Symbol Segmentation for Handwritten Math Formula

Aurel Pjetri
aurel.pjetri@stud.unifi.it

Pietro Bongini
pietro.bongini@stud.unifi.it

Abstract

Math expressions are commonly used in scientific area. For this reason, Recognizing handwritten expressions is util for many applications. In this paper we present a technique for representation of handwritten formula using Line-of-sight (LOS) graph. Subsequently, in segmentation phase, we combined some geometric features with an histogram feature. In classification, we use a binary random forest classifier to identify which LOS graph edges represent strokes of the same symbol. CROHME 2012 dataset is used for experimental results.

1. Introduction

The work treated in this paper concern the representation and segmentation of Math handwritten formulae. These formulae are written on tablets or other touch-sensitive devices. The aim of this work is to recognize, trough a classifier, which strokes belong to the same symbol.
In section 2 of this paper, we propose Line-of-sight (LOS) graph to represent handwritten formulae. Then in section 3 we describe CROHME 2012 dataset. In section 4 we focus on preprocessing and segmentation phases.
Finally, section 5 we show and analyze the results on CROHME 2012 dataset and in section 6 we draw our conclusions on this project.

2. Line-of-sight representation

In CHROME 2012 dataset, all strokes of the expression are represented by coordinates. We need to create a graph-based representation of the formula in order to highlight the spatial relationship between the strokes. For this purpose we use Line-of-sight (LOS) graph [1]. This kind of graph, as we can see in Figure 1, generates an edge between two strokes every time that a stroke can be ”seen” from the bounding box of the other stroke and vice versa. In the implementation of LOS graph is also important the use of convex hull calculated for every stroke. In fact, for each stroke we search an unobstructed angle between the bounding box of reference stroke and the vertices of the convex hull the other strokes.
At the end of this phase, we obtain a graph whose edges connect strokes with LOS relationship.

3. CROHME 2012 dataset

CHROME 2012 dataset is composed by a test set of 489 math expressions and a training set of 1339 expressions.
In test set all formulae are represented with inkml files. This type of file is used generally to represent online handwritten formulae. The strokes which compose the expression are represented by coordinates and are separated between each other. The training set is an inkml file and it represents the expressions in the same way of data set. Moreover, it contains a MathML representation of the formula and the ground truth.

4. Preprocessing and Segmentation

When we obtain the the LOS graph the aim is to classify the edges whose strokes belong or not to

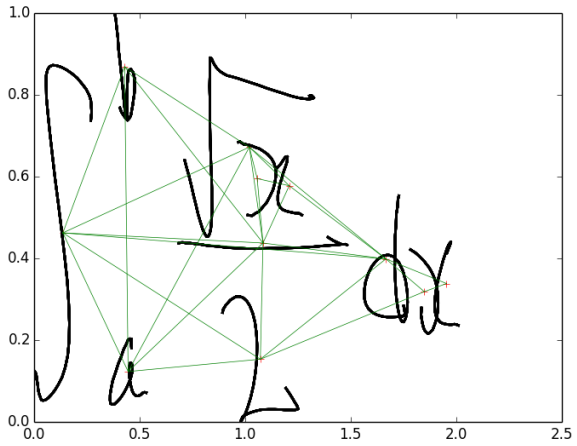

Figure 1. Example of Line-of-Sight (LOS) Graph for a Handwritten Expression. Small red nodes represent bounding box centers for eleven handwritten strokes. Edges represent mutually visible strokes.

the same symbol. An important phase of our work is the preprocessing. In fact, data have to be extracted and elaborated. We implement duplicate point deletion, size normalization, the remotion of hooks and we make the traces smoother. These operations are accomplished before the creation of LOS graph.
We delete point which have the same coordinates because they are redundant. Then we normalize coordinates in [0,1] interval. To make the expression smoother, we replace each point with the average of the previous, the current and the following point. Finally, we filter the hooks from the symbols to reduce the noise.
Segmentation. Our segmentation consists of two steps:
1) The construction of a LOS graph as it is described previously.
2) We use a binary classifier to classify all edges in the LOS graph.
For this second point we create a random forest classifier using Python Scikit-learn library[4]. We implement 104 features. These features are geometric, distance-based and histograms.
An histogram characterizes the position and density of expression's points. In our work we use histograms for couple of strokes (which compose

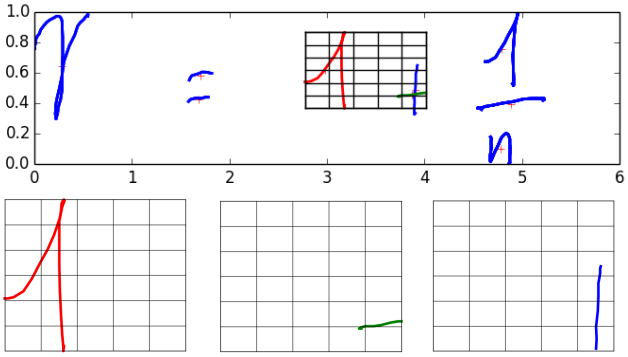

Figure 2. Example of using histograms on math expressions. An edge between the '1' and the horizontal line of '+' is considered. We produced three different histograms: one for the parent stroke (on the left), one for the child stroke (on the center), one for other strokes (on the right).

an edge) that can be merged. So, the region composed by the two stroke is divided into bins. In this way we can compute the density of strokes' points in the histogram.
As we can see in Figure 2, we use three different histograms: one histogram for each of the two strokes and one for all the other strokes in the neighborhood of the two strokes. The number of bin used is 30 (5 x 6). Therefore 90 features are obtained from histograms.

The other 14 features are geometric and distance based. We implement horizontal distance, size difference [6], minimum point distance [5], maximum distance, overlapping area, horizontal overlapping of bounding box [3], distance between stroke first points, last point distance and parallelity [2].
The geometric features are: distance between centers of mass, horizontal offset between last point of the first stroke and first point of the second stroke, vertical distance between bounding box centers, writing slope and writing curvature.

5. Experimental result

As in CROHME 2012 the training set is present in ground truth of the math expressions we can label all the edges with '*' if the strokes are part of the same symbol or with '_' if are undefined. Our

random forest classifier process 104 features, use 50 trees and a maximum depth of 40. Moreover, the Gini criterion is used for splitting. We use the first 1000 file inkml of CROHME 2012 training set to train the classifier and the other 339 for testing. The predictions of our classifier are the label "*" and "_" assigned to each edge. The fitting time of our classifier is about five minutes. The classification rate for merging or splitting stroke pairs is quite high. We obtain the 96,4% of precision in labeling edges.

6. Conclusion

In this paper we have presented an efficient technique for representation and segmentation of math formula. The implemented method, as previously observed, is quite accurate in labeling edges.
However, some improvements can be brought in fitting time of the classifier.

References

[1] Lei Hu and Richard Zanibbi. Line-of-sight stroke graphs and parzen shape context features for handwritten math formula representation and symbol segmentation. *Proc. ICFHR*, 2016.

[2] Stefan Lehmberg, H-J Winkler, and Manfred Lang. A soft-decision approach for symbol segmentation within handwritten mathematical expressions. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 6, pages 3434–3437. IEEE, 1996.

[3] Scott MacLean and George Labahn. A new approach for recognizing handwritten mathematics using relational grammars and fuzzy sets. *International Journal on Document Analysis and Recognition (IJDAR)*, 16(2):139–163, 2013.

[4] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[5] Kenichi Toyozumi, Naoya Yamada, Takayuki Kitasaka, Kensaku Mori, Yasuhito Suenaga, Kenji Mase, and Tomoichi Takahashi. A study of symbol segmentation method for handwritten mathematical formula recognition using mathematical structure information. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 630–633. IEEE, 2004.

[6] SHI Yu, LI HaiYang, and Frank K Soong. A unified framework for symbol segmentation and recognition of handwritten mathematical expressions. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 854–858. IEEE, 2007.