Научная статья
УДК 004.832
doi:10.37614/2949-1215.2023.14.7.002

ПРИВЛЕЧЕНИЕ ДОПОЛНИТЕЛЬНЫХ ЗНАНИЙ О ПРЕДМЕТНОЙ ОБЛАСТИ В ЗАДАЧАХ МАШИННОГО ОБУЧЕНИЯ

Александр Анатольевич Зуенко[1]✉, Ольга Николаевна Зуенко[2]
[1, 2]Институт информатики и математического моделирования имени В. А. Путилова Кольского научного центра Российской академии наук, Апатиты, Россия
[1]zuenko@iimm.ru✉, https://orcid.org/0000-0002-7165-6651
[2]ozuenko@iimm.ru, https://orcid.org/0000-0001-5431-7538

Аннотация

Работа посвящена рассмотрению ряда вопросов, связанных с привлечением дополнительных знаний о предметной области при решении задач машинного обучения. Описываются способы учета подобных знаний на основе модификации классических методов классификации, кластеризации, поиска ассоциативных правил.

Сделан вывод о том, что анализ фоновых знаний способен повысить достоверность и точность классических методов машинного обучения, хотя их модификация с учетом дополнительных ограничений иногда оказывается достаточно трудоемкой процедурой. К тому же в различных типах задач машинного обучения для повышения достоверности и точности их результатов требуются различные типы дополнительных ограничений. Это создает определенные сложности при решении комплексных задач, требующих поэтапного привлечения различных типов дополнительных ограничений.

Ключевые слова:

машинное обучение, задача удовлетворения ограничений, задача классификации, задача кластеризации, задача извлечения ассоциативных правил

Благодарности:

работа выполнена в рамках НИР «Разработка теоретических и организационно-технических основ информационной поддержки управления жизнеспособностью региональных критических инфраструктур Арктической зоны Российской Федерации» (регистрационный номер 122022800547-3).

Для цитирования:

Зуенко А. А., Зуенко О.В. Привлечение дополнительных знаний о предметной области в задачах машинного обучения // Труды Кольского научного центра РАН. Серия: Технические науки. 2023. Т. 14, № 7. С. 16–25. doi:10.37614/2949-1215.2023.14.7.002.

Original article

INVOLVEMENT OF ADDITIONAL KNOWLEDGE ABOUT SUBJECT DOMAIN IN MACHINE LEARNING PROBLEMS

Aleksandr A. Zuenko[1]✉, Olga N. Zuenko[2]
[1, 2]Putilov Institute for Informatics and Mathematical Modeling of the Kola Science Centre of the Russian Academy of Sciences, Apatity, Russia
zuenko@iimm.ru✉, https://orcid.org/ 0000-0002-7165--6651
[2]ozuenko@iimm.ru, https://orcid.org/ 0000-0001-5431-7538

Abstract

The study deals with the concideration of a range of issues related to the involving additional knowledge about subject domain when solving machine learning problems. The techniques for accounting such knowlwdge based on the modification of classical methods of classification, clustering, associative rule discovery are described. It is concluded that the analysis of bachgroung knowledge is able to increase reliability and accuracy of classical machine learning methods, although their modification, taking into account additional constraints, sometimes turns out to be a rather time-consuming procedure. In addition, different types of machine learning problems require different types of additional constraints to increase the reliability and accuracy of their results. This provides certaint difficulties in solving complex problems that require gradual involvement of various types of additional constraints.

Keywords:

machine learning, constraint satisfaction problem, classification problem, clustering problem, associative rule discovery problem

Acknowledgments:
the study was carried out within the framework of the Putilov Institute for Informatics and Mathematical Modeling of the Kola Science Centre of the Russian Academy of Sciences state assignment of the Ministry of Science and Higher Education of the Russian Federation, research topic “Development of theoretical and organizational and technical foundations of information support for managing the viability of regional critical infrastructures of the Arctic zone of the Russian Federation” (registration number of the research topic 122022800547-3).
For citation:
Zuenko A. A., Zuenko O. N. Involvement of additional knowledge about subject domain in machine learning problems // Transactions of the Kola Science Centre of RAS. Series: Engineering Sciences. 2023. Vol. 14, No. 7. P. 16–25. doi:10.37614/2949-1215.2023.14.7.002.

Введение

С 1980-х гг. начали появляться подходы к решению задач машинного обучения, включающие попытки применения в этом процессе некоторых дополнительных знаний. Такое небольшое нововведение послужило причиной смены подходов и существенно повлияло как на моделирование задач, так и на развитие направления машинного обучения в целом [1].

Использование ограничений в процессе извлечения знаний может быть полезно, по крайней мере, по трем причинам [2]:

- фильтрация и организация наборов данных перед применением методов извлечения данных;
- повышение производительности алгоритмов извлечения данных путем снижения пространства поиска и сосредоточении на самом поиске;
- анализ результатов на шаге извлечения для их уточнения и представления улучшенного вида извлеченных моделей.

Далее рассматривается возможности и польза от привлечения дополнительных ограничений при решении задач кластеризации, классификации, извлечения ассоциативных правил из обучающей выборки.

Использование дополнительных знаний в задачах кластеризации

Дополнительные фоновые знания положительно влияют на решение задачи кластерного анализа, повышая его эффективность и точность. *Extra Knowledge* или фоновые знания могут быть представлены различным образом. Это может быть множество помеченных объектов, т. е. объектов, которым присвоена метка класса, в который они должны попасть, связь между объектами, количество кластеров, размер кластеров и т. д.

Задача классического кластерного анализа — это задача разбиения множества объектов на классы, когда какая-либо априорная информация о принадлежности объектов этим классам отсутствует. Задача кластеризации с частичным привлечением учителя использует некоторые фоновые знания из предметной области. При этом количество классов и сами классы неизвестны, но для некоторых пар объектов известно, например, что они попадают или не попадают в один кластер. Поэтому задача называется задачей кластеризации с частичным привлечением учителя (*semi-supervised clustering*).

Часто отнесение двух объектов в один кластер только лишь на основе метрики является семантически некорректной операцией. Основным недостатком большинства существующих методов кластеризации является невозможность учитывать пользовательские ограничения на то, какие объекты обязательно должны или, наоборот, не должны попадать в один кластер. Процесс кластеризации можно сделать гораздо эффективнее, если при отнесении объектов к одному или различным кластерам анализировать не только расстояния между объектами, но и значения их признаков.

Основная идея состоит в том, чтобы использовать фоновые или базовые знания из предметной области. Эти знания предложено представлять в виде пользовательских ограничений, которые могут быть двух уровней:

- ограничения на пары объектов кластеров (*instance-level constraints*);
- ограничения на кластеры (*cluster-level constraints*), указывающие требования к кластерам.

Все методы, учитывающие ограничения на объекты [3], можно разделить на:

- основанные на ограничениях,
- основанные на расстоянии.

Ограничения на пары объектов бывают двух типов [4]: ограничения *must-link* и *cannot-link*. Ограничение *must-link* означает, что два объекта o_i и o_j должны попасть в один кластер. И наоборот: ограничение *cannot-link* гласит, что два объекта o_i и o_j должны быть в разных кластерах.

Ограничения на объекты выглядят простыми, но, по сути, являются очень полезными и эффективными для множества приложений. Их применение зачастую приводит к повышению точности результата. Кроме того, ограничения *must-link* и *cannot-link* также могут использоваться для выражения других пользовательских ограничений.

Еще одним ограничением на объекты является пороговое значение максимального диаметра, которое определяет верхнюю границу диаметра кластера, обозначающую, что между каждой парой объектов кластера расстояние не может превышать эту границу. Это требование может рассматриваться как конъюнкция ограничений *cannot-link* между всеми парами объектов с расстоянием, превышающим границу.

В подходах, основанных на ограничениях, алгоритм кластеризации модифицируется для интеграции попарных ограничений, тогда как в подходах, основанных на расстоянии, изменяется только мера расстояния.

Один из алгоритмов, основанных на ограничениях, является модификацией метода *k-means* [4]. Он интегрирует в себе ограничения *must-link* и *cannot-link*, и на каждой итерации, когда происходит обновление разбиения, все эти ограничения должны удовлетворяться. Но у этого алгоритма есть существенный недостаток, т. к. в нем не предусмотрена процедура возврата. Поэтому алгоритм может не найти никакого разбиения, даже если оно существует. Эту проблему можно обойти, если допустить удовлетворения не всех ограничений, а только большинства из них [5, 6].

Другим способом ввести ограничения на объекты является модификация метрики расстояния или целевой функции [7]. Основная идея заключается в том, что если на два объекта o_i и o_j накладывается ограничение *must-link*, то расстояние между ними может быть меньше обычного, чтобы у них было больше шансов оказаться в одном кластере. Похожий подход применяется и к ограничению *cannot-link*.

Как раз такой подход предлагается в работе [8]. Задача нахождения метрики расстояния здесь формулируется как задача оптимизации. Целевая функция является суммой расстояний между парами объектов ограничения *must-link*. Алгоритм призван минимизировать эту сумму. При этом расстояние между парами объектов ограничения *cannot-link* должно превышать некоторую константу.

Ограничения на кластеры, соответственно, накладывают требования на кластеры. Помимо ограничения на минимальное разбиение, максимальный диаметр, существуют еще ограничения на кластер:

- минимальная мощность (населенность) кластера — это ограничение требует, чтобы число объектов в каждом кластере было не меньше заданной границы α: $|C_c| \geq \alpha, \forall c \in [1,k]$;
- максимальная мощность кластера — требует, чтобы число объектов в каждом кластере было не больше заданной границы β: $|C_c| \leq \beta, \forall c \in [1,k]$;
- ограничение на среднюю населенность кластера требует, чтобы соблюдался баланс, и все кластеры были примерно одного размера, т. е. отношение между размером самого маленького и самого большого кластера было больше заданной границы θ: $\frac{\min_{i\in[1,k]}|C_i|}{\max_{j\in[1,k]}|C_j|} \geq \theta$.

В работе [9] предложен другой тип ограничений для алгоритма кластеризации, который использует относительное сравнение: *x* ближе к *y*, чем к *z*. Данный алгоритм кластеризации исследует лежащую в основе меру различия. Как показывают эксперименты на многомерных текстовых наборах данных, предложенный алгоритм точнее и надежнее, чем аналогичные алгоритмы, использующие ограничения *must-link* и *cannot-link*.

Идея данного алгоритма заключается в том, что объекты могут быть разбиты на кластеры неверно, но при этом они удовлетворяют ограничению *cannot-link*. Пары же, которые удовлетворяют ограничению *must-link*, могут соответствовать разным кластерам. Поэтому вводятся относительные сравнения под названием триплетные ограничения (*triplet constraints*). А также предлагается алгоритм кластеризации, который не только учитывает триплетные ограничения, но и одновременно изучает

меру различия набора данных. Авторы отмечают, что триплетные ограничения дают больше информации о различиях, чем попарные ограничения. Они задаются следующим образом: $d_W^C(x_3^i, x_1^i) > d_W^C(x_2^i, x_1^i)$, где $\tau_i = (x_1^i, x_2^i, x_3^i)$ — это триплет; $i = 1,...,r$.

Еще одним способом применения фоновых знаний является использование небольшого набора помеченных объектов, т. е. объектов, которым присвоена метка кластера, в который они должны попасть. В работе [10] используется множество объектов, которыми «засеиваются» или, иными словами, инициализируются кластеры. А также ограничения, которые генерируются на основании помеченных данных. Удачным образом произведенная инициализация в дальнейшем может помочь алгоритму избежать застревания в локальном оптимуме, поскольку соответствует пользовательскому определению кластеров. Для этого желательно, чтобы помеченные данные представляли все имеющиеся категории. Но не обязательно, поскольку алгоритм кластеризации способен не только группировать данные, но также при необходимости расширять и изменять имеющееся множество категорий, чтобы обеспечить разбиение, которое отражает существующие закономерности в данных.

В таком подходе применяется два алгоритма: *Seeded k-means* и *Constrained k-means*. Для первого алгоритма помеченные объекты используются только на этапе для инициализации кластеров, а затем разбиение обновляется в процессе кластеризации согласно алгоритму *k-means*. Во втором случае они распределяются по назначенным им кластерам и уже не могут менять кластер во время работы *k-means*, а алгоритм выполняет распределение только непомеченных объектов. Выбор между этими двумя алгоритмами делается на основании знаний о шуме, присутствующем в наборе данных.

Использование дополнительных знаний в задачах классификации

Решение традиционной задачи классификации происходит в два шага. На первом строится модель, которая ставит в соответствие каждому объекту заданного множества метку класса. На втором шаге полученная модель используется для классификации новых объектов. Такая классификация может выполняться при помощи деревьев решений.

Покажем, как представление результатов алгоритма интеллектуального анализа данных (классифицирующее дерево решений), представление дополнительных знаний предметной области (ограничений) и выбор метода рассуждения (абдукция) может существенно улучшить поведение и результаты классификации [11].

Абдукция — это вид логического вывода, при котором из факта того, что из A следует B и из наблюдения B, можно вывести A. Абдукцию можно рассматривать как принятие гипотезы в качестве объяснения наблюдаемых фактов в соответствии с известными законами. В последние годы абдуктивный вывод широко изучается и применяется в логическом программировании [12].

Схема абдуктивного логического программирования состоит из трех компонент $\langle P, A, Ic\rangle$, логической программы P, множества базовых абдуктивных гипотез A, которые должны объяснять наблюдения в контексте P, и множества ограничений целостности Ic, которые должны удовлетворяться.

Формально определение абдуктивного объяснения выглядит следующим образом [11].

Пусть $\langle P, A, Ic\rangle$ — абдуктивная схема, G — цель. Тогда абдуктивное объяснение G — это множество $\Delta \subseteq A$ базовых гипотез таких, что:

- $P \cup \Delta \models G$;
- $P \cup \Delta \cup Ic$ совместно.

Данное определение может быть обобщено на исходное множество гипотез Δ_0.

Пусть $\langle P, A, Ic\rangle$ — абдуктивная схема, Δ_0 — множество гипотез, а G — это цель. Тогда Δ является абдуктивным объяснением G, с учетом Δ_0, если $\Delta_0 \cup \Delta$ является абдуктивнм объяснением G.

Необходимо заметить, что заданное множество гипотез Δ_0 должно быть совместным с ограничениями Ic.

Одним из способов использования абдуктивного вывода для решения задачи классификации с недостающей информацией является добавление экспертных знаний о предметной области. Учет знаний о предметной области в стандартных алгоритмах классификации на основе дерева решений может быть непростым и потребовать существенных модификаций этих алгоритмов. С другой

стороны, абдуктивные схемы и существующие их реализации уже оснащены механизмами, которые могут быть непосредственно использованы для представления и обработки знаний, специфичных для предметной области. Для описания знаний предметной области используются ограничения целостности.

Пример 1. Рассмотрим известный пример [13], в котором описан набор объектов, представляющих некоторые ситуации с точки зрения погодных условий, в которых игра в теннис является хорошей идеей. Таблица 1 представляет обучающую выборку по атрибутам {*Погода*, *Температура*, *Влажность*, *Ветер*}. Последний столбец таблицы представляет метки классов каждого примера. На основании этой таблицы строится дерево решений, которое может быть использовано для классификации дальнейших примеров как хороших кандидатов для игры в теннис (класс *Да*) и плохих кандидатов для игры в теннис (класс *Нет*).

Таблица 1

Обучающая выборка для задачи классификации

Погода	Температура	Влажность	Ветер	Класс
Солнечная	Высокая	Высокая	Слабый	Нет
Солнечная	Высокая	Высокая	Сильный	Нет
Пасмурная	Высокая	Высокая	Слабый	Да
Дождливая	Умеренная	Высокая	Слабый	Да
Дождливая	Низкая	Низкая	Слабый	Да
Дождливая	Низкая	Низкая	Сильный	Нет
Пасмурная	Низкая	Низкая	Сильный	Да
Солнечная	Умеренная	Высокая	Слабый	Нет
Солнечная	Низкая	Низкая	Слабый	Да
Дождливая	Умеренная	Низкая	Слабый	Да
Солнечная	Умеренная	Низкая	Сильный	Да
Пасмурная	Умеренная	Высокая	Сильный	Да
Пасмурная	Высокая	Низкая	Слабый	Да
Дождливая	Умеренная	Высокая	Сильный	Нет

Дерево решений для данной задачи будет выглядеть следующим образом (рис. 1).

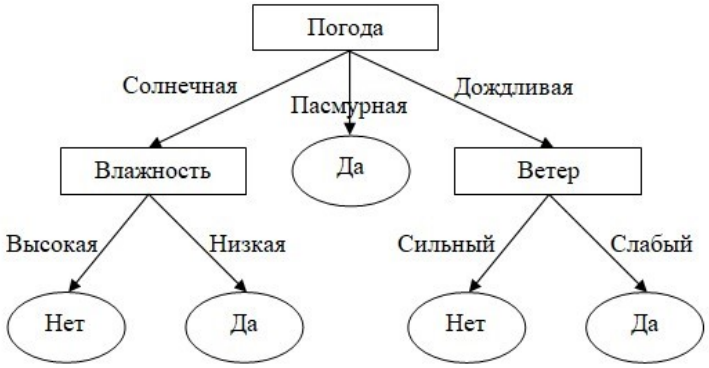

Рис. 1. Дерево решений, полученное из обучающей выборки

Предположим, что каждый раз, когда ветер сильный, влажность низкая {*Влажность* (*Низкая*), *Ветер*(*Сильный*)}. Важно отметить, что такого рода знания могут и не содержаться в обучающей выборке, на основе которой было построено исходное дерево решений. На самом деле примеры в обучающей выборке могут даже противоречить этим знаниям. Действительно, знания могут быть

получены из других источников знаний, например, из прогноза погоды. Предположим теперь, что нужно классифицировать пример e = {*Погода* = *Солнечная*, *Ветер* = *Сильный*}, собственно, это все, что известно, об остальных атрибутах информации нет.

В соответствующей абдуктивной структуре, учитывая начальный набор Δ_e = {*Погода*(*Солнечная*), *Ветер*(*Сильный*)}, класс *Да* имеет абдуктивное объяснение Δ_1 = {*Влажность*(*Низкая*)}, а класс *Нет* имеет абдуктивное объяснение Δ_2 = {*Влажность*(*Высокая*)}. Если теперь рассмотреть приведенное выше ограничение целостности, то абдуктивное объяснение Δ_2 исключается из-за того факта, что полное объяснение, данное $\Delta_e \cup \Delta_2$ = {*Погода* (*Солнечная*), *Ветер* (*Сильный*), *Влажность* (*Высокая*)}, противоречиво. Таким образом, применяя тот же самый вычислительный механизм, получаем правильную классификацию в отношении дополнительных знаний, специфичных для предметной области.

Во многих случаях деревья решений могут содержать вероятностные показатели результата проверки атрибутов/значений. Другими словами, каждая ветвь дерева помечается не только значением, соответствующим атрибуту, обозначающему родительский узел, но и мерой вероятности, которая обозначает, насколько вероятно, что в данном наблюдении атрибут принимает такое значение.

Использование дополнительных знаний
в задачах поиска частых паттернов и ассоциативных правил

Чтобы направить процесс поиска на достижение целей пользователя и сократить лишние паттерны, нужно определить ограничения [14–16]. Самым широко используемым является ограничение на частотность (*minsup*). Приведем пример.

Пример 2. База последовательностей — это множество кортежей (*sid*, *s*), где *sid* — идентификатор последовательности, а *s* — сама последовательность. В таблице 2 представлена база данных, состоящая из четырех последовательностей. Поддержка последовательности s_1 в базе данных обозначается $sup(s_1)$, это число кортежей, содержащих s_1 в базе данных. Например, в табл. 2 $\sup(\langle c\, a\rangle) = 2$.

Таблица 2

База данных последовательностей

Идентификатор последовательности	Последовательность
1	⟨*a b c d a*⟩
2	⟨*d a e*⟩
3	⟨*a b d c*⟩
4	⟨*c a*⟩

Замкнутые частые паттерны обеспечивают минимальное представление частых паттернов, т. е., можно получить все паттерны с точным значением их частотности из замкнутых.

Перечислим наиболее важные ограничения.

- Ограничение на замкнутость [17]. Частый последовательный паттерн *s* является замкнутым, если не существует другого частого последовательного паттерна *s*’ такого, что *s* содержится в *s*’ и *sup*(*s*) = *sup*(*s*’). Например, с *minsup* = 2 паттерн ⟨*b c*⟩ из табл. 2 не является замкнутым, в то время как паттерн ⟨*a b c*⟩ удовлетворяет свойству замкнутости.
- Ограничение на элементы определяет подмножество элементов, которые должны или не должны присутствовать в последовательных паттернах. Например, если наложить ограничение $C_{item} \equiv sup(p) \geq 2 \wedge (a \in p) \wedge (b \in p)$, то получаем три последовательных паттерна из табл. 2: p_1 = ⟨*a b*⟩, p_2 = ⟨*a b c*⟩ и p_3 = ⟨*a b d*⟩.
- Ограничение на длину. Длина паттерна — это количество элементов, входящих в последовательность, которое обозначается *len*(*p*). Например, если $len(p) \geq 3 \wedge sup(p) \geq 2$, то получим только два последовательных паттерна (p_2 и p_3).

• Еще одним широко распространенным ограничением является ограничение на пропуски. Последовательный паттерн с ограничением на пропуски $C_{gap} \equiv [M, N]$, обозначается $p[M, N]$, это такой паттерн, что как минимум M элементов и как максимум N элементов могут находиться между каждыми двумя соседними элементами исходной последовательности. Вернемся к *Примеру 2*: пусть $p[0, 2] = \langle c\ a \rangle$ и $p[1, 2] = \langle c\ a \rangle$, это два паттерна с двумя разными ограничениями на пропуски, и рассмотрим последовательности табл. 2. Последовательности 1 и 4 поддерживают паттерн $p[1, 2]$ (последовательность 1 содержит один элемент между (c) и (a), в то время как последовательность 4 не содержит элементов между (c) и (a)). Но только последовательность 1 поддерживает $p[1, 2]$ (только последовательности с одним или двумя элементами между (c) и (a) поддерживают этот паттерн).

• Ограничение, описываемое с помощью регулярных выражений C_{RE} — это ограничение, определенное как регулярное выражение на множестве элементов. Последовательный паттерн удовлетворяет C_{RE} тогда и только тогда, когда он распознается некоторым детерминированным конечным автоматом [18]. Например, два последовательных паттерна $\langle a\ b\ c \rangle$ и $\langle a\ d\ c \rangle$ из табл. 2 *Примера 2* удовлетворяют ограничению регулярное выражение $C_{RE} = a^*\{bb|bc|dc\}$.

• Ограничение на супер-паттерн находит паттерны, которые содержат определенную пользователем последовательность.

• Ограничение на временной интервал. База данных транзакций содержит информацию о временных метках напротив меток событий. Ограничение на временной интервал или на продолжительность — это множество последовательностей со свойством, обозначающим, что временной интервал между первой и последней транзакцией меньше или больше определенного значения $C_{dur} \equiv Dur(\alpha)\theta\Delta t$, где $\theta \in \{\leq, \geq\}$ и Δt — это заданные целые значения. Длина последовательного паттерна зависит от выбора исследуемого временного интервала. Пусть в $T \in A_t$, t_s — начальное время, а t_e — конечное время для изучения паттернов транзакции. Тогда интервал времени/события для изучения паттернов задается как: t_s–t_e для данной информационной системы S. Если сгруппировать информацию о транзакции $I \subset A_t$, соответствующую одному x_i, то получим альтернативное представление информационной системы S. Если наложить ограничение на временной интервал, то получим базу данных последовательностей с ограничением на временной интервал. Максимальную длину можно контролировать соответствующей настройкой ограничения на временной интервал.

• Совокупное ограничение — это ограничение на совокупность элементов в паттерне, где функция агрегации может быть суммой, средним, максимумом, минимумом, среднеквадратичным отклонением, и т. д. Например, в случае с анализом потребительской корзины покупателя могут интересовать товары, сумма в чеке за которые превышает некоторое заданное значение.

Естественным выводом из частых паттернов являются ассоциативные правила, выражающие ассоциацию между двумя паттернами.

Ассоциативным правилом называется импликация $X \Rightarrow Y$, где $X \subset I$, $Y \subset I$ и $X \cap Y = \varnothing$. Правило $X \Rightarrow Y$ имеет поддержку s, если s процентов транзакций из D содержат $X \cup Y$, $supp(X \Rightarrow Y) = supp(X \cup Y)$. Достоверность правила показывает вероятность того, что из X следует Y. Правило $X \Rightarrow Y$ справедливо с достоверностью c, если c процентов транзакций из D, содержащих X, также содержат $conf(X \Rightarrow Y) = \dfrac{supp(X \cup Y)}{supp(X)}$.

Иными словами, требуется выявить зависимость: если в транзакции присутствует паттерн X, то на основании этого можно сделать вывод, что паттерн Y также должен присутствовать в данной транзакции.

Выбор значений минимальной поддержки и минимальной достоверности имеет большое значение. При очень высокой поддержке алгоритм будет выявлять правила, которые являются слишком очевидными, чтобы на их основе проводить анализ. С другой стороны, очень маленькая поддержка может привести к выявлению слишком большого числа правил, которые могут оказаться статистически необоснованными, что потребует много вычислительных ресурсов. Несмотря на это, наиболее интересные и неожиданные правила зачастую можно выявить именно при низкой поддержке.

Заключение

В работе рассмотрен ряд вопросов, связанных с привлечением дополнительных знаний о предметной области при решении задач машинного обучения. Оказывается, что классические методы классификации, кластеризации, поиска ассоциативных правил плохо приспособлены для учета и анализа дополнительных ограничений экспертов. Модификация подобных методов иногда оказывается достаточно трудоемкой процедурой. Однако модифицированные методы способны существенно повысить достоверность и точность получаемых результатов машинного обучения.

В разных типах задач машинного обучения для повышения точности и достоверности их результатов требуются различные типы дополнительных ограничений. Это создает определенные сложности при решении комплексных проблем, предполагающих для их устранения интеграцию нескольких типов задач машинного обучения, и, соответственно, привлечения разнородных дополнительных ограничений.

Отдельную проблему составляет производительность вычислений методов, анализирующих дополнительные требования экспертов. Разработчики подобных методов, как правило, стремятся к тому, чтобы ограничения служили для снижения размерности пространства поиска.

Список источников

1. Russel S., Norvig P. Artificial Intelligence: A Modern Approach / S. Russel, P. Norvig — 3rd edition. New Jersey: Prentice Hall, 2010. 1132 p.
2. Grossi V., Pedreschi D., Turini F. Data Mining and Constraints: An Overview / V. Grossi, D. Pedreschi, F. Turini // Data Mining and Constraint Programming. Cham: Springer, 2016. P. 25–48.
3. Davidson I., Basu S. A survey of clustering with instance level constraints / I. Davidson, S. Basu // ACM Transactions on Knowledge Discovery from Data. 2007. № 1. P. 1–41.
4. Wagstaff K., Cardie C., Rogers S., Schrödl S. Constrained K-means Clustering with Background Knowledge / K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl // Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001). Williamstown: Williams College, 2001. P. 577–584.
5. Davidson I., Ravi S. Agglomerative hierarchical clustering with constraints: Theoretical and empirical results / I. Davidson, S. Ravi // Knowledge Discovery in Databases: PKDD 2005. Berlin, Heidelberg: Springer, 2005. P. 59–70.
6. Davidson I., Ravi S. S. Clustering With Constraints: Feasibility Issues and the k-Means Algorithm / I. Davidson, S. S. Ravi // Proceedings of the 2005 SIAM International Conference on Data Mining (SDM). 2005.
7. Bilenko M., Basu S., Mooney R. J. Integrating Constraints and Metric Learning in Semi-Supervised Clustering / M. Bilenko, S. Basu, R. J. Mooney // Proceedings of the 21st International Conference on Machine Learning. Alberta: 2004. P. 11–18.
8. Xing E. P., Ng A. Y., Jordan M., Russell S. Distance Metric Learning, With Application To Clustering With Side-Information / E. P. Xing, A. Y. Ng, M. Jordan, S. Russell // Advances in Neural Information Processing Systems. 2003. № 15. P. 505–512.
9. Kumar N., Kummamuru K. Semisupervised Clustering with Metric Learning using Relative Comparisons / N. Kumar, K. Kummamuru // IEEE Transactions on Knowledge and Data Engineering. 2008. № 20. P. 496–503.
10. Basu S., Banerjee A., Mooney R. Semi-supervised Clustering by Seeding // Proceedings of the 19th International Conference on Machine Learning, Sidney, Australia, 2002. pp. 19–26.
11. Atzori M., Mancarella P., Turini F. Abduction in Classification Tasks / M. Atzori, P. Mancarella, F. Turini // AI*IA 2003: Advances in Artificial Intelligence. Berlin, Heidelberg: Springer, 2003. P. 213–224.
12. Kakas A., Kowalski R., Toni F. Abductive Logic Programming / A. Kakas, R. Kowalski, F. Toni // Journal of Logic and Computation. 1992. № 2. P. 719–770.
13. Quinlan J. R. C4.5: Programs for Machine Learning / J. R. Quinlan — San Mateo: Morgan Fauffman Publishers, 1994. 312 p.

14. Зуенко А. А. Метод машинного обучения для выявления замкнутых множеств общих признаков объектов с применением технологии программирования в ограничениях / А. А. Зуенко // Автоматика и телемеханика. 2022. № 12. С. 156–168.
15. Dong G., Pei J. Sequence Data Mining / G. Dong, J. Pei NY: Springer, 2007. 150 p.
16. Zuenko A., Zuenko, O. Frequent Pattern Discovery as Table Constraint Satisfaction Problem / A. Zuenko O. Zuenko // Proceedings of the Sixth International Scientific Conference “Intelligent Information Technologies for Industry” (IITI’22). Cham: Springer, 2023. P. 118–130.
17. Yan X., Han J., Afshar R. Clospan: Mining: Closed sequential patterns in large datasets / X. Yan, J. Han, R. Afshar // SDM. 2003. P. 166–177.
18. Garofalakis M., Rastogi R., Shim K. Mining Sequential Patterns with Regular Expression Constraints / M. Garofalakis, R. Rastogi, K. Shim // IEEE Transactions on Knowledge and Data Engineering. 2002. № 14. P. 530–552.

References

1. Russel S., Norvig P. Artificial Intelligence: A Modern Approach. S. Russel, P. Norvig, Prentice Hall, 2010, 1132 p.
2. Grossi V., Pedreschi D., Turini F. Data Mining and Constraints: An Overview. *Data Mining and Constraint Programming*. Springer, 2016, pp. 25–48.
3. Davidson I., Basu S. A survey of clustering with instance level constraints. *ACM Transactions on Knowledge Discovery from Data*, 2007, no. 1, pp. 1–41.
4. Wagstaff K., Cardie C., Rogers S., Schrödl S. Constrained K-means Clustering with Background Knowledge. Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, 2001, pp. 577–584.
5. Davidson I., Ravi S. Agglomerative hierarchical clustering with constraints: Theoretical and empirical results. Knowledge Discovery in Databases: PKDD 2005, Berlin, Heidelberg, Springer, 2005, pp. 59–70.
6. Davidson I., Ravi S. S. Clustering With Constraints: Feasibility Issues and the k-Means Algorithm. Proceedings of the 2005 SIAM International Conference on Data Mining (SDM), 2005.
7. Bilenko M., Basu S., Mooney R. J. Integrating Constraints and Metric Learning in Semi-Supervised Clustering. Proceedings of the 21st International Conference on Machine Learning, Alberta, 2004, pp. 11–18.
8. Xing E. P., Ng A. Y., Jordan M., Russell S. Distance Metric Learning, With Application To Clustering With Side-Information. *Advances in Neural Information Processing Systems*, 2003, no. 15, pp. 505–512.
9. Kumar N., Kummamuru K. Semisupervised Clustering with Metric Learning using Relative Comparisons. *IEEE Transactions on Knowledge and Data Engineering*, 2008, no. 20, p. 496–503.
10. Basu S., Banerjee A., Mooney R. Semi-supervised Clustering by Seeding, Proceedings of the 19th International Conference on Machine Learning, Sidney, Australia, 2002, pp. 19–26.
11. Atzori M., Mancarella P., Turini F. Abduction in Classification Tasks. AI*IA 2003: Advances in Artificial Intelligence. Berlin, Heidelberg, Springer, 2003, pp. 213–224.
12. Kakas A., Kowalski R., Toni F. Abductive Logic Programming. *Journal of Logic and Computation*, 1992, no. 2, pp. 719–770.
13. Quinlan J. R. C4.5: Programs for Machine Learning, Morgan Fauffman Publishers, 1994.
14. Zuenko A. A. Metod mashinnogo obucheniya dlya vyyavleniya zamknutyh mnozhestv obshchih priznakov ob"ektov s primeneniem tekhnologii programmirovaniya v ogranicheniyah [A Machine Learning Method to Reveal Closed Sets of Common Features of Objects Using Constraint Programming]. *Avtomatika i telemekhanika* [Automation and Remote Control], 2022, no. 12, pp. 156–168. (In Russ.).
15. Dong G., Pei J. Sequence Data Mining. Springer, 2007.
16. Zuenko A., Zuenko, O. Frequent Pattern Discovery as Table Constraint Satisfaction Problem. Proceedings of the Sixth International Scientific Conference “Intelligent Information Technologies for Industry” (IITI’22). Cham, Springer, 2023, pp. 118–130.

17. Yan X., Han J., Afshar R. Clospan: Mining: Closed sequential patterns in large datasets. SDM, 2003, pp. 166–177.
18. Garofalakis M., Rastogi R., Shim K. Mining Sequential Patterns with Regular Expression Constraints. *IEEE Transactions on Knowledge and Data Engineering*. 2002, no. 14, pp. 530–552.

Информация об авторах

А. А. Зуенко — кандидат технических наук, ведущий научный сотрудник;
О. Н. Зуенко — младший научный сотрудник.

Information about the authors

A. A. Zuenko — Candidate of Science (Tech.), Leading Research Fellow;
O. N. Zuenko — Junior Research Fellow.

Статья поступила в редакцию 10.10.2023; одобрена после рецензирования 01.11.2023; принята к публикации 08.11.2023.
The article was submitted 10.10.2023; approved after reviewing 01.11.2023; accepted for publication 08.11.23.