Информативные априорные предположения в задачах привилегированного обучения

Нечепуренко И. О.[1]***, Нейчев Р. Г.***[2]***, Стрижов В. В.***[2]
ivan.nechepurenco@yandex.ru
[1]Московский Физико-Технический Институт

В данной работе рассматриваются различные методы построения моделей оптимальной вычислительной сложности. Затраты по времени и памяти для работы модели играют очень важную роль в большинстве сфер человеческой жизнедеятельности. В работе предлагается использование алгоритмов привилегированного обучения, использующих априорные предположения. Исследуются алгоритмы мультимоделирования, дистилляции и контроля сходства, проведен подробный анализ ошибки на различных данных. Применение алгоритмов согласовано с некоторыми жизненными сценариями. Улучшение качества работы алгоритмов проверено и на синтетических, и на реальных данных.

Ключевые слова: *мета-обучение, мультимоделирование, дистилляция, контроль сходства* .

1 Введение

Рассматривается задача построения алгоритмов анализа данных максимальной точности с ограниченными затратами ресурсов. Решается задача обучения с учителем. Машинное обучение связанно с обработкой больших массивов данных и итерационными процессами оптимизации, а это влечет за собой сильные затраты по ресурсам, которые не могут себе позволить даже крупные компании.

Существуют различные подходы к решению задачи ([1], [3], [2], [4]). В [2] рассматривается один из них - мультимоделирование. Рассматривается [4] алгоритм "смесь экспертов". Особенность алгоритма - использование специальной функции, именуемой шлюзовой. Шлюзовая функция определяет ценность предсказаний конкретного эксперта - одной из базовых моделей. Одной из полезных особенностей способа можно назвать возможность отфильтровывать модели, имеющие слабую предсказательную способность.

Также рассмотрен иной подход - модификация алгоритма обучения с учителем, в котором в роли учителя могут выступить не только экспериментальные данные, но и ответы машины, обученной на более полных данных либо более сложным алгоритмом. Два метода, основанных на парадигме "машина учит машину": дистилляция [2] и контроль сходства [3] , были обобщены [1] .

Рассматриваемые нами алгоритмы достаточно молодые: например,контроль сходства был представлен Вапником в 2009 г., а метод дистилляции - в 2015 -м году Д. Хинтоном. Тем не менее, эти алгоритмы уже используются на практике: например, не так давно они использовались для использования энергии датацентрами Яндекса.

Предлагается на реальных данных: данных о ценах энергопотребления в Польше в зависимости от времени протестировать известные алгоритмы, сравнить сложности вычисления итоговую точность при различных реализациях. Построение достаточно точного и малозатратного алгоритма позволит локально решать проблему оптимального плана энергопотребления электричества, позволит сэкономить определенное количество денег крупным IT-компаниям.

2 Постановка задачи

Наша задача - разаработка моделей для решения классических задач обучения с учителем: как классификации, так и регрессии, а также более сложных. Итак, предполагается, что мы имеем набор объектов $\mathbb{X}$. У каждого объекта есть набор признаков, принимающих действительные значения, и лежащих в $\mathbb{R}^m$. Такие значения можно задать матрицей $\mathbf{X} = [x_i]_{i=1}^n$, где x_i - как раз вектор признаков i-го объекта. Также есть матрица ответов $\mathbf{Y} = [y_i]_{i=1}^n$. Последняя, в завимости от задачи, может, в зависимости от постановки задачи, содежит метки классов, векторы значений и векторы распределений, $y_i \in \mathbb{R}^r$ Часто также обе матрицы объединяют в одну - $\hat{\mathbf{X}}$.

Наша цель - построение оптимальной модели $\hat{f} : \mathbb{R}^m \to \mathbb{R}^r$, с минимизирующую функцию ошибки $S(f, \mathbf{X}, \mathbf{Y})$, S принимает значения в $\mathbb{R}_+$, с ограничениями на сложность:

$$\hat{f} = \arg\min_{f, |f| \leqslant M} S(f, \mathbf{X}, \mathbf{Y})$$

Точное определение понятия сложности, впрочем, выходит за рамки нашего исседования.

2.1 Задача многоклассовой классификации.

Положим $y_i \in \delta_r$, где δ_r - пространство векторов вероятности размерности r:

$$y_i = [y_i^1, ..., y_i^r],$$

$$\forall k : 0 \leqslant y_i^k \leqslant 1,$$

$$\sum_{i=1}^{r} y_i = 1$$

Такая задача называется задачей классификации на k классов, есть множество функций ошибки, мы будем использовать кросс-энтропию:

$$S(y_i, \hat{f}(x_i)) = -\sum_{i=1}^{r} y_i^k \log \sigma(\hat{f}(x_i)^k),$$

$$\sigma(\hat{y})^k = \frac{\exp y^k}{\sum\limits_{k'=1}^{r} \exp y^{k'}}$$

Функция $\sigma(\hat{y})^k$ также называется операцией softmax.

2.2 Задача декодирования

. Если матрица ответов состоит из действительных векторов $y_i \in \mathbb{R}^r$, то задачам декодирования соответствуют следующие функции ошибки:

$$MAE(y_i, \hat{f}(x_i)) = ||y_i - \hat{f}(x_i)||_1,$$

$$MSE(y_i, \hat{f}(x_i)) = ||y_i - \hat{f}(x_i)||_2,$$

$$MAPE(y_i, \hat{f}(x_i)) = ||\frac{y_i - \hat{f}(x_i)}{y_i}||_1.$$

2.3 Прогнозирование временных рядов как частный случай декодирования.

Определение: временной ряд $s = [s_T, ..., s_i, .., s_1]$ - последовательность наблюдений $s_i = = s(t_i)$.

Также предположим, что нам дан набор из нескольких таких временных рядов $D = s^q$, $s \in \mathbb{R}^T$, $q = 1, ..., Q$. Каждому s^q соответствует частота семплирования: $\frac{1}{\tau^{(q)}} : t_i^{(q)} = i \cdot \tau^{(q)}$. Сама же задача прогнозирования временных рядов звучит так: нам дается предыстория длиной δt_p, и из этого необходимо получить прогноз $\hat{s}$, где $[\hat{s}(t_i)] : T_{max} + \delta t_r \geqslant t_i > T_{max}$, где δt_r - некоторый промежуток времени, на который прогноз и делается. Далее, в исходной формулировке можно сделать замену $y_i^{(q)} = [s^q(t_i), ..., s^q(t_i - \delta t_r)]$; $x_i = [s^q(t_i - - \delta t_r - 1), ..., s^q(t_i - \delta t_r - \delta t_p)]$. Мы можем построить матрицу плана $\hat{\mathbf{X}}$, выбрав множество моментов разбиения $t_i, i = 0...(n)$ так, что сегменты $s_i^q = [y_i|x_i]$, покрывающие временной ряд s^q, были упорядочены: $t_{i+1} > t_i \forall i$, и разбив каждый из рядов s^q.

2.4 Процедура скользящего контроля для временных рядов.

Процедура скользящего контроля - один из методов проверки адекватности модели $\hat{f}$ на базе исторических данных. На самом деле, это стандартный алгоритм кросс-валидации, только с учетом специфики случайных процессов. В рамках этой процедуры рассматривается V сегментов времени, упорядоченных хронологически. Каждый из сегментов имеет фиксированную длину δ_b, начинается во время t_B и соответствует матрице плана $\hat{\mathbf{X}}_b$.

Алгоритм:

1. Фиксируется некоторое семейство функций $\mathbb{F}$, среди которых и ищется оптимальная модель. Также полагается $b = 0$
2. Первой строкой матрицы плана $\hat{X}_b$ положим пару векторов $y_{val,b}, x_{val,b}$, соответствующую промежутку длиной δt_r
3. Теперь дополним матрицу локальной предысторией $\hat{\mathbf{X}}_{train,b} = [\mathbf{Y}_{train,b}, \mathbf{X}_{train,b}]$, соответствующей промежутку $\delta t_B = \delta t_r$: это будет
4. $\hat{f}$ ищется как решение исходной задачи минимизации на подпространстве $\hat{\mathbf{X}}_{train,b}$.
5. Ошибка оценивается на $[y_{val,b}, x_{val,b}]$
6. Повторение иттерации с пункта 2

3 Построение мультимоделей

Определение 1. *Шлюзовая функция* - $\boldsymbol{\pi} : \mathbf{X}^m \to \boldsymbol{\Delta}^K$ - *отображение, определяющее правдоподобие* π_k *модели* f_k *в мультимодели из* K *моделей. При этом* $\boldsymbol{\Delta}^K = \{\pi = = (\pi_1, .., pi_k)|\pi_i \geqslant 0 \forall i = 1..K, \sum_{k=1}^{K} p_i = 1\}$.

Определение 2. *Мультимодель* - *модель* $\overline{\mathbf{f}} : \mathbb{R}^m \to \mathbb{R}^r$, *аггрегирующая предсказания других моделей* $\mathbf{f}_1, ..\mathbf{f}_k$, *осуществляющих преобразования в тех же пространствах.*

В рассматриваемых нами задачах пользуемся предположением о том, что вектор ответов $\mathbf{y}$ представляет собой предсказание некоторой модели $\mathbf{f}(x, \mathbf{w})$ ($\mathbf{w}$ - вектор параметров), с распределенным нормально шумом: $y \sim N(\mathbf{f}(x, \mathbf{w}), \beta)$.

3.1 Смесь экспертов

Определение 3. *Смесь экспертов* - *мультимодель, определяющая правдоподобие* $\boldsymbol{b}_k$ *каждой модели* $\mathbf{f}_k$ *на объекте* x *на основе его признаков.*

$$\overline{\mathbf{f}} = \sum_{k=1}^{K} \pi_k \mathbf{f}_k$$

$$\pi_k(\mathbf{x}, \mathbf{V}) : \mathbb{R}^m \to [0; 1] \forall k = 1..K$$

$$\sum_{k=1}^{K} \pi_k(\mathbf{x}, \mathbf{V}) = 1 \forall x.$$

Здесь $\mathbf{V}$ - вектор параметров шлюзовой функции.

Будем рассматривать $\boldsymbol{\pi}$ как случайный вектор. Каждая модель $\mathbf{f}_k$, входящая в состав мультимодели порождает пару $(\mathbf{x}, \mathbf{y})$ с вероятностью $p(k|\mathbf{x}, \mathbf{w}_k)$. Распределение векторя ответов же представим в виде:

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^{K} p(\mathbf{y}, k|\mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^{K} p(k|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{y}|k, \mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k(\mathbf{x}, \mathbf{v}_k) N(\mathbf{y}|\mathbf{f}(\mathbf{x}, \mathbf{w}_k), \beta_k)$$

Шлюзовую функцию мы можем представить как softmax:

$$\pi_k(\mathbf{x}, \mathbf{v}_k) = \boldsymbol{\sigma}(\mathbf{v}_k^T \mathbf{x}) = \frac{exp(\mathbf{v}_k^T \mathbf{x})}{\sum\limits_{k'=1}^{K} exp(\mathbf{v}_{k'}^T \mathbf{x})}$$

Пользуясь этой формулой, мы можем, наконец, расписать итоговую формулу распределения ответов на объекте $\mathbf{x}$:

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k(\mathbf{x}, \mathbf{v}_k) N(\mathbf{y}|\mathbf{f}(\mathbf{x}, \mathbf{w}_k), \beta_k) = \sum_{k=1}^{K} \frac{exp(\mathbf{v}_k^T \mathbf{x})}{\sum\limits_{k'=1}^{K} exp(\mathbf{v}_{k'}^T \mathbf{x})} exp\left(-\frac{1}{2\beta_k}(\mathbf{y} - \mathbf{f}_k(\mathbf{x}, \mathbf{w}_k))^2\right)$$

Введем матрицу $\boldsymbol{\Gamma} = [\gamma_{ik}]$, где γ_{ik} - правдоподобие модели $\mathbf{f}_k$ на объекте $\mathbf{x}_i$. Также можно отметить, что строки матрицы $\boldsymbol{\Gamma}$ - значения шлюзовой функции $\boldsymbol{\pi}$ на соответствующих объектах из выборки.

3.2 EM-алгоритм

Алгоритм подбора гипперпараметров $\boldsymbol{\theta}$ используют двухшаговый алгоритм Expectation-Maximization, состоящий из **E**- и **M**- шагов соответственно. Далее s обозначает номер итерации.

E-шаг: Имеем текущие оценки $\mathbf{w}_1^s, .., \mathbf{w}_K^s, , \mathbf{V}^s, \beta^s$.

Пересчитываем матрицу $\boldsymbol{\Gamma}^{(s+1)}$ по следующему принципу:

$$\gamma_{ik}^{(s+1)} = E(z_{ik}) = p(k|\mathbf{x}_i, \boldsymbol{\theta}^{(s)}) = \frac{\boldsymbol{\pi}_k(\mathbf{x}_i) N(y_i|\mathbf{f}_k(\mathbf{x}_i, \mathbf{w}_k^{(s)}), \beta_k^{(s)})}{\sum\limits_{k'=1}^{K} \boldsymbol{\pi}_{k'}(\mathbf{x}_i) N(y_i|\mathbf{f}_{k'}(\mathbf{x}_i, \mathbf{w}_{k'}^{(s)}), \beta_{k'}^{(s)})}$$

M-шаг: получив матрицу $\mathbf{\Gamma}^{(s+1)}$, можем оптимизировать параметры для $\mathbf{f}_k$, которые входят в смесь:

$$\mathbf{v}_k = argmax_{\mathbf{v}_k} \sum_{i=1}^{n} \gamma_{ik}^{s+1} ln \pi_k(\mathbf{x}_i, \mathbf{v}),$$

$$\mathbf{w}_k = argmax_{\mathbf{w}_k} \left[-\sum_{i=1}^{m} \gamma_{ik}^{s+1} (\mathbf{y}_i - \mathbf{f}_k(\mathbf{x}_i, \mathbf{w}_k))^2 \right],$$

$$\beta_k = \arg\max_{\beta} \left[nln\beta - \sum_{i=1}^{m} \frac{1}{\beta} (\mathbf{y}_i - \mathbf{f}_k(\mathbf{x}_i, \mathbf{w}_k))^2 \right].$$

Начальная инициализация очень сильно определяет сходимость метода и итоговую точность. Для качественной инициализации используются априорные знания о данных либо множественные запуски алгоритма.

4 Привилегированное обучение

Пусть для некоторых объектов $\mathbf{x}$ доступна *привилегированная* информация $\mathbf{x}^*$. Введем функции ученика $\mathbf{f}_s \in \mathcal{F}_s$ (student) и учителя $\mathbf{f}_t \in \mathcal{F}_t$ (teacher):

$$\mathbf{f}_s : \mathbf{x} \to \mathbf{y}, \mathbf{f}_t : \mathbf{x}, \mathbf{x}^* \to \mathbf{y}$$

4.1 Контроль сходства (similarity control)

Будем строить функцию учителя $\mathbf{f}_t : \mathbf{x}^* \to \mathbf{y}$ использует только привилегированную информацию. Тогда задача сводится (ссылка) к поиску седловой точки лагранжиана L:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!можно будет доработать!!!!!!!!!!!!!!!!!!!!!!!!!!!!

$$L(\alpha, \beta) = \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i \mathbf{y}_i \mathbf{f}_s(\mathbf{x}_i) + \frac{\gamma}{2}||\mathbf{w}^*||^2 + \sum_{i=1}^{n} (\alpha_i + \beta_i - C)\mathbf{f}_t(\mathbf{x}_i^t),$$

Здесь α, β - множители Лагранжа. Привлечение учителя $\mathbf{f}_t$ помогает решать задачу в случае линейно неразделимой выборки. Предполагается, что функция учителя имеет меньшую сложность, но имеет более детальное описание.

4.2 Дистилляция(distillation)

Процесс дистилляции отличается от предыдущего тем, что модель учителя более сложная, чем ученика, но привилегированная информация отсутствует. Функция учителя задается следующим образом:

$$\mathbf{f}_t = \arg\min_{\mathbf{f} \in \mathcal{F}_t} \frac{1}{n} \sum_{i=1}^{n} l\left(\mathbf{y}_i, \boldsymbol{\sigma}(\mathbf{f}(\mathbf{x}_i))\right) + \Omega(||\mathbf{f}||),$$

где $\boldsymbol{\sigma}$ - softmax, l - кросс-энтропия, $\Omega : \mathbb{R} \to \mathbb{R}$ - некоторый регуляризатор.

Обучение $\mathbf{f}_t$ происходит на всей доступной выборке. Затем происходит сглаживание предсказания:

$$\mathbf{s}_i = \boldsymbol{\sigma}(\mathbf{f}_i(\mathbf{x}_i)/T),$$

здесь T - температура сглаживания, $\mathbf{s}_i$ - "сглаженное"предсказание. Выше T - ближе распределение вероятностей $\mathbf{s}_i$ к равномерному.

$$\mathbf{f}_s = \lim \arg\min_{\mathbf{f}\in\mathcal{F}_s} \frac{1}{n}\sum_{i=1}^{n} \left[(1-\lambda) l(\mathbf{y}_i, \boldsymbol{\sigma}(\mathbf{f}(\mathbf{x}_i))) + \lambda l(\mathbf{s}_i, \boldsymbol{\sigma}(\mathbf{f}(\mathbf{x}_i)))\right],$$

здесь $\lambda \in [0,1]$ - параметр имитации. Варьирование параметра позволяет искать баланс между обучением на исходные векторы $\mathbf{y}_i$ и сглаженные предсказания учителя $\mathbf{s}_i$. Это помогает избегать переобучения на сложные, аномальные примеры.

4.3 Обобщенная дистилляция

Вышеописанные методы можно обобщить. Учитель обучается именно на привилегированном наборе данных $\mathbf{x}^*$, и при этом сложность модели учителя не ограничена. Это объясняется тем, что привилегироанное описание может иметь гораздо более простую структуру, чем стандартное признаковое описание, и, значит, требуются более простые модели.

Общий алгоритм выглядит примерно так:

1) Выбрать параметр имитации λ и температуру сглаживания T

2) Выделить подмножество объектов, обладающих привилегированным описанием и найти оптимальную функцию учителя $\mathbf{f}_t$

3) Используя функцию учителя $\mathbf{f}_t$, построить сглаженные предсказания для всех объектов обучающей выборки.

4) Найти оптимальную функцию ученика.

Как уже было сказано, предыдущие методы являются частными случаями обобщенной дистилляции. Дистилляции соответствует следующий случай:

$$|\mathcal{F}_t|_C \gg |\mathcal{F}_s|_C, \mathbf{x}^* = \varnothing.$$

Если же, наоборот, выполняется

$$|\mathcal{F}_t|_C \ll |\mathcal{F}_s|_C, \mathbf{x}^* \neq \varnothing,$$

то метод является, фактически, контролем сходства.

Общая же идея метода такова: *ученик не может описать ничего из того, что не смог качественно описать учитель*. Таким образом, $\mathbf{f}_s$ "не обращает внимание"на сложные примеры, на которых ошибся учитель $\mathbf{f}_t$, что влечет за собой лучшее описание стандартых объектов.

5 Название раздела

Данный документ демонстрирует оформление статьи, подаваемой в электронную систему подачи статей `http://jmlda.org/papers` для публикации в журнале «Машинной обучение и анализ данных». Более подробные инструкции по стилевому файлу `jmlda.sty` и использованию издательской системы LaTeX 2_ε находятся в документе `authors-guide.pdf`. Работу над статьёй удобно начинать с правки TeX-файла данного документа.привилегированное

5.1 Название параграфа.

Нет ограничений на количество разделов и параграфов в статье. Разделы и параграфы не нумеруются.

6 Заключение

Желательно, чтобы этот раздел был, причём он не должен дословно повторять аннотацию. Обычно здесь отмечают, каких результатов удалось добиться, какие проблемы остались открытыми.

Литература

[1] Bernhard Scholkopf David Lopez-Paz, Leon Bottou. Unifying distillation and privileged information. *Journal of Machine Learning Research*, 23:1117–1193, 2012.

[2] Dean Jeffrey. Hinton Geoffrey E., Vinyals Oriol. Distilling the knowledge in a neural network. *Journal of Machine Learning Research*, 2015.

[3] Vashist A. Vapnik V. A new learning paradigm: Learning using privileged information. *Neural Networks.*, 2009.

[4] Gader Paul D. Yuksel Seniha Esen, Wilson Joseph N. Twenty years of mixture of experts. *Journal of Machine Learning Research*, 23:1117–1193, 2012.