

USING GENETIC ALGORITHMS TO SOLVE THE STRATEGIC LEARNING PROBLEM

Fidan Boylu

Operations and Information Management, School of Business, University of Connecticut, CT
fidan.boylu@business.uconn.edu

Haldun Aytug • Gary J. Koehler

Decision and Information Sciences, 351 BUS, The Warrington College of Business Administration,
University of Florida, Gainesville, FL 32611
aytugh@ufl.edu • koehler@ufl.edu

None of the existing data mining algorithms take into account the possibility that future observed attributes might have been deliberately modified by their source when the source is a human or collection of humans. They fail to anticipate that people (and collections of people) might “game the system” and alter their attributes to attain a positive classification. For example, there are many websites that show how to increase one's credit score. Typically, attributes might be altered to help achieve a positive classification. We investigate this potential strategic gaming and have developed inference methods to determine better discriminant functions in the presence of such strategic behavior and show that this strategic behavior results in an alteration of the usual learning rules. We call this problem the Strategic Learning problem. In this paper, we provide a Genetic Algorithm for solving the Strategic Learning problem. We start by reducing the Strategic Learning problem to an unconstrained search over the space of linear functions. Once we have accomplished this, we develop a Genetic Algorithm (GA) to perform this search.

An Unconstrained Formulation for Strategic Learning

Strategic Learning is defined as the task of a principal whose goal is to discriminate between certain type of agents who are self-interested, utility maximizing and decision making units. For example, a credit card company (the principal) decides which people (agents) get credit cards. The principal anticipates that agents may try to alter their attributes for a positive classification.

Here, we focus on linear discriminant functions (LDFs) for binary classification where a non-zero vector $w \in \mathfrak{R}^n$ and a scalar b are determined such that the hyperplane $w'x + b = 0$ partitions the n -dimensional Euclidian space into two half-spaces. Then, $(w, b): \mathfrak{R}^n \rightarrow \{-1, +1\}$ where $+1$ denotes the positive class and -1 denotes the negative class. Each agent i has a true vector of attributes x_i , a true label $y_i \in \{-1, +1\}$, a reservation cost r_i and a vector of costs for modifying attributes c_i . Reservation cost is the maximum effort an agent is willing to exert in order to be classified as a positive agent. On the principal's side, C_{y_i} is the penalty associated with the margin shortfall (ξ_i) of an agent of true type y_i .

In Boylu et al. (2006), the following mixed integer program (MIP) was proposed where LDFs are determined by support vector machines (Cristianini and Shawe-Taylor 2000). For

$$q_i(w, b) = \begin{cases} 0 & \text{if } 1 - b - w'x_i < 0 \\ 0 & \text{if } 1 - b - w'x_i > z_i \\ 1 - b - w'x_i & \text{otherwise} \end{cases}$$

and $z_i \equiv r_i \max_{j: (c_i)_j \neq 0} |w_j| / (c_i)_j$ for $\lambda \geq 0, C_{y_i} > 0$ and $c_i \geq 0$ with at least one j satisfying $0 < (c_i)_j < \infty$,

the MIP is

$$\begin{aligned} P1: \quad & \min_{w, b} w'w + \sum_{i=1}^{\ell} C_{y_i} \xi_i + \lambda \sum_{y_i=+1} q_i(w, b) \\ & \text{s.t. } y_i \{w'x_i + q_i(w, b) + b\} \geq 1 - \xi_i \quad i = 1, \dots, \ell \end{aligned}$$

For a 1-norm variant, $w'w$ is replaced by $\sum_i |w_i|$ in the objective. We abstract both versions using

$m(\|w\|)$ where $m(\cdot)$ an increasing function of the appropriate norm of w with $m(0) = 0$.

Now, letting $\xi_i(w, b) = \max(0, 1 - y_i [w'x_i + q_i(w, b) + b])$ we get an unconstrained version of the Strategic Learning problem

$$f(w, b) = m(\|w\|) + \sum_{i=1}^{\ell} C_{y_i} \xi_i(w, b) + \lambda \sum_{y_i=+1} q_i(w, b).$$

We can only solve this to within epsilon feasibility of the strict inequalities in the definition of $q_i(w, b)$.

Let $\varepsilon > 0$ be sufficiently small and fixed. Thus, we are interested in epsilon-minimizing $f(w, b)$. Let $b(w)$ be an epsilon optimal solution to $\varepsilon - \min_b f(b|w)$ where $f(b|w)$ denotes the function $f(w, b)$

for a fixed w . $f(b|w)$ does not have any nice features such as quasiconvexity (for example, see Figure

1). However it is still easy to find an epsilon-optimal solution (see below). Table 1 gives the possible regions of the value $C_{y_i} \xi_i(w, b) + \lambda q_i(w, b)$ for positive agents and $C_{y_i} \xi_i(w, b)$ for negative agents.

$f(b|w)$ is linear in the different regions. The key is to search over the finite starting points for the different regions. For negative points at the point of discontinuity, $1 - z_i - w'x_i$, we evaluate the function at an epsilon lower point because the function jumps up at the discontinuity. Algorithm 1, below, examines each of these points and evaluates the function $f(b|w)$ at these points. The proof of epsilon-optimality follows from knowing that between two consecutive (non-epsilon) starting points the cost function is linear, meaning only the end points need be evaluated. Furthermore, we ignore points between

an epsilon neighbor $b - \varepsilon$ and b . So we march through potential values of b noting the one with the lowest total cost. This provides an epsilon optimal value of b ($b(w)$) and reduces search to $w \in \mathfrak{R}^n$..

Algorithm 1

```

Input w.
Output an epsilon-optimal b.
Set  $\alpha = \emptyset$ 
for ( $i = 1, \dots, \ell$ ) {
    if ( $y_i = +1$ )
         $\alpha \leftarrow \alpha \cup \{1 - z_i - w'x_i\}$ 
        if ( $\lambda > 0$ )  $\alpha \leftarrow \alpha \cup \{1 - w'x_i\}$ 
    else
         $\alpha \leftarrow \alpha \cup \{1 - z_i - w'x_i - \varepsilon\}$ 
         $\alpha \leftarrow \alpha \cup \{1 - w'x_i\}$ 
        if ( $z_i < 2$ )  $\alpha \leftarrow \alpha \cup \{-1 - w'x_i\}$ 
}
Set  $f \leftarrow \infty, b \leftarrow \infty$ 
for each  $\hat{b} \in \alpha$  {
     $\hat{f} \leftarrow f(\hat{b} | w)$ 
    if  $\hat{f} < f$  then {  $f \leftarrow \hat{f}, b \leftarrow \hat{b}$  }
}
Output b;
```

Also, Lemma 1 given in Boylu (2006) provides an upper bound and removes $w=0$ from search.

Lemma 1

$$f(0) = \begin{cases} 2C_{+1} \sum_{y_i=+1} y_i & \sum_i C_i y_i \leq 0 \\ -2C_{-1} \sum_{y_i=-1} y_i & \sum_i C_i y_i \geq 0 \end{cases}$$

Thus we can solve the Strategic Learning problem by searching over $w \in \mathfrak{R}^n / \{0\}$. We have found it advantageous to search over the unit sphere $B = \{w : w'w = 1\}$ where we also determine a scaling factor $\gamma \in \mathfrak{R}$ for each w . Let $f(\gamma | w)$ be the total cost function as γ varies for a fixed w and corresponding epsilon optimal solution $b(\gamma w)$. For all the problem sets we have examined we have noticed that $f(\gamma | w)$ appears to be quasiconvex in γ when $C_{y_{+1}} < C_{y_{-1}}$. Whether this is true in general or under some reasonable set of assumptions remains unknown at this time. However, assuming it is true, we can

perform our search as follows. Generate a $w \in B$. Perform a univariate search over $\gamma \in \mathfrak{R}$ where, for each such γ we use Algorithm 1 to solve for $b(\gamma w)$. This process yields a solution γw . If our assumption about the quasiconvexity of $f(\gamma | w)$ is not true, then we just search over $w \in \mathfrak{R}^n / \{0\}$.

A Genetic Algorithm Formulation for Strategic Learning

We propose a Genetic Algorithm (GA) for solving the Strategic Learning problem. We represent population strings as a string of bits representing an n-dimensional vector w . The real-valued coefficients are limited to those that can be represented by 32 bits. We use one bit as a sign and the rest for the magnitude. The GA produces new population members through a mixing process consisting of mutation and crossover operators. With probability, χ , two selected strings are mated. The two strings produce two children formed using an affine linear crossover operator (Davis 1989). One of the children is randomly selected. If the strings do not mate (with probability $1 - \chi$), one is randomly selected to survive. Once all the new member strings are formed, mutation operators are applied. The GA mutation operator is a uniform mutation operator where each bit is flipped with probability μ (the mutation rate).

Once a string is chosen it is scaled by γ found to minimize $f(\gamma | w)$ using a one-dimensional search. Parent strings are selected using rank selection (Goldberg 1989). The fitness function used in ranking is based on a lexicographic ordering involving three values. The three values, in order of importance, are $f(\gamma w)$, the margin and the number of non-zero coefficients. A string is more fit, lexicographically speaking, if, first, its value $f(\gamma w)$ is lower. If the two strings have equal $f(\gamma w)$ values, we then choose the string having the larger margin. If these are also equal, we choose the string having the smaller number of non-zero values.

Experimental Results

To test the efficacy of this GA approach, we performed a run on a 100 point subset of the German credit data set. Of the 50 attributes in this dataset, only those listed in Table 3 could be altered by strategic manipulation. We used $C_{+1} = 1.0$, $C_{-1} = 1.1$, $\lambda = 0.5$, $\epsilon = 10^{-8}$ and $r_i = 15$.

For this run we used GA parameters of $\chi = 0.2$, $\mu = 10^{-3}$, and population size of 10. We stopped after no significant changes were observed in the last 5,000 populations. We stopped at 13,932 populations. The results of the 2-norm mixed integer programming model (MIP) developed in Boylu et al. (2006) and the GA approach developed in this paper are compared for $\lambda = 0.5$ in Table 2. We also include a non-strategic solution. The MIP was stopped after 4 hours with a gap of 12.57%. (Hence, it shows a higher objective value than the non-strategic solution.)

When the results of the approaches are compared, we see that the MIP outperforms GA in many ways. First, the MIP forces more positive agents to move and the total misclassification cost is substantially lower compared to GA results. Also, the objective value of the MIP solution is lower than GA's. The GA has lower positive effort values and larger margin (i.e., $1/w'w$) at the expense of higher misclassifications.

Of interest are the qualitative comparisons of coefficients for the attributes that can be changed. With one minor exception (-.000004 versus 0.000081), all the signs of the GA solution were the same as the MIP solutions. Furthermore, the values are not that dissimilar. The GA runs considerably faster than the MIP and scales well, so solving larger problems is possible. More work remains.

Discussion and Future Research

In this paper, we have reduced the Strategic Learning problem to an unconstrained search over $w \in \mathfrak{R}^n$ and provided a Genetic Algorithm for solving the problem. Perhaps, the most interesting capability of this model is its power to scale up to large sample sizes in comparison to the mixed integer programming model developed in Boylu et al. (2006). Also more work needs to be completed to identify the set of parameters for which the function $f(\gamma|w)$ is quasiconvex in γ . Although the results presented in this paper are promising, further analysis is required.

References

- Boylu, F., H. Aytug and G. J. Koehler, "Discrimination with Strategic Behavior," Decision and Information Sciences, University of Florida, Gainesville, FL, 2006.
- Cristianini, N., J. Shawe-Taylor. 2000. *An introduction to support vector machines and other kernel-based methods*. Cambridge University Press, Cambridge, UK.
- Davis, L. 1989. Adapting operator probabilities in genetic algorithms. *Proceedings of the Third International Conference on Genetic Algorithms*. Schaefer, J. D., ed. Morgan Kaufman, Los Altos, California, 61-69.
- Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley, Reading, MA.

Table 1. Different regions of costs

Positive Cases		Negative Cases	
$\lambda = 0$	$\lambda > 0$	$z_i < 2$	$z_i \geq 2$
$(-\infty, 1 - z_i - w'x_i)$	$(-\infty, 1 - z_i - w'x_i)$	$(-\infty, -1 - w'x_i]$	$(-\infty, 1 - z_i - w'x_i)$
$[1 - z_i - w'x_i, \infty)$	$[1 - z_i - w'x_i, 1 - w'x_i)$	$[-1 - w'x_i, 1 - z_i - w'x_i)$	
	$[1 - w'x_i, \infty)$	$[1 - z_i - w'x_i, 1 - w'x_i)$	$[1 - z_i - w'x_i, 1 - w'x_i)$

		$[1 - w'x_i, \infty)$	$[1 - w'x_i, \infty)$
--	--	-----------------------	-----------------------

Table 2. 2- norm Non-Strategic, Strategic MIP Solutions versus GA

Attribute name	Non-Strategic	MIP	GA
0 - Checking Account Balance	-0.000966	-0.000229	-0.000699
1 - Duration	-0.038404	-0.038071	-0.055754
17 - Credit Amount	0.000021	-0.000004	0.000081
18 - Savings Account Balance	0.000602	0.000540	0.000521
19 - Employment Since	0.040227	0.044727	0.050870
20 - Instalment rate	-0.078931	-0.032124	-0.184315
28 - Residence Since	0.174345	0.152243	0.268443
33 - Age	0.030289	0.028991	0.032976
40 - # of Existing Credit Cards	-0.175251	-0.190314	-0.291851
45 - Number of Dependents	0.210563	0.393598	0.083564
b^*	0.449352	0.360815	0.364540
# of Positives Moved	0	11	3
# of Negatives Moved	0	0	0
# of Positive Misclassifications	16	12	22
# of Neg. Misclassifications	21	20	23
Misclassification cost	36.269351	34.953474	46.566870
$w'w$	6.673283	7.784247	1.992520
Positive cases effort $\lambda \sum a_i$	0.000000	0.324718	0.046568
Objective Value	42.942634	43.062440	48.605959
Time (seconds)	0.078	14401.792	197.532

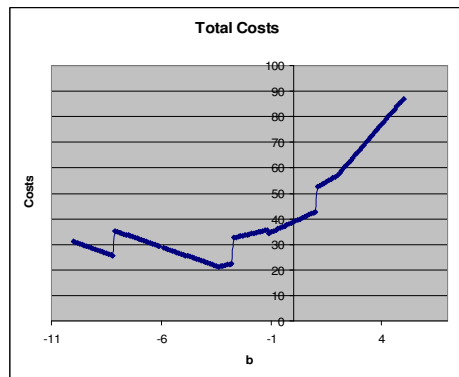


Figure 1. A typical graph of $f(b|w)$