A hierarchical ensemble of decision trees applied to classifying data from a psychological experiment

Yannick Lallement
Human-Computer Interaction Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
yannick@cmu.edu

Abstract

Classifying by hand complex data coming from psychology experiments can be a long and difficult task, because of the quantity of data to classify and the amount of training it may require. One way to alleviate this problem is to use machine learning techniques. We built a classifier based on decision trees that reproduces the classifying process used by two humans on a sample of data and that learns how to classify unseen data. The automatic classifier proved to be more accurate, more constant and much faster than classification by hand.

Introduction

Classification of complex data coming from psychological experiments is an important issue in cognitive psychology. Such classification is often done by two or more persons who subjectively rate the human behavior. This process can be long and labor-intensive, and there is often too much data to be classified by humans in a reasonable amount of time. Moreover, the psychology community considers the classification acceptable if the inter-rater reliability between the persons is at least 80%, which, given the variability of human data, often demands a lot of training.

We encountered such a problem regarding the classification of human behavior over time. We built a learning classifier to make the classification faster and more reliable. In the following, we describe our data, the classifier we built, and the results we obtained.

The KA-ATC© Task

In the Kanfer-Ackerman Air Traffic Control© task (Ackerman & Kanfer 1994) is used to study problem-solving and learning in a dynamic environment. When performing this task, participants are presented with

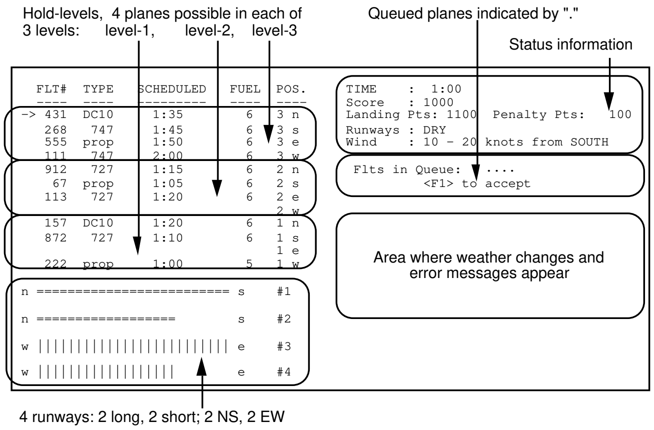

Figure 1: Startup screen of the KA-ATC© task, annotated to show the hold-levels, the runways, and areas where information is given to the participants.

the start-up screen shown in Figure 1. Planes in a hold-pattern in the upper left corner of the screen must be moved down to runways in the lower-left corner before they run out of fuel. The hold-pattern contains 3 hold levels, each of them containing 4 hold rows.

The planes are moved between adjacent hold-levels and from hold-level 1 to the runways using cursor-movement keys. A complex set of rules constrains which planes can land on which runways depending on the wind direction, wind speed, and other weather conditions. As time passes, the planes use up their fuel (indicated in the FUEL column, in minutes until crash), the wind and weather change, and more planes queue up to be admitted to the hold-pattern. Planes are accepted into an empty hold-row from the queue by hitting the F1 key. Points are awarded for landing planes; points are subtracted for crashing planes, landing planes with low fuel, and attempting to move planes to places that violate the rules (e.g., to an already occupied hold-row or to the wrong runway for the current weather conditions). Participants perform

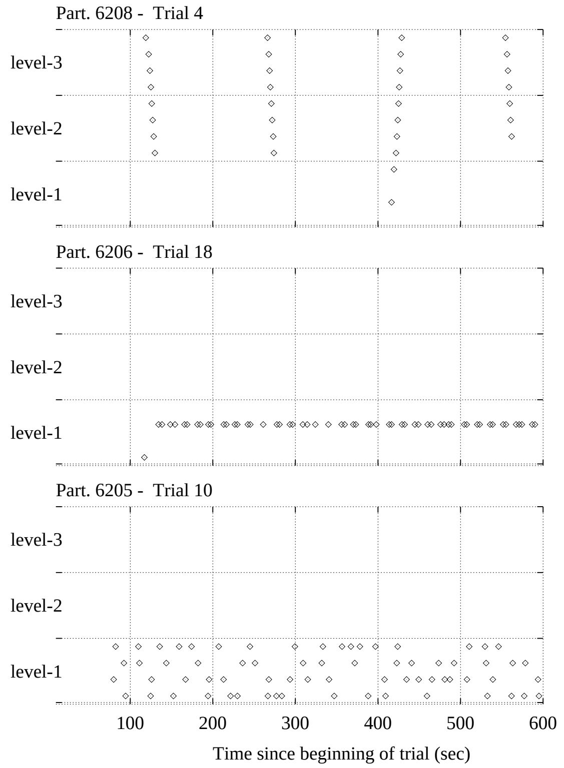

Figure 2: Prototypical strategies: stacked (top), sequential (middle), opportunistic (bottom)

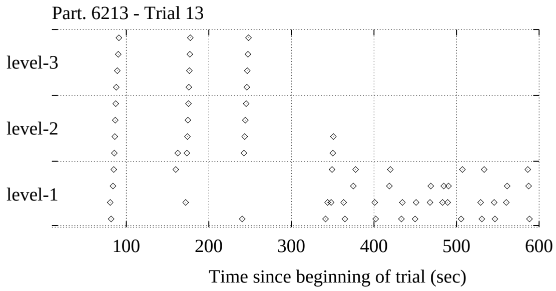

Figure 3: Abrupt shift within a trial.

	NIS	STA	SEQ	OPP
Training set	31.2%	41.2%	18.0%	9.6%
Test set	36.0%	20.3%	0.0%	43.7%

Table 1: Percentage of the quarter-trials in each class for the two sets

the task during 10-minute trials, and are instructed to maximize their score.

Timestamped keystroke data from over 3500 participants are available on a CD-ROM (Ackerman & Kanfer 1994). These data are the basis for learning models that use different AI and cognitive architectures. We are interested in a specific study (study number 2) in which each of the 58 participants performed 24 10-minute trials. Prior to writing a cognitive model of one of those participants, we became interested in a particular aspect of the task: how participants accept planes from the queue into the hold-pattern (John & Lallement 1997). Several strategies can be observed, that we present in the next section.

Queue acceptance strategies

Figure 2 shows for three trials when planes are accepted from the queue and in which hold-row they are brought. Each timeline represents a trials, with seconds since the start of the trial (x-axis) and the 12 hold-rows (y-axis). The three hold-levels are separated by dotted horizontal lines. Each time a plane is brought in from the queue, a dot appears on the timeline at the hold-row in which the plane was accepted.

We are interested in the strategies used for filling and emptying the hold rows. We identified three strategies: *stacked, sequential*, and *opportunistic*. *Stacked* indicates that the participant stacks up a series of planes one right after the other before landing them all and starting over (Figure 2, top). *Sequential* indicates that the participant is attending to one plane at a time, bringing it in to a particular hold-row and then landing it, and starting over (Figure 2, middle). *Opportunistic* indicates that the participant alternates between landing planes and filling in hold-rows whenever the other demands of the task afford an opportunity to do so (Figure 2, bottom).

In order to examine the distribution and the evolution of the strategies in study 2, we needed to identify the strategies used in 1,392 trials (58 participants, 24 trials per participant). Identifying strategies is often difficult; whereas figure 2 presents a prototypical example of each strategy, most of the trials are in fact noisy. Moreover, the strategy can change during a trial (as shown on figure 3), and this change can be abrupt or more gradual. Even if a trial gives a general visual impression that prompts for a specific classification, the precise criteria for classification are difficult to formulate. Several iterations at writing procedures and training researchers to visually classify those data were made, all failing to reach the 80% inter-rater reliability criteria (B. E. John, personal communication, October 1996). The best inter-rater reliability we obtained was 74.3% between the author and a colleague on a set of 424 trials. One of the problems faced by the humans evaluator was a not sufficient intra-rater reliability; a trial labelled a certain way could be labelled a different way at another time by the same person.

We decided to build a classifier that would automatically and consistently classify the trials into one of the three strategies, or no identifiable strategy. Because of

Feature	Type
Number of points in the trial	Integer
Average number of points per non-empty time period	Real
Average number of consecutive points on a same position	Real
Location (1=level 1, 2=l. 2, 3=l. 1&2, 4=l. 3, 5=l. 1&3, 6=l. 2&3, 7=all	Integer
strategy type (0=no strategy, 1=opportunistic, 2=sequential, 3=stacked)	Integer

Table 2: Structure of a training example

the lack of precisely formulated criteria to label the trials, the classifier would have to learn from examples. Automatic labelling seemed also necessary because of the size of the data to classify (1,392 trials for the specific experiment we are currently interested in, and well over potentially 10,000 trials on the CD-ROM).

Training and test sets

To constitute the training set, we labelled by hand 424 trials of an experiment similar to the one we are interested in. Because of the possible variation of strategy during a trial, the trials were labelled according to the following principle: try to identify a strategy for the whole trial. If possible, the whole trial, and therefore its four quarters, receive this strategy as label. If not possible, break the trial into two halves, and repeat the process until the quarter-trial level. If a quarter-trial does not show any identifiable strategy, label it as such. This multi-level classification method reflects a natural thought process; a trial is broken into smaller parts only if it cannot be classified as a whole. This method is more tolerant to noise than a mono-level one (directly classifying the quarter-trials) would be. The trials are almost always noisy, and a certain amount of noise can be more easily disregarded in a whole trial than in a quarter trial. It was therefore important to build a classifier that would replicate this multi-level classification process.

The percentage of agreement between the two labelling persons was 74.3% at the level of quarter trials. The final labelling of the training set was done by agreement or compromise between the two persons. Each quarter-trial received a label among: no identifiable strategy (NIS), stacked (STA), sequential (SEQ), opportunistic (OPP). Despite the non-satisfactory inter-rater reliability, the consensus is more trusted than any of the individual classifications, and it is from this consensus that the classifier will learn. We similarly constituted an independent test set of 48 randomly picked trials of the study we are interested in, in order to test the classifier on previously unseen data.

The frequency of each class in the two sets is given in Table 1. For each set, the base learning rate (pick the most common class) would be slightly above 40%.

Features

The classifier we built is based on decision trees, that seemed adequate for the type of complex temporal data we are handling. Decision trees classify data based on a set of features; the features must be relevant and give enough information about each data point (here, each trial) to make its classification possible. The features we used are:

- The average number of points per non-empty time period: a 600 seconds trial is broken into at most 60 10-second period (the first period starts at the first point in the chart, so there may be less than 60 periods). A higher value for this feature will prompt toward a stacked strategy.
- The average number of consecutive points on a same hold row. A higher value for that feature will prompt toward a sequential strategy.
- The location of the points: level 1 only, level 2 only, etc. If more than 90% of the points are in a certain level, then the feature takes this level as value. This feature is relevant because of a correlation between the strategies and the location; for example, sequential tends to be in level one only, whereas stacked tends to use several levels.
- The total number of points during the trial. A small number of points will prompt toward no identifiable strategy.

A program was written to compute these features for every trial we wanted to classify and every trial in the training and test sets. The structure of a training example as it will be used by the decision trees is given in table 2.

The classifier

Architecture

An initial experiment with a single ID3 (Quinlan 1986) and a single OC-1 (Murthy, Kasif, & Salzberg 1994)

Tree	Training	Pruning	Test
Full trial	175	75	174
Half trial	420	180	248
Quarter trial	840	360	496

Table 3: Number of training, pruning and test examples for each type of tree

Tree set	Average number of leaves	Average depth
Full trial	9.78	4.82
Half trial	22.40	7.96
Quarter trial	31.80	9.26

Table 4: Structure of the trees

decision tree showed that OC-1 tree gave significantly better results on our data. Unlike ID-3, OC-1 divides the data into regions using hyperplanes that are not necessarily parallel to the axis. This initial experiment prompted us to build our classifier using the OC-1 decision tree algorithm.

Ensembles of decision trees have been shown to perform better than single decision trees (Quinlan 1996; Dietterich 1997). Our classifier is an ensemble of OC-1 trees, that replicates the way we labelled the trials. It is composed of three levels, each level being a set of 50 OC-1 trees (one set for the whole trials, one set for the half trials and one set for the quarter trials). At each level, each of the 50 trees proposes a classification; the final decision is the vote of the majority of the trees, accompanied by a confidence rate, given by the percentage of trees that voted for that classification.

Training

Each tree was trained on a random subset of the training set (using the bagging method (Breiman 1994)): the trees of the full trial level were trained using 40% of our 424 hand-labelled trials for training, and 20% for pruning, leaving 40% unused. The trees of the half-trial level were trained using the same proportion of the 848 half-trials, and similarly for the quarter-trial level trees (table 3).

Table 4 shows the average number of leaves and depth for the 50 trees of each category. As expected (because the number of examples grows when trials are divided, and at the same time the examples become more and more noisy), both values raise significantly for smaller sizes of trials.

	Classifier			
Humans	NIS	STA	SEQ	OPP
NIS	22.1%	1.4%	0.1%	0.6%
STA	6.1%	39.7%	0.0%	0.2%
SEQ	0.8%	0.0%	17.9%	0.0%
OPP	2.2%	0.1%	0.0%	8.8%

Table 5: Confusion matrix between the classifier and the humans' consensus, expressed in percentage of the 1696 hand-labelled quarter trials

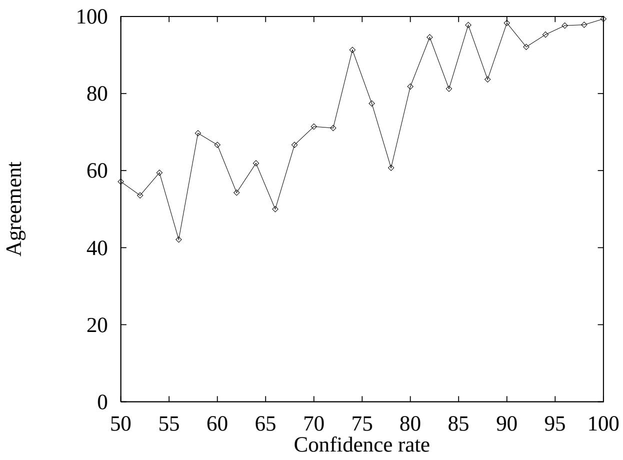

Figure 4: Agreement between the classifier and the humans' consensus *vs* confidence rate of the classifier

Labelling

To label a new trial, the classifier first tried to label the full trial, and, if no strategy was identified, then tried to label its two halves separately. The process repeats until the level of quarter trials. If no strategy was identified at the quarter-trial level, the quarter trial was labelled *no identifiable strategy.*

Results

The confusion matrix (table 5) shows in details how the classification done by the classifier compares to the humans' consensus for the whole set of trials (40% of the data were unknown to each tree). The agreement rate (trial classified the same way by the humans and the classifier) is 88.5% (sum of the diagonal). The rate of false alarms (the classifier assigns a strategy to a trial that was labelled as NIS by the humans) is 2.1%. The rate of misses (the classifier labels as NIS a trial that was assigned a strategy by the humans) is 9.1%. The rate of confusion between two identifiable strategies is 0.3%. The classifier is therefore conservative (false alarms are less common than misses) and very seldom confuses two identifiable strategies; both these results are suitable for our data analysis.

We also tested the classifier on the previously unseen 48 examples of the test set; the final agreement rate between the classifier and the humans' consensus was 85.4% at the level of quarter trials.

The confidence rate is related to the accuracy of the classification, as shown in figure 4. The average confidence rate of the classifier was 90.5% for the agreement cases, and only 68% for the disagreement cases. For confidence rates over 90%, the agreement was over 90%. The extra information provided by the confidence rate is useful: a low confidence rate for a trial will prompt for a double-check of this trial by a human evaluator.

Conclusion

These results, well above the human inter-rater reliability of 74.3%, show that this hierarchical ensemble of decision trees was able to replicate the multi-level process of labelling and to learn the unformulated criteria we used to label the training and test sets. The labels attributed are consistent with the humans' consensus, that we trust more than a single human's classification. Of course, the classifier is consistent and will always attribute the same label to a given trial. Finally, the supplementary confidence rate information allowed us to take a closer look at the trials labelled with low confidence.

It took the two trained humans hours to rate 424 trials; it takes seconds for the classifier to rate our 1,392 trials, with a better reliability. Such classification methods using machine learning techniques prove to be a valuable tool for cognitive psychology: evaluation of large quantities of fuzzy data that was intractable by psychologists before is now accessible.

Acknowledgments

Many thanks to Phil Ackerman and Ruth Kanfer for making their task and data available through the Office of Naval Research (ONR), and to Bonnie John, Tom Mitchell and Tom Dietterich for their help. This research is supported by the ONR, Cognitive Science Program, Contract Number N00014-89-J-1975N158. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ONR or the US Government.

References

Ackerman, P. L., and Kanfer, R. 1994. Kanfer-ackerman air traffic controller task cd rom database, data-collection program, and playback program. Technical report, University of Minnesota, Department of Psychology.

Breiman, L. 1994. Bagging predictors. Technical Report 421, Department of statistics, University of California, Berkeley.

Dietterich, T. G. 1997. Machine learning research: Four current directions. *AI Magazine* 18(4).

John, B. E., and Lallement, Y. 1997. Strategy use while learning to perform the kanfer-ackerman air traffic controller task. In *Proceedings of the nineteen annual Cognitive Science Society Conference*.

Murthy, S. K.; Kasif, S.; and Salzberg, S. 1994. A system for induction of oblique decision trees. *Journal of artificial intelligence research* 2:1–32.

Quinlan, J. R. 1986. Induction of decision trees. *Machine learning* 1(1):81–106.

Quinlan, J. R. 1996. Bagging, boosting and c4.5. In *Thirteenth National Conference on Artificial Intelligence*. Portland, Oregon: AAI Press / MIT Press.