УДК 681.3 https://doi.org/10.15407/pp2018.04.069

Ю.В. Рогушина

ВИКОРИСТАННЯ ОНТОЛОГІЧНИХ ЗНАНЬ У МЕТОДАХ МАШИННОГО НАВЧАННЯ ДЛЯ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ BIG DATA

Розглянуто проблеми, пов'язані з обробкою великих даних з метою здобуття з них неявних знань. Проаналізовано методи машинного навчання, що можуть застосовуватися для цього, та доцільність поєднання їх з технологіями Semantic Web та елементами штучного інтелекту, що стосуються інтелектуальної поведінки, навчання та адаптації обчислювальних систем. Наведено класифікацію типів задач інтелектуального аналізу даних, для яких застосовують засоби машинного навчання, та розглянуто їх специфіку, пов'язану з Big Data. Проаналізовано сучасні тенденції розвитку машинного навчання, пов'язані з глибоким навчанням та нейронними мережами. У роботі розглядаються сучасні засоби представлення знань про предметну область задачі, що базуються на технологіях Semantic Web, – онтології та семантична розмітка, та шляхи їх застосування для покращення результатів машинного навчання. Розглянуто приклади застосування онтологій та семантичної Wiki-розмітки для підвищення ефективності машинного навчання.

Ключові слова: машинне навчання, онтологія, Big Data.

Вступ

Сучасні тенденції розвитку інформаційних технологій (ІТ) пов'язані із обробкою даних великого обсягу та невизначеної структури із застосуванням елементів штучного інтелекту (ШІ). На потребу в застосуванні ШІ вплинули такі фактори, як: – поширення великих даних (Big Data) – інформації, для обробки якої традиційні засоби ІТ виявляються неефективними через її великий обсяг та швидкість накопичення; – зниження вартості зберeження й обробки таких даних; а також розвиток методів і засобів інтелектуального аналізу даних (Data Mining). Використання зовнішніх знань дозволить інтегрувати Big Data з різних джерел, пов'язати їх з відповідними предметними областями (ПрО) та проінтерпретувати зміст результатів їх аналізу.

Big Data

Певний набір даних доцільно розглядати як Big Data, якщо йому притаманні одна чи кілька наступних характеристик [1], що отримали назву «П'ять V»:

• *обсяг (*volume) – великі обсяги потребують спеціалізованих засобів зберeження та обробки;

• *швидкість (*velocity) – дані накопичуються з високою швидкістю;

• *різноманіття* (variety) – дані можуть бути представлені у різноманітних форматах і типах даних, що ускладнює їх інтеграцію та обробку;

• *достовірність* (veracity) – дані можуть містити помилки та шум, які не можуть бути перетворені в інформацію і, отже, не мають цінності.

• *цінність* (value) – тільки частина даних може бути корисною.

Можна класифікувати Big Data за походженням та структурованістю [2]. Дані, що обробляються рішеннями для великих даних, можуть генеруватися людиною (через різноманітні цифрові пристрої) або комп'ютерами (програмними й апаратними засобами – у відповідь на події реального світу), але здобуття з цих даних аналітичних результатів має бути автоматизованим. Серед них зустрічаються структуровані, слабо структуровані та неструктуровані дані. Через складність обробки Big Data можуть не мати моделі подання, але супроводжуються певними метаданими, що містять відомості щодо характеристики, походження і структури набору даних.

З Big Data пов'язані нові моделі даних, інфраструктура та життєвий цикл, а також нова аналітика, що передбачає ана-

ліз в реальному часі, аналіз потоків, інтерактивне машинне навчання [3].

Традиційна математична статистика, яка довгий час залишалась основним інструментом аналізу даних, так само як і засоби оперативної аналітичної обробки даних (online analytical processing – OLAP), не достатні сьогодні для вирішення таких задач: такі методи використовуються для перевірки заздалегідь сформульованих гіпотез, але саме формулювання цих гіпотез виявляється найскладнішим завданням в аналізі даних.

Використання методів ШІ в областях, пов'язаних з машинним навчанням (ML – machine learning), логічним виведенням та онтологічним аналізом, забезпечує їх взаємне вдосконалення. Але застосування ML до Big Data має значну специфіку і повинно враховувати властивості таких даних.

Data Mining та машинне навчання

Інтелектуальний аналіз даних (Data Mining – у буквальному перекладі з англійської – «розкопка даних») – напрямок в IT, ціллю якого є автоматизоване здобуття знань, які неявним чином присутні в оброблюваній інформації. Один із засновників цього напрямку Г. Пятецький-Шапіро визначив DataMining як це процес дослідження і виявлення у сирих даних комп'ютером прихованих знань, які раніше не були відомими і є нетривіальними, практично корисними та доступними для інтерпретації. Data Mining базується на методах машинного навчання (Machine Learning – ML), що призначені для розпізнавання різних типів інформаційних об'єктів [4]. Але Data Mining – це більш широке поняття порівняно з ML, яке враховує й семантичні аспекти аналізу даних.

Найбільш розповсюдженими задачами Data Mining є:

- класифікація,
- кластеризація,
- прогнозування,
- асоціація,
- візуалізація,
- аналіз і виявлення відхилень,
- оцінювання,
- аналіз зв'язків,
- підведення підсумків.

Можна використовувати дуже загальне визначення «навченості», яке дає Т. Мітчелл [5]: «Комп'ютерна програма навчається в міру накопичення досвіду щодо деякого класу задач T і цільової функції P, якщо якість рішення цих задач (щодо P) поліпшується з отриманням нового досвіду». Хоча це визначення є надзвичайно узагальненим, воно насправді дозволяє прояснити деякі важливі моменти. Наприклад, центральне місце в ML займають не дані, що обробляються, а цільова функція. Вирішуючи будь-яку практичну задачу, важливо ще до початку навчання визначити цільову функцію та засоби її оцінювання. Вибір цільової функції навіть у схожих задачах може привести до зовсім різних моделей.

Інтуїтивно зрозуміло, що «навчання» – це коли деяка модель якимсь образом «навчається», а потім починає прогнозувати нові результати [6].

ML базується на теорії ймовірностей. Ідея застосування оцінки ймовірностей апріорних й апостеріорних гіпотез для ML походить до роботи Т. Байєса «Нариси до рішення проблеми доктрини шансів» (An Essay towards solving a Problem in the Doctrine of Chances), що вийшла вже після смерті автора, у 1763 році [7]. Формула Байєса

$$p(y|x) = \frac{p(x|y) * p(y)}{p(x)}$$

дозволяє переоцінювати апріорні представлення про світ $p(y)$ на основі часткової інформації (даних), отриманих у вигляді спостережень $p(x|y)$), як висновок одержуючи новий стан представлень $p(y|x)$. Це і складає байєсівський підхід до ймовірностей. Сам термін з'явився в середині XX століття в роботі Х. Джеффриса «Теорія ймовірностей» [8] і Л. Севіджа [9].

Випадкові величини поділяють на дискретні і безупинні. Дискретна випадкова величина може мати скінчену або пере-

раховувану кількість станів. Розподіл ймовірності описує, з якою ймовірністю випадкова величина чи множина випадкових величин приймає кожне можливе значення. Спосіб завдання розподілу ймовірності залежить від того, є випадкова величина безупинною чи дискретною.

Ключові моменти сучасних ML [10]:

- формування простору ознак;
- перевірка гіпотез про об'єкти і класи об'єктів, визначення мір подібності для класів об'єктів;
- формування навчальної вибірки;
- формування тестової вибірки;
- вибір алгоритму навчання.

На жаль, в процесі аналізу "сирих" даних значна частина праці пов'язана з підготовкою та очищенням даних (за даними, наведеними у [11] – до 60 % часу досліджень, порівняно з 4 % на побудову навчаючої вибірки та 9 % – безпосередньо на дослідження даних на наявність закономірностей). Це викликає потребу використовувати там, де це можливо, вже структуровані (хоча б частково) та верифіковані дані, за якими може будуватися навчаюча вибірка для традуктивного виведення.

Задачі машинного навчання поділяють на два основні класи – навчання з учителем (supervised learning) і навчання без учителя (unsupervised learning). При навчанні з учителем на вхід подається набір класифікованих прикладів – навчальна вибірка (training set), і завдання полягає у тому, щоб класифікувати приклади з тестового набору даних (test set). Основне припущення полягає у тому, що дані з навчальної вибірки та тестового набору, схожі на ті дані, на яких потім буде застосовуватися результат навчання. Задачі навчання з учителем звичайно поділяються на задачі класифікації і регресії. У задачі класифікації потрібно поданий на вхід об'єкт визначити в один із скінченої множини класів, а в задачі регресії прогнозувати значення деякої функції, у якої може бути нескінченно багато різних значень (наприклад, за ростом людини прогнозувати її вагу).

У найбільш загальному випадку задача ML з учителем має наступний вигляд:

На вхід подається навчальна вибірка

$$X = \{< x_i, f(x_i) >\}, \quad i = \overline{1,n}$$

– набір з n класифікованих прикладів, де

$$x_i = < x_{i_1}, ..., x_{i_1} > .$$

Задачею навчання є побудова функції $g(x)$, такої, що $g(x_i) = f(x_i)$ або хоча б $g(x_i) \approx f(x_i)$.

Класифікація – найбільш проста і розповсюджена задача Data Mining. У результаті вирішення задачі класифікації виявляються ознаки, що характеризують групи об'єктів досліджуваного набору даних – класи; за цими ознаками новий об'єкт можна віднести до того чи іншого класу. Для вирішення задачі класифікації можуть використовуватися методи найближчого сусіда, дерева рішень, нейронні мережі тощо.

Для виявлення таких зв'язків можна скористатися методами індуктивного і традуктивного здобуття знань з даних, більш детальний огляд яких наведено в [12].

Існують незалежні підходи до реалізації подібних методів: ID3, ACLS, CART і т. д. Найбільш цікавим, у зв'язку зі специфікою проведеної роботи, виявився алгоритм ID3 [13], що спеціально розроблений для здобуття корисної інформації з великих обсягів слабко структурованих даних.

Незростаючий алгоритм ID3 призначений для узагальнення досвіду експериментів, параметри і результати яких описані через якісні оцінки (лінгвістичні перемінні). Він забезпечує побудову бінарного дерева рішень, а цього недостатньо зручно для представлення закономірностей багатьох ПрО. Його модифікація ID3m [14] призначена для довільної (скінченої) кількості рішень. Він також належить до незростаючих алгоритмів.

Якщо ж розміченого набору даних, відповідного конкретній задачі, немає, а є просто дані, з яких треба здобути якийнебудь зміст, то виникають задачі навчання *без учителя*. Типові приклади навчання без учителя – це кластеризація (clustering) та зниження розмірності (dimensionality reduction) та оцінки щільності. Зазвичай

такі задачі виникають на попередніх етапах дослідження даних.

Задача кластеризації є логічним продовженням ідеї класифікації і полягає в розподілі множини об'єктів на групи (кластери), при цьому в кожному кластері зібрані об'єкти, які схожі за параметрами. Варто зауважити, що на відміну від класифікації, кількість кластерів і їхніх характеристик можуть бути заздалегідь невідомими і визначатися в ході побудови кластерів, виходячи зі ступеня близькості поєднуваних об'єктів за сукупністю параметрів.

Зазначені вище задачі у залежності від використовуваних моделей, забезпечують опис (descriptive) і прогнозування (predictive) [15].

У результаті рішення *описових* задач аналітик отримує шаблони, що описують дані, які піддаються інтерпретації. Ці задачі описують загальну концепцію аналізованих даних, визначають інформативні, підсумкові, відмінні риси даних. Концепція описових задач має на увазі характеристику і порівняння наборів даних. Характеристика набору даних забезпечує короткий і стиснутий опис деякого набору даних. Порівняння забезпечує порівняльний опис двох чи більш наборів даних.

Задачі прогнозування (predictive) ґрунтуються на аналізі даних, створенні моделі, передбаченні тенденцій чи властивостей нових або невідомих даних. До них відносяться: класифікація об'єктів (для заздалегідь заданих класів); регресійний аналіз, аналіз часових рядів.

До описових задач належать: пошук асоціативних правил або патернів (зразків); групування об'єктів, кластерний аналіз; побудова регресійної моделі.

Значна частина методів ML використовує тільки параметричні моделі, які дозволяють отримувати функцію, що описана вектором параметрів скінченого розміру. У непараметричних моделей такого обмеження немає.

Деякі непараметричні моделі – це просто теоретичні абстракції (наприклад, алгоритм, пошуку серед усіх можливих розподілів ймовірності), не реалізовані на практиці. Однак існують і корисні непараметричні моделі, складність яких залежить від розміру навчального набору. Прикладом непараметричного алгоритму навчання є метод k найближчих сусідів, що не обмежений фіксованою кількістю параметрів. Звичайно вважається, що цей алгоритм узагалі не має параметрів, а реалізує просту функцію від навчальних даних.

На етапі тестування потрібно знайти в навчальному наборі X k найближчих сусідів для x, а потім повернути середнє значення від відповідних їм міток y. Ця ідея працює для будь-якого виду навчання з учителем, за умови що можна визначити поняття середньої мітки.

Алгоритм k найближчих сусідів, будучи непараметричним, може досягати дуже високої ємності, що дозволяє отримати високу правильність для великої навчальної вибірки, але це призводить до високої вартості обчислень. При малому навчальному наборі алгоритм погано узагальнюється. Одне з слабких місць алгоритму k найближчих сусідів – невміння зрозуміти, що одна ознака є більш важливою, ніж інша.

Ще один тип алгоритму навчання, що також розбиває простір входів на області, кожна з який описується окремими параметрами, – *дерево рішень* [16] і його численні варіанти. З кожним вузлом дерева рішень асоційована область простору входів, і внутрішні вузли розбивають цю область на дві частини – по одній для кожного дочірнього вузла (звичайно розсікаючи паралельно осі). Таким чином, простір входів поділяється на непересічні області, що взаємно однозначно відповідають листовим вузлам. Звичайно кожен листовий вузол зіставляє кожній вхідній точці у своїй області той самий вихід. Алгоритм навчання можна вважати непараметричним, якщо йому дозволено будувати дерево довільного розміру.

Глибоке навчання

Стимулами для розробки концепції глибокого навчання були як нездатність традиційних алгоритмів отримати узагальнення на таких задач ШІ, як розпізнаванні мови й зображень, так і відсутність масштабованості традиційних методів ML: зро-

стання обсягу даних викликає експоненційне ускладнення обчислень.

Сучасне глибоке навчання пропонує розвинуту інфраструктуру навчання з учителем. Завдяки додаванню додаткових шарів і блоків у межах одного шару глибока мережа може представляти усе більш і більш складні функції. Більшість задач, що зводяться до відображення вхідного вектора у вихідний, з якими легко справляється людина, може бути вирішено методами глибокого навчання за наявності досить великих моделей і наборів позначених прикладів. Інші задачі, які не можна описати як асоціювання одного вектора з іншим чи настільки важкі, що людині потрібно час для їхнього рішення, поки не піддаються глибокому навчанню.

Глибокі мережі прямого поширення, що називають також нейронними мережами прямого поширення, чи багатошаровими перцептронами – це типові приклади моделей глибокого навчання. Ціль мережі прямого поширення – апроксимувати деяку функцію f^*. Наприклад, у випадку класифікатора $y = f'^*(x)$ відображає вхід x у категорію y. Мережа прямого поширення визначає відображення $y = f(x;\theta)$ і шляхом навчання знаходить значення параметрів θ, що дають найкращу апроксимацію.

Глибоке навчання – окремий випадок машинного навчання. Більшість алгоритмів глибокого навчання базуються на алгоритмі оптимізації, що називається *стохастичним градієнтним спуском* (СГС)**,** який узагальнює алгоритм градієнтного спуску.

Ідея методу СГС полягає у тому, що градієнт – це математичне очікування, і, отже, його можна оцінити за невеликою множиною прикладів. Точніше, на кожнім кроці алгоритму можна взяти міні-пакет (minibatch) m' – невелику рівномірну вибірку з навчального набору m. Розмір міні-пакета m' звичайно складає кілька сотень прикладів. Важливо, що розмір m' не залежить від розміру навчального набору m. Це робить такий підхід придатним для обробки Big Data.

Майже всі алгоритми глибокого навчання можна описати як комбінацію набору даних, функції вартості, процедури оптимізації і моделі.

Більшість алгоритмів машинного навчання в тому чи іншому вигляді включає оптимізацію, тобто знаходження мінімуму чи максимуму *цільової функції* $f(x)$ при зміні x. Цільова функція може відображати розміри вартості, помилок тощо. Найчастіше в ролі функції вартості виступає негативна логарифмічна правдоподібність, тому її мінімізація дає оцінку максимальної правдоподібності.

Ключова ідея полягає у тому, що дуже велику кількість областей, порядку $O(2^k)$, можна визначити за допомогою $O(k)$ прикладів, якщо ввести деякі *залежності* між областями за допомогою додаткових припущень про істинний породжуючий розподіл. Таким чином, з'являється можливість нелокального узагальнення. Щоб скористатися нею, у багатьох алгоритмах глибокого навчання приймаються явні чи неявні припущення, дійсні для широкого кола задач ШІ.

В основі багатьох ідей машинного навчання лежить концепція різноманіття.

Різноманіття – це зв'язна область, яку можна розглядати як множину точок, асоційованих з околицею кожної точки. З будь-якої точки локальне різноманіття виглядає як евклідів простір. У ML цей термін використовують для позначення зв'язної множини точок у просторі високої розмірності, яку можна добре апроксимувати, вводячи в розгляд лише невелику кількість ступенів волі, чи вимірів. Кожен вимір відповідає локальному напрямку зміни.

Багато задач машинного навчання здаються безнадійними, якщо очікувати, що в результаті навчання алгоритм повинний знайти функції з нетривіальними змінами у всьому просторі. Алгоритми навчання різноманіть переборюють цю перешкоду, припускаючи, що велика частина – неприпустимі вхідні дані, а цікаві входи зосереджені тільки в наборі різноманіть, що містить невелику підмножину точок, причому цікаві зміни результуючої навче-

ної функції відбуваються тільки уздовж напрямків, що належать якомусь одному різноманіттю, чи при переході з одного різноманіття до іншого. Навчання різноманіть зародилося при розгляді безупинних даних у випадку навчання без учителя, хоча сама ідея концентрації ймовірності узагальнюється і на дискретні дані, і на навчання з учителем: ключове допущення полягає у тому, що маса ймовірності сконцентрована в малій області.

Припущення про те, що дані розташовані уздовж різноманіття низької розмірності, не завжди виявляється правильним чи корисним. Але в задачах ШІ, зокрема при обробці зображень, звуку чи тексту, припущення про різноманіття, принаймні, приблизно правильно.

Якщо дані розташовані на різноманітті малої розмірності, то в алгоритмі машинного навчання їх найбільше природно представляти координатами на цьому різноманітті, а не в R. У побуті ми розглядаємо дороги як одномірні різноманіття, занурені в тривимірний простір. Бажаючи повідомити адресу будинку, ми вказуємо його номер щодо вулиці, а не координати в просторі. Перехід у систему координат різноманіття – важка задача, але її рішення обіцяє помітне поліпшення багатьох алгоритмів машинного навчання. Цей загальний принцип застосуємо в самих різних контекстах.

Мережі прямого поширення важливі для практичного застосування машинного навчання. Вони лежать в основі багатьох важливих комерційних додатків. Наприклад, згорткові мережі, які використовують для розпізнавання об'єктів на фотографіях, – це окремий випадок мереж прямого поширення.

Нейронні мережі прямого поширення називаються мережами, тому що вони, як правило, утворені композицією багатьох різних функцій. З моделлю асоційований орієнтований ациклічний граф, що описує композицію. Наприклад, можна зв'язати три функції $f1, f2, f3$ у ланцюжок

$$f(x) = f1(f2(f3(x))).$$

Такі ланцюгові структури найчастіше використовуються в нейронних мережах. У даному випадку $f1$ називається першим шаром мережі, $f2$ – другим шаром і т. д. Загальна довжина ланцюжка визначає глибину моделі.

Назва «глибоке навчання» безпосередньо пов'язана з цією термінологією. Останній шар мережі прямого поширення називається *вихідним*. У ході навчання нейронної мережі потрібно наблизити $f(x)$ до $f^*(x)$. Навчальні дані – це зашумлені наближені приклади $f^*(x)$, обчислені в різних точках. Кожен приклад x супроводжується міткою $y \approx f^*(x)$. Навчальні приклади прямо вказують, що у вихідному шарі повинне відповідати кожній точці x, це має бути значення, близьке до y.

Поведінка інших шарів прямо навчальними даними не визначається. Алгоритм навчання повинний вирішити, як використовувати ці шари для породження бажаного виходу, але навчальні дані нічого не говорять про те, що саме повинний робити кожен шар. Алгоритму навчання треба самостійно вирішити, як за допомогою цих шарів домогтися найкращої апроксимації f^*. Оскільки навчальні дані не визначають виходів кожного з цих шарів, вони називаються *схованими* шарами.

Ці мережі називаються *нейронними*, тому що їхня ідея запозичена з нейробіології. Кожен схований шар мережі звичайно виробляє векторні значення. Розмірність схованих шарів визначає *ширину моделі*. Кожен елемент вектора можна інтерпретувати як нейрон. Замість того щоб розглядати шар як представлення функції з векторними аргументами і векторними значеннями, можна вважати, що шар складається з багатьох блоків, що працюють паралельно, і що кожен такий блок представляє функцію, що відображає вектор у скаляр. Кожен блок нагадує нейрон у тім розумінні, що отримує дані від багатьох інших блоків і обчислює власне значення активації. Ідея використання багатьох шарів векторних представлень прийшла з нейробіології. Вибір функцій $f_i(x)$, що використовуються для обчислення цих представлень, також походить від експеримен-

тально отриманих фактів про функції, що обчислюються біологічними нейронами. Але перед нейронною мережею не ставиться ціль змоделювати роботу мозку. Краще розглядати мережі прямого поширення не як моделі функціонування мозку, а як машини для апроксимації функцій, що спроектовані з метою статистичного узагальнення й іноді використовують деякі знання про мозок людини.

Один із способів розібратися в мережах прямого поширення полягає в тому, щоб почати з лінійних моделей і перебороти їхнього обмеження. Лінійні моделі, такі як логістична регресія і лінійна регресія, привабливі тим, що дають ефективну і надійну апроксимацію в замкнутій формі чи за допомогою опуклої оптимізації. Але в лінійних моделей є очевидний недолік – ємність моделі обмежена лінійними функціями, тому модель нездатна відобразити довільний зв'язок між двома величинами.

Щоб узагальнити лінійну модель на представлення нелінійних функцій від x, можна застосувати її не до самого x, а до результату обчислення $\phi(x)$, де ϕ – нелінійне перетворення. Можна вважати, що ϕ дає набір ознак, що описують x, чи нове представлення x.

Тоді питання зводиться до вибору відображення ϕ.

1. Один з варіантів – узяти дуже загальне відображення ϕ. Якщо розмірність $\phi(x)$ досить велика, то ємності моделі вистачить для апроксимації навчального набору, але узагальненість на тестовому наборі часто залишає бажати кращого. Дуже загальні відображення ознак звичайно базуються на принципі локальної гладкості, і закодованої в них апріорної інформації недостатньо для рішення складних задач.

2. Інший варіант – спроектувати відображення ϕ вручну. До виникнення глибокого навчання так в основному і робили. Але для кожної задачі потрібні були десятиліття людської праці і фахівці у відповідній предметній області, наприклад, з розпізнавання мови чи комп'ютерного зору, а передачі знань між різними областями майже немає.

3. Стратегія глибокого навчання складається в навчанні ϕ. При такому підході є модель

$$y = f(x;\theta;w) = \phi(x;\theta)^T w .$$

Параметри θ використовуються для навчання ϕ, обраної із широкого класу функцій, і параметри w, що відображають $\phi(x)$ у бажаний вихід. Це приклад глибокої мережі прямого поширення, де ϕ визначає схований шар. Це єдиний із трьох підходів, що не потребує припущення про опуклість задачі навчання, але його переваги переважують недоліки. У цьому випадку потрібно параметризувати представлення у вигляді $\phi(x;\theta)$ і застосувати алгоритм оптимізації для знаходження відображення ϕ, якому відповідає гарне представлення. В цьому підході є усі переваги узагальненості першого – для цього потрібно тільки взяти дуже широке сімейство функцій $\phi(x;\theta)$. Глибоке навчання може також скористатися перевагами другого підходу. Дослідник може включити в модель свої знання, спроектувавши сімейство функцій, яке, на його думку, повинне добре узагальнюватися. Перевага в тому, що людині потрібно тільки відшукати придатне сімейство функцій, а не одну конкретну функцію.

При навчанні мережі прямого поширення необхідно враховувати ті ж речі, що для лінійних моделей: вибір оптимізатора, функції вартості і вигляд вихідних блоків.

Оскільки в мережах прямого поширення є приховані шари, то потрібно вибрати функції активації, що будуть використані для обчислення вироблюваних ними значень. Крім того, потрібно спроектувати архітектуру мережі: скільки в ній схованих шарів, як ці шари зв'язані між собою, скільки блоків у кожному шарі.

Напрямки інтеграції інтелектуальних технологій з обробкою Big Data

Дослідження інформаційних ресурсів Web та Big Data спрямовані на здобуття з них потрібних користувачам відо-

мостей та знань. Такі знання можуть відображати зв'язки між різними фактами та твердженнями. Для цього доцільно застосовувати методи машинного навчання, але необхідно враховувати можливість їх масштабування для Big Data, тобто оцінювати їх обчислювальну складність і прогнозувати час роботи на великих навчальних вибірках. Щоб спростити таку обробку, виникає потреба у використанні вже наявних зовнішніх знань про ті інформаційні об'єкти, відомості про які потрібно здобути: це дозволяє виключити здобуття вже відомих закономірностей, структурувати простір ознак та конкретизувати вимоги до рішення. Це дозволяє використовувати складні методи ML для обробки великих обсягів даних. Тому наукові дослідження в цій сфері – розробка відповідних моделей і методів та оцінка їх ефективності є сьогодні одним з пріоритетних напрямків наукових досліджень.

Перші розробки з ML (приміром, машинний переклад з використанням статистичних методів) припускали, що результати можна отримувати субсимвольно, тобто без конкретних представлень про знання на неінтерпретованих даних. Такі системи здатні до оптимізації набору параметрів моделі для підвищення продуктивності з часом, але цей процес не має ніякої подібності до того, як міркують та вчаться люди.

Більш сучасні методи ШІ використовують, крім оптимізації і статистичних підходів, біоподібні нейронні мережні архітектури. Ці методи теж є субсимвольними і працюють «знизу нагору» – від даних, таких як текст і зображення. Така взаємодія з текстом і зображеннями теж сильно відрізняється від набагато більш широкого, біологічно подібного досвіду взаємодії з світом.

До теперішнього часу ШІ ще не опанував більш широкими формами навчання і розуміння, що походять з реального досвіду. Деякі вважають, що таке навчання на основі реального досвіду повинне починати створення системи з когнітивного ядра, а потім послідовно розробляти більш складні когнітивні моделі. Соціальний аспект реального досвіду містить у собі вивчення загальних знань від інших інтелектуальних агентів, а також з їхніх інформаційних продуктів, до яких відносяться текст, дані і фізичні дії. Водночас як здобуття знань щодо ПрО і створення методів міркувань для ПрО продовжують покращуватися, виявилося, що реалізувати навчання знизу нагору без яких-небудь базових знань дуже складно.

Прикладом інтересу до цього напрямку є Онтологічний самміт 2017 "AI, LEARNING, REASONING AND ONTOLOGIES" [17], на якому досліджено тенденції використання методів ШІ в області ML, міркувань і онтології для їхнього взаємного покращення. В основі цих досліджень лежить діаграма онтологічного навчання (Ontology Learning Layer Cake), що використовувалася як об'єднуючий елемент для всіх напрямків. Ця діаграма онтологічного навчання (рис. 1) містить наступні рівні: терміни; синоніми; поняття; ієрархія понять; відношення; ієрархія відношень; схеми аксіом; загальні аксіоми. Ця діаграма забезпечує концептуальну основу для обговорення того, які типи знань будуються в результаті застосування ML до Big Data.

Для онтологічного аналізу розробляються і можуть бути використані різні підходи й інструменти ML, у тому числі статистичні і лінгвістичні, які дозволяють здобувати інформацію і структуровані знання з різних джерел для полегшення розробки і підтримки онтологій, а також гармонізувати онтології для керування залежністю від особливостей наборів даних.

Велике значення набуває використання апріорних знань та онтологій ПрО для поліпшення результатів ML. Знання дозволяють поліпшити якість результатів ML, використовуючи методи логічного виводу для вибору моделей навчання і підготовки даних для навчання і тестування (скорочення великих, зашумлених наборів даних до керованих) та зробити результати ML більш зрозумілими.

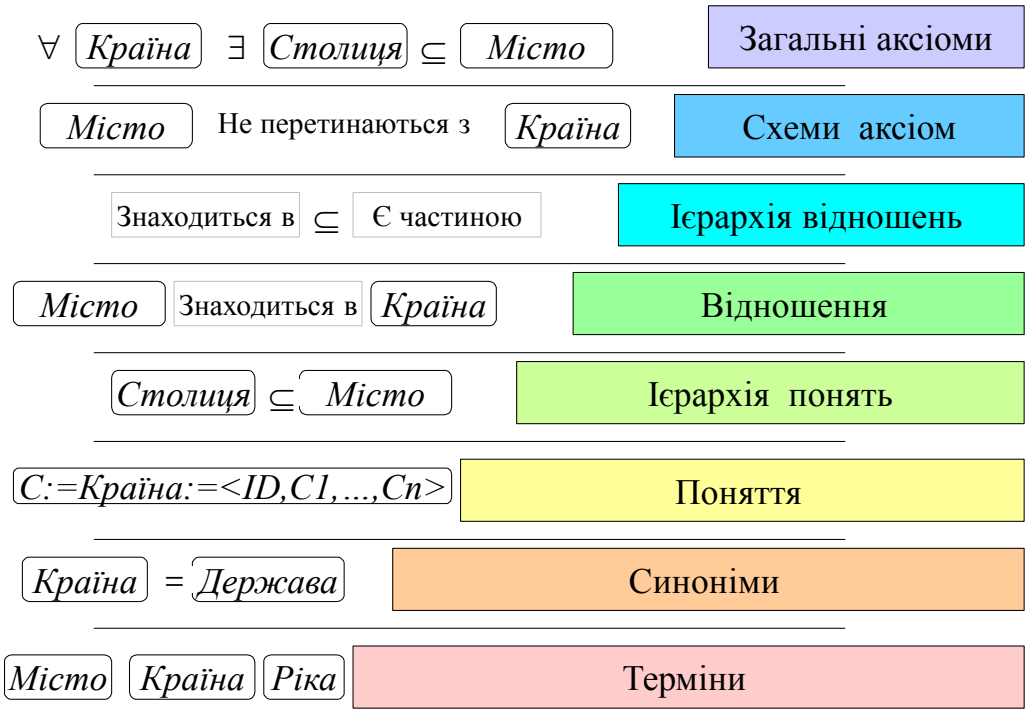

Рис.1. Діаграма онтологічного навчання

Використання ML в аналізі роботи е-ВУЕ

Розглянемо це на прикладі аналізу інформації в електронній версії Великої української енциклопедії [18].

Найпростіша задача прогнозування пов'язана з аналізом переходів між сторінками е-ВУЕ. Такий аналіз спрямований на вдосконалення навігації на порталі (переходи, що виконуються користувачами найбільш часто, доцільно зробити найбільш зручними).

Навчальна вибірка це множина пар Похідна сторінка – Сторінка переходу з значеннями їх властивостей (наявність категорій, значення семантичних властивостей, тип сторінки).

Цю інформацію можна розглядати як Big Data – хоча ці відомості структуровані, але вони надходять швидко й у великій кількості.

Якщо користуватися безпосередньо методами традиційного ML, то виникає надто складна задача з великим простором ознак. Тому доцільно застосувати апріорні знання щодо даної ПрО – створення електронних енциклопедій. Ці знання, відповідно до діаграми онтологічного навчання, – поняття, ієрархія понять, відношення та ієрархія відношень. Відповідна онтологія дозволяє виділити наступні типи шість типів сторінок (рис. 2), що відрізняються засобами навігації.

Переходом до сторінок кожного з цих типів притаманний окремий вигляд користувацького інтерфейсу (рис. 3):

- сторінки-гасла;
- категорії, організовані в набір ієрархій;
- сторінки авторів;
- сторінки медіафайлів;
- сторінки літературних посилань;
- спеціальні сторінки.

Визначення цих типів для сторінок підтримується семантичною розміткою сторінок убудованими засобами середовища Semantic MediaWiki [19].

Таким чином, не потрібно прогнозувати переходи від кожної сторінки p_i до сторінки p_j. Задача зводиться до того, щоб за властивостями і типом сторінки p_i визначити ймовірність того, що сторінка $p_j \in t_k$, $k = \overline{1,6}$.

Слід відмітити, що приклади з такої навчальної вибірки можуть бути суперечними у тому розумінні, що однаковим рядкам можуть відповідати різні результати.

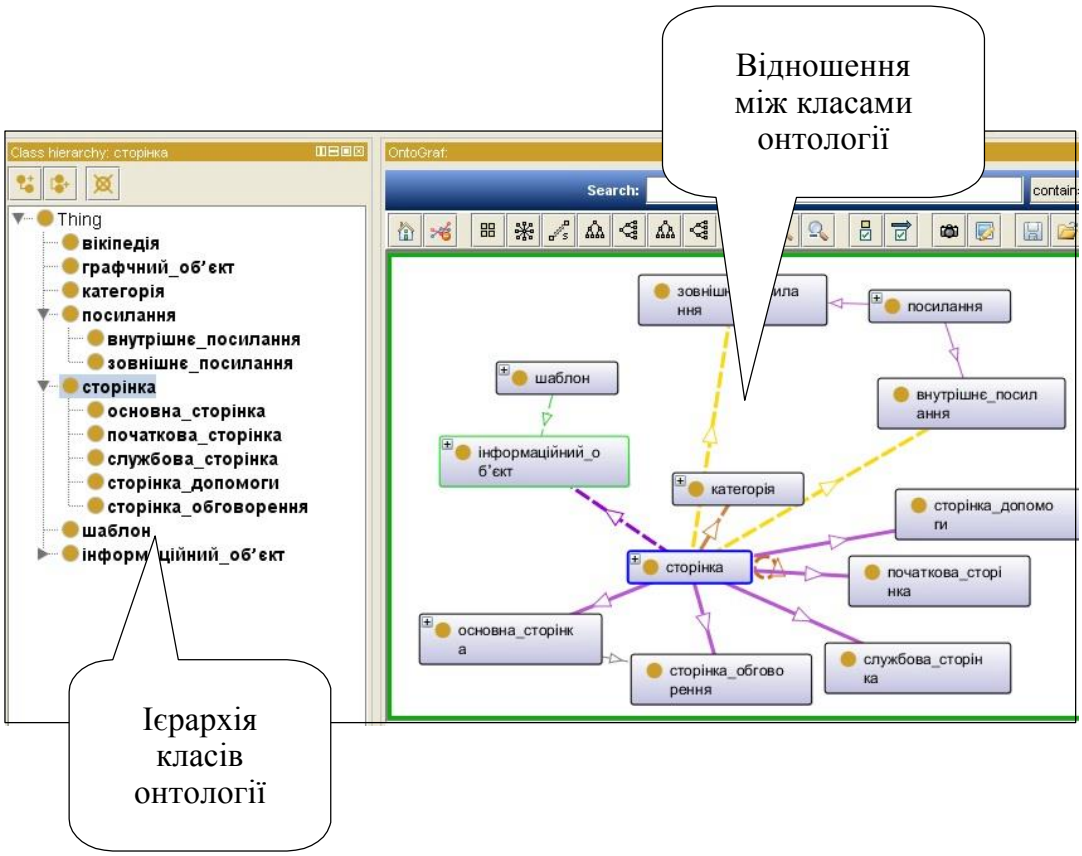

Рис. 2. Онтологічна модель е-ВУЕ

Рис. 3. Засоби навігації в е-ВУЕ на сторінці гасла

Це пояснюється тим, що за різних умов користувачі можуть переходити з тієї ж самої сторінки на різні сторінки, більш того – на сторінки різних типів. Крім того, потрібно враховувати, що значення ознак в навчальній вибірці є дискретними та якісними, а не кількісними. Через це їх неможливо впорядковувати.

Тому доцільно для обробки таких даних застосувати метод *к*-найближчих сусідів. В результаті аналізу даних буде побудовано набір комірок, що відповідають однаковим значенням параметрів. Всередині кожної такої комірки буде знаходитися множина значень типів сторінок переходу (припустима ситуація, коли таких типів буде кілька). Для того, щоб оцінити ймовірність переходу від кожної сторінки, потрібно підрахувати відношення сторінок переходу певного типу до загальної кількості сторінок переходу у комірці.

На сьогодні отримати реальні дані для такого навчання не є можливим через те, що портал е-ВУЕ працює поки що в процесі налагодження, а дії (і відповідно – переходи між сторінками) розробників порталу суттєво відрізняються від типових дій користувачів. Але розробка методів рішення таких задач має виконуватися заздалегідь.

Уявляється корисним надалі застосовувати методи ML, інтегровані з онтологією порталу, для аналізу більш довгих ланцюгів переходів (не з двох, а з більшої кількості кроків) та виконувати цю задачу до окремих підмножин сторінок порталу – приміром, окремо DL кожної галузі знань або для типу інформаційних об'єктів.

Інша, більш складна множина задач ML пов'язана з інтеграцією дій користувачів на порталі е-ВУЕ з їх діяльністю на порталі періодичних наукових видань України.

У цьому випадку кожен рядок навчальної вибірки пов'язується з діями одного користувача на обох порталах, а простір ознак складається з категорій сторінок та інформаційних ресурсів, до яких звертається цей користувач. Е-ВУЕ використовується головним чином як джерело знань про ієрархію понять та інформаційних об'єктів. Прикладом задачі, що може вирішуватися на таких даних, є класифікація наукових публікацій та їх прив'язка до певних гасел та категорій е-ВУЕ.

Висновки

Використання зовнішніх знань щодо ПрО дозволяє зменшувати простір ознак та значно спрощувати складність машинного навчання, у тому числі – для методів глибокого навчання. Тому доцільно виконувати дослідження у напрямку інтеграції аналізу Big Data з методами ML з використанням онтологій, щоб дозволити інтелектуальним застосуванням здобувати з цих даних потрібні для їх функціювання відомості. Джерелами таких знань можуть бути онтології відповідних ПрО та семантично розмічені Wiki-ресурси.

При цьому слід враховувати, що ефективність ML залежить від алгоритмів здобуття знань, даних для обробки (Big Data) та моделей подання отриманих знань, тоді як вибір моделей та методів (а також їх параметрів) повністю визначається конкретною задачею навчання.

У багатьох практичних задачах, пов'язаних з обробкою якісних ознак інформаційних об'єктів, доцільно орієнтуватися на непараметричні моделі (метод найближчих сусідів, дерева рішень), але використовувати різні форми нейронних мереж для попереднього аналізу та кластеризації Big Data. Інтеграція між різними рівнями моделі теж має базуватися на онтологічному аналізі ПрО.

Література

1. Laney D. 3-D data management: Controlling data volume, velocity and variety. *Application Delivery Strategies by META Group Inc.* 2001, P. 949. http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf.
2. Gandom A., Haide, M. Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management.* 2015. 35(2). P. 137–144. https://www.sciencedirect.com/science/article/pii/S0268401214001066.

3. Demchenko Y., De Laat C., Membrey P. Defining architecture components of the Big Data Ecosystem. Collaboration Technologies and Systems (CTS). 2014. P. 104–112.
4. Гладун А.Я., Рогушина Ю.В. Семантичні технології: принципи та практики. К.: ТОВ "АДЕФ-Україна". 2016. 308 с.
5. Mitchell T.M. Machine learning. 1997. Burr Ridge, IL: McGraw Hill. 45(37). 1997. P. 870–877.
6. Николенко С.И., Кадурин А.А., Архангельская Е.О. Глубокое обучение. Издательский дом "Питер", 2017.
7. Bayes T. An Essay Towards Solving a Problem in the Doctrine of Chances. Philosophical Transactions of the Royal Society of London. 1763. Vol. 53. P. 370–418.
8. Jeffreys H. Theory of Probability, Oxford: Oxford University Press, 1939.
9. Savage L. The Foundations of Statistics, New York: Wiley, 1954.
10. Goodfellow I., Bengio Y., Courville A., Bengio Y. Deep learning (Vol. 1). Cambridge: MIT press, 2016.
11. Эрли С. Искусственный интеллект для масштабируемой персонализации. Открытые систем. № 1. 2018. С. 20–24.
12. Рогушина Ю.В., Гладун А.Я., Осадчий В.В., Прийма С.М. Онтологічний аналіз у Web. Монографія. Мелітополь: МДПУ ім. Богдана Хмельницького, 2015. 407 с.
13. Quinlan J.R. Discovery rules from large collections of examples: a case study. Expert *Systems in the Microelectronic Age.* Edinburg, 1979. P. 87–102.
14. Рогушина Ю.В., Гришанова И.Ю. Использование метода индуктивного вывода для усовершенствования онтологии предметной области поиска. *Системні дослідження та інформаційні технології.* 2007. № 1. С. 62–70.
15. Гладун А.Я., Рогушина Ю.В. Data Mining: пошук знань в даних. К.: ТОВ "ВД "АДЕФ-Україна", 2016. 452 с.
16. Breiman L. Bagging predictors. Machine Learning. 1994. 24(2). P. 123–140.
17. Baclawski K., Bennett M., Berg-Cross G., Fritzsche D., Schneider T., Sharma R., Westerninen A. Ontology Summit 2017 communiqué–AI, learning, reasoning and ontologies. Applied Ontology. (Preprint). 2018. P. 1–16. http://www.ccs.neu.edu/home/kenb/pub/2017/09/public.pdf.
18. Рогушина Ю.В. Використання семантичних властивостей вікі-ресурсів для розширення функціональних можливостей «Великої української енциклопедії». Енциклопедичні видання в сучасному інформаційному просторі: колективна монографія / За ред. Киридон А.М. К.: Державна наукова установа «Енциклопедичне видавництво». 2017. С. 104–115.
19. Rogushina J. Semantic Wiki resources and their use for the construction of personalized ontologies. CEUR Workshop Proceedings. 1631. 2016. P. 188–195.

References

1. Laney D. 3-D data management: Controlling data volume, velocity and variety. *Application Delivery Strategies by META Group Inc.* 2001, P. 949. http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf.
2. Gandom A., Haide, M. Beyond the hype: Big data concepts, methods, and analytics. International Journal of Information Management, 35(2), 2015. P. 137–144. https://www.sciencedirect.com/science/article/pii/S0268401214001066.
3. Demchenko Y., De Laat C., Membrey P. Defining architecture components of the Big Data Ecosystem // Collaboration Technologies and Systems (CTS). 2014. P. 104–112.
4. Gladun A.Y., Rogushina J.V. Semantic technologies: principles and practics. K.: ADEF-Ukraine, 2016. 308 p. [in Ukrainian]
5. Mitchell T.M. Machine learning. 1997. Burr Ridge, IL: McGraw Hill. 45(37). 1997. P. 870–877.
6. Nikolenko S.I., Kadurin A.A., Arhangelskaya E.O. Deep Learning. Piter. 2017. [in Russian]
7. Bayes T. An Essay Towards Solving a Problem in the Doctrine of Chances. Philosophical Transactions of the Royal Society of London. 1763. Vol. 53. P. 370–418.
8. Jeffreys H. Theory of Probability, Oxford: Oxford University Press, 1939.
9. Savage L. The Foundations of Statistics, New York: Wiley, 1954.
10. Goodfellow I., Bengio Y., Courville A., Bengio Y. Deep learning (Vol. 1). Cambridge: MIT press, 2016.
11. Erli S. Artificial Intelligence for scalable personification. Open Systems. 2018. N 1. P. 20–24. [in Russian]

12. Rogushina J.V., Gladun A.Y., Osadchy V.V., Pryima S.M. Ontological Analysis for Web. Melitopol: Bogdan Hmelnitsky MDUPU. 2015. 407 p. [in Ukrainian]
13. Quinlan J.R. Discovery rules from large collections of examples: a case study. Expert *Systems in the Microelectronic Age*. Edinburg, 1979. P. 87–102.
14. Rogushina J.V., Grishanova I.Y. Use of inductive inference method for improvement of ontology of search domain. System research and information technologies. 2007. N 1. P. 62–70. [in Russian]
15. Gladun A.Y., Rogushina J.V. Data Mining: retrieval of knowlegde into data. K.: ADEF-Ukraine. 2016. 452 p. [in Ukrainian]
16. Breiman L. Bagging predictors. Machine Learning. 1994. 24(2). P. 123–140.
17. Baclawski K., Bennett M., Berg-Cross G., Fritzsche D., Schneider T., Sharma R., Westerninen A. Ontology Summit 2017 communiqué–AI, learning, reasoning and ontologies. Applied Ontology. (Preprint). 2018. P. 1–16. http://www.ccs.neu.edu/home/kenb/pub/2017/09/public.pdf.
18. Rogushina J.V. Use of semantic properties of the Wiki resources for expansion of functional posibilities of "Great Ukrainian Encyclopedia". Encyclopaedias in the modern information space: collective monograph / Ed. Kirillon A.M. Kyiv. 2017. P. 104–115. [in Ukrainian]
19. Rogushina J. Semantic Wiki resources and their use for the construction of personalized ontologies. CEUR Workshop Proceedings 1631. 2016. P. 188–195.

Одержано 07.11.2018

Про автора:

Рогушина Юлія Віталіївна,
кандидат фізико-математичних наук,
старший науковий співробітник.
Кількість наукових публікацій в українських виданнях – 140.
Кількість наукових публікацій в зарубіжних виданнях – 30.
Індекс Хірша – 3.
http://orcid.org/0000-0001-7958-2557.

Місце роботи автора:

Інститут програмних систем
НАН України,
03181, Київ-187,
проспект Академіка Глушкова, 40.
Тел.: 066 550 1999.
E-mail: ladamandraka2010@gmail.com