

Towards Economics of Education: Optimization under Uncertainty

Olga Kosheleva¹ and Richard Aló²

¹ Department of Teacher Education, University of Texas at El Paso,
El Paso, TX 79968, USA, olgak@utep.edu

² Center for Computational Sciences & Advanced Distributed Simulation,
University of Houston-Downtown, One Main Street,
Houston, TX 77002, USA, RALo@uh.edu

Abstract. Since we cannot spend as much time as we would like to on teaching all the topics, it is necessary to optimally distribute the limited amount of time between different topics. In this paper, we explain how general techniques of optimization under uncertainty can be used in education.

1 Optimizing Content Development

Basic assumptions about training. In order to find the optimal training schedule, let us make some (simplifying but realistic) assumptions about training.

In principle, there are several different types of items that we want a student to learn; for example:

- when we teach typing, we want the student to acquire the motor skills of typing all the symbols on the keyboard and all pairs of consequent symbols;
- when we teach words from a foreign language, we want the student to learn all these words;
- when we teach cross-country driving, we want the student to develop motor skills corresponding to different types of terrain: flat surface, rugged terrain, uphill, downhill, narrow bridge, etc.

Let us denote the total number of types to learn by T . For simplicity, we assume that acquiring skills necessary for each of these types takes the same number of training situations s . So, to learn all necessary types, a student needs at least $T \cdot s$ repetitions. If we denote, by T_0 , the time necessary for handling each repetition, then the total time for training a student for all necessary types is equal to $T_0 \cdot T \cdot s$.

In many learning situations, the total number T of necessary types is large, so the above time of total training is unrealistically large. Therefore, we cannot expect every single student to be 100% skilled in every possible situation type.

Since we cannot train a student to be skilled in every possible situation, it is therefore necessary to train a student in such a way that the student will be able to handle *the largest possible number* of these types.

In future applications, some of these types are more frequent, some are less frequent. So, if we know that a student can only learn, say, t different words, and we have to choose which of these words the student will learn perfectly well, we should choose t most frequent ones.

A *skill* of a student can be thus characterized by the number t of the types of items in which this student is well skilled.

To estimate frequencies of different types, we can use a general (semi-empirical) law discovered by G. K. Zipf (see, e.g., [3, 5]), according to which, if we order types from the most frequent to the least frequent one, then the frequency f_i of i -th type is proportional to $1/i$: $f_i = c/i$ for some constant c . The value of this constant can be determined from the fact that the sum of all these frequencies should be equal to 1: $f_1 + \dots + f_T = 1$. Since $1 + 1/2 + \dots + 1/T \approx \ln(T)$, we thus conclude that $c \cdot \ln(T) = 1$, $c = 1/\ln(T)$, and

$$f_i = \frac{1}{\ln(T) \cdot i}. \quad (1)$$

Traditional learning. In traditional learning of a language, a student is trained on texts from real language. In traditional typing lessons, student learn to type by typing real-life texts. In all these cases, a student is trained on a real-life flow of items.

Let us denote by I the time allocated for training. Since handling each repetition takes time T_0 , during this training time, the trainee will see $N = I/T_0$ repetitions. According to our assumption about the training time, the student will be trained only in those types i for which he or she has seen at least s repetitions. Out of the total of N repetitions, the student will see $N \cdot f_i$ repetitions of i -th type; so, the student will be trained in all the types for which $N \cdot f_i \geq s$. Substituting Zipf's expression (1) for f_i , we conclude that the student will learn all the types i for which $\frac{I}{T_0} \cdot \frac{1}{\ln(T) \cdot i} \geq s$, i.e., for which $i \leq \frac{I}{T_0 \cdot \ln(T) \cdot s}$. Therefore, the resulting student's skill level t (i.e., the total number of types in which this student will be skilled), will be equal to

$$t = \frac{I}{T_0 \cdot \ln(T) \cdot s}. \quad (2)$$

This formula describes the skill level acquired during a given training time I .

We can also consider the inverse problem: we want a student to be trained for a certain skill level t , and we need to know the time I required for this training. From the formula (2), we can conclude that

$$I = t \cdot T_0 \cdot \ln(T) \cdot s. \quad (3)$$

Optimal training. We can generate repetitions in arbitrary order, not necessarily with real-life frequencies. If we want a student to be trained on t different types, then we need to generate exactly s repetitions of this type.

If we fix the total training time I , then during this time, we can generate $N = I/T_0$ repetitions. Since learning each type requires s repetitions, the total

amount of different types in which a student can get skilled is equal to $t = N/s = (I/T_0) \cdot s$. Thus, after this training, the student will acquire the skill level

$$t = \frac{I}{T_0 \cdot s}. \quad (4)$$

This formula describes the skill level acquired during a given training time I .

We can also consider the inverse problem: we want a student to be trained for a certain skill level t , and we need to know the time I required for this training. From the formula (4), we can conclude that

$$I = t \cdot T_0 \cdot s. \quad (5)$$

Conclusion: optimal training is faster and better. By comparing the formulas (2) and (4), we conclude that during the same training time, the skill level acquired during the automated training can be much higher ($\ln(T)$ times higher) than the skill level acquired in traditional training.

Similarly, by comparing the formulas (3) and (5), we conclude that the training time necessary to acquire a given skill can be much shorter ($\ln(T)$ times shorter) for the automated training than for traditional training.

How to optimally combine classroom and field training. Usually, after classroom training, a student goes into realistic situations (field training) which solidify his training. How can we best organize this combined training?

Let us denote the time that we can allocate for classroom training by I_{au} , and the training time for the follow-up field training by I_{tr} . During the follow-up training, the student encounters $N_{\text{tr}} = I_{\text{tr}}/T_0$ repetitions. Of these repetitions, $N_{\text{tr}} \cdot f_i$ are of type i .

If this number of repetitions is $\geq s$, then for this type, the student acquires necessary skills during the follow-up training, so there is no need to simulate patients of this type during the automated training. Thus, we get all types from 1 to

$$t_{\text{tr}} = \frac{I_{\text{tr}}}{T_0 \cdot \ln(T) \cdot s} \quad (6)$$

covered.

For each type $i > t_{\text{tr}}$, we get $f_i \cdot N_{\text{tr}} = \frac{I_{\text{tr}}}{T_0 \cdot \ln(T) \cdot i} < s$ repetitions covered during traditional training. So, if we want the student to get the necessary skills, we must generate the remaining number of repetitions

$$n_i = s - \frac{I_{\text{tr}}}{T_0 \cdot \ln(T) \cdot i} \quad (7)$$

during the automated training.

We want to learn as many new types as possible. How many situation types can we thus learn? During the time I_{au} , we can only generate $N_{\text{au}} = I_{\text{au}}/T_0$ repetitions. Since learning type i requires n_i repetitions, the skill level t acquired by

a student can be determined by the formula $\frac{I_{\text{au}}}{T_0} = N_{\text{au}} = \sum_{i=t_{\text{tr}}}^t n_i$. Substituting the above expression for n_i , we conclude that

$$\frac{I_{\text{au}}}{T_0} = s \cdot (t - t_{\text{tr}}) - \frac{I_{\text{tr}}}{T_0 \cdot \ln(T)} \cdot \sum_{i=t_{\text{tr}}}^t \frac{1}{i}.$$

Since $1 + 1/2 + \dots + 1/i \approx \ln(i)$, we can rewrite this equation as

$$\frac{I_{\text{au}}}{T_0} = s \cdot (t - t_{\text{tr}}) - \frac{I_{\text{tr}} \cdot (\ln(t) - \ln(t_{\text{tr}}))}{T_0 \cdot \ln(T)}. \quad (8)$$

So, we can make two conclusions:

- If the training times I_{au} and I_{tr} are given, then the resulting acquired skill t can be determined from the equation (8), where t_{tr} is determined from the equation (6).
- Vice versa, if we know the training time I_{au} for the classroom training, and the required skill level t , then we must find t_{tr} for the equation (8), and then use the formula (6) to determine the necessary traditional training period as $I_{\text{tr}} = t_{\text{tr}} \cdot T_0 \cdot \ln(T) \cdot s$.

In both cases, the number of repetitions of different types $i = t_{\text{tr}}, t_{\text{tr}} + 1, \dots, t$ generated during the classroom training is determined by the formula (7).

Other applications. In [1], we used a similar idea to optimize the types of virtual patients used by doctors during medical training – specifically, during a training of surgeons for spinal cord stimulation procedures; see, e.g., [2].

2 Optimal Order of Presenting the Material

Formulation of the problem. In the above section, we described the optimal frequencies with which we repeat each of the items that a student has to learn. Once we know the number of repetitions of each item, the next natural question is: in what order should we present these repetitions? Should we first present all the repetitions of item 1, then all the repetitions of item 2, etc., or should we randomly mix these repetitions?

Towards mathematical formulation of the corresponding optimization problem. Each item is characterized by several (n) numerical characteristics, so we can geometrically represent each item as a point in the corresponding n -dimensional space.

- Similar items have close values of these characteristics, so the distance between the points corresponding to similar items is small.

- Vice versa, when the items are different, they at least some of these characteristics have different values on these items, so the resulting distance is large.

Thus, the distance between the corresponding points in a multi-D space can be viewed as a measure of similarity between the items.

In terms of multi-D space, an order in which we present repetitions is described as a function $x(t)$, where x is a multi-D point corresponding to the item presented at moment $t = k \cdot \Delta t$, where Δt is the time between repetitions.

As we have mentioned, when we have a few items to learn, we can easily learn them all, so there is no need for sophisticated optimization. Optimization becomes necessary when there are many items – and thus, many repetitions. In this case, similar to the way we simplify the physical problems if we approximate a collection of atoms by a continuous medium, we can approximate the discrete dependence $x(t)$ on discrete time t by a continuous function $x(t)$ of continuous time t .

What is the optimal trajectory $x(t)$? The experience of learning shows that often, presenting the items in random order is beneficial. To allow for this possibility, instead of looking for a deterministic function $x(t)$, we look for *random* processes $x(t)$. Since a deterministic function is a particular case of a random process, we are thus not restricting ourselves.

Let us consider Gaussian random processes. A Gaussian random process can be uniquely characterized by its mean $m(t) \stackrel{\text{def}}{=} E[x(t)]$ and autocorrelation function $A(t, s) \stackrel{\text{def}}{=} E[(x(t) - x(s))^2]$.

Students come with different levels of preparation. Therefore, a good learning strategy should work not only for a student that comes from 0, but also for a student that comes at moment t_0 with the knowledge that other students have already acquired by this time. From this viewpoint, a student's education starts at the moment t_0 . It is therefore natural to require that the random process should look the same whether we start with a point $t = 0$ or with some later point t_0 . Hence, the characteristics of the process should be the same, i.e., $m(t) = m(t + t_0)$ and $A(t, s) = A(t + t_0, s + t_0)$ for every t, s , and t_0 .

From the first condition, we conclude that $m(t) = \text{const}$. Thus, by changing the origin of the coordinate system, we can safely assume that $m(t) = 0$.

From the second condition, for $t_0 = -s$, we conclude that $A(t, s) = A(t - s, 0)$, i.e., that the autocorrelation function depends only on the difference between the times: $A(t, s) = a(t - s)$, where we denoted $a(t) \stackrel{\text{def}}{=} A(t, 0)$. In other words, the random process must be *stationary*.

The final question is: what autocorrelation function $a(t)$ should we use?

We must choose a family of functions, not a single function. The function $a(t)$ depends on how intensely we train. In more intensive training, we present the material faster, and thus, within the same time interval t , we can cover more diverse topics. More diverse topics means that the average change $a(t)$ can be larger. A natural way to describe this increase is by proportionally enlarging

all the distances, which leads from $a(t)$ to $C \cdot a(t)$. In other words, if $a(t)$ is a reasonable function for some training, then a new function $C \cdot a(t)$ should also be reasonable.

We can say that the functions $a(t)$ and $C \cdot a(t)$ describe exactly the same learning strategy, but with different intensities. Since intensity can be different, we cannot select a unique function $a(t)$ and claim it to be the best, because for every function $a(t)$, the function $C \cdot a(t)$ describes exactly the same learning strategy. In view of this, instead of formulating a problem of choosing the best autocorrelation *function*, it is more natural to formulate a problem of choosing the best *family* $\{C \cdot a(t)\}_C$ of autocorrelation functions.

Which family is the best? We may need non-numerical optimality criteria. Among all the families $\{C \cdot a(t)\}_C$, we want to choose the best one.

In mathematical optimization problems, numerical criteria are most frequently used, when to every alternative (in our case, to each family) we assign some value expressing its performance, and we choose an alternative (in our case, a family) for which this value is the largest. In our problem, as such a numerical criterion, we can select, e.g., the average grade on some standardized test A .

However, it is not necessary to restrict ourselves to such numerical criteria only. For example, if we have several different families that have the same average average grade A , we can choose between them the one that has the minimal level of uncomfortableness U . In this case, the actual criterion that we use to compare two families is not numerical, but more complicated: *a family F_1 is better than the family F_2 if and only if either $A(F_1) < A(F_2)$, or $A(F_1) = A(F_2)$ and $U(F_1) < U(F_2)$.* A criterion can be even more complicated. What a criterion *must* do is to allow us, for every pair of families, to tell whether the first family is better with respect to this criterion (we'll denote it by $F_1 \succ F_2$), or the second is better ($F_1 \prec F_2$), or these families have the same quality in the sense of this criterion (we'll denote it by $F_1 \sim F_2$). Of course, it is necessary to demand that these choices be consistent, e.g., if $F_1 \prec F_2$ and $F_2 \prec F_3$ then $F_1 \prec F_3$.

The optimality criterion must select a unique optimal family. Another natural demand is that this criterion must choose a *unique* optimal family (i.e., a family that is better with respect to this criterion than any other family). The reason for this demand is very simple.

If a criterion does not choose a family at all, then it is of no use.

If several different families are “the best” according to this criterion, then we still have a problem to choose among those “best”. Therefore, we need some additional criterion for that choice. For example, if several families turn out to have the same average grade, we can choose among them a with the minimal uncomfortableness.

So what we actually do in this case is abandon that criterion for which there were several “best” families, and consider a new “composite” criterion instead: F_1 is better than F_2 according to this new criterion if either it was better according to the old criterion or according to the old criterion they had the same quality and F_1 is better than F_2 according to the additional criterion.

In other words, if a criterion does not allow us to choose a unique best family, it means that this criterion is not final. We have to modify it until we come to a final criterion that will have that property.

The optimality criterion must be scale-invariant. The next natural condition that the criterion must satisfy is connected with the fact that the numerical value of the time t depends on the choice of the unit for measuring time.

If we replace the original unit of time by a new unit which is λ times larger (i.e., replace minutes by hours), then numerical values change from t to $\tilde{t} = t/\lambda$. The autocorrelation function that in the old units is described by a family $\{C \cdot a(t)\}$, in the new units, has a new form $\{C \cdot a(\lambda \cdot t)\}$,

Since this change is simply a change in a unit of time, it is reasonable to require that going from $a(t)$ from $a(\lambda \cdot t)$ should not change the *relative* quality of the autocorrelation functions, i.e., if a family $\{C \cdot a(t)\}_C$ is better than the family $\{C \cdot a'(t)\}_C$, then for every $\lambda > 0$, the family $\{C \cdot a(\lambda \cdot t)\}_C$ must be still better than the family $\{C \cdot a'(\lambda \cdot t)\}_C$.

Definitions and the main result.

Definition 1. – *By an autocorrelation function we mean a monotonically non-strictly decreasing function from non-negative real numbers to non-negative real numbers.*

- *By a family of functions we mean the family $\{C \cdot a(t)\}_C$, where $a(t)$ is a given autocorrelation function and C runs over arbitrary positive real numbers.*
- *A pair of relations (\prec, \sim) is called consistent [4] if it satisfies the following conditions:*
 - (1) *if $a \prec b$ and $b \prec c$ then $a \prec c$;*
 - (2) *$a \sim a$;*
 - (3) *if $a \sim b$ then $b \sim a$;*
 - (4) *if $a \sim b$ and $b \sim c$ then $a \sim c$;*
 - (5) *if $a \prec b$ and $b \sim c$ then $a \prec c$;*
 - (6) *if $a \sim b$ and $b \prec c$ then $a \prec c$;*
 - (7) *if $a \prec b$, then $b \prec a$ or $a \sim b$ are impossible.*

Definition 2. – *Assume a set A is given. Its elements will be called alternatives. By an optimality criterion we mean a consistent pair (\prec, \sim) of relations on the set A of all alternatives. If $b \prec a$, we say that a is better than b ; if $a \sim b$, we say that the alternatives a and b are equivalent with respect to this criterion.*

- *We say that an alternative a is optimal (or best) with respect to a criterion (\prec, \sim) if for every other alternative b either $b \prec a$ or $a \sim b$.*
- *We say that a criterion is final if there exists an optimal alternative, and this optimal alternative is unique.*
- *Let $\lambda > 0$ be a real number. By the λ -rescaling $R_\lambda(\rho)$ of a function $a(t)$ we mean a function $(R_\lambda a)(t) \stackrel{\text{def}}{=} a(\lambda \cdot t)$.*
- *By the λ -rescaling $R_\lambda(F)$ of a family F , we mean the set of the functions that are obtained from $f \in F$ by λ -rescaling.*

In this paper, we consider optimality criteria on the set \mathcal{F} of all families.

Definition 3. We say that an optimality criterion on F is scale-invariant if for every two families F and G and for every number $\lambda > 0$, the following two conditions are true:

i) if F is better than G in the sense of this criterion (i.e., $G \prec F$), then

$$R_\lambda(G) \prec R_\lambda(F);$$

ii) if F is equivalent to G in the sense of this criterion (i.e., $F \sim G$), then

$$R_\lambda(F) \sim R_\lambda(G).$$

As we have already remarked, the demands that the optimality criterion is final and scale-invariant are quite reasonable. The only problem with them is that at first glance they may seem rather weak. However, they are not, as the following theorem shows:

Theorem 1. [4] If a family F is optimal in the sense of some optimality criterion that is final and scale-invariant, then every function $a(t)$ from this optimal family F which has the form $a(t) = A \cdot t^\alpha$ for some real numbers A and α .

In other words, the optimal configuration is a fractal random process. When $\alpha = 2$, we have a straightforward trajectory, without any randomness. The value $\alpha = 0$ means that values of $x(t)$ and $x(s)$ for $t \neq s$ are completely uncorrelated, i.e., that we have a white noise. Intermediate values of α correspond to different levels of randomness.

Our experience showed that such fractal order indeed leads to improvement in learning. The exact value of the parameter α – corresponding to the fractal dimension of the corresponding trajectories – should be adjusted to the learning style of the students.

References

1. Aló, R., Aló, K., Kreinovich, V.: Towards Intelligent Virtual Environment for Training Medical Doctors in Surgical Pain Relief, *Proceedings of The Eighth International Fuzzy Systems Association World Congress IFSA '99*, Taipei, Taiwan, August 17–20, 1999, pp. 260–264.
2. Horsch, S., Claves, L.: *Spinal Cord Stimulation II*, Steinkopff Verlag, Darmstadt, 1995.
3. Mandelbrot, B. B.: *The fractal geometry of Nature*, Freeman, San Francisco, 1982.
4. Nguyen, H. T., Kreinovich, V.: *Applications of continuous mathematics to computer science*, Kluwer, Dordrecht, 1997.
5. Zipf, G. K.: *Human behavior and the principle of least-effort*, Addison-Wesley, Cambridge, MA, 1949.