УДК 519.7

ОСНОВНЫЕ ЭЛЕМЕНТЫ МАШИННОГО ОБУЧЕНИЯ

Сунчалин А.М.

Финансовый университет при Правительстве РФ, Москва, e-mail: Asunchalin@fa.ru

Машинное обучение – это изучение компьютерных алгоритмов, которые автоматически улучшаются с приобретением опыта. Машинное обучение подразумевает, что компьютеры обнаруживают, как они могут выполнять задачи, не будучи явно запрограммированы для этого. Для простых задач, назначаемых компьютерам, можно запрограммировать алгоритмы, указывающие машине, как выполнить все шаги, необходимые для решения данной проблемы; со стороны компьютера обучение не требуется. Для более сложных задач человеку может быть сложно вручную создать необходимые алгоритмы. На практике может оказаться более эффективным помочь машине разработать собственный алгоритм, а не просто программистам-людям определять каждый необходимый шаг. Методы машинного обучения используют различные подходы, чтобы помочь компьютерам научиться выполнять задачи, где нет полностью удовлетворительного алгоритма. Подходы к машинному обучению делятся на три широкие категории. 1. Обучение с учителем. Компьютер представлен примерами входных данных и их желаемых результатов, предоставленных «учителем»; цель состоит в том, чтобы выучить общее правило, которое отображает изменение входных данных в выходные. 2. Обучение без учителя. 3. Обучение с подкреплением. Компьютерная программа взаимодействует с динамической средой, в которой она должна выполнять определенную цель. При перемещении по проблемному пространству программе предоставляется обратная связь, аналогичная вознаграждениям, которую она пытается максимизировать. В статье рассматриваются подходы к машинному обучению.

Ключевые слова: машинное обучение, алгоритмы, моделирование, искусственный интеллект

BASIC ELEMENTS OF MACHINE LEARNING

Sunchalin A.M.

Financial University under the Government of the Russian Federation, Moscow, e-mail: Asunchalin@fa.ru

Machine Learning (ML) is the study of computer algorithms that automatically improve with experience. Machine learning implies that computers discover how they can perform tasks without being explicitly programmed to do so. For simple tasks assigned to computers, you can program algorithms to tell the machine how to complete all the steps necessary to solve the problem; from the computer side, no training is required. For more complex tasks, it can be difficult for a person to manually create the necessary algorithms. In practice, it may be more effective to help a machine develop its own algorithm rather than just human programmers to define each necessary step. Machine learning techniques use a variety of approaches to help computers learn how to perform tasks where there is no completely satisfactory algorithm. Machine learning approaches fall into three broad categories. 1. Teaching with a teacher. The computer is represented by examples of inputs and their desired results provided by a «teacher», and the goal is to learn a general rule that maps input to output. 2. Learning without a teacher. 3. Reinforcement learning. A computer program interacts with a dynamic environment in which it must fulfill a specific purpose. As it navigates through the problem space, the program is provided with feedback similar to the rewards it is trying to maximize. The article discusses approaches to machine learning.

Keywords: machine learning, algorithms, modeling, artificial Intelligence

Машинное обучение рассматривается как подмножество искусственного интеллекта. Алгоритмы машинного обучения строят математическую модель на основе выборочных данных, известных как «обучающие данные», для того, чтобы делать прогнозы или решения, не будучи явно запрограммированными для этого.

Машинное обучение тесно связано с вычислительной статистикой, которая фокусируется на прогнозировании с использованием компьютеров [1]. Изучение математической оптимизации поставляет методы, теории и способы применения в сфере машинного обучения. Интеллектуальный анализ данных является смежной областью исследования, сосредоточенной на поисковом анализе данных посредством обучения без учителя. Применительно к бизнес-задачам машинное обучение также называется предиктивной (предсказательной) аналитикой [2].

Целью исследования является познакомить читателя с алгоритмами машинного обучения. Для этого рассмотрим основной набор инструментов для машинного обучения.

1. Данные представляют собой входные переменные, необходимые для формирования прогноза. Данные поступают во многих формах, включая структурированные и неструктурированные данные.

2. Второе отделение панели инструментов содержит инфраструктуру, которая состоит из платформ и инструментов для обработки данных. Можно использовать библиотеки машинного обучения. Это набор предварительно скомпилированных программ, часто используемых в машинном обучении.

3. Алгоритмы. Теперь, когда среда машинного обучения настроена, можно импортировать данные непосредственно из файла CSV. Можно найти сотни интересных наборов данных в формате CSV от kaggle.com и применить к ним простые контролируемые алгоритмы обучения, такие как линейная регрессия, логистическая регрессия, деревья решений и k-ближайших соседей.

4. Визуализация. Визуальное сообщение, передаваемое с помощью графиков, диаграмм рассеяния, коробчатых диаграмм и представления чисел в формах, позволяет быстро и легко передавать информацию.

5. Большие данные. Большие данные используются для описания набора данных, который из-за своей ценности, разнообразия, объема и скорости не поддается обычным методам обработки, и для человека будет невозможным обрабатывать его без помощи продвинутой машины. Большие данные не имеют точного определения с точки зрения размера или общего количества строк и столбцов.

Обзор алгоритмов

Для анализа больших наборов данных работают со множеством продвинутых алгоритмов, включая модели Маркова, опорные векторные машины и Q-Learning. Семейство алгоритмов, которое чаще всего используют, – это нейронные сети [3].

TensorFlow – это библиотека для машинного обучения, предназначенная для глубокого обучения нейронных сетей, поскольку она поддерживает многочисленные передовые методы, включая автоматическое исчисление для обратного распространения / градиентного спуска.

Популярные альтернативные библиотеки нейронных сетей включают Torch, Caffe и быстрорастущие Keras. Написанная на Python Keras – это библиотека глубокого обучения с открытым исходным кодом, которая работает поверх TensorFlow, Theano и других сред и позволяет пользователям быстро экспериментировать с меньшим количеством строк кода. Keras является минимальной, модульной и быстрой в установке, но менее гибкой по сравнению с TensorFlow и другими библиотеками.

Наборы данных почти всегда требуют какой-либо предварительной обработки. Очистка данных (Data Scrubbing) – это технический процесс уточнения набора данных, чтобы сделать его более работоспособным. Она может включать изменение и иногда удаление неполных, неправильно отформатированных, нерелевантных или дублированных данных. Это также может повлечь за собой преобразование текстовых данных в числовые значения и изменение функций [4–6].

Работа с отсутствующими данными никогда не является желаемой ситуацией. Представьте, что при разгадке головоломки, которую вы обнаружили, 5% ее частей отсутствуют. Пропущенные значения в наборе данных могут быть в равной степени разочаровывающими и в конечном итоге способны повлиять на ваш анализ и окончательные прогнозы. Однако существуют стратегии, позволяющие минимизировать негативное влияние отсутствующих данных [7–8].

Одним из подходов является аппроксимация (научный метод, состоящий в замене одних объектов другими, в каком-то смысле близкими к исходным, но более простыми) пропущенных значений с использованием значения режима. Режим представляет собой единственное наиболее распространенное значение переменной, доступное в наборе данных. Это лучше всего работает с категориальными и двоичными типами переменных.

Второй подход к управлению отсутствующими данными заключается в аппроксимации отсутствующих значений с использованием медианного (усредненного) значения, которое принимает значение, расположенное в середине набора данных. Это лучше всего работает с целыми числами и непрерывными переменными (десятичными числами).

В крайнем случае строки с пропущенными значениями могут быть удалены полностью. Очевидными недостатками этого подхода являются наличие меньшего количества данных для анализа и потенциально меньшее количество полных результатов.

После того как очистили набор данных, следующая задача – разделить данные на два сегмента для тестирования и обучения. Очень важно не проверять вашу модель с теми же данными, которые вы использовали для обучения. Соотношение двух разделений должно быть примерно 70/30 или 80/20. Это означает, что ваши тренировочные данные должны составлять от 70% до 80% строк в наборе данных, а остальные от 20% до 30% строк – это тестовые данные. Важно разделить данные по строкам, а не по столбцам.

Перед разделением данных важно рандомизировать все строки в наборе данных. Это помогает избежать смещения в модели, поскольку исходный набор данных может быть упорядочен последовательно в зависимости от времени, когда он был собран, или какого-либо другого фактора.

После рандомизации данных можно приступить к разработке модели и применить ее к данным обучения. Остальные 30% данных откладываются в сторону и резервируются для проверки точности модели.

В случае контролируемого обучения модель разрабатывается путем подачи на машину данных обучения и ожидаемого результата (Y). Машина может анализировать и распознавать взаимосвязи между признаками (X), обнаруженными в данных обучения, для расчета окончательного результата (Y).

Следующим шагом является измерение того, насколько эффективно модель работает. Распространенным подходом к анализу точности прогнозирования является мера, называемая средней абсолютной ошибкой, которая проверяет каждый прогноз в модели и предоставляет среднюю оценку ошибки для каждого прогноза.

Как только модель сможет адекватно предсказать значения тестовых данных, она будет готова к использованию в режиме реального времени. Если модель не может точно предсказать значения из тестовых данных, необходимо проверить, были ли данные обучения и тестов правильно рандомизированы. В качестве альтернативы может потребоваться изменить гиперпараметры модели.

Каждый алгоритм имеет гиперпараметры – это настройки алгоритма. Проще говоря, эти настройки управляют и влияют на то, как быстро модель изучает шаблоны и какие шаблоны идентифицировать и анализировать.

Хотя разделение данных обучения / теста может быть эффективным при разработке моделей на основе существующих данных, остается вопрос о том, будет ли модель работать с новыми данными. Если ваш существующий набор данных слишком мал, чтобы построить точную модель, или раздел данных обучения / теста не подходит, это может привести к плохим оценкам производительности в неисследованной среде.

Существует эффективный обходной путь для этой проблемы. Вместо того чтобы разбивать данные на два сегмента (один для обучения и один для тестирования), можно реализовать перекрестную проверку (cross validation).

Перекрестная проверка может быть выполнена двумя основными методами. Первый – это исчерпывающая перекрестная проверка (exhaustive cross validation), которая включает в себя поиск и тестирование всех возможных комбинаций для разделения исходного образца на тренировочный и тестовый набор. Альтернативным и более распространенным методом является неисчерпывающая перекрестная проверка, известная как k-кратная проверка (k-fold validation). Метод проверки k-кратности включает в себя разбиение данных на k назначенных сегментов и резервирование одного из этих сегментов для проверки модели обучения в каждом раунде [9–10].

Кластеризация

Одним из полезных подходов к анализу информации служит выявление кластеров данных, которые имеют сходные атрибуты [11–12].

Кластерный анализ подпадает под понятия контролируемого обучения и неконтролируемого обучения. В качестве контролируемой методики обучения кластеризация используется для классификации новых точек данных в существующие кластеры через k-ближайших соседей (k-NN), а в качестве неконтролируемой методики обучения применяется кластеризация для идентификации дискретных групп точек данных посредством кластеризации k-средних. Хотя существуют и другие способы кластеризации, эти два алгоритма, как правило, наиболее популярны как в машинном обучении, так и в интеллектуальном анализе данных [5].

Простейшим алгоритмом кластеризации является k-ближайших соседей (k-NN) – контролируемая методика обучения, используемая для классификации новых точек данных на основе отношения к близлежащим точкам данных.

k-NN похож на систему голосования или конкурс популярности. Подумайте об этом как о новом ребенке в школе, который выбирает группу одноклассников для общения, основываясь на пяти одноклассниках, которые сидят рядом с вами. Среди пяти одноклассников трое – фанаты, один – фигурист, а другой – диск-жокей. Согласно k-NN, вы бы предпочли тусоваться с фанатами, основываясь на их численном преимуществе. Давайте рассмотрим другой пример.

Диаграмма рассеяния позволяет вычислить расстояние между любыми двумя точками данных. Точки данных на диаграмме рассеяния уже были разделены на две группы. Затем на график добавляется новая точка данных, класс которой неизвестен. Мы можем предсказать категорию новой точки данных, основываясь на ее отношении к существующим точкам данных.

Метод k-средних кластеризации. Как популярный алгоритм обучения без учителя, k-средняя кластеризация пытается разделить данные на k отдельных групп и эффективна при раскрытии базовых шаблонов данных. Примерами потенциальных груп-

пировок являются виды животных, клиенты с похожими характеристиками и сегментация рынка жилья. Алгоритм кластеризации k работает, сначала разбивая данные на число k кластеров, где k представляет количество кластеров, которые вы хотите создать. Если решили разделить набор данных на три кластера, то, например, для k будет установлено значение 3. Таким образом, исходные (некластеризованные) данные были преобразованы в три кластера (k равно 3).

Если бы мы установили k на 4, из набора данных был бы получен дополнительный кластер, чтобы произвести четыре кластера.

При настройке k важно указать правильное количество кластеров. В общем, с увеличением k кластеры становятся меньше, а дисперсия падает. Однако недостатком является то, что соседние кластеры становятся менее отличными друг от друга при увеличении k.

Если вы установите k равным количеству точек данных в вашем наборе данных, каждая точка данных автоматически преобразуется в автономный кластер. И наоборот, если вы установите k в 1, то все точки данных будут считаться однородными и создавать только один кластер. Установка k на любое из крайних значений не даст какой-либо достойной информации для анализа.

Более простой и нематематический подход к настройке k – применение знаний предметной области.

Регрессионный анализ

Рассмотрим один из алгоритмов машинного обучения – регрессионный анализ. Он представляет собой контролируемую технику обучения, используемую для поиска наилучшей линии тренда в целях описания набора данных.

Самый простой метод регрессионного анализа – это линейная регрессия, в которой для описания набора данных используется прямая линия.

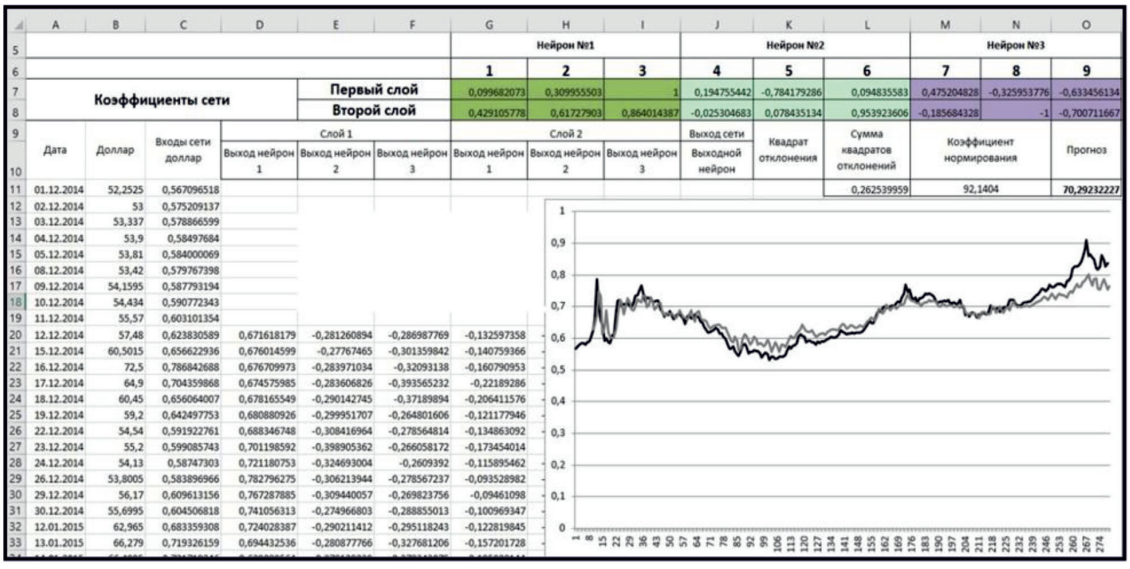
Линейная регрессия включает в себя прямую линию, которая разделяет точки данных на диаграмме рассеяния. Цель линейной регрессии – разделить данные таким образом, чтобы минимизировать расстояние между линией регрессии и всеми точками данных на диаграмме рассеяния.

В целом, чем ближе точки данных находятся к линии регрессии, тем точнее окончательный прогноз. Если существует высокая степень отклонения между точками данных и линией регрессии, наклон даст менее точные прогнозы.

Логистическая регрессия. Логистическая регрессия принимает сигмоидную функцию для анализа данных и прогнозирования дискретных классов, которые существуют в наборе данных.

Логистическая регрессия обычно используется для двоичной классификации, чтобы спрогнозировать, например, два дискретных класса. Логистическая регрессия применяется во многих областях, включая обнаружение мошенничества, диагностику заболеваний, обнаружение чрезвычайных ситуаций и т.д.

Логистическая регрессия с более чем двумя значениями результата известна как полиномиальная логистическая регрессия.



				Нейрон №1			Нейрон №2			Нейрон №3		
				1	2	3	4	5	6	7	8	9
Коэффициенты сети			Первый слой	0,099682073	0,309555503	1	0,194755442	-0,784179286	0,094835583	0,475204828	-0,325953776	-0,633456134
			Второй слой	0,429105778	0,61727903	0,864014387	-0,025304683	0,078435134	0,953923606	-0,185684328	-1	-0,700711667
Дата	Доллар	Входы сети доллар	Слой 1			Слой 2			Выход сети	Квадрат отклонения	Сумма квадратов отклонений	Коэффициент нормирования		Прогноз
			Выход нейрон 1	Выход нейрон 2	Выход нейрон 3	Выход нейрон 1	Выход нейрон 2	Выход нейрон 3	Выходной нейрон		0,262539959	92,1404		70,29232227
01.12.2014	52,2525	0,567096518												
02.12.2014	53	0,575209137												
03.12.2014	53,337	0,578866599												
04.12.2014	53,9	0,58497684												
05.12.2014	53,81	0,584000069												
08.12.2014	53,42	0,579767398												
09.12.2014	54,1595	0,587793194												
10.12.2014	54,434	0,590772343												
11.12.2014	55,57	0,603101354												
12.12.2014	57,48	0,623830589	0,671618179	-0,281260894	-0,286987769	-0,132597358								
15.12.2014	60,5015	0,656622936	0,676014599	-0,27767465	-0,301359842	-0,140759366								
16.12.2014	72,5	0,786842688	0,676709973	-0,283971034	-0,32093138	-0,160790953								
17.12.2014	64,9	0,704359868	0,674575985	-0,283606826	-0,393565232	-0,22189286								
18.12.2014	60,45	0,656064007	0,678165549	-0,290142745	-0,37189894	-0,206411576								
19.12.2014	59,2	0,642497753	0,680880926	-0,299951707	-0,264801606	-0,121177946								
22.12.2014	54,54	0,591922761	0,688346748	-0,308416964	-0,278564814	-0,134863092								
23.12.2014	55,2	0,599085743	0,701198592	-0,398905362	-0,266058172	-0,173454014								
24.12.2014	54,13	0,58747303	0,721180753	-0,324693004	-0,2609392	-0,115895462								
26.12.2014	53,8005	0,583896966	0,782796275	-0,306213944	-0,278567237	-0,093528982								
29.12.2014	56,17	0,609613156	0,767287885	-0,309440057	-0,269823756	-0,09461098								
30.12.2014	55,6995	0,604506818	0,741056313	-0,274966803	-0,288855013	-0,100969347								
12.01.2015	62,965	0,683359308	0,724028387	-0,290211412	-0,295118243	-0,122819845								
13.01.2015	66,279	0,719326159	0,694432536	-0,280877766	-0,327681206	-0,157201728								

Прогноз USD/RUB на основе нейронной сети

При выполнении логистической регрессии следует помнить два момента: в данных не должно быть пропущенных значений и все переменные не зависят друг от друга. Также должно быть достаточно данных для каждого исходящего значения, чтобы обеспечить высокую точность. Хорошей отправной точкой будет наличие примерно 30–50 точек данных для каждого результата, т.е. 60–100 общих точек данных для бинарной логистической регрессии.

Приведем пример создания экономической модели на основе логистической регрессии. Как правило, логистическую функцию используют в нейронных сетях. Нейронные сети на сегодняшний день являются наиболее популярными моделями для прогнозирования финансовых временных рядов. С их помощью можно прогнозировать как высоковолатильные акции, фьючерсы, облигации, так и макроэкономические показатели, к которым относятся: курс рубля, цена на золото, нефть и др.

Предположим, что курс USD/RUB зависит не только от собственных тенденций, но и от цен на нефть и золото. Рассмотрим однослойную нейронную сеть. Подадим на входы различных нейронов данные за предшествующие три дня (1-й нейрон: стоимость нефти BRENT; 2-й нейрон: стоимость унции золота; 3-й нейрон: курс USD/RUB). На рисунке представлен прогноз на основе многофакторной нейронной сети.

Список литературы

1. Theobald O. Machine learning for absolute beginners: a plain English introduction. Scatterplot Press, 2017. P. 157.

2. Иванюк В.А., Арутюнов А.Л., Цвиркун А.Д. Разработка инструментальных средств прогнозирования в социально-экономических системах // Препринт доклада. ИПУ РАН. 2012. С. 43.

3. Krose B.P. van der Smagt An introduction to neural networks. English Edition, November. 1996. P. 136.

4. Philo J.R. Kevin Kelly. The inevitable: Understanding the 12 technological forces that shape our future. New York: Penguin, 2016. P. 297.

5. Samuel A.L. Some studies in machine learning using the game of checkers. IBM Journal of research and development. 2000. Vol. 44. № 1.2. P. 206–226.

6. Sun S., Cao Z., Zhu H., ZhaoJ. A survey of optimization methods from a machine learning perspective. IEEE transactions on cybernetics. 2019. Vol. 50. № 8. P. 3668–3681.

7. Иванюк В.А., Андропов К.Н., Егорова Н.Е. Методы оценки эффективности и оптимизации инвестиционного портфеля // Фундаментальные исследования. 2016. № 3–3. С. 575–578.

8. Станик Н.А., Иванюк В.А., Попов В.Ю. Феномен пузырей на финансовых рынках // Современные проблемы науки и образования. 2012. № 6. [Электронный ресурс]. URL: http://www.science-education.ru/ru/article/view?id=7474 (дата обращения: 28.04.2021).

9. Athey S., Imbens G.W. Machine learning methods that economists should know about. Annual Review of Economics. 2019. Vol. 11. P. 685–725.

10. Sra S., Nowozin S., Wright S.J. Optimization for machine learning. Mit Press, 2012. P. 494.

11. Mitchell J.B.O. Machine learning methods in chemoinformatics. Wiley Interdisciplinary Reviews: Computational Molecular Science. 2014. Vol. 4. № 5. P. 468–481.

12. Komura D., Ishikawa S. Machine learning methods for histopathological image analysis. Computational and structural biotechnology journal. 2018. Vol. 16. P. 34–42.