

# Получение распределений в задачах регрессии

Валерий В. Дмитриев\*  
Уфа, Россия  
ufabiz@gmail.com

## Аннотация

Приводится алгоритм нахождения функций распределения в качестве решения задачи регрессии.

В общем виде задачу регрессии можно сформулировать как восстановление зависимости  $\phi : X \rightarrow L_1(\Omega)$ , сопоставляющей элементам некоторого фазового пространства  $X$  случайную величину  $\xi \in L_1(\Omega)$ .

Классический подход к решению задачи регрессии состоит в нахождении среднего значения  $E[\phi(x)]$  для каждого  $x \in X$ .

В статье предлагается простой алгоритм оценки распределений случайных величин  $\phi(x) \in L_1(\Omega)$ .

## 1 Мотивация

В анализе данных значительное место занимают два класса задач – задачи классификации и регрессии.

Так сложилось, что, хотя эти задачи очень похожи, подход к их решению отличается.

Большинство алгоритмов решения задач *классификации* позволяют не просто оценить среднее значение  $E[\phi(x)]$ , для каждого элемента фазового пространства  $X$ , но и найти плотность распределения.

Для задач *регрессии* обычно находят лишь некоторую числовую оценку  $\widehat{\phi(x)}$ , которая, чаще всего, является средним значением, но не находят плотность распределения.

Знание плотности распределения дает гораздо больше возможностей для принятия решений, чем просто оценка среднего.

Например, для заданной точки  $x \in X$  мы можем:

- Оценить уверенность прогноза в каждой конкретной точке.
- Найти не среднее, а наиболее вероятное значение случайной величины. Это особенно актуально, если распределение  $\phi(x)$  является мультимодальным.
- Определить доверительный интервал возможных значений оценки  $\widehat{\phi(x)}$ .
- Вычислить любые характеристики распределения, определяемые конкретной задачей и позволяющие более взвешенно и точно принимать решения на основе прогноза модели.

## 2 Описание подхода

### 2.1 Постановка задачи

В целях простоты изложения опишу подход для одномерной задачи регрессии. В многомерном случае подход аналогичен.

---

\*<https://github.com/valmat>

Имеем некоторое фазовое пространство  $X$  и закономерность

$$\phi : X \rightarrow L_1(\Omega, \mathbb{R}).$$

$\phi$  сопоставляет случайные величины из  $L_1(\Omega)$  точкам фазового пространства  $X$ .

Таким образом, мы имеем семейство вероятностных мер  $\{P_x\}_{x \in X}$ , порождаемых закономерностью  $\phi$ .

Нам нужно построить модель, порождающую параметрическое семейство вероятностных мер  $\{Q_{x\theta}\}_{x \in X, \theta \in \Theta}$ , и найти оптимальное значение параметра  $\theta_0 \in \Theta$ , дающее наилучшее, в некотором смысле, приближение реальных распределений  $\{P_x\}_{x \in X}$ :

$$Q_{x\theta_0} \sim P_x.$$

При этом мы располагаем выборкой точек  $\{(x_i, y_i)\}_{i=1}^N$ , порожденной  $N$  независимыми испытаниями:  $x_i \in X$ ,  $y_i = \phi(x_i)$ .

$y_i \in L_1(\Omega)$  – являются независимыми случайными величинами.  $x_i \in X$ , в общем случае, случайными величинами могут и не быть.

Чтобы понять как строить модель, решающую поставленную задачу, посмотрим как она решается в случае задач классификации.

В приведенной выше постановке задачи единственным отличием задачи классификации от задачи регрессии является то, что для задач классификации вероятностное пространство  $L_1(\Omega)$  является дискретным.

Когда задача моделирования распределения решается для дискретного  $L_1(\Omega)$ , т.е. для классификации, реальную плотность распределения приближают функциями вида

$$\sum_{k=1}^K \mathbb{1}_{A_k},$$

где  $A_k \subseteq \Omega$ ,  $\mathbb{1}_A$  – характеристическая функция множества  $A$ .

Именно так мы и поступим.

Только для решения задачи регрессии моделировать лучше не плотность, а функцию распределения. На это есть ряд причин.

Во-первых, использование функции распределения является более робастным, чем использование плотности.

Во-вторых, плотность распределения должна удовлетворять свойству  $\int_{\mathbb{R}} p(t) dt = 1$ . Это свойство может быть сложнее удовлетворить при построении модели, чем соответствующее ограничение на функцию распределения:

$$\begin{aligned} \lim_{t \rightarrow -\infty} F(t) &= 0, \\ \lim_{t \rightarrow +\infty} F(t) &= 1. \end{aligned}$$

## 2.2 Построение модели

Для построения модели вместо классического для задач регрессии подхода с построением модели как параметрического отображения

$$M_\theta : X \rightarrow \mathbb{R}$$

и последующим нахождением  $\theta$  путем оптимизации, будем строить модель сразу приближающую функции распределения:

$$M_\theta : X \rightarrow (\mathbb{R} \rightarrow [0, 1])$$

или, что тоже самое:

$$M_\theta : X \times \mathbb{R} \rightarrow [0, 1]$$

То есть каждой паре  $(x, t)$ ,  $x \in X, t \in \mathbb{R}$ , наша модель будет сопоставлять число в интервале  $[0, 1]$ .

Например, для нейронных сетей этого легко добиться, поместив сигмоиду последним слоем сети.

Информация о реальном семействе распределений  $\{P_x\}_{x \in X}$ , которой мы располагаем, отражена в имеющейся у нас обучающей выборке  $\{(x_i, y_i)\}_{i=1}^N$ .

Эта обучающая выборка порождает набор тривиальных функций распределения  $\{F_i\}_{i=1}^N$ :

$$F(t) = \begin{cases} 1, & t \geq y_i \\ 0, & t < y_i \end{cases}$$

Чтобы уйти от задачи построения модели аппроксимирующей выборку функций к хорошо изученной задаче построения модели аппроксимирующей выборку точек, перейдем от выборки  $\{(x_i, y_i)\}_{i=1}^N$  к выборке

$$\bigcup_{i=1}^N \{(x_i, t_j, F_i(t_j))\}_{j \in J_i}.$$

Для этого для каждого  $i = 1 \dots N$  случайным образом подберем числа  $t_j$ , для  $j \in J_i$ , из некоторого диапазона допустимых значений  $y$ .

Таким образом, мы приходим снова к классической задаче регрессии. Но фазовым пространством для нее будет не исходное пространство  $X$ , а пространство  $X \times Y$ , где  $Y \subseteq \mathbb{R}$  – множество допустимых значений  $y$ .

Т.е. мы получили обычную задачу регрессии для выборки  $\{(z_k, u_k)\}_{k=1}^M$ , где  $z_k = (x_l, t_s)$ , а  $u_k = F_l(t_s) \in [0, 1]$ , для некоторых  $l$  и  $s$ .

Для решения этой задачи можно применить любой алгоритм обучения с учителем из огромного арсенала методов решения задач регрессии.

## 2.3 Ограничения

Поскольку описанный выше способ моделирует построение функций распределения, наша модель должна удовлетворять некоторым дополнительным ограничениям. Пусть

$$M_\theta : X \times Y \rightarrow [0, 1], \theta \in \Theta$$

параметрическое семейство моделей и  $\theta_0$  – оптимальная оценка параметра, дающая приближение реального семейства распределений  $\{P_x\}_{x \in X}$  и

$$M = \lim_{\theta \rightarrow \theta_0, \theta \in \Theta} M_\theta$$

итоговая модель.

Тогда должны быть выполнены требования, что для каждого  $x \in X$   $M(x, \cdot) : t \rightarrow [0, 1]$  – является функцией некоторого распределения.

То есть должны быть удовлетворены следующие условия:

**DF.1**  $\lim_{t \rightarrow -\infty} M(x, t) = 0, \lim_{t \rightarrow +\infty} M(x, t) = 1$

**DF.2**  $t_1 \leq t_2 \Rightarrow M(x, t_1) \leq M(x, t_2)$

**DF.3**  $M(x, t) \in [0, 1], \forall t \in \mathbb{R}$

Все эти условия, в общем случае, не обязаны выполняться по построению моделей  $M_\theta$  способом, описанным выше.

Условие **DF.3** может быть удовлетворено путём наложения ограничений на саму модель. Например, для нейронных сетей можно последующим слоем разместить сигмоиду.

Практика показала, что для правильно построенной модели при достаточном объеме обучающей выборки условия **DF.1** и **DF.2** будут выполнены автоматически. Но эти условия должны быть вынесены на этап валидации в качестве дополнительного обязательного критерия правильности построения модели.

### 3 Итог

Кратко опишем алгоритм.

Дана обучающая выборка  $\{(x_i, y_i)\}_{i=1}^N$ .

1. Находим диапазон допустимых значений  $Y$ .

Например,

$$Y = [\min_i y_i - a, \max_i y_i + a],$$

где  $a$  – некоторое число, подбираемое исследователем.

2. Затем для каждой пары  $(x_i, y_i)$  случайно генерируем набор точек  $\{t_j\}_{j \in J_i} \subseteq Y$ .

$\{t_j\}$  нужно генерировать так, чтобы было достаточно точек, лежащих левее  $y_i$  и достаточно точек, лежащих правее  $y_i$ .

Можно задать разбиение  $\{t_j\}_{j \in J} \subseteq Y$  одинаковое для всех  $i$ , но тогда мы теряем разнообразие обучающей выборки в тех случаях, когда  $(x_i, y_i)$  и  $(x_k, y_k)$  – близкие, но не совпадающие точки.

3. После того, как точки  $\{t_j\}_{j \in J_i}$  сгенерированы, генерируем новую обучающую выборку, как объединение выборок

$$\bigcup_{i=1}^N \{(x_i, t_j, u_{ij})\}_{j \in J_i}$$

$$u_{ij} = \begin{cases} 1, & t_j \geq y_i \\ 0, & t_j < y_i \end{cases}$$

Для удобства обозначим  $z_{ij} = (x_i, t_j)$ .

$z_{ij}$  будут лежать в области определения нашей модели, т. е. будут являться признаками, а  $u_{ij}$  в области значений, т. е. будут являться таргетами.

4. На практике новую полученную выборку лучше случайно перемешать, перед тем как приступить к обучению модели.
5. Строим модель обучения с учителем на обучающей выборке  $\{(z_{ij}, u_{ij})\}$  как для обычной задачи регрессии.
6. Проводим валидацию модели.

В частности, на удовлетворение условия того, что  $M(x, t)$  является функцией распределения по  $t$  для каждого  $x \in X$ , т. е. проверяем **DF.1**, **DF.2**, **DF.3**.

## 4 Валидация

Как и для обычных задач регрессии, невозможно дать какие-то универсальные критерии оценки качества построения модели. Но можно дать несколько рекомендаций, позволяющих оценить это качество.

В любом случае, модель  $M(x, t)$  должна быть функцией распределения по  $t$  для всех  $x \in X$ . Если ограничения **DF.1**, **DF.2**, **DF.3**, налагаемые на функции распределения, не выполнены для  $M(x, \cdot)$ , то такую модель следует отвергнуть как некачественную.

Сам алгоритм по построению является обычной задачей регрессии. И к его результатам применимы все метрики качества, применяемые к задачам регрессии.

Для получения этих метрик тестовую выборку  $\{(x_i, y_i)\}$  нужно привести к виду  $\{(z_{ij}, u_{ij})\}$  тем же способом, что и обучающую.

Кроме того, мы можем перейти на уровень исходных данных и для каждой  $x_i$  из тестовой выборки посчитать среднее значение  $\hat{y}_i$  как:

$$\hat{y}_i = \int_{t \in Y} t dM(x_i, t)$$

Таким образом, мы можем оценивать качество модели так, как если бы мы не строили распределения, а решали обычную задачу регрессии.

Замечу, что в некоторых случаях, вместо оценки среднего  $\hat{y}$  более уместным будет оценивать наиболее вероятное значение  $y$ :

$$\hat{y}_i = \arg \max_{t \in Y} \frac{\partial M(x_i, t)}{\partial t}$$

В целом, подход с моделированием распределений вместо моделирования значений дает не меньше, а даже больше способов оценки качества модели.

## 5 Эксперименты

В качестве базовой закономерности возьмём функцию

$$f(x) = 1 - x^2 + \frac{3}{2}x - \sin(2\pi x^2), \quad (1)$$

на отрезке  $x \in [0, 1]$  (Рис. 1).

Моделируем <sup>1</sup>. закономерность, определяемую выражением (1) плюс нормальный шум  $\mathcal{N}(f(x), \sigma(x))$ , где

$$\sigma(x) = 0.05 + \frac{x}{2}.$$

То есть для каждой точки  $x \in [0, 1]$  нашего фазового пространства значения соответствующей случайной величины, определяемой моделируемой закономерностью, распределены по закону:

$$\mathcal{N}(f(x), \sigma(x)).$$

Если решать обычную задачу регрессии с помощью нейронной сети, то можно увидеть, что выдаваемые моделью ответы будут довольно хорошо ложиться на средние значения (Рис. 1). Как это и ожидалось.

Нахождение распределений методом, описанным в настоящей статье, тоже даёт хорошие результаты.

На рисунке 2 представлены функция распределения (а) и построенная по этой функции плотность распределения (б).

---

<sup>1</sup>Исходный код экспериментов можно найти по адресу:  
[https://github.com/valmat/regress\\_distr/blob/master/experiments.ipynb](https://github.com/valmat/regress_distr/blob/master/experiments.ipynb)

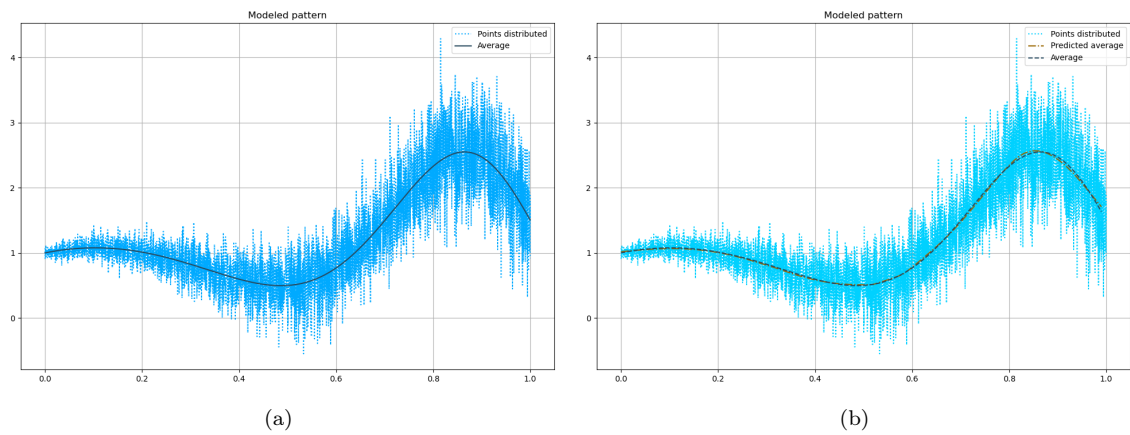


Рис. 1: (a) Моделируемая закономерность (b) Решение обычной задачи регрессии

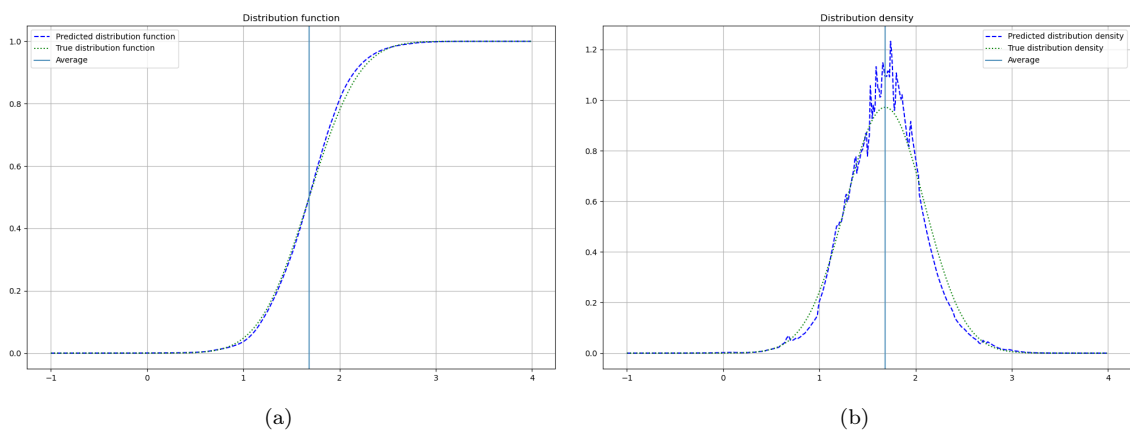


Рис. 2: (a) функция распределения (b) плотность распределения

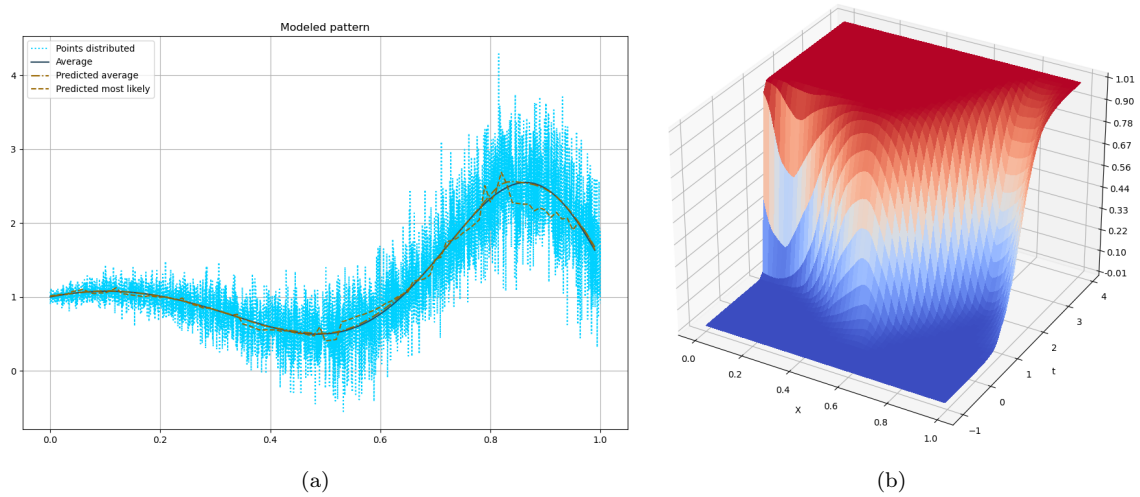


Рис. 3: (a) Средние и наиболее вероятные значения (b) Модельная функция распределения  $F(x, y)$

На рисунке 3 (a) изображены средние и наиболее вероятные значения, построенные по модельным функциям распределения  $F(x, y)$ .

На рисунке 3 (b) проиллюстрирован вид смоделированной функции распределения  $F(x, y)$ .

## 6 Заключение

На практике, при достаточной объеме обучающей выборки, непрерывные алгоритмы машинного обучения, такие как нейронные сети, дают хорошее приближение для функций распределения.

В обучающей выборке могут быть образцы с близкими значениями признака  $x$ , но различными значениями таргета  $y$ . Все они вносят вклад в обучение функций распределения.

Эксперименты и практический опыт показывают, что ограничения, накладываемые на функцию распределения, удовлетворяются.

Прогнозирование распределений вместо прогнозирования средних значений дает намного более богатые возможности для принятия решений.

Моделирование распределений вместо моделирования значений требует меньше дополнительных и часто невыполнимых ограничений.

Например, если рассмотреть решение одной и той же задачи моделированием распределений  $M_\theta(x, t) \in [0, 1]$  и моделированием значений  $R_\theta(x) \in \mathbb{R}$ , то применение МНК, то есть MSE в качестве функций потерь, для  $R_\theta(x)$  равносильно предположению

$$M_\theta(x, t) \sim \mathcal{N}(t, \hat{y}, \sigma),$$

что, чаще всего, неверно.

Конечно для нахождения оптимальной модели  $M_{\theta_0}(x, t)$  мы тоже вынуждены сделать некоторое предположение на вид распределения ошибки  $M_{\theta_0}(x, t) - \hat{u}$ , но это предположение ограничивает нас менее жестко.

Платой за преимущества, даваемые моделью, предсказывающей распределения, является необходимость обучать более емкую модель. А следовательно, более медленная скорость сходимости по сравнению с классическим подходом.

Действительно, нам нужно выучить не просто среднее, но и дополнительную информацию о форме распределения.

Кроме того, мы вынуждены искусственно увеличить объем обучающей выборки, выполняя пополнение ее таким образом, как это было описано в (формула \*).

Это дополнительно приводит к замедлению обучения и требует больше вычислительных ресурсов.