Мультимоделирование, привилегированное обучение

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

Цель данной работы - опробовать различные методы построения моделей оптимальной вычислительной сложности. Затраты по времени и памяти для работы модели играют очень важную роль в большинстве сфер человеческой жизнедеятельности. Носимая электроника и защищенные устройства для решения задач биометрии, устройства автоматической обработки телеметрических данных, системы потоковой аналитики результатов коллизий Большого Адронного Колайдера — это лишь малая доля случаев, когда требуются достаточно быстрые алгоритмы машинного обучения. Существует широкиий спектр способов уменьшения сложности модели. Один них - это разбиение объектов на подобласти, и описание данных в каждой своей собственной моделью. Этот способ называется мультимоделированием (в иноязычной литературе Mixture experts). Другой способ - мета-обучение (Metalearning), метод, основанный на парадгме учителя-ученика, но при этом в качестве учителя могут выступать не только ответы на экспериментальных данных, но и результаты работы другой модели учителя. В случае добавления дополнительной, априорной информации, можно значительно улучшить результаты сходимость происходящего во время обучения оптимизационного процесса, снизить сложность модели, а также повысить итоговое качество модели. Эти свойства были проверены на реальных данных.

Ключевые слова: *машинное обучение, мета-обучение, мультимоделирование* .

1 Введение

Рассматривается задача построения алгоритмов анализа данных максимальной точности с ограниченными затратами ресурсов. Конкретно работа посвящена самому распространенному типу машинного обучения - обучению с учителем. Обычно в промышленности и науке 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. Рассматриваемая нами задача применима практически во всех сферах науки и бизнеса.

Существуют различные подходы к решению задачи. Один из них - мультимоделирование. Нам интересен один конкретный [4] алгоритм: смесь экспертов. В реализации этого алгоритма специальная функция, именуемая шлюзовой, определяет ценность предсказаний конкретного эксперта. Одной из полезных особенностей способа можно назвать возможность отфильтровывать "слабые"модели.

Другой 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] .

Рассматривваемые 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 $s_i = = s(t_i)$. (Выходит так, что время течет из настоящего в прошлое).

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 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$_\varepsilon$ находятся в документе `authors-guide.pdf`. Работу над статьёй удобно начинать с правки TeX-файла данного документа.

3.1 Название параграфа.

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

4 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.