Using Fuzzy Logic to Analyze Superscript and Subscript Relations in Handwritten Mathematical Expressions

Ling Zhang
School of Computing
Queen's University
Kingston, Ontario, Canada
zhangl@cs.queensu.ca

Dorothea Blostein
School of Computing
Queen's University
Kingston, Ontario, Canada
blostein@cs.queensu.ca

Richard Zanibbi
CENPARMI
Concordia University
Montreal, Quebec, Canada
zanibbi@cenparmi.concordia.ca

Abstract

Handwritten mathematical notation contains ambiguities of various kinds. Here we focus on ambiguity in spatial relationships; in particular, we use fuzzy logic to treat ambiguity in subscript-or-inline and inline-or-superscript spatial relationships. We extend an existing system for recognizing handwritten mathematical notation, adding the capability of producing a ranked list of interpretations rather than a single top-choice interpretation. Fuzzy membership values are assigned to each spatial relationship; a given pair of symbols can have non-zero membership in fuzzy sets subscript *and* inline*, or in fuzzy sets* inline *and* superscript*. These fuzzy membership values are combined to produce an overall confidence value for the entire interpretation. We have modified the user interface of our system so that a user can quickly view and select from the ranked interpretations when the highest confidence interpretation is incorrect.*

1. Introduction

Spatial relationships in handwritten math expressions are highly variable [10], and often lead to ambiguity in the meaning of the expression (Figure 1). Fully automatic resolution of all ambiguities is unachievable: since people do not always agree on one "correct" interpretation of an expression, it is impossible for a computer to find a single correct interpretation in all cases. We use fuzzy logic to model symbol layout ambiguities. Using this model, our system presents several interpretations to the user (Figure 2). Each interpretation has a confidence value. Section 3 discusses the use of fuzzy logic to compute confidence values.

Many researchers have investigated the recognition of mathematical notation. Surveys are provided in [2] [4]. Here we review selected papers.

Wang and Faure extensively investigate subscript, inline, and superscript relationships in handwritten input, statistically characterizing the ambiguity that is present [16]. In their tests, symbols are characterized only by bounding boxes; the symbol class (such as ascender, descender, normal) is not known.

Figure 1. Spatial relationships in handwritten math expressions are often inexact or ambiguous. (a) The alignment of x+y does not reflect an intended baseline. (b) This expression has baseline ambiguity. (c) Ambiguity in dominance of division bars.

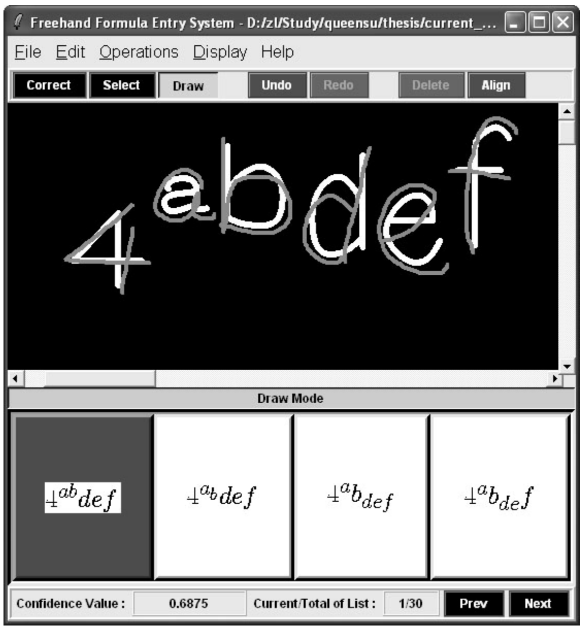


Figure 2. An ambiguous expression is shown at the top. Four interpretations are shown at center. These are the four interpretations with the highest confidence values.

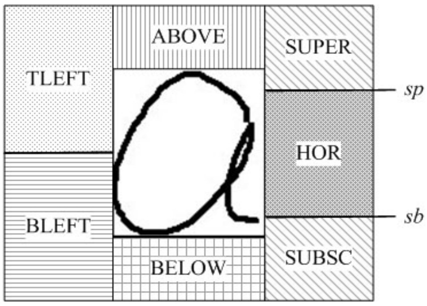


Figure 3. Regions used for baseline detection in DRACULAE [1]. Region definition depends on symbol class; for example, the subscript region is larger for a descender symbol such as *y*, than it is for this normal symbol *a*.

Chou uses a stochastic context-free grammar to create a math recognition system that handles noise and low-resolution images [3]; the math notation is assumed to be typeset using strict, known layout rules. Graph grammars have been used to interpret handwritten mathematical expressions that contain poorly-aligned symbols [8]; this approach involves extensive search and backtracking. Tapia and Rojas use support vector machines and baseline structure analysis to recognize on-line handwritten mathematical expressions [15].

Winkler et al. recognize the structure of handwritten mathematical expressions using a directed graph with soft decision making to generate multiple interpretations in ambiguous cases [17]. Special symbols (fraction, summation, product, integral, root) are located, and used to group surrounding symbols. This grouping involves soft decisions, with calculation of probabilities for ambiguous spatial relations, and generation of alternative interpretations by duplicating the currently processed graph. The work we report here uses fuzzy logic to address ambiguities in a different computational framework.

Eto and Suzuki handle math expressions with OCR ambiguities by constructing a network with one node for each symbol, where the node can represent several possible OCR results for that symbol [5]. A labeled edge represents a choice of two particular OCR interpretations and a choice of a particular spatial relation. The overall interpretation is found by constructing a low-cost spanning tree.

We extend an existing deterministic mathematical notation recognition system, which can be downloaded under GNU public license from [6]. This system consists of three main components: the user interface and OCR engine are from the Freehand Formula Entry System (FFES), as described in [14]. The third component is a parser called DRACULAE [1]. The original DRACULAE returns one interpretation for each input expression. Our fuzzy logic extension to DRACULAE produces a list of alternative interpretations, each with a confidence value.

DRACULAE operates without backtracking, and is able to interpret handwritten expressions containing poorly aligned symbols. Layout analysis in DRACULAE occurs by repeated extraction of baselines. For the expression in Figure 1(a), the main baseline contains three symbols: — + y. The first step in baseline extraction locates the start symbol (the leftmost symbol in the baseline, here —); this is done using a scan of the sorted symbol list, and dominance analysis. Once the start symbol has been located, subsequent baseline symbols are located by testing their membership in regions defined around existing baseline symbols (Figure 3). Once a complete baseline has been extracted, recursion is used to process secondary baselines. In the example of Figure 1(a) the two secondary baselines consist of single symbols: *x* above the —, and *2* below the —. A detailed description of these algorithms is available in [1].

The math symbols are recursively organized into baselines, producing a baseline structure tree (BST) that represents the internal structure of the interpretation. The baseline structure tree can be directly converted to LaTeX; with additional processing and additional analysis of baseline structure, an operator tree can be produced as well.

Fuzzy logic has a long history and many applications [7] [9] [11] [12] [13] [18]. Our use of fuzzy logic is discussed below.

2. Alternative Interpretations

Symbols and spatial relationships are basic elements in our DRACULAE system for interpreting math expressions. Ambiguity in the spatial relationships between symbols forms the basis for our generation of alternative interpretations. Thus we begin by summarizing how symbols are treated. Every symbol has bounding-box attributes (*minx*, *miny*, *maxx*, *maxy*), and a symbol class (Ascender, Descender, Normal, Open Bracket, Non-scripted, Root, or Variable Range). The *centroid* for a symbol is calculated based on the bounding box and the symbol class. A descender symbol's centroid lies high in the bounding box, whereas the centroid for an ascender symbol lies lower in the bounding box.

Spatial relationships are associated with a symbol. The spatial relationship between this symbol and another symbol is determined by finding which region the other symbol's centroid is located in. Our original DRACULAE uses crisp regions (Figure 3), so any two symbols have a uniquely-determined spatial relationship. We extend this by adding fuzzy membership in the spatial relationships, as shown in

Figure 4. Symbols whose centroids land in the SUPER/HOR region have non-zero memberships in the two spatial relations SUPER and HOR. Similarly, symbols with centroid locations in the SUBSC/HOR region result in non-zero memberships in the two spatial relations SUBSC and HOR. These regions are parameterized by the thresholds *sp* (superscript threshold), *ub* (upper base threshold), *lb* (lower base threshold), and *sb* (subscript threshold). These thresholds define the membership functions of the fuzzy sets *super*, *inline*, and *subsc*, as illustrated in Figure 4.

Mathematically, if the symbol's centroid *x* coordinate lies between *ub* and *sp*, then the symbol's membership

- in *super* is $(x-ub)/(sp-ub)$
- in *inline* is $(sp-x)/(sp-ub)$

Similarly, if the symbol's centroid *x* coordinate lies between *lb* and *sb*, then the symbol's membership

- in *subsc* is $(x-sb)/(lb-sb)$
- in *inline* is $(lb-x)/(lb-sb)$

We use the term *confidence value* for the membership value of a symbol in the fuzzy sets *inline*, *super*, and *subsc*. If a symbol's centroid lies in the overlapping ranges, the two confidence values of the symbol in different fuzzy sets (*super* and *inline*, or *inline* and *subsc*) sum to 1 according to the definitions of membership functions. As a result, membership functions in the overlapping ranges sum to 1.

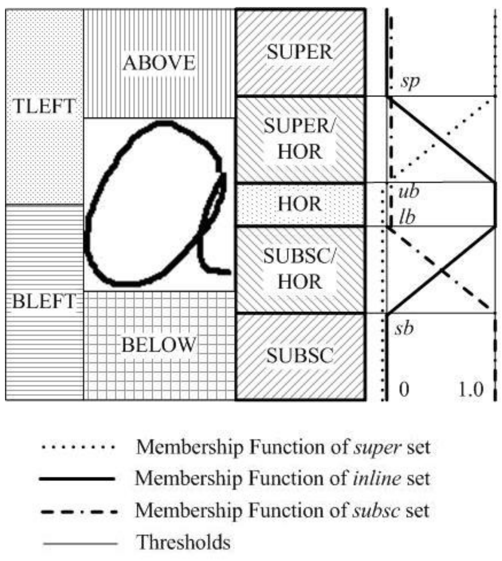


Figure 4. Definition of fuzzy regions relative to a symbol. Nonzero memberships in two fuzzy sets are possible in the regions SUPER/HOR and SUBSC/HOR. The right portion of the figure shows the membership functions for the three fuzzy sets, *super, inline, subsc.*

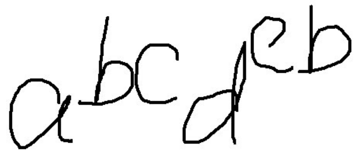


Figure 5. An example to illustrate the operation of Extended DRACULAE.

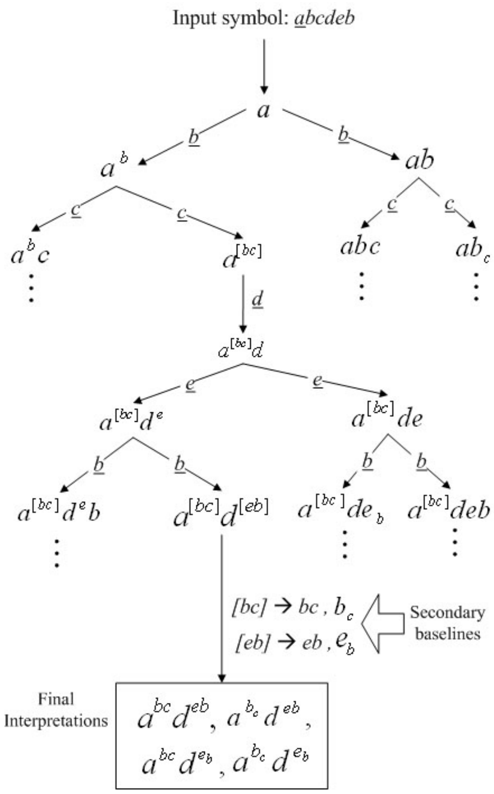
Mathematical expressions are analyzed using multiple passes over the input symbols. The four major passes perform layout analysis, lexical analysis for tokens, lexical analysis for relations, analysis of expression syntax, and analysis of expression semantics [1]. Our focus is on the layout pass; this is where most of the changes were needed, to extend DRACULAE into a system that computes alternative interpretations of the input symbols.

The operation of the extended layout pass is illustrated in Figure 6, using the input shown in Figure 5. The layout pass recursively finds baselines. The first step of the layout pass is to find the start symbol of the baseline that dominates the current set of symbols. In this example, the start symbol is *a*. The next symbol of the baseline is *inline* with the start symbol. Here, many possible baselines are produced, by considering the various combinations of membership values in the fuzzy sets *inline*, *subsc*, and *super*. For the example in Figure 5, 11 different dominant baselines are produced. The lower portion of Figure 6a illustrates the dominant baseline *ad*; the notation $a^{[bc]}d^{[eb]}$ means that secondary baselines will be constructed from symbol sets *bc* and *eb*. After construction of the secondary baselines, which have two alternatives each, four final interpretations are produced using the dominant baseline *ad*. In total, the 11 dominant baselines expand into 18 final interpretations.

The number of interpretations is affected by the size of the fuzzy regions SUPER/HOR and SUBSC/HOR. We use two threshold ratios, base threshold ratio (*b*) and threshold ratio (*t*), as well as the y center (*center*) of a symbol, to control the size of these regions:

- $h = maxy - miny$
- $lb = center - b * h$
- $ub = center + b * h$
- $sp = 2 * (maxy - t * h) - ub = ub + (1 - 2t - 2b)h$
- $sb = 2 * (miny + t * h) - lb = lb - (1 - 2t - 2b)h$

The condition $t + b < 0.5$ must be applied to make the definitions of *sp* and *sb* work. The example in Figure 6 uses a value of 1/8 for the base threshold ratio and a value of 1/6 for the threshold ratio, producing the 18 final interpretations described above. If we change both threshold ratios to 1/6, this increases the size of HOR, and reduces the size of the SUPER/HOR and SUBS/HOR regions in Figure 4. As a result, the number of final interpretations drops to 10.



(a) Extracting baselines

Interpretations	confidence value
$a^{bc}d^{eb}$	0.553571
$a^{bc}d^{e}b$	0.446429
abc_{d}^{eb}	0.225
$abc_{d}^{e}b$	0.225
$a^{bc}d^{e_b}$	0.2
$abc_{d}^{e_b}$	0.2
$a^{bc}de_b$	0.196429
$a^{bc}deb$	0.196429
$abc_{d}e_b$	0.125
$abc_{d}eb$	0.125
$a^{b_c}d^{e_b}$	0.0625
$a^{b_c}d^{eb}$	0.0625
$a^{b_c}d^{e}b$	0.0625
$a^{b_c}de_b$	0.0625
$a^{b_c}deb$	0.0625
$ab_{c_d}^{e}b$	0.0625
$ab_{c_d}e_b$	0.0625
$ab_{c_d}eb$	0.0625

(b) Final interpretations with fuzzy confidences

Figure 6. (a) Alternative interpretations produced when the Layout Pass processes the input in Figure 5. The top part of the figure illustrates the extraction of the dominant baseline. In this case, 11 different dominant baselines are extracted. The dominant baseline *ad* is illustrated in detail: the two secondary baselines (using symbol sets *bc* and *eb*) have two interpretations each, resulting in four final interpretations built on the dominant baseline *ad*. (b) Final interpretations with fuzzy confidences with base threshold ratio = 1/8, threshold ratio = 1/6.

3. Computing Confidence Values

Confidence values for individual spatial relationships are given by the fuzzy membership functions illustrated in Figure 4. These individual confidence values need to be combined to compute a confidence value for an entire interpretation. As is common in fuzzy logic systems, we use combination by minimum, computing the confidence of an entire interpretation as the minimum of the confidence values of the spatial relationships used in that interpretation. This combination rule can result in many ties in confidence values: if many final interpretations share a lowest-confidence spatial relationship, then differences in other parts of the interpretations are not reflected in the confidence values assigned to the overall interpretations.

We experimented with another method of combining confidence values: combination by multiplication. Here, the confidence of a final interpretation is computed as the product of the confidence values of the spatial relationships used in the interpretation. This combination rule produces fewer ties in confidence values. Unfortunately, it penalizes interpretations that have larger numbers of component spatial relationships. In future work, it would be interesting to compare the rank-order of confidence values produced algorithmically with the rank-order of interpretations produced by human subjects.

4. Conclusion

We have applied fuzzy logic to characterize the ambiguity in subscript and superscript relations in handwritten mathematical expressions. Two threshold ratios are used to define the membership functions in fuzzy sets *super*, *inline* and *subsc*. If the fuzzy regions are defined to be large spatially, then a large set of alternative interpretations is produced, and conversely, small fuzzy regions produce a small list of alternative interpretations. In future work, an adaptive method for adjusting the size of fuzzy regions may be found.

References

[1] R. Zanibbi, J. Cordy, D. Blostein, "Recognizing Mathematical Expressions Using Tree Transformation", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 24, No. 11, pp. 1455-1467, Nov. 2002.

[2] D. Blostein, A. Grbavec, "Recognition of Mathematical Notation", *Handbook of Character Recognition and Document Image Analysis*, pp. 557-582, World Scientific, 1997.

[3] P. Chou, "Recognition of Equations Using a Two-Dimensional Stochastic Context-Free Grammar", *Visual Communications and Image Processing IV*, SPIE Proceedings Vol. 1199, pp. 852-863, 1989.

[4] K. Chan, D. Yeung, "Mathematical Expression Recognition: A Survey", *Int'l J. Document Analysis and Recognition*, Vol. 3, No. 1, pp. 3-15, Aug. 2000.

[5] Y. Eto, M. Suzuki, "Mathematical Formula Recognition using Virtual Link Network," *Proc. Sixth Int'l Conf. Document Analysis and Recognition (ICDAR 2001)*, Seattle, Washington, Sept. 2001, pp. 762-767.

[6] http://www.cs.queensu.ca/drl/ffes/.

[7] *Proc. IEEE International Conferences on Fuzzy Systems*, annually since 1992.

[8] A. Grbavec, D. Blostein, "Mathematics Recognition Using Graph Rewriting", *Third Intl. Conf. Document Analysis and Recognition*, pp. 417-421, Montreal, Aug. 1995.

[9] G. Klir, B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hill, Upper Saddle River, NJ, 1995.

[10] E. Miller and P. Viola, "Ambiguity and Constraint in Mathematical Expression Recognition," *Proc. 15th National Conf. on Artificial Intelligence (AAAI 98)*, Madison, Wisconsin, July 1998, pp. 784-791.

[11] T. Ross, J. Booker, W. Parkinson, *Fuzzy Logic and Probability Applications: Bridging the Gap*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002.

[12] T. Ross, *Fuzzy Logic with Engineering Applications*, Second Edition, John Wiley & Sons, Ltd, 2004.

[13] N. Singpurwalla, J. Booker, "Membership Functions and Probability Measures of Fuzzy Sets", *Journal of the American Statistical Association*, Vol. 99, No. 467, pp. 867-876, Sept. 2004.

[14] S. Smithies, K. Novins, J. Arvo, "Equation Entry and Editing via Handwriting and Gesture Recognition", *Behaviour and Information Technology*, Vol. 20, No. 1, pp. 53-67, 2001.

[15] E. Tapia, R. Rojas, "Recognition of On-line Handwritten Mathematical Formulas in the E-Chalk System," *Proc. Seventh Int'l Conf. on Document Analysis and Recognition, ICDAR 2003.* Edinburgh, Scotland, Aug. 2003, pp. 980-984.

[16] Z. Wang, C. Faure, "Structural Analysis of Handwritten Mathematical Expressions", *Proc. Ninth Int'l Conf. Pattern Recognition*, pp. 32-34, 1988.

[17] H.-J. Winkler, H. Fahrner, M. Lang, "A Soft Decision Approach for Structural Analysis of Handwritten Mathematical Expressions," *Proc. IEEE Intl. Conf. Acoustics, Speech, and Signal Processing*, Detroit, pp. 2459-2462, 1995.

[18] L. Zadeh, "Fuzzy sets", *Inf. Control*, Vol. 8, pp. 338-353, 1965.