We therefore define the distance d $(0 \leq d \leq 1)$ between the centroid of the triangle formed by the points $P(P_1), P(P_2), P(P_3)$ and the origin to represent the team score. Note that if $P_1,\ P_2,\ P_3$ form an equilateral triangle, the centroid will be at the origin and the d will be zero, which is at its smallest. This makes sense because when the personalities of the three members are evenly spread apart, they are the least efficient and the team score should be at its minimum. On the other hand, if $P_1,\ P_2,\ P_3$ are all at the same point, the theoretical centroid will also be at that point, and d is at its maximum, 1. This also makes sense as when the three teammates all have the same personalities, the team should be the most efficient and the team score should thus also be the highest.

We will therefore derive the formula of d. Given $P_1,\ P_2,\ P_3$, their coordinates are $(sin\frac{P_1\pi}{15}, cos\frac{P_1\pi}{15}), (sin\frac{P_2\pi}{15}, cos\frac{P_2\pi}{15}), (sin\frac{P_3\pi}{15}, cos\frac{P_3\pi}{15})$. Therefore, the coordinate of the centroid [3] is,

$$\left(\frac{\sin\frac{P_1\pi}{15} + \sin\frac{P_2\pi}{15} + \sin\frac{P_3\pi}{15}}{3}, \frac{\cos\frac{P_1\pi}{15} + \cos\frac{P_2\pi}{15} + \cos\frac{P_3\pi}{15}}{3}\right)$$

Therefore,

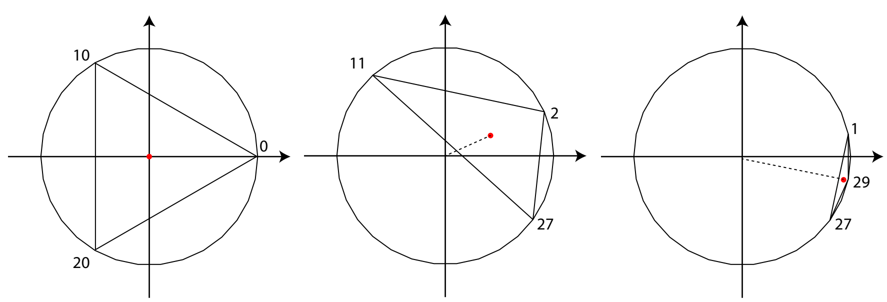
$$d = \sqrt{\left(\frac{\sin\frac{P_1\pi}{15} + \sin\frac{P_2\pi}{15} + \sin\frac{P_3\pi}{15}}{3}\right)^2 + \left(\frac{\cos\frac{P_1\pi}{15} + \cos\frac{P_2\pi}{15} + \cos\frac{P_3\pi}{15}}{3}\right)^2}$$

Through algebraic manipulations, we will arrive at the solution

$$d = \frac{1}{3}\sqrt{3 + 2\cos\frac{(P_1 - P_2)\pi}{15} + 2\cos\frac{(P_2 - P_3)\pi}{15} + 2\cos\frac{(P_3 - P_1)\pi}{15}}$$

The above function represents the phenomenon that when P_1, P_2, P_3 are very close, then the results are better (d is larger), and when the individual scores are distinct, the team's results suffer (d is smaller). Let's consider three teams with personality scores team 1 $(0, 10, 20)$, team 2 $(2, 11, 27)$, and team 3 $(1, 27, 29)$. By intuition, team 1 should have the worst performance as they are evenly separated, and team 3 should have the best performance as they are the closest. Team 2 should fall somewhere in between team 1 and team 3. Our model also predicts a similar outcome as $F(1, 27, 29) > F(2, 11, 27) > F(0, 10, 20)$. This can also be visualized in Figure 1 below. The dotted line is the longest in the third 30-gon whereas the dotted line does not exist or has a length of 0 in the first 30-gon.

Figure 1: Three teams with scores $(0, 10, 10)$, $(2, 11, 27)$, $(1, 27, 29)$ are shown below. The red dot represents the centroid, and the dotted line represents the distance between the centroid and the origin.



Assumption 3: Team writing scores are modeled by the harmonic mean of the individual writing scores.

That is, for a given W_1, W_2, W_3

$$H(W_1, W_2, W_3) = \frac{3}{\frac{1}{W_1} + \frac{1}{W_2} + \frac{1}{W_3}}$$

Where H is the harmonic mean function.

Note: W_i is never zero, so there is no concern for undefined team writing scores.

The harmonic mean model seeks to provide a lower score for teams with a particularly poor writer, while at the same time limiting the upside of having a single outstanding writer. [4] We want to model the need for a balanced team of writers, as modeling competitions require.

For example, the following relations showcase our desired characteristics:

Defining $W(a, b, c)$ to be the team writing score when the individual scores are a, b and c.

1. $W(5, 5, 5) > W(10, 3, 2)$
2. $W(6, 5, 4) > W(7, 5, 3)$
3. $W(10, 10, 1) > W(3, 3, 3)$

In example 1, we show the property of "worst score bias." That is, the worst score in team writing score calculation has disproportionate influence on the outcome.

Example 2 also illustrates worst score bias, but in a less obvious way. Notice that the arithmetic means of (6,5,4) and (7,5,3) are equivalent, but the nature of the harmonic mean causes it to return lower values for inputs with small outliers.

Finally, example 3 shows us that worst score bias cannot overcome large differences in sample mean. Even though 1 is a massive outlier, the two writing score 10 teammates more than make up for the deficiency, at least when compared to a mediocre (3,3,3) team.

3.2 The Predictor Distribution

Assumption 4: The score for the competition lies in $[0, 100]$

Assumption 5: The score for a team has a distribution $\sim N(\mu, \sigma^2)$

We make **Assumption 5** to incorporate a stochastic element into our model. This accounts for potential conflicts amongst team members despite compatible Personality traits and similar skills in Math, Computer Science, and Writing. The normal distribution was chosen primarily due to its easily manipulable mean and variance.

We first tackle the standard deviation (and hence, the variance) of the distribution. Consider the following,

$$P(|N(\mu, \sigma^2) - \mu| \leq k\sigma)$$

Note that when $k = 2$, the above probability evaluates to be approximately 95%., we decide on a standard deviation based on the above fact and the following assumption. [5]

Assumption 6: A team's actual score will have a 95% chance of lying in the range $[\mu - 5, \mu + 5]$

This assumption implies that a team will perform consistently regardless of potential conflicts that may emerge. This assumption is made to strike a balance between the deterministic and stochastic aspects of our model. By reducing the standard deviation of the predicted scores, our algorithms can work solely based off the mean values of the predicted scores. [1]

Therefore for any team, we have the following,

$$2\sigma = 5 \implies \sigma = 2.5$$

With a constant standard deviation (and hence, variance) decided on, we proceed to obtain a mean value for the distribution. Note that given the specified criteria, the team's Math and Computer Science scores are 'additive', while the Writing score is 'multiplicative'. Note that the Personality parameter can modify the score, so we treat its effect as 'multiplicative' given that our model for personality differences produces a value in $[0, 1]$. We further make the following assumption.

Assumption 7: A team's personality score that is closer to 1 will have a net positive effect on their mean score and a team's personality score that is closer to 0 will have a net negative effect on their mean score.

This fairly simple assumption allows a personality score of 0.5 to have a 'neutral' effect on the mean score of a team. Now, to incorporate a team's Math, Computer Science, and Writing score, consider the following,

$$\mu' = \lambda a_W W(P + a_P)(a_M M + a_C C)$$

The above statement incorporates our assumptions about the Probability score of a team. Further, the team's Writing score is used as a multiplier, while the team's Math and Computer Science scores are added together. The constant λ allows us to modify the value to fit a specific range. To determine all the required constants consider the following fringe case, where we have the perfect team of $(10, 10, 10, P)$, $(10, 10, 10, P)$, $(10, 10, 10, P)$. Here (M, C, W, P) is the Math, Computer Science, Writing, and Personality value respectively. Given that we use a distribution, we do not want this perfect team to have a mean score of a 100%, but we want them to have a high chance of obtaining the perfect score. We take this aspect into consideration when we decide on a final mean. We therefore have,

$$\mu = \frac{95}{1.5}\left(P + \frac{1}{2}\right)\frac{W}{10}\left(\frac{M}{20} + \frac{C}{20}\right)$$

This was chosen so that the maximum possible mean score is 95. This simplifies to get,

$$\mu = \frac{19W(2P+1)(M+C)}{120}$$

So the predictor distribution for Team (A, B, C) is modeled by,

$$Pred(A, B, C) \sim N\left(\frac{19W(2P+1)(M+C)}{120}, 2.5^2\right)$$

4 The Algorithms

With the baseline methods for estimation established, it's important to see that our models can actually be applied to problem solving. To test this, we seek to automatize the process of:

1. Finding the optimal member for a team when given two members already.
2. Constructing a set of 100 teams whose mean is greater than if the teams were chosen randomly.

4.1 Algorithm 1: Optimizing the Third

Let Student-A = (M_a, C_a, W_a, P_a) Where P_a is the personality score of student-A, W_a is the writing score of student A, and so on. (Student-A is given to be part of the team)

Let Student-B = (M_b, C_b, W_b, P_b) Where P_b is the personality score of student-B, W_b is the writing score of student B, and so on. (Student-B is given to be part of the team)

Let $M = Max(P_a, P_b)$ and $m = Min(P_a, P_b)$.

Step 1: IF $3 \leq M - m \leq 15$ **RETURN** the list of students with personality scores x that satisfy $m \leq x \leq M$ in sorted order.

IF $3 > M - m$ **RETURN** the list of students with personality scores x that satisfy $m - 1 \leq x \leq M + 2$ in sorted order.

ELSE (If $M - m > 15$) **RETURN** the list of students with personality scores x that *do not* satisfy $M \geq x \geq m$ in sorted order.

Step 2: Apply the mean score function

$$\mu = \frac{19W(2P+1)(M+C)}{120}$$

for Student-A with scores (M_a, C_a, W_a, P_a) , Student-B with scores (M_b, C_b, W_b, P_b), and Student-i with scores (M_i, C_i, W_i, P_i) to produce μ_i for each Student within the output list of step 2.

Step 3: Using the output μ_i

$$\text{Max}(\mu_l, \mu_j, \mu_k \cdots) = \mu_z$$

Where z is the index of the student who gives the team of Student-A and Student-B the best projected overall score.

The following is the pseudo-code for algorithm 1.

Algorithm 1 Optimizing the Third Person

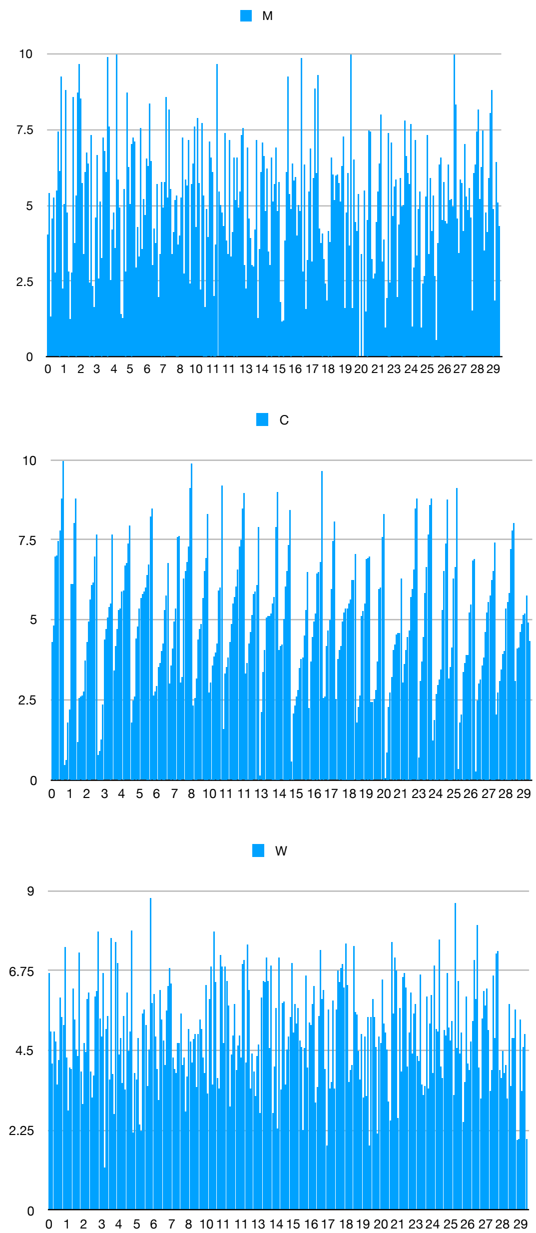
procedure FINDTHIRD
    $P_A \leftarrow$ personality of *Person A*
    $P_B \leftarrow$ personality of *Person B*
    $M \leftarrow max(P_A,\ P_B)$
    $m \leftarrow min(P_A,\ P_B)$
    $diff \leftarrow M$ - m
    if $3 \leq diff \leq 15$ **then**
        $Arr[\ ] \leftarrow x, m \leq x \leq M$
    if $diff < 3$ **then**
        $Arr[\ ] \leftarrow x, m-1 \leq x \leq M+2$
    if $diff > 15$ **then**
        $Arr[\ ] \leftarrow x, \neg(m < x < M)$
loop
    $x \leftarrow Arr[\ ]$
    $Max[\ ] \leftarrow ExpPred(A, B, x)$.
    goto *loop*.
    close;
    $Final \leftarrow max(Max[0],\ Max[1],\ \cdots)$.
    return Final

Assumption 8: It is unreasonable to expect the optimal teammate for a given team of two to exist outside of a certain range of personality.

We conjecture that it's possible to, given a two-person team, provide an personality interval (that is not the whole personality spectrum) for which we have 100% confidence in the interval containing the optimal teammate for the given team of two. This conjecture goes unproven due to time constraints, but our numerical testing leaves the conjecture unchallenged for the intervals we specified in algorithm 1.

Note: This is only true because P is *independent* of M, C, and W. It need not be true in general. For example, it's easy to imagine a world where people of personality 12 through 20 tend to be better at computer programming than those who have other personality scores. Assumption 8 should not apply to situations such as those.

Figure 2: Graphs of M vs P, C vs P, and W vs P show that people's math, CS, or writing scores are independent of their personality scores.



4.2 Algorithm 2: Above Average Team-making

Step 1: Sort the list of the 300 students by their personality score in increasing order

Step 2: Select the first two people in the sorted list, N. Name them Student-A and Student-B.

Step 3: Invoke **Algorithm 1** with Student-A and Student-B as inputs. **Algorithm 1** will return an index z. The student of index z will be teamed up with Student-A and Student-B.

Step 4: Remove Student-A, Student-B, and Student-Z from the sorted list, N, and store them in the first entry of a new list, say T. N should now have a size of 297, and T should have a size of 1.

Step 5: Repeat **Step 2** with list N until the size of N becomes 0 and the size of T becomes 100.

The following is the pseudo-code for algorithm 2.

Algorithm 2 Above Average Team-making

procedure OptTeams
  Sort Arr[Students] by Personality in $0 \leq x \leq 30$
  $A_1 \leftarrow Arr[0]$
  $A_2 \leftarrow Arr[1]$
  $A_3 \leftarrow FindThird(A_1,\ A_2)$
  $Pred[\,] \leftarrow ExpPred(A_1,\ A_2,\ A_3)$
  Remove A_1, A_2, A_3 *from Arr[Students]*
  OptTeams(Arr[Students], Pred[Scores])

Algorithm 2 attempts to maintain simplicity by recycling the functionality of algorithm 1. By nature, the first two teammates that algorithm two selects will be of very similar (if not identical) personality. Furthermore, the third teammate will be chosen in a near optimal matter. Thus, it is clear that algorithm 2 will be at least better than random choice. We analyze the level of success for algorithm 1 and 2 in the next section.

5 Evaluation and Results

Now we share the outcomes of numerous tests and simulations of all sorts: ranging from easily interpretable raw data models, trends of our models with random team assignment, tests for algorithm 1 and algorithm 2, and analysis of their effectiveness.

5.1 Algorithm 1

Since we want to support the claim that algorithm 1 picks an optimal third teammate when given two teammates, we compare its results to a procedure which we know for certain will produce the optimal teammate, the "brute force" algorithm[2]. In the the figure below, we illustrate the result on our algorithm operating on an array of two-person teams.

Figure 3:

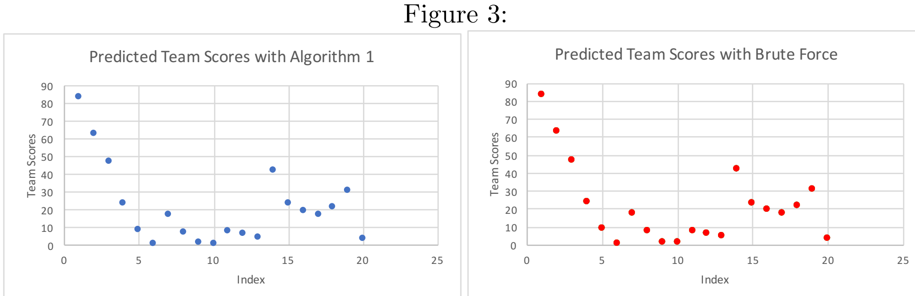


Figure 4:

	Person 1 (P, M, C, W)	Person 2 (P, M, C, W)	Predicted Mean (Alg 1)	Predicted Mean (Brute Force)
1	Person(1 | 10.0 | 10.0 | 10.0)	Person(1 | 10.0 | 10.0 | 10.0)	83.37811679	83.37811679
2	Person(1 | 10.0 | 10.0 | 10.0)	Person(14 | 10.0 | 10.0 | 10.0)	62.78611356	62.78611356
3	Person(3 | 9.0 | 1.0 | 8.0)	Person(3 | 1.0 | 9.0 | 8.0)	47.05852306	47.05852306
4	Person(10 | 7.0 | 7.0 | 5.0)	Person(20 | 7.0 | 7.0 | 5.0)	23.8812643	23.8812643
5	Person(0 | 5.0 | 9.0 | 10.0)	Person(0 | 9.0 | 5.0 | 1.0)	8.912830358	8.912830358
6	Person(10 | 1.0 | 1.0 | 1.0)	Person(25 | 1.0 | 1.0 | 1.0)	0.615222863	0.615222863
7	Person(16 | 3.0 | 1.0 | 5.0)	Person(16 | 9.0 | 2.0 | 4.0)	17.31874314	17.31874314
8	Person(29 | 5.0 | 1.0 | 2.0)	Person(1 | 1.0 | 4.0 | 3.0)	7.42436519	7.42436519
9	Person(26 | 3.0 | 3.0 | 3.0)	Person(11 | 1.0 | 1.0 | 1.0)	1.685010133	1.685010133
10	Person(6 | 1.0 | 1.0 | 1.0)	Person(5 | 1.0 | 2.0 | 1.0)	1.157509077	1.157509077
11	Person(11 | 7.0 | 4.0 | 1.0)	Person(7 | 1.0 | 5.0 | 4.0)	7.702952446	7.702952446
12	Person(8 | 2.0 | 5.0 | 2.0)	Person(2 | 2.0 | 7.0 | 2.0)	6.521637939	6.521637939
13	Person(5 | 3.0 | 7.0 | 5.0)	Person(13 | 4.0 | 4.0 | 1.0)	4.524623798	4.524623798
14	Person(17 | 5.0 | 9.0 | 7.0)	Person(10 | 5.0 | 10.0 | 8.0)	42.052691	42.052691
15	Person(2 | 7.0 | 2.0 | 4.0)	Person(5 | 7.0 | 2.0 | 9.0)	23.35803203	23.35803203
16	Person(29 | 6.0 | 4.0 | 3.0)	Person(21 | 9.0 | 1.0 | 7.0)	19.45753478	19.45753478
17	Person(3 | 1.0 | 5.0 | 5.0)	Person(23 | 2.0 | 5.0 | 10.0)	17.35526638	17.35526638
18	Person(13 | 9.0 | 6.0 | 6.0)	Person(1 | 4.0 | 6.0 | 5.0)	21.40499025	21.40499025
19	Person(15 | 10.0 | 7.0 | 7.0)	Person(7 | 7.0 | 3.0 | 6.0)	30.5980066	30.5980066
20	Person(21 | 4.0 | 3.0 | 2.0)	Person(17 | 6.0 | 1.0 | 1.0)	3.677247138	3.677247138

[2]This is done by taking the maximum of the outputs of the mean function applied with each of the other 298 students

Figure 4 compares the outputs of the brute force algorithm and algorithm 1 given an array of examples. The leftmost column of figure 4 serves as an index for the data points so that they can be easily identified in figure 3.

Notice that algorithm 1 produces a team score within machine precision (15 decimal places) of the brute force algorithm. It is, however, unclear if the algorithms are actually selecting the same *individual*. Thus, we cannot say for certain that algorithm 1's result is strictly equal to the brute force algorithm's, but we can say without doubt that algorithm 1 is extraordinarily accurate in its optimization.

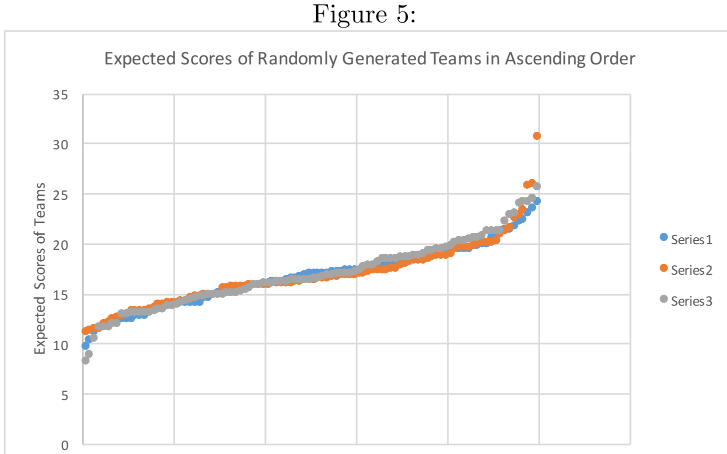
Additionally, if the 20 sample inputs outlined in Figure 4 are examined closely, one can observe that extreme cases towards the higher end (For instance, group 2), extreme cases towards the lower end (For instance, group 9), and average cases were all included (For instance, group 17). Covering a wide range of sample cases better attests to the validity of our model, as it is proven to function perfectly under all types of situations.

Based on this, algorithm 1 searches for an optimal third member in an informed way. In most cases, we only check a small subset of the data, yet we still achieve accuracy within 15 decimal places of the brute force algorithm[3].

5.2 Algorithm 2

For this algorithm, we want to improve the mean of our overall team scores. Before we do so, it is important to consider the distribution of overall scores in randomly assigned teams.

Figure 5:



[3]In a perfect world, we'd like to make sure that our third member algorithm always picked out *exactly* the best member.

In figure 5, each color represents a different randomized set of teams. We see that, when we assign teams randomly,

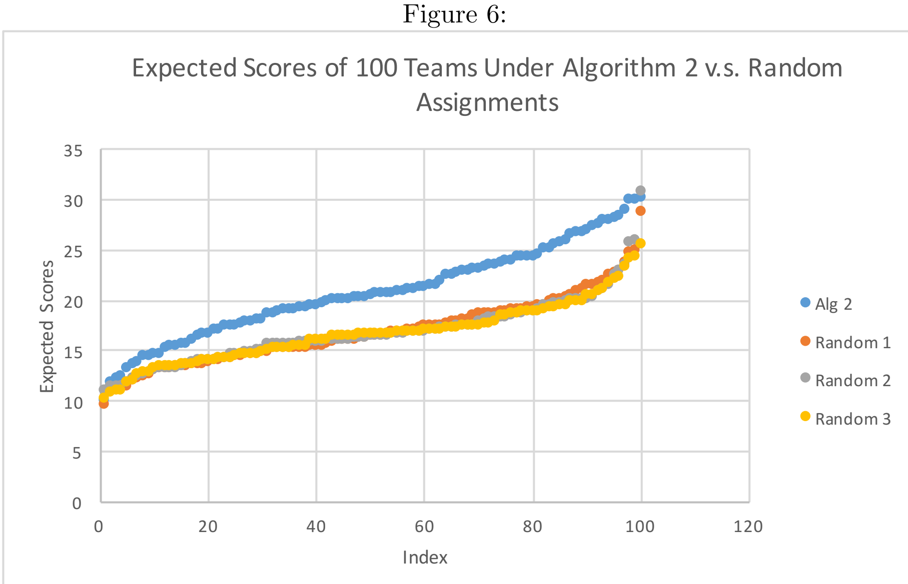
$$\mathbf{E}[X] \approx 16.56$$

Where X is the score of a random team.

We see that our mean function is very harsh on non-coordinated teams. That is, teams suffer more from a certain degree of dissimilarity than teams are helped by the same degree of coercion. When considering the overall contest score range (0,100) a mean of 16.56 is quite low, and is, in fact, a good model of the importance of coercion in a competition team.

Now, using algorithm 2, we construct 100 teams of 3 from the 300 participant sample. The mean scores for teams constructed using algorithm 2 and using random assignments are compared (in increasing order) in figure 6.

Figure 6:



The expected scores of the teams constructed using algorithm 2 have the property,

$$\mathbf{E}[Y] \approx 20.68$$

Where Y is the score of a team constructed by algorithm 2.

An expected value increase of 4.12 is not exceptional, but is quite large when considered in ratio to the standard deviation of our random team assignment algorithm mean.

$$\frac{4.12}{\sigma} = \frac{4.12}{2.5} = 1.648$$

In addition, it can be observed that the expected score line with Algorithm 2 (blue) lies completely above the other three lines. This property also reinforces the strength of the algorithm, as it ensures that the mean team score of every one of the 100 teams, ordered from smallest to largest, is equal to or greater than the result of the corresponding indexed team in a random assignment.

Our algorithm certainly accomplishes the primary goal, but it's degree of success is under question. There appears to be great opportunity for improvement, though we cannot make a justifiable conjecture on what an optimal solution might be.

6 Conclusion

6.1 Strengths and Weaknesses

Due to the time constraint and some overly simplified assumptions, our model does possess several weaknesses in certain aspects.

Strength: Our methods compute the team scores in M, C, and W model to surprising accuracy and with remarkable simplicity, and they also include the properties of best score bias and worst score bias. For Mathematics and computer programming, we capture the fact that a single high score is sufficient to carry the team, while not discounting the contribution of other team members. For W, we emphasize the importance for every team member to be competent in writing , without undercutting the value of big individual writing scores.

Weakness 1: In our **Assumption 5**, we assumed the variance of the normal distribution of team results to be a constant, 2.5. We made this assumption because with the limited time, we wanted to focus more on how each variable attribute will affect the mean value of the normal model instead of the variance. The variance was thus assumed to be constant to slightly reduce the complexity. However, in reality, the variance of the teams' performances will not be a constant value. They can in fact be influenced by the variable attributes. For instance, consider two teams with individual math scores $(10, 1, 1)$ and $(7, 7, 7)$. By **Assumption 1**, the team math scores for these two teams will be the same. They have a similar impact on the mean and do not impact the variance of the distribution of the team results. However, by intuition, team 1 with scores $(10, 1, 1)$ should have a higher variance. If the person with the math score 10 is sick or in an emergency, the this team will completely crash. In contrast, even if one person is not performing as expected in team 2, the other two people can still perform relatively well. If more time is permitted, the phenomenon stated above should be accounted when setting up our variance.

Weakness 2: The four functions that we derive to determine the team scores in individual sections are all deterministic. In other words, given the score of each team member, we predict that the team score will be a specific value. However, this model is simplified, and in a more realistic situation, the team score should contain a stochastic element. To illustrate this weakness, consider team 1 and 2 with personality score $(0, 10, 20)$ and $(1, 3, 5)$. By our **Assumption 2**, team 2 has a team personality score that is strictly larger than that of team 1's. However, in reality, there might be occasions, although very rare, where a team with very distinct personalities will perform better than a team with similar personalities. Our deterministic functions failed to consider such a possibility.

Weakness 3: In our algorithm, we have filtered out a significant portion of the sample by setting the personality scores of the two given team members as the bounds. However, there is a possibility that such a bound will filter out the absolute best candidate for the third person. We believe that there exists a bound which is close to the personality scores of the two given members and which will for sure not exclude the absolute best candidate for the third person. Due to the time limit, however, we cannot provide a rigorous proof for the existence of such a bound.

6.2 Further Improvements

There are several other properties that would be expected concerning the impact of the relationships between M, C, P, and W on the final score. For example, we might want to implement a property of our model were having large discrepancies between a team's mathematics and computer programming score has a negative effect on the overall score. The intuition is that much of the math talent would "go to waste" if there was nobody with the computer knowledge to implement the mathematician's ideas.

It also seems reasonable that, in a math modeling competition, M might have a greater influence on the overall score than C. That is, deficiency in mathematics is probably much more costly than an inability to program.

Finally, if time had permitted, we would have preferred to implement an "ego effect." It seems logical that if a team had two members who were very good at the same thing, two members with scores of 9 in computer programming for example, there would be many arguments on how things should be done. Whereas, if there was a single star computer programmer, the team would be more likely to defer to him all the responsibilities of programming, without objection.

7 Bibliography

References

[1] George Casella and Roger L Berger. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002.

[2] Erhan Cinlar. *Introduction to stochastic processes.* Courier Corporation, 2013.

[3] David Kay. *College geometry: A discovery approach.* Pearson Higher Ed, 2011.

[4] Walter Rudin et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1964.

[5] Dennis Wackerly, William Mendenhall, and Richard Scheaffer. *Mathematical statistics with applications.* Nelson Education, 2007.