Inferring Strategies from Observed Actions: A Nonparametric, Binary Tree Classification Approach *

Jim Engle-Warnick
Nuffield College
Oxford University
Oxford OX1 1NF, UK
jim.engle-warnick@nuff.ox.ac.uk

August 1, 2001

Abstract

This paper introduces a non–parametric binary classification tree approach to inferring unobserved strategies from the observed actions of economic agents. The strategies are in the form of possibly nested if–then statements. We apply our approach to experimental data from the repeated ultimatum game, which was conducted in four different countries by Roth et al. (1991). We find that strategy inference is consistent with existing inference, provides new explanations for subject behavior, and provides new empirically–based hypotheses regarding ultimatum game strategies. We conclude that strategy inference is potentially useful as a complementary method of statistical inference in applied research.

Keywords: binary tree, classifier, strategy, bargaining, nonparametric, resampling, experimental economics

*I wish to thank John Duffy, Jean-Francois Richard, Alvin E. Roth and Bradley Ruffle. I also thank Alvin E. Roth for providing the data from the Roth, Prasnikar, Okuno–Fujiwara, and Zamir (1991) ultimatum game experiments. Special thanks to Jayne Engle Warnick. A previous version of this paper was a winner in the Society for Computational Economics Graduate Student Paper Contest, July 2000. This research was supported by the University of Pittsburgh through an Andrew W. Mellon pre–doctoral fellowship.

1 Introduction

One approach to the problem of statistical inference in applied research is to make assumptions about how actions or decisions might be conditioned on strategically important variables and then to conduct a regression analysis. Another approach is to use non–parametric techniques to determine the statistical significance of the difference in behavior between experimental treatments. For applied work with a theoretical basis in game theory, however, the solution (i.e., an equilibrium) to a game requires the specification of strategies for every player. While a regression may assist in describing behavior, it is generally not interpretable as a strategy. If instead we could characterize observed decisions using strategies we could bridge a gap between theory and observed behavior. The problem is that we observe *actions* but not the *strategies* that generated the actions.

In this paper we introduce a computational procedure to infer unobserved strategies in the form of nested if–then statements from the observed actions of economic agents. An example of a bargaining strategy could be "if my opponent offers me more than 40% of a pie accept the offer, otherwise reject the offer". We use classification analysis (Breiman et al., 1984) to fit strategies to the data, and extend the procedure using a resampling scheme that tests the robustness of the inference result.

The procedure is an algorithm which automates part of the model specification process. The researcher specifies strategy primitives that are thought to be important to decision–making such as time, past actions or payoffs, or realizations of variables such as interest rates or unemployment rates. From the list of strategy primitives the algorithm simultaneously selects both the explanatory variables and their respective coefficients. Resampling provides an estimate of the distribution of not only the estimated coefficients but also the functional form of the unobserved strategy.

This article builds on the literature in which researchers investigate the decision rules and strategies that people play in games. Selten and Mitzkewitz (1997) and

Selten and Buchta (1999) elicit strategies from subjects that are played against each other. The "strategy method" makes strategies observable; this paper focuses on environments in which strategies are unobservable. For unobservable decision rules El–Gamal and Grether (1995), e.g., introduce a maximum likelihood approach in a game with incomplete information using cut-off rules similar to those in this paper. Engle–Warnick and Ruffle (2001) present a maximum likelihood approach derived from El–Gamal and Grether (1995) for inferring repeated–game strategies for individual agents. Duffy and Engle–Warnick (2001a) use symbolic regression in combination with a genetic algorithm to infer strategies from actions, and Engle–Warnick and Slonim (2001) present experiments which were specifically designed to infer repeated-game strategies in the form of deterministic finite automata in a trust game. This paper differs by applying binary tree classification to the problem of strategy inference.

We find by computer simulation that our technique accurately recovers known data generating strategies and interpret its output under behaviorally plausible types of misspecification. We find evidence for the existence of population strategies in the data of the Roth et al. (1991) bargaining experiment which was conducted in four different countries. In two of the four countries we infer nearly identical strategies, in a third country we infer a clearly different strategy, and in a fourth country we find that decision making cannot be characterized by a strategy of this type at all. We conclude that the strategy inference approach provided new information regarding decision making in the experiments and that it provided new hypotheses regarding empirically–based strategy models in the ultimatum game.

We begin by defining the binary classification tree and its measure of predictive performance. We then introduce the estimation procedure and discuss its properties and limitations. We introduce resampling and summarize results from simulated data (the details are presented in the Appendix). Lastly, we present results from the bargaining experiments. We conclude with a summary and argue that the technique complements existing inference approaches for experiments, and that it is potentially useful for applied work in general.

2 Binary Classification Tree Definition

We model strategies with binary classification trees because they are interpretable as if–then statements, because there is an extensive literature on a wide variety of applications from hospital triage to military friend–or–foe identification (e.g., Breiman et al., 1984), and because statistical properties of their estimation are known (Devroye et al., 1994). Binary classification tree analysis has also been used for modelling multiple regimes in cross-country growth behaviour (Durlauf and Johnson, 1995) and United States output fluctuations (Cooper, 1998).[1] What's new in this paper is the interpretation of the classification tree as a strategy.

2.1 Classification Systems

Imagine your car is making an unusual noise when you drive it. You take it to the auto shop and the technician asks you a series of diagnostic questions. Is the noise in the front or back of the car? Is it intermittent or constant? Is the pitch of the noise constant with the speed of the car? If the answers to these questions are front, constant, and variable the mechanic may check the history of cars for which the answers were the same and find that the vast majority of the time the transmission had failed. These questions are based on past experience with cars that exhibit unusual noises.

This is essentially the problem we face when inferring strategies from actions. Where the auto shop has data on the symptoms of the failures we have the realizations of economic variables; where the auto shop has data on the final diagnoses we know the actions taken by the economic agents. Using their data the auto shop has determined the best questions to ask to arrive at the diagnosis of the problem. Our goal is to use our data to find the most likely questions the decision–makers asked when making their decisions, i.e., to find their strategies.

[1] Hansen (2000) introduces a related threshold regression model and applies it to the growth model of Durlauf and Johnson (1995).

2.2 The Strategy Model: The Binary Classification Tree

Binary classification trees can be interpreted as nested if–then statements; a graphical representation is presented in Figure 1. The context of the strategy is the "ultimatum game", in which a proposer makes an offer to split 1000 units of currency.[2] The responder then accepts or rejects the proposal. The strategy in Figure 1 is for a responder. An acceptance implements the proposal and a rejection sends both players home with nothing.

The binary tree that models a responder strategy in Figure 1 consists of two nodes. The top node, which is called the *root node*, is labelled "Proposal $\leq$ 384.9". The lower node is an *internal node* and is labelled "Previous Proposal $\leq$ 368.9". We will refer to nodes equivalently as *splits* because they split the data into different decision categories (i.e., "accept" and "reject"). The bottom nodes, which are labelled "accept" and "reject", are referred to as *terminal* nodes. The tree consists of three *sub–trees*, which are formed by eliminating exactly one of the internal nodes. Since there are two internal nodes, exactly zero, one, or two internal nodes may be removed leaving behind the entire tree, the tree consisting of only the root node, or no tree.

After receiving a proposal a responder who is playing the strategy in Figure 1 asks "is the proposal less than or equal to 384.9?". If the answer is "no" she proceeds down the right branch (labelled "F" for false) of the root node of the tree and accepts the proposal. If the answer is "yes" then she proceeds down the left branch of the tree (labelled "T" for true) and then asks the question "was the previous proposal less than or equal to 368.9?". If this answer is "no" then she rejects the proposal, and accepts it otherwise.

[2] Although the context here is specific to the ultimatum game, classification trees may be applied to any environment in which if–then statements are plausible strategies. Other functional forms may be used as well: see Duffy and Engle–Warnick (2001b) for an application to linear regression analysis.

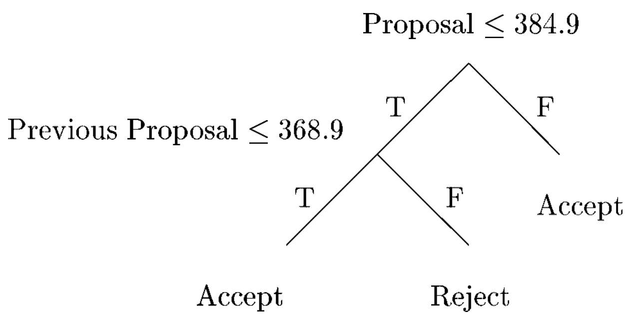

Figure 1: A Binary Regression Tree

3 Binary Classification Tree Estimation

To estimate a binary classification tree the data are divided into two disjoint sets: a *training sample* and a *testing sample.*[3] The training sample is used to identify a class of binary trees that best–fit the data. The testing sample is used to specify the best tree in this class according to out–of–sample predictive power. The purpose of the testing sample is to control for overfitting the data. The following sub–sections provide the steps of the algorithm.

3.1 Splitting the Data: Training and Testing Samples

The first step is to divide the dataset $D_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, which consists of decisions $Y_i = \{0, 1\}$ and explanatory variables $X_i \in \Re^d$, into two sets: a *training* set $D_\ell = \{(X_1, Y_1), \ldots, (X_\ell, Y_\ell)\}$ and a *testing* set $D_m = \{(X_{\ell+1}, Y_{\ell+1}), \ldots, (X_{\ell+m}, Y_{\ell+m})\}$, where $\ell + m = n$. In our application, we take $\ell = m$ because of good performance using simulated data. Data points are selected for the training set by drawing from the set D_n without replacement and with equal probability placed on each data point for each draw. After ℓ data points are drawn for training the remaining data are used for testing.

[3] See Devroye et al. (1994) for binary classification tree estimation theory.

3.2 Fitting the Data: The Impurity Function

The second step is to find the tree that best–fits the training set D_ℓ. Fitness is defined with respect to how well the tree sorts the decisions at its terminal nodes. To motivate the concept of fitness imagine two observations of data. The first observation is an acceptance of a proposal of 425, and the second is a rejection of a proposal of 370 when the previous proposal had been 350. Now take the first data point and hold it above the root node of the strategy in Figure 1 and drop it through the tree. Since the proposal was greater than 384.9 it falls to the right–most terminal node. Do the same with the second data point; it falls first down the left branch of the root node (the proposal is less than 384.9) and then down the left branch of the second non–terminal node (the previous proposal was less than 368.9) to the left–most terminal node.

Once each observation is dropped through the tree there is a collection of acceptances and rejections at the terminal nodes. The fitness of the tree depends on the homogeneity of the decisions at the terminal nodes and is represented by an *impurity function,* which takes on a maximum value for equal numbers of each decision, and a minimum value when all decisions are of the same category.

Figure 2 is presented to illustrate how the impurity function works. At each terminal node we present the number of acceptances and rejections that occurred in a hypothetical data set. One can get a sense that the right–most terminal node, where there are seven times as many acceptances as there are rejections (70 vs. 10), is helping to fit the data well, the middle terminal node somewhat less with a ratio of less than 2 to 1 for rejections to acceptances, and the left–most node even less with an even smaller ratio of acceptances to rejections.

The impurity function we use here is the Gini Function, a common measure of inequality. The Gini score is computed as $2p(1-p)$ where p is the proportion of either decision type (acceptance or rejection) at the terminal node and $1-p$ is the proportion of the other decision type.[4] The fitness score of a node is the sum of

[4] For simplicity the algorithm which was used in this paper takes a monotonic transformation of this function and uses the product of the number of acceptances and number of rejections as the

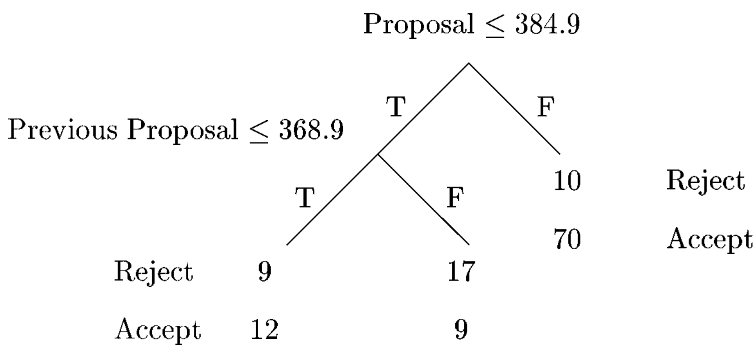

Figure 2: Impurity Function Example

the Gini scores at its two terminal nodes. The tree is "grown" one node at a time by finding the variable–coefficient pair that results in the lowest node fitness score.

Taking p as the proportion of rejections at each node, the fitness score for each terminal node from left to right in the example in Figure 2 is $2\frac{9}{21}(1-\frac{9}{21}) = 0.49$, $2\frac{17}{26}(1-\frac{17}{26}) = 0.453$, and $2\frac{10}{70}(1-\frac{10}{70}) = 0.245$. These fitness scores order the effectiveness of the terminal nodes from right to left, with the right–most terminal node most effective and left–most node least effective in classifying the data.

The best–fit tree is grown as follows. Beginning with the root node of the tree, search over every possible coefficient of every variable to find the split that minimizes the fitness of the resulting tree. Repeat this procedure at all terminal nodes of the resulting tree one split at a time until a stopping criterion is reached (e.g., an upper limit on impurity or a maximum number of splits). In this study the stopping criterion is three splits.[5] The tree is forced to make the majority decision at each terminal node.

score.

[5] As the tree becomes more complex, fewer observations are found at each node, increasing the number of splits that fit the data equally well. In the data it was not uncommon to find more than 1000 strategies that fit the data equally well when four splits were permitted; since this did not occur with three splits or less we identified three splits as a breakpoint. For examples of reasoning to levels of three in a different context see Nagel (1995) and Stahl and Wilson (1997). Endogenizing the stopping rule is a subject for further research.

3.3 Specifying the Model: Selecting the Best Sub–tree

The third step is to specify the model. Consider the set of sub–trees (defined in Section 2.2) denoted C_ℓ of the best–fitting tree. Let ϕ denote a member of this set. The *testing* sample D_m is used to select a classifier ϕ_{best} from C_ℓ. This is done by minimizing the number of classification errors, $\hat{L}_{\ell,m}(\phi)$, which are committed on the testing sample by choice of $\phi \in C_\ell$:

$$\phi_{best} = \min_{\phi} \hat{L}_{\ell,m}(\phi) = \frac{1}{m} \sum_{i=\ell+1}^{\ell+m} I_{\{\phi(X_i) \neq Y_i\}},$$

where $\phi(X_i)$ is the binary tree that maps the explanatory variables to an action and I is an indicator variable. Hence $I_{\{\phi(X_i) \neq Y_i\}}$ takes on the value of 1 when a misclassification occurs and 0 otherwise.

As an example let Figure 1 represent the decisions of a tree which have been determined by data on a *training* sample and let Figure 2 represent the collection of *testing* sample decisions which fell to the terminal nodes. Since the tree specifies (from left to right across terminal nodes) accept, reject, and accept the number of misclassified data points is the number of (from left to right) rejections + acceptances + rejections $= 9 + 9 + 10 = 28$. Now to find the optimal subtree, begin by removing the split that specifies the "previous proposal" leaving only the tree with the single split on the "proposal" and its two terminal nodes. Assume now that from the training sample, the left terminal node specified the action accept and the right one reject. Now the number of misclassified data points is equal to the number of acceptances + rejections $= 21 + 10 = 31$. Since removing the node resulted in a sub–tree that committed more errors on the testing sample than the original tree, we specify the entire tree as the inferred strategy.

In general we remove internal nodes from the best–fitting tree, one at a time, and test for an improvement in the out–of–sample predictive power of the tree. We continue the procedure from the bottom of the tree up until removing nodes no longer improves the out–of–sample fitness, or until the tree itself is completely eliminated.

Table 1: Training Sample: Growing the Tree

1	Divide the data into a training and a testing sample
2	Begin with the root node of the tree
3	Select a terminal node
4	Select an explanatory variable
5	Select a coefficient value
6	Attach the split to the tree and store its fitness score
7	If any coefficient values remain, go to 5
8	If any explanatory variables remain, go to 4
9	If any terminal nodes remain, go to 3
10	Add the split with the best classification value to the tree
11	If stopping criterion not reached, go to 3

Table 2: Testing Sample: Specifying the Tree

1	Count the number of misclassification errors for best–fitting tree
2	Remove an internal node which is attached to two terminal nodes
3	Count the number of misclassification errors for resulting sub–tree
4	If sub–tree error count is less than tree error count remove the node
5	If any internal nodes remain go to 2

3.4 The Estimation Procedure: Summary

The algorithm finds the best–fit tree using the *training* sample by minimizing the impurity score of the tree one split at a time until a stopping criterion is reached. From the best–fit tree it selects the sub–tree that minimizes the error count in the *testing* sample as the output of the regression. Tables 1 and 2 summarize the tree–growing and tree specification algorithms.

3.5 Resampling: A Robustness Test

The estimation procedure returns the best out–of–sample predictor from the class of trees defined by the best–fit tree in–sample. How accurate is this estimate? There could, for example, be more than one tree that fit the data equally well. This *observational equivalence* of trees occurs when there is no realization of decisions in the data that enable the researcher to distinguish between competing theoretical models of strategies in terms of their ability to fit the data. Two related issues are (1) there exist distributions for which trees that are generated by impurity functions are not consistently estimated, and (2) estimates of the predictive performance of such strategies can be inaccurate (see the bootstrap example in Efron, 1982).

To address these issues we introduce a resampling scheme. Resampling validates the impurity function method of binary tree inference by checking for consistency between randomly drawn samples, and avoids the arbitrary assignment of a single best strategy to the data. We randomly select a training and corresponding testing sample from the data set 1000 times and estimate a binary classification tree on each of these random samples. By reporting the frequency of occurrence of each type of strategy that results from each of the 1000 regressions we produce an estimate of the distribution of the *functional form* of the strategy we are inferring.

From sample to sample the functional form of the estimated strategy may differ considerably. Trees may differ from each other by both the *number* of splits and the *type* of splits they contain. In the strategy presented in Figure 1 there is one split on the proposal level and one on the previous proposal level. With a different randomly drawn sample there may, for example, have been only a single split on the proposal level, or there may have been two splits on the proposal level and none on the previous proposal level. The number of possible types of strategies depends on the number of candidate explanatory variables.[6]

To report the types of strategies that result from the regression output on each random sample we introduce the following notation. Define the strategy type as a vector $z \in I^d$, where each element of z represents the number of times the variable $x_i \in X$ (X is the vector of candidate explanatory variables) occurs in a split in the strategy. For example if $d = 4$ (i.e., there are four explanatory variables, e.g., x_1: the proposal, x_2: the previous proposal, x_3: the previous decision, and x_4: time), and the inferred strategy is given in Figure 1, then the strategy type is $z = \{1, 1, 0, 0\}$. After reporting the frequency distribution of the inferred strategy types we will report the functional form of the modal strategy, an estimate of the mean and the variance of the split coefficients, and an estimate of the mean and the variance of

[6] There are four possible one–split, ten possible two–split and 20 possible three–split strategy types in the set of strategies we consider in the following section.

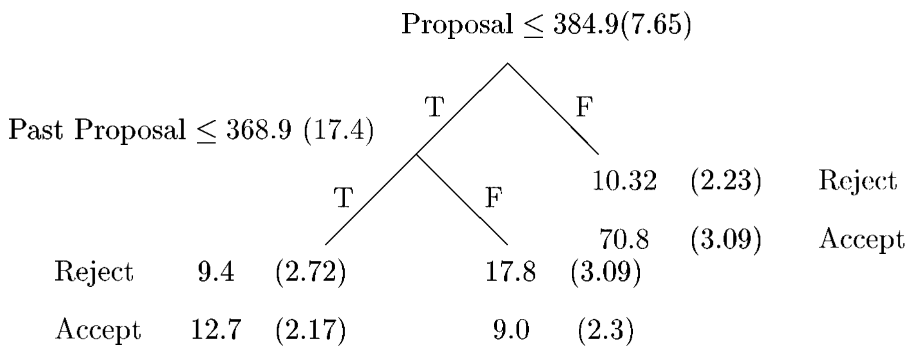

Figure 3: Modal Strategy Report Format

the number of acceptances and rejections at the terminal nodes.

For example, Figure 3 displays the modal strategy from an estimate on actual data. At each split we report the variable, and the mean and the standard deviation of its estimated coefficient. At the split labelled " Proposal $\leq$ 384.9 (7.65)" the variable is "Proposal", the mean of the estimated coefficient is 384.9, and its standard deviation is 7.65. At each terminal node we report the mean and variance of the number of rejections and acceptances that occurred in the data. For example, the right–most terminal node in Figure 3 reveals that the mean number of rejections at this node was 10.32 with a standard deviation of 2.23, and the mean number of acceptances was 70.8 with a standard deviation of 3.09.

In summary we resample the data and estimate a strategy on each randomly selected sample. We report the output of this regression in two stages. In the first stage we report the number of times we find each specific realization of a combination of variables that make up the splits of the strategy (i.e., each realization of a strategy type). In the second stage we report the functional form of the modal strategy type, the estimates of the variable coefficients, and the mean number of acceptances and rejections at the terminal nodes.

4 Simulated Data Inference Results

In the Appendix we present the results of the inference procedure using simulated data. We infer strategies when the data are generated by a single known strategy (with errors), by two strategies that condition on different variables, and by players who learn over time. We show that the procedure accurately recovers an estimate of the known strategy, that results are interpretable under the condition of heterogeneity when there are two strategies, and that results are interpretable under the learning condition. With these results we turn to the experimental data.

5 The Four Country Ultimatum Game Experiment

In this section we illustrate the procedure with data from the four country ultimatum game experiment of Roth et al. (1991). This experiment is a good test of the procedure because the game and the experiment are well known, and because it provides an environment where levels may well be important to decision making. The random repairing of subjects makes the assumption of independence of observations reasonable. Inferring reasonable levels in an experiment that has been well studied will help to validate the new estimation technique while providing a demonstration of the additional information that can be learned from strategy analysis. [7]

Subjects were randomly and anonymously paired to play a one–shot ultimatum game. Proposers made an offer to responders to split a pie of 1000 units of currency in increments of 5 units, and then responders accepted or rejected the proposal. An acceptance implemented the proposal and a rejection resulted in a payoff of 0 to both subjects. The experiments were conducted in Jerusalem, Ljubljana, Pittsburgh, and Tokyo. It is well known that the proposal levels differed in the four treatments, and that the probability of rejection was lower in countries where the proposals were lower. Proposals were highest in Pittsburgh and Ljubljana and lowest in Jerusalem. The rate of acceptance of proposals was lowest in Ljubljana and highest in Jerusalem

[7] For different interpretations of the behavior in this experiment see Costa–Gomes and Zauner (2001) and the references therein.

Table 3: Ultimatum Game Variables and Coefficients

Explanatory Variables			
Variable	Notation	Coefficients	Increment
Time t proposal	P(t)	0-1000	5
Time t-1 proposal	P(t-1)	0-1000	5
Time t-1 decision	D(t-1)	0	-
Round number	t	1-9	1

and Tokyo. We will further investigate this behavior by inferring strategies from the actions of the responders.

The set of conditioning variables will be the time t proposal, $P(t)$, the time $t-1$ proposal, $P(t-1)$, the decision to reject or accept the proposal at time $t-1$, $D(t-1)$, and time (the round number), t. The set of conditioning coefficients for $P(t)$ and $P(t-1)$ is the set of integers from zero to 1000 in increments of five. For $D(t-1)$ we include only the integer 0 for conditioning and interpret the statement "if $D(t-1) \leq 0$" as a rejection if true and acceptance otherwise. The conditioning set for t is one through nine. We expect $P(t)$ to figure prominently in decision making, and allow the algorithm to determine whether any of the other variables should be included in strategies. Time is included to test for an end game effect. The variables and their parameters are summarized in Table 3.

6 Experimental Results

Table 6 presents the frequency distribution of the number of splits in the inferred strategy for each of the 1000 randomly drawn samples in the experimental treatments. Each entry represents the number of strategies that were inferred with the corresponding number of splits. For example, in the Jerusalem treatment 806 zero–split, 0 one–split, 138 two-split, and 56 three-split strategies were inferred. The table reveals that the modal number of splits in the Jerusalem, Ljubljana, Pittsburgh and Tokyo treatments were zero, two, two, and two with the smallest frequency of occurrence of 669 in Ljubjana and the largest 936 in Pittsburgh. There appears to be evidence of population strategies in three of the four treatments.

Table 4: Frequency Distribution of Number of Strategy Splits

Number of Splits	0	1	2	3
Jerusalem	806	0	138	56
Ljubljana	35	270	669	26
Pittsburgh	39	5	936	20
Tokyo	36	163	800	1

Table 5: Inferred Strategy Distribution for Sessions in Jerusalem

Distribution of Inferred Strategy Types					
	Strategy Type (z)				No. of Strategies
Split Type	P(t)	P(t-1)	D(t-1)	T	
Number of Splits	0	0	0	0	806
	1	0	0	1	115
	1	0	1	0	5
	1	1	0	0	10
	2	0	0	0	8
	1	0	1	1	1
	1	1	0	1	29
	1	1	1	0	12
	1	2	0	0	5
	2	0	0	1	4
	2	0	1	0	1
	2	1	0	0	3
	3	0	0	0	1

In the Jerusalem treatment zero splits were inferred 80.6% of the time (Table 5). Intuitively this is because there is no level associated with any of the four conditioning variables at which the population majority decision switched from rejection to acceptance. It is known that the overall level of acceptance was higher in these sessions than in the other three, but also that the average proposal was lower. A conjecture is that the proposals were not low enough to trigger a switch–over in the majority decision, i.e., that the proposals did not achieve a low enough level to identify the responder strategies.

The distribution of inferred strategies in Ljubljana is quite different (Table 6). The modal strategy occurred 624 times and is shown in Figure 4. There are two splits, both of which occur on the variable $P(t)$. From the mean number of decisions at the terminal nodes one can see that the strategy is highly accurate whenever the

Table 6: Inferred Strategy Distribution for Sessions in Ljubljana

Distribution of Inferred Strategy Types					
	Strategy Type (z)				No. of Strategies
Split Type	P(t)	P(t-1)	D(t-1)	T	
Number of Splits	0	0	0	0	35
	1	0	0	0	270
	1	0	1	0	3
	1	1	0	0	42
	2	0	0	0	624
	1	1	0	1	7
	1	1	1	0	2
	2	0	1	0	4
	2	1	0	0	5
	3	0	0	0	8

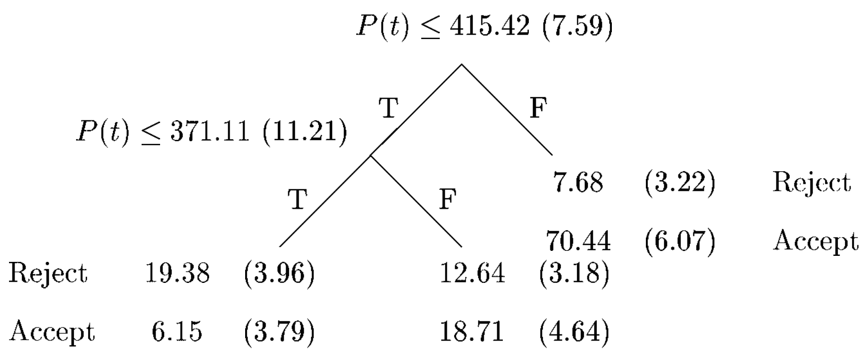

Figure 4: Inferred Modal Strategy for Sessions in Ljubljana

proposal is less than 371 (where it predicts the action reject) or greater than 415 (where it predicts the action accept), but in the interval between 370 and 415 its prediction is not as accurate. This suggests that decision–making is more predictable above and below the two different thresholds than in the interval between them.

There is further evidence for the significance of the modal inferred strategy. The single split strategy with the variable $P(t)$ occurred 270 times, had a mean coefficient of 412.20, and is consistent with the root node of the modal strategy. These two strategies accounted for 894 of the 1000 strategies inferred from the different samples.

The modal strategy in the Pittsburgh treatment occurred 872 times and is nearly identical to the strategy inferred from the data from Ljubljana. Table 7 summarizes

Table 7: Inferred Strategy Distribution for Sessions in Pittsburgh

Distribution of Inferred Strategy Types					
	Strategy Type (z)				No. of Strategies
Split Type	P(t)	P(t-1)	D(t-1)	T	
Number of Splits	0	0	0	0	39
	1	0	0	0	5
	1	0	0	1	7
	1	0	1	0	47
	1	1	0	0	10
	2	0	0	0	872
	1	0	2	0	4
	1	1	0	1	3
	1	1	1	0	2
	2	0	1	0	4
	2	1	0	0	7

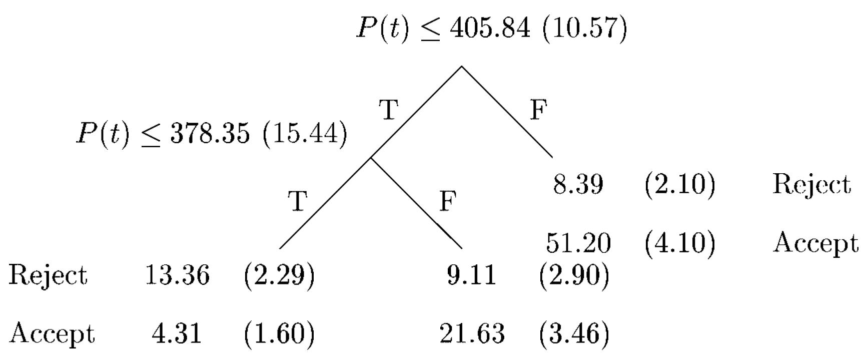

Figure 5: Inferred Modal Strategy for Sessions in Pittsburgh

the strategy distributions and Figure 5 shows the estimate of the modal strategy.

The results from the sessions that were conducted in Tokyo are summarized in Table 8. The modal strategy type occurred 681 times; it includes splits on the variables $P(t)$ and $P(t-1)$. Figure 6 shows the form of this two–split strategy, which initially splits the data along $P(t)$ at a level of 385.47. If the proposal is less than 385.47 then there is another split at the $P(t-1)$ level of 365.69.

For a look at how the strategy assigns decisions, note the number of acceptances and rejections at the terminal nodes in Figure 6. From the first split, if $P(t) > 384.9$,

Table 8: Inferred Strategy Distribution for Sessions in Tokyo

Distribution of Inferred Strategy Types					
	Strategy Type (z)				No. of Strategies
Split Type	P(t)	P(t-1)	D(t-1)	T	
Number of Splits	0	0	0	0	36
	1	0	0	0	163
	1	0	0	1	8
	1	0	1	0	107
	1	1	0	0	681
	2	0	0	0	4
	1	1	1	0	1

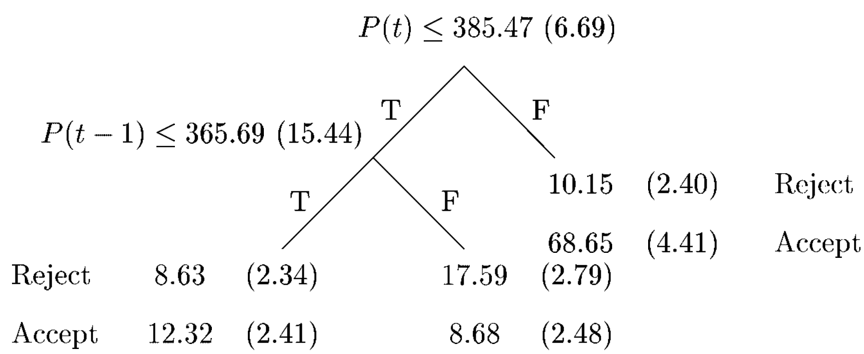

Figure 6: Inferred Modal Strategy for Sessions in Tokyo

the mean number of rejections is 10.15 and the mean number of acceptances is 68.65. At the second split, if $P(t-1) \leq 365.69$, the mean number of rejections is 8.63 and the mean number of acceptances is 12.32. If $P(t-1) > 365.69$ the mean number of rejections is higher than that of acceptances: 17.59 vs. 8.68. Subjects tended to accept proposals above the cut–off level $P(t) = 385.47$, but if the time t proposal was lower than this level, they tended to reject if the previous proposal was relatively high. Subjects in the Tokyo treatment appear to have been conditioning in part on the change in proposal levels from one period to the next. A relatively high proposal in the previous period appears to have created a preference for a relatively high proposal at time t.

Recall from Table 8 that the strategy with a single split on $P(t)$ occurred 163 times. The mean of the $P(t)$ coefficient for this single–split strategy was 383.25.

This is consistent with the root node of the modal strategy reported above, and provides further evidence of the importance of that particular variable and coefficient combination.

To compute the estimate of overall decision strategy accuracy we report the average proportion of misclassification errors on the testing sample for each of the modal strategies. In the sessions that were conducted in Tokyo, Ljubljana, and Pittsburgh this estimated error rate is 0.241, 0.196, and 0.202 respectively. The strategies are able to correctly classify the data in the testing sample (i.e, they are able to predict the correct action taken) roughly 75%–80% of the time.

Summarizing the findings from the data we find evidence of heterogeneity with regard to strategies between treatments. The lack of an inferred strategy in the Jerusalem treatment may reveal as much about proposer behavior as responder behavior: proposals may not have been below the threshold level for rejection. In Ljubljana and in Pittsburgh a high and a low threshold assist in classifying decisions: in the interval between these thresholds decision–making is less predictable. In Tokyo the inferred behavior is dynamic: a relatively high proposal in the previous period appears to increase the probability of rejection in the current period. Most importantly, the classifier approach suggests that decision–making in three of the populations may be characterized by a binary classification tree strategy.

7 Conclusion

We introduced a computational procedure to infer strategies from observed actions and demonstrated it using both simulated and experimental data. Using simulated data we showed that the procedure is capable of uncovering known strategies and showed how to interpret its output under misspecification. We inferred unobserved strategies from the observed actions of subjects in a classic ultimatum game experiment. Results were consistent with existing data analysis: levels were important in the ultimatum game experiments, conditioning levels were different across countries, and responder behavior in two of the countries was similar.

The results provided new information regarding strategic behavior. We did not, for example, infer a strategy from the actions of the responders in the Jerusalem treatment. We can interpret this as a lack of evidence of the existence of a strategy of the type we hypothesized, or as proposal behavior that did not permit the identification of responder strategies. Further research into proposer strategies may clarify this result. In Ljubljana and Pittsburgh we found evidence for behavior that is not predictable within a range of proposals but is predictable outside of the range. Responder behavior in Tokyo was different in that it exhibited a dynamic element by conditioning on level of the proposal in the previous period of the game.

We tested the robustness of the procedure in four ways: (1) by fitting with a training sample and selecting the model with a testing sample, (2) by reporting results from resampling, (3) by comparing the results to inference previously performed on these data, and (4) by testing performance on simulated data. Each of these steps was taken to ensure that we did not infer unmeaningful behavior from the actions of the subjects. In particular we reported an estimate of the distribution of the functional form of the unobserved strategy through resampling. This provided more information regarding the accuracy of the specification than existing procedures that report the single functional form of the best fitting model.

The application of binary classification estimation is not limited to experimental data nor is it limited to population strategies. Any decision making environment where nested if–then statements may characterize repeated decision making may be investigated using the methods in this paper (e.g. federal reserve decisions or strike decisions). An extension to a three–category decision case can be found in Engle–Warnick and Ruffle (2001), and an extension to strategies estimated by linear regression can be found in Duffy and Engle–Warnick (2001b).

The benefits of the procedure are the direct interpretation of the model as strategies, and the generality of the approach which simultaneously and automatically specifies the specific functional form of the model and estimates its coefficients. When combined with existing methods of statistical inference the binary classification estimator can advance the ability of game theory to describe and understand observed behavior.

A Results from Simulated Data

In the Appendix we report on the effectiveness of recovering strategies from simulated data. We show that when the model is properly specified, i.e., when there is a population strategy that generates the data, the procedure accurately recovers it. When there are multiple strategies generating the data we find evidence for the actual conditioning variables and their coefficients, but must take care when interpreting the regression output. Lastly, when agents learn over proposal levels with a common strategy type, we recover the correct strategy type with a relatively high variance estimate for the level.

A.1 Constructing the Strategies for the Simulations

We simulate responder behavior against the actual proposer decisions in the data, hence no distributional assumptions are necessary for the behavior of the proposers, and variance of the actions of the simulated strategy when played against the data is ensured. An error is defined as transitioning down the wrong branch of the tree at an internal node. The distribution of the error process is modelled as a truncated normal density and is shown in Figure 7. The mean of the density is the median proposal level and is estimated from the actual data. The standard deviation of the error density (labeled $\hat{\sigma}$) was selected by finding the 5th and 95th percentiles of the proposals and placing them two standard deviations from the median.[8] The density is truncated at $3\hat{\sigma}$, hence the error under the curve shown in Figure 7 is normalized to 1.

At proposal levels below the median proposal less $3\hat{\sigma}$ no error can occur; at proposal levels between the median proposal less $3\hat{\sigma}$ and the median proposal the probability of committing an error is $F(\cdot)$ (the error density shown in Figure 7); at proposal levels between the median proposal and the median proposal plus $3\hat{\sigma}$ the

[8] Using the estimate of the mean and standard deviation from the data does not qualitatively affect the results. The 5th and 95th percentiles happened to be symmetrical about the median.

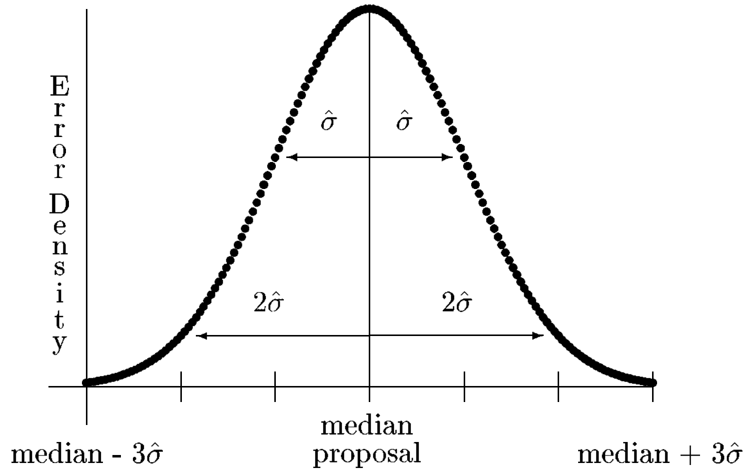

Figure 7: Modelling the Strategy Error Term

probability of committing an error is $1 - F(\cdot)$; at proposal levels above the median proposal plus $3\hat{\sigma}$ the probability of committing an error is zero. Hence the further away the proposal is from the median proposal the lower the probability of deviating from the strategy specification.

A.2 One Responder Strategy: Conditioning on the Time t Proposal

In this sub–section we report results from the simulation of a single responder strategy with a single split that specifies rejecting a proposal if it is less than or equal to 400. The error density is centered on a proposal level of 400 and has a standard deviation of 100.[9] Thus the probability of committing an error is zero whenever proposal levels are above 700 and below 100, and is 0.5 whenever the proposal level is exactly 400. These simulated subjects are more likely to reject lower proposals at time t than higher proposals.

The distribution of the inferred strategies for this simulated data is given in Table 9. We inferred a strategy with one split 993 times (out of 1000 randomly selected samples from the simulated data), a strategy with two splits five times, and a strategy with three splits twice. The table presents the distribution of the strategy types in the inferred strategies. The four numbers in the rows which are

[9] We used the data from the sessions which were run in Japan for the simulations.

Table 9: Inferred Strategy Distribution for Single Strategy Simulation

Distribution of Inferred Strategy Types					
	Strategy Type (z)				No. of Strategies
Split Type	P(t)	P(t-1)	D(t-1)	T	
Number of Splits	1	0	0	0	993
	2	0	0	0	5
	2	0	0	1	2

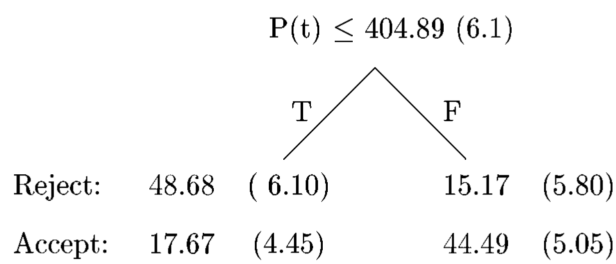

Figure 8: Inferred Modal Strategy for Single Strategy Simulation

labelled "number of splits" make up the type vector z. In this simulation we inferred a strategy with exactly one $P(t)$ split 993 times (first row), a strategy with two $P(t)$ splits five times (second row), and a strategy with one split each containing $P(t)$ and t twice (third row).

For the single split strategy, the mean of the split level was 404.89 with a standard deviation of 6.1. Thus 993 times out of 1000 we inferred a strategy with the correct number of splits and with the correct conditioning variable. Further, the mean of the coefficient was very close to the true coefficient of 400. Figure 8 shows the tree representation of the inferred modal strategy and also reveals that the majority votes of the decisions at the terminal nodes are correct. We conclude that the procedure accurately uncovered the known population strategy.

A.3 Two Responder Strategies: Conditioning on Either the Time t or $t-1$ Proposal

In this sub–section we demonstrate the interpretation of the regression output when there is heterogeneity in the population of decision–makers. We simulated two

Table 10: Inferred Strategy Distribution for Two Strategy Simulation

Distribution of Inferred Strategy Types					
	Strategy Type (z)				No. of Strategies
Split Type	P(t)	P(t-1)	D(t-1)	T	
Number of Splits	1	0	0	0	602
	0	1	0	0	258
	1	1	0	0	111
	0	2	0	0	18
	2	0	0	0	3
	1	2	0	0	4
	2	1	0	0	4

responder strategies, with half of the subjects in the population playing according to the strategy in the previous sub–section, and the other half playing according to a second strategy that contains a single split on the time $t-1$ proposal $P(t-1)$. The error term for the second strategy is centered on a coefficient of 400 with a standard deviation of 100 (i.e., the error parameterization is the same as that for the first strategy).

Table 10 presents the distribution of the inferred strategies. Table 10 reveals that 602 times we inferred a one–split strategy that conditioned on $P(t)$ and 258 times we inferred a one–split strategy that conditioned on $P(t-1)$. In fact every single strategy type contains either one or the other or both of these two variables, which in fact were components of the two data generating strategies.

For the strategy type $\{1,0,0,0\}$ the mean and variance of the $P(t)$ coefficient was 400 (6.47); for the strategy type $\{0,1,0,0\}$ the mean and variance of the $P(t-1)$ coefficient was 408.64 (21.92); for the strategy type $\{1,1,0,0\}$ the means and variances of the $P(t)$ and $P(t-1)$ coefficients were 398.64 (17.52) and 408.74 (23.82). Although the regression is unable to take this type of heterogeneity into account explicitly, the combination of these three strategy types proxied for the two strategies that actually generated the data.

Table 11: Inferred Strategy Distribution for Reinforcement Learning Simulation

Distribution of Inferred Strategies Types					
	Strategy Type (z)				No. of Strategies
Split Type	P(t)	P(t-1)	D(t-1)	T	
Number of Splits	1	0	0	0	963
	2	0	0	0	36
	2	0	0	0	1

A.4 Learning Rule: Learning to Condition on Different Levels

In this sub–section we interpret the regression output when the data are generated by a learning model that has been successful in describing subject behavior in these experiments. The data are generated using reinforcement learning (see Roth and Erev, 1995). In reinforcement learning subjects choose strategies probabilistically with propensities that are updated according to payoffs. Strategies that result in relatively high payoffs are more likely to be repeated. For proposers we take the strategy space to be the proposals 100, 200, 300, 400, 500, 600, 700, 800, and 900 as in Roth and Erev (1995). Responders condition on these same levels and accept a proposal if it is less than or equal to the level and reject it otherwise.[10] All players began the game with uniform (equal) propensities to play each strategy. The agent based computer simulations replicate exactly the conditions in the experimental laboratory.

Results are given in Table 11. The table shows that in 963 cases out of 1000 we inferred a strategy with a single–split on the variable $P(t)$. This is a sensible result because responders were all playing a strategy that matches this description. Figure 9 shows that the mean split level was 463.95 and the standard deviation was 35.51. The estimate of the standard deviation is higher than in the first simulation (with a single population strategy) because different subjects begin the game conditioning on different levels and learned to play using different levels over the ten rounds of the game.

[10] For consistency with Roth and Erev (1995) we do not include the error term in these simulations; subjects play the chosen strategies deterministically.

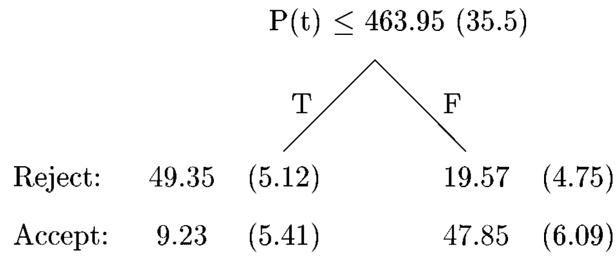

Figure 9: Inferred Modal Strategy for Reinforcement Learning Simulation

References

Breiman, L., Friedman, J., Olshen, R., and Stone, C., 1984. Classification and Regression Trees. Wadsworth International, Belmont CA.

Cooper, J., 1998. Multiple Regimes in U.S. Output Fluctuations. Journal of Business and Economic Statistics 16:92–100.

Costa–Gomes, M., Zauner, K., 2001. Ultimatum Bargaining Behavior in Israel, Japan, Slovenia, and the United States: A Social Utility Analysis. Games and Economic Behavior 34:238–269.

Devroye, L., Gyorfi, L., Lugosi, G., 1996. A Probabilistic Theory of Pattern Recognition. Springer–Verlag, New York.

Duffy, J., Engle–Warnick, J., 2001a. Using Symbolic Regression to Infer Strategies from Experimental Data, forthcoming in: Chen, S. (Ed.), Evolutionary Computation in Economics and Finance, Springer– Verlag.

Duffy, J., Engle–Warnick, J., 2001b. Multiple Regimes in U.S. Monetary Policy? A Nonparametric Approach. Unpublished manuscript.

Durlauf, S., Johnson, P., 1995. Multiple Regimes and Cross-Country Growth Behaviour. Journal of Applied Econometrics 10:365–384.

Efron, B., 1982. The Jacknife, the Bootstrap and Other Resampling Plans. Society for Industrial and Applied Mathematics, Philadelphia, PA.

El–Gamal, M., Grether, D., 1995. Are People Bayesian? Uncovering Behavioral Strategies. Journal of the American Statistical Association 90:1137–1145.

Engle–Warnick, J., Slonim, R., 2001. A Method to Infer Repeated Game Strategies from Actions: Evidence from Trust Games. Unpublished manuscript.

Engle–Warnick, J., Ruffle, B., 2001. Inferring Buyer Strategies to Deterimine the Effect of Buyer Concentration on Pricing. Unpublished manuscript.

Hansen, B., 2000. Sample Splitting and Threshold Estimation. Econometrica 68:575–603.

Nagel, R., 1995. Unravelling in Guessing Games: An Experimental Study. American Economic Review 85:1313–1326.

Roth, A., Erev, I., 1995. Learning in Extensive Form Games: Experimental Data and Simple Dynamic Models in the Intermediate Term. Games and Economic Behavior, Special Issue, Nobel Symposium 8:164-212.

Roth, A., Prasnikar, V., Okuno–Fujiwara, M. Zamir, S., 1991. Bargaining and Market Behavior in Jerusalem, Ljubljana, Pittsburgh, and Tokyo: An Experimental Study. American Economic Review 81:1068–1095.

Selten, R., Buchta, J., 1999. Experimental Sealed Bid First Price Auctions, in: Budescu, D., Erev, I., Zwick, R. (Eds.), Games and Human Behavior, Lawrence Earlbaum, New Jersey.

Stahl, D., Wilson, P., 1995. On Players' Models of Other Players: Theory and Experimental Evidence. Games and Economic Behavior 10:218–254.

Selten, R., Mitzkewitz, M., 1997. Duopoly Strategies Programmed by Experienced Traders. Econometrica 65:517–555.