

Classifier Combination and Feature Extraction

Concavity Features

Luiz S. Oliveira and Marisa Morita

Federal University of Parana (UFPR)

Department of Informatics (DInf)

R. Rua Cel. Francisco H. dos Santos, 100, Curitiba, PR, Brazil

lesoliveira@inf.ufpr.br

1 Introduction

In the previous reports we have introduced a feature set based on global features and reported the performance of this feature set using three different architectures of HMMs to recognise handwritten month words. In this report we performed some experiments to combine those three architectures and also implemented a new feature set based on concavity information.

The source code for this new feature set is available in the SVN repository. Before compiling the code make sure to update the PPROJ variable in the `Make_orand_concfeat` file available in the directory `./tools/orand/batch/feature/gen` and also to update the defines in the `global.h` file, available in the directory `./tools/orand/inter/include`. To compile the source code use

```
make -f Make_orand_concfeat
```

And to execute the program, type

```
./orand_globfeat feat2 concf
```

The program generates three different files for training, validation and testing, respectively. Those files are located in the directory `./tools/orand/param/feature/feat2/concf/vq`

2 Combining Classifiers

A classical strategy for constructing classifiers is to evaluate different models and topologies in order to select the one with the highest performance with respect to a testing data set. However, the construction of a universal classifier that performs an appropriate classification for any kind of input is a utopian goal. Ranawana [1] states that, frequently, the construction of an ideal

classifier for a certain application is almost impossible. An interesting alternative to deal with the aforementioned problem is the construction of multiple-classifiers based systems. In this approach, many classifiers produce independent decisions, which are then combined to yield a final response (the estimated class). In fact, starting from the assumption that different kinds of classifiers allow us to classify accurately different patterns, a combination of these classifiers may achieve more accurate decision on the overall data set, instead of the performance achieved by only one classifier [2, 3, 4].

With that in mind, we have performed some experiments to combine the three different HMM architectures reported in the previous report. However, we were conscious that the expected diversity may not be available since all the classifiers were trained with the same feature set. In such a case, the only source of diversity would be the different architectures of the HMMs. In spite of that, we have implement and tested all the combination rules described by Kittler et al in [4]. Table 1 shows the results of each HMM architecture and also the results os all combination rules.

Table 1: Summary of the recognition rates obtained in the experiments

Experiment	Number of HMM Models	Rec. Rate (%)
I	24	80.9
II	36	82.2
III	12	82.4
Combinaton Rule		
Voting		82.0
Max		82.5
Sum		82.5
Product		82.2

As we may notice, the combination did not bring any improvement in terms of performance. The Sum and Max rules produced a recognition rate of 82.5%, which is similar to the best classifier (82.4%). Other rules, such as voting and Product, did not increase the recognition rate. The Product rule is effectively a severe rule of fusing the classifier outputs as it is sufficient for a single recognition engine to inhibit a particular interpretation by outputting a close to zero probability for it [4]. In other words, if only one classifier produces a low score for a given testing pattern, the combination of all classifiers would be a low score, independently of the scores produced by the other classifiers.

It is worth of remark, though, that even not improving the recognition rate of the system, combining these different architectures certainly provide a more reliable system. In order to increase the diversity in our classification system, we have implemented another feature set, which is described in the next section.

3 Concavity Features

Concavity features have been successfully used as pattern representation for different handwriting recognition problems [5, 6]. The basic idea of concavity measurements is the following: for each white pixel in the component, we search in 4-Freeman direction (Figure 1a), the number of black pixels that it can reach as well as which directions the black pixel is not reached. When black pixels are reached in all directions (e.g. point x_1 in Figure 1c), we branch out in four auxiliary directions (s_1 to s_4 in Figure 1b) in order to confirm if the current white pixel is really inside a closed contour. Those pixels that reach just one black pixel are discarded.

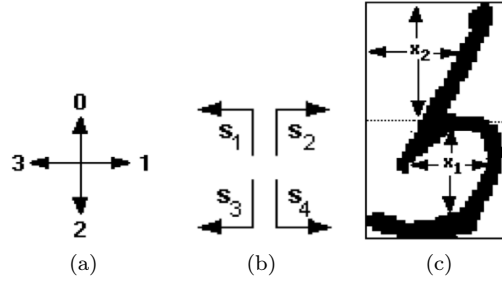


Figure 1: Concavity Features [6]

To reduce the feature vector as well as the handwriting variability, we are considering only the more relevant concavities. We are labelling the white pixels which have black pixels in three or four directions (five configurations). The white pixels with two or less directions are often located on the grapheme borders (e.g., point x_2 in Figure 1c). The labelling leads to a vector composed of nine components for each grapheme: s_1 to s_4 and five configurations. Since we are zoning the grapheme into two equal parts along the vertical axis, the final feature vector contains 18 components normalised between 0 and 1. In order to use this feature vector to train the HMM-based classifier, it needs to be discretised using a vector quantisation technique. Such a technique is being implemented right now.

4 Conclusions

In this report we have reported the results of the experiments related to the combination of classifiers based on global features. We have seen that the diversity produced by the different HMM architectures was not enough to produce the diversity that could be exploited by the combination techniques. To mitigate this problem, we have implemented a feature set based on concavities. Since this representation creates a feature vector composed of continuous values, it will be discretised before training the HMMs. For the next report we expect to have the recognition rates

produced by this new feature vector as well as the results of the combination of concavities and global features.

References

- [1] R. Ranawana, “Multi-classifier system – review and a roadmap for developers,” *International Journal of Hybrid Intelligent Systems*, vol. 3, no. 2, pp. 35–61, 2006.
- [2] G. Brown, J. Wyatt, R. Harris, and X. Yao, “Diversity creation methods: a survey and categorization,” *International Journal of Information Fusion*, vol. 51, no. 2, pp. 5–20, 2005.
- [3] L. I. Kuncheva and C. J. Whitaker, “Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy,” *Machine Learning*, vol. 51, no. 2, pp. 181–207, 2003.
- [4] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, “On combining classifiers,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.
- [5] R. Jayadevan, S. R. Kolhe, P. M. Patil, and U. Pal, “Handwritten digit recognition: benchmarking of state-of-the-art techniques,” *Pattern Recognition*, vol. 36, pp. 2271–2285, 2003.
- [6] M. Morita, R. Sabourin, F. Bortolozzi, and C. Y. Suen, “Segmentation and recognition of handwritten dates: an hmm-mlp hybrid approach,” *International Journal on Document Analysis and Recognition*, vol. 6, no. 4, pp. 248–262, 2004.