2009 10th International Conference on Document Analysis and Recognition

A Probabilistic Framework for Soft Target Learning in Online Cursive Handwriting Recognition

Xiaoyuan Zhu Yong Ge Fengjun Guo Lixin Zhen
*Motorola Shanghai Lab, No. 11 Building, No. 1257 MingYue Road, PuDong District,
Shanghai, PRC, 201206
{XVKT76, A19118, A16708, A17090}@motorola.com*

Abstract

To develop effective learning algorithms for online cursive word recognition is still a challenge research issue. In this paper, we propose a probabilistic framework to model the inherent ambiguity of cursive handwriting by using soft target vector of each character class. In the proposed algorithm, the values of soft targets are estimated by introducing a lower bound on the log likelihood and optimizing this lower bound via an EM like algorithm. In the experiments on 207K collected cursive words written by 1060 subjects, the proposed algorithm clearly outperforms baseline method with word error reduction up to 11.6%. Furthermore, the estimated soft target values are useful for measuring the separability between output classes.

1. Introduction

Online cursive word recognition [1, 2, 3, 4] is a problem of finding the most probable sequence of characters given a sequence of unsegmented cursive input. It offers a very fast and natural way for the communication between human and computers (or mobile devices). A normal online cursive word recognition system [5] includes three parts: front-end, character recognition, and searching. To recognize a cursive word, front-end module segments the input cursive sequence into strokes and extracts salient features from the strokes. Afterward, character recognition module is performed on the feature sequence to estimate the score of each character. Finally, searching module implements a decoding algorithm based on the character scores under the constraints of a lexicon.

The character based classifier is at the heart of cursive word recognition system. However, because of the highly ambiguous nature of cursive handwriting as illustrated in Figure 1, there is great challenge to learn a character classifier from cursive inputs. In the common method for learning a multi-class character classifier, target vector is set by 1-of-K coding scheme with a target value of "1" in the nth position if the observation falls in class n and "0" otherwise. The target values of "0" and "1" can be considered ideally or hard target values. However, using hard targets for learning can lead to over-fitting. Moreover, the hard target values fail to represent the inherent ambiguity of cursive characters properly. Recently, researchers have proposed many methods [6, 7] to address the above problems by relaxing the pressure towards hard target values during train procedure. The previous results [8, 9] have shown the effectiveness of these methods in cursive word recognition.

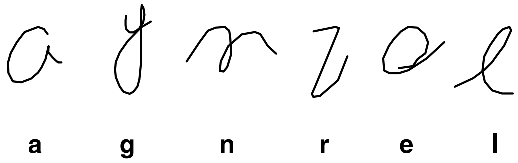

Figure 1. The ambiguity of cursive characters: a/o, g/y, n/m/r, r/v/i, e/o, l/e.

In this paper, we proposed a probabilistic framework to model the inherent ambiguity of cursive inputs for online handwriting word recognition. In the proposed method, the ambiguity of cursive handwriting is modeled by a posterior distribution serving as the soft target vector of each character class. To estimate the value of soft targets, we introduce a lower bound on the log likelihood under Bayesian framework, and maximize this lower bound via an EM like algorithm during training. The proposed algorithm is evaluated on 207K collected cursive words (including letters (both lower and upper cases), digits, and punctuations) written by 1060 subjects. The results show that by modeling the inherent ambiguity of cursive characters properly, the proposed method clearly outperforms baseline method with word error reduction up to 11.6%. Furthermore, the estimated soft targets are useful for measuring the separability between output classes.

The structure of this paper is organized as follows. We derive the proposed probability framework in section two. The experimental results on cursive word

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.6

recognition are analyzed in section three. Conclusions are drawn in section four.

2. The probabilistic framework for online cursive recognition

In this section, we propose a probabilistic framework for online handwriting recognition to handle the ambiguity of cursive characters. First we formulate the learning problem as follows: Let $D=\{x_i,w_i\}_{i=1}^{N_D}$ stand for the learning dataset of N_D independent and identically distributed (i.i.d.) items, where x_i denotes the cursive input of the i^th target word w_i. Let $w_i=\{w_{i1,\ldots}w_{ij,\ldots}w_{iJ}\}$, w_{ij} is the j^th character in the word w_i and J is the total number of characters in w_i. Let $w_{ij}\in\Phi$, where Φ is character label set. For a specified segmentation S (this can be obtained automatically via forced alignment method), let $x_i=\{x_{i1},\ldots x_{ij},\ldots x_{iJ}\}$, where x_{ij} is a feature vector extracted from the j^th segment of the i^th cursive word input. Let θ denote the parameter vector of character based classifier which maps the input space of x_{ij} into label set Φ.

In the proposed method, parameter θ is estimated under Maximum Likelihood criterion as follow:

$$\begin{aligned}\arg\max_{\theta} P(D\mid\theta,S) &= \arg\max_{\theta}\prod_{i=1}^{N_D} P(x_i,w_i\mid\theta,S)\\ &= \arg\max_{\theta}\prod_{i=1}^{N_D}\prod_{j=1}^{J_i} P(x_{ij},w_{ij}\mid\theta)\end{aligned} \qquad (1)$$

where we assume that the character data pairs $\{(x_{ij}\ ,\ w_{ij})\}$ are independent of each other.

To model the ambiguity of cursive input, we further model the probability $P(x_{ij,}\ w_{ij}|\ \theta)$ as follows:

$$\begin{aligned}P(x_{ij},w_{ij}\mid\theta) &= \sum_{c_{ij}\in\Phi} P(x_{ij},w_{ij},c_{ij}\mid\theta)\\ &= \sum_{c_{ij}\in\Phi} P(w_{ij}\mid c_{ij})P(c_{ij}\mid x_{ij},\theta)p(x_{ij})\\ &= \sum_{c_{ij}\in\Phi}\frac{P(c_{ij}\mid w_{ij})P(w_{ij})}{P(c_{ij})}P(c_{ij}\mid x_{ij},\theta)p(x_{ij})\\ &= \frac{1}{Z_{ij}}\sum_{c_{ij}\in\Phi} P(c_{ij}\mid w_{ij})P(c_{ij}\mid x_{ij},\theta) \qquad (2)\end{aligned}$$

where $c_{ij}\in\Phi$ is the hidden variable used to model the inherent ambiguity of cursive input x_{ij}, and Z_{ij} is a normalization constant which can be omitted in the rest derivation. In the last step of equation (2) we assume that the priors $P(w_{ij})$ and $P(c_{ij})$ are uniform distributions.

Substituting equation (2) into (1), we obtain:

$$\begin{aligned}\ln P(D\mid\theta,S) &= \sum_{i=1}^{N_D}\sum_{j=1}^{J_i}\ln P(x_{ij},w_{ij}\mid\theta)\\ &= \sum_{i=1}^{N_D}\sum_{j=1}^{J_i}\ln\sum_{c_{ij}\in\Phi} P(c_{ij}\mid w_{ij})P(c_{ij}\mid x_{ij},\theta)\end{aligned} \qquad (3)$$

It is difficult to directly estimate θ from equation (3). To facilitate the estimation procedure of θ, we derive a lower bound on the log likelihood by introducing an auxiliary distribution Q on the hidden variable c_{ij} as follow:

$$\begin{aligned}&\ln P(D\mid\theta,S)\\ &= \sum_{i=1}^{N_D}\sum_{j=1}^{J_i}\ln\left[\sum_{c_{ij}\in\Phi}\frac{Q(c_{ij}\mid w_{ij})}{Q(c_{ij}\mid w_{ij})}P(c_{ij}\mid w_{ij})P(c_{ij}\mid x_{ij},\theta)\right]\\ &\geq \sum_{i=1}^{N_D}\sum_{j=1}^{J_i}\sum_{c_{ij}\in\Phi}Q(c_{ij}\mid w_{ij})\ln\frac{P(c_{ij}\mid w_{ij})P(c_{ij}\mid x_{ij},\theta)}{Q(c_{ij}\mid w_{ij})}\\ &= F_M\end{aligned} \qquad (4)$$

Then the estimation procedure of θ can be performed by maximizing the lower bound F_M in equation (4) using an EM like algorithm [10]. In the E-step, auxiliary distribution $Q(c_{ij}|w_{ij})$ is estimated by maximizing F_M under the constrain $\Sigma_{cij}Q(c_{ij}|w_{ij})=1$. In the M-step, we estimate class parameter θ by maximizing F_M.

After some manipulation, we obtain the proposed algorithm as follows:

E-step:

$$Q^{(n)}(c\mid w)=\frac{P(c\mid w)\sqrt[N_w]{\prod_{(m,n)\in\{(m,n):w_{mn}=w\}}P(c_{mn}=c\mid x_{mn},\theta^{(n)})}}{\sum_{c\in\Phi}P(c\mid w)\sqrt[N_w]{\prod_{(m,n)\in\{(m,n):w_{mn}=w\}}P(c_{mn}=c\mid x_{mn},\theta^{(n)})}} \qquad (5)$$

M-step:

$$\theta^{(n+1)}=\arg\min_{\theta}\left\{\sum_{i,j}\sum_{c\in\Phi}Q^{(n)}(c\mid w_{ij})\ln\frac{Q^{(n)}(c\mid w_{ij})}{P(c\mid x_{ij},\theta)}\right\} \qquad (6)$$

where N_w denotes the number of characters labeled as w in the segmented training dataset.

In the proposed algorithm, $P(c|x_{ij},\theta)$ serves as the character based classifier modeled by a feed-forward neural network classifier. Auxiliary distribution $Q(c|w)$ describes the ambiguity of the cursive input of character w. In the E-step, $Q(c|w)$ is iteratively estimated by combining both the prior knowledge $P(c|w)$ and the information extracted from the training data. In the M-step, $Q(c|w_{ij})$ serves as the soft targets which supervises the learning process by minimizing the KL divergence between $Q(c|w_{ij})$ and $P(c|x_{ij},\theta)$. Since $P(c|x_{ij},\theta)$ is model as a feed forward neural

Table 1. A comparison of word and character error rate for online cursive recognition.

Error Rate (%)	All Cases		Letters In Vocabulary		Letters OOV 15%	
	Word	Char	Word	Char	Word	Char
Baseline method	22.5±0.2	8.1±0.1	16.5±0.2	5.7±0.1	22.1±0.2	8.3±0.1
Proposed method	21.0±0.2	7.9±0.1	14.6±0.1	5.1±0.1	20.6±0.3	8.1±0.1
Error Reduction	6.5±0.6	2.4±0.9	11.6±0.8	9.6±1.1	6.8±0.8	3.4±1.1

network, the minimization in the M-step can be easily performed by gradient based optimization methods [11]. The convergence of the proposed algorithm can be monitored by using the lower bound F_M as proposed in [12].

Furthermore, it is worth mentioning that if we assume $Q(c{=}w|w)$=1 and all $Q(c\bullet w|w)$=0, then the proposed EM like algorithm degenerates to:

$$\theta^* = \arg\min_{\theta}\left\{-\sum_{i,j}\ln P(c = w_{ij} \mid x_{ij}, \theta)\right\} \qquad (7)$$

which is the traditional training method of neural network with hard target value. For comparison, we take equation (7) as the base line method.

3. Experimental results

To evaluate the proposed algorithm, a series experiments are performed on the handwriting dataset collected by our lab. Our dataset consists of 207K English cursive words including letters (both lower case and capital letters), digits, and punctuations, written by 1060 subjects. We divided the whole dataset into two parts for training and testing with 186K and 21K words respectively. In training dataset, we randomly selected 10K words for evaluation and used the rest data for training. The lexicon used for both the proposed method and the baseline method contains 25000 words. It should be noted that there are no digits and punctuations in the lexicon.

In the proposed algorithm, we use standard softmax output nonlinearity with cross-entropy lose function for the feed-forward neural network classifier. The input layer of our network contains geometric properties of the trajectory and shape features, which are extracted from cursive data. The output layer contains 72 notes including lower case and capital letters, numbers, punctuations and one extra for “blank”. The weights are initialized randomly in the range [-0.2, 0.2]. For the proposed method, the initial value of posterior distribution $Q(c|w)$ is set as follows. For each $w\in\Phi$, we set $Q^{(0)}(c{=}w|w)$=1 and all $Q^{(0)}(c\bullet w|w)$=0. Then an individual M-step is performed to obtain more proper initial values of θ. The prior probability $P(c|w)$ is a key parameters of the proposed algorithm and needed to be set before optimization. The setting of $P(c|w)$ will be discussed in detail latter. For comparison, we also applied the baseline method on the above dataset. In baseline method, we fixed the value of $Q(c|w)$ as discussed in the preview section and optimized classifier under the criterion formulated in equation (7). The setting of neural network in baseline method is the same as in the proposed algorithm.

3.1. Results and comparison

To benchmark the proposed method, all the experiments are repeated four times with random initialization of neural networks. Thus, word error rate and character error rate are stated as the mean ± the standard deviation as listed in Table 1. The character error rate is calculated by using Levenshtein distance. In the first experiment, both algorithms are tested on 11K cursive words containing all the symbols, including lower case and capital letters, digits, and punctuations. The results are listed in the second column of Table 1. From these results we can see that the proposed algorithm clearly outperforms the baseline method by better modeling the inherent ambiguity of cursive inputs. In the second experiment, to show the influence of Out-Of-Vocabulary (OOV) words on the proposed algorithm, we first tested the algorithms on 9K in vocabulary cursive words, containing only lower cases and capital letters. And then we obtained another test set by adding 15% OOV words into the in vocabulary test set. The results on these two datasets are listed in the third and the fourth columns of Table1 respectively. From the results we can see that although increasing OOV rate decreases the

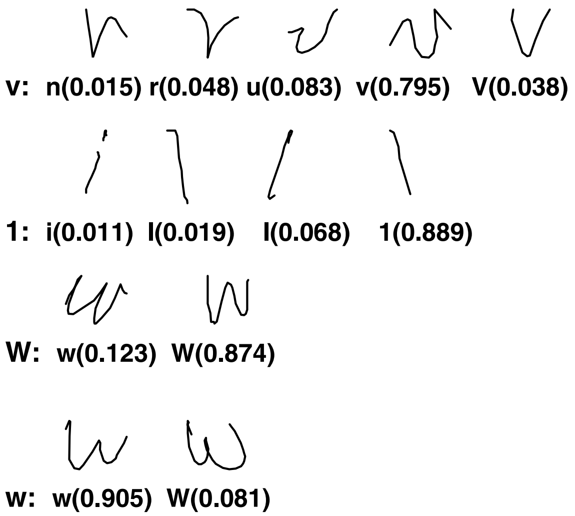
v: n(0.015) r(0.048) u(0.083) v(0.795) V(0.038)
1: i(0.011) l(0.019) I(0.068) 1(0.889)
W: w(0.123) W(0.874)
w: w(0.905) W(0.081)

Figure 2. The soft target values of similar cursive characters.

performance of both methods, the proposed method still achieves 6.8% Word Error Reduction (WER) according to the baseline method. It is also worth mentioning that WER of the proposed algorithm are larger than Character Error Reduction (CER) on all the three datasets. This fact implies that the proposed soft target learning algorithm performs better on word recognition than on character recognition by properly modeling the legitimate ambiguity between characters in training process. Similar results have been previous observed in [8].

Furthermore, in the proposed algorithm, the estimated soft target values (posterior probability) $Q(c|w)$ can be used to measure the separability between output classes. Thus they are helpful to characterize the estimated classifier. In the first line of Figure 2, we list the soft targets label for target character "v" with posterior probability $Q(c|w=$"v") larger than 0.01. We also illustrate the corresponding cursive sample above each target label. In Figure 2, characters with larger soft target value are less distinguishable from the target character according to the estimated classifier. From this example we can see that the cursive input of character "v" is similar with "n", "r", "u", "V" and the ambiguity is properly represented by $Q(c|w=$"v"). From Figure 2 we can also see that there is great ambiguity between the lower and the upper case of some English characters, such as w- W.

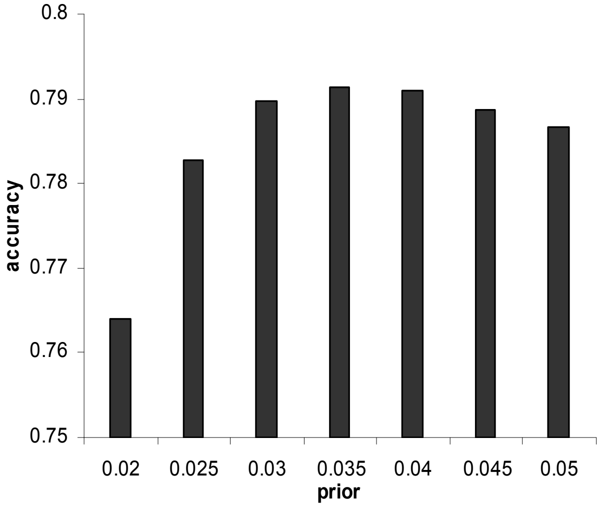

Figure 3. A comparison of recognition accuracy on different initial prior probabilities.

3.2. Influence of prior probability

In the proposed algorithm, the setting of prior probability has significant effect on the recognition performance. As described in section 2, for each $w \in \Phi$ we need to set the prior distribution $P(c|w)$ before training start. In our experiment, $P(c|w)$ is set as follows:

$$P(c = w \mid w) = P_0$$
$$P(c \neq w \mid w) = \frac{1 - P_0}{|\Phi| - 1} \qquad (8)$$

where $|\Phi|$ denotes cardinality of the label set Φ. For the "blank" output of neural network, we set P_0=1. While for other outputs, P_0 are the same constant required to be initialized.

To study the influence of prior probability, P_0 is increased 0.005 at each step from 0.02 to 0.05 (It should be noted that the averaged value for a 72 output neural network is 0.014). For each P_0 we train a classifier by using the proposed algorithm and evaluate it on the test set containing all the symbols. The results are illustrated in Figure 3. From these results, we can see that the classifier reach its best performance at P_0=0.035. Either increase (approaching 0.05) or decrease (approaching 0.02) P_0 will lead to the reduction of recognition accuracy.

To further study this fact, we take character "v" as an example to illustrate the influence of the prior $P(c|w)$ on the posterior $Q(c|w)$ in Figure 4. In Figure 4, x axis denotes the five most similar characters of character "v" (including itself), and y axis denotes the estimated posterior of each similar character. From the results in Figure 4, we can see that if P_0 is too small (e.g. P_0=0.02), the peak of posterior distribution $Q(c|w=$"v") may not focus on $c=$"v". Thus in this case, $Q(c|w)$ does not supervise the

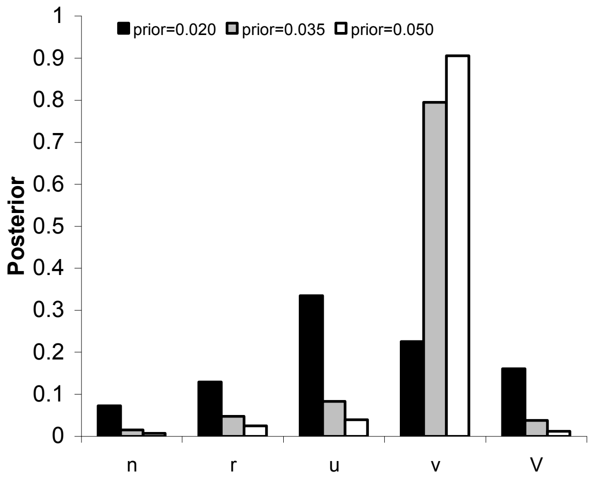

Figure 4. The influence of prior probability on the posterior probability for character v.

learning process well. On the other side, if P_0 is too large (e.g. P_0=0.05), the posterior probability $Q(c|w=\text{“}v\text{”})$ is more centralized on character “v”. However, this makes $Q(c|w)$ fail to model the ambiguous cursive input properly. Thus both large and small P_0 will do harm to the recognition performance which is confirmed by the results in Figure 3.

4. Conclusions

Classification methods play an important role in cursive word recognition system for the great ambiguity of cursive handwriting. However, the common training procedure supervised by the hard target values fails to represent the inherent ambiguity of cursive characters properly. In this paper, we proposed a probabilistic framework to model the inherent ambiguity of cursive inputs using the soft target vector $Q(c|w)$ of each character class. To estimate the values of $Q(c|w)$, we introduce a lower bound on the log likelihood $\ln P(D|\theta,S)$ under Bayesian framework, and maximize the bound via an EM like algorithm during training. We evaluated the proposed algorithm on 207K collected cursive words including lower case and capital letters, digits, and punctuations, written by 1060 subjects. The results show that the proposed method clearly outperforms the baseline method with word error reduction up to 11.6% by modeling the inherent ambiguity of cursive characters properly. Furthermore, the estimated soft targets values can be used to measure the separability between output classes, which is helpful to characterize the estimated classifier.

References

[1]. Rejean Plamondon, Sargur N. Srihari, “On-line and off-line handwriting recognition: a comprehensive survey”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.22. no.1, 2000, pp. 63-84.

[2]. Giovanni Seni, Rohini K. Srihari, Nasser Nasrabadi, “Large vocabulary recognition of on-line handwritten cursive words”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.18. no.7, 1996, pp. 757-762.

[3]. C. Tappert, C. Suen, and T. Wakahara, “The state of the art in online handwriting recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 8, 1990, pp. 787–808.

[4]. Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, “Gradient-Based Learning Applied to Document Recognition”, *Proceedings of the IEEE*, vol.86, no.11, November 1998, pp. 2278-2324.

[5]. Giovanni Seni, Tasos Anastasakos, “Non-cumulative character scoring in a forward search for online handwriting recognition”, *In Proceeding of International Conference on Acoustics, Speech and Signal Processing*, 2000, pp 3450-3453.

[6]. Y. LeCun, L. Bottou, G. Orr and K. Muller, “Efficient BackProp”, in Orr, G. and Muller K. (Eds), *Neural Networks: Tricks of the trade, Springer*, 1998.

[7]. Michael Rimer, Tony Martinez, “Classification-based objective functions”, *Machine Leaning*, vol.63, no.2, 2006, pp. 183-205.

[8]. L. Yaeger, R. Lyon, and B. Webb, “Effective training of a neural network character chassifier for word recognition”, in *Proc. Advances in Neural Information Processing Systems 9*, Cambridge, MA, MIT Press, 1997.

[9]. Xuefeng Chen, Xiabi Liu and Yunde Jia, “A soft target method of learning posterior pseudo-probabilities based classifiers with its application to handwritten digit recognition”, *11th international conference on frontiers in handwriting recognition*, motreal, Canada, August 2008.

[10]. R. M. Neal, G. E. Hinton, “A view of the EM algorithm that justifies incremental, sparse, and other variants”, in *Learning in Graphical Models*, Dordrecht, Kluwer Academic Publishers, 1998, pp. 355-368.

[11]. C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, 1995.

[12]. C. M. Bishop, D. Spiegelhalter, and J. Winn, “VIBES: a variational inference engine for Bayesian networks”, in *Proc. Advances in Neural Information Processing Systems 15*, 2002, pp. 15-22.