ФІЗИКО-МАТЕМАТИЧНІ НАУКИ

УДК 002.5

ПОБУДОВА ПОШУКОВИХ СИСТЕМ НА ОСНОВІ МЕТОДІВ МАШИННОГО НАВЧАННЯ

Семчишин О.М., Карабін О.Й.
Тернопільський національний педагогічний університет
імені Володимира Гнатюка

Розглянуто теоретичні основи пошукових систем на основі методів машинного навчання; з'ясовано за допомогою яких мов програмування і їх бібліотек можна реалізувати; визначено аспекти задіяння методів машинного навчання над вже існуючими; використано спрямування обґрунтувати роботу пошукових систем з одним запитом та провести їх порівняльний аналіз. На організаційно-діяльнісному етапі визначено етапи роботи пошукових систем та їх можливості для одного запиту. Перспективно-пошуковий етап передбачає використання більшого спектра бібліотек для реалізації найкращого запиту.
Ключові слова: освітній процес, машинне навчання, штучний інтелект, алгоритми, бібліотека машинного навчання, середовище програмування, мови програмування.

Постановка проблеми. Щоденно пошукова система Google надає безліч відповідей на запити. Четвертина з них є неповторюваними. Тому, неможливо написати для пошукової системи таку програму, в якій передбачено кожен запит, і для кожного запиту відому кращу відповідь. Пошукова система повинна вміти приймати рішення самостійно, тобто, сама вибирати з мільйонів документів той, який найкраще відповідає користувачеві. Для цього потрібно навчити її навчатися. Завдання навчити машину навчатися існує не тільки в пошукових технологіях. Без машинного навчання (далі МН) неможливо, наприклад, розпізнавати рукописний текст або мову. В результаті машинного навчання комп'ютер може демонструвати поведінку, яку в нього не було явно закладено. Пошукова система повинна навчитися будувати правило, яке визначає достовірність пошукового запиту. Пошукова машина повинна аналізувати властивості веб-сторінок і пошукових запитів.

Аналіз останніх досліджень і публікацій. Теоретичні засади машинного навчння розкриваються у дослідженнях вітчизняних науковців – М.С. Лавренюк, О.О. Марченко, Є.А. Мельников, О.П. Мосалов, А.О. Ніконенко, О.М. Новіков, Д.В. Прохоров, В.Г. Редько, Т.В. Россада та ін.; зарубіжних науковців – R. Sutton, A. Barto, T. Prescott та ін.

Виділення не вирішених раніше частин загальної проблеми. На сьогодні немає такої пошукової системи, яка б дала найкращий варіант відповіді, вона не може застосувати всі можливі характеристики вибірки. Потрібно врахувати топографічні фактори та опиратися на історію пошуків задіюючи різні завдання машинного навчання.

Головною метою цієї роботи є розгляд аспектів побудови пошукових систем на основі методів машинного навчання.

Виклад основного матеріалу. Нині інформаційні технології – невід'ємна частина інформатизації суспільства. Використовуючи сучасні персональні комп'ютери, можна інтенсифікувати процес навчання, зробити його більш наочним і динамічним, встановлювати правила опрацювання інформації. Саме функції автоматизації дозволять оптимізувати роботу користувача – поглиблене вивчення нового матеріалу і засвоєння пройденого, за допомогою методів машинного навчання. Це забезпечить якісне та ефективне опанування навчального матеріалу.

Завдання машинного навчання поділяють на категорії або навчальний «сигнал» або «зворотний зв'язок»:

1. Навчання з учителем (кероване навчання, *supervised learning*). Метою є навчання загального правила, яке відображає входи на виходи. Комп'ютерові представляють приклади входів та бажаних виходів, задані «вчителем». В окремих випадках вхідний сигнал може бути доступним лише частково, або бути обмеженим особливим зворотним зв'язком.
2. Напівавтоматичне навчання (*semi-supervised learning*): комп'ютерові дають лише неповний тренувальний сигнал: тренувальний набір, в якому відсутні деякі цільові виходи.
3. Активне навчання (*active learning*): комп'ютер може отримувати тренувальні мітки лише для обмеженого набору екземплярів, а також має оптимізувати свій вибір об'єктів для отримування міток. За інтерактивного застосування, вони можуть надаватися для мічення користувачеві.
4. Навчання з підкріпленням (*reinforcement learning*): тренувальні дані надаються лише як зворотний зв'язок на дії програми в динамічному середовищі, як при керуванні автомобілем, або грі в гру з опонентом.
5. Навчання без учителя (спонтанне навчання, *unsupervised learning*): алгоритмові навчання

не дається міток, залишаючи його самому знаходити структуру в своєму вході. Навчання без учителя може бути метою саме по собі або засобом досягнення мети [6].

Вони широко використовуються, і їх використання є індивідуальним. Для реалізації проблем машинного навчання використовують алгоритми. Алгоритми мають бути реалізовані певною мовою програмування, в певному середовищі, розраховані на певний вид продукту тощо. Якщо програміст розуміє принцип роботи певного алгоритму, то будь-яка мова програмування може підійти для реалізації того чи іншого методу і вирішення прикладної задачі. Звісно, це буде потребувати додаткового часу, можливо ефективність реалізації буде гіршою за аналоги. Але розуміння логіки процесу робить вибір мови чи сфери застосування алгоритму – необмеженими [5].

У машинному навчанні весь освітній процес, як правило, зводиться до мінімізації помилки. Тобто, якщо система мінімізувала помилку до досить низької величини, вважаємо, що її можна використовувати [6]. Одним із прикладів процесу мінімізації помилки є нейронні мережі. Нейронні мережі займаються поетапним проектування вхідних векторів, щоб на виході отримати правильний вектор очікуваної розмірності [1].

Система машинного навчання навчається на прикладах для того, щоб їх генералізувати, тобто знайти приховані закономірності між вхідними та вихідними даними так, щоб на тестовій вибірці демонструвати хороші результати. Якість результатів досить сильно залежить від того, які ознаки оберемо для описання і яким чином їх будемо описувати. Є декілька способів описання цих ознак.

Наприклад, є бінарні ознаки: якщо у нас є певні характеристики, які можуть бути описаними відповіддю на питання так або ні, є певна характеристика або немає, 1 або 0, значить можемо їх описувати певним бінарним вектором. Цих ознак може бути декілька, тобто, наприклад, у об'єкта може бути 10 або 100 000 ознак в залежності від прикладної задачі.

Номінальні ознаки: один із прикладів – це координати точки, які можуть бути від нуля до плюс нескінченності в залежності від того, який у нас тип даних. Координати точки (–5; 16) – це будуть дві номінальні ознаки –x та y.

Порядкові ознаки – це ознака позиції, наприклад, в списку задачі ранжування.

Кількісні ознаки – припустимо, описання кількості доходу фізичної особи або зріст, або будь-які інші властивості, які можна описати кількісною характеристикою.

Нейронні мережі, як один із видів машинного навчання, на вхід завжди отримують певний вектор ознак, на виході надають вектор відповідей. Якщо говорити про нейронні мережі, слід пам'ятати, що все, що робить нейронна мережа, це деформує багатовимірний простір ознак таким чином, щоб внаслідок всіх деформацій приклади наблизились до очікуваних зон у вихідному просторі.

Розмірність простору, в якому задаємо ознаки, як правило відрізняється від розмірності простору, в якому очікуємо відповіді. Припустимо, простір відповідей може бути одновимірний або двовимірний в залежності від задачі.

Окрім описання об'єкту також описуємо відповіді, тобто мітки, які ставимо до об'єктів, які хочемо отримати внаслідок роботи системи. Мітки також можуть бути бінарні, тобто 1 або 0, наявна або відсутня, це можуть бути індекси класів, до яких належить об'єкт.

Задачі класифікації теж між собою дещо відрізняються, оскільки існують класифікації на класи, які між собою не перетинаються, наприклад, класифікація «котики чи собачки». Або це може бути класифікація на набір класів, які перетинаються, наприклад, класифікація наявних об'єктів на зображенні. Одночасно можуть бути в наявності кілька різних об'єктів і фотографії можуть належати до декількох класів одночасно.

У задачах регресії міткою може виступати будь-який номінальний вектор. Наприклад, можемо сконструювати функцію, яка одну двовимірну точку проектує на виході у якусь іншу двовимірну точку.

У задачах ранжування кінцевою метою є впорядкована послідовність результатів, які отримаємо.

У процесі навчання будь-якої системи машинного навчання ключовим моментом у досягненні бажаного результату є вибір правильного функціоналу якості.

Функціонал якості – це певна функція, яка видає нам рівень помилки, яку робить система. У процесі навчання моделі ми намагаємося мінімізувати помилку, яку видає система. Тобто, є набір прикладів, набір очікуваних результатів. Система спочатку видає результати, які дуже відрізняються від того що ми очікуємо. Відповідно по кожному із прикладів можемо вирахувати помилку і скорегувати систему таким чином, щоб помилка була менше. У залежності від того, яку функцію втрат функціоналу якості виберемо, буде залежати яким чином буде навчатися система.

Функціонал якості в залежності від задачі відрізняється. Наприклад, для задачі регресії досить доцільне використання так званої функції квадратичної помилки, коливіднімаємо від очікуваного результату результат, який видала система і беремо квадрат цієї різниці. Також можна обрати абсолютну помилку, наприклад, в задачі мультикласової класифікації функцією помилки може бути – віднесла система до правильного класу чи не віднесла. Тобто 1 або 0 [4].

Попри те, що після етапу формулювання системи рівнянь методи вирішення цієї системи не такі очевидні і описання математичного апарату такого рішення потребує значної кількості сил та часу, для того, щоб використовувати цей метод для вирішення прикладних задач із використанням доступних бібліотек машинного навчання, достатньо лише розуміти, що застосування методу опорних векторів – це пошук прямої, яка розділяє максимально ефективно два класи.

За останні роки виникла ціла екосистема бібліотек, в яких досить грамотно реалізовані найбільш відомі базові алгоритми МН. Відтак не обов'язково реалізовувати кожен з алгоритмів з нуля. Зрозуміло, що для людини, яка займається МН, потрібно розуміти, як працює кожен із алгоритмів.

Однією з мов програмування, яка максимально використовується для вирішення прикладних

задач, є мова Python. Ця мова має ряд переваг. Вона досить проста у вивченні і, як правило, це мова, яка потребує низького рівня входу.

Суб'єкт, який має базові знання з теорії алгоритмів і математики, досить просто може освоїти базовий функціонал, методи і синтаксис для того, щоб вирішувати прикладні задачі.

Саме на базі цієї мови реалізована велика кількість бібліотек, які надають у зручному виді більшість доступних алгоритмів.

Більшість із цих бібліотек використовують бібліотеку **NumPy**. Це бібліотека, яка дозволяє швидко і ефективно працювати з числовими даними, матрицями, таблицями чисел в різних форматах, проводити велику кількість типових операцій, що потрібні в процесі вирішення прикладних задач машинного навчання.

Дана мова має зручний менеджер пакетів pip, який теж потрібно встановити, а також при необхідності є можливість використання бібліотеки NumPy, в коді при ініціалізації достатньо буде його імпортувати.

Загалом для машинного навчання існує декілька базових бібліотек на Python, які мають досить велику перевагу у порівнянні з бібліотеками інших мов програмування. І основна з них характеризується детальною і якісною документацією.

Одною з бібліотек машинного навчання з хорошою документацією, яка реалізує більшість типових методів є scikit-learn. У даній бібліотеці реалізовані десятки алгоритмів для задач кластеризації, регресії, класифікації, методу опорних векторів, лінійної та логістичної регресія та багато інших.

У кожному із доступних алгоритмів є велика кількість параметрів, які слід враховувати при налаштуванні під індивідуальне звдання.

Машинне навчання завжди потребує попереднього форматування даних для того, щоб натренувати певну модель (необхідно дані попередньо підготувати в відповідному форматі, в певній векторній репрезентації).

Однією із дуже зручних бібліотек усередині мови Python, які допомагають працювати з великою кількістю табличних даних (досить часто тренувальні і тестувальні вибірки виглядають, як .csv-таблиці з сотнями тисяч і мільйонами рядків і колонок параметрів) є бібліотека **pandas**. Вона дозволяє дуже швидко завантажувати дані, препроцесити їх (готувати у відповідний формат), щоб у зручному вигляді відправляти на опрацювання алгоритму. Наприклад, вибраного з бібліотеки scikit-learn.

Сьогодні, актуальним є завдання зменшення вимірності. Тому є багатовимірні дані і необхідно переглянути їх двовимірну проекцію – карту і як дані структурно залежні один від одного.

Дане задання можна вирішити вибравши алгоритмом зменшення вимірності t-SNE. При встановленні бібліотеки та імпортованому методі, якщо є 2000 векторів у стовимірному форматі та в діапазоні від –1 до 1, процес зменшення вимірності буде виглядати так: спершу відбувається імпортування даних з .csv формату, використовуючи pandas, далі конвертація у вид певного NumPy-масиву, врахувавши припущення з float 32 типом даних, врешті ініціалізація t-SNE-оптимізатор. Для початку можна не вказувати унікальні гіперпараметри, а використовувати стандартні. Якщо необхідно отримати двовимірні точки, які відповідають кожному із стовимірних векторів так, щоб їх можна було розглянути на площині, необхідно застосувати їх в нашій моделі t-SNE метод fit або fit transform, в який передаємо масив із списком векторів.

У результаті отримано також список векторів, де кожному відповідному стовимірному елементу з аналогічним індексом буде відповідати набір двовимірних координат, які можна накласти на площину і подивитися, як розподілені вказані дані, щоб визначити певні закономірності. відповідно увесь даний код процесу зменшення вимірності сумарно з імпортами бібліотек буде не більше 20-30 рядків коду.

Одна із досить поширених бібліотек низькорівневих операцій для реалізації алгоритмів МН – бібліотека Theano. Дана бібліотека реалізує комплексні матричні мультиплікації, швидкі методи згортки з множенням, вичленовуванням, методи регресії і всю backend-логіку роботи нейронних мереж.

Одним із ключових бонусів таких бібліотек є те, що окрім CPU-реалізації (тобто реалізації роботи алгоритму на процесорі), бібліотеки типу Theano або TensorFlow (від Google) є open source, тобто з відкритим кодом (можна його подивитися чи дописати модулі, яких Вам не вистачає).

Один із ключових бонусів – у них реалізована підтримка обчислень на відеокартах. Процес роботи або тренування МН пришвидшується у 70-300 разів у порівнянні із швидкістю роботи на процесорі, оскільки він виконує операції виконує послідовно, по черзі. При виконанні обчислень на відеокарті, що являє собою систему з великої кількості маленьких процесорів, одна складна задача розбивається на велику кількість маленьких задач і вони виконуються паралельно. Саме бібліотеки Theano, TensorFlow досить ефективно використовують можливості GPU-обчислень (обчислення на відеокартах).

Лаконічності в ініціалізації машинного нвчання можна досягнути, використовуючи бібліотеку Keras. Це бібліотека, яка одночасно дозволяє і використовувати або backend бібліотеки Theano, або backend бібліотеки TensorFlow. Оскільки основна логіка зберігається, а відрізняються методи реалізації, то високорівневу структуру наших МН можливо описувати на високорівневому API, а вже в залежності від потреби змінювати backend, на якому це все обчислюється – на процесорі чи відеокарті.

Кількість бібліотек зростає щодня і навіть виник певний набір бібліотек, які стабільно встановлюються. Розроблені цілі пакети бібліотек, які можна встановити за один раз. Це дозволяє перетворити персональний комп'ютер на готову станцію, де можна реалізовувати будь-які алгоритми машинного навчання, аналізу великих даних. Один з таких пакетів – Anaconda Python – це повністю налаштоване середовище програмування, в якому попередньо встановлені десятки і сотні бібліотек, версії яких не конфліктують між собою.

Висновки і пропозиції. Машинне навчання досліджує вивчення та побудову алгоритмів, які можуть навчатися з даних, і виконувати передбачувальний аналіз на них. Такі алгоритми ді-

ють шляхом побудови моделі зі зразкового тренувального набору вхідних спостережень, щоби здійснювати керовані даними прогнози або ухвалювати рішення, виражені як виходи, 2 замість того, щоби суворо слідувати статичним програмним інструкціям. Сфера машинного навчання розвивається неймовірними темпами. Лише за останній рік ефективність вирішення типових задач зросла в десятки разів.

У пошукового запиту теж є властивості, наприклад, геозалежні – це означає, що для хорошої відповіді на цей запит потрібно врахувати регіон, з якого він був заданий. Властивості запиту і сторінки, які важливі для ранжирування і які можна виміряти числами, називаються факторами ранжирування. Для точного пошуку важливо враховувати багато різних чинників.

Процес оптимізації пошукових запитів можна зробити ефективнішим із врахуванням розуміння логіки процесу, принципу роботи алгоритму, вибору мови програмування, методу реалізації, сфери застосування алгоритму.

Список літератури:

1. Машинне навчання. URL: http://company.yandex.ru/technologies/spectrum/index.xml (дата звернення: 21.01.18).
2. Курс «Машинне навчання» на Prometheus. URL: https://prometheus. org.ua/ (дата звернення: 12.02.18).
3. Машинне навчання. URL: https://uk.wikipedia.org/wiki/Машинне_навчання (дата звернення: 25.03.18).
4. R. Kohavi and F. Provost, "Glossary of terms, " Machine Learning, vol. 30, no. 2-3, pp. 271-274, 1998 (англ.).
5. Narayanan Arvind (August 24, 2016). Language necessarily contains human biases, and so will machines trained on language corpora. Freedom to Tinker (англ.).
6. Машинне навчання. URL: https://uk.wikipedia.org/wiki/Машинне_ навчання (дата звернення: 28.11.18).

Семчишин Е.М., Карабин О.И.
Тернопольский национальный педагогический университет
имени Владимира Гнатюка

ПОСТРОЕНИЕ ПОИСКОВЫХ СИСТЕМ НА ОСНОВЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

Аннотация
Рассмотрены теоретические основы поисковых систем на основе методов машинного обучения; выяснено с помощью которых языков программирования и их библиотек можно реализовать; определены аспекты задействования методов машинного обучения над уже существующими; использовано направления обосновать работу поисковых систем с одним запросом и провести их сравнительный анализ. На организационно-деятельностного этапе определены этапы работы поисковых систем и их возможности для одного запроса. Перспективно-поисковый этап предусматривает использование большого спектра библиотек для реализации наилучшего запроса.
Ключевые слова: образовательный процесс, машинное обучение, искусственный интеллект, алгоритмы, библиотека машинного обучения, среда программирования, языки программирования.

Semchyshyn O.M., Karabin O.Y.
Volodymyr Hnatiuk National pedagogical University of Ternopil

BUILDING SEARCH SYSTEMS BASED ON METHODS OF MASTER TRAINING

Summary
The theoretical foundations of search systems on the basis of machine learning methods are considered; it is clarified with which programming languages and their libraries can be realized; the aspects of the use of methods of machine learning over existing ones are determined; the direction used to justify the work of search engines with one query and to conduct their comparative analysis. At the organizational and activity stage, the stages of the search engines' operation and their capabilities for one query are determined. The prospect-search phase involves using a larger library of libraries to implement the best query.
Keywords: educational process, machine learning, artificial intelligence, algorithms, machine learning library, programming environment, programming languages.