

Методы структурного обучения для построения прогностических моделей*

А. А. Варфоломеева

Московский физико-технический институт, ФУПМ, каф. «Интеллектуальные системы»

В работе исследуется задача прогноза существенно нелинейной модели методами структурной регрессии. Предлагается алгоритм, определяющий структуру оптимальной параметрической модели заданной сложности. Приводится сравнение качества работы предлагаемого алгоритма и алгоритма символьной регрессии на синтетических данных.

Ключевые слова: *структурное обучение, нелинейные модели, индуктивное порождение, сложность моделей, символьная регрессия.*

Введение

В работе предложен и изучен метод прогнозирования структуры нелинейной модели при помощи структурного обучения. При этом структура модели представляет из себя суперпозицию из набора заранее определенных элементарных функций.

Одним из методов для решения задачи восстановления функциональной зависимости по набору исходных данных является символьная регрессия. Джон Коза предложил реализацию этого метода с помощью аналога эволюционного алгоритма [1]. Иван Зелинка предложил дальнейшее развитие этой идеи [?], получившее название аналитического программирования.

Алгоритм построения математической модели в аналитическом программировании выглядит следующим образом: задан набор элементарных функций, из которых можно строить различные формулы (например, степенная функция, $+$, \sin , \tan). Начальный набор формул строится либо произволь-

Работа выполнена при поддержке Министерства образования и науки РФ в рамках Государственного контракта 07.524.11.4002.

ным образом, либо на базе некоторых предположений эксперта. Затем на каждом шаге производится оценка каждой из формул согласно некоторому функционалу качества. На базе этой оценки у части формул случайным образом заменяется одна элементарная функция на другую (например, \sin на \cos или $+$ на \times), а у некоторой другой части происходит взаимный попарный обмен подвыражениями. Данный подход может быть описан в терминах эволюционного алгоритма: каждый индивид является формулой, изображенной в свою очередь в виде дерева. Тогда набор формул, существующий в определенный момент, представляет собой одно поколение. При этом хромосомы представляются поддеревьями, и, в отличие от классического генетического алгоритма, могут быть различного размера (длины). Описанный выше обмен подвыражениями представляет собой в этом случае генетическое скрещивание, замена одной элементарной функции у некоторых деревьев — мутацию. При этом возникает ряд сложностей, связанных с областями определения и арностями элементарных функций, записанных в узлах дерева. Данный метод фактически является ненаправленным поиском и перебирает большое количество неподходящих деревьев до того момента, как приблизится к оптимуму.

Альтернативой аналитическому программированию можно считать подход обучения в глубину (Deep Learning) [2, 3]. Этот подход заключается в иерархическом представлении данных, в котором на нижнем уровне находятся сам набор данных, а на каждом уровне выше — более абстрактное его представление, которое представляет собой некую скрытую комбинация из данных, указанных ниже. Так, например, при использовании данного метода в обработке изображений, набором данных является матрица яркости пик-

селей некоего изображения, на следующем уровне — данные о выраженных геометрических закономерностях на изображении (отрезки, кривые, окружности), на более высоких уровнях иерархии — более сложные и абстрактные выявленные закономерности. В одном из основных алгоритмов, использующих данный подход, иерархия строится при помощи нейронной сети с несколькими скрытыми слоями [4]. В одном из основных методов обучения в глубину нейронная сеть обучается, получая на вход и на выход одинаковый набор данных, после чего каждый из уровней сети представляется как вид задания данных на определенном уровне абстракции.

В данной работе предлагается рассмотреть метод построения математической модели, основанный на прогнозировании структуры функциональной зависимости. Предполагается, что функциональная зависимость существенно нелинейна и, аналогично описанному выше, является суперпозицией элементарных функций. При этом делается ограничение на максимальную сложность модели, и дерево суперпозиции представляется в виде матрицы. В таком виде задача сводится к задаче структурного обучения, описанной, например, в [5, 6]. Методы структурного обучения решают задачу нахождения структуры или зависимостей, имеющихся внутри исходных данных. Метод широко применим для синтаксического разбора предложений [?], компьютерного зрения [?], структуризации неформатированных библиографических записей [?].

Цель предлагаемой работы заключается в разработке такого алгоритма, который бы прогнозировал структуру регрессионной модели, описывающую предъявленную выборку оптимальным образом. Параметры алгоритма оцениваются по набору, в котором содержатся прецеденты — множество регрес-

сионных выборок и множество моделей, оптимально соответствующих этим выборкам.

Решаются следующие задачи:

1. выбор способа задания структуры регрессионной модели;
2. выполнение заданных ограничений на структуры модели;
3. выбор функционала ошибки алгоритма;
4. выбор прогностического алгоритма.

Работа предложенного метода иллюстрируется на данных..(каких?)

Постановка задачи

Задан набор $\mathfrak{D} = \{(\mathbf{D}_k, f_k)\}$, состоящий из регрессионных выборок \mathbf{D} . Каждая пара $\mathbf{D}_k = (\mathbf{X}_k, \mathbf{y}_k)$ состоит из $(m \times n)$ -матрицы \mathbf{X} и $(m \times 1)$ -вектора \mathbf{y} .

Каждой выборке \mathbf{D} требуется поставить в соответствие оптимальную модель $f : \mathfrak{Z} \times \mathbb{X} \rightarrow \mathbb{R}$ из порождаемого множества моделей $\mathcal{F} = \{f_s\}$, где \mathfrak{Z} — пространство параметров, доставляющую минимум заданной функции ошибки, определяемой ниже.

Другими словами, для множества моделей

$$\mathcal{F} = \{f_s \mid f_s : (\omega, \mathbf{X}) \mapsto \mathbf{y}, s \in \mathbb{N}\},$$

требуется найти такой индекс \hat{s} , что функция f_s среди всех $f \in \mathcal{F}$ доставляет минимум функции ошибки S при фиксированной регрессионной выборке \mathbf{D} :

$$\hat{s} = \arg \min_{s \in \mathbb{K}} S(f_s \mid \hat{\mathbf{w}}_k, \mathbf{D}_k), \quad (1)$$

где $\hat{\mathbf{w}}_k$ — оптимальный вектор параметров модели f_s для каждой $f \in \mathcal{F}$ при данной регрессионной выборке \mathbf{D} :

$$\hat{\mathbf{w}}_k = \arg \min_{\mathbf{w} \in \mathfrak{Z}} S(\omega \mid f_s, \mathbf{D}_k). \quad (2)$$

В качестве функции ошибки S используется сумма квадратов регрессионных остатков:

$$S(\mathbf{w}_k \mid f_s, \mathbf{D}_k) = \| \mathbf{y} - f(\mathbf{w}_k, \mathbf{X}) \|_2. \quad (3)$$

Множество \mathcal{F} порождается следующим образом. Задано множество \mathcal{G} порождающих функций. Для каждой функции $g : \mathbb{R} \times \dots \times \mathbb{R} \rightarrow \mathbb{R}$ из набора \mathcal{G} определены её арифность $v = v(g)$, области определения и значений: $\text{dom}(g), \text{cod}(g)$. Заданы правила индуктивного порождения функции f в виде суперпозиции функций из \mathcal{G} .

Каждой суперпозиции f ставится в соответствие дерево Γ_f , строящееся по следующим правилам:

- корнем дерева является специальный символ “ $*$ ”, имеющий одну дочернюю вершину;
- в остальных вершинах V_i дерева Γ_f находятся соответствующие элементарные функции $g_r, r = r(i)$ из набора \mathcal{G} ;
- число дочерних вершин V_j у некоторой вершины V_i равно арифности соответствующей функции $g_r: v = v(g_r)$;
- область определения элементарной функции дочерней вершины V_j содержит область значений функции родительской вершины $V_i: \text{dom}(g_{r(i)}) \supset \text{cod}(g_{r(j)})$;
- в листьях дерева Γ_f находятся свободные переменные x_i .

!!тут пример дерева(?inkscape)

Каждому дереву Γ_f ставится в соответствие бинарная матрица Z размера $(l + n) \times (m + 1)$, где l – число элементарных функций набора \mathcal{G} , n – число свободных переменных \mathbf{x}_i . Элементы матрицы Z отвечают за наличие ребра между двумя вершинами в дереве. При этом столбцы матрицы отвечают

только за те вершины, которые могут быть родительскими: вершина дерева “ * ” и элементарные функции g_s . Строки матрицы отвечают за потенциальные дочерние вершины: элементарные функции g_r и свободные переменные x_i . Таким образом, матрица Z состоит из квадратного блока, отвечающего за набор элементарных функций, добавленного слева столбца, отвечающего за вершину дерева “ * ”, и n добавленных снизу строк, отвечающих за свободные переменные x_i . Таким образом, на матрицу Z по построению накладываются следующие ограничения:

- в каждом столбце i содержится либо количество единиц, равное арности $v = v(g_{r(i)})$ элементарной функции $g_{r(i)}$, отвечающей за i -ый столбец матрицы, либо ноль;
- в каждой строке может содержаться только одна единица;
- заполнение строк и столбцов проходит сверху–вниз и слева–направо, т.е. для записи очередного ребра в матрицу выбирается самый левый и верхний из “свободных-столбцов и строк, отвечающий тем же родительским и дочерним элементам.

Требуется найти такой алгоритм $a : \mathbb{D}_k \mapsto f_s$, который ставит в соответствие каждому набору \mathbb{D}_k модель f_s , которая описывает этот набор оптимальным в смысле $(1, 2, 3)$ образом.

Описание алгоритма

В терминах структурного обучения найти модель, оптимально описывающую регрессионную выборку \mathbb{D}_s , значит указать соответствующую ей матрицу Z_s , по которой однозначно восстанавливается структура дерева суперпозиции Γ_f .

(Полет мысли) Данная матрица Z_s не может быть спрогнозирована в явном виде. Но могут быть спрогнозированы матрица вероятностей переходов, т.е. известна вероятность найти некоторое ребро в дереве суперпозиции f_s . Для нахождения матрицы вероятности P используется нейронная сеть, получающая на вход регрессионную выборку D_s , и имеющая сигмоидную функцию на выходном слое. Ставится задача построения оптимального дерева суперпозиции Γ_{f_s} по данной матрице вероятностей P . В работе предложено два метода решения данной задачи: ()

Жадный алгоритм построения дерева. Задан взвешенный ориентированный граф $\Gamma = \mathcal{G} \times E$, где \mathcal{G} — множество порождающих функций $\mathcal{G} = \{g_1, \dots, g_n\}$. Графу соответствует матрица переходных вероятностей P_1 ,

$$(P_1)_{ij} = p(g_i \rightarrow g_j).$$

Задано множество свободных переменных

$$X = \{x_1, \dots, x_n\}$$

и матрица переходных вероятностей P_2 размера $m \times n$

$$(P_2)_{ik} = p(g_i \rightarrow x_k).$$

Заданы арности порождающих функций из набора \mathcal{G} . Будем обозначать

$$D = \mathcal{G}' \times E', \quad |E'| = K,$$

поддерево графа Γ сложности K .

Предлагается следующий жадный алгоритм. Пусть на шаге $k - 1$ построено дерево D_{k-1} с «открытыми» вершинами $g_1, \dots, g_{r_{k-1}}$. Для каждой из этих открытых вершин подсчитаем оценки:

$$c_i = \frac{\max_{j=1, \dots, m} (P_1)_{ij}}{\max_{k=1, \dots, n} (P_2)_{ik}}, \quad i = 1, \dots, r_{k-1}.$$

Пока количество ребер в дереве не превышает наперед заданной сложности K : $D_k \leq K$,

«закрываем» ребро $i^* = \arg \max_i c_i$ наибольшей оценкой c_i функцией g_{ij} :
 $j^* = \arg \max_j g_{ij}$.

Если количество ребер в дереве превышает K , «закрываем» все открытые ребра переменной x_{k^*} , $k^* = \arg \max_k (P_2)_{ik}$ для всех $i = 1, \dots, r$.

Вычислительный эксперимент

Алгоритм был протестирован на выборке синтетических данных, полученных следующим образом. Экспертно задан набор элементарных функций \mathcal{G} , для каждой из которых известны её аргументность $v = v(g)$, области определения и значений: $dom(g), cod(g)$. По набору \mathcal{G} по указанным выше правилам было построено конечное множество суперпозиций \mathcal{F} — библиотека функций. Экспертно заданы значения независимых переменных \mathbf{X} , для каждого элемента f_s из библиотеки функций \mathcal{F} получены значения зависимых переменных \mathbf{y}_s . Таким образом, сгенерировано множество регрессионных выборок $\mathfrak{D} = \{(\mathbf{D}_s, f_s)\}$, где $\mathbf{D}_s = (\mathbf{X}, \mathbf{y})$. Также указаны максимальные числа вхождений элементарных функций k и переменных p в суперпозицию.

Для обучения алгоритма была использована стандартная нейронная сеть в среде Matlab с двумя скрытыми слоями, имеющая на последнем слое сигмоидную функцию. Входом такой нейросети является регрессионная выборка $\mathbf{D}_s = (\mathbf{X}, \mathbf{y})$, выходом — матрица вероятностей P .

Литература

- [1] *Koza, J. R.* Genetic programming. In Williams, J. G. and Kent, A. (editors) // Encyclopedia of Computer Science and Technology, 1998. Vol. 39. P.: 29-43.
- [2] *Yoshua Bengio* Learning Deep Architectures for AI // Foundations and Trends in Machine Learning, 2009. Vol: 2, No. 1.P.: 1–127.
- [3] *Itamar Arel, Derek C. Rose, Thomas P. Karnowski* Deep Machine Learning—A New Frontier in Artificial Intelligence Research // IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE, November 2010. P. 13-19.
- [4] *Yoshua Bengio, Aaron Courville, Pascal Vincent* Representation Learning: A Review and New Perspectives // Department of computer science and operations research, U. Montreal.
- [5] *Martins A. F. T.* The Geometry of Constrained Structured Prediction: Applications to Inference and Learning of Natural Language Syntax. Carnegie Mellon University, 2012.
- [6] *Jaakola T., Sontag D.* Learning Bayesian Network Structure using LP Relaxations // Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), 2010. Vol: 9. Issue: 1. P.: 358–365.
- [7] *Г.И. Рудой, В.В. Стрижов* Алгоритмы индуктивного порождения суперпозиций для аппроксимации измеряемых данных // Информатика и её применения, 2013. Vol: 1. P. ??-??.
- [8] *Г.И. Рудой, В.В. Стрижов* Упрощение суперпозиций элементарных функций при помощи преобразований графов по правилам // Интеллектуализация обработки информации. Доклады 9-й международной конференции, 2012. P. 140-143.
- [9] *А.Н. Фирстенко* Метаописание временных рядов.
- [10] *Стрижов В.В., Крымова Е.А.* Выбор моделей в линейном регрессионном анализе // Информационные технологии, 2011. Вып. 10. Стр. 21–26.
- [11] *Lampert C. H.* Maximum Margin Multi-Label Structured Prediction. IST Austria (Institute of Science and Technology Austria), 2011.
- [12] *Bishop C.M.* Pattern Recognition and Machine Learning. LLC: Springer Science, 2006.
- [13] *Kwok T.-Y., Yeung D.-Y.* Constructive Algorithms for Structure Learning in Feedforward Neural Networks for Regression Problems // IEEE Transactions on Neural Networks, 1997. Vol. 8. P. 630–645.
- [14] *Riccardo Poli, William B. Langdon, Nicholas F. McPhee* A Field Guide to Genetic Programming, 2008.

-
- [15] *Vladislavleva, Ekaterina J.* Order of Nonlinearity as a Complexity Measure for Models Generated by Symbolic Regression via Pareto Genetic Programming.