Getting It Right at the Very Start -- Building Project Models where Data Is Expensive by Combining Human Expertise, Machine Learning and Information Theory

Frank D. Francone
RML Technologies, Inc. & Chalmers University of Technology
frank.d@francone.com

Larry M. Deschaine
Science Applications International Corporation, Inc. & Chalmers University of Technology
Larry.M.Deschaine@alum.mit.edu

Keywords: Environmental Science, Geophysics, Information Theory, Underground Anomaly Detection, Machine Learning, Genetic Programming, Integrated Decision Support Systems, Expert Systems

Abstract

Building models using machine learning techniques requires data. For some projects, gathering data is very expensive. In this type of project—which we refer to as 'Incremental Learning Projects'—there are two significant costs to using machine learning techniques: (1) Machine learning models cannot even begin to make predictions until the project has already spent significant amounts of money gathering data; and (2) While the data is being gathered to train the machine learning system, unnecessary costs are incurred in making inefficient decisions.

Engineers may address this type of problem efficiently when enough human expertise exists about the problem domain to be modeled. This work proposes an approach to combining human expertise, machine learning and information theory that makes efficient and effective decisions from the start of the project, in concert with project data collection.

INTRODUCTION

This work describes an approach to creating integrated decision support systems for certain types of projects, which we refer to as 'Incremental Learning Projects.'

Characteristics of Incremental Learning Projects

Properly-designed, decision-support systems can yield significant cost savings and improved decision support for Incremental Learning Projects, which have the following characteristics:

- The project requires that engineers repeatedly make similar decisions based on relatively low cost measurements of the domain in which the decisions are to be made;
- Human experts can give a first-pass assessment for each of these required repetitive decisions. From their assessment, project engineers should be able to determine how certain the human experts were of their decision;
- The cost of acquiring actual ground-truth for the decisions required by the project is relatively high;
- One of the principal goals in the project is to avoid incurring the costs of acquiring actual ground-truth for a large number of the decision points—that is, it would be desirable that a large number of the repetitive project decisions be made using only the information available from the project's low-cost, domain measurements; and
- Finally, sufficient information exists in the low-cost measurements for machine learning systems alone or in conjunction with human experts to make better judgments than the human expert alone.

We will sometimes refer to projects with these characteristics as 'Incremental Learning Projects.'

Incremental Learning Project Example

An example of an Incremental Learning Project, which we will follow and expand upon throughout this paper, will make our discussion more clear.

Geophysicists frequently have to discriminate between different types of underground objects from readings made by electromagnetic or other above-ground, non-destructive, sensors. [1] Each signal from the above-ground sensors is

relatively inexpensive to obtain. From these above-ground measurements, geophysicists can make predictions about which signals represent target objects and which do not.

But to get ground-truth, a hole must be dug at considerable expense. Not having to dig empty holes is a primary objective of this type of project. [1] Adding a machine learning system to the mix can improve decision accuracy considerably. [2]

Obviously, this underground anomaly discrimination project fits each of the criteria set forth above.

The Chicken-And-Egg Problem in Using Machine Learning in Incremental Learning Projects

The problem with applying machine learning systems to Incremental Learning Projects is that ground-truth must be provided to machine learning systems for the purpose of deriving predictive models. [3] This poses a chicken-and-egg problem.

One technique to bypass the chicken-and-egg problem is described in [4]. That work involved a physics-based simulator that predicted the location of the fringe of an underground, groundwater-contaminant-plume. A Kalman filter system looked at randomly generated outputs from the simulator to determine where to take the next ground sample to optimally reduce the uncertainty of the simulator.

Where physics-based simulation models are not available, machine-learning techniques may be used to develop them to help make better predictions and reduce costs. But building inductive models requires expensive process information such as ground-truth to validate them. [3][5][6]

This suggests that, at best, machine learning can only begin to contribute to project decisions after the project has been ongoing for some time and has made enough mistakes to provide examples of good—and bad—decisions to the machine learning algorithm.

Step-By-Step Approach to Incremental Learning Projects

This paper proposes a step-by-step approach to Incremental Learning Projects. This approach solves the chicken-and-egg problem. Our approach involves five Steps:

1. Make a preliminary expert assessment for each of the repetitive decisions. The decisions should be ranked by probability. For example, for binary decision-making requiring a classification of each decision into ‘yes’ or ‘no’ categories, each decision should be ranked by how probable is it that this decision should be a ‘yes’ decision. In our example project, each underground anomaly detected by the sensors would be labeled with the expert assessment of the probability that it is a target that should be dug up;
2. Using the expert probability rankings from Step 1 as ground-truth, train the machine learning system to produce predictions about the result of making each of the required repetitive decisions. In our example project, the machine learning system would be trained using the experts predictions of ground-truth in lieu of actual ground truth. The machine learning system would then generate predictions for each anomaly;
3. Using the predictions from the previous Step, for each of the expected repetitive project decisions, do the following:

   (a) Determine the expected additional project cost of acquiring ground-truth by making that decision; and

   (b) Determine the information gain predicted by Shannon as a result of making that decision.

   Then incur the expense of obtaining ground-truth (dig the hole in the example project) for the decision or decisions that yield(s) the highest expected amount of new information per net-dollar expended obtaining the information;
4. Retrain the machine learning system using known ground-truth, including the new ground-truth information just acquired in Step 3. Where ground-truth is not available, use the most current expert-based assessment for each decision in lieu of ground-truth; and
5. Repeat Steps 3 and 4 until the project ends.

Improvement in Incremental Learning Projects Due to Step-by-Step Approach

The advantages of this step-by-step approach are straightforward:

- Problem-solving with machine learning techniques may begin at the start of a project rather than having to wait until sufficient data samples have been gathered on which to train the models;
- Early use of machine learning may improve decision making substantially and throughout the life of the project; and
- Improved decision making means lower cost and better results. This is particularly critical where the cost of obtaining data is high. In such projects, minimizing the number of data points for which high-cost measurement is required is often the largest cost-determinant.

Information Theory and Incremental Learning Projects

Information Theory suggests that the step-by-step approach we propose should be the most efficient means of gathering information for training a machine learning system in Incremental Learning Projects. Simply put, information theory suggests that more information is acquired about a physical system when one samples from the system and obtains an unexpected result than when the result is expected. [7]

So long as the experts' preliminary ranking of the probable effect of decisions to be made is not random, then the most unexpected results should occur in the region where the experts are *most uncertain*. Thus, by sampling ground-truth in the region of maximum uncertainty, we gather additional information about the system as efficiently as possible per sample.

In addition to the information gain expected as a result of acquiring new ground truth, there is another factor in determining the optimal ordering of project decisions—that is, which hole to dig next. That factor is the expected additional cost to the project of acquiring the ground-truth by digging any particular hole.

When we consider added project cost in conjunction with Shannon's definition of information gain, we conclude that the most efficient order of decision-making is, at any given time, to make that decision of which the predictive system is *most confident* and then feed the ground-truth thereby acquired back into the machine learning system. In other words, dig the next hole where the predictive system (human-expert-based or machine-learning-based) is most certain there is a target.

MACHINE LEARNING ALGORITHMS

Machine Learning is a term that includes a number of algorithmic approaches. When we refer to machine learning systems, we are actually referring to a subset of machine learning called 'supervised' machine learning. [3][5] Various and probably familiar machine learning techniques include back-propagation and various other neural network approaches [8], decision-tree algorithms [3][5], and genetic programming, [3][9][10][11]

The authors prefer linear genetic programming ('LGP') for the particularly difficult learning domains that are typical of Incremental Learning Projects. [2] Thus, while our discussion here will frequently refer to LGP, other sufficiently powerful machine learning tools may sometimes be substituted for LGP in this technique.

What all supervised machine learning techniques have in common is that the algorithm trains on known data with known answers. In one way or another, the algorithm develops a mapping between the domain measurements provided to it by the project engineers and the known 'answer'—or the 'ground-truth'.

Thus, and by way of example, LGP automatically produces a C program that maps the domain measurements provided by the project engineers to ground-truth. [11] With a properly trained and tested model in hand, the engineer can then apply the LGP model to inputs for which the answer is not known and use the model as a predictive tool for the remainder of the project. [3][11]

Our approach is related to, but different from other work we have done to improve process decision-making accuracy. [2] In that work, we fused the information content of human-derived models and machine-based models. This resulted in improved decision-quality accuracy beyond that which was possible using either approach separately. In that study, we assumed that ground-truth was available to train the machine-learning system from the start. By way of contrast, the present approach addresses quite a different problem—how to fuse expert analysis with machine-learning when there is no preexisting ground-truth.

The purpose of this paper is to suggest that by appropriate step-by-step sampling of ground-truth, machine learning may be integrated with human and traditional computer simulation tools even in Incremental Learning Projects to produce a better, and more cost effective, approach to this type of project.

DESCRIPTION OF INCREMENTAL LEARNING PROJECTS

We briefly described Incremental Learning Projects in the Introduction. The following observations flesh that discussion out:

First: Incremental Learning Projects involve similar decisions made over-and-over. It would be desirable to make these decisions based on relatively low-cost measurements. But these low-cost measurements do not provide experts enough information to make their decisions with certainty.

Although we have referred to human experts as having made the preliminary assessment, the nature of the preliminary assessment is not so important here. This assessment may be grounded on one or more of the following techniques, depending on the project:

- Human experts using the low-cost measurements to make preliminary decisions;
- Statistical analysis of the low cost measurements;
- A simulation model of the project using the low-cost measurements as inputs; or

- A combination of the foregoing.

Second: This approach is suitable for projects for where machine learning can make better predictions out of whatever ground-truth information is available at any given time than can the human experts or the existing simulation models.

Surprisingly to some, this project requirement is no longer uncommon for noisy, complex modeling problems. The underground objects example above is only one domain in which genetic programming outperforms human experts when looking at the same data. [2][12] Noisy domains where the physics of the problem are complex or poorly understood frequently present situations where genetic programming or other machine learning techniques outperforms human experts and human-designed rule systems or simulators.

Finally: This approach is suitable only for projects in which the cost of obtaining ground-truth is high. If it is not, then the project engineer should gather the ground-truth (at low cost, of course) and apply machine learning techniques in a traditional manner.

APPLICATION ISSUES

Our step-by-step approach for applying machine learning to Incremental Learning Projects is presented in the Introduction. Those steps raise a number of interesting application issues, which we address in this section.

Acquiring the Training Data for the Machine Learning Algorithm

The essential ingredient of the approach outlined in this paper is the manner in which the training examples are derived for the machine learning algorithm. The traditional machine learning approach would be to select training examples containing both: (1) low-cost measurements of the domain; and (2) the known ground-truth for those measurements.

Simulated Ground Truth

For Incremental Learning Projects, we suggest deriving the ground-truth for the initial training examples in a non-traditional manner. In our approach, the training examples would be derived from two sources: (1) If available, training examples should be chosen using actual ground-truth. As the project proceeds, more and more of the examples would be based on real ground-truth; or (2) If actual ground-truth is not available, the training examples should be derived from the best human, statistical, or simulator based judgment that may be derived from the low-cost measurements in lieu of ground-truth. Model development should, of course, follow the Department of Defense guidelines for verification, validation, accreditation and credibility. [6]

To use the example above, suppose the decision that must be made in the project is whether to dig expensive holes at particular spots where electromagnetic anomalies have been detected by above-ground measurements. Of course, the goal is to remove the target objects. An empty hole represents a large and unnecessary expense.

To do this, Geophysicists would examine the low-cost measurements and assess the probability of whether these measurements reveal a target object that must be removed. Anomalies with an assessed probability in excess of a project specific threshold would be labeled, for the purpose of training the machine learning algorithm, as "TARGET." Those below the threshold would be labeled as "NOT TARGET." That would provide 'ground-truth' to the machine learning system in the absence of actual ground-truth.

The Effect of Using Simulated Ground Truth

The effect of constructing training examples in this manner is that machine learning may be integrated with human expertise in an almost risk-free manner. That is, a properly trained and tested machine learning model would begin making predictions at or above the level of the human experts, whose judgment it has effectively reverse engineered. Machine learning systems (in particular linear genetic programming) have been quite successful in such 'reverse engineering the experts' type applications. [2]

The machine learning system typically starts by making predictions as good as those that would be made by the best human-experts or the best available simulators. As more ground-truth is acquired during the course of the project—in the above example, as more holes are dug—those actual examples supplement and/or replace the human-expertise based examples. Typically, a machine learning system's predictions will improve as it acquires more—and more accurate—training data.

Accordingly, we should expect such an approach to do no worse than the human experts. In reality, the machine learning system, after training on the human expert predictions, often immediately improves on the human predictions. What happens is that the machine learning system finds regions of the input space where the expert predictions are inconsistent with expert predictions elsewhere. By identifying those inconsistencies *ab initio*, the machine learning system clarifies the expert's domain knowledge for better predictions at the start. [2]

And, as the project moves forward—and more ground-truth is acquired—we should expect such an approach frequently to out-perform the alternative approach which

uses only human, statistical or simulator-based expertise throughout the project. [2]

Choosing the Cost Function

All supervised machine learning systems require a 'cost function.' In effect, a cost function tells the algorithm when a particular model is doing better, or worse, at solving the problem at hand. The cost function is used by machine learning algorithms to move thru the search space of possible models. [3][5]

In linear genetic programming, the cost function is referred to as a 'fitness function.' This nomenclature comes from LGP's history as an evolutionary algorithm. Evolutionary algorithms draw on analogies to Darwinian natural selection—survival of the fittest. Thus, the genetic programming fitness function is used to determine which models survive and 'reproduce' during training. [3]

In the above example, a simple fitness function would just tally up how many anomalies a particular model has classified correctly as "DIG" or "DON'T DIG." Models that classify training examples more accurately would be assessed in the cost function as more 'fit.'

But in this approach, all training examples are not equal. Off-hand, at least three general categories of training examples may be delineated:

1. Examples based on human-expert evaluations—that is, there is no known ground-truth for the example—and in which the experts have **low confidence** in their prediction.
2. Examples based on human-expert evaluations—that is, there is no known ground-truth for the example—and in which the experts have **high confidence** in their prediction.
3. Examples based on actual, measured ground-truth. In almost all cases, these examples should be regarded as the examples in which we have the most confidence.[1]

One important decision that project engineers must make is whether and how to weight these different cases in the cost, or fitness function. Assigning different costs to different training examples is a frequently used technique in machine learning [5] [11] and it seems particularly applicable in this situation. Manifestly, an error by the algorithm on a training example that involves known ground truth seems more serious than an error on an example where the training example is based on an expert's low-confidence judgment.

The details of using differential cost functions would be the subject of a different, and much longer paper. Nevertheless it is important to note here:

First, that the issue should be explicitly resolved by project engineers based on the particulars of the project, and

Second, that effect of similar weighting schemes can be very different depending on which machine learning algorithm is used. For example, the author's experience with differential cost functions in evolutionary algorithms such as LGP suggests a very small weighting differential can have much more substantial effects than the same differential applied in decision-tree algorithms. [11]

Ordering the Acquisition of Ground-Truth for Optimal Project Performance

Improvements in machine learning predictions for the project will depend on how much new information is acquired during the project. New information is acquired by learning ground-truth. Thus, Shannon's measure of the amount of information that may be acquired by choosing to dig here instead of there is very useful.

Shannon described the foundations of Information Theory in 1948. [7] The information obtained by making an observation that has an *a priori* probability, *p*, of occurring—$I(p)$—is defined as:

$$I(p) = p * Log(1/p); \quad (1)$$

Using Equation (1), the remainder of this section describes two different approaches to integrating information theory into the project decision-making process. One approach seeks to maximize the amount of *information acquired per decision made*. The later seeks to maximize *the information acquired per additional project dollar expended obtaining that information*.

The Simple Information Theoretic Approach to Ordering the Samples to Obtain Ground-Truth

In the above example, assume that the geophysicists have made preliminary assessments of each of the geophysical anomalies. In doing so, they have assigned probabilities that each anomaly represents a target that should be dug up. In that case, it is trivial to show that the maximum Shannon information (see Equation 1) would be obtained by digging the anomaly about which the geophysicists are most uncertain about whether their decision is correct.

Put another way, where the geophysicists assign a probability of 0.50 that a particular anomaly is a target that

[1] The exception to this general rule would occur where engineers determine there is a reason to suspect the low-cost measurements such as instrument error, calibration problems and the like.

should be dug up, their uncertainty is highest. Regarding that anomaly, the expected value of the information to be obtained from digging it up is greater than or equal to the expected information obtainable from any other anomaly in the domain.[2]

Thus, the strategy suggested by this simple information theoretic approach is to dig next, that anomaly for which the TARGET vs. NOT TARGET classification is most uncertain.

A More Sophisticated Information Theoretic Approach to Ordering Samples to Obtain Ground-Truth--Integrating Project Costs and Information Theory

A second measure by which project engineers could order the digging of anomalies would be to maximize the expected amount of information acquired by digging a hole per dollar of expected additional project-costs caused by digging.

The cost of digging a hole would be the simplest way to measure the cost of obtaining ground-truth for a particular anomaly. But that does not really represent the expenditure of additional monies for that anomaly if there is a probability assessed to the anomaly that the anomaly is a target. Rather, if $p(i)$ represents the probability that the *i*th anomaly is a target, the expected incremental cost to the project of digging that anomaly $Cost(inc)$ is:

(2) $$Cost(inc,i) = (1 - p(i)) * Cost(dig);$$

where $Cost(dig)$ is the expected cost of digging a hole.

This may be illustrated by an example. Suppose the probability that the *i*th anomaly is actually a target is 0.90. In that case, project engineers expect to dig that hole one way or the other. So digging it now adds an additional expected cost to the project measured by the probability that the hole will be empty—that is, 0.10, as suggested by Equation (2).[3] Thus, if it costs \$200 to dig a hole, the expected additional cost to the project of digging this anomaly to acquire information is only \$20.

The information gain per dollar of additional cost expected from digging up the *i*th anomaly may be formalized by combining Equations 1 and 2. The expected information gain per dollar of added project cost from digging the *i*th anomaly— $I(\$, i)$ — is stated in Equation 3.:

(3) $$I(\$, i) = \frac{p(i) * \log(1/p(i))}{(1 - p(i)) * Cost}$$

It is simple to demonstrate computationally that, as $p(i)$ increases from 0 to 1, $I(\$, i)$ increases steadily, for all positive values of *Cost* (Equation 3).

We can therefore conclude as the anomaly digging project proceeds, engineers should, at any point, dig the anomaly that then has the highest probability of being a target. By doing so, they can maximize the expected information gain per dollar of added costs— $I(\$)$.

Furthermore, so long as project engineers can rank the anomalies in the project from most likely to least likely to be targets, the same dig ordering holds, even if we cannot assign specific probability numbers. This follows from the fact that, if $p(i) > p(j)$, then $I(\$, i)$ is greater than $I(\$, j)$ (see Equation (3)).

This leads to a somewhat different conclusion than the simple application of Equation (1), discussed above. Instead of digging up the most uncertain anomaly first, this analysis suggests that project engineers can minimize the cost of acquiring information by, at each point in the project, digging up the anomaly that they are *most* certain is not a false positive.

Machine Learning as a Decision Support Tool in Concluding the Project

At some point in the project, the engineers have to stop digging. Otherwise, the machine learning system saves no money. In this section, we propose a simple metric for making that determination.

The effectiveness in this metric depends on project engineers adopting the second method of ordering acquisition of ground-truth proposed above—that is, ground-truth is acquired by starting with the decision about which human or machine predictors are most certain. Then the next most certain. Then the next. And so forth.

---

[2] Although we refer in the preceding paragraph to 'geophysicist' based estimates of probability, such estimates will be made purely by the geophysicists only in the early stages of the project. As the project continues, and the machine learning system begins to make predictions, those probabilities would be assigned by the machine learning system alone or (more likely) by the machine learning system after review of its predictions by the geophysicists.

[3] Similar reasoning leads to the conclusion that project engineers may ignore the cost of *not* digging until the very end of the project. The decision not to dig imposes no additional costs on the project until the decision becomes irrevocable—that is, at the end of the project. Until the end of the project, the hole can always be dug. So deciding not to dig it at an earlier point in the project adds no cost to the project.

That metric is also based on the project designers assigning a cost—$Cost(fn)$—to making a false-negative decision. In our example project, $Cost(fn)$ represents the cost of a decision not to dig up an anomaly that turns out to be a target. Our example project is over when engineers decide not to dig up all anomalies remaining.

At each step, the machine learning system has assigned a probability that the *i*th anomaly is a target, $p(i)$. The incremental cost to the project of not digging the hole—$Cost(nd)$—is:

(4) $$Cost(nd) = p(i) * Cost(fn).$$

As long as $Cost(nd)$ is greater than $Cost(dig)$, engineers should keep digging because the cost of digging the hole is less than the cost of a false negative.

So, for example, if the cost of a false negative is $5,000 and the cost of digging a hole is $200, engineers should keep digging until $p(i) = 0.04$.

CONCLUSION

In this paper we have presented a novel approach to integrating machine learning techniques with human expertise and human-built simulators on projects with a high cost of obtaining data.

Engineers with projects similar to those described in this paper should consider utilizing the techniques described herein to integrate machine learning capabilities into their projects.

REFERENCES

[1] Ernesto R. Cespedes, (September 2001). *Advanced UXO Detection / Discrimination Technology Demonstration--U.S. Army Jefferson Proving Ground, Madison, Indiana*. US Army Corps of Engineers, Engineer Research and Development Center.

[2] Francone, F. D., Deschaine, Larry, M. (2004). *Extending the Boundaries Of Design Optimization by Integrating Fast Optimization Techniques with Machine-Code-Based, Linear, Genetic Programming*. (In press) Journal of Information Sciences, Elsevier Press.

[3] Banzhaf, W., Nordin, P., Keller, R, and Francone, F. (1998) *Genetic Programming, An Introduction,* Morgan Kaufmann Publishers, Inc, Stanford CA.

[4] Deschaine, L. M., *Simulation and Optimization of Large Scale Subsurface Environmental Impacts; Investigations, Remedial Design and Long Term Monitoring*. In press: Journal of "Mathematical Machines and Systems", National Academy of Sciences of Ukraine, Kiev. In press. 2003.

[5] Langley, Pat. (1998) *Elements of Machine Learning,* Morgan Kaufmann Publishers, Inc, Stanford, CA.

[6] Department of Defense, *DoD Defense Modeling and Simulation Guidelines for model verification, validation, accreditation and credibility* (available at https://www.dmso.mil/public/transition/vva/)

[7] Shannon, C. E., (1948), *A Mathematical Theory of Communication,* Bell System Technical Journal 27, 379-423.

[8] Masters, Timothy. (1995) *Advanced Algorithms for Neural Networks*, John Wiley & Sons, Inc. New York, New York.

[9] Koza, John. (1996) *Genetic Programming, On the Programming of Computers by means of Natural Selection.* MIT Press, Cambridge Massachusetts.

[10] Nordin, P., Banzhaf, W. and Francone, F. (1999) "Efficient Evolution of Machine Code for CISC Architecture Using Instruction Blocks and Homologous Crossover." In *Advances in Genetic Programming, Volume III*, edited by Spector, Lee *et al.* MIT Press, Cambridge MA, 275-300.

[11] Francone, F., (2000) *Discipulus™ Owner's Manual*. Available at www.aimlearning.com.

[12] Koza, J., Bennett, F., Andre, D., & Keane, M. (2001). *Genetic Programming III: Automatic Programming and Automatic Circuit Synthesis.* MIT Press, Cambridge, MA.