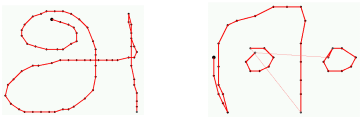
Dynamic Time Warping applied to Tamil Handwriting Recognition

Ralph Niels and Louis Vuurpijl
Nijmegen Institute for Cognition and Information (NICI)
Radboud University Nijmegen, The Netherlands

Data and techniques

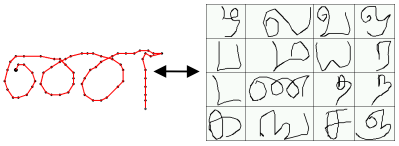
Data

40 writers each wrote 10 instances of 156 Tamil characters. All data was recorded *online* (on a writing tablet).



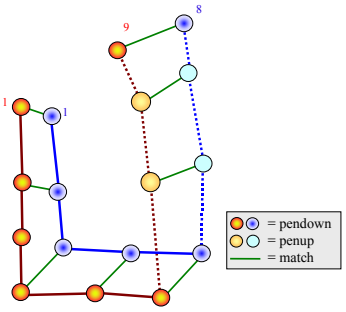
Classifier

The unknown sample is compared to all labeled prototypes*. The most similar prototype (with the smallest DTW-distance) is used for classification.



* Prototypes were created by clustering instances of characters using HCLUS (Vuurpijl & Schomaker, 1997), and merging clusters into prototypes using a technique based on Learning Vector Quantization.

Dynamic Time Warping

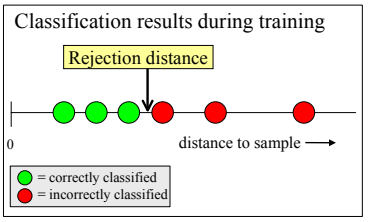


Our modification of DTW makes a match between two curves by making a point-to-point comparison based on 3 conditions. Two points match if:
(i) They are both first or both last points of their curve.
(ii) They are both penup or both pendown (modification of original DTW).
(iii) They are at the same relative position on their curve.

Rejection

Rejection allows the classifier to reject a classification if the certainty is below a certain threshold. Two measures were implemented:

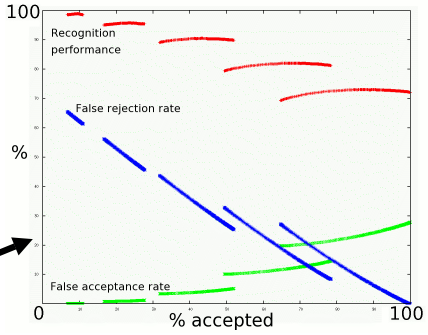
(i) Agreement: Indicates how many of the 4 next-nearest prototypes have the same label as the nearest one.
(ii) Rejection distance: Prototype dependent value that indicates the distance range in which, during training, only correct classifications were made.



Experiments and conclusion

Experiments

	Procedure	Training	Testing	Recognition performance
Writer dependent test	Separate test for 10 writers			Average: **87.76%**
Writer independent test	One test			**72.11%**
Rejection test	Monte Carlo simulation			



Conclusion

Performance	Rejection	Future work
Literature reports between 71% and 96%. Alphabet size varies from 26 to 156 Tamil characters, so comparisons are hard to make. Our classifier performs relatively good.	Rejection can improve the reliability of the classifier, but can also increase the number of false rejects.	Gaining more insight in strong and weak points of the classifier. Which characters are problem cases? Which characters are typically mixed up? This information can help to improve the system, for Latin characters too.