Структурное обучение для генерации моделей

Бочкарев Артем Максимович
Научный руководитель: Стрижов В. В.

Московский физико-технический институт
Вычислительный центр им. А.А. Дородницына Российской академии наук

6 июня 2018 г.

Введение

Проблема

- Генетический алгоритм символьной регрессии находит точные модели аппроксимации, но требует значительных вычислительных ресурсов
- При аппроксимации выборки не учитываются модели, полученные на похожих задачах

Цель работы

- Автоматизировать построние моделей аппроксимации
- Ускорить нахождение моделей символьной регрессии

Методы

- Мета-обучение
- Прогнозирование структуры модели в виде дерева

Литература

- Символьная регрессия
  - Kulunchakov, A. S., & Strijov, V. V. (2017). Generation of simple structured information retrieval functions by genetic algorithm without stagnation. *Expert Systems with Applications, 85, 221-230.*
  - Koza, J. R. (1994). Genetic programming as a means for programming computers by natural selection. *Statistics and computing, 4(2), 87-112.*
- Мета-обучение
  - Zoph, B., & Le, Q. V. (2016). Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578.*
  - Lemke, C., Budka, M., & Gabrys, B. (2015). Metalearning: a survey of trends and technologies. *Artificial intelligence review, 44(1), 117-130.*
- Прогнозирование структуры деревьев
  - Alvarez-Melis, D., & Jaakkola, T. S. (2016). Tree-structured decoding with doubly-recurrent neural networks.
  - Jin, W., Barzilay, R., & Jaakkola, T. (2018). Junction Tree Variational Autoencoder for Molecular Graph Generation. *arXiv preprint arXiv:1802.04364.*

Постановка задачи

Задача аппроксимации

Пусть $\mathbf{X}$ – матрица объект-признак, а $\mathbf{y}$ – вектор зависимой переменной.

Задача аппроксимации должна удовлетворять следующим условиям.

Требования к задаче аппроксимации

- $\mathbf{x}_i$ неслучайны
- $\{\mathbf{x}_i\}_{i=1}^n$ – упорядоченное множество
- $\mathbf{y}$ случайны
- $y_i = f(\mathbf{x}_i) + \varepsilon_i$
  - ε_i независимы
  - ε_i гомоскедастичны
  - $\varepsilon_i \sim \mathcal{N}(0, \sigma)$

Описание задачи аппроксимации – выборка $D = (\mathbf{X}, \mathbf{y})$.

Постановка задачи

В качестве моделей для задачи аппрокисмации рассматривается пространство $\mathfrak{F}$ моделей символьной регрессии.

Модель символьной регрессии

- Порождается грамматикой G:

$$g \to B(g, g) | U(g) | S,$$

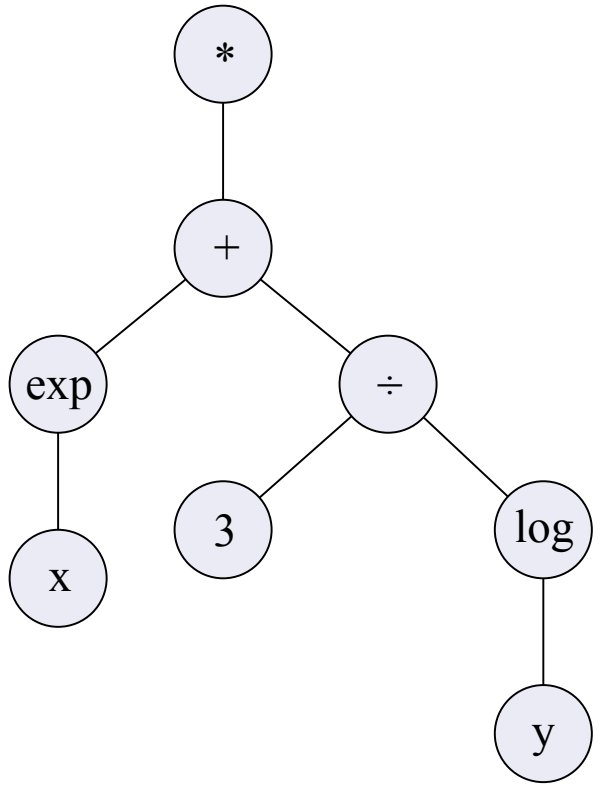
  где B – бинарные функции $(+, *)$, U – унарные функции $(\text{sqrt}, \log, \exp)$, а S – множество переменных.
- $f = g_1 \circ g_2 \circ \cdots \circ g_k$
- Модель символьной регрессии f представляется в виде дерева Γ_f

Постановка задачи

Дерево Γ_f удовлетворяет следующим условиям:

Дерево Γ_f

1. Символ $*$ является корнем дерева;
2. Листья Γ_f содержат переменные $x \in S$.
3. Узлы дерева v содержат соответствующие функции g;
4. $\#\text{child}(v) = \text{arity}(g)$;
5. Если v_j дочерняя вершина к v_j, то $\text{dom}(g_j) \supset \text{cod}(g_i)$;
6. Дочерние вершины g упорядочены;



$$f = e^x + \frac{3}{\log(y)}$$

Постановка задачи

Выборка мета-обучения

Набор пар $\mathfrak{D} = \{D_i = (\mathbf{X}_i, \mathbf{y}_i), f_i\}_{i=1}^m$ назовем мета-выборкой. Она удовлетворяет условиям:

- $\mathrm{dom}(\mathbf{x}_i) = \mathrm{dom}(\mathbf{x}_j)\ \forall i, j$ (все $\mathbf{X}$ имеют одну область определения)
- f_i является оптимальной моделью для D_i в пространстве $\mathfrak{F}$:

$$f_i = \arg\min_{f \in \mathfrak{F}} \mathrm{MSE}(\mathbf{y}_i, f_i(\mathbf{X}_i))$$

Задача мета-обучения

Для мета-выборки $\mathfrak{D}$ найти оптимальную мета-модель $\mathfrak{g} : D \to f$, минимизирующую ошибку на всех задачах аппроксимации:

$$\mathcal{L}(\mathfrak{g}, \mathfrak{D}) = \frac{1}{m} \sum_{i=1}^{m} MSE(\mathbf{y}_i, \mathfrak{g}(D_i)(\mathbf{X}_i))$$

Решение

Представление модели

Матрица смежности $\mathbf{Z}_f$ для дерева Γ_f.

Представление задачи аппроксимации

Вектор $\mathbf{d} = [\text{vec}(\mathbf{X}), \mathbf{y}]^T$ является представлением выборки для задачи аппроксимации.

Декомпозиция мета-модели

Мета-модель $\mathfrak{g} : D \to f$ является отображением из пространства векторов $\mathbb{R}^n$ в пространство матриц смежности деревьев $\mathbb{Z}$.
Мета-модель $\mathfrak{g}$ является суперпозицией двух функций:

$$f = \mathfrak{g}(D) = g_{\text{rec}}(g_{\text{clf}}(D))$$

Решение

Классификация

Функция классификации $g_{\text{clf}} : \mathbb{R}^n \to \mathbb{P}$ является отображением из пространства представлений задачи аппроксимации в пространство матриц вероятностей.

$$g_{\text{clf}}(\mathbf{d}) = \mathbf{P}_f,$$

где $\mathbf{P}_f$ – матрица вероятностей ребер в дереве Γ_f. g_{clf} – алгоритм классификации (логистическая регрессия, нейронная сеть).

Восстановление структуры

Функция восстановления структуры $g_{\text{rec}} : \mathbb{P} \to \mathbb{Z}$ это отображение из пространства матриц вероятностей ребер в пространство матриц смежности дерева. Предлагается два метода восстановления структуры дерева g_{rec}:

- Жадный алгоритм
- Динамическое программирование

Восстановление структуры

Жадный алгоритм

Алгоритм начинает восстановление из корня $*$. На каждом шаге достраивается ребро с наибольшей вероятностью. Конец работы, если достигли максимальной глубины дерева или в листьях только переменные.

Динамическое программирование

На каждом шаге задача разбивается на подзадачи, соответствующие возможным поддеревьям, для каждой ищется решение, максимизирующее $s(f)$.

- $s(f) = \prod_{e \in f} P_e$ – правдоподобие дерева;
- $s(f) = \frac{1}{n} \sum_{e \in f} P_e$ – средняя вероятность ребра в дереве.

Параметризация

Чтобы метод работал на реальных данных, необходима параметризация. Пусть лучшая непараметрическая модель представима в суперпозиции $f = f_1 \circ \cdots \circ f_n$.

Параметризация

Введем параметры для каждой элементарной функции f_i:

$$f_i(\mathbf{x}, \alpha_{i1}, \alpha_{i0}) = \alpha_{i1} f_i(\mathbf{x}) + \alpha_{i0}.$$

Параметрами суперпозиция f являются параметры ее элементарных функций:

$$f(\mathbf{x}) \to f(\mathbf{x}, \boldsymbol{\alpha})$$

Полученная функция дифференцируема, вектор параметров $\boldsymbol{\alpha}$ находится градиентным спуском.

Обзор метода

Обучение

1. Удалить константы из моделей f
2. Обучить g_{clf} на предсказание матрицы вероятностей $\mathbf{P}$

Тестирование

1. Предсказать матрицу $\mathbf{P}$ используя g_{clf}
2. Восстановить модель f с помощью g_{rec}
3. Параметризовать модель f
4. Найти оптимальные параметры α используя метод градиентного спуска

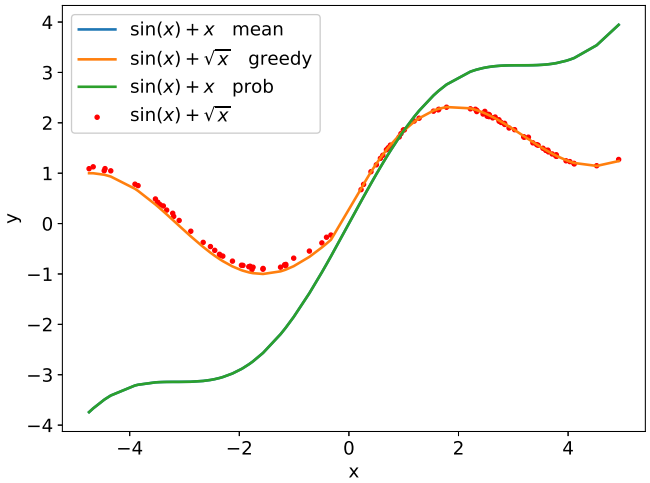
Вычислительный эксперимент

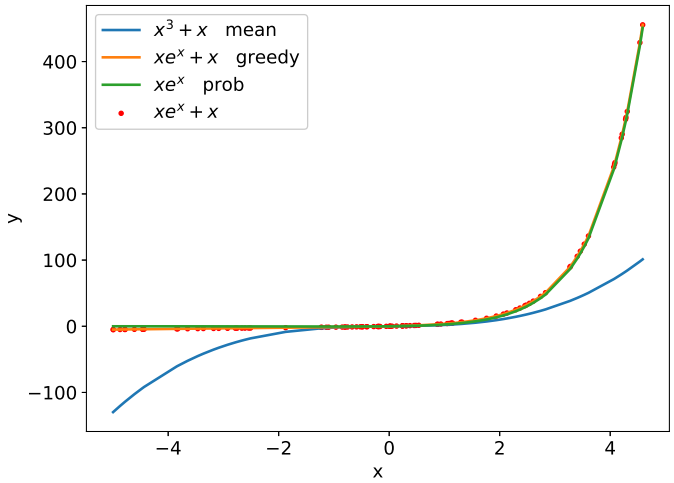
Сегменты сгенерированного временного ряда, порожденные случайными моделями символьной регрессии.

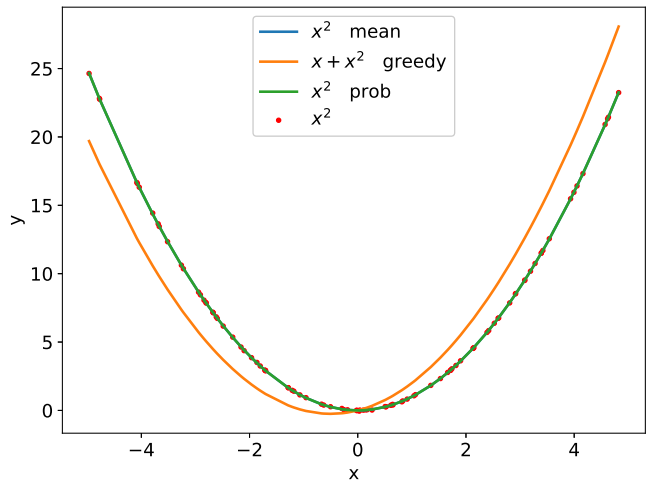
Схема эксперимента

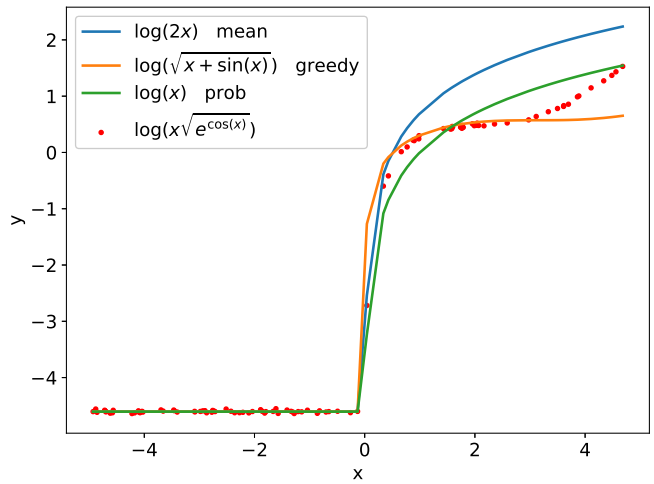
1. Сгенерировать ≈ 5000 1-D задач аппроксимации
   - $\mathbf{X}$ равномерно распределен на $[-5, 5]$
   - f – случайная модель символьной регрессии
   - $\mathbf{y} = f(\mathbf{X}) + \mathcal{N}(0, 0.05)$
2. Разделить задачи аппроксимации на обучение и контроль
3. Обучить g_{clf}
4. Предсказать матрицы вероятнстей $\mathbf{P}$ для тестовых задач и восстановить модели с помощью g_{rec}

Непараметрический случай

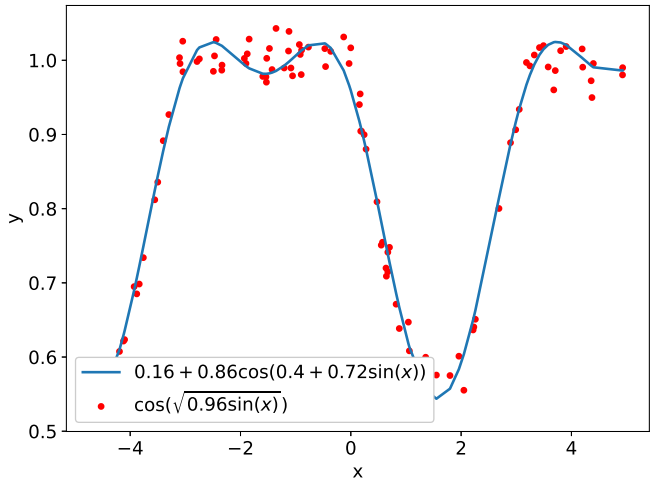


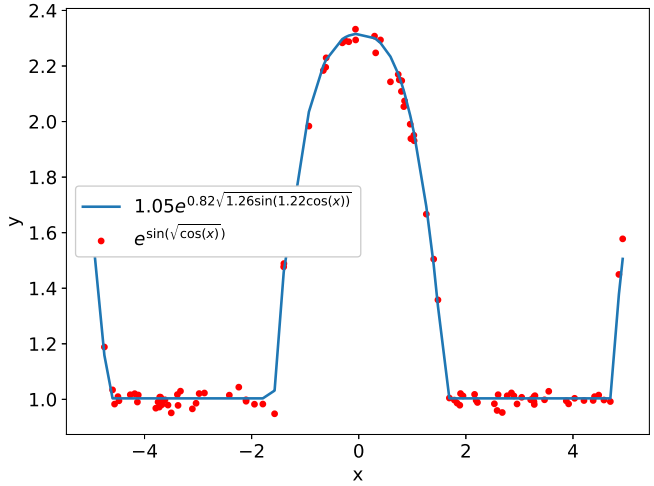


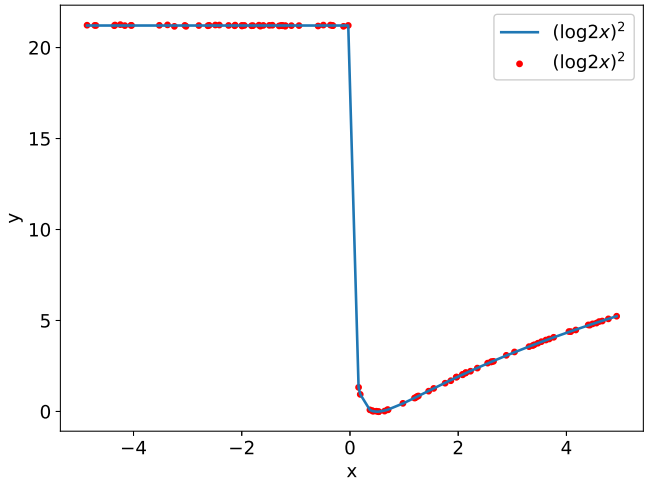


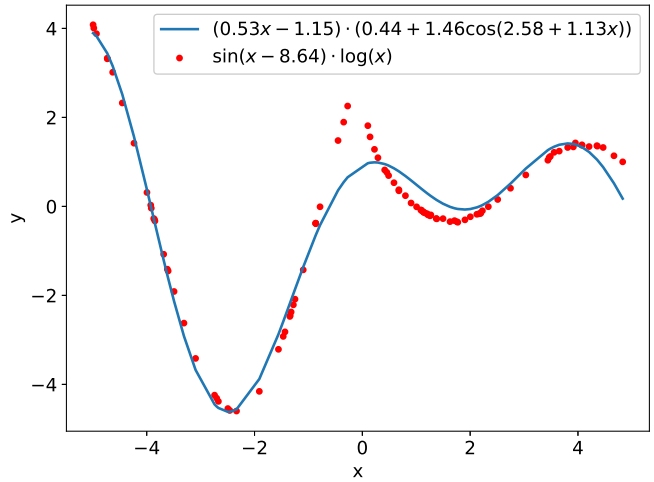


Параметрический случай









Результаты

Непараметрический случай

	Random Forest	*NN*	*Logistic regression*
Greedy algorithmr	5.45	5.81	6.3
DP (tree likelihood)	5.41	5.65	5.97
DP (mean probability)	**5.32**	5.72	6.12

Параметрический случай.

	Random Forest	*NN*	*Logistic regression*
Greedy algorithm	7.02	7.13	7.35
DP (tree likelihood)	**6.88**	6.93	7.01
DP (mean probability)	6.92	6.94	6.99

Реальные данные

Вычислительный эксперимент проводился на трех выборках: временные ряды с акселерометра, курса валют и стоимости акций на бирже. Сравнивались результаты и время работы предложенного метода и генетического алгоритма.

Результаты на реальных данных

	Акселерометер		Курс валют		Стоимость акций	
	MSE	t, сек	MSE	t, сек	MSE	t, сек
Символьная регрессия	0.052	5.12	0.012	6.02	3.13	6.34
Мета-модель	0.054	0.23	0.014	0.28	3.28	0.31

Заключение

- Предложен метод предсказания структуры дерева модели символьной регрессии
- Метод мета-обучения используется для прогнозирования оптимальных моделей аппроксимации
- Вычислительный эксперимент на реальных данных показал значительный прирост скорости в сравнении с генетическим алгоритмом