Statistical classification of spatial relationships among mathematical symbols

Walaa Aly, Seiichi Uchida
Department of Intelligent Systems, Kyushu University
744 Motooka, Nishi-ku, Fukuoka-shi, Japan
walaa,uchida@human.is.kyushu-u.ac.jp

Akio Fujiyoshi
Department of Computer and Information Sciences, Ibaraki University
4-12-1 Nakanarusawa, Hitachi, Ibaraki, 316-8511, Japan
fujiyosi@mx.ibaraki.ac.jp

Masakazu Suzuki
Department of Mathematics, Kyushu University
6-10-1, Hakozaki, Higashi-ku, Fukuoka-shi, Japan

Abstract

In this paper, a statistical decision method for automatic classification of spatial relationships between each adjacent pair is proposed. Each pair is composed of mathematical symbols and/or alphabetical characters. Special treatment of mathematical symbols with variable size is important. This classification is important to recognize an accurate structure analysis module of math OCR. Experimental results on a very large database showed that the proposed method worked well with an accuracy of 99.57% by two important geometric feature relative size and relative position.

1 Introduction

Automatic recognition of mathematical expressions is considered as a basic process in converting scientific and engineering documents into an electronic form. This process is composed of two parts; structure analysis and recognition of mathematical symbols. There have been many attempts to recognize mathematical documents, an overview of previous attempts are found in [1].

In this paper, we consider the structure analysis part, that is, the automatic classification of spatial relationships between each adjacent pair (hereafter, simply called *discrimination task)*. The spatial relationships are composed of baseline, subscript, superscript, upper, and lower relations.

Determination of spatial relationships is very important to recognize mathematical expressions because the same set of symbols convey different meaning depending on the spatial relationships. For example, "ab", "a_b", "a^b" have the same symbols but introduce different meaning.

Throughout this paper, we assume the correct category is given for every characters and symbols; that is, we assume that recognition of characters and mathematical symbols has done already. This assumption is rather realistic when we focus on the structure analysis part; in most math OCRs, in fact, the structure analysis is done after recognizing individual characters and symbols.

The structure analysis part is discussed by many researchers started from Anderson [7], who used a purely syntactic approach for parsing mathematical expressions. Okamoto and his colleagues [8, 9] determined the spatial relationships using geometric informations. Zanibbi etal. [10] presented a system for recognizing typeset and handwritten mathematical expressions. They used classes of symbols to determine the relationships. Suzuki etal. [11] introduced an OCR system called INFTY to recognize mathematical expression. They used geometric features to determine the spatial relationships. Garaian and Chaudhuri [12, 13] proposed an approach for understanding mathematical expressions, in which they used geometric information to determine spatial relationships.

Unfortunately, most of the previous attempts did not give details about the discrimination task (e.g. [2, 3, 4, 5, 6]); they gave only the total performance of the system and specified neither quantitative nor qualitative analysis of their results. This may be because (i) the discrimination task is one module of a large math OCR system, (ii) it employs many heuristics whose details are often hidden from readers, and

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.90

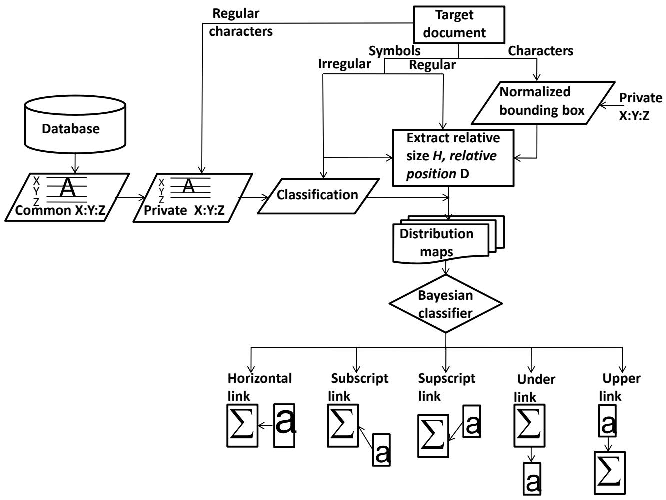


Figure 1. The proposed method.

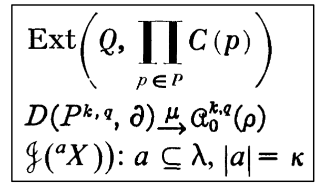


Figure 2. Examples of mathematical expressions.

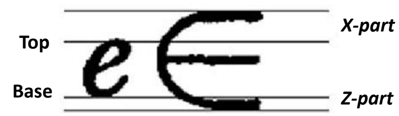


Figure 3. Detection the type of symbol "∈".

(iii) it should be evaluated with a large-scale database.

The main contribution of this paper is to tackle the discrimination task by a statistical decision method grounded by huge database. In the proposed method, the importance of using document-dependent processing and symbol types will be fully emphasized.

In this method we will use two features called *relative size* and *relative position*. These features are very important to specify the spatial relationships. For example, the relative size H is used to discriminate between baseline and non baseline classes and the relative position D is used to discriminate among subscript, superscript, upper, and under classes. Experimental results revealed that the discrimination can be done almost perfectly (~ 99.57%).

Figure 1 shows an overview of the proposed method. Our task is the classification of spatial relationships between each adjacent pair (parent-child), where parent is the first symbol of each pair and child is the second symbol. The features relative size H and relative position D will form distribution maps which plot the features in a two dimensional space. The spatial relationships are determined using Bayesian classifier which classifies each point in the distribution maps into one of 5 classes.

The proposed method introduces several new techniques; for example, symbol types is used to compensate the variation of symbol sizes; each symbol has a type depending on its size and position. These types are used to discriminate the distribution maps and therefor each symbol type is classified in a different distribution map. Beside, document-dependent processing is introduced to improve the performance of the discrimination task. Furthermore, very large databases are used in the discrimination task, these databases are suitable for the evaluation of the usefulness of the proposed features.

Our initial work on this task was reported in [14]. In this work only the spatial relationships among characters were focused. Thus, the subscript pair of "$\prod$" and "p" and the horizontal pair of "D" and "(" in Fig 2 were out of its scope. The present work is a large extension of the previous work. In this work, we consider both symbols and characters.

The remainder of this paper is organized as follows. Section 2 introduces the features which were used in the distribution map. Section 3 presents a document dependent processing. Section 4 shows experimental results with very large databases. Finally, Section 5 presents a conclusion.

2 Feature extraction for discriminating the spatial relationships

2.1 Symbols types

A type for each symbol is defined to compensate the variation in positions and heights of symbols. Figure 3 shows an example of determining the type of symbol "∈". To specify the type of symbol "∈"; first the top and base of its parent character is calculated using the hight ratio of three regions called X:Y:Z regions. Figure 4 shows these regions. Then the type is decided according to the value of X-part and Z-part.

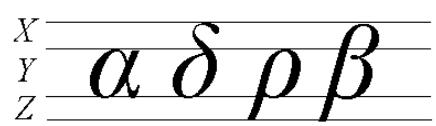


Figure 4. X, Y and Z regions.

Table 1. Symbol types.

Types	Examples of symbols				
X+Y+Z	$\leq$	$\int$	$]$	$\sum$	$\oplus$
X+Y	$<$	$>$	$\wedge$	$\in$	$\subset$
Y	$-$	$=$	$\sim$	$\leftarrow$	$\smile$
Y+Z	$\Subset$	$\perp$	$\supseteqq$		
X	$\hat{a}$	$\check{a}$	$\breve{a}$	$\tilde{a}$	$\dot{a}$
Z	$\underline{a}$	$\underbrace{a}$			

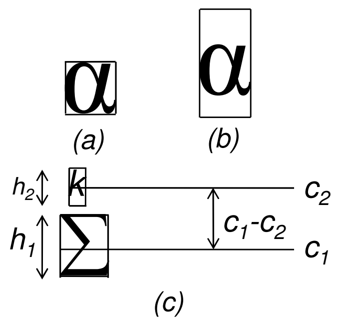


Figure 5. (a) Actual bounding box for character "α". (b) Normalized bounding box for character "α". (c) Normalized sizes (h_1, h_2), normalized centers (c_1, c_2).

Symbols will have 6 types according to X:Y:Z regions such as ("X", "$X + Y$", "$X + Y + Z$", "Y", "$Y + Z$", "Z"). Table 1 shows some examples of symbol types. This estimation of symbol types improves the performance of the proposed method. These types are used to discriminate the distribution maps and therefore each type will be classified in different distribution maps.

2.2 Feature Extraction

As stated in [14], to estimate the proposed features, the normalized bounding boxes are used for characters instead of actual bounding boxes. Figure 5 (a) shows the actual bounding box for character "α" and Fig 5 (b) shows the normalized bounding box for character "α". The normalized bounding boxes are estimated by adding a virtual ascender or a virtual descender or both depending on the character category.

Unfortunately, this technique does not valid for symbols; symbols have more variation than characters. For example, some symbols such as "$-, =$" have smaller hight than "Y" hight and others such as "$[, |$" have longer heights than "$X + Y + Z$" heights. Therefore, the actual bounding boxes are used for symbols.

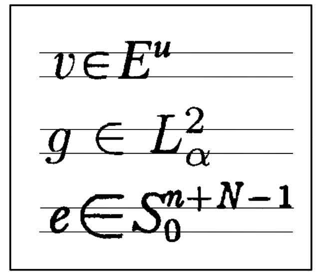


Figure 6. Symbol "$\in$" in different documents.

Let h_1 and h_2 denote the heights of bounding box of the parent and child, respectively. Similarly, let c_1 and c_2 denote the centers of the bounding box of those pairs. Figure 5 (c) shows these parameters.

The relative size H and the relative position D can be extracted for each adjacent pair as follows:

$$H = \frac{h_2}{h_1}, \quad D = \frac{c_1 - c_2}{h_1}. \tag{1}$$

3 Document dependent processing

Each document has its own characteristics which differers from the other document. Observing these characteristics improved the performance of the proposed method.

3.1 Private X:Y:Z

We estimate X:Y:Z ratio for baseline characters and X:Y:Z ratio for non-baseline characters because they have different font shapes[1]. These ratios are common for all characters in each document and therefore we hereafter call them *private* X:Y:Z *ratios*. In contrast, we also can estimate the X:Y:Z ratios common for any document by using all the characters of a large-scale multi-document database and therefore we hereafter call the ratios *common* X:Y:Z *ratios*. Private X:Y:Z ratio outperforms common X:Y:Z ratio as will be shown in the experimental results.

3.2 Irregular symbols and characters

After determining the symbol types, we noticed that, some symbols have different types in different document. These symbols will called *irregular symbols*. For example,

[1] Readers may be confused by the fact that we need to discriminate between baseline characters and non-baseline characters for estimating their own X:Y:Z ratio during the process toward our final goal. For this discrimination we used X:Y:Z which calculated from all characters contained in the database. Of course, the result from this discrimination includes some errors. These errors do not affect the estimation seriously because we use the average of the heights.

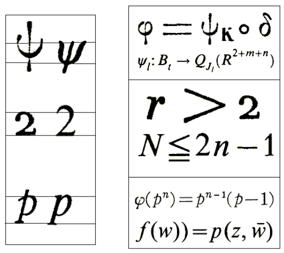


Figure 7. Examples of irregular characters.

"∈" symbol occupies only "*Y*" region in some documents and occupies "*X* + *Y*" regions in another documents , yet occupies the entire "*X* + *Y* + *Z*" regions in the other documents and therefor they have 3 different types. Figure 6 shows these types.

As stated in [14], there are some characters which have different sizes and occupy different "*X*","*Y*", "*Z*" regions in different documents. These characters are called *irregular characters*. Figure 7 shows some examples of these characters.

Special treatment is applied for irregular characters/symbols, in which each irregular characters/symbols was discriminated into many cases depending on its heights and positions. This special treatment improved the performance as will be shown in the experimental results.

4 Experimental results

4.1 Database

The discrimination task was applied on 158,308 adjacent pair of symbols and characters of mathematical expressions. For example, 37,263 adjacent pairs have characters-characters type, 52,057 adjacent pairs have symbols-characters type, 54,924 adjacent pairs have characters-symbols type, and 14,064 adjacent pairs have symbols-symbols type.

To the authors' best knowledge, these databases are the largest of those used in past attempts on the discrimination task. For example, they are larger than the database used in [18], which consists of 297 pages. Such large databases are extremely well suited to derive universal properties (e.g., the discrimination task) of mathematical expressions.

4.2 Distribution maps analysis

Figure 8 (a) shows the distribution map without using symbol types. Heavy overlaps between the classes can be observed on this map. These overlaps come from the variation of the sizes and positions of the actual bounding boxes of symbols. For example, horizontal symbols which occupy "*X* + *Y*" regions may overlapped with superscript symbols because their centers will be up.

Figure 8 (b) shows the distribution map after using symbols types. The overlaps were drastically decreased because we avoided the variation of the sizes of the symbols; symbols with the region ("*X*", "*X*+*Y*", "*X*+*Y*+*Z*", "*Y*", "*Y*+*Z*", "*Z*") were classified in different distribution maps. Thus we can conclude that, using the types of symbols is very powerful for the discrimination task with (H, D)-features.

Table 2. Discrimination accuracy (%) by quadratic classifier on H-D space.

No. of symbol type	Doc. dependent processing		Accuracy rate
	irregular treatment	private *X*:*Y*:*Z* ratio	
1	X	X	93.69
1	X	O	93.69
1	O	X	93.86
1	O	O	94.04
6	X	X	99.14
6	X	O	99.15
6	O	X	99.51
6	O	O	99.57

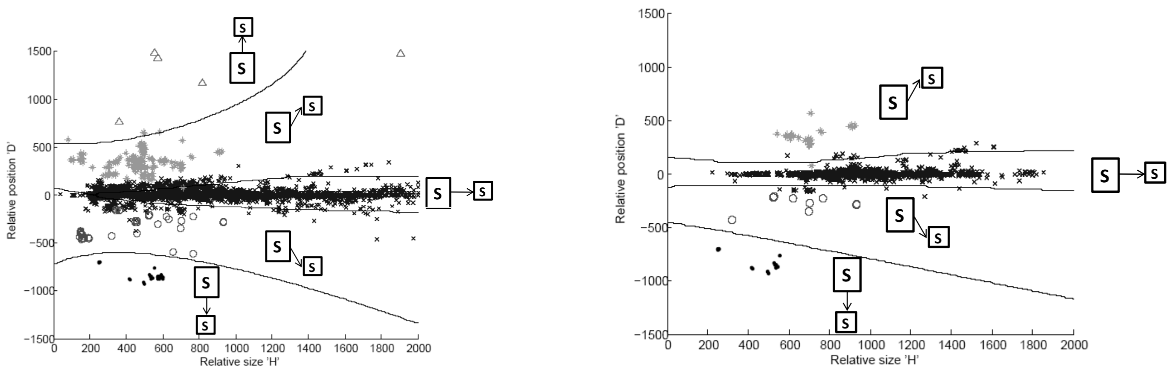
Table 2 shows the accuracy rate using a simple Bayesian classifier. From this table, we notice that, the results improved very much after using symbol types. The best accuracy rate occurred when applying document-dependent processing, in which private $X\!:\!Y\!:\!Z$ ratio and special treatment for irregular characters/symbols were applied. These results prove the importance of using both symbol types and document-dependent processing in the discrimination task.

5 Conclusion

In this paper, the spatial relationships between each adjacent pair of mathematical expressions is classified into one of five classes (horizontal class, subscript class, superscript class, upper class, and lower class) for realizing an accurate structure analysis module of math OCR.

In this task very large databases are used which are suitable to covey the geometric information of characters and symbols of mathematical expressions.

Experimental results shows that, symbol types and document dependent processing improved the performance of the proposed method by observing the distribution maps which defined by two features *relative size* and *relative position*. These two points were overlooked in the



(a)Without using symbol types for all symbol-symbol pairs.

(b) Using symbol types for symbol-symbol pairs, symbols have the "$X + Y + Z$" type.

Figure 8. Distribution maps for different cases. The curves show the decision boundaries by quadratic classifier. The values of H and D are multiplied by 1000.

past attempts, while they give us an important aspect that documents-dependent processing are necessary on the structure analysis.

References

[1] K. Chan, and D. Yeung, "Mathematical expression recognition: A survey," Int. J. Document Analysis and Recognition, vol. 3, no. 1, pp. 3–15, 2000.

[2] J. Ha, R. M. Haralick, and I. T. Phillips, "Understanding mathematical expressions from document images," Proc. 3rd Int. Conf. Document Analysis and Recognition, vol. 2, pp. 956–959, 1995.

[3] J. -Y. Toumit, S. Garcia-Salicetti, and H. Emptoz, "A hierarchical and recursive model of mathematical expressions for automatic reading of mathematical documents," Proc. 5th Int. Conf. Document Analysis and Recognition, pp. 119-122, 1999.

[4] Y. Guo, L. Huang, C. Liu , and X. Jiang, "An automatic mathematical expression understanding system," Proc. 9th Int. Conf. Document Analysis and Recognition, vol. 2, pp. 719–723, 2007.

[5] H. J. Lee, M.c. Lee, "Understanding mathematical expressions using procedure-oriented transformation, " Int. Journal of Pattern Recognition, vol. 27, no. 3, pp. 447–457, 1994. pp. 1084–1087, 1995.

[6] D. Blostein, and A. Grbavec, "Recognition of mathematical notation," In Handbook of Character Recognition and Document Image Analysis, pp. 557–582, 1997.

[7] R.H. Anderson, "Syntax-directed recognition of hand-printed two-dimensional mathematics," in *Interactive Systems for Experimental Applied Mathematics*, M. Klerer and J. Reinfelds, Eds. Academic Press, pp. 436-459, 1968.

[8] M. Okamoto, and B. Miao, "Recognition of mathematical expressions by using the layout structure of symbols," Proc. 1st Int. Conf. Document Analysis and Recognition, pp. 242–250, 1991.

[9] H. Twaakyondo, and M. Okamoto, "Structure analysis and recognition of mathematical expressions," Proc. 3th Int. Conf. Document Analysis and Recognition, pp. 430–437, 1995.

[10] R. Zanibbi, D. Blostein, and J.R. Cordy, "Recognizing mathematical expressions using tree transformation, ", vol. 24, no. 11, Int. J. Pattern analysis and machine intelligence, pp. 1455–1467, 2002.

[11] M. Suzuki, F. Tamari, R. Fukuda, S. Uchida, and T. Kanahori, "INFTY- An integrated OCR system for mathematical documents, " Proc. Int. Conf. ACM Symposium on Document Engineering, pp. 95–104, 2003.

[12] U. Garain, and B. B. Chaudhuri, "A syntactic approach for processing mathematical expressions in printed documents," Proc. Int. Conf. Pattern Recognition, vol. 4, pp. 523–526, 2000.

[13] U. Garain, and B. B. Chaudhuri, "An approach for recognition and interpretation of mathematical expressions in printed documents," Proc. Int. Conf. Spring-Verlag London, vol. 3, pp. 120–131, 2000.

[14] A. Walaa , S. Uchida, and M. Suzuki, "A Large-Scale Analysis of Mathematical Expressions for an Accurate Understanding of Their Structure," Proc. 8th Int. Document Analysis Systems, pp. 549–565, 2008.

[15] M. Suzuki, S. Uchida, and A. Nomura, "A Ground-truthed mathematical character and symbol image database," Proc. 8th Int. Conf. Document Analysis and Recognition, pp. 675–679, 2005.

[16] S. Uchida, A. Nomura, and M. Suzuki, "Quantitative analysis of mathematical documents," Int. J. Document Analysis and Recognition, vol. 7, no. 4, pp. 211–218, 2005.

[17] M. Suzuki, C. Malon, and S. Uchida, "Databases of mathematical documents," Research Reports on Information Science and Electrical Engineering of Kyushu University, vol. 12, no. 1, pp. 7–14, 2007.

[18] U. Garain and B. B. Chaudhuri, "A corpus for OCR research on mathematical expressions," Int. Journal Document Analysis and Recognition, vol. 7, no. 4, pp. 241–259, 2005.