Using machine learning to detect non-triviality of knots via 
colorability of knot diagrams

Alexei Lisitsa\textsuperscript{1} and Alexei Vernitski\textsuperscript{2}

\textsuperscript{1} University of Liverpool, Liverpool, UK
\texttt{a.lisitsa@liverpool.ac.uk}

\textsuperscript{2} University of Essex, UK
\texttt{asvern@essex.ac.uk}

Abstract

We apply machine learning to combinatorial knot theory, specifically, we consider a 
classical problem of deciding if a knot diagram represents the trivial knot as a classification 
problem. As a part of this process, we use a reformulation of this problem expressed via 
so-called Fox coloring of knot diagrams or, more generally, coloring knot diagrams with 
elements of algebraic structures called quandles.

Introduction

Knot theory is a branch of mathematics in which being assisted by machine learning feels 
especially attractive and promising, since small and numerous illuminating examples and counterexamples can be built successfully; let us discuss recent examples of such studies. In \cite{24} the authors consider the problem of classification of 5 types of simple knots in the polymers where polymers are encoded by sequences of monomers, and train feed-forward neural networks and (with much better results) recurrent neural networks for this classification task. In \cite{15} encoding of knots by rectangular diagrams was used and bidirectional LSTM networks were trained to recognize 36 knots types. In \cite{12} reinforcement learning was used to untangle knot diagrams presented in braid encoding. In \cite{16} and in our ongoing research we used reinforcement learning (multi-agent Q-learning and deep learning) to untangle braids. In \cite{18} we compared performance of machine learning in testing realizability of Gauss diagrams with that of humans. In \cite{4, 5, 6} machine learning is applied to studying various knot invariants.

A \textit{quandle} is an algebraic structure whose binary operation is a generalization of the operation of conjugation in a group; see, for example, \cite{7}. Quandles were introduced in \cite{14, 23} as a powerful knot invariant. To be precise, the fact whether the arcs of a knot diagram can be colored by elements a given quandle (with certain conditions satisfied at the crossings) is a knot invariant. In \cite{11, 10, 9} this approach was combined with automated reasoning and SAT solving to detect trivial knots and, more generally, to recognize knots; see also \cite{3}. In this study we use machine learning to recognize colorability of knot diagrams with quandles and, therefore, to detect non-trivial knots.

In general, the efficient detection of non-trivial knots remains a challenge. The problem belongs to a complexity class $\text{NP} \cap \text{co-NP}$ \cite{21, 13} and polynomial time algorithms for it are unknown. Very recently quasi-polynomial time algorithm for unknot detection was proposed in \cite{22}. The recent work on machine learning applied to unknot detection \cite{24, 15, 12} has shown encouraging performance of learned classifiers for this algorithmically difficult problem. The research reported in this paper continues the work in this direction and has the following novel features. We use the most traditional encoding of knots by realizable Gauss codes/diagrams \cite{1} and by more recent petal diagrams \cite{2}. We apply classical machine learning algorithms, such as multilayered perceptrons/feed-forward neural networks. We use approximations of unknotedness by quandle colorability.
Methodology and details of implementation

In our experiments in this study we used two approaches to representing knot diagrams in the computer. In one approach, we used classical Gauss codes/diagrams [1]. To produce a dataset, a pre-defined amount of random Gauss diagrams is generated using our tool [17], then diagrams are checked for realizability using the algorithm for signed realizability from [20], and then, the variants are produced by varying at each crossing, which arc goes above or below the crossing. In another approach, we used petal diagrams of knots [8, 2], and to produce a dataset, we chose random permutations indicating in which order arcs pass behind each other at the crossing. Whereas standard permutation matrices were successful, we were more successful when we represented permutations by new ternary matrices, inspired by an encoding of permutations as a certain list of numbers called the Lehmer code or the inversion table [19]. Namely, we represent a permutation \( p \) by a matrix in which the entry at \( i, j \) is equal to 1 (or \(-1\), or 0) if \( p \) does not swap the order of \( i \) and \( j \) (if \( p \) swaps the order of \( i \) and \( j \), if \( i = j \)).

The second step in creating the training set and the test set was finding out, for these randomly generated knot diagrams, whether they represent the trivial knot or a non-trivial knot. In this study, instead of attempting to untangle the knot, we replace this question by the question of colorability by certain quandles. At this step, we used two approaches. One approach was coloring by quandles of small sizes. Another approach was coloring by quandles induced by cyclic groups, which is equivalent to finding the number called the determinant of the knot diagram. Why do we consider the question of quandle colorability instead of the question of being the trivial knot? There are several reasons for this. Firstly, quandle colorability is an interesting research area in its own right [11, 10, 9, 3]. Secondly, it is known that for small sizes of diagrams, a diagram represents the trivial knot if and only if it cannot be colored by one of several small quandles [11, 3] or has a particular value of the determinant. Thirdly, even for larger diagrams, colorability by quandles of small size is a good approximation to being a non-trivial knot.

Experiment results

Table 1 presents some of the results of ongoing work in the first approach. G and EG in the names of datasets are referring to Gauss and Extended Gauss notation, respectively, in a sense of [1]; SQ-\( N \) is referring to the the initial segment of \( N \) quandles from a sequence SC of all simple quandles of small size used in [10]. #Frames are referring to the number of different unsigned diagrams used in the generation of datasets of signed diagrams.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size of dataset</th>
<th>Size of diagrams</th>
<th>#Frames</th>
<th>Quandle set</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-SQ-EG-8all</td>
<td>3072</td>
<td>8</td>
<td>6</td>
<td>SQ-1</td>
<td>75.3%</td>
</tr>
<tr>
<td>2-SQ-EG-8all</td>
<td>3072</td>
<td>8</td>
<td>6</td>
<td>SQ-2</td>
<td>65.2%</td>
</tr>
<tr>
<td>5-SQ-EG-8all</td>
<td>3072</td>
<td>8</td>
<td>6</td>
<td>SQ-5</td>
<td>62.3%</td>
</tr>
<tr>
<td>25-SQ-EG-8all</td>
<td>3072</td>
<td>8</td>
<td>6</td>
<td>SQ-25</td>
<td>55.2%</td>
</tr>
<tr>
<td>3Q-11-G-1x1024</td>
<td>2048</td>
<td>11</td>
<td>1</td>
<td>SQ-1</td>
<td>86.5%</td>
</tr>
<tr>
<td>3Q-11-G-4x250</td>
<td>2048</td>
<td>11</td>
<td>4</td>
<td>SQ-1</td>
<td>65.3%</td>
</tr>
<tr>
<td>3Q-11-G-20x200</td>
<td>8000</td>
<td>11</td>
<td>20</td>
<td>SQ-1</td>
<td>59.2%</td>
</tr>
</tbody>
</table>

Table 1: The accuracy of MLP (Multi-Layered Perceptron) of recognition of colorability of knot diagrams by sets of quandles (by any in a set); diagrams are encoded by “one hot” encoding from [18]; 70% training/30% testing split; WEKA Workbench [25] is used with default settings for MLP.

Our initial results shows that the classical machine learning model of perceptron demon-

---

1 random permutations and encoding of diagrams by permutations [17] are used here
strates good performance for the recognition of quandle colorability of knot diagrams, especially for the cases of colorability by a single quandle (SQ-1 set consists of single 3-element quandle) and for the datasets with small number of frames. Increasing the number of quandles and the number of frames leads to some degradation of the accuracy of learned models.

In the second approach we considered petal diagrams\(^2\) of size 7 (there are \(7! = 5040\) petal diagrams of this size, in total) and trained a binary classifier to distinguish between the trivial knot and non-trivial knots. We used the training set consisting of an equal number (500+500 = 1000) of petal diagrams whose determinant is 1 (they represent the trivial knot) and petal diagrams whose determinant is not 1 (they represent non-trivial knots \(3_1, 4_1, 5_1,\) or \(5_2\) [2]). If permutations are presented by their permutation matrices, some learning occurs successfully, with the accuracy on the training set 100% and the accuracy on the test set around 80%. We also introduced a new way of presenting permutations by ternary matrices (see the definition above), instead of permutation matrices, and the accuracy on the test set increased to around 96%. For these experiments, we use Keras and TensorFlow in Python, and the binary classifier is a feed-forward neural network with one hidden layer of size 100, with an input layer of size \(7 \times 7 = 49\) and a softmax output layer. Our results for this approach indicate that indeed the recognition of diagrams with determinant 1 is learnable and the accuracy is dramatically increased by using novel encoding by ternary matrices.

References


\(^2\)We are grateful to Chaim Even-Zohar for sharing some of his code with us.
The Class File


