Data augmentation in mathematical objects

Tereso del Río and Matthew England
Coventry University
delriot@coventry.ac.uk

1 Introduction

Cylindrical Algebraic Decomposition (CAD) is a mathematical algorithm that given a set of polynomials decomposes the space into regions in which they are sign invariant. This algorithm requires the choice of a variable ordering. In fact, the variable ordering can have a huge impact on the complexity [BD07].

Since the community realised the importance of variable ordering, some heuristics and machine learning models have been proposed for this task. One of the barriers that the researchers have found while training machine learning models is the unbalancedness and lack of existing data.

In this text, a new idea is proposed to both balance and augment the existing datasets by exploiting the arbitrarity of the variable representations (names). We note that this idea has been independently proposed also by [HHP+23].

2 Main idea of this paper

Data augmentation consists on generating new instances from existing ones. Imagine we are interested in training a model to detect the direction of an arrow. We know that a picture of an arrow pointing to the right that is rotated 90 degrees clockwise results in an arrow pointing down. If the dataset of arrows is imbalanced, this idea can be used to balance the dataset.

It is possible to exploit this idea further, by generating three extra images from each of the images in the original dataset, resulting in a balanced dataset of four times the original one. This can also help fighting against biases in the dataset (e.g. arrows in traffic signals never point downwards so the model could learn to recognise traffic signals)

Similarly, given a set of polynomials (e.g. $\{x_1^2-x_2, x_3^3-1\}$) for which the ideal variable ordering has been computed ($x_2 \succ x_1 \succ x_3$). Simply by swapping the names of the variables x_1 and x_2 we obtain the new set of polynomials (in the example, $\{x_2^2-x_1, x_3^3-1\}$), in which we know, without any computations what is the ideal variable ordering (in the example, $x_1 \succ x_2 \succ x_3$).

3 Experiments

For our experiments the methodology in [FE19] is followed. The obtained dataset has 1019 instances (labelled sets of polynomials). This dataset is unbalanced, the sizes of different classes are as follow:

0: 406 1: 93 2: 135 3: 51 4: 202 5: 132

3.1 Datasets

We split this dataset into an original testing dataset containing 20% of the instances and an original training dataset containing the rest.

We randomly change the label of each instance of the original datasets (training and testing), obtaining a balanced training dataset and a balanced testing dataset.

Nothing is stopping us from using the six possible reorderings to the dataset. By doing this, we obtain perfectly balanced testing and training datasets with six times the size they had originally.

3.2 Training the models

As in [FE19], cross-validation is used to choose the hyperparameters of various ML models available in sklearn. This methodology is repeated with the three training datasets available, normal, balanced and augmented. And all these models are tested in the balanced dataset that was obtained after randomly changing the labels of the original testing dataset.

Training dataset	Normal	Balanced	Augmented
KNN	0.3	0.42	**0.55**
DT	0.35	0.43	**0.54**
MLP	0.35	0.45	**0.47**
SVC	0.23	0.29	**0.48**
RF	0.46	0.53	**0.61**

Table 1: Accuracies in the balanced testing dataset of the models trained in the different training datasets.

4 Conclusion and further work

It has been seen that using a balanced dataset instead of an unbalanced one of the same size the accuracies of the models improved on average 27%. And when the dataset was augmented to multiply the size by six the accuracies of the models improved on average 63%. This increases in accuracy are an amazing result.

This simple idea to augment the dataset for the purpose of CAD can be easily generalised to any mathematical dataset containing objects in which the variables have arbitrary representations.

Furthermore, similar data augmentation ideas are possible in mathematical objects, one should reflect on which parts of the representation of a mathematical object are arbitrary.

For example, in the recently published [LC19] where mathematical expressions are represented as natural text, the order of the operands in commutative operations is arbitrary (e.g. $x \wedge 2 + y * z$ is the same expression as $z * y + x \wedge 2$). This could be exploited to generate an exorbitant amount of new instances that do not require labelling.

References

[BD07] Christopher W. Brown and James H. Davenport. The complexity of quantifier elimination and cylindrical algebraic decomposition. *Proceedings of the International Symposium on Symbolic and Algebraic Computation, ISSAC*, pages 54–60, 2007. Publisher: ACM Press ISBN: 9781595937438. doi:10.1145/1277548.1277557.

[FE19] Dorian Florescu and Matthew England. Algorithmically generating new algebraic features of polynomial systems for machine learning. *CEUR Workshop Proceedings*, 2460, 2019. arXiv: 1906.01455 Publisher: CEUR-WS. doi:10.48550/1906.01455.

[HHP$^+$23] John Hester, Briland Hitaj, Grant Passmore, Sam Owre, Natarajan Shankar, and Eric Yeh. Revisiting Variable Ordering for Real Quantifier Elimination using Machine Learning. 2023. Publisher: arXiv Version Number: 1. URL: https://arxiv.org/abs/2302.14038, doi:10.48550/ARXIV.2302.14038.

[LC19] Guillaume Lample and François Charton. Deep Learning for Symbolic Mathematics. 2019. Publisher: arXiv Version Number: 1. URL: https://arxiv.org/abs/1912.01412, doi:10.48550/ARXIV.1912.01412.