УДК 004.3 DOI: 10.17587/it.26.499-506

И. Я. Львович[1], д-р техн. наук, проф., office@vivt.ru,
Я. Е. Львович[1], д-р техн. наук, проф., office@vivt.ru,
А. П. Преображенский[1], д-р техн. наук, проф., app@vivt.ru,
О. Н. Чопоров[2], д-р техн. наук, проф., choporov_oleg@mail.ru,
[1] Воронежский институт высоких технологий,
[2] Воронежский государственный технический университет

Особенности методов машинного обучения

Проведен комплексный анализ возможностей подходов, базирующихся на машинном обучении, описаны их слабые и сильные стороны, отмечены основные характеристики алгоритма логистической регрессии и рассмотрены задачи, выполняемые с помощью данного алгоритма. Проведен сравнительный анализ алгоритмов.

Ключевые слова: *машинное обучение, алгоритм, обработка данных, нейросети, глубокое обучение*

Введение

В настоящее время наблюдается рост объема данных в различных сферах, который приводит к необходимости развития методов, связанных с корректной обработкой таких данных. Например, такие крупные компании, как Яндекс или Google, привлекают в качестве инструментария язык программирования R [1] или библиотеки для Python [2]. Это дает возможности для существенного повышения эффективности обработки данных. Помимо программного обеспечения имеют значение разработки по повышению производительности компьютеров. В связи с этим появляется возможность для того, чтобы изучать так называемые большие данные. Их анализ ведется на основе алгоритмов машинного обучения.

Машинное обучение рассматривается в виде подраздела науки об искусственном интеллекте (ИИ). Она связана с построением интеллектуальных машин, которые способны к творческой деятельности, характерной для людей. Например, в 1959 г. работником компании IBM Артуром Самуэлем была написана компьютерная программа для игры в шашки. С определенным положением на доске связывали вес, показывающий вероятность выигрыша. Вначале вероятность рассчитывалась по формуле, в которую входило число шашек и дамок соперников. Но затем Самуэль смог улучшить эффективность такого подхода за счет анализа результатов тысячи партий, и были уточнены позиционные веса. Уже через 15 лет программу по уровню можно было сопоставить с хорошо подготовленным непрофессиональным игроком. Происходило обучение программы, что можно рассматривать как один из первых примеров машинного обучения [3].

Разработка технологий глубинного обучения осуществлялась подобно тому, как формируется структура мозга людей. При этом в целях обработки данных привлекались искусственные нейронные сети, поскольку их работа аналогична работе нейронов мозга. Сама нейронная сеть должна обрабатывать очень большие потоки данных. Это возможно на базе действующих суперкомпьютеров и внедрения технологий Big Data (большие данные) [4]. Для реализации машинного обучения необходим опыт, данные. Большее число данных в системе определяет повышение точности работы с ними компьютера.

Целью работы является обзор методов машинного обучения, используемых для решения большого числа практических задач.

Анализ алгоритмов классификации в машинном обучении

Машинное обучение бывает нескольких видов:

- контролируемое:
  — классификация,
  — регрессия,
  — обнаружение аномалий;
- неконтролируемое.

Рассмотрим подзадачу контролируемого обучения на примере классификации объектов.

Основное различие между двумя типами обучения заключается в том, что контролируемое обучение выполняется с использованием известных выходных значений для образцов. Поэтому целью данного вида обучения является изучение функции, которая с учетом выборки данных и желаемых результатов наилучшим образом приближает входные и выходные данные. Неконтролируемое обучение не имеет целью определить естественную структуру, присутствующую в наборе точек данных [3, 5].

Задача классификации заключается в том, что имеется некий набор объектов, которые были разделены по неким признакам на определенные классы. Набор объектов конечен, и заранее предопределено, к какому из классов относится целевой объект, — такой набор является обучающей выборкой. Требуется построить алгоритм, который сможет классифицировать произвольный объект из множества.

Отличительной чертой задачи классификации является то, что набор ответов ограничен, и эти ответы называются лэйблами или метками классов. Можно сказать, что класс является множеством всех объектов с данным значением метки. Входные данные — это признаковое описание объектов в большинстве случаев, поэтому можно принять, что каждый объект имеет описание, состоящее из набора признаков, которые могут быть числовыми и нечисловыми. Классификация бывает как биноминальной (когда требуется разделить объекты по бинарному признаку на два класса), так и мультиноминальной (когда по общим признакам алгоритм распределяет объекты на классы, которые он сочтет нужным).

В качестве первой обучаемой нейронной сети можно рассматривать персептрон Розенблатта. Он дает возможности для решения задач классификации по двум классам при бинарных входных сигналах. Совокупность входных сигналов обозначают в виде n-мерного вектора x. Его элементы рассматриваются как булевы переменные.

Алгоритм усредненного персептрона дает оценку, которая может использоваться для выбора между двумя разными классами. Фактически, алгоритм усредненного персептрона с двумя классами представляет собой простую реализацию нейронной сети [3, 5].

Укажем примеры задач, где алгоритм эффективен.

- *Проведение прогнозов на фондовых рынках.* Имея информацию по ценам акций в течение последней недели, можно сделать прогноз завтрашней цены акций.
- *Обеспечение предоставления кредита.* Необходимо сделать оценку степени риска по предоставлению кредитов для частных лиц. В качестве исходных данных можно рассматривать доход, предыдущую кредитную историю и др.
- *Процессы управления.* Требуется определить, какие действия должен предпринимать робот, используя видеокамеру (делать поворот направо или налево, двигаться вперед и др.), для достижения цели.

Особенность бинарного персептрона состоит в том, что он в качестве функции активации применяет не сигмоид, а единичную ступеньку с уровнем 0,5. Если проводится расчет средних значений весовых коэффициентов, то анализируется бинарный усредненный персептрон.

В практических случаях подбор коэффициента скорости обучения персептрона большей частью осуществляют экспериментальным образом [3—5]: больше шаг — быстрее модель, меньше точность; меньше шаг — число шагов увеличивается, модель замедляется, точность обучения повышается.

Рассмотрим характеристики алгоритма, базирующегося на бинарной байесовской точечной машине. Алгоритм использует байесовский подход к линейной классификации, называемый "машиной байесовской точки". Этот алгоритм эффективным образом позволяет аппроксимировать оптимальное с точки зрения теории байесовское среднее линейных классификаторов за счет выбора одного "среднего" классификатора — точки Байеса. Так как Байес-Пойнт-машина является байесовской классификационной моделью, она не подвержена переобучению. С точки зрения использования на практике байесовской оценки в первую очередь обращают внимание не на точность, а на ковариантность наблюдаемой вероятности. Это связано с тем, что оценка будет иметь большее значение, если будет больше наблюдаемая вероятность события. Также для байсовского подхода характерен тот факт, что параметров мало, нет необходимости в том, чтобы осуществлять нормализацию данных, тогда разные параметры будут вносить разный вклад.

Для большинства наборов данных стандартная установка 30 итераций обучения достаточна для того, чтобы алгоритм делал точные прогнозы. Иногда точные прогнозы можно сделать, используя меньшее число итераций. Рассмотрим бинарное расширенное дерево принятий решений. Идея метода заключается в обучении ансамбля, где второе дерево исправляет ошибки первого дерева, третье дерево исправляет ошибки предыдущих. Прогнозы базируются на совокупности этих деревьев, что на выходе дает

предсказание. Требуются тегированные данные со столбцом значений для каждого ряда.

В каждом внутреннем узле есть соответствие одной из входных переменных; существуют ребра для детей для каждого из возможных значений этой входной переменной. Каждый лист в дереве решений представляет значение целевой переменной, учитывая значения входных переменных, представленных путем от корня до листа [6]. Деревом решений или деревом классификации является дерево, в котором каждый внутренний (не листовой) узел помечен функцией ввода. Дуги, идущие от узла с меткой входной функции, помечены каждым из возможных значений целевого или выходного признака, или дуга приводит к подчиненному узлу решения с другой функцией ввода [7, 8]. Каждый лист дерева помечен классом или распределением вероятности по классам.

Существуют бинарные леса принятия решений, представляющие собой быстрые, контролируемые ансамблевые модели. Методы ансамбля основаны на общем принципе. В нем вместо рассмотрения одной модели осуществляют построение обобщенных моделей, что значительно улучшает результаты. Далее модели связываются и объединяются. Как правило, модели ансамбля обеспечивают лучший охват и точность, чем отдельные деревья решений.

Преимуществами ансамблевых моделей являются:

- охват границы нелинейных решений;
- интегрирование выбора функций в процессы обучения и классификации;
- использование зашумленных данных;
- необходимость меньшего числа действий для очистки данных;
- тип данных не является ограничением: он может обрабатывать как числовые, так и категориальные переменные.

Недостатки ансамблевых моделей:

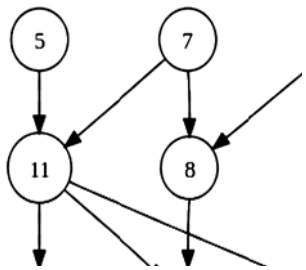
- переобучение модели не всегда простым образом реализуется на практике для деревьев решений. Эта проблема решается путем установки ограничений на параметры обучающейся модели;
- при работе с непрерывными числовыми переменными дерево решений теряет информацию, когда оно классифицирует переменные в разных категориях.

Существуют бинарные джунгли принятия решений. Рандомизированные деревья решений и леса характеризуются богатой историей машинного обучения. Можно увидеть существенные успехи в применении, особенно это относится к компьютерному зрению. При этом в них есть фундаментальное ограничение: когда есть достаточно данных, то число узлов для деревьев решений будет экспоненциальным образом возрастать с глубиной. В определенных приложениях, например, для мобильных или встроенных процессоров, можно рассматривать память в виде ограниченного ресурса. В связи с этим вследствие экспоненциального роста деревьев будет ограничение по их глубине. Ограничена и их потенциальная точность. Для решения можно предложить использовать метод джунглей. Анализ по ансамблям деревьев решений, связанных с ациклическими графами (DAG) (см. рисунок) демонстрирует, что они рассматриваются в виде компактных и мощных дискриминационных моделей в ходе классификации. DAG в принятии решений отличается от традиционных деревьев решений тем, что есть возможности для реализации нескольких путей от корня к каждому из листов.

Направленным ациклическим графом является орграф, для которого нет направленных циклов. При этом могут наблюдаться "параллельные" пути, которые будут идти из одного узла и вести к конечному узлу. Направленный ациклический граф рассматривается в виде обобщения леса деревьев. Каждый узел знает только о предыдущем узле, возможность прийти другим путем, который будет более коротким или точным, неизвестна [7, 8].

Направленные ациклические графы часто применяются при обработке данных, планировании, поиске наилучшего маршрута в навигации и сжатии данных. В основном указанный подход применяется в области финансов и ценных бумаг. Данная структура помогает отслеживать и предугадывать ценовые тренды несмотря на общие финансовые показатели рынка.

Опорные векторные машины (SVM) — весьма популярный и хорошо изученный класс контролируемых моделей обучения, которые могут использоваться в задачах линейной и нелинейной классификации [9]. В последних исследованиях основное внимание уделялось способам оптимизации этих моделей для эффективного масштабирования до более круп-



Схематичное представление взаимодействия узлов ациклического графа (DAG)

ных наборов учебных материалов. В этой реализации от Microsoft Research функция ядра, используемая для сопоставления точек данных с пространством объектов, специально предназначена для сокращения времени, необходимого для обучения, при сохранении большей части точности классификации [10, 11].

После того, как исследователь определяет параметры модели, подготавливается помеченный набор данных в качестве входных данных для модели в целях обучения или для определения гиперпараметров. Затем обученная модель может использоваться для прогнозирования значений для новых входных данных. Данный способ классификации характеризуется весьма широким применением: от задач по распознаванию образов или формирования спам-фильтров до вычислений, показывающих распределения по горячим алюминиевым частицам внутри ракетных выхлопов. Несмотря на то, что подход первоначально был связан с бинарными классификаторами, могут быть возможности для его работы в задачах мультиклассификации.

Рассмотрим устройство алгоритма машины опорных векторов на простом примере. Допустим, у нас на столе лежит множество синих и красных шариков. Если они расположены не в абсолютном беспорядке, то плоскостью мы можем поделить стол на части, попытавшись максимально точно отсортировать шарики по цветам, и, когда добавляется новый шар одного из цветов, будет известно на какую часть стола его положить [10].

Самым большим преимуществом алгоритма машины опорных векторов является тот факт, что он самостоятельно определяет функцию гиперплоскости. Если же шары находятся в сильно перемешанном состоянии, то плоскость здесь не поможет. Но если все шары подкинуть в воздух, то их уже можно будет попробовать разделить листом бумаги. Таким образом, используя повышение разрядности пространства, мы получаем новые возможности для классификации. Использование нуль-пространства (или ядра) матрицы дает отличный инструмент для работы в пространствах более высокой разрядности, где лист бумаги по-прежнему является гиперплоскостью, но уже используется не линейная функция, а функция плоскости.

Предположим, у нас есть набор данных о пациентах, у каждого из них есть показатели: пульс, возраст, анализы, привычки, патологии и т.д. Каждый из параметров является измерением пространства. В результате работы алгоритм машины опорных векторов отображает эти параметры в высшее измерение, находит функцию гиперплоскости и классифицирует объекты. Бинарная, усиленная машина опорных векторов (LD-SVM) является реализацией Microsoft для класса векторных машин.

Рассмотрим особенности *бинарной нейронной сети.* Проведение классификации на базе нейронных сетей можно считать контролируемым методом обучения. Например, такая модель нейронной сети может применяться, чтобы прогнозировать бинарные результаты: существование некоторого заболевания у пациентов или поломку компьютера в течение заданных временных интервалов. Входы являются первым уровнем нейронной сети и подключены к выходному уровню ациклическим графом, состоящим из взвешенных ребер и узлов. Когда нейронная сеть обучается на базе входных данных, изучается связь значений на входах и выходах. Все нейроны, которые находятся на одном уровне, связаны ребрами со всеми нейронами следующего уровня, при этом каждое ребро обладает весом. Для вычисления выхода сети для конкретного входа вычисляется значение для каждого узла в скрытых слоях и на выходном уровне и вычисляется взвешенная сумма значений узлов из предыдущего слоя. После этого к взвешенной сумме применяется функция активации. При обучении нейронных сетей важно правильно выбирать коэффициент скорости обучения, который позволяет управлять величиной коррекции весов на каждой итерации. Большие значения этого параметра ведут к ускорению работы алгоритма, Но при этом может произойти снижение точности настройки модели на минимум функции ошибки, тогда есть вероятность увеличения ошибки обучения. Малые значения коэффициента определяют меньший шаг в коррекции весов. Тогда получается, что число итераций в обучении растет, это определяет уменьшение ошибки обучения. На практике подбор коэффициента скорости обучения выполняют экспериментальным образом.

Алгоритм логистической регрессии для бинарной классификации

Логистическая регрессия (Logistic regression) — это подход, позволяющий строить линейный классификатор для оценок апостериорных вероятностей того, принадлежат ли объекты классам [12, 13]. Другими словами, это тип алгоритма классификации, включающий линейный дискриминант. Регрессия — это метод моделирования целевого значения на основе независимых предсказателей. Этот метод в основном используется для прогнозирования и выяснения причинно-следственной связи между переменными.

Методы регрессии в основном различаются в зависимости от числа независимых переменных и типа отношения между независимыми и зависимыми переменными [14]. В отличие от фактической регрессии логистическая регрессия не пытается предсказать значение числовой переменной с учетом набора входных данных. Вместо этого выходом (результатом) логистической регрессии является вероятность того, что данная входная точка принадлежит определенному классу. Для простоты предположим, что мы рассматриваем только два класса "+" и "−" (для многоклассовых задач существует многолинейная логистическая регрессия), а выходом P_+ является вероятность того, что определенная точка данных принадлежит классу "+". Для определения принадлежности входной точки классу "−" будем соответственно говорить о вероятности $P_- = 1 - P_+$. Центральной посылкой логистической регрессии является предположение о том, что входное пространство можно разделить на две четкие "области" для каждого из двух классов линейной границей. Для двух измерений это прямая линия, для трех измерений это уже плоскость и т.д.

Как в логистической регрессии используется такая линейная граница для количественной оценки вероятности точки данных, принадлежащей определенному классу? Дадим геометрическую интерпретацию этого "деления" входного пространства на две разные области. Рассмотрим для простоты двумерное пространство (x_1, x_2), уравнение прямой в котором имеет вид

$$B_0 + B_1x_1 + B_2x_2,$$

где B_0, B_1, B_2 — коэффициенты.

Теперь в зависимости от расположения точки с координатами (a, b) есть три возможных сценария:

- точка (a, b) лежит в области, определяемой точками класса "+". В результате значение $B_0 + B_1a + B_2b$ будет положительным, лежащим в интервале (0, ∞). Чем больше это значение, тем больше расстояние между точкой и границей и больше вероятность того, что точка (a, b) принадлежит классу "+". Следовательно, P_+ будет лежать в интервале [0,5; 1] [12];
- точка (a, b) лежит в области, определяемой точками класса "−". В этом случае значение $B_0 + B_1a + B_2b$ будет отрицательным, лежащим в интервале (−∞, 0). Чем больше по модулю это значение, тем больше расстояние между точкой и границей и больше вероятность, что точка (a, b) принадлежит классу "−". Следовательно, P_- будет лежать в интервале [0; 0,5];
- точка (a, b) лежит на границе. В этом случае $B_0 + B_1a + B_2b = 0$. Это означает, что модель не может сказать, принадлежит ли точка (a, b) это классу "+" или классу "-". В результате P_+ будет равно 0,5.

Таким образом, получена функция, которая имеет значение, лежащее в интервале (−∞, +∞) с учетом входной точки данных. Но как сопоставить бесконечный интервал возможных значений функции с вероятностью P_+, которая лежит в ограниченном интервале [0, 1]?

Ответ дает функция шансов [12].

Обозначим $P(X)$ — вероятность события X. Отношение шансов $\mathrm{OR}(X)$ определим выражением $P(X)/(1 - P(X))$, которое является отношением вероятности события к вероятности того, что оно не происходит. $P(X)$ принимает значения от 0 до 1, а $\mathrm{OR}(X)$ — значения от 0 до ∞. Но этого по-прежнему недостаточно, потому что граничная функция должна быть определена в диапазоне (−∞, +∞). Для этого необходимо взять логарифм от $\mathrm{OR}(X)$, потому что $\log(\mathrm{OR}(X))$ определен в диапазоне от −∞ до +∞.

В рассматриваемом примере логистическая регрессия заключается в следующем:

Шаг 1. Вычисление граничной функции (в качестве альтернативы, коэффициенты логарифмической функции) $t = B_0 + B_1a + B_2b$.

Шаг 2. Получение коэффициента $\mathrm{OR}_+ = \mathbf{e}^t$ (так как t — это логарифм OR_+).

Шаг 3. Вычисление P_+ с использованием простого соотношения $P_+ = \dfrac{\mathrm{OR}_+}{1 - \mathrm{OR}_+}$.

Таким образом, как только известно t с шага 1, можно комбинировать шаги 2 и 3, чтобы получить соотношение $P_+ = \dfrac{\mathbf{e}^t}{1 - \mathbf{e}^t}$, что и будет являться логистической функцией.

Анализ задач, решаемых с помощью логистической регрессии в машинном обучении

Рассмотрим задачи бинарной классификации на основе логистической регрессии, где необходимо разделить объекты по признаку на два класса.

- *Задача диагностики.* В медицине одной из популярных задач является задача подтверждения диагноза на основе различных факторов и признаков пациентов. Алгоритм принимает решение о вероятности события подтверждения диагноза, основываясь на связях, данных, на которых он обучается. Обучающей выборкой здесь выступают данные о пациентах, которые имеют различные параметры: возраст, пол, социальная группа, результаты анализов, привычки и. т. д. Для

каждого пациента установлен бинарный параметр, который указывает на результат (0 — диагноз не подтвердился, 1 — диагноз подтвержден). После обучения на этих данных, модель строит для себя связи, и при подаче параметров нового неизвестного пациента делает предположение о том, подтвержден диагноз у пользователя или нет.

- *Скоринговые модели.* Задача "подсчета очков" возникает в тех сферах деятельности, где требуется предсказать поведение объекта на основе признаков и данных, которые для него характерны. Наиболее популярная сфера — банковская. Банк, основываясь на большом числе признаков, должен принять решение о выдаче кредита заявителю. Параметры могут быть самыми разнообразными: возраст, средний доход, максимальный доход, кредитная история, стаж работы, общий стаж работы, пол, наличие судимости.

Таких признаков может быть сколь угодно много в обучающей выборке, но главное, что для каждого объекта определен результирующий параметр, например, 0 — клиент не надежный, 1 — кредит можно одобрять.

Задачи бинарной классификации возникают всегда, когда нужно выбрать между двух вариантов. Примерами могут быть следующие ситуации:

- сотрудник безопасности, проверяя посетителя, должен принять решение, основываясь на внешних признаках посетителя, уровне доступа и т. д., пропустить его или нет;
- поисковый робот, найдя документ в мировой сети, должен принять решение — добавлять его в базу индексации или нет;
- клиент смотрит на различные условия банков, принимает решение о том, открывать счет или нет;
- инвестор принимает решение о финансировании проекта.

В основе алгоритма логистической регрессии лежит принцип, что прогнозируется не значение результата (или номера класса), а вероятность того, что результат для нового объекта принимает значение 1. На основе этой вероятности принимается решение о том, к какому классу отнести новый экземпляр. Несмотря на то что решения вроде бы очевидны, сравниваются полученные вероятности с пороговым значением 50 %, но при этом необходимо учитывать дополнительные ошибки при принятии решения.

К примеру, в задачах медицинской диагностики врач заподозрил, что пациент серьезно болен. Варианты здесь следующие:

- с одной стороны, если предположение неверно, но гипотеза одобрена, это приведет минимум к лишним расходам на лекарства, а в худшем — к ухудшению состояния здоровья, потому что пациент лечился не от того заболевания;
- с другой стороны, если предварительный диагноз был правильным, но гипотеза отвергнута, пациент может не получить должного лечения, и это может привести к печальным последствиям.

Можно сделать вывод, что пороговое значение должно быть выбрано не по умолчанию, а исходя из соображения минимизации возможных потерь, вызванных совершением ошибок.

Достоинства алгоритма логистической регрессии:

- он хорошо и подробно изучен;
- адаптирован под большие объемы данных, скорость позволяет молниеносно обрабатывать огромные наборы данных;
- является практически единственным алгоритмом для разрозненных данных с огромным числом признаков;
- логистическая регрессия определят вероятность отношения к разным классам;
- в модели может быть построена и нелинейная граница, если на вход будут поданы полиномиальные признаки.

Недостатки алгоритма логистической регрессии:

- плохо работает для задач, по которым зависимость ответов от признаков — сложная, нелинейная;
- в практических случаях нередко не выполняются предположения теоремы Маркова—Гаусса, в связи с этим работа линейных методов чаще будет менее лучшей, если, например, сравнивать с SVM и методом ансамблей.

Результаты анализа методов

В таблице приведены характеристики алгоритмов бинарной классификации. Как можно заметить, все алгоритмы могут использоваться как для бинарной, так и для мультиноминальной классификации.

Анализ алгоритмов бинарной классификации, которые были рассмотрены выше, показывает, что лучшие результаты по точности предсказания показывают алгоритмы лесов принятия решений, джунглей, усиленные деревья, а также нейронные сети и алгоритмы на основе байесовских принципов (SVM [14] и байесовская точечная машина [15]). Алгоритмы логистической регрессии, персептрона и усиленной машины опорных векторов показывают хороший результат, но в силу особенностей самих алгоритмов это вполне предсказуемый ре-

Сравнительный анализ алгоритмов бинарной классификации

Подход для классификации	Точность	Время обучения	Линейность	Адаптируемость	Объем данных
Логистическая регрессия	Хорошо	Быстрое	Отлично	Хорошо	Малый—большой
Лес решений	Отлично	Умеренное	Хорошо	Хорошо	Малый—большой
Джунгли решений	Отлично	Умеренное	Хорошо	Хорошо	Большой
Нагруженное дерево решений	Отлично	Умеренное	Хорошо	Хорошо	Большой
Нейронная сеть	Отлично	Медленное	Умеренное	Отлично	Среднее
Взвешенный персептрон	Хорошо	Умеренное	Отлично	Умеренное	Среднее
Машина опорных векторов	Отлично	Умеренное	Отлично	Хорошо	Большой
Усиленная машина опорных векторов	Хорошо	Медленное	Хорошо	Отлично	Большой
Байесовский классификатор	Умеренное	Умеренное	Отлично	Умеренное	Среднее

зультат, так как в зависимости от числа классов будут меняться параметры и регуляризация.

По скорости обучения логистическая регрессия выигрывает, она действительно признана одним из популярных алгоритмов за счет скорости обучения и своей простоты. В то же время нейронные сети и усиленная машина опорных векторов являются довольно медленными в силу сложности устройства самих алгоритмов, остальные алгоритмы варьируются по показателям в зависимости от настроек. По кастомизации, очевидно, преуспевают такие алгоритмы, как нейронные сети и усиленная машина опорных векторов, поскольку их можно настроить за счет различных параметров, увеличивая их продуктивность и точность (например, в нейронных сетях можно настраивать число скрытых слоев, число узлов в каждом слое). Одним из важных критериев является то, с какими данными работают алгоритмы; в сводной таблице приведены результаты, показывающие, что для относительно небольших наборов данных будут полезны алгоритмы логистической регрессии и лес принятия решений, в то время что для больших наборов данных более приемлемы алгоритмы джунглей принятия решений и усиленное дерево принятия решений, а также алгоритмы на базе машин опорных векторов.

Рассмотренные выше подходы объединены в студии машинного обучения Azure [16]. Она является целой платформой, которая предназначена для создания, тестирования и развертывания решений для бизнес-аналитики и создания прогнозов на основе данных. Она зарекомендовала себя среди крупных компаний, использующих данный продукт, так как это центральный узел, где собраны многие элементы аналитики, данных и ресурсов. Помимо этого, платформа предлагает на выбор большой набор инструментов и готовых самых популярных алгоритмов для предоставления желаемых мощностей пользователям.

Студия машинного обучения Azure поддерживает язык R [1] для работы с Big Data, а также язык Python [2], который содержит в себе огромные дополнительные возможности для работы с алгоритмами машинного обучения. Основные преимущества, которые получает исследователь:

- для импорта данных из хранилищ Azure и систем hdfs нет установленного предела данных;
- гибкость в политике ценообразования;
- существует множество готовых наборов данных и алгоритмов;
- исследователь может опубликовать свою модель данных в виде веб-сервиса;
- исследователи могут публиковать эксперименты для моделей данных всего за несколько минут, тогда как ученым-экспертам может потребоваться несколько дней, чтобы сделать то же самое;
- чем больше времени исследователь применяет модель, тем больше данных прогнозируется. Например, если тренировать собственную модель в течение нескольких месяцев, можно ожидать ошибок всего один или два раза за все время.

Среди недостатков данной студии можно указать:

- набор GUI алгоритмов довольно ограниченный, но можно подключать модули на языках R и Python;
- цена будет существенно возрастать по мере возрастания используемых мощностей. При использовании бесплатной подписки в Azure предоставляется ограниченное хранилище данных;
- большинство алгоритмов и преобразований имеют ограниченный набор настроек параметров.

Заключение

В работе проведен анализ различных алгоритмов бинарной классификации, которые

используются в машинном обучении. Кроме того, рассмотрены ситуации, когда лучше всего использовать тот или иной алгоритм машинного обучения. Кратко описаны возможности студии машинного обучения Azure.

Список литературы

1. **Кабаков Р.** R в действии. М.: ДМК-Пресс. 2014. 588 с.
2. **Маккинли У.** Python и анализ данных / Пер. с англ. М.: ДМК Пресс, 2015. 482 с.
3. **Goodfellow I., Bengio Y., Courville A.** Deep Learning. The MIT Press, 2016. 801 с.
4. **Айвазян С. А.** Прикладная статистика: классификация и снижение размерности. М.: Финансы и статистика, 1989. 607 с.
5. **Hastie T., Tibshirani R., Friedman J.** The Elements of Statistical Learning. Springer, 2009. 533 p.
6. **Howard D.** ITIL Release Management: A Hands-on Guide. M.: CRC Press, 2010. 344 с.
7. **Rokach L., Maimon** O. Data Mining with Decision Trees: Theory and Applications. World Scientific Publishing, 2008.
8. **Long J.** ITIL Version 3 at a Glance: Information Quick Reference. Boston, MA: Springer, 2008. 974 с.
9. **Воронцов К. В.** Лекции по линейным алгоритмам классификации. URL: http://www.machinelearning.ru/wiki/images/6/68/voron-ML-Lin.pdf (дата обращения: 30.03.2020).
10. **Statnikov A., Aliferis C. F., Hardin D. P.** A Gentle Introduction to Support Vector Machines in Biomedicine: Theory and methods. World Scientific, 2011.
11. **Cristianini N., Shawe-Taylor J.** An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, 2000.
12. **Hosmer D. W.**, **Lemeshow S.** Applied Logistic Regression. New York: Chichester — Wiley, 2002. 383 p.
13. **Дрейпер Н., Смит Г.** Прикладной регрессионный анализ. Пер. с англ. М.: Издательский дом "Вильямс", 2007. 912 с.
14. **Kecman V.** Learning and Soft Computing. Support Vector Machines, Neural Networks, Fuzzy Logic Systems. The MIT Press, Cambridge, MA, 2001. 608 p.
15. **Вьюгин В.** Математические основы теории машинного обучения и прогнозирования. М.: МЦМНО, 2013. 390 с.
16. **Документация** по Azure. URL: https://docs.microsoft.com/ru-ru/azure/ (дата обращения: 30.03.2020).

I. Ya. Lvovich[1], Professor, e-mail: office@vivt.ru, **Ya. E. Lvovich** [1], Professor, e-mail: office@vivt.ru,
A. P. Preobrazhensky[1], Associate Professor, e-mail: app@vivt.ru,
O. N. Choporov[2], Professor, e-mail: choporov_oleg@mail.ru,
[1]Voronezh Institute of High Technologies, Voronezh, Russian Federation,
[2]Voronezh State Technical University, Voronezh, Russian Federation

The Features of Machine Learning Methods

The paper provides a comprehensive analysis of the capabilities of approaches based on machine learning, describes their weaknesses and strengths. The types of machine learning are indicated. The subtask of supervised learning is considered on the example of the classification of objects. The parameters of the binary averaged perceptron are noted. The characteristics of an algorithm based on a binary Bayesian point machine are considered. The advantages and disadvantages of ensemble models are indicated. The features of a binary neural network are considered. The main characteristics of the logistic regression algorithm, advantages and disadvantages are noted. The tasks performed using this algorithm are considered, in which it is necessary to classify objects according to their characteristics into two classes: diagnostic problems, scoring models. The summary table compares the binary classification algorithms for accuracy, training time, linearity, adaptability, data volumes. The features of the Azure machine learning studio, in which the analyzed approaches are applied, are shown.

Keywords: *machine learning, algorithm, data processing, neural networks, deep learning*

DOI: 10.17587/it.26.499-506

References

1. **Kabakov R.** R in action, Moscow, DMK-Press, 2014, 588 p. (in Russian).
2. **Makkinly U.** Python and data analysis. Trans. from English, Moscow, DMK-Press, 2015, 482 p. (in Russian).
3. **Goodfellow I.** Deep Learning, Publishing house The MIT Press, *2016,* 801 p.
4. **Ayvasyan S. A.** Applied statistics: classification and dimensionality reduction, Moscow, Finance and statistics, 1989, 607 c. (in Russian).
5. **Hastie T.** The Elements of Statistical Learning, 2nd edition, Springer, 2009, 533 p.
6. **Howard D.** ITIL Release Management: A Hands-on Guide, Moscow, CRC Press, 2010, 344 p.
7. **Rokach L.** Data Mining with Decision Trees: Theory and Applications, World Scientific Publishing, 2008.
8. **Long J.** ITIL Version 3 at a Glance: Information Quick Reference, Boston, MA, Springer, 2008, 974 p.
9. **Vorontsov K. V.** Lectures on linear classification algorithms, available at: http://www.machinelearning.ru/wiki/images/6/68/voron-ML-Lin.pdf (accessed: 03.30.2020).
10. **Statnikov A.** A Gentle Introduction to Support Vector Machines in Biomedicine: Theory and methods, World Scientific, 2011 (in Russian).
11. **Cristianini N.** An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, 2000.
12. **Hosmer D. W.**, **Lemeshow S.** Applied Logistic Regression, New York, Chichester, Wiley, 2002, 383 p.
13. **Draper N.** Applied regression analysis, Moscow, Williams Publishing House, 2007, 912 c.
14. **Kecman V.** Learning and Soft Computing. Support Vector Machines, Neural Networks, Fuzzy Logic Systems, The MIT Press, Cambridge, MA, 2001, 608 p.
15. **Vyugin V.** Mathematical foundations of the theory of machine learning and forecasting, ICMNO, 2013, 390 p. (in Russian).
16. **Azure** Documentation, available at: https://docs.microsoft.com/ru-ru/azure/ (accessed: 03.30.2020) (in Russian).