How to Compute an Overall Grade for a Class that Has Both Theoretical and Practical Components?

Vignesh Ponraj and Vladik Kreinovich

Abstract In a class where the students learn both theoretical and practical skills, at the end of the class, we need to combine the overall grades for these two skills into a single final grade. Usually, instructors use weighted average of these grades – e.g., a simple arithmetic average. However, this approach has a problem. For example, if a student got 100 for the theoretical part of a computing class and 60 for the practical part, his/her average grade 80 would provide a false impression to future employers that this student is good in programming – while in reality, his/her programming skills are below satisfactory. To avoid this problem, some instructors use the smallest of the two grades. However, this can also be misleading. For example, the above 100-60 student would get a not-good overall grade of 60, so future employers will consider hum/her useless for computing-related tasks – while in reality, he/she can help with theoretical aspects of the computing tasks. Recently, a new heuristic formula was proposed for such a combination that seems to avoid the limitations of the weighted average and minimum approaches. In many situations when we value both components equally, this formula leads to a more adequate description of the student skills. In this paper, we provide a theoretical explanation for this formula's success. We also use the ideas behind this theoretical explanation to show how this formula can be naturally extended to situations when one of the class components is valued more – and also to situations when the class has more than two natural components.

1 Formulation of the problem

Situation: a brief introduction. In many classes, we teach both theoretical and practical skills. For example, after taking the Introduction to Computer Science

Vignesh Ponraj and Vladik Kreinovich
Department of Computer Science, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, e-mail: vigneshponraj97@gmail.com, vladik@utep.edu

class, the students must be able to write simple programs – and also to understand and explain, on examples, how the basic constructions (for-loops, while-loops, if-then statements) work. Some assignments and test questions test their ability to code, other assignments and test questions tests their ability to explain and utilize the main concepts.

To each assignment and for each test, we assign a numerical grade. In particular, in the US, usually, the following numerical grades are assigned:

- excellent work (denoted by A) means between 90/100 and 100/100;
- good work (denoted by B) means between 80/100 and 90/100;
- satisfactory work (denoted by C) means between 70/100 and 80/100;
- below satisfactory (denoted by D) means between 60/100 and 70/100; and
- unsatisfactory (failing grade, denoted by F) means below 60/100.

The grades for different assignments and different questions are communicated to the student, but they are not communicated to anyone else. The only grade that goes into the student's transcript is the overall grade for the class.

The question that we analyze in this paper is: how to best combine the grades for different assignments and tests into a grade for the class?

How the class grade is usually computed. Usually, the grade for the class is a weighted average of the grades for all assignments and test questions. Typically, this is done in two stages:

- first, we combine the grades for all practical assignment into a single "practical" grade g_p, and we combine the grades for all theoretical assignments and question into a single "theoretical" grade g_t;
- then, we combine g_p and g_t into a single grade for the class:

$$g = w_p \cdot g_p + (1 - w_p) \cdot g_t. \tag{1}$$

In many cases, we just take the arithmetic average of the two grades:

$$g = 0.5 \cdot g_p + 0.5 \cdot g_t. \tag{2}$$

Why is this usual scheme not perfect? To explain why the usual scheme is not perfect, let us give a simple example. Suppose that a student has a perfect 100 on all theoretical questions on all the tests, but on all programming assignments, only gets 60, below satisfactory. So, if we consider them equally, then, according to the formula (2), we get 80.

As a result, not only the student passes the class, he passes it with a good grade. If this student is being hired by a company, the company sees that on the Introduction to Computer Science course, that student got B – so they think that he will be good for programming jobs. But actually this student's programming skills are lacking. When the students performs not so well on these tasks, the company may believe that the student's university is inflating the grades – and make a decision never again to hire students from this university: while in reality, many other graduates from this same university program very well.

A known way to avoid such consequences: description and limitations. A solution to this problem that some instructors use is to take the minimum of the two grades. So, if a student scored 100 for one part and 60 for another part, this student gets an overall grade of 60 for this class – and thus, fails the class. If this is a CS student, then this student doesn't progress further – until he/she retakes this class and gets a passing grade.

If this is a student of some other major, a hiring company will look at his transcript, see a failing grade in the Introduction to Computer Science class, and company will think that this student is useless in computer-related tasks. However, in reality, for some tasks – namely, for more theoretical tasks – the student can be useful. This student may not be able to write good code, but he/she can understand others' codes and thus help. So this solution has its own limitations.

A recent proposal. To resolve this situation, Dr. Nigel Ward from our university's Computer Science Department proposed the following formula for the overall grade:

$$g = \min(\alpha \cdot g_p + (1-\alpha) \cdot g_t, \alpha \cdot g_t + (1-\alpha) \cdot g_p). \tag{3}$$

In other words, we pick some value $\alpha > 0.5$, and then take the minimum of the two weighted averages:

- the one in which the theoretical part get the weight α and
- the one in which the practical part gets the weight α.

Dr. Ward has used this formula in his classes, and it seems to work.

Examples.

- When $\alpha = 0.5$, the formula (3) becomes the average.
- When $\alpha = 1$, the formula (3) becomes the minimum.
- When α is in between 0.5 and 1, then the value computed by the formula (3) is in between the average and the minimum. For example, in the above case $g_p = 60$ and $g_t = 100$, for $\alpha = 0.8$, we have $1 - \alpha = 0.2$, so:

$$g = \min(0.8 \cdot 60 + 0.2 \cdot 100, 0.8 \cdot 100 + 0.2 \cdot 60) =$$

$$\min(48 + 20, 80 + 12) = \min(68, 92) = 68,$$

  which is indeed between the minimum (which is 60) and the average (which is 80).

Natural questions. The first natural question is: Why this particular formula and not some other one?

The second question is related to the fact that this formula views both theoretical and practical parts equally. How can we modify this formula if we would like to put more emphasis on the practical part?

What we do in this paper. In this paper, we provide answers to both questions.

2 Analysis of the problem and the resulting answers

To solve our problem, it is reasonable to use decision theory. Like in many other situations, we need to make a decision – namely, we need to decide what grade to give to the student. So, a natural idea is to use decision theory, a theory specifically designed to help us make decisions; see, e.g., [1, 2, 3, 4, 5, 6, 7].

How decisions are made according to decision theory: a brief reminder. In decision theory, decision maker's preferences are described by a value called *utility*: the larger this value, the better the alternative.

This numerical value is defined in such a way that if an action a leads to different outcomes $o_1, \ldots, o_n$ with probabilities $p_1, \ldots, p_n$, then the utility $u(a)$ of the action is equal to the expected values of the utility $u(o_i)$ of possible outcomes:

$$u(a) = p_1 \cdot u(o_1) + \ldots + p_n \cdot u(o_n). \tag{4}$$

Let us apply this technique to our problem. We want to come up with an overall grade that reflects the student's utility to his/her future employers. To estimate this utility, we need to take into account that, in general, possible tasks that the student can do for a company can be of three different types:

- the company may need to use the person's practical knowledge, e.g., to code something; we will denote this situation by p;
- the company may need to use the person's theoretical knowledge, e.g., to analyze someone else's code without writing a new code; we will denote this situation by t;
- the company may need to use both the person's practical knowledge and his/her theoretical knowledge; we will denote this situation by pt.

For simplicity, let us assume that for each part – theoretical and practical – the utility is simply proportional to the grade.

Then, in the first task, the student's utility is proportional to this student's practical grade. In the second task, the student's utility is proportional to this student's theoretical grade. In the third task, when both skills are required, the student's utility is proportional to the minimum of the two grades.

So, if we denote by p_i the probability of the i-th type of task, so that $p_p + p_t + p_{pt} = 1$, we get the following formula for the student's utility:

$$u = p_p \cdot g_p + p_t \cdot g_t + p_{pt} \cdot \min(g_p, g_t), \tag{5}$$

where $p_{pt} = 1 - p_p - p_t$.

This decision-theory-based expression generalized the empirical formula. Let us show that the theory-based formula (5) generalizes the empirical formula (3). Thus, we answer both questions formulated in Section 1:

- we provide a theoretical justification for the empirical formula, and

- we provide a more general theoretically justified expression.

Indeed, it is well known that if $a \le b$, then $a+c \le b+c$ for all c, and that $c \cdot a \le c \cdot b$ for all $c \ge 0$. This means that

$$\min(a+c, b+c) = \min(a,b) + c \tag{6}$$

for all c and that

$$\min(c \cdot a, c \cdot b) = c \cdot \min(a,b) \tag{7}$$

for all $c \ge 0$. In particular, for $a = g + p$, $b = g_t$, and $c = p_{pt}$, the formula (7) leads to

$$p_{pt} \cdot \min(g+p, g_t) = \min(p_{pt} \cdot g_p, p_{pt} \cdot g_t),$$

so the formula (5) takes the form:

$$u = p_p \cdot g_p + p_t \cdot g_t + \min(p_{pt} \cdot g_p, p_{pt} \cdot g_t). \tag{8}$$

Now, for $a = p_{pt} \cdot g_p$, $b = p_{pt} \cdot g_p$, and $c = p_p \cdot g_p + p_t \cdot g_t$, the formula (6) leads to

$$u = \min(p_p \cdot g_p + p_t \cdot g_t + p_{pt} \cdot g_p, p_p \cdot g_p + p_t \cdot g_t + p_{pt} \cdot g_t), \tag{9}$$

i.e., to

$$u = \min((p_p + p_{pt}) \cdot g_p + p_t \cdot g_t, p_p \cdot g_p + (p_t + p_{gt}) \cdot g_t), \tag{10}$$

which can be also described as

$$u = \min(\alpha_p \cdot g_p + (1-\alpha_p) \cdot g_t, \alpha_t \cdot g_t + (1-\alpha_t) \cdot g_p), \tag{11}$$

where we denote $\alpha_p \stackrel{\text{def}}{=} p_p + p_{pt}$ and $\alpha_t \stackrel{\text{def}}{=} p_t + p_{pt}$ So, when $p_p = p_t$, and thus, $\alpha_p = \alpha_t$, i.e., when we take both parts equally, then we get exactly the empirical formula (3) with $\alpha = \alpha_p = \alpha_t$.

Conclusion of this section. So, not only we explained the empirically successful formula (3) we also have a more general formula that we can use if we want to give more weight to one of the parts.

This derivation also explains which value α – or, correspondingly, which values of α_p and α_t – we should use: this can determined based on the probability of different types of tasks at the jobs for which we prepare our students.

3 An interesting observation

What we do in this section. In this section, we show that the same formula (5), or, equivalently, (11) appears if we start with the cases when the answer is very clear, and use linear interpolation – the simplest interpolation technique – to extend this answer to all other combinations of the grades.

What are the cases for which the answer is clear? We want to find a formula $f(a_1,a_2)$ that calculates the overall grade for the class based on the grades a_1 and a_2 for the two knowledge components.

If the students gets exactly the same grade a for both components, then clearly, this should be this student's overall grade. So, we should have $f(a,a)=a$.

We also need to know the student's grade in two extreme situations:

- when the student is perfect in theory but does not much about programming, and
- when the student in perfect in coding and does not know much about theory.

Let us denote the corresponding overall grades by $v_1=f(1,0)$ and $v_2=f(0,1)$.

The general formula for linear interpolation. If for some unknown function $F(x)$, we know its values $y_1=F(x_1)$ and $y_2=F(x_2)$ for two inputs x_1 and x_2, then linear interpolation means that for all other inputs x, we take

$$F(x)=\frac{x_2-x}{x_2-x_1}\cdot F(x_1)+\frac{x-x_1}{x_2-x_1}\cdot F(x_2). \tag{12}$$

Let us use linear interpolation. Let us show that based on this data, linear interpolation indeed leads to the same formula as the utility approach. For this purpose, we consider two possible cases: when $a_1\ge a_2$ and when $a_1<a_2$.

In this first case, when $a_1>a_2$, we first apply linear interpolation by the first variable to the known values $f(0,0)=0$ and $f(1,0)=v_1$ to find the value $f(a_1,0)$. In this case, $F(x)=f(x,0)$, $x_1=0$, $x_2=1$, $y_1=F(x_1)=F(0)=f(0,0)=0$, and $y_2=F(x_2)=F(1)=f(1,0)=v_1$, and $x=a_1$. Thus, the formula (12) leads to:

$$f(a_1,0)=F(a_1)=\frac{1-a_1}{1-0}\cdot 0+\frac{a_1-0}{1-0}\cdot v_1=a_1\cdot v_1. \tag{13}$$

Then, we apply linear interpolation by the second variable to the values $f(a_1,0)=a_1\cdot v_1$ and $f(a_1,a_1)=a_1$ to find the desired value $f(a_1,a_2)$. In this case, $F(x)=f(a_1,x)$, $x_1=0$, $x_2=a_1$, $y_1=F(x_1)=F(0)=f(a_1,0)=a_1\cdot v_1$, $y_2=F(x_2)=F(a_1)=f(a_1,a_1)=a_1$, and $x=a_2$. Thus, the formula (12) leads to:

$$f(a_1,a_2)=F(a_2)=\frac{a_1-a_2}{a_1-0}\cdot a_1\cdot v_1+\frac{a_2-0}{a_1-0}\cdot a_1=$$
$$(a_1-a_2)\cdot v_1+b=v_1\cdot a_1+(1-v_1)\cdot a_2. \tag{14}$$

This is exactly the formula (11) for the case when $a_1=g_p\ge g_t=a_2$.

In this second case, when $a_1<a_2$, we first apply linear interpolation by the second variable to the known values $f(0,0)=0$ and $f(0,1)=v_2$ to find the value $f(0,a_2)$. In this case, $F(x)=f(0,x)$, $x_1=0$, $x_2=1$, $y_1=F(x_1)=F(0)=f(0,0)=0$, and $y_2=F(x_2)=F(1)=f(0,1)=v_2$, and $x=a_2$. Thus, the formula (12) leads to:

$$f(0,a_2)=F(a_2)=\frac{1-a_2}{1-0}\cdot 0+\frac{a_2-0}{1-0}\cdot v_2=a_2\cdot v_2. \tag{15}$$

Then, we apply linear interpolation by the first variable to the values $f(0,a_2) = a_2 \cdot v_2$ and $f(a_2,a_2) = a_2$ to find the desired value $f(a_1,a_2)$. In this case, $F(x) = f(x,a_2)$, $x_1 = 0$, $x_2 = a_2$, $y_1 = F(x_1) = F(0) = f(0,a_2) = a_2 \cdot v_2$, $y_2 = F(x_2) = F(a_2) = f(a_2,a_2) = a_2$, and $x = a_1$. Thus, the formula (12) leads to:

$$f(a_1,a_2) = F(a_1) = \frac{a_2 - a_1}{a_2 - 0} \cdot a_2 \cdot v_2 + \frac{a_1 - 0}{a_2 - 0} \cdot a_2 =$$

$$(a_2 - a_1) \cdot v_2 + a_2 = v_2 \cdot a_2 + (1 - v_2) \cdot a_2. \tag{16}$$

This is exactly the formula (16) for the case when $g_p < g_t$. the slide).

4 What if we have three or more components?

This was all about the case when we only have two components to combine. But what if we have three or more components?

In this case, you may have different utilities depending on which skills you have. For example, for the 3 components, you have the following formula:

$$\begin{gathered} u = p_1 \cdot g_1 + p_2 \cdot g_2 + p_3 \cdot g_3 + \\ p_{12} \cdot \min(g_1,g_2) + p_{13} \cdot \min(g_1,g_3) + p_{23} \cdot \min(g_2,g_3) + \\ p_{123} \cdot \min(g_1,g_2,g_3), \end{gathered} \tag{17}$$

where for each set I, p_I denotes the probability that we need all the skills from the set I and no other skills. For example, p_1 that we only need Skill 1, and p_{12} is the probability of a task that requires Skills 1 and 2.

5 Acknowledgments

This work was supported by the AT&T Fellowship in Information Technology, by the Institute for Risk and Reliability, Leibniz Universitaet Hannover, Germany, by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Focus Program SPP 100+ 2388, Grant Nr. 501624329, by the European Union under the project ROBOPROX (No. CZ.02.01.01/00/22 008/0004590), by the Center of Excellence in Econometrics, Faculty of Economics, Chiang Mai University, Thailand, by the Ho Chi Minh City University of Banking, Vietnam, and by Thang Long University, Hanoi, Vietnam.

The authors are thankful to all the participants of the NMSU/UTEP Workshop on Mathematics, Computer Science, and Computational Science, Las Cruces, New Mexico, April 11, 2026.

References

1. P. C. Fishburn, *Utility Theory for Decision Making*, John Wiley & Sons Inc., New York, 1969.
2. P. C. Fishburn, *Nonlinear Preference and Utility Theory*, The John Hopkins Press, Baltimore, Maryland, 1988.
3. V. Kreinovich, "Decision making under interval uncertainty (and beyond)", In: P. Guo and W. Pedrycz (eds.), *Human-Centric Decision-Making Models for Social Sciences*, Springer Verlag, 2014, pp. 163–193.
4. R. D. Luce and R. Raiffa, *Games and Decisions: Introduction and Critical Survey*, Dover, New York, 1989.
5. H. T. Nguyen, O. Kosheleva, and V. Kreinovich, "Decision making beyond Arrow's 'impossibility theorem', with the analysis of effects of collusion and mutual attraction", *International Journal of Intelligent Systems*, 2009, Vol. 24, No. 1, pp. 27–47.
6. H. T. Nguyen, V. Kreinovich, B. Wu, and G. Xiang, *Computing Statistics under Interval and Fuzzy Uncertainty*, Springer Verlag, Berlin, Heidelberg, 2012.
7. H. Raiffa, *Decision Analysis*, McGraw-Hill, Columbus, Ohio, 1997.