А.А. Балатай, С.К. Сагнаева

ВНЕДРЕНИЕ ИНСТРУМЕНТОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ОБНАРУЖЕНИЯ АНОМАЛИЙ

В статье рассмотрены широкие категории задач, которые могут использовать машинное обучение, также рассмотрим валидацию точности МО – моделей для прогнозирования. Разработка функций является критическим аспектом в МО, который включает в себя выбор и извлечение функций. Он используется для уменьшения размерности объемных данных и выявления дискриминантных признаков, которые снижают вычислительные издержки и повышают точность моделей МО.

Введение

В 1959 году Артур Сэмюэл ввел термин «машинное обучение», как «область исследования, которая дает компьютерам возможность учиться без явного программирования». Машинное обучение-это набор инструментов, которые, вообще говоря, позволяют нам «научить» компьютеры выполнять задачи, предоставляя примеры того, как они должны выполняться. Например, предположим, что мы хотим написать программу для различения допустимых сообщений электронной почты и нежелательного спама. Мы могли бы попытаться написать набор простых правил, например, помечать сообщения, содержащие определенные функции (например, слово «продажи» или явно поддельные заголовки). Тем не менее, написание правил, чтобы точно определить, какой текст является действительным, на самом деле может быть довольно трудно сделать хорошо, что приводит либо ко многим пропущенным спам-сообщениям, либо, что еще хуже, ко многим потерянным электронным письмам. Хуже того, спамеры будут активно корректировать способ отправки спама, чтобы обмануть эти стратегии (например, написать «s@les»). Написание эффективных правил — и поддержание их в актуальном состоянии- быстро становится непреодолимой задачей. К счастью, машинное обучение дало решение. Современные спам-фильтры «учатся» на примерах: мы предоставляем алгоритму обучения примеры писем, которые мы вручную обозначили как «ham» (действительная электронная почта) или «spam» (нежелательная электронная почта), и алгоритмы учатся различать их автоматически [1].

I. Основные типы МО

Машинное обучение (далее – МО) – это разнообразная и захватывающая область, и существует множество способов ее определения:

- Представление об Искусственном Интеллекте.
- Представление Программной Инженерии.
- Просмотр Статистики.

Часто методы машинного обучения разбиваются на две фазы:

- Обучение: модель изучается на основе набора обучающих данных.
- Применение: модель используется для принятия решений о некоторых новых тестовых данных [1].

Существует четыре широкие категории задач, которые могут использовать МО, а именно: кластеризация, классификация, регрессия и извлечение правил [2]. В задачах кластеризации цель состоит в том, чтобы сгруппировать сходные данные вместе, одновременно увеличивая разрыв между группами. В то время как в задачах классификации и регрессии цель состоит в том, чтобы сопоставить набор новых входных данных с набором дискретных или непрерывных значений выходных данных соответственно. Задачи извлечения правил существенно отличаются, когда цель состоит в выявлении статистических взаимосвязей в данных. На рис.1 представлены категории задач, которые выигрывают от машинного обучения.

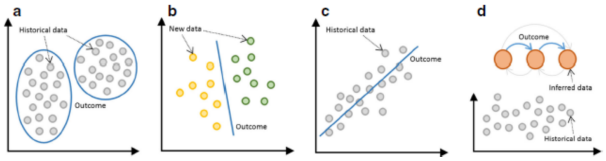

Рис. 1 – a. Кластеризация b. Классификация c. Регрессия d. Извлечение правил

Некоторые из основных типов машинного обучения:

- Контролируемое обучение, при котором обучающие данные помечаются правильными ответами, например «спам» или «ветчина». Двумя наиболее распространенными типами контролируемого обучения являются классификация (где выходы представляют собой дискретные метки, как при фильтрации спама) и регрессия (где выходы являются вещественными).
- Неконтролируемое обучение, при котором нам дают набор немаркированных данных, которые мы хотим проанализировать и обнаружить закономерности внутри. Два наиболее важных примера-это уменьшение размеров и кластеризация.

- Подкрепляющее обучение, при котором агент (например, робот или контроллер) стремится научиться оптимальным действиям, основанным на результатах прошлых действий. Как правило, обучение основано на примерах из обучающих наборов данных. Однако в rl есть агент, который взаимодействует с внешним миром, и вместо того, чтобы обучаться на примерах, он учится, исследуя окружающую среду и используя знания. Действия вознаграждаются или наказываются. Таким образом, обучающие данные в rl представляют собой набор пар «состояние-действие» и вознаграждений (или штрафов). Агент использует обратную связь от окружающей среды, чтобы узнать наилучшую последовательность действий или "политику" для оптимизации совокупного вознаграждения. Например, извлечение правил из данных, которые статистически поддерживаются и не прогнозируются. В отличие от генеративных и дискриминационных подходов, которые близоруки по своей природе, rl может пожертвовать немедленными выгодами ради долгосрочных вознаграждений. Следовательно, rl лучше всего подходит для принятия когнитивных решений, таких как принятие решений, планирование и планирование [1].

После сбора данных они разлагаются на обучающие, проверочные и тестовые наборы данных. Обучающий набор используется для поиска идеальных параметров (например, весов связей между нейронами в нейронной сети (NN)) модели машинного обучения [1].

Общая декомпозиция набора данных может соответствовать 60/20/20% среди обучающих, валидационных и тестовых наборов данных или 70/30% в случае, если валидация не требуется. Эти эмпирические декомпозиции являются разумными для наборов данных, которые не очень велики. Однако в эпоху больших данных, когда набор данных может содержать миллионы записей, допустимы и другие экстремальные декомпозиции, такие как 98/1/1% или 99/0.4/0.1%. Однако важно избегать асимметрии в обучающих наборах данных по отношению к интересующим классам. Это препятствует обучению и обобщению результатов, что приводит к чрезмерной и/или недостаточной подгонке модели. Кроме того, как проверочные, так и тестовые наборы данных должны быть независимы от обучающего набора данных и следовать тому же распределению вероятностей, что и обучающий набор данных. Временная и пространственная устойчивость модели машинного обучения может быть оценена путем предоставления модели обучающим и валидационным наборам данных, которые находятся на временном и пространственном расстоянии.

II. Разработка характеристик

Собранные необработанные данные могут быть зашумленными или неполными. Прежде чем использовать данные для обучения, он должен пройти фазу предварительной обработки, чтобы очистить данные. Еще одним важным шагом перед изучением или обучением модели является извлечение признаков. Эти особенности действуют как дискриминаторы для обучения и вывода.

Разработка функций является критическим аспектом в машинном обучении, который включает в себя выбор и извлечение функций. Он используется для уменьшения размерности объемных данных и выявления дискриминантных признаков, которые снижают вычислительные издержки и повышают точность моделей МО. Извлечение признаков часто представляет собой вычислительно интенсивный процесс получения расширенных или новых признаков из существующих признаков с использованием таких методов, как энтропия, преобразование Фурье и анализ главных компонент (PCA).

Выбор и извлечение функций можно выполнять с помощью таких инструментов, как NetMate [2] и WEKA [3]. Однако в этом случае методы извлечения и отбора ограничены возможностями используемого инструмента. Поэтому для выбора объектов часто используются специализированные методы фильтрации, встраивания и обертывания. Фильтрация отсеивает обучающие данные после тщательного анализа набора данных для выявления нерелевантных и избыточных функций. В отличие от этого, методы, основанные на оболочках, используют итеративный подход, используя различные подмножества функций в каждой итерации для определения оптимального подмножества. Крайне важно тщательно выбрать идеальный набор функций, который обеспечивает баланс между использованием корреляции и уменьшением/устранением чрезмерной подгонки для повышения точности и снижения вычислительных затрат. Поэтому тщательное извлечение и отбор признаков имеет решающее значение для производительности моделей машинного обучения [4].

III. Показатели производительности и валидация модели

После того, как модель МО была построена, крайне важно оценить эффективность модели МО, которая будет описывать, предсказывать или оценивать результаты. Однако важно понимать, что нет никакого способа отличить алгоритм обучения как «лучший», и несправедливо сравнивать частоту ошибок в целом ряде приложений. Показатели производительности могут использоваться для измерения различных аспек-

тов модели, таких как надежность, точность и сложность.

Рассмотрим валидацию точности МО – моделей для прогнозирования. Обычно эта проверка точности проходит анализ ошибок, который вычисляет разницу между фактическими и прогнозируемыми значениями. Напомним, что прогнозирование – это результат использования моделей МО для задач классификации и регрессии. В классификации общие метрики для анализа ошибок основаны на логистической функции, такой как двоичная и категориальная кросс-энтропия – для двоичной и многоклассовой классификации соответственно. В регрессии общепринятыми метриками ошибок являются Средняя абсолютная ошибка (MAE) и Средняя квадратическая ошибка (MSE). Обе регрессионные метрики ошибок игнорируют направление заниженных и завышенных оценок в прогнозах. MAE проще и легче интерпретировать, чем MSE, хотя MSE более полезен для строгого наказания за большие ошибки.

Вычисление функции стоимости обучающих и валидационных наборов данных позволяет диагностировать проблемы производительности из-за высокого смещения или высокой дисперсии. Высокая предвзятость относится к простой модели МО, которая плохо отображает отношения между характеристиками и результатами (недостаточная подгонка). Высокая дисперсия подразумевает модель МО, которая соответствует обучающим данным, но плохо обобщается для прогнозирования новых данных (чрезмерная подгонка). В зависимости от диагностируемой проблемы модель МО может быть улучшена путем возврата к одной из следующих составляющих проектирования: (i) сбор данных для получения большего количества обучающих данных (только для высокой дисперсии), (ii) проектирование признаков для увеличения или уменьшения набора признаков и (iii) обучение модели для построения более простой или более сложной модели или для корректировки параметра регуляризации [1].

После настройки модели машинного обучения для наборов данных обучения и проверки метрики точности для тестового набора данных сообщаются как проверка производительности модели.

В классификации общепринятым показателем, отражающим эффективность модели МО, является точность. Метрика точности определяется как доля истинных предсказаний T для каждого класса $C_i \forall i = 1...N$ среди общего числа предсказаний следующим образом:

$$Точность = \frac{\sum_1^N Tci}{Всего\ прогнозов}$$

Например, рассмотрим классификационную модель, которая предсказывает, должно ли письмо попасть в папку «спам», «входящие» или «продвижение» (т. е. в этом случае точность-это сумма писем, правильно предсказанных как спам, входящие и продвижение, деленная на общее количество предсказанных писем. Однако метрика точности не является надежной, когда данные искажены по отношению к классам. Например, если фактическое количество спама и рекламных писем очень мало по сравнению с входящими письмами, простая модель классификации, которая предсказывает каждое письмо как входящее, все равно достигнет высокой точности. Чтобы устранить это ограничение, рекомендуется использовать метрики, полученные из матрицы, как показано на рис. 2.

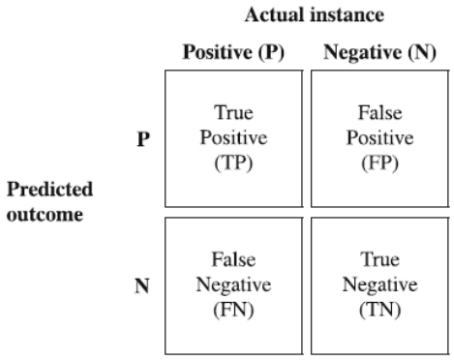

Рис. 2 – Матрица

Учтите, что каждая строка в матрице представляет собой прогнозируемый результат, а каждый столбец –фактический экземпляр. Таким образом, Истинный позитив (TP) – это пересечение между правильно предсказанными исходами для фактических положительных случаев. Точно так же Истинно Отрицательный (TN) – это когда модель классификации правильно предсказывает фактический отрицательный экземпляр. В то время как ложноположительные (FP) и ложноотрицательные (FN) описывают неправильные предсказания для отрицательных и положительных фактических случаев соответственно. Заметим, что TP и TN соответствуют истинным предсказаниям для положительных и отрицательных классов соответственно. Поэтому метрика точности также может быть определена в терминах матрицы:

$$Точность = \frac{TP+TN}{TP+TN+FP+FN}$$

Матрица на рис. 2 работает для бинарной модели классификации. Поэтому его можно использовать в многоклассовой классификации, построив матрицу для конкретного класса. Это достигается путем преобразования задачи многоклассовой классификации в несколько подзадач бинарной классификации с использованием стратегии «один против остальных». Например,

в многоклассовой классификации электронной почты матрица для класса спама устанавливает положительный класс как спам, а отрицательный – как остальные классы электронной почты (т. е. inbox and promotion), получение бинарной модели классификации спама, а не спам-почты [2].

Следовательно, Истинная положительная скорость (TPR), описывающая количество правильных предсказаний, выводится из матрицы следующим образом:

$$TPR\,(Recall) = \frac{TP}{TP + FN}$$

Обратная, Ложноположительная скорость (FPR) - это отношение неверных прогнозов и определяется как:

$$FPR = \frac{FP}{FP + TN}$$

Аналогично, Истинная отрицательная ставка (TNR) и Ложная отрицательная ставка (FNR) используются для вывода количества правильных и неправильных отрицательных прогнозов соответственно. Следует отметить, что классификационную модель с отрицательным градиентом на ROC-кривой можно легко улучшить, перевернув предсказания из модели или поменяв местами метки реальных экземпляров. Таким образом, можно сравнить несколько классификационных моделей для одной и той же задачи и получить представление о различных условиях, при которых одна модель превосходит другую. Таким образом, точность модели МО может быть формально определена как частота правильных предсказаний для фактических положительных примеров:

$$Точность = \frac{TP}{TP + FP}$$

Компромисс между значениями отзыва и точности позволяет настраивать параметры классификационных моделей и достигать желаемых результатов. F-мера позволяет анализировать компромисс между отзывом и точностью, предоставляя среднее гармоническое значение, в идеале 1, этих метрик:

$$F - = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Коэффициент вариации (CV) - это еще один показатель точности, особенно используемый для отчетности о производительности неконтролируемых моделей, которые проводят классификацию с использованием кластеров (или состояний). CV-это стандартизированная мера дисперсии, представляющая внутрикластерное (или внутригосударственное) сходство. Более низкий CV подразумевает небольшую вариабельность каждого результата по отношению к среднему значению соответствующего кластера. Это означает более высокое внутрикластерное сходство и более высокую точность классификации.

IV. Алгоритмы машинного обучения

Линейная регрессия наименьших квадратов. При линейной регрессии задача состоит в том, чтобы провести линию через распределение, которое ближе всего к большинству точек в обучающем наборе. В простой линейной регрессии линия регрессии минимизирует сумму расстояний от отдельных точек, то есть сумму «Квадрата невязок».

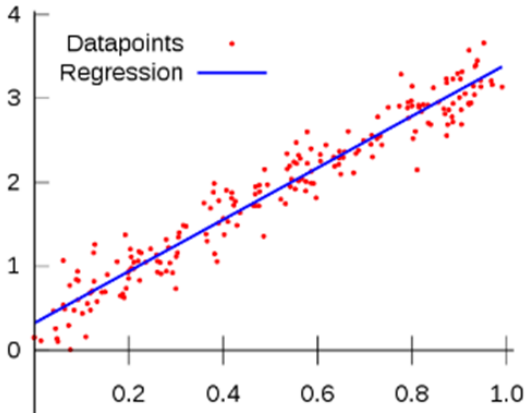

Рис. 3 – Пример линейной регрессии

Линейная регрессия используется для прогнозирования количественного отклика Y от предикторной переменной X. Линейная регрессия производится с предположением, что существует линейная зависимость между X и Y. Математически мы можем записать линейную зависимость в виде:

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + \cdots + b_n * x_n$$

где, y – это ответ, значения b называются коэффициентами модели. Эти значения «изучаются» на этапе подгонки/обучения модели. b_0 – это перехват, b_1 – коэффициент для x_1 (первый признак), b_n – коэффициент для x_n (n-й признак).

V. Алгоритмы машинного обучения

Машина опорных векторов (SVM) [4] - это контролируемый алгоритм машинного обучения, который может быть использован как для задач классификации, так и для задач регрессии. В SVM мы строим точки данных в N-мерном пространстве, где N-число объектов, и находим гиперплоскость для дифференциации точек данных.

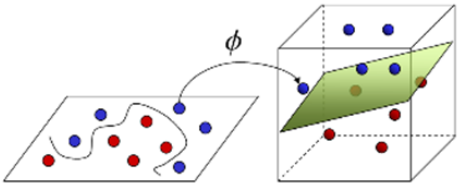

Рис. 4 – Гиперплоскость SVM

Чтобы понять основную идею, мы можем предположить следующий сценарий: на рисунке 5 (a) у нас есть три гиперплоскости (A, B и C). Теперь определите правильную гиперплоскость для классификации звезды и круга. Нам нужно запомнить правило большого пальца, чтобы определить правильную гиперплоскость: «Выберите гиперплоскость, которая лучше разделяет два класса». В этом сценарии гиперплоскость «Б» превосходно выполнила эту работу.

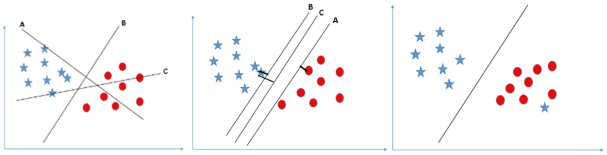

Рис. 5 – Различные сценарии SVM

В (b) у нас есть три гиперплоскости (A, B и C), и все они хорошо разделяют классы. Теперь мы должны определить правильную гиперплоскость. Здесь максимизация расстояний между ближайшими точками данных (любым классом) и гиперплоскостью поможет нам выбрать правильную гиперплоскость. Это расстояние называется маржей. В SVM легко иметь линейную гиперплоскость между этими двумя классами. Но возникает еще один животрепещущий вопрос: нужно ли добавлять эту функцию вручную, чтобы иметь гиперплоскость? Нет, у SVM есть техника, называемая трюком ядра. Это функции, которые берут низкоразмерное входное пространство и преобразуют его в более высокое размерное пространство, то есть преобразуют неразделимую задачу в сепарабельную задачу, эти функции называются ядрами.

VI. Заключение

Выводя итоги, можно сказать, что для обнаружения аномалий может быть использован алгоритм кластеризации K-mean, который определяет два центроида для нормального и аномального поведения. Таким образом, можно сравнить несколько классификационных моделей для одной и той же задачи и получить представление о различных условиях, при которых одна модель превосходит другую.

Список литературы

1. Raouf Boutaba, Mohammad A. Salahuddin, Noura Limam1, Sara Ayoubi1, Nashid Shahriar, Felipe Estrada-Solano and Oscar M. Caicedo, “A comprehensive survey on machine learning for networking: evolution, applications and research opportunities”, Journal of Internet Services and Applications, 2018
2. Arndt D. HOW TO: Calculating Flow Statistics Using NetMate. 2016. https://dan.arndt.ca/nims /calculating- flow-statistics -using- netmate/.
3. Machine Learning Group, University of Waikato. WEKA. 2017. http:// www.cs.waikato.ac.nz/ml/weka/.
4. Brill E, Lin JJ, Banko M, Dumais ST, Ng AY, et al. Data-intensive question answering. In: TREC, vol. 56. 2001. p. 90.

Балатай Алия Амангелдікызы, докторант Евразийский национальный университет им. Л.Н. Гумилева, г.Нур-Султан, Республика Казахстан

Сагнаева Сауле Кайроллиевна, к.ф.-м.н., доцент