УДК 519.87

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ САМОНАСТРАИВАЮЩЕГОСЯ АЛГОРИТМА ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ С НАСТРАИВАЕМЫМИ КОЭФФИЦИЕНТАМИ ДЕРЕВЬЕВ ПРИ РЕШЕНИИ ЗАДАЧ СИМВОЛЬНОЙ РЕГРЕССИИ

Хритоненко Д.И.
Научный руководитель д-р техн. наук Семенкин Е.С.
Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева.

Задача *символьной регрессии* заключается в нахождении математического выражения в символьной форме, аппроксимирующего зависимость между конечным набором значений независимых переменных и соответствующими значениями зависимых переменных.

Эволюционные алгоритмы (ЭА) – стохастические алгоритмы, в которых искусственно воссозданы эволюционные процессы реального мира[1]. В ЭА решения определенным образом кодируются *индивидами,* а приспособленность каждого индивида измеряется некоторым вещественным числом, называемым его *пригодностью*. Так, например, при решении задачи символьной регрессии при помощи алгоритма генетического программирования (ГП)[2] каждый индивид будет представлен в виде *дерева* (направленного графа, в котором каждая последующая вершина связанна с одной и только одной предыдущей). Пример математического выражения $5 \cdot x \cdot \sin(y) + (y + 1) \cdot \cos(x)$, закодированного деревом:

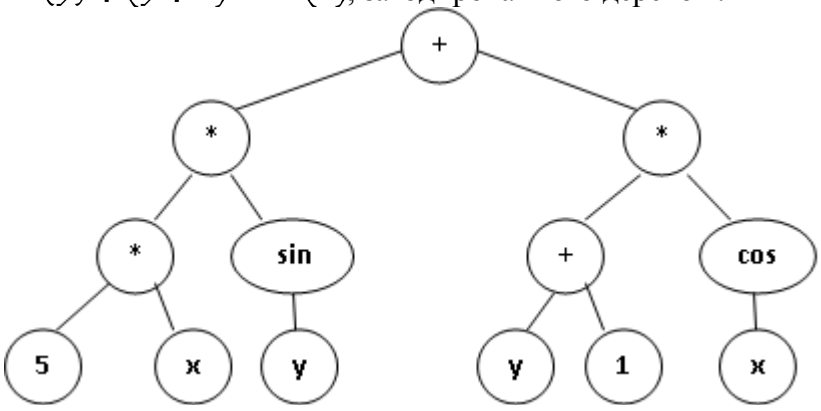

Рисунок 1. Пример кодирования

Важными являются понятия *функционального* и *терминального* множества. Под функциональным множеством понимается множество всех возможных внутренних вершин дерева. Под терминальным – множество всех возможных внешних вершин дерева. В задаче символьной регрессии терминальное множество определяется как множество констант и переменных-аргументов. Полезным является оценивание *оптимального* набора констант каждого дерева, т.к. в случае выбора «удачной» структуры (структуры, которая близка, совпадает с истинной или точно аппроксимирует истинную), мы можем существенно уменьшить ошибку аппроксимации. Пусть $Const_j$ – набор констант выражения, закодированного деревом T_j. Тогда оптимальный набор констант будет определяться следующим образом:

$$Const_j^{opt} = \arg \min_{Const \in R^k} \left(\sum_{i=1}^{n} [f(\overline{x_i}) - \widehat{f_j}(\overline{x_i}, Const_j)]^2 \right)$$

В данной формуле k – число констант дерева T_j, n – объем выборки, $\overline{x_i}$ – вектор переменных-входов, $\hat{f}_j(*)$ – вычисленное по дереву T_j значение выражения.

В данной работе для отыскания оптимального набора констант дерева используется самонастраивающийся генетический алгоритм (ГА), который был разработан и протестирован ранее. Идея самонастройки данного ГА была перенесена и на алгоритм ГП. Эффективность была также проверена на ряде тестовых задач, а также на задачах классификации австралийских и немецких кредитов. В задачах классификации алгоритм ГП аппроксимировал разделяющую поверхность между двумя различными классами. Тестировались различные варианты ГП:

Название	GP_1	GP_2	GP_3	GP_4
Особенности	Стандартный алгоритм ГП	GP_1 + настройка коэффициентов дерева	GP_1 + настройка параметров самого алгоритма ГП	GP_2+ GP_3

Таблица 1. Виды тестируемых алгоритмов ГП

Тестирование проводилось по 50-ти прогонам на 24-х тестовых задачах размерности 2 [3], а также на двух задачах классификации размерности 14 и 20 соответственно [4]. Результаты тестирования и сравнения алгоритмов на тестовых задачах:

Алгоритм	Ошибка	Сложность	Надежность	Трудоемкость	Формулы		
					P1	P2	P3
GP_1	0.00793	87	63	826	17	20	63
GP_2	0.00497	73	87	671	30	9	61
GP_3	0.00475	86	92	751	23	12	65
GP_4	0.00263	61	100	693	41	10	49

Таблица 2. Результаты тестирования

Ошибка – среднеквадратичное отклонение выхода полученного выражения и истинного значения. *Сложность* – число вершин дерева. *Надежность* – процент решений, удовлетворяющих заданной точности (ошибка<0.01). *Трудоемкость* – номер поколения, на котором было найдено решение, удовлетворяющее заданной точности. Формулы (решения), аппроксимирующие с заданной точностью тестовые задачи, в свою очередь были разделены на 3 класса:

- *Точные* (P1) – формулы, точно совпадающие с истинной структурой
- *Условно точные* (P2) – формулы, которые приводятся при помощи элементарных математических преобразований и округлений к истинным
- *Не приводимые* (P3) – формулы, не приводимые к истинным (разложение в ряд и т.д.)

В данной таблице приведены результаты усреднения по всем тестовым задачам и прогонам. В случае тестирования алгоритмов ГП без настройки параметров, проводилось также усреднение и по всем параметрам алгоритма.

Результаты тестирования и сравнения алгоритмов на задачах классификации:

Название алгоритма	Australian credit	German credit
SCGP	0.9022	0.7950
MGP	0.8985	0.7875
2SGP	**0.9027**	**0.8015**
GP	0.8889	0.7834
Fuzzy classifier	0,8910	0,7940

C4.5	0.8986	0.7773
LR	0.8696	0.7837
Bayesian approach	0,8470	0,6790
Boosting	0,7600	0,7000
Bagging	0,8470	0,6840
RSM	0,8520	0,6770
CCEL	0,8660	0,7460
k-NN	0.7150	0.7151
CART	0.8744	0.7565
MLP	0.8986	0.7618
GP_4	0.8960	0.7693

Таблица 3. Результаты тестирования

Критерий эффективности алгоритма:

$$I = 1 - \frac{ERR}{n}$$

В данной формуле ERR – число неправильно классифицированных объектов, n – общее число объектов. Сравнительные результаты взяты из научной литературы.

Сравнивая полученный алгоритм (GP_4) с базовым (GP_1), можно сделать вывод, что применение самонастраивающегося алгоритма ГП с настраиваемыми коэффициентами деревьев позволило существенно уменьшить ошибку аппроксимации, уменьшить сложность выражения, а также повысило надежность. Как недостаток, следует отметить существенное увеличение времени работы. Это связано с тем, что на оптимизацию коэффициентов дерева тратится большое количество ресурсов. В связи с этим, полезным может являться использование алгоритма GP_3. На задачах классификации разработанный алгоритм с настраивающимися коэффициентами деревьев показал результаты, сравнимые с существующими аналогами.

Библиографические ссылки

1. *Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы : пер. с польск. И. Д. Рудинского. М. : Горячая линия. Телеком, 2006. – 383 с.*
2. *Koza J. R. Genetic programming: on the programming of computers by means of natural selection. : MIT, 1998. – 609 c.*
3. *[Электронный ресурс] – URL: http://coco.gforge.inria.fr/doku.php*
4. *[Электронный ресурс] – URL: http://archive.ics.uci.edu/ml/datasets.html*