Automated Clock Drawing Test through Machine Learning and Geometric Analysis

Anupam Guha, Hyungsin Kim, Ellen Yi-Luen Do
College of Computing and GVU Center
Georgia Institute of Technology
Atlanta, Georgia
{aguha7, hyungsin, ellendo}@gatech.edu

***Abstract*—In this paper, we discuss the challenges of sketch recognition accuracy and automation of the Clock Drawing Test (CDT). Sketch recognition in the context of the CDT is a complex problem due to the lack of knowledge of the preference bias among the sketches drawn by neuro-atypical patients. However, machine learning provides a viable solution to detect measurable patterns among sketches drawn in the CDT. The paper posits that these sketches are constrained, and the majority of the recognition work can be completed by doing handwritten digit recognition. All other properties to be measured in the CDT are primarily geometric in nature. The paper explores various machine learning classification techniques for handwritten digit recognition. Finally, based on analysis, we present the best one in context of CDT.**

Keywords: Clock Drawing Test, Machine Learning, Geometric Analysis, pattern recognition

INTRODUCTION

The Clock Drawing Test (CDT) is an instrument to screen people with cognitive impairment and dementia [1]. Generally, CDT is administered manually in a hospital environment in which the test subject is instructed to draw the face of a clock, and the hands indicating a specific time (e.g., 10 past 11) [2]. A variation of CDT asks the subject to copy a drawn figure of a clock. The difficulties in developing automated CDT software lies in the amount of variances in the predicted results due to the neuro-atypical nature of the patients involved. Not only is it difficult to predict with any degree of accuracy how much the errors on parts of the subject alter the sketches to be recognized from the expected outcomes but also it is hard to distinguish situational drawing errors from genuine symptoms of cognitive debility.

In this paper, we present a system that attempts to address these challenges and automates some if not all of the processes that occur in the conduction of the CDT. It becomes somewhat easier to address the problem if it is broken down into subparts. Each has multiple potential solutions and then we use build and fix to test each one of them. The computational model of the CDT has several exploitable constraints which shortens our task. First we chose a rubric for scoring once the clock face has been drawn. The system has to generate a score out of 13, each point awarded for a factor drawn correctly. These factors are:

1. Only numbers 1-12 are present without additions or omissions
2. Only Arabic numerals are used
3. Numbers are in the correct order.
4. Numbers are in approximately correct positions
5. Numbers are inside the circle
6. Only two hands are present
7. The hour target number is indicated
8. The minute target number is indicated
9. The hands are in correct position
10. There are no superfluous markings
11. The hands are relatively joined
12. A centre of the circle is present or inferred at the joining of the hands

Looking at this scoring rubric one fact emerges. Recognition of handwritten digits is the key. If all the digits are correctly identified it becomes easier to process the other sketches, like that of the hands, where they are pointing to, etc.

HANDWRITTEN DIGIT RECOGNITION

We posit that handwritten digit recognition provides a solution to the automation of CDT. For one it is a fairly robust and tested paradigm [4], which is being currently developed for various other applications [3] using a lot of different techniques. If the digits are determined to be all present and accounted for (on conversely unidentifiable) then more than half of the work of sketch recognition in CDT is over. Also even if the recognition is not completely accurate it does not alter the score much. All that needs to be decided is that most of the sketches are Arabic digits and their locations. The first task of the system is to identify the discrete sketches drawn in the clock-face. Counter intuitively this proves to be a non-trivial task because the patient undergoing the test may not draw every digit neatly away from each other (See Figure 1). Often the patient would join adjoining sketches like clock hands with digits. While this is a small error from the human point of view and presents no problem to the medical examiner it may play havoc with an Machine Learning (ML) system. Conversely the patient may choose to draw the sketches of the digits in such a manner that the digits themselves are drawn with separations which ought not to occur (Figure 2).

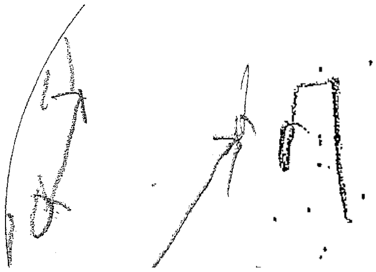

Figure 1. Cases of patients smudging two discrete sketches

The data collected for analysis of such anomalies was taken from a set of 50 CDTs obtained from the tests conducted by the Emory's Alzheimer Research Center on a random sample set patients suffering from mild cognitive impairment due to aging. The results were:

1. Three cases of sketches being smudged together – for example, hands overlapping or connecting the digits as shown in Figure 1.
2. Eight cases of sketches being discrete without cause – for example, digits drawn with space between different strokes in Figure 2

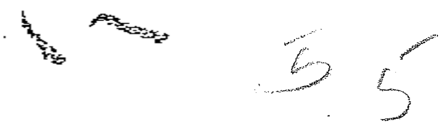

Figure 2. Cases of patients breaking a continuous sketch

Considering that 50 CDTs yield over 700 discrete sketches these are fairly low numbers. Patients undergoing treatment for cognitive impairment somehow are able to maintain digit integrity. Analyzing the CDTs we came to a conclusion that sketch integrity is fairly stable across a spectrum of patients and there is a justifiable case for doing machine learning classification techniques on them after some elementary noise cleaning of the CDT images.

The second test we conducted was to identify which machine learning classification algorithm worked best when the data to be classified was handwritten digits made by neuro-atypical patients. The goal was not to test for algorithms which are considered state of the art for handwritten digit recognition but to use algorithms which can be used easily to determine the suitability of other algorithms based on the way they work and which are relatively fast ('lazy learning' like kNN) as we wanted to see what, if any, differences are there from 'normal' digits drawn by neurotypical patients besides extracting features which were 'interesting'. It was quite possible that the handwritten digit data here would be sufficiently different from that of a standard dataset and it was interesting to find out if so then where. It was not expected that these general machine learning algorithms will perform as well as those specifically designed for handwritten digit recognition but absolute accuracy was not the goal here, rather comparisons and identifying useful features to be extracted was. The results obtained are different from those obtained by standard datasets like MNIST (http://yann.lecun.com/exdb/mnist/). We have a dataset of handwritten digits that have already been identified for the purpose of recognizing the sketches. The dataset was obtained from the MNIST dataset of which the first 5000 instances were taken as training data. The algorithms used were kNN, Multilayer Perceptron and Decision Trees.

Feature Extraction

To test for which algorithm would prove to be the best, the test set was created out of the digits obtained from the 50 datasets. Each was resized into a 16 x 16 pixel binary image. If all the pixels were to be taken as features then each sketch was represented as an array of 256 '0's and '1's. Also a second dataset was created by taking these images, and taking the horizontal, vertical, right diagonal and left diagonal histograms. The histograms will be arrays of 16, 16, 31, and 31 integers, concatenating them would give an array of 94 integers which were used to represent the sketches.

Pruning	Train%	Test%
---	100%	59.5%
minimum leaf instance=2 stop splitting nodes with less instances than=2	90.3%	58.6%
minimum leaf instance=4 stop splitting nodes with less instances than=4	80.4%	54.8%
post pruning m=2	79.9%	56.1%

Table 1: Decision Trees Digit-Raw

Table 1 displays the results for decision trees on the raw data with no feature extraction. The results are not very encouraging and even a little amount of 'pruning' (lossy compression of the trees) results in further degradation. When the same algorithms have been used for 'normal samples' from the MNIST dataset with the same algorithms for the same settings the accuracy has crossed 95% in kNN as well as neural networks for digit-hist and 90% for digit raw and a little above 72% for unpruned decision trees for digit hist and 65% for digit raw. Accuracy increases in Digit-Hist (Table 2) which uses histograms as features compared to Digit-Raw. We believe this is due to the fact aggregation of attributes rather than taking them individually smoothens out the errors and decreases 'overfitting' (which means that the algorithm trusts faulty data too much and takes into account each little deviation in the graph and misclassifies when test data doesn't follow those deviations exactly). We believe some of the damage done by overfitting is slightly improved by post pruning as post pruning smoothens out the small deviations.

This behavior is consistent on both Digit datasets. It is evident that decision trees are not a viable option for such noisy data.

Pruning	Train%	Test%
---	99.98%	63.1%
minimum leaf instance=2 stop splitting nodes with less instances than=2	89.1%	62.4%
minimum leaf instance=4 stop splitting nodes with less instances than=4	78.6%	61%
post pruning m=2	78.1%	61.4%

Table 2: Decision TreesDigit-Hist

Table 3 and 4 display the results for kNN. A drastic improve in accuracy is observed. This has to do with the properties of the datasets and the way kNN works. kNN is extremely suitable for handwritten digit recognition as we are taking into consideration all attributes simultaneously, and errors are more likely to get smoothened out. 'Lazy' machine learning (machine learning simply involving storing and matching like kNN, no 'training' involved) makes a lot of sense for classification where there is a pretty straightforward feature extraction from images. It takes a lot less time than, say, neural networks while giving comparable accuracy. However there is not much scope for improvement, once a maximum has been reached.

Nearest Neighbours	Train%	Test%
1	99.9	85.6
5	99.9	89.4
10	99.9	85.8
20	99.9	85.8
50	99.9	84.9

Table 3: k-Nearest NeighboursDigit-Raw

Nearest Neighbours	Train%	Test%
1	99.9	88.7
5	99.9	89.1
10	99.9	89.6
20	99.9	89.5
50	99.9	88.4

Table 4: k-Nearest NeighboursDigit-Hist

Table 5 displays the results of the perceptrons on the two datasets. We trained the neural networks for different number of epochs. A very surprising result is observable. In the two digit datasets, doubling the epochs lead to a slight decrease in accuracy. This may be due to overfitting. Also unlike the last two algorithms in neural network, Digit-Raw would actually give a better result than Digit-Hist. The reason is simple, while in other classification algorithms, presence of too many attributes becomes a hindrance at times, and prevents smoothening over small deviations, in neural networks; it means a large input network, which is almost always good news. Also the input network in Digit-Raw is binary, whereas that in Digit-Hist is numerical. The same is observed with the 'normal digits' but with accuracy crossing 95%. Evidently there is some degree of 'sameness' inside the MNIST digits which is fundamentally different from the digits obtained from the CDT. Whether this difference is the result of the neuroatypiccal patients drawing the digits or whether this is due to the test not done in conditions the MNIST digits were collected is a discussion beyond this paper. We observe that the digits are different; why so is something we will not speculate on.

Digit-Raw	
256 input layer, 50 hidden layer, 10 output layer	
500 epochs	
Correctly Classified Instances	87.1 %
1000 epochs	
Correctly Classified Instances	86.8 %
1500 epochs	
Correctly Classified Instances	86.5 %
2000 epochs	
Correctly Classified Instances	86.4 %
Digit-Hist	
94 input layer, 20 hidden layer, 10 output layer	
500 epochs	
Correctly Classified Instances	84.5 %
1000 epochs	
Correctly Classified Instances	84.3 %
1500 epochs	
Correctly Classified Instances	84.1 %
2000 epochs	
Correctly Classified Instances	84 %

Table 5: Multilayer Perceptron

As can be observed it proved that the decision tree algorithm was not adequate at all in identifying accurately the sketches made by the patients in the CDT. Though the perceptrons proved most accurate they were incredibly slow. We posit that for the wants of the CDT the kNN algorithm suffices as it is sufficiently accurate as well as quite fast. These results advocate the necessity of large scale collection of digits drawn by neuroatypical patients and comparing them with the 'standard' datasets to observe repeatable patterns of differences. As our work here was to develop a system which does automated CDT, The best algorithm which is reasonably fast and can be used for the CDT system is kNN with 10 neighbors using the histograms as the features.

COMPONENTS OF THE SYSTEM

The first order of business of the system under development is to extract the individual blobs. Each blob is a sketch segment from the drawing of the clock face. The system uses MATLAB® as its platform. It uses the connected component functionality to discover these sketches. The idea is that each sketch component is a sequential connected group of pixels removed from any other sketch segment. Thus extraction of each ‘blob’ as a separate image is possible where each is a separate sketch. In this manner the entire drawing is broken up into fragments, each is resized and cleaned. The larger sketches are hypothesized to be the hands and stored separately. ‘Larger’ is defined as the sketches whose major axis is longer by 100% of the average length.

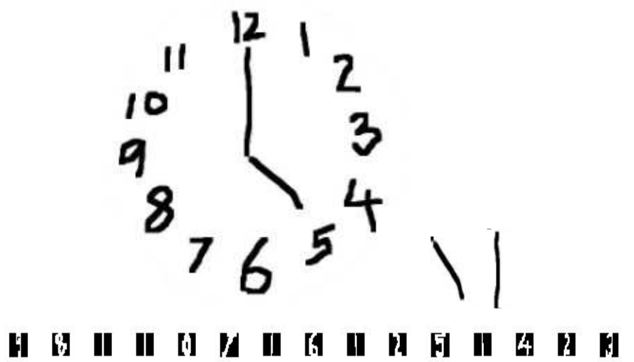

Figure 3. Clock face being broken up into discrete components like hands and digits

After this is done those digits are identified which are ‘1’ using kNN. The ‘1’s are important as there are five ‘1’s. Then each digit is identified. The ‘1’s which are in close proximity with a ‘0’, a ‘1’ and a ‘2’ (‘proximity’ defined as Euclidean distance between centroids being 50% lesser than average) are designated as 10, 11 and 12. Now the system knows whether there all the 12 numbers are present or not and where their centroids are. This solves about half of the CDT. The general observation was that the patients are most prone to errors like skipping a few digits, or drawing them at faulty distances or confusing the centre of the clock face.

Since the system assumes the larger sketches to be the hands, the lengths of their major axes (measured by functionality provided in MATLAB®) are known which is used to identify the minute and the hour hand (shorter). The system identifies the angles of the hands in relation to the horizontal axis to determine the pointing directions. Also it is easy to check for superfluous markings, extra hands, and position of centre. The ‘normal’ position of centre is defined as anywhere within 20% distance of the radius from the centre. Any more deviation would not look like the clock centre to the human eye as well. Of course it may be possible that the patient does not draw (or imply) the centre at all. But since the major axes of hands are known it is possible to measure whether they will converge within that 20% from centre distance.

Thus the entire CDT is evaluated. Of course to process these sketches a lot of assumptions are being made. Firstly, we concede that ML doesn’t seem robust enough. Secondly, the hands may be so short they get confused for digits. Thirdly the major axes of the hands might not be the direction the patient was intending to point to, fourthly there are chances that 10, 11 and 12 are identified incorrectly (both false positives and negatives). Some smudges may be construed as superfluous markings. And there is no real way to test point 2 that only Arabic numerals are present. If the patient uses roman (or any other numeral system) the system will diagnose those as superfluous markings and not as numerals. Similarly some patients have been known to write the words for the numbers rather than draw them.

ANALYSIS AND CONCLUSION

Even if this system will almost never give a highly accurate score for each patient, it will be useful to detect crests and troughs to measure the progress (or otherwise) of a large number of patients easing the load on the medical officers. The handwritten digit detection part almost reaches 90% accuracy. What this system displays is that in a constrained environment it is easier to do sketch recognition. Also a lot of extra information is being generated which might prove useful for the medical investigators who use the CDT. In the CDT the sketches are either digits or hands which can be handled by machine learning techniques and some geometrical measurements. We state that it is the context of the sketch which plays a vital role in determining what techniques should be used to analyze those sketches because the term ‘sketch’ may cover a wide spectrum. To do any meaningful sort of analysis of sketches, we need to have a preference bias, something which is a well known rule of thumb in machine learning. Without an a priori preference bias it is difficult to determine what will work before truing it out which becomes intractable. Also when dynamic recognition (pen motion) is not essential, machine learning may be used to identify sketches taking them as final images and converting sketch recognition into a classification problem. We posit that knowing features such as motion points would make recognition even more accurate.

REFERENCES

[1] Steenland NK, Auman CM, Patel PM, et al. Development of a rapid screening instrument for mild cognitive impairment and undiagnosed dementia. Journal of Alzheimer's disease: JAD. 2008;15(3):419-27.

[2] Strauss E, Sherman EM, Spreen O. A Compendium of Neuropsychological Tests: Administration, Norms, and Commentary. Oxford University Press, USA; 2006

[3] FAISAL TEHSEEN SHAH, KAMRAN YOUSAF (2007)*Handwritten Digit Recognition Using Image Processing and Neural Networks* Proceedings of the World Congress on Engineering Vol I WCE 2007, July 2 - 4, 2007, London, U.K.

[4] CHENG-LIN LIU, KAZUKI NAKASHIMA, HIROSHI SAKO, HIROMICHI FUJISAWA, *Handwritten Digit Recognition Using State-of-the-Art Techniques* 2002 IEEE