A System for Recognizing Online Handwritten Mathematical Expressions and Improvement of Structure Analysis

Anh Duc LE, Truyen Van PHAN, and Masaki NAKAGAWA

Department of Computer and Information Sciences
Tokyo University of Agriculture and Technology, Japan
{leducaanh841988, truyenvanphan}@gmail.com
nakagawa@cc.tuat.ac.jp

Abstract—This paper presents a system for recognizing online handwritten mathematical expressions (MEs) and improvement of structure analysis. We represent MEs in Context Free Grammars (CFGs) and employ the Cocke-Younger-Kasami (CYK) algorithm to parse 2D structure of on-line handwritten MEs and select the best interpretation in terms of symbol segmentation, recognition and structure analysis. We propose a method to learn structural relations from training patterns without any heuristic decisions by using two SVM models. We employ stroke order to reduce the complexity of the parsing algorithm. Moreover, we revise structure analysis. Even though CFG does not resolve ambiguities in some cases, our method still gives users a list of candidates that contain expecting result. We evaluate our method in the CROHME 2013 database and demonstrate the improvement of our system in recognition rate as well as processing time.

Keywords—online recognition; handwritten mathematical expressions; 2D Context Free Grammar; CYK parsing algorithm

I. INTRODUCTION

In recent years, devices such as tablets, PDAs and electronic white boards are becoming more and more popular. They allow users to annotate on documents, draw figures, write mathematical expressions, etc. more naturally, easily than traditional PCs with keyboard and mouse. A new type of electronic pen and paper devices such as Anoto pen is also available to input handwriting. Researchers have proposed many approaches to recognize handwriting in the latter half of the 20th century.

Recognition of handwritten MEs can be divided into three major processes: symbol segmentation, symbol recognition, and structure analysis. Firstly, a sequence of input strokes is segmented into hypothetical symbols. Then, each hypothetical symbol is recognized by a symbol classifier. Finally, structure analysis is made by parsing recognized symbols to determine the most likely interpretation as a ME. Handwritten ME recognition is one of the current challenges in handwriting recognition. It requires complex tasks of symbol segmentation, recognition, structure analysis, context analysis and total likelihood evaluation. Some local ambiguities can be solved by context, while the other ones cannot be solve even using context. In this case, we must verify semantics of MEs. This verification process is so complicated and need much more computational time. Instead of using the semantics of MEs, we have a simple way that gives users a list of candidates. For example, Fig. 1(a) gives an example of local ambiguities that can be solved by context. The first expression is “$z_1 + z_2$”, while the second expression is “1.23”. Our method tries to make both of them appear in the list of candidates.

(a) $\hat{Z}_1 + \hat{Z}_2 \quad \hat{1.23}$

(b) $(1+2)(a+b)=3a+3b$

Figure 1. Examples of Local ambiguities.

Zanibbi et al. proposed a method based on tree transformation [1]. Input symbols are processed in three passes. The first pass constructs a Baseline Structure Tree describing the two-dimensional arrangement of input symbols. The second pass groups tokens comprised of multiple input symbols such as decimal numbers and function names. The last pass analyzes expression syntax and produces operator tree that is suitable for computation.

Recently, Stochastic Context-Free Grammars (SCFGs) have proven to be useful in recognizing MEs. Yamamoto et al. took an approach of formulating the recognition problem as a search problem of the most likely ME candidate [2]. They used stochastic CFGs in order to model handwritten MEs and the CYK algorithm to parse the input strokes. Structure probability is calculated by a concept called Hidden Writing Area. The experiment showed that the result was improved in different grammatical constraint levels.

MacLean et al. presented a method for recognizing MEs by using a top-down parsing algorithm [3]. In their approach, the incremental parsing process constructs the shared parse forest that presents all recognizable parses of the input. Then, the extraction process finds the nth-most highly-ranked tree from the forest. They provided correction mechanism to help users to edit recognition errors.

Alvaro et al. proposed a formal model for on-line handwritten MEs recognition based on 2D-SCFGs and Hidden
Markov Models (HMM) [4]. HMM used both online and offline features to recognize mathematical symbols. The CYK algorithm was chosen to implement structure analysis. Support Vector Machine (SVM) helped to learn geometric features between bounding boxes for determining structural relations.

Awal et al. introduced a global approach allowing learning mathematical symbols and structural relations directly from expressions [5]. During training phase, symbol hypotheses are generated without the language model. The dynamic programming algorithm finds the best segmentation and recognition of the input. The classifier learns both of the correct and incorrect segmentations. This process is repeated to update the classifier. Gaussian models are used for modeling structural relations. The differences in position and size of two bounding boxes are applied for the purpose of estimating the Gaussian parameters.

These three systems took the promising results in the Competition on Recognition of Handwritten Mathematical Expressions (CROHME). The system by Alvaro et al. won the first price in CHROME 2011 and got second price in CHROME 2013. The systems by MacLean et al. and that by Awal et al. received the second and third prices in CHROME 2012, respectively.

We are now working on a self-learning system for elementary, junior-high, and high school students to learn math and making a MEs recognition system to recognize and mark handwritten MEs. For this purpose, we need to develop an automatic handwritten ME recognition system. The systems which have joined CHROME show high recognition rates but there remain large problems for handwritten ME recognition to be employed in real environment of education. The recognition speed and memory size are practically important. Moreover, if the correct answer is within the candidates even if the top candidate is not correct, a user can select it without rewriting. Thus, the cumulative recognition rate is important.

In this paper, we present an approach to solve local ambiguities through applying stochastic 2D CFGs. The recognition system optimizes simultaneously symbol segmentation, recognition, and 2D structure. We propose to use two SVM models to determine structural relations to improve recognition rate. Furthermore we employ stroke order to achieve smaller time complexity than the above two methods [3, 4]. Through the experiment, we prove that the recognition rate and processing time of our method are improved. The rest of this paper is organized as follows. Section 2 introduces an overview of our system and processes of symbol segmentation/recognition, extraction of structural relation, and 2D structure analysis. Section 3 shows an evaluation of this system and section 4 concludes the paper.

II. SYSTEM OVERVIEW

This section describes an overview of the system. The handwritten MEs recognition problem is formulated as a search problem of the most likely interpretation of handwritten strokes. Then, we model the search problem as the following formula:

$$P = \prod P_{\text{seg}}(S_j, S_{j+1}) \times \prod P_{\text{rec}}(S_j) \times \prod P_{\text{rel}}(R_k | SE_j) \times \prod P_{\text{Gram}}$$  (1)

Where $P_{\text{seg}}(S_j, S_{j+1})$ is the probability of separation between stroke $j$ and $j+1$. $P_{\text{rec}}(S_j)$ stands for the probability that a group of strokes $G_i$ is recognized as the symbol $S_j$. $P_{\text{rel}}(R_k | SE_j)$ is the probability of 2 sub-expressions are combined into a larger expression in relation $R_k$. $P_{\text{Gram}}$ is the production probability in the grammar. The detail of these probabilities is described as follows.

A. Symbol Segmentation

In our system, we compute the separation probability of each pair of adjacent strokes and generate all symbol hypotheses of MEs in the segmentation process. The separation probability is calculated from 12 geometric features shown in [6, 7] and 9 additional features. Consequently, we extracted 21 geometric features such as minimum distance of adjacent strokes/H, length of off-stroke/H, overlap area projected onto the x and y axes/H, overlap between 2 bounding boxes/S1, distance between centers/H and so on, where H and S1 are the average height of ME and the area of the bounding box for the first stroke, respectively. We use a SVM classifier for segmentation. From the distribution of the output score of the SVM classifier, we could set a threshold $T$ so that a pair of strokes is segmented if its score $t$ is higher than $T$ and otherwise if the score is lower. However, this hard decision does not allow the later processes to recover from error decisions. The figure 2 shows the distribution of the SVM output score for training patterns. We set the segmentation $T_s$ and non-segmentation threshold $T_{ns}$ to divide the output score to 3 areas: segmentation (SP), non-segmentation (NSP) and uncertain segmentation (US). The separation probabilities in SP and NSP are 1 and 0 respectively. In US area, we use the Sigmoid function to transform the SVM score to probability.

![Figure 2. Distribution of SVM output scores.](image)

From the SVM output score, we generate groups of strokes called symbol hypotheses. We discard invalid hypotheses by the following constraints:
- Employ SP and NSP areas to avoid invalid hypotheses.
- Restrict the maximum number of strokes as 4 (the largest
number for math symbols) plus 1, considering an additional or separated stroke.  
- Reject hypotheses having recognition score too low.

B. Symbol Recognition

After generating symbol hypotheses, we recognize each symbol hypothesis by a character recognizer which combine offline and online recognition methods [8]. This combination is robust to stroke connections and cursive strokes due to the elastic matching by the online method and also robust to stroke disorders or duplicated strokes due to the image matching by the offline method.

For ME recognition, however, the original recognizer for Japanese characters which are more than 8,000 categories has been modified. First, we have removed compression of its dictionaries and coarse classifier, which have been employed to reduce the memory size and computation time, since they are not necessary for a small set of symbols for ME’s, which are less than 100. The resultant memory size for the handwritten ME recognition is 1.38 MB for 100 categories. Second, we keep only 3 candidates for each symbol pattern although more candidates have been kept for later stages in Japanese text recognition. Third, we introduce an auxiliary process to recognize ‘dot’ and ‘comma’. We treat them uniformly here and discriminate them in the later stage by the grammars.

For each symbol recognition, we add another candidate of ‘dot/comma’, if both the height and width are less than half of the average height and width. Its likelihood is calculated as follows:

\[ P_{\text{dot/comma}} = F(\text{height}) \times G(\text{width}) \]  

Where F and G are fuzzy functions shown in Figure 3. If the height of the bounding box is less than H/10, then F = 1. Otherwise, F decreases linearly to 0 at H/2. The function G is similar for the width.

On the other hand, the likelihoods of other candidates are derived from symbol recognition scores.

The reasons of this process are as follows:
1) These symbols are often misrecognized when they are normalized into a standard size.
2) Distinction between ‘dot’ and ‘comma’ is difficult and somehow meaningless since they can be distinguished in the grammars.

\[ F(h) \]  

\[ G(h) \]  

\[ 0 \text{ H/10} \]  

\[ H/2 \]  

\[ 0 \text{ W/10} \]  

\[ W/2 \]  

Figure 3. Membership Function for ‘dot/comma’.

C. Structural Relations.

The relation in MEs are sometimes ambiguous even for humans. For example, Figure 4(a) shows the case that bounding boxes are similar, but relations are different. Moreover, handwritten symbols are written in various size and shape. To estimate structural relations, bounding boxes were often employed but their problems were pointed out in [2]. We follow the concept of Hidden Written Area (HWA) which was proposed to represent logical relation more stably in [2], but we simplify the concept by assuming a body box for each symbol, which decrease side effect of ascendant and descendant strokes. First, we classify symbols into 4 groups: ascendant, descendant, normal and big symbols. Ascendant or descendant symbols extend above the mean line or below the base line. Normal symbols fall between the mean and base lines. Big symbols extend both the mean and base lines as shown in 4(b). For each group, we assume a body box which includes the main body of each symbol. Therefore, the body box is different for each symbol recognition candidate. Figure 4(c) shows them recognition candidates ‘d’ and ‘a’.

To determine the structural relation between two sub-expressions, there are some approaches presented in [3, 4, 5]. MacLean et al. presented a fuzzy logic method. The membership function is defined based on the distance, angle and amount of overlap between the bounding boxes. Mouchère used a Gaussian model for calculating structural relation probability. This model is based on the differences of the baseline position and relation of height between sub-expressions. However, both of the two approaches must define several thresholds to identify structural relation. Another approach presented by Alvaro et al. learns spatial relations from training patterns without heuristic decision. They used 9 geometric features to classify relations. However, distinguishing some of relations do not need all 9 features, we prune redundant features. Figure 5 presents the distribution of feature Dx (relation of horizontal center between 2 body boxes for sub-expressions). We realize that the above, below, and inside relations are distinguished clearly with the horizontal, superscript, and subscript relations. Therefore, we use Dx feature to divide relations into 2 groups: Group1 (above, below, inside) and Group 2 (horizontal, superscript, subscript). Then, we build 2 SVM models to get relative score of the structural relations within each group. We use the H, Dy features for training SVM of Group 1 and the H, Dy and O features for training SVM of Group 2. All of them are defined in Fig. 6.
is a finite set of relations between 2 sub-expressions, is a distinguished start symbol. For each rule (P → ...)
production rules to reduce it further to another
→ k,j production rules to reduce 2 sub-MEs to a
= i,j = i+j
→ does not satisfy constraints of rejecting invalid hypotheses
for each production X → a
if(prob > 0 ){
Add node (X → a, P_{seg}(f|α) * P_{seg}(f) into cell(i,j)

Parsing stage: CYK operates only on SCFGs given in CNF. To our grammar, however, more production rules (X → A)
have been added, so that we have to modify the original CYK. We employ X → r A B production rules to reduce 2 sub-MEs to a
non-terminal. Then, from the reduced non-terminal, we employ X → A production rules to reduce it further to another
non-terminal. At each cell in the CYK table, we store 40 best candidates for a ME. The candidates of the final result are
extracted at cell (i,N).

For i = 2...n
For j = 1...n - i
For k = 1 .. l
For each rule (X → r A B)
C1 = getCell ((k,j)
C2 = getCell((i-k, j+k+1)
prob = P(C1| A) * P(C2| B) * P_{seg}(C1, C2| r)
* P_{seg}(s_{i+k+1}) * P_{gram}(X → r A B)
if(prob > 0 ){
Add node (X → r A B, prob, C1, C2) into cell(i,j)
}
For each rule (X → A)
if cell(i,j) has node A{
Add node (X → A, P_{s} into cell(i,j)
}

The complexity of this algorithm is still O(n^3|P|) like the original CYK whereas the complexity of the method by Alvaro et al. [4] and that by Maclance et al. [3] are O(n^3 log n|P|) and O(n^3|P|), respectively. We used stroke order to arrange strokes instead of traveling all strokes in the parsing stage. On the other hand, we must prepare all particular writing orders such as those before and after fractions, roots, and parentheses in the grammar to avoid parsing failure.

III. EXPERIMENTS

For evaluating our recognition system, we use the CROHME 2013 database. CROHME 2013 is a contest in online handwritten recognition, organized at ICDAR 2013 [9]. It contains 8,836 MEs for training and 761 MEs for testing, selected from 5 different MEs databases. We participated in CROHME 2013 with 8 other systems. The experiments are implemented on an Intel(R) Core(TM)2 Dual E8500 CPU 3.16GHz with 2.0 GB memory.
The first experiment is to compare the symbol segmentation performance of the two methods which using 1 threshold (Method 1) and 2 thresholds (Method 2), respectively. For Method 1, we choose default threshold of SVM $T = 0.0$. For Method 2, from training patterns, we choose $T_{s} = -1.0$ and $T_r = 1.5$. Table 1 shows that Method 2 has significantly improved the result of segmentation.

<table>
<thead>
<tr>
<th>Method</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method 1</td>
<td>56.27</td>
<td>48.10</td>
</tr>
<tr>
<td>Method 2</td>
<td>81.95</td>
<td>83.49</td>
</tr>
</tbody>
</table>

The second experiment is to evaluate the performance of determination of the structural relation. From the CROHME 2013 database, we extracted 78,802 relations for training and 5,844 relations for testing. We implemented the method of Alvaro et al. to compare with our method. We test the classification rate of our method with changing the $D_x$ parameter from 0.2 to 0.6. Table 2 shows the classification result where their method produces 86.87% while our method achieves the maximum rate being 95.85% at $D_x = 0.4$.

<table>
<thead>
<tr>
<th>Method</th>
<th>$D_x$</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alvaro et al.</td>
<td>~</td>
<td>86.87</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>95.78</td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td>95.80</td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>95.85</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>95.70</td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>95.60</td>
</tr>
</tbody>
</table>

Table 3 presents 4 measurements including: Sym Seg as symbol segmentation, Sym Seg + Rec as symbol segmentation and recognition, Rel Tree as relation tree, and Exp Rec as expression recognition in CROHME 2013. Our system is ranked third in this contest among 8 participants with 19.97% accuracy in expression recognition. System VII got the best for Exp Rec using their own training patterns. Except the system VII, the system IV got the best for the CROHME database.

<table>
<thead>
<tr>
<th>System</th>
<th>Sym Seg</th>
<th>Seg+Class</th>
<th>Rel Tree</th>
<th>Exp Rec</th>
</tr>
</thead>
<tbody>
<tr>
<td>VII</td>
<td>97.86</td>
<td>93.03</td>
<td>88.65</td>
<td>60.36</td>
</tr>
<tr>
<td>IV</td>
<td>84.97</td>
<td>73.94</td>
<td>49.73</td>
<td>23.40</td>
</tr>
<tr>
<td>II</td>
<td>80.70</td>
<td>66.41</td>
<td>22.44</td>
<td>19.97</td>
</tr>
<tr>
<td>III</td>
<td>85.24</td>
<td>62.63</td>
<td>53.24</td>
<td>9.39</td>
</tr>
<tr>
<td>VI</td>
<td>57.86</td>
<td>47.68</td>
<td>33.63</td>
<td>8.35</td>
</tr>
<tr>
<td>I</td>
<td>46.93</td>
<td>25.19</td>
<td>24.85</td>
<td>2.68</td>
</tr>
<tr>
<td>VIII</td>
<td>90.32</td>
<td>73.84</td>
<td>50.19</td>
<td>18.33</td>
</tr>
<tr>
<td>V</td>
<td>84.45</td>
<td>66.66</td>
<td>41.34</td>
<td>14.31</td>
</tr>
<tr>
<td>Improved</td>
<td>81.95</td>
<td>67.06</td>
<td>49.12</td>
<td>20.72</td>
</tr>
</tbody>
</table>

Our system was very low in Rec Tree because we used heuristic rules to identify relations at that time. After competition, we improved symbol segmentation and structure analysis as described in Section II A and II C. Then, Rec Tree and Exp Rec have been improved to 49.12% and 20.72%, respectively. The accuracy of top 5 candidates reaches 41.13% in Exp Rec.

The average processing time per one ME is 0.32 seconds. Although there is no information about processing time to compare with other methods, we believe that it is small enough to be employed in real applications.

IV. CONCLUSION

In this paper, we presented an approach to recognize online handwritten mathematical expressions and solve local ambiguities in symbol segmentation/recognition and structure analysis by using SCFGs. In the case when SCFGs do not solve ambiguities, our method tries to give a list of candidates that contains correct result. Our method can learn symbol segmentation and structural relation from training patterns to improve recognition rate. Moreover, we employ stroke order to reduce the complexity of the parsing algorithm. Experimental results show that the recognition rate and processing time are improved.

ACKNOWLEDGEMENT

This research is being supported by Grant-in-Aid for Scientific Research under the contract number (B) 24300095 and (S) 25220401. The authors would like to thank Dr. Bilan Zhu and other members for availing the symbol recognizer and Mr. Cuong Tuan Nguyen and Mr. Khanh Minh Phan for giving helpful comments and suggestions.

REFERENCES