CARNAP’S INDUCTION AND MACHINE LEARNING:  
A REMARK  
Vladik Kreinovich  
Computer Science Department  
University of Texas at El Paso, El Paso, TX 79968, USA

Abstract. In our 1992 paper, we showed that Carnap’s formula for updating probabilities can be deduced from very general and reasonable assumptions. In this remark, we show that a similar approach can help in choosing a learning algorithm. Namely, it justifies the use of a well-known linear learning model.

Carnap’s formula, and our justification of Carnap’s formula. Suppose that our initial probability estimate of some event is \( p_0 \). To get a better estimate, we perform \( N \) additional experiments, and this event occurs in \( M \) of them. How to update the probability? In [Carnap 1952], R. Carnap proposed a formula

\[
p = \frac{p_0 + \lambda M}{1 + \lambda N}
\]

for an updated probability (here, \( \lambda \) is some real number). In our paper [Kreinovich et al. 1992], we showed that this formula can be deduced (using group-theoretic mathematical techniques) from very general and reasonable assumptions.

A similar situation in machine learning. Because of the general character of our arguments, they can be applied not only to the situation where expert’s uncertainty values are changed, but also to the case of machine learning. Namely, let us consider the so called reinforcement algorithms (see, e.g., [Zeidenberg 1991], Section 2.21): in these algorithms, for each situation, we know beforehand a finite set of possible actions \( a_1, ..., a_i, ..., a_r \). On each learning stage \( n = 0, 1, 2, ..., \) the learning device generates one of these actions. Before the learning starts, we must somehow show guess the probabilities \( p_0(a_i), 1 \leq i \leq r \) of all the actions. After each action, the device is told whether this action was successful or not. Depending on that, we must modify the probabilities of the actions, i.e., change \( p_n(a_i) \) to \( p_{n+1}(a_i) \). How to change?

Case of 2 actions. If we have only two possible actions, then the success of each action can be viewed as a confirmation of the hypothesis that this action is correct, and the success of the opposite action can be viewed as an argument against this hypothesis. In
other words, \( p_n(a_i) \) is the initial probability of the hypothesis \( H_i \) that action 1 is adequate, 
\( p_{n+1}(a_i) \) is an updated probability of this hypothesis. We are in a typical probability update situation, so we can apply Carnap’s formula to estimate the updated probability: 
Namely, we have performed \( N = 1 \) additional experiment. If as a result, action 1 was successful or action 2 was unsuccessful, then for action 1, this experiment was successful (\( M = 1 \)), so the updated probability equals

\[
p_{n+1}(a_i) = \frac{p_n(a_i) + \lambda}{1 + \lambda}.
\]

If action 1 was unsuccessful, or if action 2 was successful, then \( M = 0 \), and thus

\[
p_{n+1}(a_i) = \frac{p_n(a_i)}{1 + \lambda}.
\]

If we denote \( \theta = \lambda/(1 + \lambda) \) (so that \( 1/(1 + \lambda) = 1 - \theta \)), then the above two updating formulas can be rewritten in a somewhat simplified form:

\[
p_{n+1}(a_i) = (1 - \theta)p_n(a_i) + \theta
\]

if action 1 was successful and

\[
p_{n+1}(a_i) = (1 - \theta)p_n(a_i)
\]

if this action was a failure.

**General case: arbitrarily many actions.** Let’s now consider the general case of arbitrarily many actions. In this case, the success of an action \( a_i \) means a confirmation of a hypothesis \( H_i \) that this very action is correct, and it is also an argument against all other hypotheses \( H_j, j \neq i \). However, the failure of an action \( a_i \) cannot be explained in the same simple way: it is an argument against a hypothesis \( H_i \), but, unlike the case \( r = 2 \), it is not a proof of any other hypothesis. The only thing that we can say is that one of the remaining \( r - 1 \) hypotheses is true. We can say that in this case, each of the hypotheses \( H_j, j \neq i \), gets \( 1/(r - 1) \) of a confirmation.

**Application of Carnap’s formula to machine learning.** Applying Carnap’s formula with \( M = 1/(r - 1) \), we arrive at the following updating algorithm:

\[
p_{n+1}(a_i) = (1 - \theta)p_n(a_i) + \theta
\]

when on this learning step, we performed an action \( a_i \), and this action was successful;

\[
p_{n+1}(a_i) = (1 - \theta)p_n(a_i)
\]

when we performed an action \( a_i \), and this action was a failure, or when we performed some other action \( a_j, j \neq i \), and that other action was a success; and

\[
p_{n+1}(a_i) = (1 - \theta)p_n(a_i) + \theta/(r - 1)
\]

2
when some action \( a_j (j \neq i) \) was performed and failed.

**The resulting algorithm is well known in machine learning.** We arrive at a well-known linear learning model (see, e.g., [Estes et al 1959], [Narendra et al 1974], [Barto et al 1985]). This algorithm is still being used in many efficient learning systems, see, e.g., [Suppes et al 1991]). So, we come to the following conclusion:

**Conclusion.** *The same group-theoretic considerations that justify Carnap’s induction formula, also justify linear learning model.*

**Acknowledgments.** This work was sponsored by NSF grant No. CDA-9015006, NASA Research Grant No. 9-482, and Grant No. PF90–018 from the General Services Administration (GSA), administered by the Materials Research Institute. The author is also greatly thankful to Patrick Suppes for valuable comments and inspiring papers.

**REFERENCES**


