The Problem of Handwritten Mathematical Expression Recognition Evaluation

Ahmad-Montaser Awal, Harold Mouchère and Christian Viard-Gaudin
IRCCyN/IVC – UMR CNRS 6597
Ecole polytechnique de l'université de Nantes - France
{ahmad-montaser.awal, harold.mouchere, Christian.Viard-Gaudin }@univ-nantes.fr

Abstract

We discuss in this paper some issues related to the problem of mathematical expression recognition. The very first important issue is to define how to ground truth a dataset of handwritten mathematical expressions, and next we have to face the problem of benchmarking systems. We propose to define some indicators and the way to compute them so as they reflect the actual performances of a given system.

1. Introduction

Due to the importance of mathematics in scientific documents, online/offline handwritten mathematical expression (ME) recognition obtained special attention in the last few years. Thus, a significant number of researches attempts to resolve the problem of ME recognition [15][16]. Regardless the good performance achieved by many systems, this domain lacks a unified method to assess their performances. In consequences, from a global point of view, comparison of results coming from different systems is limited for many reasons. Firstly, there is no available public dataset of online/offline handwritten MEs. Hence, researchers tend to collect their own set of handwritten expressions which might be sometimes limited to a sub-class of expressions or to certain domains. Though direct comparison of different systems performance is not possible. Secondly, each system adopts its own data structure to represent ground truth of MEs. Though, there is no common evaluation measures available.

We propose in this paper to describe advantages and drawbacks of some interesting ME ground truth representations and evaluation methodologies of ME recognition systems.

2. ME Specificities

A mathematical expression is a 2D layout of math symbols. In offline ME recognition problems, a symbol consists of a set of black pixels not necessary connected. While a symbol in online ME consists of one or more strokes; with possibilities of delayed strokes, a stroke being the sequence of points between a pen down and a pen lift. Thus, recognizing an expression consists in finding the best possible grouping of those strokes or pixels to represent symbols. At the same time, spatial relations among symbols must be found in order to find out the structure (layout) of the recognized expression.

Most of recognition systems considers ME recognition as a set of three sub problems [2]: segmentation, recognition and structural analysis and interpretation. The segmentation step consists in grouping strokes or pixels belonging to the same symbol. The symbol recognition step consists in associating a label to each found symbol. Then, structural analysis evaluates the relationships between the symbols and uses a grammar to propose a valid interpretation of the ME.

2. ME ambiguities

There are many sources of ambiguity in ME recognition. The first one is the ambiguity of expressions themselves, and a second one is a natural ambiguity of handwriting production, which can be considered as the output of a stochastic process with a large intra-class variance and at the same time a large overlap between classes.

The language producing MEs is not a completely formal language. The same expression can be

interpreted differently in different contexts. For example, the expression $f(y+1)$ can have two different interpretations: it can be considered as the variable *f* multiplied by the expression (*y*+1); or the function *f* applied to the value *y+1*. Similarly, the expressions $a/2b$ can be interpreted as $\frac{a}{2}b$ or as $\frac{a}{2b}$. Another example is the expression $sin^2 \alpha/2$, which can be interpreted as: $(sin(\alpha/2))^2$ or $(sin\ \alpha)^2/2$. This type of ambiguity could be solved by an exhaustive bracketing which is not comfortable in a handwriting task. Furthermore, handwriting is well known to be naturally ambiguous. Not only, a given pattern can be distorted and thus the assignation to a class is ambiguous, but in addition, different interpretations are possible even without any distortion. For example with the first sample of Figure 1, we can interpret this sample as the absolute value of the variable *x*, or as being 1 times 1. With the second example, it can be interpreted as the sequence of the implicit product of three variables *a*, *x*, and *b*, or the explicit product between *a* and *b*.

Figure 1 Examples of ambiguity of handwritten mathematical symbols

3. ME Representation

An important question is to know what to represent in the ground truth of a ME. Is it the layout of the set of symbols or is it the interpretation of the expression? Usually, the final result of a recognition process is a LaTeX string, or a MathML structure. A LaTeX string is a very popular representation of a ME but it has some limitations. First, it represents the layout of the expression, and does not intend to interpret the mathematical expression. Second, this description is not unique: the same layout can be described with some variants regarding for instance the number of braces. If we consider the expression: $(x+2)^3$, it will be written either as $(x+2)^3$ or ${(x+2)}^{3}$.

On the other side, MathML [1] is an emerging XML format designed to draw ME in portable documents like web pages. Moreover, it aims at encoding either mathematical layout or mathematical meaning (for graphical displays; speech synthesizers; input for computer algebra systems; plain text displays; print media). In the first case, a given description should define a single layout, while in the second case, the ME will be displayed differently by different renders. Figure 2 presents these two possible MathML descriptions of the Expression $(x+2)^3$. We can note that with the second description, the occurrence of the brackets are not explicit, while they are defined explicitly with the first description. One consequence is that different layouts which represent the same ME from an interpretation point of view will be defined by the same description.

```
<msup>
   <mrow>
      <mo>(</mo>
      <mrow>
         <mi>x</mi>
         <mo>+</mo>
         <mn>2</mn>
      </mrow>
      <mo>)</mo>
   </mrow>
   <mn>3</mn>
</msup>
```

(a) Presentation markup

```
<apply>
  <power/>
  <apply>
      <plus/>
      <ci>x</ci>
      <cn>2</cn>
  </apply>
  <cn>3</cn>
</apply>
```

(b) Content markup

Figure 2 MathML markup of $(x+2)^3$

For example the three expressions displayed in Figure 3 will have the same MathML description defined by:

```
<apply>
    <minus/>
    <apply>
      <divide/>
      <ci>a</ci>
      <ci>b</ci>
    </apply>
</apply>
```

$-\frac{a}{b}$ $-(a/b)$ $-\left(\frac{a}{b}\right)$

Figure 3 The same MathML ME displayed by different renders (a) Test Suite of MathML [1] (b) FireFox 3.5.7 and (c) MathMagic 4.81 [2]

Thus, if we use this description to print an expression, as no layout information is provided then we will obtain one of the three layouts presented in Figure 3 according to the used render.

Based on these observations, it becomes clear that MEs must not be groundtruthed only by their content but foremost with their displayed symbols and their layout. However, in advanced steps of ME recognition

[1] http://www.w3.org/Math/testsuite/mml2-testsuite/index.html
[2] http://www.mathmagic.com/

it is possible to apply some conversions in order to obtain the content if needed with the difficulties related to ME ambiguities presented in the previous section.

On the other hand, the recognition systems have their own ME representation depending on how they perform their recognition. Many systems generate trees to represent expressions as a result of structural and syntactic analysis. Hence, these trees hold more information about the structure of the expression. Intuitively, trees are very useful to evaluate recognition systems because they contain not only spatial and logical relations between symbols, but also the symbols recognition and segmentation information.

In [4], the authors simplify the use of trees as structure presentation. They introduce hidden writing area (HWA) associated to each input stroke that defines the relation with previous one. A more common way to represent the structure of an expression is to use relational trees. Binary trees were used in [10]; where non-terminals are the possible logical relations between symbols, and terminals are the recognized symbols. More recently, [3] re-adopt the use of symbol relation trees (SRT), as in Figure 4. A SRT is formed with a dominant symbol and then its sub-expressions as child nodes. The spatial relationship between a dominant symbol and its children is coded using the edges. Then each sub-expression is represented recursively by a SRT using a new dominant symbol. Spatial relations are chosen among six possible types: inside, over, under, superscript, subscript and right.

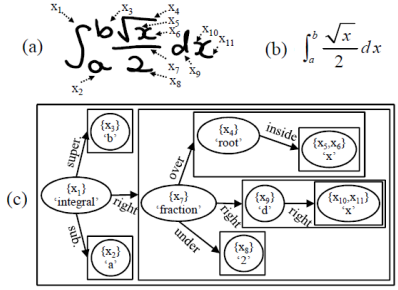


Figure 4 Symbol relation tree (SRT) representation, (a) handwritten input, (b) intended interpretation (c) the SRT representation [3]

Another approach is proposed in [11] in order to construct a syntactic and semantic free representation, the authors use a baseline structure trees (BST) that represents the hierarchical structure baselines in an expression. In the system proposed in [8], an expression is represented as a relational tree, Figure 5, containing two types of nodes:

- Non-terminals (NT), contain:
  - A set of strokes that defines a sub-expression, the root of the tree being a non-terminal that represents the solution.
  - A corresponding LateX string.
  - A type of relation (R) between its children. A relation describes not only the spatial relation but also its logical interpretation. It can be one out of the following nine: superscript, subscript, sqrt, parenthesis, sum, integral, horizontal pair, operator, and fraction.
- Terminals (T), contain:
  - A set of strokes, corresponds to a unique symbol.
  - A corresponding label.

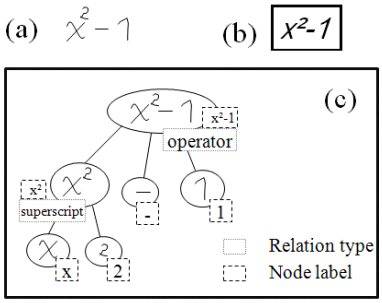


Figure 5 Relational tree (a) handwritten input, (b) intended interpretation (c) the relational tree representation

4. System evaluation

In order to know how to evaluate a ME recognition system, we have first to define what is the expected output of the system. Ideally, the output of a system should be presented in the same way as the ground truth of the input expression. The simplest way would be to count the correctly recognized expressions by comparing the output with the ground truth. However, as we mentioned before different ground truths can represent the same expression. Thus, this method of evaluating is hard to be automated. As a consequence, several published results are still computed manually or semi manually [3][5][11]. The main drawback of this intuitive measure is that one single missed symbol or segmentation error in an expression lead to a wrong recognition result. Though, more in-depth measures are required.

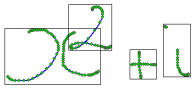
In [13], Garain proposed an integrated performance measure to evaluate recognized expressions. Segmentation, symbol recognition and structures are compared simultaneously using the presentation format of MathML of input and corresponding output expressions. Then, a *performance index* γ is calculated giving a value in the range 0 to 1 evaluating the

correctness of the recognized expression. Another measure has been proposed in [18], the tree representations of MathML of input and output expressions are transformed into Euler strings. Then, the edit distance between Euler strings is the evaluation measure. The advantage of the two previous methods is that they are totally automatic with reasonable complexity and though applicable to large expression test databases. Furthermore, they are global measures independent of the system architectures. Hence, they allow the comparison between different systems.

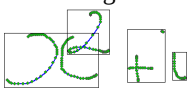
The measurements which are done at the expression level are very global and hence can hide some intricate problems. Such measures do not allow to understand the system's performance at intermediate levels. A similar example is sentence recognition, where it is important to have a word recognition rate; rather than evaluating only at the sentence level, since when a sentence is very long it is very likely to have at least one error. But, with such a methodology, there is no difference in case of one or multiple errors. In addition, the structure of the recognition system also determines what we are looking to evaluate. For example, a system with simple structure and syntactic analysis is better to be evaluated at segmentation and recognition levels.

Several different indicators can be proposed to evaluate ME recognition systems at various levels. One of the most straightforward is simply the symbol recognition rate, which is the rate of correctly recognized symbols in the expression [3][8]. For example, if the expression x^2+i is recognized as $x2+1$ we will have one recognition error. It is evident that although this measure is easy to calculate but it somehow reflects only the performance of the symbol classifier. For example, recognizing the previous expression as $\alpha^3 x1$ will indicate a 0% symbol recognition rate as well as 0% at the expression recognition level. However this does not mean that the system fails totally. To overcome this limitation, another measure is widely used, it is the segmentation rate. By definition, the segmentation rate is the rate of correctly segmented symbols. A symbol is said to be segmented correctly if all its strokes or pixels are correctly grouped together regardless if it has been recognized correctly or not. Back to the previous example, if we consider the result of the segmentation process as displayed in Figure 6-a, then the corresponding segmentation rate is of 100%, meaning that all segments of symbols are correctly extracted. Though, the segmentation step of the overall system is correctly applied. In fact, as with text recognition, symbol segmentation and recognition are interdependent measures. For example, assume that the expression x^2+i is recognized as x^2+1 because of the missing of the small dot of the "*i*" in the last component as displayed in Figure 6-b. Then, in that case, we will obtain a segmentation rate of 50% (2/4) and also the symbol recognition rate will of course be impacted with a value of 75% (3/4); while in addition recognition at the expression level would be 0%. This dependency and such issues make it essential to have performance measures on more intricate levels.

Another interesting property of the result given by the ME interpretation system would be to know to what extent the relations between components have been correctly interpreted. Assume that with the ink given in Figure 6-a, the result would be $c^3 x1$. In that case, the segmentation is fully correct (100%), the recognizer is definitively very poor (0%), but however the system has been able to correctly extract the structure. It is relevant to have access to this kind of information to be able to analyze and compare different systems.



a) Correct segmentation



b) Incorrect segmentation

Figure 6 Result of the segmentation process

In this direction, some systems were evaluated by simply checking the system's ability to correctly recognize the structure of some input expressions [9]. Such evaluation methods ignore the symbol recognition evaluation and focus whether structural analysis is applied correctly [10] [4][11].

This measure is less discussed in the literature for many reasons. First, spatial relations are not necessary explicitly present in the recognition results. In the SRT and BST trees, spatial relations are implicit and logical relations are used instead. Moreover, when spatial relations are present in the result trees, matching spatial relations requires complex tree matching algorithms. At the same time, similarly to the edit distance used on LaTeX or MathML strings, an edit distance can be calculated between two trees [16]. Nevertheless, tree edit distance use only insertion, deletion and replacement operations but not inversion due to complexity of matching operation (non ordered tree matching is a NP problem [17]). Then $(a+b)/(c+d)$ recognized as $(c+d)/(a+b)$ will have a very high edit distance even if there is only one inversion error.

5. Expressions database

Another main difference between different ME recognition systems is the expression databases. Unlike from texts, there is no ME database publicly available. As a result, each research group has its own collected database in order to test their systems. Usually a corpus of expressions covering some domains is considered. Then printed expressions are collected from available scientific documents by scanning them [11]. This task becomes more difficult with handwritten data (offline or online). In this case, large number of writers is required to obtain a good representative database [3]. Furthermore, expression labeling is a long and tedious task to achieve in order to be able to evaluate the system on different levels. As seen in sections 0 and 3, it is complex to have a unique labeling of the given database because of ME ambiguities. Conversely, writers might write differently the same expression from a fixed corpus. However, the performance of the system depends on the chosen expressions corpus which might be less or more complex. It is also inappropriate to compare the performance of systems on different datasets and corpora.

In most of the cases, the language model is adapted to the chosen corpus. While in reality the contrast case is true. Notice that when we were children, we learned how to write new expressions and not simply a set of them. Even if trainable grammar has been presented in [17], but it is not well studied in the literature and requires more investigation. Describing the domain of a recognition system with a grammar rather than a set of expressions is more benefic and assures more generalizing capacity.

We have proposed a tool [8] that allows to produce any corpus of handwritten mathematical expressions starting from previously collected isolated symbols. It generates pseudo-synthetic handwritten mathematical expressions using a stochastic layout guided by the Latex string defining the expression [7]. Although it is less interesting to test a system with synthetic data, this tool is still useful for its ability of generating easily any new handwritten expression with a specific corpus.

Currently, the expression corpus is extracted from the base "Aster" [6]. A set of 36 expressions is chosen covering a majority of mathematical domains. Each expression contains in average 11 symbols, where the total number of distinct symbols is 34 classes. Each expression is artificially generated from a base of isolated symbols collected from 280 writers [8], therefore this dataset contains 10,080 mathematical expressions. In addition, each of those 36 expressions is written twice resulting a total of 72 expressions written by 10 writers, examples shown in Figure 7.

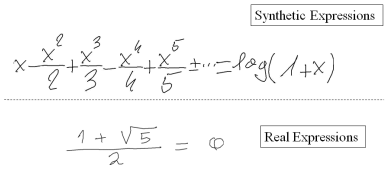


Figure 7 Examples of real and synthetic expressions

Collected and generated expressions are groundtruthed at the expression level by the corresponding "LaTeX" string. They are also labeled at symbol and stroke levels.

6. Propositions

Currently, our research focuses on the recognition of online handwritten mathematical expression recognition; however our proposition can be extended to offline handwritten ME recognition. We have proposed various methods and algorithms. One of the main concern deals with the constitution of expression database, and how to evaluate the performance of the system and compare it with other systems.

We propose the use of an open expression database for training the system, rather than using a fixed corpus. As it is the case for the text, we aim at defining a grammar and a set of mathematical symbols. To achieve this task, it is required to use a very large variety of MEs. Such variety can be found in MEs appearing in real scientific documents. An interesting source is the web pages of Wikipedia. For example, we have extracted all mathematical expressions from web pages of the French Wikipedia. Almost 77 000 LaTeX expressions were found in 7 000 web pages. Additional variety could be found in scientific papers databases and also in academic books. By doing this, we preserve symbol and relation frequencies. Then it is possible to filter this corpus to extract sub-sets for different purposes. The filtering can be based for instance on scientific domains, on length in the number of symbols, on a symbol set, on structural constraints, etc.

We propose also different levels of labeling: pixel/stroke level, symbol level, relation level, and finally at the expression level. To achieve it, the use of presentation format of MathML appears to be a good choice. This labeling allows an independent evaluation of the diverse stages of the system. We retain the following three evaluation measures: segmentation, symbol recognition and expression recognition rates. In addition, we propose also the use of a fourth measure to evaluate the ME output at the relation level.

At this relation level, we propose to define two concepts. They are the concepts of *correct relation*,

and of *found relation*. A *correct relation* is found in the result relational tree with a correspondence to the same relation in the ground truth tree **and** involving the same symbols in the children nodes of this relation. Secondly, the relation is called a *found relation* in case of not having same symbols. Matching relational trees of result and ground truth is done by a simple matching algorithm. Relations that do not have correspondences are said to be *invalid relations*.

Then, we propose also to define three degrees of dependency. A *free* relation measure does not consider any segmentation or recognition results reflecting only the capacity of finding correct relations. In this case, we might have 100% of correct relations even if segmentation and recognition are totally wrong. A second degree consist in considering also segmentation information, so that the relation is considered as correct if in addition the components involved in the relation correspond to the right segments. With the third degree, it also requested that the symbols have been correctly recognized to count a correct relation. Hence, considering also recognition information converts the correct relation measure into an integrated measure reflecting recognition, segmentation, and structure analysis performances.

Going back to the example displayed in Figure 6, if it is recognized as $c^3 \times 1$, considering the segmentation given in Figure 6-a, with the three degrees of dependency, we will get respectively the scores of 100% (the two relations superscript and operator being extracted), 100% (the two relations are based on correct segments) and 0% (none of the two relations has correctly recognized symbols). While if we consider the segmentation defined in Figure 6-b, only superscript relation has correct segments, so the three scores turns to be 100%, 50% and 0%.

7. Conclusion

This paper presents an overview of issues related to the benchmarking of handwritten mathematical expression recognition systems. Although ambiguities may exist in using MathML or LaTeX as recognition results or ground truth, they are still the most standardized way of representing MEs.

However, relational trees are widely used to represent ME during the recognition process. Although some integrated measures were proposed to evaluate recognition systems, the community tends to use other performance measures at lower levels. Segmentation, symbol recognition, and baseline rates are the most common used measures.

The domain of handwritten mathematical expression recognition achieved lot of progress. Hence, it is very important to standardize evaluation measures and construct public benchmarking to put in light the achievements of different systems. A good way of resolving those issues would be to organize ME recognition competition in the coming conferences in this domain.

References

[1] http://www.w3.org/TR/MathML/.

[2] K-F. Chan, D-Y. Yeung. An efficient syntactic approach to structural analysis of on-line handwritten mathematical expressions. *Pattern Recognition* 33: 375-384, 2000.

[3] T-H Rhee, J-H Kim. Efficient search strategy in structural analysis for handwritten mathematical expression recognition. *Pattern Recognition 42(*12): 3192-3201, 2009.

[4] R. Yamamoto, S. Sako, T. Nishimoto, S. Sagayama. On-Line Recognition of Handwritten Mathematical Expressions Based on Stroke-Based Stochastic Context-Free Grammar. 10th IWFHR, La Baule, France: 249-254, 2006.

[5] U. Garain, B. Chaudhuri. Recognition of Online Handwritten Mathematical Expressions. IEEE Transactions on Systems, Man and Cybernetics 34: 2366-2376, 2004.

[6] T.V. Raman. Audio system for technical readings. Cornell University, 1994.

[7] A.M. Awal, R. Cousseau, C. Viard-Gaudin. Convertisseur d'équations LATEX2Ink. 10th CIFED, Rouen, France: 193-194, 2008.

[8] A.M. Awal, H. Mouchère, C. Viard-Gaudin. Towards handwritten mathematical expression recognition. 10th ICDAR, Barcelona, Spain:1046-1050, 2009.

[9] D Prusa., V. Hlavac. 2D Context-Free Grammars: Mathematical Formulae Recognition. The Prague Stringology Conference: 77-89, 2006.

[10] R. Geneo, J.-A. Fitzgerald, T. Kechadi. A Purely Online Approach to Mathematical Expression Recognition IWFHR: 255-260, 2006.

[11] R. Zanibbi, D. Blostein. Recognizing Mathematical Expressions Using Tree Transformation. Pattern Analysis and Machine Intelligence(24): 1455-1467, 2002.

[12] A. Belaid, J-P. Haton. A syntactic approach for handwritten mathematical formulae recognition. *Pattern Analysis & Machine Intelligence* (6): 105-111, 1984.

[13] U. Garain, B.B. Chaudhuri. A corpus for OCR research on mathematical expressions. *IJDAR* (7): 241-259, 2005.

[14] K.-F. Chan, D.-Y. Yeung. Mathematical expression recognition: A survey. *IJDAR*(3): 3-15, 2000.

[15] E. Tapia, R. Rojas. A Survey on Recognition of on Line Handwritten Mathematical Notation. Freie Universit¨at Berlin, Institut fur Informatik, Germany, 2007.

[16] P. Bille. A survey on tree edit distance and related problems. Theor. Comput. Sci., Vol(337): 217-239, 2005.

[17] J. F. Hull. Recognition of mathematics Using a Tow-Dimensional Trainable Context-free Grammar. Massachuestts Institute Of Technology, 1996.

[18] K. Sain, A. Dasgupta, U. Garain. A Tree Matching Based Performance Evaluation for OCR of Mathematical Expressions. *IJDAR* (2010).