Genetic Programming for Discrimination of Buried Unexploded Ordnance (UXO)

Dr. Edwin Roger Banks,
Dr. Edwin Núñez, Paul Agarwal
COLSA Corporation
6726 Odyssey Drive
Huntsville, AL 35806

Dr. Claudette Owens,
Marshall McBride, Ron Liedel
U.S. Army Space & Missile Defense
Command (USASMDC)
Huntsville, AL 35807-3801

ABSTRACT

According to the Department of Defense, over 10 million acres of land in the US need to be cleared of buried unexploded ordnance (UXO). Worldwide, UXO injures thousands each year. Cleanup costs are prohibitively expensive due to the difficulties in discriminating buried UXO from other inert non-UXO objects. Government agencies are actively searching for improved sensor methodologies to detect and discriminate buried UXO from other objects. This paper describes the results of work performed on data gathered by the GeoPhex GEM-3 electromagnetic sensor during their attempts to discriminate buried UXO at the U.S. Army Jefferson Proving Ground (JPG). We used a variety of evolutionary computing (EC) approaches that included genetic programming, genetic algorithms, and decision-tree methods. All approaches were essentially formulated as regression problems whereby the EC algorithms used sensor data to evolve buried UXO discrimination chromosomes. Predictions were then compared with a ground-truth file and the number of false positives and negatives determined.

Categories and Subject Descriptors

G.6-Global Optimization

Keywords

Genetic programming, UXO, ordnance, classification algorithms, data mining

1 INTRODUCTION

At GECCO 2004, Francone *et al* [1] presented a late breaking paper titled “Discrimination of Unexploded Ordnance from Clutter using Linear Genetic Programming”. Later last year, we attempted to attack the same problem using a variety of evolutionary computing approaches.

Francone *et al’s* main result was based on data from the U.S. Army Jefferson Proving Ground Phase V (JPG-V) tests. Unfortunately, to date we have not been able to obtain the same data sets (sensor data plus ground truth data). In lieu of that data, we accessed data from an earlier experiment at the same location called JPG Phase IV (JPG-IV). This data is available at http://www.eoir.com/uxo/JPG4.htm.

We wanted to apply genetic programming and other similar techniques to determine how well our independently derived results compared with those reported earlier by the original JPG-IV participants.

2 PROBLEM DESCRIPTION

Ten vendors participated in the original JPG-IV experiment. Participants included private companies and government groups, as well as combinations of both. Several participants used sensors of their own design while some used those designed by others. Three main types of sensor technologies were available: magnetic (M), electromagnetic (EM), and ground penetrating radar (GPR). From the available data, we chose to re-examine the data obtained by the GeoPhex Limited GEM-3 sensor. This sensor takes data using electromagnetic coils. We chose this data set because it seemed to have enough density of data to allow for the development of adequate discrimination chromosomes, yet enough variety to challenge the technology.

In this JPG-IV experiment, a total of 160 targets were catalogued and deliberately buried underground: 50 ordnance and 110 other assorted “fragments.” A flag marked the position of all ordnance and fragments. Vendors did not know what kind of object was buried beneath the flags. Targets varied in size, mass, and the depth at which they were buried. Indeed, target weight varied over a scale of 450 to 1 (20 mm to 155 mm ordnance). Readings were taken at 25 grid-points separated by 9 inches and with the center grid-point directly over the target. Each reading involved 8 different frequencies, ranging from 30 to 23,970 Hz. The in-phase (I) and quadrature (Q) components were measured at each point and frequency. This yielded 400 measurements for each target (25 points x 8 frequencies x 2 components) to use with our genetic software program. A ground truth file was also available. It listed not only the ordnance type, but also its depth, length, weight, azimuth, declination, and other properties. Figure 1 below provides examples of the ordnance buried in the JPG-IV test.

3 CHROMOSOME ENCODING

Our first task in the processing of the available sensor data was to obtain a set of “features” or “operators” that could be used to develop the chromosomes. In a problem of this nature, these are not obvious. Creativity is required to determine what might work at extracting information useful for discrimination purposes. For example, the average of the I and Q values over all sensor measurement points was one such feature. Some of these operators were parameterized, taking one or more arguments, typically using the frequency as the parameter. An early attempt to simply use the raw data only and evolve the features did not work as well as the use of such pre-computed features. Some of the features we selected were based on what we deemed to be reasonable physics models, although we are by no means subject matter experts. Other features were rather capricious, such as one that measures how the spatial gradient of signal intensity rotates with frequency. The idea was to let the selection process weed out poor operators.

In addition to the derived operators, we also allowed a variety of arithmetic operators such as addition, subtraction, exponentiation, logarithms, as well as some conditionals (*if-then-else*, *greater than*, etc.).

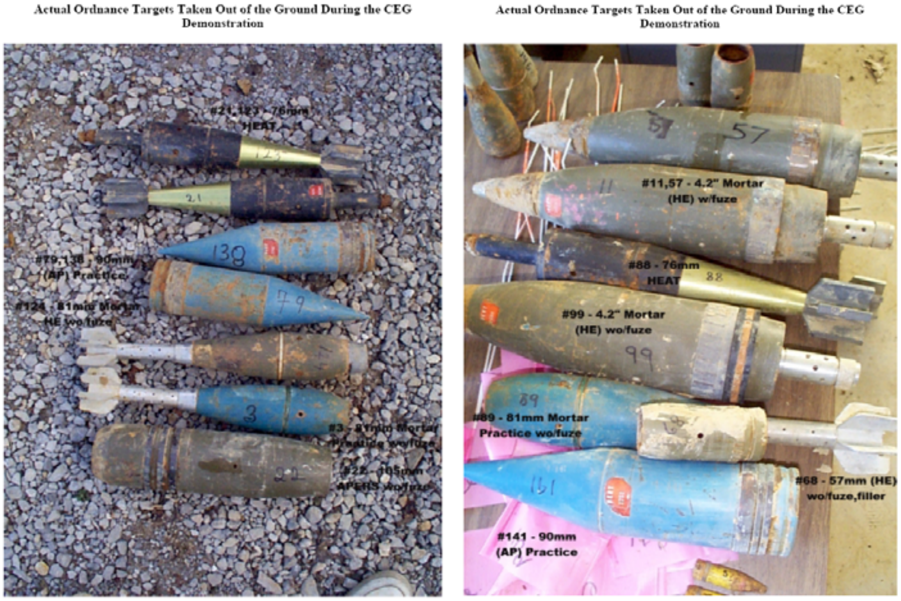
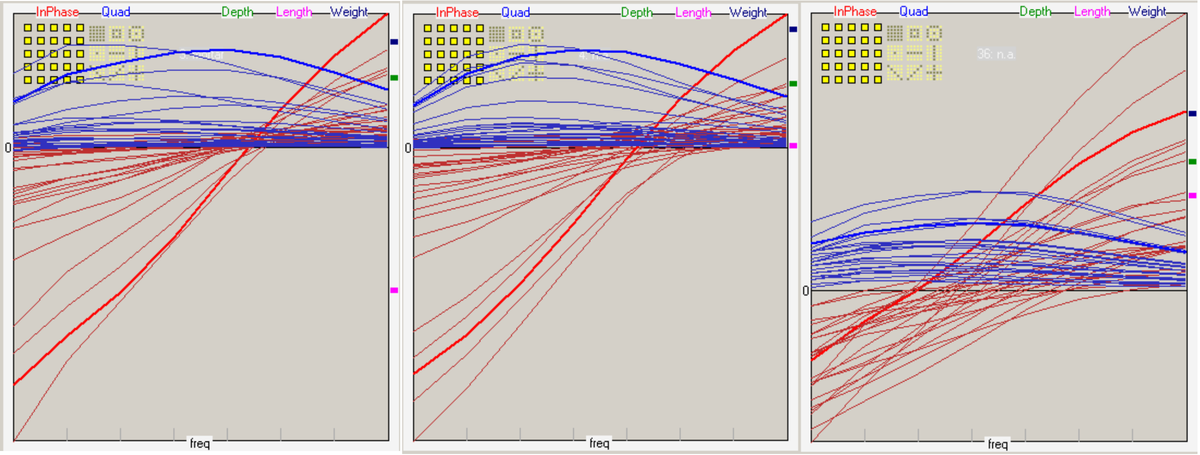

Actual Ordnance Targets Taken Out of the Ground During the CEG Demonstration
#21,123 - 76mm HEAT
#79,138 - 90mm (AP) Practice
#124 - 81mm Mortar HE wo/fuze
#3 - 81mm Mortar Practice wo/fuze
#22 - 105mm APERS wo/fuze
Actual Ordnance Targets Taken Out of the Ground During the CEG Demonstration
#11,57 - 4.2" Mortar (HE) w/fuze
#88 - 76mm HEAT
#99 - 4.2" Mortar (HE) wo/fuze
#89 - 81mm Mortar Practice wo/fuze
#68 - 57mm (HE) wo/fuze,filler
#141 - 90mm (AP) Practice

Figure 1. Examples of ordnance buried in the JPG-IV test

Figure 2 illustrates the difficulty in discrimination of these buried objects. This figure shows 3 charts of the in-phase (I) and quadrature (Q) sensor readings as a function of frequency for each of the 25 grid-points where measurements were taken. The bold lines are for the point directly at the flag in the center of the grid. Note that the first chart is for a mortar (ordnance) and the latter two are for fragments although the first two are far more similar!



Target-3 (Mortar) Target-4 (Fragment) Target-36 (Fragment)

Figure 2. Three plots of I and Q as a function of frequency

Data for each target could be examined individually using this analysis tool that we called the "target explorer". The target explorer provided some visual insight as to possible features that might be used for discrimination.

4 GENETIC PROGRAMMING ENCODING

In terms of our discrimination results, the most successful approach was to use genetic programming (GP). We encoded the chromosomes as linear strings of operators (genotype) composed of several genes. For GP we used chromosomes of 10 genes, with 2 genes each encoding for depth, length, and weight, and 4 genes encoding for discrimination, that is, determination of whether the target represented ordnance or not. Each pair of genes was simply evaluated over the sensor data for a target. This evaluation produced two numbers that were added to yield an estimate for depth, length, and weight. For the ordnance discrimination the last four genes were combined in various ways (arithmetic and logical). The best results turned out to be simply their sum. The sum was compared to a reference number (typically 1.0) and if greater, the target was presumed to be ordnance, otherwise it was classified as non-ordnance.

5 GENETIC ALGORITHM ENCODING

A GA encoding was also attempted in the form of a single large gene, comprising a constant weight value for each of the available features. The fitness F_{pn} (see later) was then simply the linear weighted sum of features as compared to a reference value. The result was quite inferior to GP, but due to the nature of the features selected, this was to be expected. We may yet return to GA to combine the features in a less-linear way. At present, we are not using this approach.

6 DECISION TREE ENCODING

Based on a discrimination approach described in Freitas [2], we also encoded the chromosome as a decision tree (DT) problem. It offered the advantage of much faster evaluation since the result of evaluating the first gene determined which half of the remaining genes should to be evaluated. The balanced binary tree resulted in only logarithmically increasing evaluations when compared with the normal GP approach. Leaf nodes then answered whether a target was ordnance or not. The DT approach also proved to be inferior to GP and suffered more strongly from overfitting, a problem discussed in the next section.

7 THE OVERFITTING PROBLEM

One of the most troubling problems we had to overcome was the development of chromosomes that would show excessive overfitting to the data. In the total of 50 ordnance objects in the data set, we had many different types and sizes. Weight varied from 0.1 kg (20 mm high explosive) to over 43 kg (155 mm high explosive with lifting lug). The data contained only 3 or 4 occurrences of each type of item. Thus it was very easy for the GP to evolve some almost irrelevant expression that just happened to match those very few instances. Thus, that chromosome would only fit that data set: the chromosome would be overfit.

To avoid the problem of overfitting we divided the whole 160-target data set into three groups: a Training set, a Test set, and a Validation set. Typically these sets included 50%, 40%, and 10% of the data, respectively. Great care was taken to have a similar distribution of all target types in all three data sets. The chromosomes were evolved using the Training data set only. Chromosome fitness was tested against the Test set and the best result selected. Finally, we report the score as this chromosome was applied to the previously unseen validation set. To ensure that we obtained a representative chromosome, each 10% of objects in the Validation set were held back. As a result, each Validation set did not overlap with any other and the entire data set was covered and used in the final evaluation (in a jackknife technique).

The classification problem is aggravated by the fact that the classification was binary: ordnance or non-ordnance (O/NO). Both types of classification exhibited very large variations in mass and length, as well as the depth at which they were buried. To alleviate this problem we included a measure of confidence in the result as a magnitude adjustment. When used, this adjustment considered the value of the chromosome as compared to the reference value. Thus, a value of 1.0002 when the chromosome was applied to a particular target would be a determination of ordnance (since it exceeds the reference value of 1.0), but the confidence level is very low, so its effect on the overall fitness was diminished. Similarly we looked at the signal strength as another sort of magnitude adjustment. When the overall signal strength was very weak, we diminished its effect on fitness because it was hardly anything more than noise. We would rather misclassify such ordnance than have an adverse affect on training on targets with stronger signals.

In all cases a “dig list” was produced listing UXO digging priority. The items with most confidence (highest values) almost certainly were ordnance, while the ones with lowest values were almost certainly non-ordnance. It was those intermediate values that were more likely to be in error.

8 GENOTYPE

We use a genotype encoding called Karva that follows Ferreira [3, 5-6]. We also sometimes use other encodings such as Reverse Polish Notation (RPN), but have found Karva usually superior. In brief, if one views the tree structure for an expression, then reads the operators and variables in a top to bottom, left to right order, the resulting string of symbols is a Karva expression. Thus each gene is simply a linear string of operators and features (variables). These genes are then “compiled” for maximum speed.

9 FITNESS FUNCTION FOR GP

Fitness is computed by evaluating the Karva expressions to produce a number which is then in turn compared to a fixed reference value to decide whether the target is ordnance or not. This decision is then compared with the ground truth file. If the decision is incorrect, the result is a false positive (FP) or false negative (FN). Then the fitness value to be minimized is given through the following two equations:

$$F_{pn} = FN + R*FP \qquad (1)$$

$$F = DLW_{fitness} + F_{pn} \qquad (2)$$

In equation (1) F_{pn} indicates the total number of UXO false positives and false negatives. It is a number we want to minimize. This means that FN and FP give the number of misclassified buried objects. FN designates the targets classified as non-UXO that in fact were UXO. In contrast, FP corresponds to the number of targets classified as UXO when they really did not belong in that category. R is a coefficient that weighs the relative importance of FN to FP. Experts in charge of clearing an area of buried UXO want to avoid as much as possible leaving any genuine UXO buried in place. Thus, they give greater importance to discrimination algorithms that reduce FN. The coefficient R is always less than 1 to accomplish this task.

Although we concentrate on F_{pn}, we typically evolve on the full fitness function F given by equation (2). F, which is to be minimized, includes a measure of the error in depth, length, and weight, designated as $DLW_{fitness}$. We typically set the F_{pn} score with a much higher relative importance so that $DLW_{fitness}$ may not be the predominant feature. The genes for O/NO have reference to the $DLW_{fitness}$ gene's values as additional operators. They were added to put a bit of selection pressure on discovering good measures of these properties.

10 PARAMETERS

Our genetic programming software provides a very large set of adjustable parameters, including several types of mutation, recombination, population manipulation operations, selection pressure, and others. There are approximately 85 adjustable items in total. Through experience we can usually establish a reasonable set of values for starting the evolution, and then refining the set by making many runs.

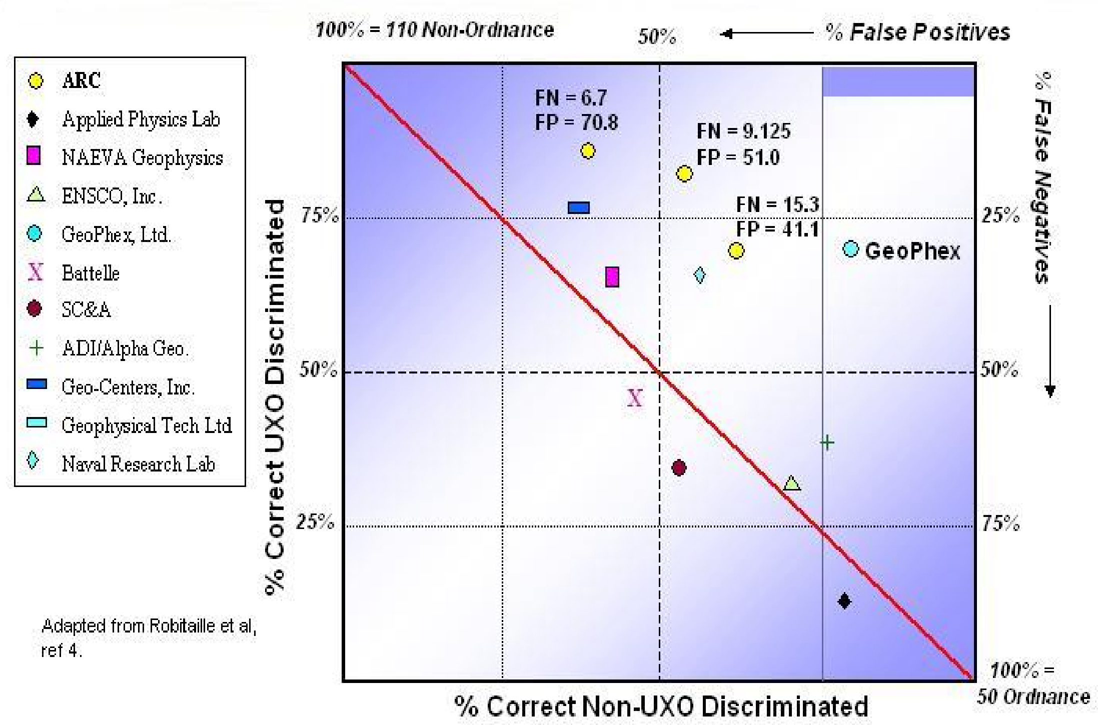
We typically used a basic mutation rate of about 3%, (μ, λ) selection, a selection pressure of about 25% (100% is pure rank selection, 0% is pure random selection), a population of a few hundred chromosomes, and computation duration of 6-12 minutes. Ultra-long computation durations of 10 days or so generally were inferior due to overfitting.

11 RESULTS

Figure 3 on the next page shows our results, with FN and FP scores. This chart illustrates a single value for most vendors and shows how they fared on the Jefferson Proving Ground Area IV test range. We show three of our results. The three results vary due to the use of a different ratio parameter R that specifies the relative importance of reducing FN and FP. The shaded rectangle in the upper right corner is the "region of desired operational performance" according to [4]. This rectangle corresponds to a maximum of 5% FN (ordnance left in the ground) and 25% FP (non-ordnance unnecessarily to be dug up). The plotted FN and FP scores are fractional due to averaging a large number of separate runs.

12 CONCLUSION

Although we have been unable to show an apples-to-apples comparison with the results of Francone [1] due to lack of access to that data set, we have validated their use of EC in addressing this type of classification problem. Furthermore we have expanded thereon by providing additional information such as predicted burial depth, length, and weight as well as a confidence level in our predictions.



Adapted from Robitaille et al, ref 4.

Figure 3. SMDC/ARC Results

13 REFERENCES

1. Frank D. Francone, Larry M. Deschaine, Tom Battenhourse, Jeffery J. Warren, “Discrimination of Unexploded Ordnance from Clutter using Linear Genetic Programming”, GECCO 2004

2. Alex A. Freitas, Data Mining and Knowledge Discovery with Evolutionary Algorithms, Springer, Berlin, 1998

3. Ferreira, C., 2001. Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. Complex Systems, 13 (2): 87-129

4. George Robitaille, Jane Adams, Chris O’Donnell, Pope Burr. “Jefferson Proving Ground Technology Demonstration Program Summary”, 1999

5. Ferreira, C.: Gene Expression Programming in Problem Solving, WSC6 Tutorial, (2001)

6. Ferreira, C., Gene Expression Programming in Problem Solving. In R. Roy, M. Köppen, S. Ovaska, T. Furuhashi, and F. Hoffmann, eds., Soft Computing and Industry – Recent Applications, pages 635-654, Springer-Verlag, 2002

7. George Robitaille, Project Officer. “UXO Technology Demonstration Program at Jefferson Proving Ground, Phase IV”, Report SFIM-AEC-ET-CR-99051. May1999