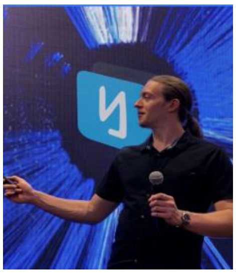
УДК [004.422.833+004.62]

ТЕОРИЯ ИГР КАК ИНСТРУМЕНТ ВИЗУАЛИЗАЦИИ МАШИННОГО ОБУЧЕНИЯ

А.В. Саскевич
Аспирант БГУИР,
инженер-программист Synesis



М.В. Стержанов
Доцент кафедры информатики БГУИР

Белорусский Государственный Университет Информатики и Радиоэлектроники, факультет Компьютерных Систем и Сетей, кафедра Информатики, Республика Беларусь
ООО "Синезис", Республика Беларусь
E-mail: asaskevich@yandex.com

А.В. Саскевич

Окончил Белорусский Государственный Университет Информатики и Радиоэлектроники. Аспирант БГУИР. Работает в компании Synesis в должности инженера-программиста. Проводит научные исследования в области применимости методов машинного обучения и обработки больших объемов информации в сфере образования.

М. В. Стержанов

Окончил Белорусский Государственный Университет Информатики и Радиоэлектроники. Работает в должности доцента кафедры информатики БГУИР. Занимается научным руководством магистрантов и аспирантов.

Аннотация. В данной работе рассматриваются способы визуализации выходных результатов моделей машинного обучения (модели регрессии и модели глубокого обучения) в контексте теории игр посредством вектора Шепли и способы интерпретации с его помощью полученных результатов.

Ключевые слова: теория игр, машинное обучение, визуализация, вектор Шепли.

Введение.

Процесс обучения моделей в машинном обучении, как моделей регрессии и моделей глубокого обучения, так и в целом концептуально подразумевает поиск оптимального решения для некоторой задачи глобального экстремума. Данный процесс может быть представлен в терминах теории игр как некоторая абстрактная игра между различным количеством акторов [1]. В таком случае результат поиска коэффициентов для модели может быть представлен как результат разыгрывания игровых партий.

Для адаптации задачи машинного обучения к терминологии теории игр следует уточнить следующие термины. В терминах машинного обучения коэффициенты (или features) являются основой модели – они определяют результат поиска оптимального решения, в то время как в терминах теории игр таким параметром могут стать как сами игроки, так и их итоговая игровая результативность за некоторое число проведенных партий.

Рассмотрим сведение к игре на примере задачи линейной регрессии [2]. В задаче линейной регрессии □ ∗ □ + □ необходимо подобрать такие коэффициенты □, □, чтобы сумма ошибок при заданной функции ошибок для всех известных точках $x_i, y_i = f(x_i)$ была минимальной, например, при использовании суммы наименьших квадратов, т.е.

$\sum_{i}^{n}(f(x_i)-y_i)^2 \to min$. Коэффициенты *a, b* можно представить в аналогии с игрой двух спортсменов, каждый из которых участвует в соревнованиях, например, в прыжках и художественной гимнастике. Прыгун способен заработать $a * x$ очков исходя из его результата прыжка x метров и количества очков a за каждый преодоленный метр, в то же время гимнаст получает только фиксированное число баллов b в виде оценки за выступление (например, балл от 1 до 10). Общий результат выступления команды представляется как сумма очков первого и второго атлета, а именно $y = a * x + b$. Вопрос задачи стоит следующим образом – зная результаты выступлений разных команд y_i на некотором соревновании и результат прыжка первого атлета x_i , необходимо установить сколько очков комиссия спортивного мероприятия давала за прыжок на *1* метр и за выступление гимнаста. Данная задача может быть решена как с помощью регрессии, так и через теорию игр. Как коэффициенты могут быть связаны между собой или быть независимыми, так и игроки могут либо играть в одной команде ради общего результата, либо соперничать между собой. В конечном итоге каждый коэффициент вносит свой вклад в итоговую модель, аналогично все игроки вносят свою плату за участие в игре и в конце всех партий оказываются со своим итоговым выигрышем или проигрышем. Как накладываются определенные ограничения или область значений на коэффициенты модели, так и игроки следуют определенным заранее определенным правилам, например, в примере выше гимнаст не может получить число баллов, не входящее в промежуток $b \in [-10; 10]$.

Вектор Шепли.

Естественно, при сведении модели машинного обучения к реальным задачам, количество игроков или признаков становится велико и влияние каждого параметра на итоговый результат может сильно варьироваться от признака к признаку, как итог затрудняется отладка модели при обнаружении ошибок в ее работе. Расчет вектора Шепли основан на идее вычисления вклада каждого игрока в результат партии или предсказания модели машинного обучения [3][4]. В упрощенном виде происходит вычисление справедливой ставки каждого игрока (признака) для игры. Для примера рассмотрим следующую ситуацию. Три семьи желают построить общий парк. Первая семья желает построить небольшой парк, например, *20* квадратных метров. Вторая семья занимается спортом и хочет парк на *30* кв.м. Третья семья хочет большой парк со спортивной зоной на *50* кв.м. Постройка каждого метра парка займет, например, *1* у.е. Естественно, каждой семье выгодно, если в результате голосования будет принят ее проект. В случае, если в результате голосования выиграет проект третьей семьи – первой и второй семье придется переплатить, что может быть неприемлемо. Выбрать усредненный вариант постройки – *33* кв.м. (*(20 + 30 + 50) / 3)* тоже неприемлемо, первая семья переплачивает *13* у.е., вторая – *3* у.е., а третья – недоплачивает *17* у.е., не получая того результата, который хотели.

Для вычисления вектора Шепли для указанной задачи поступим следующим образом. Рассмотрим последовательно три ситуации. В первой ситуации каждая семья заплатит *6.67* у.е. за парк в *20* кв.м. Во второй ситуации для постройки парка в *30* кв.м. вторая и третья семья доплатят по *5* у.е. (парк больше на *10* кв.м., а первой семье это превышение не нужно). В третьей ситуации для постройки парка в *50* кв.м. лишние *20* кв.м. постройки ложатся на плечи третьей семьи – они доплачивают *20* у.е., так как парк в *50* кв.м. нужен только им. В итоге получается, что для парка в *50* кв.м. семьи заплатят *6.67, 11.67* и *31.67* у.е. соответственно исходя из их потребностей и предпочтений, что также является справедливой ценой для каждой семьи. В таком случае все получат парк *50* кв.м., а наибольший вклад в общую сумму окажет третья семья. Соответственно, вектор Шепли получает те же значения, что и вклад каждой семьи в итоговую сумму. Кроме того, он определяет, что при изменении параметров парка в большую или меньшую сторону в первую очередь меняться будет вклад третьей семьи.

Техническое решение.

Для визуализации выходного результата модели предлагается фреймворк SHAP (SHapley Additive exPlanations) [5], который поддерживает работу с деревьями решений, регрессионными моделями и моделями глубокого обучения. Кроме того, фреймворк поддерживает интеграцию с популярными фреймворками машинного обучения – Sci-Kit, TensorFlow, XGBoost, CatBoost.

Результаты визуализации.

Рассмотрим применение модели дерева решений для визуализации результата на примере набора данных о ценах на жилье в Бостоне [6]. Исходный датасет содержит 506 строк, каждая из которых содержит 14 различных характеристик, например, уровень преступности, возраст построек, развитость инфраструктуры, уровень загрязненности, число жителей окружающего района, средняя арендная плата. В результате обучения модели может быть построен график, представленный на рисунке 1. Данный график ранжирует параметры модели по уровню их влияния. Соответственно, при оптимизации модели можно опираться на те параметры, которые ведут к улучшению результата. Представленный график подразумевает, что параметры *PTRATIO* (отношение количества учеников к количеству учителей) и *LSTAT* (средний возраст жителей) оказывает наибольшее влияние на модель.

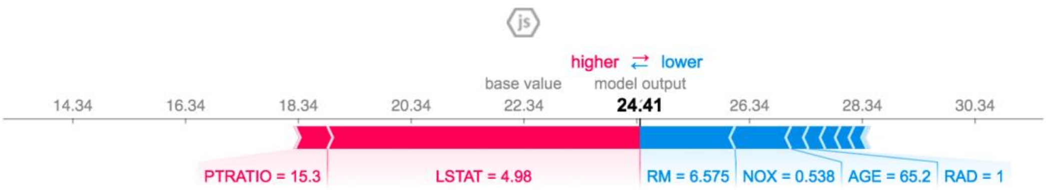


Рисунок 1. Влияние параметров модели на результат модели. Красным цветом – параметры, более влияющие на результат, синим – менее

Обучив модель глубокой нейронной сети на наборе данных mnist [7] можно определить график следующего содержания, представленный на рисунке 2. Также, как и предыдущий график, он отображает (уже в контексте распознавания образов) те пиксели, которые влияют на улучшение результата распознавания (красный цвет) и на ухудшение распознавания (синий цвет) для каждого класса. Как видно из графика, основной результат модели основывается на центральных пикселях входного изображения. Кроме того, видно, что для каждого входного примера основное влияние на распознавание оказывают пиксели, входящие конкретно в тот класс, которому принадлежит входной пример, в то время как остальные классы к распознаванию не приводят.

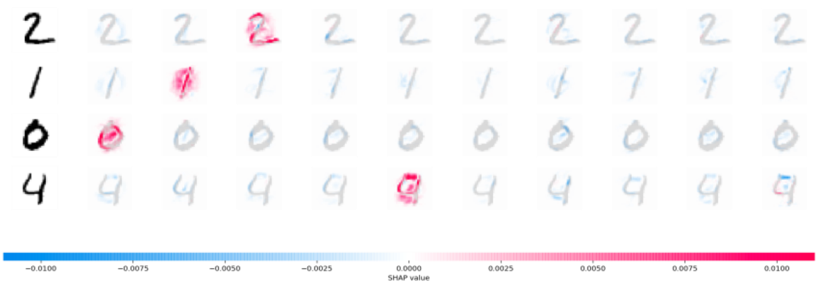


Рисунок 2. Результат обработки модели глубокой нейронной сети. Красным цветом – параметры, более влияющие на результат, синим – менее

Аналогично, на рисунке 3 представлен результат работы для сверточной нейронной сети. По нему также видно, какие группы пикселей (красный цвет) изображения в каждом отдельном классе влияют на результат лучше. При ручной отладке сверток модели можно увидеть похожую картину, которая укажет, какие части изображений являются ключевыми при распознавании объектов и классов для данной модели.

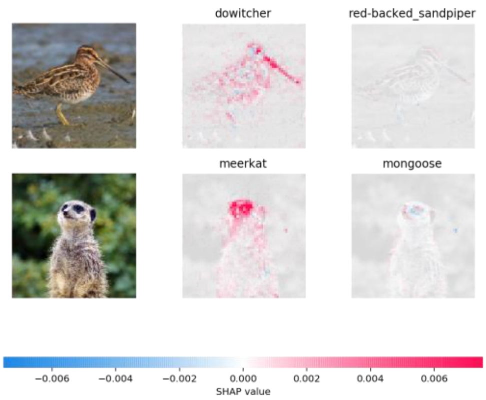


Рисунок 3. Результат обработки модели сверточной сети. Красным цветом – параметры, более влияющие на результат, синим – менее

Заключение.

Работа с вектором Шепли и адаптация машинного обучения к теории игр позволяет детально и обоснованно определять, какие параметры и элементы моделей влияют на итоговый результат. Это поможет как при разработке новых моделей и новых параметров для моделей, ориентируясь на уже существующие и их обоснование инструментами теории игр, так и при отладке уже созданной модели. Разработчик или другой специалист могут при помощи теории игр установить, какая стратегия и какие параметры оказались для каждой конкретной модели преобладающими и ключевыми.

Список литературы

[1] Rezek I. et al. On similarities between inference in game theory and machine learning // Journal of Artificial Intelligence Research. – 2008. – Т. 33. – С. 259-283.

[2] Lipovetsky S., Conklin M. Analysis of regression in game theory approach // Applied Stochastic Models in Business and Industry. – 2001. – Т. 17. – №. 4. – С. 319-330.

[3] Winter E. The shapley value // Handbook of game theory with economic applications. – 2002. – Т. 3. – С. 2025-2054.

[4] Cohen S. B., Ruppin E., Dror G. Feature Selection Based on the Shapley Value // IJCAI. – 2005. – Т. 5. – С. 665-670.

[5] SHAP (SHapley Additive exPlanations) is a game theoretic approach to explain the output of any machine learning model [Электронный ресурс]. – Режим доступа: https://github.com/slundberg/shap. – Дата доступа: 15.02.2021.

[6] The Boston Housing Dataset. A Dataset derived from information collected by the U.S. Census Service concerning housing in the area of Boston Mass. [Электронный ресурс]. – Режим доступа: https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html. – Дата доступа: 17.02.2021.

[7] The MNIST database of handwritten digits [Электронный ресурс]. – Режим доступа: http://yann.lecun.com/exdb/mnist/. – Дата доступа: 19.02.2021.

GAME THEORY AS A VISUALIZATION TOOL FOR MACHINE LEARNING

A.V. SASKEVICH
Postgraduate student of the BSUIR, software engineer Synesis

M.V. STERJANOV
Associate professor at BSUIR

Belarusian State University of Informatics and Radioelectronics, Department of Computer Systems and Networks, Department of Informatics, Republic of Belarus
LLC "Synesis", Republic of Belarus
E-mail: asaskevich@yandex.com

Abstract. This paper discusses how to visualize the output of machine learning models (regression models and deep learning models) in the context of game theory using the Shapley vector and how to interpret the results obtained with it.

Keywords: game theory, machine learning, visualization, Shapley vector.