11-1-2021

Why Moments (and Generalized Moments) Are Used in Statistics and Why Expected Utility Is Used in Decision Making: A Possible Explanation

R. Noah Padilla
The University of Texas at El Paso, rnpadilla2@miners.utep.edu

Vladik Kreinovich
The University of Texas at El Paso, vladik@utep.edu

Comments:
Technical Report: UTEP-CS-21-87

Why Moments (and Generalized Moments) Are Used in Statistics and Why Expected Utility Is Used in Decision Making: A Possible Explanation

R. Noah Padilla and Vladik Kreinovich

Abstract Among the most efficient characteristics of a probability distribution are its moments and, more generally, generalized moments. One of the most adequate numerical characteristics describing human behavior is expected utility. In both cases, the corresponding characteristic is the sum of results of applying appropriate nonlinear functions applied to individual inputs. In this paper, we provide a possible theoretical explanation of why such functions are efficient.

1 Formulation of the Problem

In this paper, we provide a new explanation of two seemingly unrelated phenomena:

- that moments (and, more generally, generalized moments) are effectively used in statistics; see, e.g., [8], and
- that expected utility is effectively used in decision making; see, e.g., [1, 2, 3, 4, 5, 6, 7].

Before we provide the corresponding explanations, let us first briefly describe these two phenomena.

Moments and generalized moments: a brief reminder. One of the most frequent ways to characterize a random variable x is to use moments – i.e., expected values $E[x^k]$ of some integer power of this variable – and, more generally, generalized moments, i.e., expected values $E[f(x)]$ of some function of the random variable.

For each random quantity q, its expected value is equal to the limit of its average observations $q_1, \ldots, q_n, \ldots$:

R. Noah Padilla and Vladik Kreinovich
Department of Computer Science, University of Texas at El Paso
500 W. University, El Paso, TX 79968, USA
e-mail: rnpadilla2@miners.utep.edu, vladik@utep.edu

$$E[q] = \lim_{n\to\infty} \frac{q_1 + \ldots + q_n}{n}.$$

By definition of the limit, this means that when n becomes larger and larger, the average

$$\frac{q_1 + \ldots + q_n}{n}$$

gets closer and closer to the expected value. Thus, a reasonable way to estimate the mean based on the observations q_i is to take the arithmetic average of all the observed values:

$$E[q] \approx \frac{q_1 + \ldots + q_n}{n}.$$

In particular, to estimate the value $E[f(x)]$ of the generalized moment (or, in particular, of a usual moment corresponding to $f(x) = x^k$) based on the observations $x_1, \ldots, x_n$, it is reasonable to use the corresponding arithmetic average

$$E[f(x)] \approx \frac{f(x_1) + \ldots + f(x_n)}{n}. \tag{1}$$

Alternative formulas for moments and generalized moments. In some cases, we have limited number of values $v_1, \ldots, v_k$ ($k \ll n$) that the variables x_i can take. In this case, each term $f(x_i)$ in the sum

$$s \stackrel{\text{def}}{=} f(x_1) + \ldots + f(x_n) \tag{2}$$

is equal to one of the k values $f(v_j)$, $1 \le j \le k$. In such cases, we can simplify the formula (2) by grouping together terms equal to $f(v_1)$, terms equal to $f(v_2)$, etc. Then, we get

$$s = f(v_1) + \ldots + f(v_1)\ (n_1 \text{ times}) + \ldots + f(v_k) + \ldots + f(v_k)\ (n_k \text{ times}),$$

where n_j denotes the number of terms $f(x_i)$ which are equal to $f(v_k)$, or, equivalently,

$$s = f(x_1) + \ldots + f(x_n) = n_1 \cdot f(v_1) + \ldots + n_k \cdot f(v_k).$$

Substituting this expression into the formula (1), we conclude that

$$E[f(x)] \approx \frac{n_1}{n} \cdot f(v_1) + \ldots + \frac{n_k}{n} \cdot f(v_k). \tag{3}$$

Here, the ratio $\dfrac{n_j}{n}$ is the frequency with which the value v_j appears in the observations, i.e., in effect, the probability p_j of this value – to be more precise, the probability is defined as the limit of such a frequency, but since we are considering large n, probability and frequency are approximately the same. Thus, the formula (3) takes the form

$$E[f(x)] \approx p_1 \cdot f(v_1) + \ldots + p_k \cdot f(v_k). \tag{4}$$

Expected utility: a brief reminder. It is known – see above references – that a rational person, when making a decision, should maximize the value of a special expression known as *expected utility*

$$u \stackrel{\text{def}}{=} p_1 \cdot u(v_1) + \ldots + p_k \cdot u(v_k), \tag{5}$$

where:

- $v_1, \ldots, v_k$ are possible consequences of the selected action,
- p_j is the (subjective) probability of getting an alternative v_j, and
- $u(v_j)$ is a number – called *utility* – that characterize the value of the alternative v_j to the decision maker.

Comment. The main use of expected utility is to decide which alternative is better, i.e., which decision we should make. From this viewpoint, what is important are not the numerical values (5) themselves, but which values are larger and which are smaller. From this viewpoint, instead of the values u, we could use the values $g(u)$ for any increasing function $g(u)$ – since for an increasing function $u < u'$ if and only if $g(u) < g(u')$.

Is there a common explanation for these two formulas? There exist explanations for both formulas (4) and (5), explanations based on different ideas; see, e.g., the above references. However, the fact that the expressions (4) and (5) are very similar – in both cases, we have a linear combination of the values of some function ($f(v)$ in the first case, $u(v)$ in the second case) applied to different values $v_1, \ldots, v_k$ – made us think that there also be a joint explanation for these two seemingly unrelated formulas. In this paper, we provide a possible common explanation.

2 Main Ideas Behind Our Explanation

In many practical problem, computation time is a big issue. Nowadays, we get a lot of data, and we have a lot of computational ability. However, still, computation time remains a big issue. For example, with numerous weather sensors almost everywhere, we get a lot of data that enables us to predict tomorrow's weather reasonably well – but because of the huge amount of data and, as a result, a huge amount of computations, the only way to predict weather is to use high-performance computers, where a large number of processors are working in parallel, and even on such computers, weather prediction takes hours (and became possible only after special time-saving algorithms were implemented).

In many other problems we still cannot perform computations in desired time. For example, in principle, it is possible to predict somewhat accurately in what direction a potentially deadly tornado will go in the next 15 minutes – but the resulting

computations so far require much longer than 15 minutes and are, therefore, practically useless. From this viewpoint, it is desirable to come up with computations that can be performed as fast as possible.

Which computations are the fastest? Of course, to make computations faster, we need to parallelize computations as much as possible. On a parallel computer, first, all the processors perform one computation step, then they all perform another step, etc. To minimize the overall computation time:

- we need to minimize the number of steps, and
- we need to minimize the time needed for each step – i.e., in other words, perform, at each step, computations which are as fast as possible.

Which computational steps are the fastest? When we process numbers, computation on a deterministic computer means, in effect, computing the value of some function of an input. Overall, the function we compute is a composition of functions computed on consequent steps.

Among different functions of several variables, linear functions, i.e., functions of the type

$$f(x_1,\ldots,x_n) = a_0 + a_1 \cdot x_1 + \ldots + a_n \cdot x_n \tag{6}$$

are the easiest (thus fastest) to compute.

However, if we only use linear computational steps, then, due to the fact that a composition of linear functions is linear, we will only be able to compute linear functions, while in real life, many processes are nonlinear. Thus, in addition to linear computational steps, we also need some nonlinear ones.

In general, the more inputs a function has, the longer it takes to process all these inputs and to compute the value of this function. From this viewpoint, among all nonlinear functions, the fastest to compute are nonlinear functions of one variable $y = s(x)$. Thus, fastest computations should consist of two types of computational steps:

- linear steps, on which we compute a linear combination (6) of the inputs, and
- nonlinear steps, on which we compute the value of a function of one variable

$$y = s(x).$$

To make computations fast, consequent computational steps must be of different types. Indeed, if we have a linear step followed by a linear step, then all these two steps compute is a composition of two linear functions – which, as we have mentioned, is also a linear function. Thus, instead of these two steps, we can have a single linear step, in which we directly compute this composition.

Similarly, if we have a nonlinear step $y = s(x)$ followed by another nonlinear step $z = s'(y)$, then all these two steps compute is a composition $z = s'(s(x))$ of these two functions – i.e., also a nonlinear function of one variable. Thus, instead of these two steps, we can have a single nonlinear step, in which we directly compute this composition.

So, in general, to make computations faster, we need to make sure that consequent computational steps are of different types, i.e., that:

- a linear computational step is followed by a nonlinear one, and
- a nonlinear computational step is followed by a linear one.

What can we compute with the smallest possible number of computational steps. Now that we know which are the fastest computational steps, let us analyze which functions can be computed by using the smallest possible number of computational steps.

The smallest possible number of computational steps is 1. In one step, we can compute either a linear function or a function of one variable. In both statistics and decision making applications, we need to process several numbers:

- in the statistics cases, we need to take into account (and thus, to process) several observations $x_1, \ldots, x_n$, and
- in the decision making cases, we need to take into account (and thus, to process) several different possible consequences $v_1, \ldots, v_k$ of the analyzed decision.

Thus, if we limit ourselves to a single computational step, we cannot use a function of one variable. Therefore, we have to use a linear function. In case of the statistical analysis, this corresponds to using the first moment

$$E[x] \approx \frac{x_1 + \ldots + x_n}{n} = p_1 \cdot v_1 + \ldots + p_k \cdot v_k,$$

for some values p_j. In case of decision making, this corresponds to having utility proportional to the numerical value v_j of each alternative:

$$u = p_1 \cdot v_1 + \ldots + p_k \cdot v_k.$$

In line with the general fact that some real-life dependencies are nonlinear, both in statistical analysis and in decision making, we may need to use nonlinear functions to get a more adequate description. In this case, we need to use at least two computational steps.

Two stages: possible options. Due to the above, these stage must be different. So, we have two options:

- the first option is to have a linear stage followed by a nonlinear stage, and
- the second option is to have a nonlinear stage followed by a linear stage.

Two stages: first option. If the first stage is linear and the following one nonlinear, then, in general, we compute a function

$$f\left(a_0 + \sum_{j=1}^{k} a_j \cdot v_j\right).$$

Comparing such values is equivalent to comparing the corresponding linear combinations $a_0 + \sum_{j=1}^{k} a_j \cdot v_j$, and we know that such a linearized approach does not work for many real-life phenomena.

Two stages: second option. If the first stage is nonlinear and the second one linear, then we compute expressions $a_0 + \sum_{j=1}^{k} a_j \cdot f_j(v_j)$. This provides a more general opportunities for comparison.

In particular, if a priori, we have no reason to prefer some j's, then it makes sense to use the same nonlinear function $f_j(v) = f(v)$ to process all the inputs. Thus, we get the expression

$$a_0 + \sum_{j=1}^{k} a_j \cdot f(x_j). \tag{7}$$

This expression is exactly what we wanted to explain. The formula (7) is exactly what is used when we use generalized moments or expected utility. Thus, we have indeed explained the desired expressions.

3 Acknowledgments

This work was supported in part by the National Science Foundation grants:

- 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science), and
- HRD-1834620 and HRD-2034030 (CAHSI Includes).

It was also supported:

- by the AT&T Fellowship in Information Technology,
- by the program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478, and
- by a grant from the Hungarian National Research, Development and Innovation Office (NRDI).

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.
8. D. J. Sheskin, *Handbook of Parametric and Non-Parametric Statistical Procedures*, Chapman & Hall/CRC, London, UK, 2011.