Automatic Derivation of Formulas by Graph Embedding and Pattern Matching Network*

MinCong Luo
China Institute of Atomic Energy
Peking
luomincentos@ciae.ac.cn

ABSTRACT

Automatic formula derivation (symbolic computation) is a challenging problem that has been widely used in scientific research. The key to this research is to compute and output results automatically based on input formulas, which requires large amount of prior mathematical knowledge and complex derivation techniques. The previous researches mainly rely on pattern matching, which requires large number of complex algebraic systems and basic formula libraries.

In this work, a formula derivation framework with graph embedding and pattern matching network is proposed. We represent high-order logic formula as graph, which is embedded into a feature matrix with structure and logic information. Then the pattern matching network(PatternNet) is utilized for formula derivation pattern selection. The differential equation solving dataset(DESD) for formula derivation training is also proposed and our approach achieves best results compared to baseline models.

KEYWORDS

Automatic Formula Derivation ,Graph Embedding,Pattern Network

ACM Reference Format:
MinCong Luo. 2018. Automatic Derivation of Formulas by Graph Embedding and Pattern Matching Network. In *Proceedings of ACM SAC Conference (SAC'19)*. ACM, Limassol, Cyprus, Article 4, 3 pages. https://www.sigapp.org/sac/sac2019/10.475/123_4

1 INTRODUCTION

Automatic formulas derivation (symbolic computation) is widely used in various fields of scientific research and engineering. Because of the requiring of large amount of prior mathematical knowledge, including basic formulas and related theorems, and advanced logic reasoning ability [3], the previous researchs are mostly based on derivation by pattern matching [2], which is limited to do automatic derivation of formulas in a particular sub-domain, such as geometric calculation [1], calculation of electromagnetic [6], or automatic proof of theorems [4].

We propose a general automatic formulas derivation framework, which is not limited to the derivation in a sub-domain of scientific research. First we represent the formula as a graph [5], then the graph is embedded into the feature matrix with structure and logic information via the neural network. The pattern matching network is then proposed to derive the formula transformation mode selection, specifically, each node (as a sub-formula) in the graph needs to select the transformation mode by the pattern matching network (PatternNet)(e.g. $sin^2(x)$ is a sub-formula of $sin^2(x)+g^2(x)$ and can optionally be transformed to $1-cos^2(x)$ or $(1-cos(2x))/2$, or no transformation). The derivation result X_r can be derived by our approach based on the inputing formula X_f needs to be derived and the target X_t.Our main contributions:

- **Graph representation and embedding for formula.** Which maps the formulas into the computable space.
- **PatternNet for formula derivation.** Which can compute the corresponding transformation mode for each sub-formula to achieve the automatic derivation.
- **Dataset DESD for training.** We propose the differential equation solving dataset (DESD) for model training, which contains more than 2000 differential equations and relevant solutions.

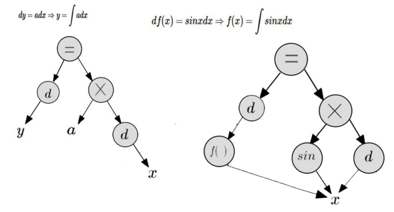
$dy = adx \Rightarrow y = \int adx$
$df(x) = sinxdx \Rightarrow f(x) = \int sinxdx$
=
d
×
y
a
d
x
=
d
×
f( )
sin
d
x

Figure 1: A mathematical formula is represented as a directed graph.

2 AUTOMATIC DERIVATION OF FORMULAS

2.1 Formula Representation

We propose a method representing a mathematical formula as a directed graph $G(V, E)$, where the node set V represents a function($sin(\cdot)$,$exp(\cdot)$), an operator($+$,$\times$), or a symbol(x,π)

*The Source Code:https://github.com/Luomin1993/SciBot in the formula, and the edge set E represents the affiliation of the nodes in the formula.

2.2 Graph Embedding

It is necessary to embed the formula into the computable vector space, so that we can measure the difference between the formulas and use the feature vectors to train the neural network for fuzzy reasoning. Set the initial feature vector of each node v in the graph to be x_v^0, which can be updated in an iterative way:

$$x_v^{ID} = F_c(\underbrace{W_O \frac{1}{d_O} \sum_{u \in O} F_O(x_u, x_v)}_{Nodes\quad output}$$

$$+ \underbrace{W_I \frac{1}{d_I} \sum_{u \in I} F_I(x_u, x_v)}_{Nodes\quad input} + \underbrace{W_H \frac{1}{d_H} \sum_{u \in H} F_H(x_u, x_v)}_{Nodes\quad at\quad the\quad same\quad level})$$

where we build a node structure prediction network for graph embedding. $x_v^{ID} \in R^{L_m}$ is the encoding vector of subgraph starts from the node v parsed by breadth-first traversal, which is unique. The node set O represents the nodes start from the node v, the node set I represents the nodes reaching v, the node set H represents nodes at the same level with v. $d_O = |O|$, and $F_O(x_u, x_v)$ is a network measures the correlation between node v and node u. In the training step, x_v is the learnable parameter can be trained, x_u is the input. Finally we get the feature matrix $X_G = [x_1^T, \cdots x_i^T, \cdots x_N^T]$ for the entire formula graph with structure and logic information.

2.3 Pattern Matching Network

Pattern matching network gives the result based on the inputing formula X_f needs to be derived and the target X_t, that is, transformation mode T_i for per node v_i in formula X_f(transformation mode T_i belongs to a finite formula transformation methods set T),then the final result X_r is calculated. PatternNet can be defined as:

$$T_r^{D_T \times L} = fc(W_f^{H \times D} X_f^{D \times L} + W_t^{H \times D} X_t^{D \times L} + b)$$

where $fc(\cdot)$ represents the fully connected network, $T_r^{D_T \times L}{}_{:,i} \in R^{D_T}$ is the transformation mode vector for node i in the transformation methods space T.

2.4 Expiriment And Evaluation

Baseline models for image generation:

- **DNN-5,DNN-10** and **DNN-20:** The encoding matrix X_f and X_t are directly inputed into a 5-layer neural network in DNN-5 to predict transformation mode T_r, without graph embedding.DNN-10 and DNN-20 are same as above but 10-layer and 20-layer neural network are used.
- **Pure pattern matching:** Derivation method only based on pattern matching.

Algorithm 1 The embedding algorithm using SGD with learning rate η

```
Input: Graph G; Output: Embedding vector x_v for node v in Formula;
1: for $n = 1$ to $S$ do do
2:     For $v \in G$;
3:         $x_v^{ID} = F_c(W_O \frac{1}{d_O} \sum_{u \in O} F_O(x_u, x_v) + W_I \frac{1}{d_I} \sum_{u \in I} F_I(x_u, x_v) + W_H \frac{1}{d_H} \sum_{u \in H} F_H(x_u, x_v))$;
4:     $L_v \leftarrow ||x_v^{ID} - \hat{x}_v^{ID}||$;
5:     $x_v \leftarrow x_v - \eta \bigtriangledown_{x_v} L_v$;
6: end for
```

The derived result F_r' and the correct result F_r are compared by two scoring methods to evaluate the performance of the model:

$$S_G(F_r', F_r) = \lambda_0 / (|G_s \in F_r \wedge G_s \notin F_r'| + |G_s \in F_r' \wedge G_s \notin F_r|)$$

$$S_E(X_r', X_r) = \lambda_1 / (||X_r' - X_r||)$$

where $S_G(F_r', F_r)$ measures the similarity of the graphs of F_r' and F_r, and $S_E(X_r', X_r)$ measures the similarity of the feature vectors. The models are trained by differential equation solving dataset dataset (DESD) containing more than 2000 differential equations and relevant solutions.

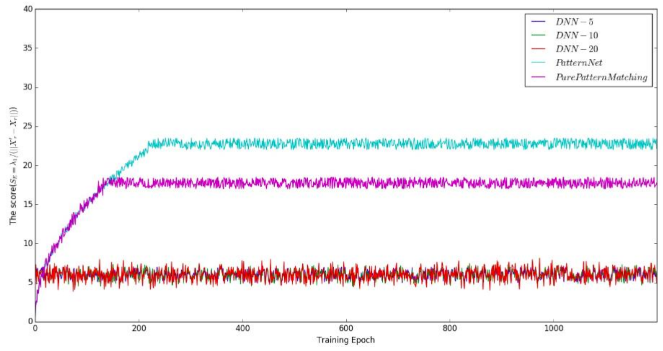

Figure 2: The score S_E of the proposed model and the baseline models.

Table 1: The Score Of The Proposed Model And Baselines.

Score	$S_G(F_r', F_r)$	$S_E(X_r', X_r)$
DNN@5	0.37	5.59
DNN@10	0.76	6.71
DNN@20	0.75	6.80
Pattern Match	4.17	17.56
PatternNet	**4.27**	**23.18**

3 RESULT

In this paper, we propose a method for automatic formula derivation with graph embedding and pattern matching network. Experimental results have showed that our approach combined with pattern matching and network fitting achieves best result.

REFERENCES

[1] Xiao Shan Gao and Ding Kang Wang. 1995. On the automatic derivation of a set of geometric formulae. *Journal of Geometry* 53, 1-2 (1995), 79–88.
[2] Paul Houston and Nathan Sime. 2018. Automatic symbolic computation for discontinuous Galerkin finite element methods. *Siam Journal on Scientific Computing* 40, 3 (2018).
[3] J. Robinson and Andrei Voronkov. 2001. *Handbook of automated reasoning.* Elsevier ;. 179272 pages.
[4] Jano Vidali. 2018. Using symbolic computation to prove nonexistence of distance-regular graphs. (2018).
[5] Mingzhe Wang, Yihe Tang, Jian Wang, and Jia Deng. 2017. Premise Selection for Theorem Proving by Deep Graph Embedding. (2017).
[6] Valery Yakhno and Meltem Altunkaynak. 2018. Symbolic computation of the timedependent electric and magnetic fields in bianisotropic media with polynomial inputs. *International Journal of Numerical Modelling Electronic Networks Devices and Fields* 2 (2018), e2339.

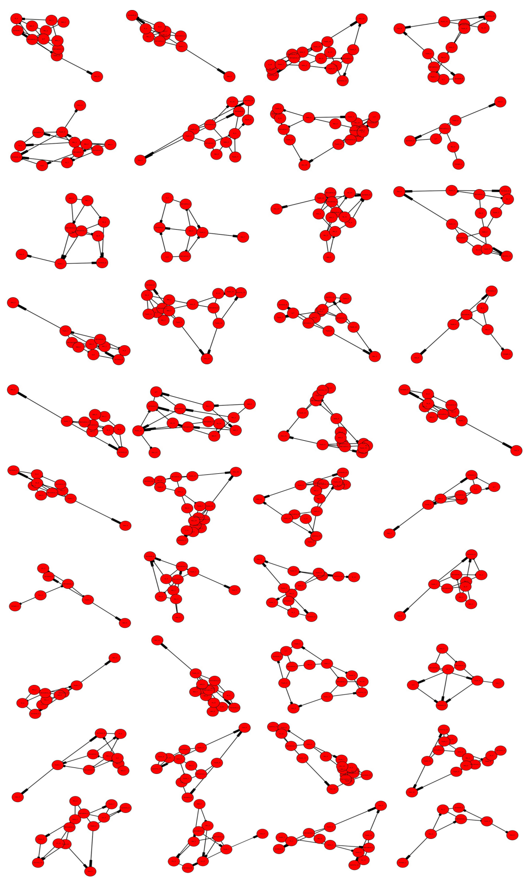

Figure 3: Some formula examples represented by directed graph in the dataset.

$$\bullet\text{equation}: aY_0(u)\frac{d^2}{dx^2}f(x)+uY_0(b)\frac{d}{dx}f(x)+yKr(a)f(x)=u^2g(x)$$

$$\diamond\text{solution}: f(x)=C_1e^{\frac{x\left(-uY_0(b)-\sqrt{-4ayKr(a)Y_0(u)+u^2{Y_0}^2(b)}\right)}{2aY_0(u)}}+C_2e^{\frac{x\left(-uY_0(b)+\sqrt{-4ayKr(a)Y_0(u)+u^2{Y_0}^2(b)}\right)}{2aY_0(u)}}$$
$$-\frac{u^2e^{\frac{x\left(-uY_0(b)-\sqrt{-4ayKr(a)Y_0(u)+u^2{Y_0}^2(b)}\right)}{2aY_0(u)}}\int g(x)e^{\frac{x\sqrt{-4ayKr(a)Y_0(u)+u^2{Y_0}^2(b)}}{2aY_0(u)}}e^{\frac{uxY_0(b)}{2aY_0(u)}}dx}{\sqrt{-4ayKr(a)Y_0(u)+u^2{Y_0}^2(b)}}$$
$$+\frac{u^2e^{\frac{x\left(-uY_0(b)+\sqrt{-4ayKr(a)Y_0(u)+u^2{Y_0}^2(b)}\right)}{2aY_0(u)}}\int g(x)e^{-\frac{x\sqrt{-4ayKr(a)Y_0(u)+u^2{Y_0}^2(b)}}{2aY_0(u)}}e^{\frac{uxY_0(b)}{2aY_0(u)}}dx}{\sqrt{-4ayKr(a)Y_0(u)+u^2{Y_0}^2(b)}}$$

$$\bullet\text{equation}: aPm(z)\frac{d^2}{dx^2}f(x)+yPm(z)\frac{d}{dx}f(x)+zPm(z)f(x)=Pr(x)$$

$$\diamond\text{solution}: f(x)=C_1e^{\frac{x\left(-y-\sqrt{-4az+y^2}\right)}{2a}}+C_2e^{\frac{x\left(-y+\sqrt{-4az+y^2}\right)}{2a}}$$
$$-\frac{e^{\frac{x\left(-y-\sqrt{-4az+y^2}\right)}{2a}}\int Pr(x)e^{\frac{xy}{2a}}e^{\frac{x\sqrt{-4az+y^2}}{2a}}dx}{\sqrt{-4az+y^2}Pm(z)}$$
$$+\frac{e^{\frac{x\left(-y+\sqrt{-4az+y^2}\right)}{2a}}\int Pr(x)e^{\frac{xy}{2a}}e^{-\frac{x\sqrt{-4az+y^2}}{2a}}dx}{\sqrt{-4az+y^2}Pm(z)}$$

$$\bullet\text{equation}: uPt(y)\frac{d}{dx}f(x)+yPm(a)\frac{d^2}{dx^2}f(x)+zLm(b)f(x)=k(x)$$

$$\diamond\text{solution}: f(x)=C_1e^{\frac{x\left(-uPt(y)-\sqrt{u^2Pt^2(y)-4yzLm(b)Pm(a)}\right)}{2yPm(a)}}+C_2e^{\frac{x\left(-uPt(y)+\sqrt{u^2Pt^2(y)-4yzLm(b)Pm(a)}\right)}{2yPm(a)}}$$
$$-\frac{e^{\frac{x\left(-uPt(y)-\sqrt{u^2Pt^2(y)-4yzLm(b)Pm(a)}\right)}{2yPm(a)}}\int k(x)e^{\frac{x\sqrt{u^2Pt^2(y)-4yzLm(b)Pm(a)}}{2yPm(a)}}e^{\frac{uxPt(y)}{2yPm(a)}}dx}{\sqrt{u^2Pt^2(y)-4yzLm(b)Pm(a)}}$$
$$+\frac{e^{\frac{x\left(-uPt(y)+\sqrt{u^2Pt^2(y)-4yzLm(b)Pm(a)}\right)}{2yPm(a)}}\int k(x)e^{-\frac{x\sqrt{u^2Pt^2(y)-4yzLm(b)Pm(a)}}{2yPm(a)}}e^{\frac{uxPt(y)}{2yPm(a)}}dx}{\sqrt{u^2Pt^2(y)-4yzLm(b)Pm(a)}}$$

$$\bullet\text{equation}: aPm(u)\frac{d}{dx}f(x)+bEv(y)\frac{d^2}{dx^2}f(x)+yPt(y)f(x)=yPr(x)$$

$$\diamond\text{solution}: f(x)=C_1e^{\frac{x\left(-aPm(u)-\sqrt{a^2Pm^2(u)-4byEv(y)Pt(y)}\right)}{2bEv(y)}}+C_2e^{\frac{x\left(-aPm(u)+\sqrt{a^2Pm^2(u)-4byEv(y)Pt(y)}\right)}{2bEv(y)}}$$
$$-\frac{ye^{\frac{x\left(-aPm(u)-\sqrt{a^2Pm^2(u)-4byEv(y)Pt(y)}\right)}{2bEv(y)}}\int Pr(x)e^{\frac{x\sqrt{a^2Pm^2(u)-4byEv(y)Pt(y)}}{2bEv(y)}}e^{\frac{axPm(u)}{2bEv(y)}}dx}{\sqrt{a^2Pm^2(u)-4byEv(y)Pt(y)}}$$
$$+\frac{ye^{\frac{x\left(-aPm(u)+\sqrt{a^2Pm^2(u)-4byEv(y)Pt(y)}\right)}{2bEv(y)}}\int Pr(x)e^{-\frac{x\sqrt{a^2Pm^2(u)-4byEv(y)Pt(y)}}{2bEv(y)}}e^{\frac{axPm(u)}{2bEv(y)}}dx}{\sqrt{a^2Pm^2(u)-4byEv(y)Pt(y)}}$$

$$\bullet\text{equation}: aQs(a)\frac{d^2}{dx^2}f(x)+bf(x)g(z)+yh(u)\frac{d}{dx}f(x)=Pt(x)$$

$$\diamond\text{solution}: f(x)=C_1e^{\frac{x\left(-yh(u)-\sqrt{-4abQs(a)g(z)+y^2h^2(u)}\right)}{2aQs(a)}}+C_2e^{\frac{x\left(-yh(u)+\sqrt{-4abQs(a)g(z)+y^2h^2(u)}\right)}{2aQs(a)}}$$
$$-\frac{e^{\frac{x\left(-yh(u)-\sqrt{-4abQs(a)g(z)+y^2h^2(u)}\right)}{2aQs(a)}}\int Pt(x)e^{\frac{x\sqrt{-4abQs(a)g(z)+y^2h^2(u)}}{2aQs(a)}}e^{\frac{xyh(u)}{2aQs(a)}}dx}{\sqrt{-4abQs(a)g(z)+y^2h^2(u)}}$$
$$+\frac{e^{\frac{x\left(-yh(u)+\sqrt{-4abQs(a)g(z)+y^2h^2(u)}\right)}{2aQs(a)}}\int Pt(x)e^{-\frac{x\sqrt{-4abQs(a)g(z)+y^2h^2(u)}}{2aQs(a)}}e^{\frac{xyh(u)}{2aQs(a)}}dx}{\sqrt{-4abQs(a)g(z)+y^2h^2(u)}}$$

$$\bullet\text{equation}: ah(y)\frac{d^2}{dx^2}f(x)+uPr(u)f(x)+un(b)\frac{d}{dx}f(x)=z^2n(x)$$

$$\diamond\text{solution}: f(x)=C_1e^{\frac{x\left(-un(b)-\sqrt{u\left(-4aPr(u)h(y)+un^2(b)\right)}\right)}{2ah(y)}}+C_2e^{\frac{x\left(-un(b)+\sqrt{u\left(-4aPr(u)h(y)+un^2(b)\right)}\right)}{2ah(y)}}$$
$$-\frac{z^2e^{\frac{x\left(-un(b)-\sqrt{u\left(-4aPr(u)h(y)+un^2(b)\right)}\right)}{2ah(y)}}\int n(x)e^{\frac{x\sqrt{u\left(-4aPr(u)h(y)+un^2(b)\right)}}{2ah(y)}}e^{\frac{uxn(b)}{2ah(y)}}dx}{\sqrt{u\left(-4aPr(u)h(y)+un^2(b)\right)}}$$
$$+\frac{z^2e^{\frac{x\left(-un(b)+\sqrt{u\left(-4aPr(u)h(y)+un^2(b)\right)}\right)}{2ah(y)}}\int n(x)e^{-\frac{x\sqrt{u\left(-4aPr(u)h(y)+un^2(b)\right)}}{2ah(y)}}e^{\frac{uxn(b)}{2ah(y)}}dx}{\sqrt{u\left(-4aPr(u)h(y)+un^2(b)\right)}}$$

Figure 4: Some formula examples in the dataset.