Determination of the Number of Writing Variants with an HMM based Cursive Word Recognition System

Marc-Peter Schambach
Siemens Dematic AG
78467 Konstanz, Germany
Marc-Peter.Schambach@siemens.com

Abstract

An important parameter for building a cursive script model is the number of different, relevant letter writing variants. An algorithm performing this task automatically by optimizing the number of letter models in an HMM-based script recognition system is presented. The algorithm iteratively modifies selected letter models; for selection, quality measures like HMM distance and emission weight entropy are developed, and their correlation with recognition performance is shown. Theoretical measures for the selection of overall model complexity are presented, but best results are obtained by direct selection criteria: likelihood and recognition rate of training data. With the optimized models, an average improvement in recognition rate of up to 5.8 percent could be achieved.

1. Introduction

When building a system for cursive handwriting recognition, an important step is finding the right model that best describes cursive script. One decision concerns the question of how many writing variants, or "*allographs*" of each letter have to be considered to get a model representative of all occuring writing styles. Mostly, these decisions are made manually, based of assumptions that are made about the writing. This way, upper- and lowercase letters are distinguished often, as well as hand block and cursive writing styles. But a good model has to consider those variants that really occur. Therefore, it is useful to determine the writing variants automatically, especially if there is no detailed knowledge about writing styles. This is e.g. the case in postal automation systems, where recognition systems specific to different countries with different writing styles and even alphabets have to be developed. Then allographs have to be determined automatically by analyzing the same sets of training data, which are used for setting the recognition system parameters.

Identifying writing variants means clustering the allographs. Therefore, a distance between allographs which are modelled in the recognition system by linear HMMs has to be defined. Furthermore, a quality measure for HMMs is proposed which allows comparison of the usefulness of different allograph models. Based on these measures, an iterative method to adapt the HMM script model automatically to the best topology is developed. Experiments show how the recognition rate is influenced by the model, and Bayesian model selection criteria are presented that measure how well the model generalizes to unseen test data. However, no over-adaptation to the training data could be detected, so ratings of the training data are useful selection criteria.

This paper is organized as follows. First, an overview of the underlying recognition system and its applications is given. Then, in section 3, a distance measure for letter HMMs is defined and it is shown that the recognition performance is influenced by it. Section 4 proposes the emission weight entropy as a quality measure for letter models. In section 5, the algorithm for model adaptation is presented and model selection criteria are defined. In section 6, effects of the different selection criteria to character modelling and recognition performance are discussed. Finally, a summary and outlook to further work are given.

2. System architecture

The script recognition system is based on linear left-to-right HMMs, with a semi-continuous, tied-mixture probability modelling structure. The script model is defined by a set of *graphemes* (letters, numbers and special characters). Different writing variants of a grapheme (allographs) are combined in a *multipath letter model* with multiple, parallel state "paths" [4].

The system is applied in postal automation systems to recognition tasks in cursive script, hard-to-segment hand

block and machine print, and Arabic script recognition. For testing the algorithm, eight configurations from different countries have been selected: Canada (alphanumeric), Germany, the USA (numeric and alphabetical), and the United Arab Emirates (arabic). Additionally, the CEDAR database (numeric and alphabetical) [3] has been used. Word recognition tests have been performed with dictionaries of size 100–200 (country, city, street names) and 27000–43000 (ZIP codes).

3. HMM distances

Determining the writing variants in script samples means clustering the training data. For any clustering algorithm, it's useful to define a distance between objects, in this case between allograph HMMs. The distance proposed is calculated for left-to-right allograph models, but it can be also applied to multipath letter models.

3.1. Definition

The method is called Statistical Dynamic Time Warping, and has been presented in [1]. The idea is the following: instead of using the Viterbi algorithm for classifying sequences of input vectors, the *state sequence* of the first HMM λ_1 is classified by the second, λ_2. In a semicontinous HMM system, the class probabilities of the input data can be replaced simply by the class emission weights of the states, and standard Viterbi algorithm can be performed. To make the measure symmetrical, each of the models to compare is classified by the other,

$$D(\lambda_1, \lambda_2) = \tfrac{1}{2}\left(D(\lambda_1, \lambda_2) + D(\lambda_2, \lambda_1)\right). \qquad (1)$$

With this method it is possible to calculate the distance between models, even when they have differing numbers of states.

3.2. Effects in recognition systems

A good method to examine the quality of the distance measure proposed is given by visual inspection. A method for visualization is presented in another paper [6]. Fig. 1 shows some examples of letters models with smallest or highest distance in a recognition system for German ZIP codes. The results correspond quite well to human perception of similarity. To prove the usefulness of the used distance measure, a correlation between misclassification and model distance is supposed. An example with German adress data (34 letter models) is shown in Fig. 2. It can be seen that the possibility of misclassifications, i.e. confusion of letters inside a word, grows with smaller distance between letter HMMs. The highest misclassification rates appear with those models having the smallest distance. This

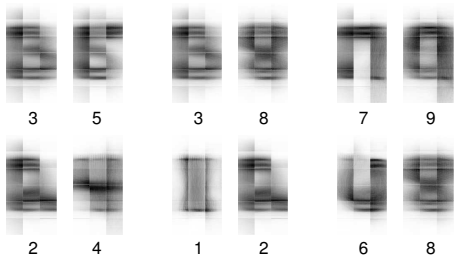

Figure 1. Visualization of HMM-Parameters for model pairs with low (top) and high distance (bottom)

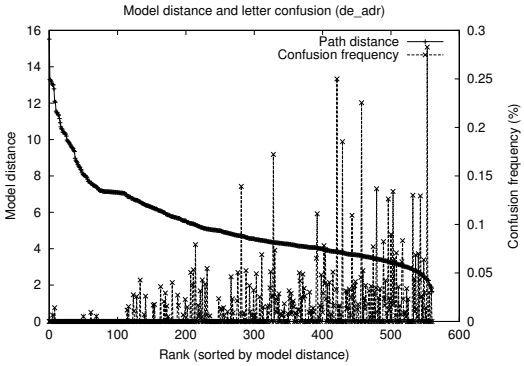

Figure 2. Correlation between model distance and misclassification

correlation has been observed in all eight recognition systems studied.

4. Letter model entropy

To test whether allographs are useful to model, an independent measure for model quality is needed. For semicontinuous HMM systems, an easy measure is the entropy of emission weights $b_s(c)$ for classes c in state s,

$$H(s) = -\sum_c b_s(c) \log b_s(c). \qquad (2)$$

No properties of the classes themselves are considered; but when classes are well separable, the value indicates how well defined model states are. The letter model quality is defined by averaging the entropy of all states. The influence on recognition performance can be seen in Fig. 3 for US numbers. Dictionary effects influence the recognition rate for single characters, but still the expected correlation can be seen.

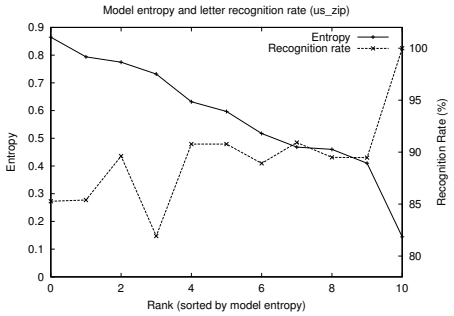

Figure 3. Correlation between model entropy and letter recognition rate

5. Allograph adaptation

Starting from a trained system with default model, the allograph clustering algorithm iteratively adds and removes allograph models. In every iteration, the following tasks are performed:

- Select allograph state paths for modification. Different strategies have been developed and tested.
- Modify the selected allographs. Remove a state path by deleting or mixing similar paths, or add one by modifying a selected, existent path.
- Retrain the HMM, including codebook calculation.

Break when a maximum or minimum size of the allograph model is reached. The final script model is selected by criteria presented in section 5.2.

5.1. Strategies for selection and modification

Different strategies for allograph selection and modification have been developed, three for adding and three for removing allograph models.

5.1.1 Model quality: distance and entropy

Similar allographs within a single letter model represent the same data and are therefore candidates for merging. Using the distance defined in section 3, the grapheme model with the pair of paths closest to each other is chosen and the respective paths are joined by averaging the emission weights.

As quality measure for letter models, the emission weight entropy (2) has been proposed in section 4. The model which is modelled worst, indicated by the highest average allograph entropy, is chosen to have an additional writing variant. As initialization, the best allograph (the one with lowest entropy) is doubled, and emission weights are shifted randomly.

5.1.2 Model likelihood

The effect of grapheme modelling within the recognition system can be given by model likelihood. This is the contribution of a particular letter model to training data likelihood, and it can be calculated from the Viterbi path, where likelihood is assigned to each model state.

To remove an allograph, the one with the overall worst contribution in likelihood is chosen. Model likelihood also can be used to add allographs: the grapheme with the average worst likelihood of all allographs is chosen, and the allograph with the highest likelihood variance is doubled.

5.1.3 Amount of represented data

A pragmatic approach to improving the recognition system selects model detailing by frequency in training data. Graphemes that appear more often have higher influence on overall performance, and are modelled with more allographs.

The allograph frequency is defined by the product of absolute number of the grapheme in training data and the transition probability to the first state in the allograph state path. Allographs with low frequency are removed; those with high frequency are doubled.

5.2. Model selection criteria

After adaptation iterations, the best script model has to be selected. Generally, when trying to find the "right" model, the danger of over-adaptation to the available data exists. In our case, more allograph models could result in worse generalization to test data.

Bayesian model selection criteria are a general approach to deal with this subject. Searching the best model structure M_s for data X, you have to maximize

$$P(M_s|X) = \frac{P(X|M_s) \cdot P(M_s)}{P(X)}. \qquad (3)$$

As model selection is independent of data evidence $P(X)$, and no explicit prior for the model structure $P(M_s)$ is given, the best model structure is estimated by the maximum likelihood criterion $P(M_s|X) \propto P(X|M_s)$. The likelihood has to be calculated by intergrating all parameters Θ:

$$P(X|M_s) = \int_\theta P(X|\theta, M_s) P(\theta|M_s) d\theta. \qquad (4)$$

This calculation is called Bayesian integration. Because with HMM it is not possible to solve this integral analytically, approximations like Laplace or Cheeseman-Stutz [2] have been made, which lead to criteria penalizing high model complexity [5] [7].

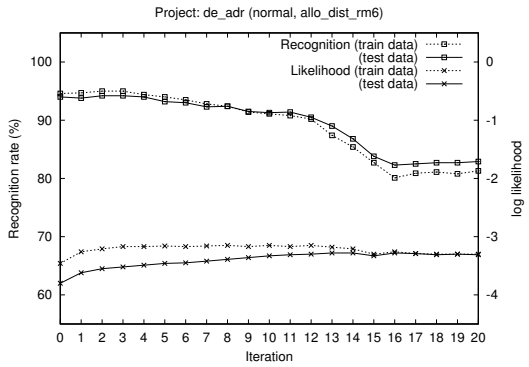

Figure 4. Evolution of likelihood and recognition rate during allograph adaptation iterations

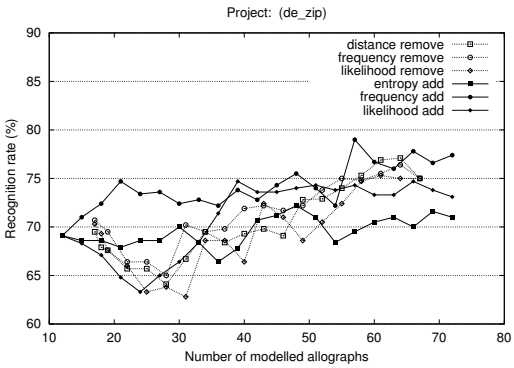

Figure 5. Recognition rate for different number of allographs during iterations

None of these applied well to HMM script word recognition, because no effects of over-adaptation have been seen with reasonable maximum model complexity. Generalization effects can be detected, but maximizing adaptation to training data still achieves best test results. Two simple criteria have been used: likelihood of training data, and training recognition rate with synthetic dictionary data.

6. Experimental results

For each of the eight projects tested, experiments for all six selection and modification strategies in section 5.1 have been performed. A maximum of 20 iterations has been carried out, in each modifying 30 percent of all models. For adding allographs, the baseline system modelled one allograph per grapheme; for removing, the baseline modelled six allographs.

6.1. Iterations

The characteristic of iterations is exemplified in Fig. 4 for the German address recognition system. Likelihood and recognition rate for both training and test data are plotted while allograph models are iteratively removed by the distance criterion. Looking at the likelihood curves, generalization improves with the classifier getting less complex: the gap between training and test data gets smaller when models are removed. A similar effect can be seen for the recognition rate. [1] However, with the maximum number of models used in the experiments, no overadaptation effects could be seen in any recognition project.

The recognition rate forms a tableau whose edge around iteration 11 could indicate an "appropriate" modelling: the right balance between model complexitiy, resulting in computational costs, and generalizable recognition performance.

6.2. Recognition rates

Recognition rates are shown in Table 1. Best recognition rates can be achieved when removing allographs selected by distance or freqency, and chosing the final script model by recognition rate of training data. An average improvement of 6.36 percent could be obtained. Looking only at the test data and selecting the best, even a maximum improvement of 6.74 percent could be possibly reached. So better model selection criteria have to be developed to make full use of this potential.

To compare the different selection and modification strategies for adding and removing models, see Fig. 5 for plots of recognition rate against number of models in the German ZIP recognition system. The adding strategies start from a common point on the left, while removing strategies start from the right. From those plots, the best number of models can be derived easily; in the example, it is around 60 models. Interestingly, when removing models, the recognition rate is at a minimum around 25 models, while the adding strategies are able to generate better configurations for this number of models. So aside from the overall better performance of the removing strategies, adding strategies show to have their advantages, too. A combination of both approaches seems to be reasonable.

6.3. Discussion of selection strategies

An inspection of the images of the final grapheme models gives interesting insights into the properties of the different selection strategies. For the CEDAR address system (letters only) as example, the iterations which give best

[1] The recognition rate on test data is *better* than on training data because the dictionary is smaller.

Project	baseline	add paths by ...			remove paths by ...		
		entropy	likelihood	frequency	distance	likelihood	frequency
Arab Emirates	76.9	75.9 / 79.2	55.7 / 82.4	79.2 / 82.4	85.3 / 85.3	85.3 / 85.3	85.3 / 85.3
Canada (address)	93.4	94.1 / 94.3	93.8 / 93.9	94.6 / 95.0	94.8 / 94.9	95.0 / 95.0	94.8 / 95.0
Germany (address)	92.8	93.2 / 93.3	93.3 / 93.9	92.7 / 93.4	94.2 / 94.2	94.4 / 94.0	94.2 / 94.2
Germany (ZIP)	69.1	70.0 / 71.0	73.8 / 74.7	77.4 / 77.4	76.9 / 77.1	75.3 / 75.0	74.8 / 76.4
USA (address)	81.2	82.8 / 82.8	83.9 / 83.9	81.7 / 81.7	83.5 / 84.2	82.9 / 84.1	83.1 / 84.1
USA (ZIP)	60.8	64.3 / 64.3	68.6 / 68.6	69.6 / 69.5	67.6 / 69.7	66.3 / 68.9	67.7 / 68.9
CEDAR (cities)	84.2	84.6 / 84.6	85.0 / 84.6	82.0 / 82.0	85.5 / 85.3	84.3 / 85.7	84.2 / 85.7
CEDAR (ZIP)	58.2	62.8 / 62.8	65.1 / 65.1	63.0 / 65.5	66.9 / 65.5	67.6 / 64.8	65.5 / 65.7
average	77.1	78.5 / 79.0	77.4 / 80.9	80.0 / 80.9	81.8 / 82.0	81.4 / 81.6	81.2 / 81.9
relative improvement	—	1.82 / 2.46	0.39 / 4.93	3.76 / 4.93	6.10 / 6.36	5.58 / 5.84	5.32 / 6.23

Table 1. Word recognition rates (given in %), chosen by likelihood / recognition rate.

recognition results for test data are selected, and the resulting modelling is discussed.

When adding allographs, selection by entropy gives the best visual impression. "Easy" characters like 'c', or 'u' are modelled only twice, while 'e' and 'r' have a maximum of 8 different variants, reflecting their frequent and variable occurence; no duplicate representation of the same form exists. Adding allographs by frequency results in best recognition performance, but the modeling is not-intuitive: Some models like 'a', 'l', 'n', 'o' are modelled 11 times, producing similar allographs, others like 'g', 'j', 'k' etc. have only one writing variant, not reflecting the complexity of these characters. Closest to the baseline systems are the results from adding by likelihood. In general, the final models have 2 to 4 writing variants, representing the typical writing styles. Complicated or frequent models, like 'x', 'y', 'n' and 'r', consist of up to 8 allographs.

Regarding the iterations where allographs are removed, selection by likelihood gives worst results in all projects, because models with low likelihood are worsened by removing allographs, resulting in many models reduced to one allograph only. Selection by frequency shows similar results as adding by frequency: most freqent characters like 'a', 'e', 'o' are modeled a maximum of five times. Because the baseline system models 6 allographs only, the effect of similar models isn't as strong as when adding. A good visual impression without double representations is obtained by removing by distance, but a few models like 'b', 'k' seem to be overly simplified, by modelling only one writing variant.

7. Summary and outlook

In this paper, an algorithm for automatical adaptation of the number of writing variants in a script recognition system has been presented. An improvement in recognition performance of up to 6.36 percent, compared to the baseline system, could be obtained. Computational cost increase only in training, not during recognition. Several adaptation strategies have been proposed, which show different behaviour. While selection by frequency results in best recognition performance, entropy and distance criteria better reflect the letter complexity. — An adaptation algorithm with alternating steps for adding and removing allographs is envisaged for future work, in order to get the full benefit of the different selection strategies. The combination with a complementary approach to model topology adaptation by adjusting the model length [6] will further improve the recognition performance of the system.

References

[1] C. Bahlmann and H. Burkhardt. Measuring HMM similarity with the Bayes probability of error. In *Proc. of the 6th Int. Conf. on document analysis and recognition*, pages 406–411, Seattle, WA, Sept. 2001. IEEE Computer Society Press.

[2] P. Cheeseman and J. Stutz. Bayesian classification (AutoClass): Theory and results. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 153–180. MIT Press, 1996.

[3] J. J. Hull. A database for handwriting recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, May 1994.

[4] A. Kaltenmeier, T. Caesar, J. Gloger, and E. Mandler. Sophisticated topology of hidden Markov models for cursive script recognition. In *Proc. of the 2nd Int. Conf. on document analysis and recognition*, pages 139–142, Tsukuba Science City, Japan, Oct. 1993. IEEE Computer Society Press.

[5] C. Li. *A Bayesian Approach to Temporal Data Clustering using Hidden Markov Models*. Dissertation, Vanderbilt University, 2000.

[6] M.-P. Schambach. Model length adaptation of an HMM based cursive word recognition system. In *Proc. of the 7th Int. Conf. on document analysis and recognition*, Aug. 2003.

[7] A. Stolcke and S. M. Omohundro. Best-first model merging for hidden Markov model induction. Technical Report TR-94-003, International Computer Science Institute, Berkeley, CA, 1994.