IDENTIFYING LAYOUT CLASSES FOR MATHEMATICAL SYMBOLS USING LAYOUT CONTEXT

Ling Ouyang and Richard Zanibbi

Document and Pattern Recognition Lab
Rochester Institute of Technology, Rochester, NY USA
rogerouyang@gmail.com, rlaz@cs.rit.edu

ABSTRACT

We describe a symbol classification technique for identifying the expected locations of neighboring symbols in mathematical expressions. We use the seven symbol layout classes of the DRACULAE math notation parser (Zanibbi, Blostein, and Cordy, 2002) to represent expected locations for neighboring symbols: Ascender, Descender, Centered, Open Bracket, Non-Script, Variable Range (e.g. integrals) and Square Root. A new feature based on shape contexts (Belongie et al., 2002) named *layout context* is used to describe the arrangement of neighboring symbol bounding boxes relative to a reference symbol, and the nearest neighbor rule is used for classification. 1917 mathematical symbols from the University of Washington III document database are used in our experiments. Using a leave-one-out estimate, our best classification rate reaches nearly 80%. In our experiments, we find that the size of the symbol neighborhood, and number and arrangement of key points representing a symbol affect performance significantly.

***Index Terms*—** shape contexts, document layout analysis, character recognition, math recognition

1. INTRODUCTION

Recognizing mathematical notation is a challenging pattern recognition problem due to the large number of math symbols, and complexities in interpreting the two dimensional arrangement of symbols and their intended semantics [1]. Much of the information in mathematical expressions is carried by the relative spatial position between symbols, such as superscript, subscript, adjacent and containment (e.g. in a square root). Many methods have been proposed to extract spatial relationships between symbols such as coordinate grammars [2], Projection Profile Cutting [3], minimum spanning trees for penalty graphs representing alternative symbol layouts [4], recursive baseline structure analysis [5], and others.

We present an algorithm for classifying all mathematical symbols into seven layout classes (see Figure 1 and Table 1), which identify the expected locations of neighboring symbols with a significant spatial relationship [6]. From these seven classes, existing techniques (in particular, baseline structure analysis [5]) may be used to identify the spatial arrangement of symbols in an expression. Surrounding regions of a symbol may include below, above, superscript, subscript, horizontal adjacency and containment. Different classes have different associated regions, as shown in Fig 1.

Successfully identifying the layout class of symbols may permit symbol layout to be recognized without recognizing the specific identity of characters (i.e. OCR-free structure recognition), and may provide features for *subsequent* OCR.

We use a feature named *layout context* to describe the arrangement of neighboring symbols relative to a reference symbol. A number of key points sampled from the side and/or interior of the symbol bounding box are used to represent symbol locations. A circle placed at the reference bounding box center with adjustable radius is used to define the neighboring symbol region. We examine a variety of key point models and neighborhood sizes, and use the nearest neighbor rule for classification. Depending on the chosen parameters, we obtain a classification accuracy between 43% and almost 80% on symbols taken from math expressions in the University of Washington III document database.

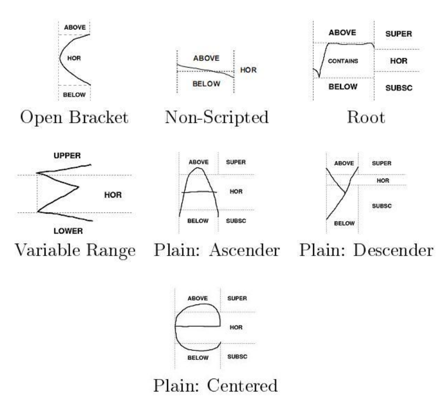


Fig. 1. Symbol layout classes [5]

Table 1. Class Membership [5]

Class	Symbols
Ascender	0...9, $A...Z$
Descender	$g, p, q, y, r, \eta, \rho$ $\Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi$
Open Bracket	[{(
Non Scripted	Unary binary operators and relation ($\times, \backslash, \geq, \div, \equiv$)
Root	$\sqrt{}$
Variable Range	$\sum\prod\int\cap\cup$
Center	All other symbols

2. METHODOLOGY

We make use of a new feature named *layout context*. The inspiration for this feature comes the shape contexts of Belongie et al. [7]. Layout context is designed to depict the spatial distribution of neighboring symbols relative to a reference symbol within a mathematical expression.

We represent the spatial location of a symbol using a key points model. We assume that symbols are previously segmented, and take key points from symbol bounding boxes. We consider taking key points from three locations: from the bounding box boundary, the interior of the bounding box, and both the interior and exterior of the bounding box. For each type of key point model, we use n to denote the total number of points for the instance. Fig 2 shows some example key point models.

For the layout context, we also need to define the local neighborhood area of the symbol. This neighborhood is defined by a circle centered at the reference symbol center. Any key points from symbols that lay in this area are included in the layout context. We use r to denote the ratio between the neighborhood circle radii and the unit length, which is half of the reference symbol bounding box diagonal (from the center to a corner). The larger r is, the larger the neighborhood area is, and in turn the more symbols will be included in the feature calculation.

After choosing a key points model and the radius of the neighborhood area (r), a layout context is computed using the three steps below. Figure 3 provides an example of computing the layout context for the symbol '+' within an expression.

1. Compute the vectors connecting the neighboring symbol key points to the symbol center o, and calculate the length and angle of each vector.

2. Divide the neighborhood region into 60 bins, consisting of 12 equal angle bins and 5 distance bins. The ratio of the five distance bins radii moving out from the center of the symbol are $\frac{1}{16} : \frac{1}{8} : \frac{1}{4} : \frac{1}{2} : 1$, with the whole (1) being the radius of the circle neighborhood (as determined by r).

3. Compute the histogram of key points by their distance and angle relative to the reference symbol center over the 60 bins. Normalize the histogram by dividing each bin by the total number of key points in neighborhood. The resulting histogram is the layout context of the reference symbol.

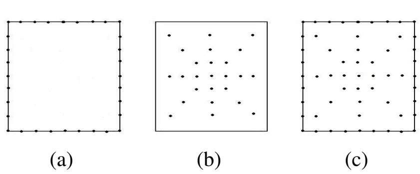


Fig. 2. Key points models. (a) key points sampled from the bounding box sides, (b) key points sampled evenly from the diagonals, horizontal and vertical center lines, and (c) key points from both the bounding box interior and sides

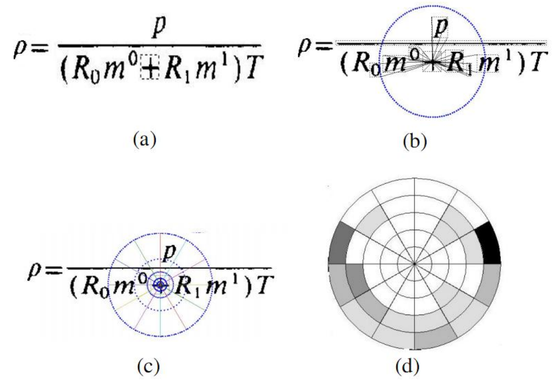


Fig. 3. Example: Calculating layout context for "+," using $r = 4$ and 5 key points (bounding box corners and center). (a) Expression where the reference symbol "+" lies. (b) Vectors that connect other key points within the neighborhood to the bounding box center. (c) 60 log polar bins of the circle neighborhood area. (d) Visual representation of the layout context of symbol "+". Darker bins contain more points.

After feature extraction, classification is performed using the Nearest-Neighbor algorithm (NN). The histogram matching cost between the a reference symbol and training instances is used as as the distance metric for the NN algorithm. As layout context is represented by a histogram, it is natural to use the χ^2 metric [7] as shown in Equation(1) (with Yates' correction). Here p_i and q_i are symbols, and C_{ij} represents the cost of matching the layout context histograms $h(p_i)$ and $h(q_i)$ for these two symbols.

$$C_S = \frac{1}{2} \sum_{i=1}^{K} \frac{[h(p_i) - h(q_i)]^2}{h(p_i) + h(q_i)} \quad (1)$$

We use the leave-one-out (LOO) [8] method to estimate recognition rates. LOO is the extreme case of k-fold cross validation, where all data is used for training and validation. A shortcoming of the approach is that the method may be computationally expensive for large data sets.

3. EXPERIMENT

To evaluate the accuracy of our classification method, we performed an experiment in which we examined different key point models and neighborhood sizes. We used the University of Washington III database for training and testing data (1917 mathematical symbols within 73 expressions). A summary of our results is provided in Table 2.

Table 2. Classification rates for combinations of neighborhood sizes (r-radius) and key point locations (i: inside bounding box, s: bounding box side). n is the number of key points with the highest accuracy for each condition

Condition	r	n	i	s	Accuracy
Control	1	1	•		0.420
1	1	121	•		0.707
2	2	25	•		0.721
3	4	57	•		0.740
4	8	25	•		0.670
5	16	57	•		0.571
6	1	128		•	0.689
7	2	128		•	0.739
8	4	128		•	0.733
9	8	16		•	0.662
10	16	4		•	0.559
11	1	89	•	•	0.735
12	**2**	**89**	•	•	**0.792**
13	4	89	•	•	0.748
14	8	41	•	•	0.662
15	16	249	•	•	0.557

Our control condition is when the radius of the outermost circle is just half of the bounding box diagonal ($r = 1$, encircling the bounding box) and the key point model uses only bounding box centers ($n = 1$). As seen in Table 2, accuracy ranges from 42.0% (control) to 79.2% (condition 12). Performance is best when $r = 2$ and both inner and side points are used with $n = 89$. The worst condition is when the $r = 16$ ($55.7\% \sim 57.1\%$,), which includes key points from most or all symbols in an expression.

Including key points from neighboring symbols very close to the reference symbol, located within the circle whose radius is the twice the unit length for the reference symbol BB ($r = 2$), gives the highest accuracy for all three sources of key points (indicated by i and s). Accuracy decreases when larger or smaller neighborhoods are used.

In addition, the number and type of the key points affect accuracy. In general, using too few key points, or only one location for key points (inner or side) degrades accuracy. This may be because the layout context feature will not accurately reflect the spatial distribution of symbols if the number of key points included in the histogram is small.

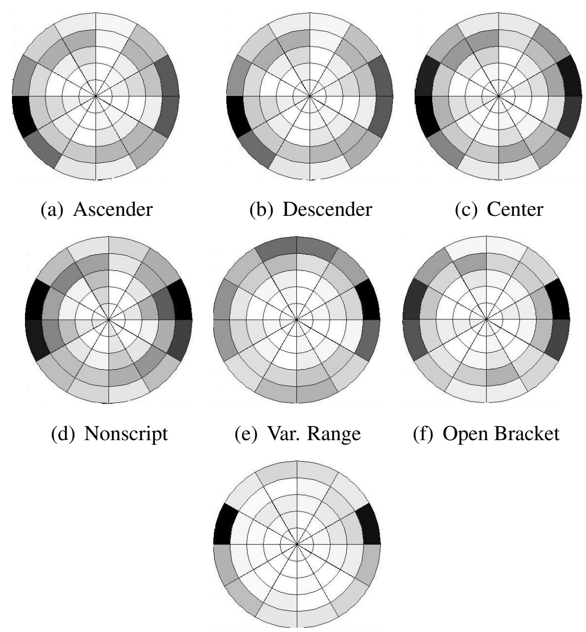
A confusion matrix for the best condition in Table 2 (condition 12) is shown in Table 3. The last row of Table 3 contains the most frequent confusion for each layout class. Center is the layout class that is most frequently confused with other ones. This may be because of the definition of the Center layout class (as shown in Table 1), which includes all symbols that do not belong to the other six classes, including some we may not have anticipated.

The Root class has the highest accuracy, possibly because

Table 3. Confusion matrix for best condition (12) in Table 2

				Correct	Class			
Predict Class	Ascender	Descender	Center	Open Bracket	Non-script	Variable Range	Root	Predicted Frequency
Ascender	**82.8%**	11.8%	12.0%	5.0%	6.3%	15.9%	0.0%	660
Descender	3.2%	**69.2%**	4.2%	2.1%	2.3%	2.2%	0%	144
Center	7.3%	13.3%	**75.6%**	10.0%	9.5%	4.5%	4.5%	501
Open Bracket	0.9%	0.7%	2.8%	**82.0%**	0.7%	2.2%	0	139
Non-script	5.3%	3.9%	5.0%	0	**80.3%**	9.0%	0	413
Variable Range	0.4%	0.7%	0.4%	0.7%	0.7%	**65.9%**	0.0%	39
Root	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	**95.4%**	21
Actual Frequency	657	127	500	139	428	44	22	694
Common Confusion	Center	Center	Ascender	Center	Center	Ascender	Center	

the layout contexts for root symbols are quite distinct from the other classes (see Fig 4). The neighborhood area for square roots is usually larger than for other layout classes for a given r, due to a longer bounding box diagonal. Consequently, there are more symbols covered in the ring zone between the most two distant circles from the bounding box center. For the Root class, differences between the densities of the leftmost-upper and rightmost-upper bins are much larger than that of other layout classes; this is partly an artifact of our data set, where roots are often located in the denominator of a fraction (see Figure 3).



(a) Ascender (b) Descender (c) Center

(d) Nonscript (e) Var. Range (f) Open Bracket

(g) Root

Fig. 4. Average layout context histogram for each layout class

We also conducted an experiment to investigate the contribution of reference symbol and neighboring symbols to the classification performance. A comparison between the highest classification rates when using the reference symbol only, using neighboring symbols only, and using both reference and

$$f_{cl}(\ln m) = \frac{N_{cl}}{\sigma_{cl}\sqrt{2\pi}} \exp\left[-\frac{(\ln m - \mu_{cl})^2}{2\sigma_{cl}^2}\right]$$

$$B = \frac{K_o(\sqrt{\omega p}\, r_e)}{K_0(\sqrt{\omega p}\, r_e)[pI_o(\sqrt{\omega p}) - \sqrt{\omega p}\, I_1(\sqrt{\omega p})] - I_o(\sqrt{\omega p}\, r_e)[pK_0(\sqrt{\omega p}) + \sqrt{\omega p}\, K_1(\sqrt{\omega p})]}$$

$$\overline{h}_s = \frac{-I_o(\sqrt{\omega p}\, r_e) \cdot K_0(\sqrt{\omega p}\, r_D) + K_o(\sqrt{\omega p}\, r_e) I_o(\sqrt{\omega p}\, r_D)}{K_0(\sqrt{\omega p}\, r_e)[pI_o(\sqrt{\omega p}) - \sqrt{\omega p}\, I_1(\sqrt{\omega p})] - I_o(\sqrt{\omega p}\, r_e)[pK_0(\sqrt{\omega p}) + \sqrt{\omega p}\, K_1(\sqrt{\omega p})]}$$

Fig. 5. Expressions in our dataset containing square roots

neighboring symbols (as used to produce Table 2), is shown in Fig 6. In the results, classification performance is usually better when using both reference and neighborhood symbols key points than using either alone.

In the condition where reference symbol key points are used alone, the classification rate is nearly constant (62%) for all the r values. However, in the other two conditions where neighboring symbols are used, the classification rates follow a similar trend, in which accuracy arises when r changes from 1 to 2 and decreases when r continues to increase until $r = 16$. This indicates that both the reference symbol and neighboring symbols contribute to the classification performance but with different effects. This again shows that selecting the size of neighborhood area to cover closely surrounding neighboring symbols improves classification accuracy. On the other hand, including symbols that are too far away from the reference symbol (when $r = 16$) may deteriorate performance.

4. FUTURE WORK

In order to improve classification accuracy, the layout context feature might be enhanced by weighting key points, e.g. by distance from the reference symbol center. We have used one of the simplest classification techniques (nearest neighbor), and the use of a well-designed neural network or support vector machine, and/or boosting classifiers (e.g. via AdaBoost) might also increase accuracy.

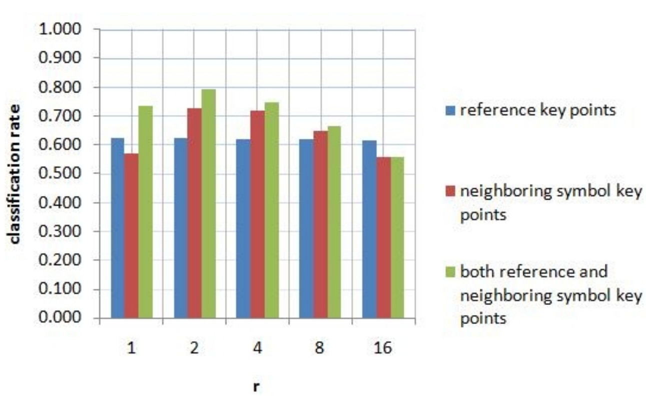


Fig. 6. Comparison of highest classification rates for key points taken from the reference symbol alone, neighborhood symbols alone, and both reference and neighbor symbols for different neighborhood sizes (r)

One might consider using a 'global' layout context, or set of shape contexts sampled from an expression to identify the layout context of mathematical symbols. One could also combine layout contexts with other visual features such as aspect ratio or (regular) shape contexts for the reference symbol. Finally, an obvious extension to our work is to take key points from contours and/or foreground pixels of symbols directly, rather than from bounding box locations.

This work provides a useful baseline for comparison with these other approaches.

5. ACKNOWLEDGEMENT

This work was supported by a Xerox Corporation UAC Grant and a Chester F. Carlson Center for Imaging Science scholarship held by the first author.

6. REFERENCES

[1] Y. Dit-Yan C. Kam-Fai, "Mathematical expression recognition: A survey," *International Journal on Document Analysis and Recognition*, vol. 3, pp. 3–15, 2000.

[2] R. Anderson, *Syntax-Directed Recognition of Hand-Printed Two-Dimensional Mathematics*, Phd Thesis, Harvard University. Cambridge, MA, US, 1968, Cambridge, MA, USA, 1968.

[3] M. Okamoto and B. Miao, "Recognition of mathematical expressions by using the layout structures of symbols," in *Proc. ICDAR '91*, 1991, pp. 242–250.

[4] Y. Eto and M. Suzuki, "Mathematical formula recognition using virtual link network," in *Proc. ICDAR*, 2001, pp. 762–767.

[5] R. Zanibbi, D. Blostein, and J.R. Cordy, "Recognizing mathematical expressions using tree transformation," *IEEE Transactions On Pattern Aanlysis and Machine Intelligence*, pp. 1455–1467, 2002.

[6] L. Ouyang, *Symbol Layout Classification For Mathematical Expressions Using Layout Context*, Master's Thesis, Rochester Insititute of Technology, Rochester, NY, USA, (expected 2009).

[7] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape context," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 24, no. 24, pp. 509–522, 2002.

[8] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," 1995, pp. 1137–1143, Morgan Kaufmann.