Graph matching over Hypothesis Graphs for the Analysis of Handwritten Arithmetic Operations

Arnaud Lods, Eric Anquetil, Sébastien Macé

Graph matching over Hypothesis Graphs for the Analysis of Handwritten Arithmetic Operations

Arnaud Lods
Learn&Go, Univ Rennes, CNRS, IRISA
F-35000 Rennes
arnaud.lods@learn-and-go.com

Éric Anquetil
Univ Rennes, CNRS, IRISA
F-35000 Rennes
eric.anquetil@irisa.fr

Sébastien Macé
Learn&Go
F-35000 Rennes
sebastien.mace@learn-and-go.com

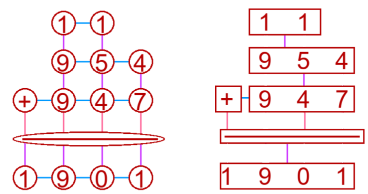
***Abstract*—This paper presents a preliminary research work for the analysis of handwritten arithmetic operations in the context of e-education. Given a mathematical exercise, an answer in the form of an arithmetical operation is expected from a child. This answer can be represented by a graph containing both expected numbers and their corresponding relationship one to another. We propose to compute several hypotheses Fuzzy Visibility Graph of symbols over a child's input. The widely used A* algorithm to compute exact graph edit distance is applied over those graphs to match the expected answer. To reduce the search complexity, simplified graphs of operands are first generated and used. Each operand is a sub-graph of the original graph, the algorithm is then applied on the pairs of matched sub-graphs. The hypothesis graph with the smallest graph edit distance with the expected graph and the matched differences can be used to produce an adapted feedback. The result of an experiment over a given example is presented.**

***Index Terms*—graph edit distance, A* algorithm, handwritten arithmetic operation, fuzzy visibility graph, graph analysis.**

I. Introduction

The improvement of pen-based devices over the recent years offers new ways of teaching in school. We now have the opportunity to provide interfaces with complete liberty for the children. Thus one can devise an adapted system to create personalized and extensive feedback on mistakes made without a costly human analysis. For our domain of application, learning mathematics, such problem can be represented by a graph $G_T = \{V_T, E_T\}$ where V_T, the set of vertices, represents the set of mathematical symbols and E_T, the set of edges, represents the mathematical relationships between symbols, as displayed in Fig. 1a.

The scientific problematic boils down to a problem of graph matching. Given a source graph G_S produced from the child's input and a target graph G_T which is the expected solution, we are looking for the best graph edit distance (GED) that minimizes the number of required operations to transform G_S into G_T. The community of pattern recognition has since long tackled the problematic of graph matching. A study in [1] covers most of the related works on the matter, from exact graph matching to inexact graph matching. Recent works focus on the improvement of the computation of a higher bound for classification purpose. In [2] the authors present several optimization algorithms. However, to compute the best GED, the A* algorithm [3] is a robust but costly solution. In [4] an algorithm is presented to compute the exact GED in half the time and with a lower memory consumption.



(a) Target graph of symbols G_T (b) Target graph of operands G_TO

Fig. 1: The graph G_T generated with rules over the arithmetic operation: 954 + 947.

As a first milestone for our research work, we present in Section II the application of a naive A* algorithm on an intermediate then complete representation of the graph to reduce overall computational complexity. Then we present a first experiment in Section III. In Section IV we discuss our ongoing research work on this problematic.

II. Our system

To produce a graph that can be matched to the target graph presented earlier, we extend a system previously built in [5] called Fuzzy Visibility Graph. Fuzzy landscapes, corresponding to fuzzy areas in the space each representing an observed mathematical relationship, are learned over pairs of symbols and are used to build the graph from a set of symbols. This set of symbols is produced by two different Random Forest classifiers to segment the strokes into symbols and to classify those symbols with two set of geometrical features. The classifiers give us a probability for each class. As children are still in the learning process and can have very different and ambiguous input, instead of generating one graph G_S to match to the target graph G_T, we generate a set of hypotheses graphs $\mathcal{G} = \{G_{S1}, G_{S2}..G_{Sn}\}$ for each uncertain classification. As such the goal is to match each graph G_{Si} to the target graph G_T to find the graph with the lowest GED. Fig. 2 display an example of two hypotheses graphs made on the same input where the segmentation was ambiguous.

As stated earlier, a simplified A* algorithm is used to match each graph G_{Si} to G_T. The costs of transformations (Eq. 1) are defined to put more emphasis on *how* the child has built his

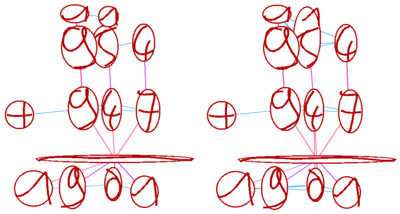


Fig. 2: The two Fuzzy Visibility Graphs of symbols hypotheses G_{S1} and G_{S2}.

operation with a higher cost for edges deletion/insertion rather than *what* he has written with a lower cost for different vertices matching. For vertices substitution, $||l_u - l_v||$ currently refers to the Levenshtein distance on the string of symbols labels. For edges substitution, $f(r_{(u,v)} - r_{(u',v')})$ has a cost of 0 if the relationship is similar or a cost of 1 otherwise. We also impose a restriction on the deletion and insertion of vertices. Let N_T and N_S be respectively the number of vertices for the target and source graph considered. We only allow the insertion of vertices if $N_T > N_S$ and the deletion of vertices if $N_T < N_S$. In other words, as we are working with operations that are expected to contain lots of mistakes from both positioning and calculus, the goal is to force a matching, even if symbols have different labels, rather than deleting and adding new vertices with correct labels if the child made mistakes on numbers. Thus, if we observe too mane or too few vertices, we allow a number of insertion/deletion operations as to match exactly those which are missing/in excess. Once the best GED for the pairs of graphs of operands is computed, we can compute the GED on the graph of symbols for each hypothesis.

$$\begin{cases} c(u \to v) = ||l_u - l_v|| \\ c(u \to \epsilon) = c(\epsilon \to v) = \tau \\ c((u,v) \to (u',v')) = f(r_{(u,v)}, r_{(u',v')}) \\ c((u,v) \to \epsilon) = 0 \\ c(\epsilon \to (u',v')) = 2\tau \end{cases} \tag{1}$$

We now have the matched pairs of operands between each graph G_{SOi} and G_{TO}, each operand being a sub-graph of the original graph of symbols. The algorithm A* is applied to find the best GED for each pair of matched sub-graphs. We identify two types of edges: internal edges between symbols of the sub-graphs, that are matched as usual, and external edges between symbols and operands. For the latter case, the cost of edges edition is computed from a symbol vertex to an operand vertex: $\{(u, V)\}$. This allows us to avoid matching larger sub-graphs of related symbols which would in turn increase a lot the number of matches. The lowest sum of the GED and its corresponding edit path on the set of sub-graphs represent the best matching for each graph G_{Si} to G_T. The hypothesis graph with the lowest cost is kept and the resulting GED represents the analysis result with the differences on what was expected.

III. Experiment

We ran the system on the given arithmetic addition displayed in Fig. 2. Two hypotheses are generated while producing the fuzzy visibility graph (Fig. 2). These hypotheses (G_{S1}, G_{S2}) are converted to two operands graphs (G_{SO1}, G_{SO2}) using the same rules to produce G_{TO}. The two matching steps presented in Section II are applied and the expected hypothesis graph has the lowest GED with a missing relationship between the + and the horizontal bar. Otherwise if we were to select the second graph as the best hypothesis, then the edit path: Insert node, Insert relation, Insert relation on the carry-over in G_T give us the information that the carry-over was forgotten, and thus an adapted feedback can be displayed.

IV. Conclusion and perspectives

We presented the first step of a system that make use of fuzzy visibility graph representation to compute the graph edit distance from hypotheses graphs to a given target graph. The standard A* algorithm is used, and an intermediary representation is used to match sub-graph of the initial pair of graphs to reduce the computation time. Simple cost functions for the graph edition are used. On a first sample, the correct hypothesis graph is matched at a lower cost and the expected matching between symbols is found in a reasonable time.

In the future we intend to learn the cost functions on a complete dataset to take into account the probabilities of the classifiers producing the initial hypotheses graphs. Then we will be able to produce experimental validation on a corresponding dataset. Another step is to implement a more efficient algorithm to reduce computation for complex operation, and to merge the different hypotheses graphs in a single graph with vertices and hypotheses edges to avoid matching the same pairs several times. Other improvement can be made on the initial graph representation. We also intend to expand experiments on a large dataset collected from children aged 6 to 8-years-old to evaluate the effectiveness of the matching with several methods and different types of arithmetical operations. Eventually transformations resulting from the graph edit distance will be used to produce adapted feedback for the children.

Acknowledgment

With the support from the LabCom **ScriptAndLabs** founded by the **ANR** ANR-16-LVC2-0008-01. With the support from the ANRT.

References

[1] M. Vento, "A long trip in the charming world of graphs for pattern recognition," *Pattern Recognition*, vol. 48, no. 2, pp. 291–301, 2015.

[2] K. Riesen, "Structural pattern recognition with graph edit distance," *Advances in computer vision and pattern recognition, Cham*, 2015.

[3] K. Riesen, S. Fankhauser, and H. Bunke, "Speeding up graph edit distance computation with a bipartite heuristic.," in *MLG*, pp. 21–24, 2007.

[4] Z. Abu-Aisheh, R. Raveaux, J.-Y. Ramel, and P. Martineau, "An exact graph edit distance algorithm for solving pattern recognition problems," in *4th International Conference on Pattern Recognition Applications and Methods 2015*, 2015.

[5] A. Lods, E. Anquetil, and S. Macé, "Fuzzy visibility graph for structural analysis of online handwritten mathematical expressions," in *2019 15th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, 2019.