On-Line Recognition of Handwriting Mathematical Expression via Network

Y. Sakamoto[*], M.Xie[*], R.Fukuda[†‡] and M.Suzuki[*§]

May 15, 1998

Abstract

The recent developments of technologies on networks and computers are giving us the prospect to various new approaches in both the research and the education of mathematics. For example, an electric white board connected by a network to a computer algebra system or some mathematical data base enables us to give a new style of lectures or discussions on mathematics.

However, it is ture that the user interfaces of the current computer systems are not convenient to input mathematical expression. The most frequently used TEX system, for example, is not adequate here. In order to overcome this difficulty, we are developing a real time recognition system of handwriting mathematical expression.

There are many researchers on the recognition of handwriting characters, but there are few research articles devoted to recognize the characters written on various places and in various sizes like a mathematical expression. The methods of our system is based on the Hidden Markov Model(HMM) for the character recognition, and on the Dynamic Programming(DP) for the segmentations of a sequence of strokes into character units. The system recognizes well constructed mathematical expressions of high school level.

1 Introduction

The explosive growth of the World Wide Web (WWW) has developed the computer network and international communications. Informations and softwares on the computer connected to the Internet can be easily operated by

[*]Graduate Schoos of Mathematics Kyushu University 36, Fukuoka 812 Japan.
[†]Faculty of Engineering Oita University, Oita 870-1192 Japan.
[‡]Email: rfukuda@cc.oita-u.ac.jp
[§]Email: suzuki@math.kyushu-u.ac.jp

the WWW browser using the Hyper Text Markup Language (HTML) and the Java technology. These will enable us effective communications for research and education on WWW. In Japan, there is a plan that all high schools will be connected by the computer network in a few years, and the communication on WWW will be more popular in educational environments.

Various advantages are easily found for this style of communications. For example: Participants of the lectures or the discussions may stay anywhere in the world as long as they can use a computer connected to the Internet. Various informations on WWW from all over the world are available at all times. WWW is suitable for many-to-one and many-to-many mutual communications, and so on. However, at the present time, the user interface is not convenient enough especially for mathematical expressions.

Our primitive two ideas are a electric white board (a free drawing area in the graphical interface of the browser) and the rule of TEX system. The former is familiar and easy but needs large data size when it is stored as an image file. On the other hand, the expression using the TEX rule does not need large data size, but it needs special skill and the expression is not suitable for a communication. Our system, which will solve the above problem, recognize mathematical expressions written on an electric white board and express the results according to the TEX rule. We develop this system as a Java application. Possibly, it also will be a good tool for making a source file of TEXsystem.

Java is a platform for application development adapted to the network, and nowadays, its applications are available to most Internet browser. This guarantees our system will be useful for world wide communications. Java was developed at Sun Microsystems, one of the giants in the computer industry. Computer programs are very closely tied to the specific hardware and the operating systems, and this is a problem with distributing executable programs from web pages. Java solves this problem by using byte code. The Java compiler does not produce native executable code for a particular machine like a C compiler does. Instead it produces a special format called byte code. This looks a lot like machine language, but unlike machine language Java byte code is exactly the same on every platform.

The idea of our system is based on the theory of Hidden Markov Model (HMM). This model is the most popular stochastic dynamical system and is applied to many areas. Especially for speech or handwriting recognition, there is a lot of research which is based on this model [3, 5, 7]. In general, its performance depends on what states the system use. Our object, handwriting mathematical expression, is not cursive but has various sizes and positions which give special meanings to characters or blocks. We therefore choose, as the states of the Markov process, the tangent of the pen movement, the position and the size of the character. We use the method of Baum-Welch [7] for the training and recognition of characters, basic DP method for the

segmentation, and the algorithm of Inoue-Miyazaki-Suzuki [4] for the analysis of a construction of mathematical sentences. Thus we have developed the system which recognizes the well written mathematical expression of the high school level.

2 Model design

In this section we shall describe the modeling of our system as a left-to-right Hidden Markov Model.

2.1 Hidden Markov Model

Hidden Markov Model is determined by a Markov chain (discrete time and finite states Markov process) $\{S_t\}_{t=1}^{\infty}$ and an observation process $\{O_t\}_{t=1}^{\infty}$. Let $\{s_i\}_{i=1}^{N}$ be the set of all states of the Markov process and $\{o_i\}_{i=1}^{M}$ be the observation set. Suppose that S_t describe a real phenomenon like voice or human act. Then usually we can not observe it perfectly. So that the observation is a random variable of this Markov process. Therefore even in the case that the set of states is identical with the observation set, the conditional probability $\Pr(O_t = s_i | S_t = s_i)$ is not always 1.

In this paper we only consider stable HMMs, that is, the transition distribution of $\{S_t\}$ and the conditional distribution of the observation process do not depend on the time. Then the stochastic model (S_t, O_t) determined by the transition matrix $A = \{a_{i,j}\}_{i,j \le N}$ ($a_{i,j} = \Pr(S_{t+1} = s_j | S_t = s_i)$ and the observation matrix $B = \{b_{i,\ell}\}_{i \le N, \ell \le M}$ ($b_{i,\ell} = \Pr(O_t = o_\ell | S_t = s_i)$). The assumption "left-to-right" implies $a_{i,j} = 0$ when $i > j$.

We will estimate A and B using a training data set. Then the recognition and the segmentation are based on the maximum likelihood method with respect to the above distribution. For the detail of HMM see [6, 8].

2.2 The States of HMM

The data obtained from the interface consist of the position (a two dimensional vector), the on-off data of mouse button and the time. Though the raw data of points obtained periodically, the time is a critical information since needless points are removed to eliminate redundancy. Briefly speaking, the point is removed when the angle of successive segments and the length of segment is too small.

A member of the observation set $o_\ell = (o_{\ell,1}, o_{\ell,2}) \in \mathbf{O}$ is a direct product of the on-off data of mouse button ($o_{\ell,1} = 0, 1$) and the direction of segments which is separates into 16 parts ($o_{\ell,2} = 1, \ldots, 16$, see Figure 1). A segment, which consists of elements of $\mathbf{O}$ between two point where the mouse button is pushed and released, is called a *stroke*. When the mouse button is on (or

off), it is called *a pen-down stroke* (resp. *a pen-up stroke*). We ignore the direction $o_{\ell,2}$ for an element of $\mathbf{O}$ which belongs to a pen-up stroke(see Figure 2).

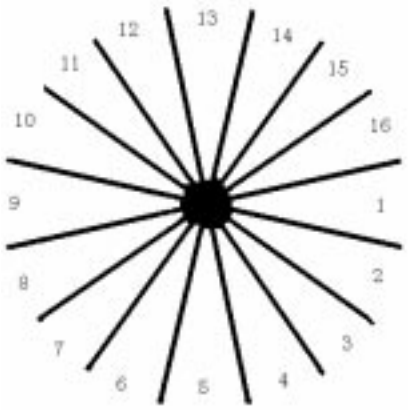

Figure.1 direct code

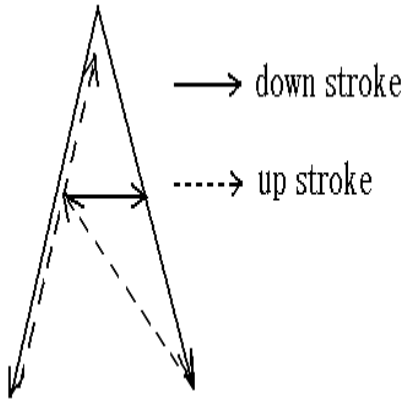

Figure.2 up/down stroke

For each character, we consider its own HMM. While these have same observation sets $\mathbf{O}$, the states are determined individually according to the data set. For a observation sequence in the training data set, they are first divided into strokes. When the frequencies of the observations are not stable enough, the stroke is divided into several states. In general, there may be several sets of states for one character, because a character is not written in the same manner.

We assume that the transition matrix $(a_{i,j})_{1\leq i,j\leq N}$ ($a_{i,j} = \Pr(S_{t+1} = s_j | S_t = s_i)$, it is not depend on t by the assumption) satisfies

$$a_{i,j} = 0, \quad \text{if } i \neq j \text{ and } i \neq j-1 \tag{1}$$

that is, the next state is uniquely determined for each state. We consider several HMMs for one character if it does not satisfy (1).

3 Training and Recognition of one character

3.1 Initial estimation

Our method of making the states also give the information when the state changes. The initial values of the transition matrix and the observation matrix are determined by them. The condition (1) implies that its distribution is determined by Poisson arrival times $\{A_i\}_{i=1}^N$, which represents the staying time, the time between the arrival and the departure, of the process for each state. Then, we have only to estimate the parameters of exponential distributions of $A_i (i = 1, 2, \ldots, N)$ for the estimation of the transition matrix.

When the sample data $a_1, a_2, \ldots, a_k$ of A_i are given, the maximum likelihood estimation of the parameter is

$$\lambda_i = \frac{k}{\sum_{j \le k} a_j} \tag{2}$$

that is, the density function of its distribution is

$$\lambda_i e^{-\lambda_i x} \qquad (x \ge 0) \tag{3}$$

Then the transition matrix is given by

$$a_{i,j} = \begin{cases} 1 - e^{-\lambda_j} & \text{if } i = j \\ e^{-\lambda_j} & \text{if } i = j+1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The observation matrix is estimated by simple frequencies for each states.

3.2 Reestimation

To improve the above estimation, we use the well known algorithm of Baum-Welch [7].

Let $\overline{O}_k = \{\overline{o}_{k,t}\}_{t=1}^{T(k)}$, $(k = 1, 2, \ldots, K)$ be observation sequences obtained from the data sets where $T(k)$ is the length of kth observation sequence. Consider

$$P = \prod_{k=1}^{K} \Pr(O_1 = \overline{o}_{k,t}, \ldots, O_{T(k)} = \overline{o}_{k,T(k)}) \tag{5}$$

which is the probability of the event that all O_k $(k = 1, 2, \ldots, K)$ are actually observed. We will reestimate the parameters to maximize the probability P.

Set

$$\alpha_t^{(k)}(j) = \Pr(O_1 = o_{k,1}, \cdots, O_t = o_{k,t}, S(t) = s_j) \tag{6}$$

$$\beta_t^{(k)}(j) = \Pr(O_{t+1} = o_{k,t+1}, \cdots, O_{T(k)} = o_{k,T(k)} | S(t) = s_j) \tag{7}$$

Then, the reestimation $\{\overline{a}_{i,j}\}_{i,j \le N}$ and $\{\overline{b}_{i,\ell}\}_{i \le N, \ell \le M}$ of the transition matrix $\{a_{i,j}\}_{i,j \le N,}$ and the observation matrix $\{b_{i,\ell}\}_{i \le N, \ell \le M}$ are given by

$$\overline{a}_{i,j} = \frac{\displaystyle\sum_{k=1}^{K} \sum_{t=1}^{T(k)-1} \alpha_t^{(k)}(i) a_{i,j} b_{j,o_{k,t+1}} \beta_{t+1}^{(k)}(j)}{\displaystyle\sum_{k=1}^{K} \sum_{t=1}^{T(k)-1} \alpha_t^{(k)}(i) \beta_t^{(k)}(i)} \tag{8}$$

$$\overline{b}_{j,\ell} = \frac{\displaystyle\sum_{k=1}^{K} \sum_{\overline{o}_{k,t}=o_\ell} \alpha_t^{(k)}(j)\beta_t^{(k)}(j)}{\displaystyle\sum_{k=1}^{K}\sum_{t=1}^{T(k)} \alpha_t^{(k)}(j)\beta_t^{(k)}(j)} \tag{9}$$

Then we can increase the probability P. After several iteration, we stop the reestimation when the increment gets small enough.

To decrease the amount of the calculation, $\alpha_t^{(k)}(j)$ and $\beta_t^{(k)}(j)$ are calculated inductively, and for $\beta_t^{(k)}(j)$ the induction execute from the opposite side. These values often gets too small, then it needs some scalings to avoid underflows.

Set $c_t^{(k)} = (\sum_{j=1}^{N} \alpha_t^{(k)})^{-1}$, and $\alpha_t^{(k)}(j)$ or $\beta_t^{(k)}(j)$ is replaced by $c_t^{(k)}\alpha_t^{(k)}(j)$ or $c_t^{(k)}\beta_t^{(k)}(j)$ respectively, as soon as they are obtained during the induction. Then, the ratio of new $\alpha_t^{(k)}(j)$ (or $\beta_t^{(k)}(j)$) and old one is $\prod_{s\le t} c_s^{(k)}$ (or $\displaystyle\prod_{s>t} c_s^{(k)}$ respectively). So that the above value by a reestimation is uninfluenced of this scaling (for details see [7]).

3.3 Recognition

We have obtained well estimated HMMs for every characters. Let us consider a given observation sequence which will configure one character. Then, for all HMMs corresponding to characters, we calculate the probability of the event that this sequence appears in actual and it reachs to the final state of the model. Then the character which attains the maximum of the above probabilities is the result of the system. We can not remove the assumption reaching to the final states. If it is removed, the data of 'V' may return a high probability under the HMM with respect to 'W'.

4 Segmentation

A Mathematical expression is given as a sequence of strokes. Then constitutive characters are given as its subsets. We will explain the way to find the optical segmentation which divide a mathematical expression into characters. This method is fundamentally based on the method of Dynamic Programing.

4.1 Dynamic Programing

To begin with, we generalize our segmentation. Let $\{\sigma_1, \sigma_2, \ldots, \sigma_J\}$ be a sequence of nodes and $\Phi = \{\phi_k\}_{k=1}^{K}$ be an arbitrary segmentation. The following (a)$\sim$(c) are detailed account for the segmentation:

(a) An element ϕ_k $(k \leq K)$ is called an arc, and it consists of two nodes $(\phi_k = (\sigma_{j(k,1)}, \sigma_{j(k,2)})$, $j(k,1) < j(k,2))$.

(b) ϕ_k and ϕ_{k+1} are connected, that is, $j(k,2) = j(k+1,1)$ for any $k \leq K$.

(c) $j(1,1) = 1$ and $j(K,2) = J$.

For our system, we choose a pen-up stroke as a node. Adding a pen-up stroke at the head or the end of a sequence of strokes, we may regard a character as an arc.

Consider a cost function $f(\Phi)$ which give a grade for Φ as a mathematical expression, that is, the better mathematical expression the segmentation Φ is, the larger the cost function $f(\Phi)$ is.

Let Φ be a segmentation which attains the maximum of f, that is,

$$f(\Phi) = \max\{f(\Psi) | \Psi : \text{a segmentation of } \{\sigma_1, \sigma_2, \ldots, \sigma_J\}\} \qquad (10)$$

and set $\overline{\Phi} = \{\phi_k\}_{k=K_1}^{K}$ where K_1 is an arbitrary positive number. Then, for the consistency, we assume that $\overline{\Phi}$ attains the maximum of f among the segmentations of $\{\sigma_{j(K_1,1)}, \sigma_{j(K_1,1)+1}, \ldots, \sigma_J\}$. For the cost function f satisfying the above property, the DP-system find the optimal segmentation Φ (see [1, 2] for detail). So that the remaining problem is to define a good cost function.

4.2 Cost function and Details

For our system, likelihoods are main informations for the cost function. However a careless estimation often make it an inadequate one. For example, we consider the event E which satisfy $\Pr(E) = \frac{4}{5}$. Then probability of the event that E is observed 10 times without a break is $(\frac{4}{5})^{10} = 0.1073\ldots$ when the 10 trials are stochastically independent. Comparing samples of ten times trials, this value is the largest probability. But the probability of 'E does not happen' is $\frac{1}{5} = 0.2$, which is greater than the above value. This may imply that the probability of "the system returns correct answer 10 times" is less than the probability of "the system return wrong answer once".

Then likelihoods for the cost function are determined according to the following requirements.

(a) Each stroke has its own HMM, and indistinctive strokes are identified.

(b) For each HMM, the Markov process is adjusted according to the number of step of the object, and the likelihood is normalized according to actual frequencies.

(c) For each character, we have the information of strokes which the character consist of. When a set of strokes is found among them, the likelihood for the set is given as a weighted geometrical average.

(d) For a set of characters, we also consider a weighted geometrical average.

(e) Time and position informations are used for secondary estimations, since strokes are not enough data to recognize a character. For example, "E" and "F-" or "H" and "1-1" are indistinctive by strokes.

Thus we obtain a cost function $f(\Phi)$ which will return large value when $\Phi = \{\phi_k\}$ is a good segmentation.

Suppose that $\{\phi_k\}_{k=1}^{K}$ is the best segmentation. Then, for any $K_1 < K$, $\{\phi_k\}_{k=K_1}^{K}$ is also the best segmentation for corresponding sequence of strokes. According to this idea, the DP-method determine the best segmentation inductively from the end side of the sequence as follows.

Let $\{\sigma_j\}_{j \leq J}$ be a given sequence of nodes and J_1 be an arbitrary positive number such that $J_1 < J$. Suppose that each Φ_r $(r = J_1 + 1, J_1 + 2, \ldots J)$ is the best segmentation for $\{\sigma_j\}_{j=r}^{J}$. Then the best segmentation is found among

$$\{[\sigma_{J_1}, \sigma_r] \cup \Phi_r\}_{r=J_1+1}^{J} \tag{11}$$

Thus the best segmentation of $\{\sigma_j\}_{j \leq J}$ is obtained inductively.

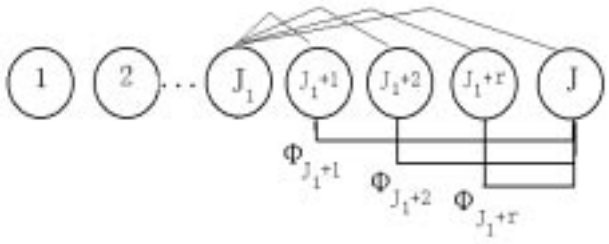

Figure.3 segmentation

However our objects, sequences of strokes, are not always observed correctly. We therefore choose the best segmentation among combinations of at most five candidates for each stoke. Then slight errors of recognition of strokes and characters will be cleared away by this segmentation.

4.3 Analysis of Construction

The system of Inoue,Miyazaki and Suzuki [4] recognizes a printed mathematical expression. For the analysis of constructions, the key data are a result of recognitions as a character, position and size. Our system obtains these data by a stocastical way. This is also a most likelihood estimation, the model of which is a simple Markov chain and the distribution is determined by training data sets.

References

[1] R.Bellman, "Dynamic Programming", *Princeton Univ. press 1957*

[2] R.A.Howard, "Dynamic Programming and Markov Processes", *Wiley, 1960*

[3] Jianying Hu, Michael K.Brown and William Turin, "HMM Based On-line Handwriting Recognition", *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, VOL.18, NO.10,OCTOBER 1996

[4] K.Inoue, R.Miyazaki and M.Suzuki, "Optical Recognition of Printed Mathematical Documents", *ATCM98*

[5] B.H.Juang and L.R.Rabiner, "Hidden Markov Models for Speech Recognition" *TECHNOMETRICS*,AUGUST 1991, VOL 33 NO.3

[6] S.E.Levinson, L.R.Rabiner and M.M.Sondhi, "An Introduction to the Application of the Theory of Probabilistic Function of a Markov Process to Automatic Speech Recognition", *The Bell System Technical Journal*, (Vol.62,No.4, April 1983)

[7] L.R.Rabiner, "A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition", *Proceeding of the IEEE*,Vol.77,No.2, February 1989

[8] L.R.Rabiner and B.H.Juang "An Introduction to Hidden Markov Models", *IEEE ASSP MAGAZINE*, January 1986

[9] T.Wakahara and K.Odaka, "On-Line Cursive Kanji Character Recognition Using Stroke-Based Affine Transformation", *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* VOL.19, NO.12,DECEMBER 1997