Analysis of Different Writing Styles with the Self-Organizing Map

Vuokko Vuori and Erkki Oja

Laboratory of Computer and Information Science
Helsinki University of Technology
P.O. Box 5400, FIN-02015 HUT, Finland

Abstract

This work shows how a Self-Organizing Map (SOM) can be applied in the analysis of different handwriting styles. Handwriting styles are represented with vectors whose components reflect the tendencies of the writers to use certain prototypical styles for isolated alphanumeric characters. The study shows that the correlations between different writing styles congruent with prior human knowledge can be found with SOM. It turns out that the SOM can make a distinction between writers with cursive style or a mixture of print and block styles. The former group of subjects forms a clear cluster in a writing-style-space, and in their case, the correlations between the writing styles are very strong and understandable.

1 Introduction

In this work, natural writing styles of several writers are analyzed. The aim of the study is to find a representation for writing styles which enables their comparison and detection of possible clusters in a writing-style-space. In addition, correlations between the writing styles of characters of different classes are searched for. This work tries to find answers to questions such as: "If I know how you write letter 'a', can I infer something about the way you write letter 'd' based on what I know about other writers?". This kind of information might be useful in automatic recognition of handwritten characters [1] in a hand-held device with text input by 1) helping to distinguish confusing characters without using the context of the characters or any other language model, and 2) speeding up the system's adaptation to a new writing style.

The writing style of a single writer is represented with a vector whose components indicate the tendency of the writer to use some predefined writing styles. These writing styles have been selected by a clustering algorithm and then further tuned with the Learning Vector Quantization (LVQ) [2] algorithm using character data collected from several subjects. In order to find correlations between and within the writing styles of different writers, the writing style vectors are ordered with a Self-Organizing Map (SOM). The SOM-algorithm maps similar writing style vectors close to each other and correlated writing styles of individual character classes produce similar-looking component planes.

2 Writing style vectors

The writing style of an individual writer is presented with a so called writing style vector. The components of the vector indicate the tendency of the writer to use a particular style of writing for a given character class. The tendency is measured by calculating the average similarity between the character samples and their closest correct prototypes. The next sections will explain in detail the steps taken to form the writing style vectors for the writers. First, the method for comparing the characters, the clustering algorithm, and the selection and fine-tuning of the prototypes are described. Next, the transformation from a dissimilarity measure into a similarity measure is presented.

2.1 Dissimilarity measure

The characters were compared with each other by using a dissimilarity measure based on the Dynamic Time Warping (DTW) algorithm [3]. The DTW-algorithm matches two curves represented by sequences of data points so that the sum of the squared Euclidean distances between the matched data points is minimized. The matching is constrained by boundary and continuity conditions. Boundary conditions ensure that the first and last data points of the two curves are matched against each other. The continuity condition requires that all the data points are matched at least once and in the same order that they have been produced. The characters are compared stroke wise – the dissimilarity between two characters is the sum of the dissimilarities between the stroke pairs. Dissimilarity between characters of different numbers of strokes is set to infinity. More detailed description of the dissimilarity measure can be found from references [4, 5].

2.2 Cluster Tree

A simple clustering algorithm was used for forming a tree-like structure for each class and stroke-number variation. In the beginning of the algorithm, there were as many clusters as there were character samples. Next, those two clusters whose middle items were the most similar were merged and the middle item of the new cluster was found. Then again, two clusters were merged into one in a similar manner. The algorithm continued until there was only one cluster left. The benefit of this algorithm is that it does not require any prior information on the number of writing styles. It also has a tendency to keep malformed, rare or even erroneous samples as their own clusters.

All the cluster trees were examined and the prototypes for the fine-tuning phase were manually selected. Some of the very rare styles (only one or two samples from one writer) were omitted and all the selected prototypes were significantly different (drawing order or direction of the strokes) from each other. The number of prototypes selected in this phase was 327 and it determines the dimension of the writing style vectors. Some of the prototypes can be seen in Figure 3.

2.3 Tuning with LVQ

The prototypes selected by using the clustering algorithm were fine-tuned with the LVQ-algorithm using the rest of the characters as training samples. The original LVQ-algorithm was modified so that vectors of different length can be compared. The LVQ-algorithm is able to reshape the prototypes gradually so that they are more general representatives of a group of similar learning samples.

The learning samples were introduced one by one and the best matching prototype belonging to the same class was found for each of them. The data points of the prototype were moved towards corresponding data points of the training sample. The DTW-algorithm used for the matching establishes the required point-to-point correspondence between the data points. The suitable number of learning epochs (30) and the value of the learning rate (linearly decreasing from 0.05 to zero) were selected by studying the evolution of the average dissimilarity per data point during the fine-tuning process. The modified version of the LVQ-algorithm is described in detail in reference [6].

2.4 Transforming dissimilarity into similarity

The dissimilarity measure obtained with the DTW-algorithm has a range from zero to infinity and it depends on the numbers of data points and strokes. Therefore, the dissimilarities between strokes were normalized by the number of data point matchings and the total dissimilarities were divided by the number of strokes. After these normalizations, the dissimilarities (D) were transformed into similarity measures (S) in the following way:

$$S = \frac{1 - \tanh(\alpha(D - \beta))}{2}. \qquad (1)$$

Similarity is a decreasing function of the normalized dissimilarity and in this specific case its range is between zero and one. Suitable values for parameters $\alpha = 0.0003$ and $\beta = 20\,000$ were selected so that the clear majority of the dissimilarities for the best matching prototypes have similarity values close to one and the dissimilarity values for the rest of the prototypes are close to zero.

3 Data

The data used in the experiments consists of isolated handwritten characters ('a'-'z', 'å', 'ä', 'ö', 'A'-'Z', 'Å', 'Ä', 'Ö', '0'-'9') collected from 45 subjects. The writing equipment consisted of a special pressure sensitive tablet attached to a Unix workstation. The resolution of the tablet is 100 lines per millimeter and the sampling rate is at maximum 205 data points per second. A character can therefore be presented with a series of the x- and y-coordinates of the moving pen point. The total number of characters was approximately 40 000. Half of the subjects wrote the characters after a dictation of a short story and without any visual feedback. The other half wrote characters in random order. This time, the characters were shown on the computer screen and were recognized on-line. All the characters were written by using natural style or, in other words, without any constrains on the style. There seems to be no significant difference between the quality of the characters collected with the two setups. However, the distribution and number of the characters per subject are different. In the former case, the distribution of characters is somewhat similar to that of the Finnish language. The number of characters per subject was in the latter collection at least twofold compared to that of the former collection.

4 Analysis of the writing styles

As the main interest of this work is on the correlations between the writers, all the styles used by only a

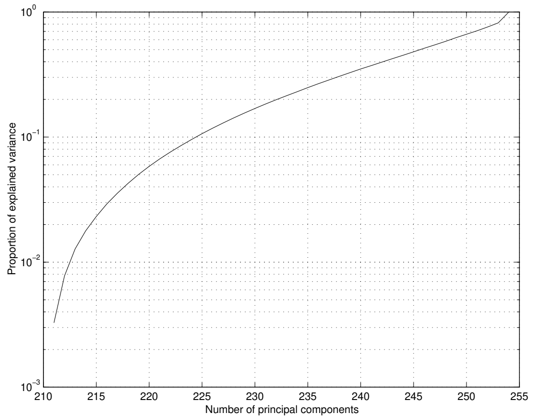

Figure 1: Principal Component Analysis.

single writer were omitted from the writing style vectors. As the first thing, the resulting 254-dimensional writing style vectors were analyzed with other commonly used techniques: Principal Component Analysis (PCA), Curvilinear Component Analysis (CCA), and Sammon mapping. The aim of these analyses was to estimate the true dimensionality of the data. Finally, the data was presented with a SOM in a hope of finding interesting structures such as clusters.

4.1 Dimensionality of the data

A PCA, CCA and Sammon mapping carried out for the writing style vectors showed that the dimensionality of the data is high compared to the number of writers. These experiments were carried out by using SOM Toolbox [7], a software package for Matlab. A PCA, see Figure 1, showed that the ten most important component directions can explain only a half of the variations in the data. To capture 90% of the variations, as many as 25 component directions are required. Therefore, it seems that the data is truly high dimensional and obvious clusters could not be detected by using linear projections to lower, say two or three, dimensional space. Results obtained with nonlinear projection methods, namely Sammon mapping and CCA, were congruent with those of PCA: no clear clusters could be found. However, some of the writing style vectors were mapped close to each other. This could be seen by looking at the three-dimensional clouds of data points from different angles.

4.2 Creating the SOM

Several alternatives for the Self-Organizing Map's size, lattice, neighborhood function, training algorithm, training parameter and epochs, initialization, winner search, and updating rule were experimented with. Different maps were compared with each other by using two quality measures: quantization error and ability to preserve the topology of the data. The former measure is the average distance between each data vector and its best matching map unit (BMU). The latter one is the proportion of all data vectors for which the first and second BMUs are not adjacent units.

The map size was soon fixed to 10×5 units which is approximately 10% more than the number of writers. Next, the topology of the map was selected to be a sheet with hexagonal lattice and Gaussian neighborhood. A linear initialization along the first two principal directions of the data proved to produce better results than a random initialization. The batch training algorithm was applied with Euclidean metric as their combination provided much faster and reliable convergence than an on-line training algorithm or a metric based on the angle between two vectors.

The training was carried out in two phases. In the first phase, rough training, the radius of the neighborhood was linearly decreased from 5 to 2 during 100 training epochs. Next, in the fine-tuning phase, the radius was decreased from 2 to 1 during 400 epochs. The numbers of the epochs are perhaps unnecessarily large but there was no need to optimize them as the batch training was rather fast taking less than one minute in total and the extra epochs did not have any unwanted side-effects. The quantization error of the SOM was approximately 4.3 while the topological properties of the data were preserved perfectly.

5 Results obtained with the SOM

The U-matrix of a SOM is helpful in detecting clusters on the map. Its coloring is based on the distances between neighboring map units. Areas where the neighboring map units are similar to each other are colored with dark gray whereas light shades indicate that the differences between the units are more significant. The U-matrix and some interesting component planes of the constructed SOM are shown in Figures 2 and 3. The interesting component planes are those whose range is at least 0.7, which means that there is significant variance between the map units. The component planes in Figure 3 are ordered according to their quadrants. The values of the map units were summed up in each quadrant and the ordering was performed on the basis of these four sums.

From the U-matrix of the SOM, see Figure 2, it can be seen that the writing styles can be roughly divided into two groups. The first group is concentrated in the upper right corner of the map and the other group is mapped on the lower parts of the map. The clusters

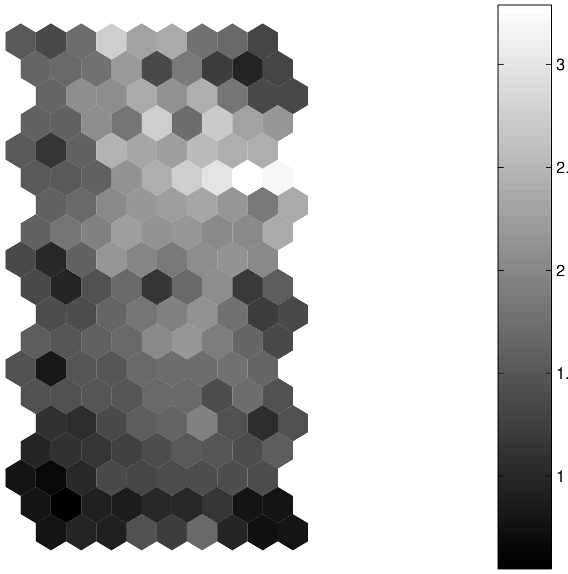

Figure 2: U-matrix formed from the complete, 254-dimensional data vectors

can be explained by studying the component planes shown in Figure 3. Light shades on the component planes indicates that writers mapped in that location have a high tendency to use the corresponding writing style. Dark shades means that the writing style is not likely to be used. Writers who exclusively use cursive writing styles very similar to those taught in Finland in the elementary school are located in the upper right corner. The rest of the writers mostly use a mixture of print and block characters and the interpretations of the differences between the writers, say, in the left and right lower corners, cannot be made as easily.

The correlations between the writing styles of single characters and individual writers found with the SOM are congruent with the human prior knowledge on the writing styles. For example, prototypes which contain similar parts, such as cursive lower case 'a', 'g', and 'd', have strikingly similar component planes. In addition, writers who have adopted the writing styles learned in school and therefore have less inter-writer variation in their writing style form a clear cluster located in the right upper corner of the map.

6 Conclusions

This study has showed that the correlations between different writing styles can be found with a Self-Organizing Map. Writing styles of several subjects were characterized with vectors whose components re-

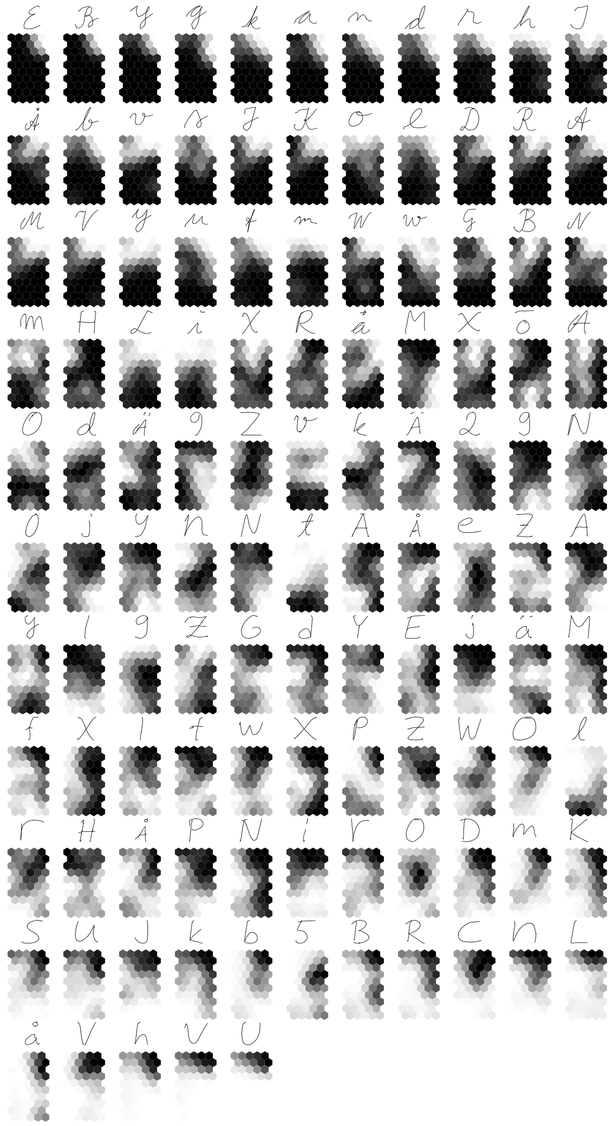

Figure 3: Some interesting component planes with the corresponding prototypes. The range of the component values is at least 0.7.

flected the tendency of a writer to use some prototypical styles for isolated characters. The SOM was able to make distinction between subjects who use cursive style and those whose style is a mixture of print and block styles. The correlations between writing styles of single characters found by looking at the component planes of the SOM are intuitively pleasing. They are congruent with the prior human knowledge which was not used in the construction of the map.

These results justify the use of the knowledge on the writing styles of other writers in the adaptation of a recognition system into a new writing style. The next steps in our work will involve testing whether a writing style vector formed by using only a few characters samples and the most similar SOM units can help to improve the initial accuracy of a prototype-based recognition system.

References

[1] R. Plamondon and S. N. Srihari, "On-line and off-line handwriting recognition: A comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 63–84, January 2000.

[2] T. Kohonen, *Self-Organizing Maps*, vol. 30 of *Springer Series in Information Sciences*. Springer-Verlag, 1997. Second Extended Edition.

[3] D. Sankoff and J. B. Kruskal, *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison*. Addison-Wesley, 1983.

[4] V. Vuori, "Adaptation in on-line recognition of handwriting," Master's thesis, Helsinki University of Technology, 1999.

[5] J. Laaksonen, V. Vuori, M. Aksela, E. Oja, and J. Kangas, *Cognitive and Neural Models for Word Recognition and Document Processing*, ch. Experiments with Adaptation Methods in On-line Recognition of Isolated Latin Characters. World Scientific Press, To appear in 2000.

[6] J. Laaksonen, J. Hurri, E. Oja, and J. Kangas, "Comparison of adaptive strategies for on-line character recognition," in *Proceedings of International Conference on Artificial Neural Networks*, pp. 245–250, 1998.

[7] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, "SOM Toolbox for Matlab 5," Tech. Rep. A57, Helsinki University of Technology, April 2000.