# COACH - Cumulative Online Algorithm for Classification of Handwriting Deficiencies

**Ariella Richardson** and **Sarit Kraus**
Department of Computer Science
Bar-Ilan University
Ramat-Gan, Israel
{richara,sarit}@cs.biu.ac.il
Tel:(972) 8-971-9268, Tel:(972) 3-531-8762

**Patrice L. Weiss** and **Sara Rosenblum**
Department of Occupational Therapy
Faculty of Social Welfare & Health Studies
University of Haifa
Haifa, Israel
{rosens,tamar}@research.haifa.ac.il
Tel: 972-4-8240474, (972) 4-8249955

**Abstract**

In this paper we present COACH - a Cumulative Online Algorithm for Classification of Handwriting deficiencies. A description of our algorithm along with a performance evaluation of COACH on real data is provided. COACH is an innovative algorithm designed for building an online handwriting evaluation tool to be used for classifying and remediating handwriting deficiencies. COACH adapts learning and data mining techniques from AI to handwriting deficiency classification in an innovative fashion. Until now handwriting classification has been performed manually by trained therapists causing expensive and subjective evaluation. This application lowers the cost of evaluation, increases objectiveness, and enables repeated testing that can accompany therapy. COACH is evaluated on real data obtained from children with poor handwriting using a digitizer tablet. Results show that COACH manages to successfully differentiate between poor to proficient handwriting. Differentiation is obtained even after using data from only a few words. These results prove that COACH is a promising emerging application for online evaluation.

## Introduction

Many people suffer from handwriting deficiencies of different kinds. These deficiencies can be of various origins and have many characterizations. The number of people with problems such as these is increasing all the time. The diagnosis of such problems is usually performed by trained occupational therapists using a set of Handwriting Evaluation tests such as described in (Erez & Parush 1999). There are many problems with this type of testing. The tests are limited to characteristics of the writing observable by humans. The testing is subjective and if performed by an unexperienced therapist may be wrong. Testing is very time consuming and expensive since it requires professional evaluation and therefore is usually only performed once for diagnosis. An application that provides diagnosis would lower costs of evaluation, provide support to unexperienced therapists, enable re-evaluation throughout therapy to test for improvement and is therefore an important contribution to this domain. Another problematic aspect of existing evaluation techniques is their inability to use information hidden from the human eye. This includes the pen's pressure used when writing, or the pen's tilt and azimuth. This data can add insight to what causes the difficulty and how to intervene in order to improve handwriting. Online data from the handwriting process are collected using a digitizing tablet and instrumented pen. These data provide much information on the handwriting process, but are complex to analyze since they have multiple attributes and are collected over time. We do not have enough expert information concerning the relative importance of the various attributes so we must look at all of them. COACH uses learning and data mining from AI and applies them in an innovative fashion to handwriting deficiency classification.

This paper presents COACH (a Cumulative Online Algorithm for Classification of Handwriting deficiencies) which is an online innovative classification algorithm that provides the user with immediate feedback on handwriting. COACH can be used as a diagnostic tool for subjects. In addition COACH can be used to test various handwriting interventions by the therapist or to practice alone after the correct intervention is found. The algorithm uses pressure, tilt and inAir (the time that the pen is not in contact with the surface) and can provide details on the proficiency of a writer for each attribute. COACH is trained on data collected from various writers both proficient and poor and can provide an online evaluation of new handwriting samples.

The problem of handwriting recognition is known to be a difficult task and much research has been conducted in this domain e.g. (Bahlmann 2006). However previous research has not addressed the special characteristics of handwriting belonging to writers with various deficiencies. Most handwriting studies have been made on proficient writers and it is obvious even to the naked eye that the writing from deficient writers is shaped differently to proficient writing. It therefore seems that standard techniques are not applicable. Some exploration of handwriting deficiencies using computerized methods has already been done for example by Rosenblum, Parush, and Weiss (Rosenblum, Parush, & Weiss 2003); one of the unique issues in the present study is the cooperation between disciplines. The need for creating a handwriting classification tool evolved within the occupational therapy (OT) community. This interaction between OT and computer science provided us not only with real data, but also with focus on which points are of interest in this domain. Finding the attributes that contribute to good classification provides insight into understanding the mechanisms of poor handwriting. It seems that the OT community is ready to embrace new technologies to assist with the diagnosis of various conditions. COACH demonstrates potential for AI integration in OT. It shows how AI techniques can contribute to solving OT problems and why this cooperation seems promising.

In our research AI learning is applied to a new domain. One of the challenging aspects of our research was the structure of the data. The data used is real data collected by occupational therapists for their ongoing handwriting research. This data was not collected specifically for our research. The evaluation of COACH on real data will allow successful integration into a deployed system in the future. In order to use the data with AI learning mechanisms we needed to adapt the data. This was a difficult task because of the nature of the data. The data is in the form of multivariate time series. Most classification algorithms are not capable of directly classifying multivariate time series. Multivariate data that are not dependent on time such as a vector of information gathered on a subject (age, sex, height, temperature) are easy to classify (Witten & Frank 2005). Time series with a single attribute, e.g. temperature at different time intervals $\{temp(t_1), temp(t_2), ..., temp(t_n)\}$ can also be classified using available classifiers, for example Morabito and Versaci (Morabito & Versaci 2003). However, multivariate time series are difficult to classify. Using common solutions (Dash & Liu 1997) it is possible to choose a single attribute from among the set of attributes and use it for classification. However, this results in the loss of the information found in the other attributes. Alternatively, one can look at multiple attributes in separate time intervals, but then data from other time intervals are lost. In both cases the process could be repeated for each attribute or time interval and then the classifications could be integrated. Another approach is to integrate data over time for each attribute, for example using averages and standard deviations on the whole time series for each attribute and creating a vector of the results for each subject (Baxter, Williams, & He 2001). We chose to separate the attributes and create time series for each attribute. This is performed for various time intervals in order to simulate online behavior. These time series are classified and the integration of the results for all the attributes and time intervals are combined to provide the current classification.

One of the difficulties encountered during the adaptation of the classification algorithms to handwriting was the choice of the unit of text that was best suited for analyzing. In order to discuss this let us define the term stroke. *A stroke* is a continuous line that the subject draws on the paper. Some letters are typically composed of multiple strokes - such as the letter 'K' and others composed of a single stroke such as 'O'. Poor writers however, tend to break each letter into more strokes. In contrary to our initial assumption that letters (or even words) would be used as a basic text unit for handwriting deficiencies we discovered that our analysis can be based on strokes. The ability to extract information from strokes rather than letters is very important as it saves the need to use a text identification algorithm (Bahlmann 2006). This may be important when dealing with poor writers who tend to form letters in unconventional ways. Using strokes also means that the algorithm can be applied to character sets from many languages. We found that it is possible to perform classification after a very small amount of writing and this contributes to our ability to build an online system.

COACH may provide insight to many disabilities. Alzheimer's disease, multiple sclerosis, and other diseases affect handwriting abilities and it would be interesting to study these diseases through handwriting. Understanding handwriting mechanisms may provide a vantage point for understanding the broader area of all motor control.

COACH is also applicable to anomaly detection. Anomalies are observed events which deviate from what is expected. When sensors are used to measure the behavior of the system, it is easy to report problems when a sensor measurement exceeds a defined threshold. However in the case that this threshold is unknown, or some other unknown collective sensor behavior causes trouble it is not easy to find the anomaly. A similar process to the one described in COACH can be used to learn to classify the anomalous behavior. This opens a broad array of possible applications for example: evaluating robot performance and accessing security in computer systems.

The rest of this paper is organized as follows: We first present related work. Then define the formal classification problem. This is followed by a description of our approach to the solution. The COACH algorithm is described in detail together with experimental results for two handwriting datasets. We show that COACH can differentiate between poor and proficient handwriting after a small amount of writing. The paper concludes with a discussion and suggestions for future studies.

## Related Work

Much research has been carried out in the area of handwriting recognition. In Bharath et. al (Bharath, Deepu, & Madhvanath 2005) an online clustering algorithm for handwriting recognition is presented. They cluster strokes to identify writing styles, and then perform recognition for each style. Another handwriting recognition project is *frog on hand* by Bahlmann (Bahlmann 2006). *Frog on hand* is an online handwriting recognition project that uses digital pen letter recognition. However we are not interested in handwriting recognition; it is not only the shaping of the letters but also how they are formed by various writers that interests us.

Kalera et. al (Kalera, Srihari, & Xu 2004) describe a signature verification and identification algorithm for an offline environment. They attempt to differentiate between an original signature and a forgery. However, signatures are different to other handwriting as shown in (Bouletreau *et al.* 1998).

The algorithms described mainly use the data on pen position and ignore other aspects such as pen pressure and time. These aspects are important for diagnosis of handwriting deficiencies. Another drawback of these systems is that they are developed for proficient writers. Their performance for poor writers is unclear. The reliability on letter shapes for identification may be problematic since many deficient writers do not form letters in a regular fashion, this being the nature of their disability.

Work on analyzing handwriting data collected from a digitizer offline has been done by Rosenblum, Parush, and Weiss (Rosenblum, Parush, & Weiss 2003), (Rosenblum, Weiss, & Parush 2003). In their work they discovered the inAir phenomenon (the time spent with the pen in the air between strokes) where poor writers tend to spend more time than proficient writers moving the pencil in the air between strokes written on the paper. This important discovery provides motivation to further investigate the data collected using such a digitizer.

Research on data with large numbers of attributes does not contribute to the solution of our problem since it lacks the information regarding time. Some examples include Zhu et. al (Zhu *et al.* 2007) and Liu et. al (Liu *et al.* 2007).

Baxter et al. (Baxter, Williams, & He 2001) solve a multivariate time series problem. Each attribute is a medical test performed at various time intervals. They capture the behavior of a specific attribute (medical test) into a single feature that is used for classification. Kadous and Sammut (Kadous & Sammut 2004) use metafeatures to convert each series into several features and then perform classification. This may result in loss of information. We actually tried compressing each time series into a single value before classification. However better results were achieved using all data (more details follow).

Zaki and Lesh (Zaki, Lesh, & Ogihara 2000) use multivariate subsequences for failure detection in planning. They collect subsequences that are typical of successful and failed plans. Subsequences that appear only in failed plans are used for failure detection. This seems to be beneficial, however it seems that the pruning process can be refined and we plan to explore these expansions in the future. Liu and Liu (Liu & Liu 2002) perform multivariate classification by extending the naive Bayesian classifier and decision trees to suit temporal prediction.

## Problem Definition

Let $M$ be a set of labeled matrices, one matrix per subject, where each matrix $m_i^x$ is of the form:

$$m_i^x = \left\{ \begin{array}{cccc} a_{t_1}^1 & a_{t_1}^2 & \dots & a_{t_1}^n \\ a_{t_2}^1 & a_{t_2}^2 & \dots & a_{t_2}^n \\ . & . & \dots & . \\ a_{t_l}^1 & a_{t_l}^2 & \dots & a_{t_l}^n \end{array} \right\}$$

- $a_{t_j}^o$ is the value of attribute $a^o$ sampled at time $t_j$.
- all subjects have the same number and type of attributes $a^1$ to $a^n$ but the number of time samples $t_l$ can vary for different subjects.
- $x$ is the class label of the subject, $X_1$, $X_2$ ... $X_K$

A model is built using the labeled matrices in $M$. Given a new matrix $m'$ we want to know for which $k$ $m' \in X_k$, $1 < k < K$. It would also be beneficial to obtain some information on which attributes $a^i$ contribute or affect the classification.

For example: in our domain $M$ is a set of matrices of data collected on 9 yr old children, where each matrix $m_i$ is the data from one subject. The subjects are labeled according to their handwriting abilities, $X1 = \{\text{proficient}\}$, $X2 = \{\text{poor}\}$. There are $n = 3$ attributes $a^1$ =pressure, $a^2$ =tilt, $a^3$ =inAir time. The data is sampled at times $t_1 = 0$, $t_2 = 0.01$ to $t_l = 10$ Given data from a new subject $m'$ we would like to determine whether $m' \in X1$ (proficient) or $m' \in X2$ (poor).

## Proposed Solution

The task of classifying handwriting consisted of several stages. First we analyzed the data offline. This analysis determined the units of data used for classification, the classifier used, and the manipulations made on the data. Then we tested our algorithm. Once preliminary results were obtained we decided on heuristic improvements to be made to the algorithm and tested it again. Details on how this was done are presented below.

The first task was to determine which units of the handwriting need to be used. Initially it seemed natural to use letters as a basic unit for model building and classification. Letters have different lengths and shapes. Our assumption was that it would be meaningless to compare a long complicated letter to a short easy one since deficient writers are expected to have more trouble with complex letters. However segmenting the data into letters is a non-trivial research issue (Bharath, Deepu, & Madhvanath 2005), (Bahlmann 2006), (Kalera, Srihari, & Xu 2004). Furthermore we have no interest in recognizing the letters but rather want to uncover characteristics of the handwriting style or deficiency. Therefore we decided to use strokes (a continuous line drawn without lifting the pen). Using strokes is very helpful as it avoids the need to segment the data into letters; it is easy to extract strokes. Strokes may be used for many languages or perhaps even drawing.

We proceeded to choose a classifier. We decided to perform classification of a single attribute time series, for each attribute, and then integrate the results. Classification of the single attribute time series was performed using WEKA (Witten & Frank 2005), a collection of machine learning algorithms for data mining tasks. COACH uses Decision Trees (C4.5) (Quinlan 1993) for classification, with Leave-one-out cross-validation for evaluation.

After selecting the data units (strokes) and the classifier (C4.5) we explored the option of processing the data before performing the classification. We performed the classification both on the data in its raw form and also used various manipulations of the data. This is similar to (Baxter, Williams, & He 2001) and (Kadous & Sammut 2004). Our manipulations involved taking derivatives of the data. We also used means and standard deviations on the whole time-series. The conclusion of these preliminary experiments was that it is best not to perform any manipulations on the data.

The next stage was to classify all the data for each attribute separately and build a model. The main aim of this is to provide input for the next stage of the classification process where we integrate the classifications obtained on different attributes. The important byproduct of this process is information on how each attribute contributes to the classification process. This in itself is valuable output for an occupational therapist researching handwriting. The single attribute classification provides information on the behavior of attributes that are typical of a deficiency.

Rather than building the model on all the data, the text is divided into $N$ parts and a model built for each part and for each attribute $Mod_{i,Att}$. There are two reasons for dividing the text into parts. The first is that this simulates an online classification. The second is that it is known that for poor writers writing usually deteriorates over time. It is therefore important to classify strokes of an unclassified writer with the model that corresponds to the same part of the writing task. Once we have found the classification for single attributes we want to create a final classification based on all attributes. For this we must find ways to combine the results obtained in the single attribute classification. We later suggest several heuristics along with experimental evaluations.

## The COACH Algorithm

**Algorithm 1** COACH($text$)

1: $FinalClass \leftarrow \emptyset, C \leftarrow \emptyset$
2: **for all** Att **do**
3:     Divide $text$ into N parts
4:     **for** $i \leftarrow 1$ to $N$ **do**
5:         $S_i \leftarrow$ first $M$ strokes from part $i$
6:         $C_{i,Att} \leftarrow ClassifyAttSet(S_i, Mod_{i,Att})$
7:         $C = C \bigcup C_{i,Att}$
8: $FinalClass \leftarrow CombineHeuristic(C)$
9: **return** $FinalClass$

**Algorithm 2** ClassifyAttSet($S_i, Mod_{i,Att}$)

$PROF \leftarrow \emptyset$,$POOR \leftarrow \emptyset$;
**for** $s_j \in S_i$ **do**
    $Class_j = classify(s_j, Mod_{i,Att})$;
    **if** $Class_j = prof$ **then**
        $PROF++$
    **if** $Class_j = poor$ **then**
        $POOR++$
**if** $PROF > POOR$ **then**
    **return** $prof$
**if** $PROF < POOR$ **then**
    **return** $poor$
**if** $PROF = POOR$ **then**
    **return** $Random(prof, poor)$

The classification of a new subject is performed using $COACH$. The $COACH$ algorithm appears in Algo. 1. $COACH(text)$ is provided with the text belonging to a new subject that we wish to classify. For each attribute we divide the text into $N$ parts, select the first $M$ strokes from each part and classify them using $ClassifyAttSet(S_i, Mod_{i,Att})$. Once we have a classification for each ('attribute','text part') pair we use a heuristic to combine all classifications. We tried different heuristics for combining these into one multiple attribute classification performed in $CombineHeuristic()$ (line 8). The first is to simply use a majority vote and choose the classification that was found most often in the single attribute classifications. The second is to use one attribute on part of the text and another for other parts. The third is to choose the attribute we use to classify based on the models we find. Details on these heuristics along with some experimental results appear later on.

The $ClassifyAttSet(S_i, Mod_{i,Att})$ in Algo. 2 performs iterative single attribute classification. $S_i$ is the set of strokes currently being classified. $Mod_{i,Att}$ is the model built from training data that corresponds to part $i$ of the text for attribute $Att$ (pressure, tilt or inAir). For each unclassified subject we classify single strokes from one part of the text and use a majority vote to determine the classification. In case of a tie we use random classification.

## Experimental Results

### Dataset

The data used was collected on a WACOM x-y digitizing tablet using a wireless electronic pen with pressure sensitive tip. At each time interval samples of the x and y coordinates, pressure, tilt and azimuth are taken, which creates a time series for each attribute. We used two datasets. Both of them include children in elementary school. Each set has two groups of subjects, one including the poor writers, and the other of proficient writers (the control). The sets are

- **DCD: Developmental Coordination Disorders.** - 42 subjects, 22 poor (**DCD**), 20 proficient. **DCD** is a motor impairment that affects a subjects ability to perform the skilled movements necessary for daily living and among other things affects handwriting proficiency.
- **Dysgraphic** - 94 subjects, 49 poor (**Dysgraphic**), 45 proficient. "Dysgraphia" is a learning disability resulting from difficulty in expressing thoughts in writing.

Each subject is labeled by a trained occupational therapist using a standardized evaluation tool. It must be noted that not all subjects classified as poor are the same, some may be more similar to proficient writers than others. Furthermore in the **Dysgraphic** dataset the labeling was performed initially by teachers and only then updated by occupational therapists and may be less reliable. Thus, we are working with a noisy data-set.

### Evaluation

The classification was done using Leave-one-out cross-validation. The **success rate** is the percentage of subjects classified correctly. This is shown as *% success* on the Y axis in the graphs. The classification is performed on five parts of the text. The results for each part use all strokes obtained from current and previous text parts ('cumulative'). This corresponds to the *text part* on the X axis. We have results on both the DCD and the Dysgraphic data sets. For both groups we ran ClassifyAttSet($S_i, Mod_{i,Att}$) (Algo. 2) with Att=tilt, Att=pressure and Att=inAir. We chose $M = 10$ strokes from each part for our experiments, because this number provided good classification. When we used smaller numbers such as $M = 4$ success rates dropped, however using $M = 50$ did not improve success rates. We use $N = 5$ parts of text, this provided us with as many sections as possible while maintaining enough strokes in each section, if we set $M = 10$.

### Results for Single Attribute Classification

We first present the average results for the **DCD** data in (Fig. 1). For tilt and inAir the classification reaches over 60% on average. For pressure we obtain a success rate of 70% after the first text part (after only 10 strokes), and reach over 80% on average when using the entire text.

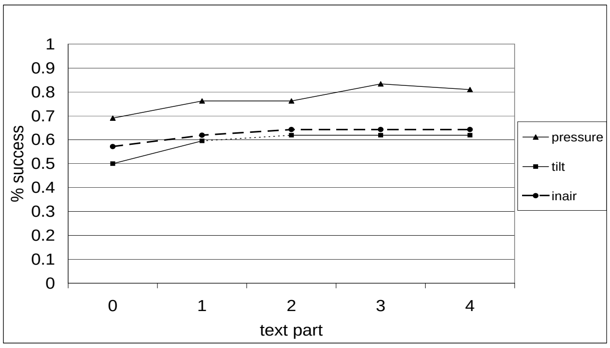


Figure 1: DCD results for classification.

We follow with the **Dysgraphic** results in Figure 2. The results for tilt are 60%, for inAir they reach 65%.For pressure we obtain a success rate of 75% for the entire text.

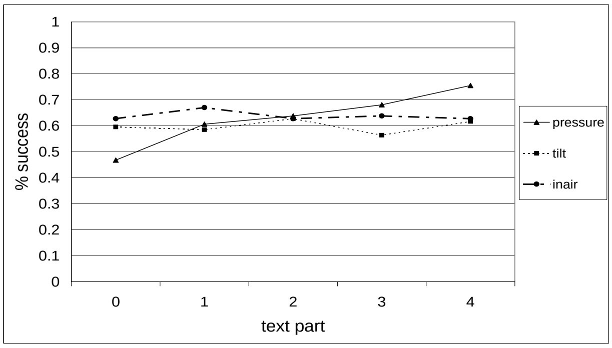


Figure 2: Dysgraphic results for classification.

### Heuristics for Multivariate Classification

As mentioned earlier once we have a single attribute classification we also experimented with some heuristics. The description of these heuristics together with experimental results follow:

- **TD-pres**: for **DCD** data the main attribute that contributes to classification is pen pressure. We noticed that the classification using tilt was not successful overall. However when the classification using tilt was '**DCD**' it was nearly always correct. This is shown in Fig. 3. Therefore we introduced the following heuristic: Use classification of tilt when it classifies as '**DCD**', otherwise use pressure classification. In Fig. 4 we show the benefit obtained from the **'TD-pres'** heuristic and reach 85% average success rate.
- **A-P**: for **Dysgraphic** data we noticed that the inAir attribute provides a success rate of over 60% after only 10 strokes have been written. The pressure attribute only starts contributing later (probably because the writers get tired - a known symptom of dysgraphia). We used this to derive the following heuristic: use classification obtained from inAir attribute for first parts of text and then transfer to using the pressure attribute as text proceeds. In Figure: 5 we present the improvement made by using the **'A-P'** heuristic and reach close to 76% average success rate.

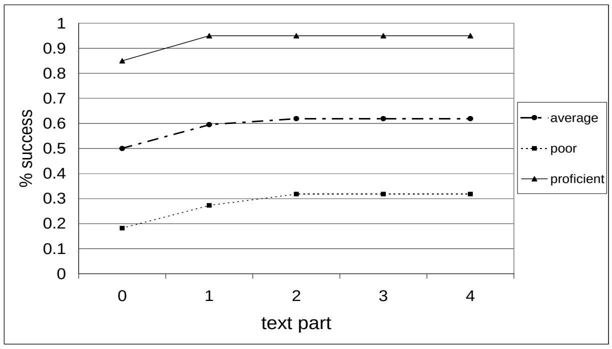


Figure 3: DCD results for classification using tilt.

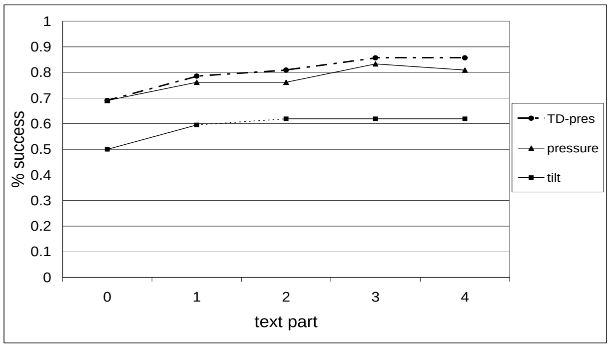


Figure 4: DCD classification using TD-pres heuristic.

## Discussion and Future Work

This research shows how AI techniques can be adapted and enhanced for handwriting deficiency classification. We have shown that the COACH algorithm can classify the writers with an 80% success rate on average without any additional information about the type of deficiency we are trying to classify. These results are considered to be very good in this domain, as diagnosing writing deficiencies is not an exact science and perfect classification is not expected. Our

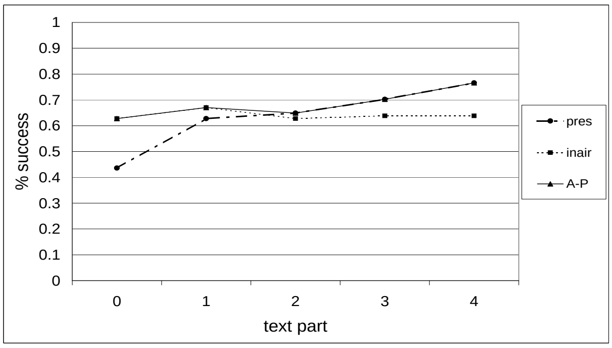


Figure 5: Dysgraphic classification using A-P heuristic.

results prove that AI is suitable for this domain. The results for the Dysgraphic dataset are lower than the DCD group for a number of reasons. First, the labeling of the initial data for the Dysgraphic set is noisy, which in turn affects our ability to classify correctly. Another cause may be that writers labeled Dysgraphic may not be very different from proficient writers, as opposed to writers with DCD that have very poor writing skills. DCD writers may be more similar to each other since they all suffer from the same problem. Dysgraphia however results from various causes and therefore is more diverse. For both groups pressure is the main attribute that contributes to the classification. However, how pressure affects the writing in each group is different. For Dysgraphic writers the pressure only discriminates after a large proportion of the text is written, since writers tend to tire over time. DCD writers have trouble with pen pressure right from the start because of the nature of their disability. InAir is also especially important in the Dysgraphic set where it is the main attribute that is immediately distinguishable. This is of great importance to an online system where fast discrimination is desired. Hence even when one attribute is dominant there is information hidden in other attributes. This analysis enabled by COACH is valuable information for correct diagnosis and choosing an appropriate remediation approach.

These results presented in this paper can be expanded to other cases. Alzheimer's disease, multiple sclerosis, and other diseases, which affect handwriting abilities, would be interesting to study. We would like to expand our classification to differentiate between more than two classes in order to develop a deployed classification system that can be used for diagnosis and remediation of deficiencies. Another issue we plan to address is understanding the unique features of the subjects that were not correctly classified in comparison to those that were classified correctly. Finally, in the future we plan to further explore multivariate sequence learning in a general fashion in order to classify data from many domains.

## References

Bahlmann, C. 2006. Directional features in online handwriting recognition. *Pattern Recogn.* 39(1):115–125.

Baxter, R. A.; Williams, G. J.; and He, H. 2001. Feature selection for temporal health records. *Lecture Notes in Computer Science* 2035:198–209.

Bharath, A.; Deepu, V.; and Madhvanath, S. 2005. An approach to identify unique styles in online handwriting recognition. In *ICDAR '05: Proceedings of the Eighth International Conference on Document Analysis and Recognition*, 775–779. Washington, DC, USA: IEEE Computer Society.

Bouletreau, V.; Vincent, N.; Sabourin, R.; and Emptoz, H. 1998. Handwriting and signature: one or two personality identifiers? In *Fourteenth International Conference on Pattern Recognition*, volume 2, 1758–1760.

Dash, M., and Liu, H. 1997. Feature selection for classification. *Intelligent Data Analysis* 1(1-4):131–156.

Erez, N., and Parush, S. 1999. The Hebrew Handwriting Evaluation (2nd ed.). Israel, Jerusalem: School of Occupational Therapy. Faculty of Medicine. Hebrew University of Jerusalem.

Kadous, M. W., and Sammut, C. 2004. Constructive induction for classifying multivariate time series. In *European Conference on Machine Learning*.

Kalera, M. K.; Srihari, S. N.; and Xu, A. 2004. Offline signature verification and identification using distance statistics. *International Journal of Pattern Recognition and Artificial Intelligence* 18(7):1339–1360.

Liu, B., and Liu, J. 2002. Multivariate time series prediction via temporal classification. In *ICDE*, 268.

Liu, T.-Y.; Yang, H.-Y.; Zheng, X.; Qin, T.; and Ma, W.-Y. 2007. Fast large-scale spectral clustering by sequential shrinkage optimization. In *ECIR*, 319–330.

Morabito, F. C., and Versaci, M. 2003. Fuzzy neural identification and forecasting techniques to process experimental urban air pollution data. *Neural Netw.* 16(3-4):493–506.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Rosenblum, S.; Parush, S.; and Weiss, P. L. 2003. The in air phenomenon: temporal and spatial correlates of the handwriting process. *Perceptual Motor Skills* 96(3 pt 1):933–954.

Rosenblum, S.; Weiss, P. L.; and Parush, S. 2003. Computerized temporal handwriting characteristics of proficient and non-proficient handwriters. *The American Journal of Occupational Therapy* 57(2):129–138.

Witten, I. H., and Frank, E. 2005. *Data Mining: Practical machine learning tools and techniques*. San Francisco: Morgan Kaufmann, 2nd edition.

Zaki, M. J.; Lesh, N.; and Ogihara, M. 2000. Planmine: Predicting plan failures using sequence mining. *Artif. Intell. Rev.* 14(6):421–446.

Zhu, S.; Takigawa, I.; Zhang, S.; and Mamitsuka, H. 2007. A probabilistic model for clustering text documents with multiple fields. In *ECIR*, 331–342.