УДК 004.85
DOI: 10.15827/0236-235X.140.660-669
Дата подачи статьи: 11.05.22, после доработки: 22.07.22
2022. Т. 35. № 4. С. 660–669

Алгоритмы генерации обучающих множеств в системе с прецедентным выводом на основе ситуаций-примеров

И.Н. Глухих [1], *д.т.н., профессор*
Д.И. Глухих [1], *аспирант, d.i.glukhikh@utmn.ru*

[1] *Институт математики и компьютерных наук,*
Тюменский государственный университет, г. Тюмень, 625003, Россия

В статье рассматривается проблема создания обучающих множеств и их масштабирования в задачах машинного обучения. Предметом исследования является процесс генерации обучающих множеств на основе примеров в целях их аугментации.

Для реализации идеи расширения предлагается использовать преобразование имеющихся примеров ситуаций. Преобразование примеров осуществляется на основе известного метода оптимизации – метода покоординатного спуска.

Описывается постановка задачи преобразований ситуаций-примеров в терминах введенной модели представлений. Предлагаются алгоритмы, позволяющие из исходного множества ситуаций-примеров, заданных с помощью формальных представлений, получать расширенное множество, которое будет включать в себя ситуации, отвечающие критериям сходства с данными примерами.

В статье представлена апробация предложенных алгоритмов при исследовании нейросетей для отбора ситуаций в системах вывода по прецедентам. Полученные результаты имеют практическую значимость для обучения искусственных нейросетей, используемых в интеллектуальных системах поддержки принятия решений. Предложенные алгоритмы позволяют автоматизировать формирование наборов данных дата-сетов, используя имеющиеся подготовленные и одобренные примеры характерных ситуаций и решая задачу преобразований как задачу поиска оптимума целевой функции схожести.

Ключевые слова: *обучение нейросетей, обучающие данные, case-based reasoning, искусственный интеллект, координатный спуск.*

Метод вывода решений по прецедентам (Case-Based Reasoning, CBR) – один из известных методов искусственного интеллекта, который находит применение при разработке *систем поддержки принятия решений* (СППР) в различных областях [1–3]. В СППР на основе CBR при возникновении проблемной ситуации решение для нее ищется в базе знаний, в которой хранятся прецеденты – пары <Ситуация, Решение>, где Ситуация представляет известную из прошлого опыта проблемную ситуацию, а Решение – то решение, которое считается рациональным или необходимым в этой ситуации. В качестве решения могут выступать набор управляющих воздействий, план мероприятий, программа действий персонала для разрешения проблемной ситуации и др. Если такая база знаний, которая еще называется базой (библиотекой) прецедентов, имеется в CBR-системе, то остается найти в ней ситуацию, которая наиболее похожа на вновь возникшую проблемную ситуацию, чтобы потом выдать пользователю решение из найденной пары. Сравнение и оценка похожести ситуаций является одной из ключевых задач CBR-метода. Для ее решения используются разные способы, среди которых наиболее распространенным, по-видимому, является применение разнообразных метрик в пространстве параметров, которыми описываются ситуации [4].

Вычисление сходства между ситуациями и возможность применения тех или иных метрик определяются их формальным представлением [5, 6].

Однако применение метрик и соответствующих математических вычислений не всегда может обеспечить точную и детальную оценку сходства с тем, чтобы в базе знаний уверенно выбрать ситуацию с «правильным» решением и не выбрать ситуацию с решением «неправильным». Проблема особенно актуальна для комплексных ситуаций, которыми характеризуются сложные объекты, состоящие из многих неоднородных элементов и отношений между ними. В подобных случаях для описания ситуаций приходится использовать большое число параметров разного типа с использованием разных шкал измерений (количественных и качественных), разрабатывая локальные метрики сходства [7], агрегирование которых, в свою

очередь, связано с рисками потери информации и неоднозначностью получаемых результатов.

Такие ситуации возникают, например, в системах эксплуатации сложных технологических объектов (на крупных производствах, в нефтегазодобывающей и перерабатывающей промышленности, на предприятиях городской инфраструктуры), когда при принятии решений необходимо учитывать состояния различных подсистем, связи между ними, а также их операционное окружение, внешнюю среду и др. [8].

Перспективным подходом к определению схожести ситуаций является подход на основе машинного обучения и, в частности, искусственных нейронных сетей, которые уже показали свою работоспособность в ряде научных исследований [7–9].

Однако этот подход требует значительного объема обучающих примеров с известными реакциями, которые должны быть достигнуты обучаемой моделью. В рассматриваемой задаче такие обучающие наборы данных образуются парами сравниваемых ситуаций с метками, в качестве которых выступают количественные оценки их похожести или, в упрощенном случае, качественные оценки вида похож/непохож [7, 9].

Поскольку на практике в реальных системах далеко не всегда есть достаточные по объему наборы обучающих данных, формирование таких наборов – обучающих и валидирующих дата-сетов становится самостоятельной и актуальной научно-практической задачей. Ее решение в условиях недостатка обучающих данных и при отсутствии возможностей сформировать их по наблюдениям за системой связывают с идеей расширения (аугментации) того небольшого числа обучающих примеров, которые уже имеются [10].

В данной работе для реализации идеи расширения предлагается использовать преобразование имеющихся примеров ситуаций, заданных в рамках разработанной ранее модели формальных представлений ситуаций [11]. В этой модели ситуация предметной области трактуется как совокупность состояний, в которых находятся элементы сложного объекта и связи между ними. Формальным представлением ситуации выступает вектор в пространстве состояний или, в случае сложного объекта, набор таких векторов (мультивектор ситуации), каждый из которых соответствует своему элементу сложного объекта или связи между ними. Отдельному компоненту вектора состояний соответствует свое состояние из множества возможных. На этапе идентификации текущей ситуации распознаются состояния элементов сложного объекта и формируются векторы состояний, компоненты которых принимают значение 0 или 1 (при точном распознавании состояний) или в интервале от 0 до 1 (при нечетком распознавании состояний) [12]. Сформированный мультивектор ситуации вместе с другим мультивектором – ситуации из базы знаний подается на вход нейронной сети, которая вычисляет значение функции схожести Sim, то есть определяет степень сходства между двумя ситуациями [13,14].

Задача преобразования ситуаций-примеров

Метод преобразований для создания обучающего набора данных на основе примеров состоит в следующем. Пусть задано множество примеров ситуаций $SIT = \{Sit_r | r = 1, \ldots, R\}$ таких, что любые две различные ситуации из этого множества не являются схожими в смысле некоторой функции похожести ситуаций Sim (.), то есть $Sim\ (Sit_r, Sit_p) < Th$, где Th – порог, после превышения которого можно говорить о схожести ситуаций.

Применительно к базе знаний в системе с выводом на прецедентах это означает, что есть R пар <Ситуация (Sit_r), Решение (Sol_r)>, где все решения отличаются друг от друга. Иначе говоря, признак решения выступает в качестве классификационного признака, разделяющего все множество ситуаций на классы.

Задача – путем преобразований расширить имеющееся множество SIT новыми элементами так, чтобы для каждого r получить множество $SIT_r = \{Sit_r, Sit_1, Sit_2, \ldots\}$, где все элементы – ситуации, похожие с Sit_r в соответствии с заданным критерием схожести.

Введем новый индекс для упрощения записи: $SIT_r = \{Sit_k | k = 1, \ldots, Rr\}$, где $Sit_k \equiv Sit_r$ при $k = 1$.

В общем виде задача преобразований формулируется как следующая оптимизационная задача.

Найти такую Sit', что

$$h(Sit') = Sim(Sit, Sit') \rightarrow \max \quad (1)$$

при $Sit \neq Sit'$.

Ограничение вводится для того, чтобы в процессе решения данной задачи не получить в ответ Sit', полностью совпадающую с Sit, то есть новая ситуация должна отличаться от исходной.

Если *Sit* – вектор в пространстве состояний, то решение этой задачи состоит в переборе компонентов данного вектора, которые могут принимать значение 0 или 1 (в случае точной классификации) или в интервале от 0 до 1 (в случае нечеткой классификации) [8]. Далее рассматриваем только случай точной классификации. Для нечеткой классификации вместо перебора нулей и единиц будут изменяться значения в соответствующей позиции вектора от 0 до 1 с некоторым шагом Δ.

Алгоритмы решения задачи преобразований

В основе предложенных алгоритмов лежит известный метод оптимизации – метод покоординатного спуска, где в качестве исходной точки поиска выступает преобразуемый пример ситуации, точнее, ее вектор в пространстве состояний [14]. Первый алгоритм является базовым, он позволяет получить на выходе один дополнительный вектор *Sit′*, который отвечает критерию наибольшей похожести с исходным примером *Sit*.

Алгоритм 1.

1. Начало
2. Устанавливаем $MAXH = 0$, $Sit_{out} = Sit$
3. Для j от 1 до M делать
// M – число компонентов вектора ситуации в пространстве состояний
{
4. Преобразуем значение j-го компонента вектора ситуации на противоположное
$Sit_{out}[j] = |Sit\,[j] - 1|$
5. Вычисляем $h(Sit_{out})$
6. Если $h(Sit_{out}) > MAXH$ То
{
7. $MAXH = h(Sit')$
8. $Sit' = Sit_{out}$
}
}
9. Вывод Sit'
10. Конец

По окончании работы данного алгоритма получаем такую ситуацию *Sit*′, для которой выполняется $h(Sit') \rightarrow \max$ на всем множестве ситуаций, полученных преобразованием (пересчетом) отдельных компонентов вектора ситуации.

Развитием базового алгоритма является алгоритм 2, на выходе которого выдается не единственный вариант *Sit*′, а некоторое множество ситуаций, удовлетворяющих требованиям сходства с исходным примером *Sit*.

Алгоритм 2.

1. Начало
2. Устанавливаем $Sit_{out} = Sit$
3. Для j от 1 до M делать
// M – число компонентов вектора ситуации в пространстве состояний
{
4. Преобразуем значение j-го компонента вектора ситуации на противоположное
$Sit_{out}[j] = |Sit\,[j] - 1|$
5. Вычисляем $h(Sit_{out})$
6. Если $h(Sit_{out}) > Th$ То
// Включаем Sit_{out} в искомое множество *SIT*, если новая ситуация удовлетворяет порогу сходства и значение $h(Sit_{out})$ в множество меток *SIM*
7. Sit_{out} in *SIT* и $h(Sit_{out})$ in *SIM*
}
8. Упорядочиваем элементы множества *SIT* по значению $h(Sit_{out})$
9. Отбираем из *SIT* первые V элементов (где V – заданное число искомых элементов) и, добавляя в него начальное преобразуемое *Sit*, формируем выходное множество *SIT′* – расширенное множество для ситуации *Sit* и, соответственно, множество меток *SIM′*.
10. Конец

Модификацией этого алгоритма становится проверка на шаге 6 не только условия $h(Sit_{out}) > Th$, но и дополнительных условий-ограничений, например, равенства двух ситуаций по заданному подмножеству позиций (компонентов) в сравниваемых векторах состояний.

Повторение этого алгоритма по всем r позволяет из $SIT = \{Sit_r | r = 1, \ldots, R\}$ получить расширенные множества исходных ситуаций-примеров $\{SIT'_r | r = 1, \ldots, R\}$ и их меток:

$SIT'_r = \{Sit'_k | k = 1, \ldots, Rr\}$,

$SIM'_r = \{Sim_k | k = 1, \ldots, Rr\}$,

где для всех k имеем $Sim(Sit_1, Sit_k) > Th$, то есть все элементы множества ситуаций удовлетворяют требованию сходства с исходной ситуацией (исходная ситуация включена в множество в виде Sit_1).

Вычисление сходства $h(Sit_{out})$

Для оценки степени сходства $h(Sit_{out}) = Sim(Sit, Sit')$ используются экспертный и вычислительный подходы.

При экспертном подходе пары ситуаций предъявляются экспертам и решается задача экспертного оценивания, которая может быть поставлена в различных формулировках. В частности, это прямое оценивание похоже-

сти в терминах «Похоже ($Sim(.) = 1$)/Непохоже ($Sim(.) = 0$)» или на более детальной шкале с промежуточными значениями; оценка возможности применить одно и то же решение для обеих ситуаций «Возможно ($Sim(.) = 1$)/Невозможно ($Sim(.) = 0$)»; оценка принадлежности к одному классу ситуаций и т.п.;

При вычислительном подходе вводится дополнительная функция или набор правил, позволяющие оценить значение $Sim(Sit, Sit')$ путем сравнения векторов двух ситуаций. В частности, это могут быть правила, построенные на основе экспертных знаний и выполняющие классификацию двух ситуаций (оценка схожести в значениях 0 или 1 по принадлежности к одному классу). Более детальной будет оценка схожести векторов состояний, если она вычисляется как отношение одинаковых компонентов двух векторов к их общему числу (с возможностью дальнейшей модификации путем введения дополнительных весов и ограничений).

Стоит отметить, что качественная оценка схожести путем выбора из двух возможных значений 0 или 1 не способна отделить порождаемые ситуации друг от друга по уровню схожести. Такая оценка не позволяет использовать алгоритм 1 и его модификации. Однако, как будет показано далее, в сложных ситуациях, характеризующихся не одним вектором состояний, а набором таких векторов – мультивектором, такая качественная оценка тоже дает работоспособный способ расширения обучающих данных.

Комплексные ситуации на сложном объекте

Рассмотрим случай с комплексной (сложной) ситуацией. Такая ситуация возникает, например, на сложных объектах, которые состоят из множества разнородных элементов [8]. Если каждому из N элементов сложного объекта сопоставить свой вектор в пространстве состояний, то вся комплексная ситуация будет характеризоваться набором из N таких векторов. Далее он будет называться мультивектором. Если до сих пор ситуации Sit соответствовал один вектор в пространстве состояний, то сложной ситуации Sit соответствует более одного такого вектора: $Sit \Leftrightarrow (Sit_1, Sit_2, \ldots, Sit_N)$. Таким же образом новой порождаемой ситуации соответствует $Sit' \Leftrightarrow (Sit'_1, Sit'_2, \ldots, Sit'_N)$.

Теперь задача (1) может быть записана как многокритериальная оптимизационная задача:

$$H(Sit') = (Sim_1, \ldots, Sim_i, \ldots, Sim_N) \rightarrow \max \quad (2)$$

при ограничениях $Sim_k \geq Th_k$ при $k \in K$, где K – множество индексов тех элементов сложного объекта, по которым обязательно должно быть достигнуто сходство не ниже некоторого порога Th_k.

Здесь Sim_i – функция сходства между ситуациями по i-му элементу, $Sim_i = Sim(Sit_i, Sit'_i)$.

В частности, это ограничение может отражать тот факт, что при оценке сходства ситуаций и выборе решений может потребоваться учитывать не только состояние собственно управляемого объекта, но и его контекст, окружение, состояние которого может влиять на решение, но управлять которым невозможно. Чтобы сравнивать ситуации и выбирать решения с учетом данного требования, для элементов такого контекста задается значение $Th_k = 1$.

Для решения задачи (2) очевидным образом может быть использован алгоритм 2, на вход которого подается конкатенация векторов состояний. Тогда в цикле на шаге 3 число M заменяется на $M \times N$ (количество из N векторов по M компонентов), а на шагах 5 и 6 вместо локального сходства $h(.)$ определяется глобальное сходство $H(.)$.

Следующие алгоритмы позволяют дополнить набор инструментов расширения исходных множеств за счет дополнительных возможностей для их комбинирования. Они оперируют результатами применения алгоритма 2 к каждому из i-х векторов в пространстве состояний. Таким образом получаются множества SIT'_i с соответствующими метками – локальными оценками $h(.)$. Далее перебираются комбинации из элементов этих множеств и выполняется отбор этих комбинаций согласно критерию и ограничениям задачи (2).

Алгоритм 3.

1. Начало
2. Для i от 1 до N делать
   {
3. Выполнить Алгоритм 2

// На выходе Алгоритма 2 формируется множество множеств $SIT'_1, SIT'_2, \ldots, SIT'_i, \ldots, SIT'_N$, где $SIT'_i = \{Sit'_{ik} \mid k = 1, \ldots, R_i\}$, R_i – число векторов ситуаций, сгенерированных путем преобразований исходной ситуации и удовлетворяющих критерию схожести (1)

   }

// Устанавливаем начальный набор мультивектора выходной ситуации, в который включаются первые элементы каждого из множеств

4. $Sit_{out} = (Sit'_{i1} \mid i = 1, \ldots, N)$
5. Для i от 1 до N делать

6. Для k от 1 до R_i делать
{
7. $Sit_{out} = (Sit'_{ik})$
8. $H = H(Sit_{out})$
9. Если $H > MAXH$ То
{
10. $MAXH = H$
11. $Sit' = Sit_{out}$
}
}
12. Конец

На выходе алгоритма формируется один мультивектор ситуации Sit', компоненты которого – векторы из множеств SIT'_1, SIT'_2, …, SIT'_i, …, SIT'_N и оценка схожести $H(Sit')$ удовлетворяет критерию (2).

Алгоритм 4 аналогичен алгоритму 2, на его выходе формируется не один мультивектор ситуации, а упорядоченное по критерию (2) множество, каждый из элементов которого является представлением комплексной ситуации, полученной из исходной ситуации Sit и отвечающей требованиям схожести. Чтобы не приводить полное содержание этого алгоритма, покажем только те шаги, которые заменяют шаги 9–11 в алгоритме 3.

Алгоритм 4.

1. Начало
…
9. Если $H > Th$ то
// Если схожесть ситуации с исходной Sit выше принятого порога, то она включается в целевое множество вместе с включением $H(Sit_{out})$ в соответствующее множество меток
10. Sit_{out} in SIT, $H(Sit_{out})$ in SIM
11. Упорядочение элементов SIT по $H(Sit_{out})$ и формирование на выходе SIT', SIM'
12. Конец

В результате работы алгоритма формируется искомое множество ситуаций в их формальном представлении мультивекторами, каждая из которых удовлетворяет требованиям пороговой схожести с исходной ситуацией.

Вычисление сходства $H(Sit)$

Векторное представление целевой функции схожести в (2) требует выбора способа вычисления $H(Sit)$ путем агрегирования локальных оценок сходства Sim_i. Для этих целей используется аддитивная свертка, где весовые коэффициенты α отражают относительную важность для общей оценки сходства двух ситуаций их похожести по i-му элементу сложного объекта:

$$H(Sit') = \sum_{i=1}^{N} \alpha_i Sim_i \quad (3)$$

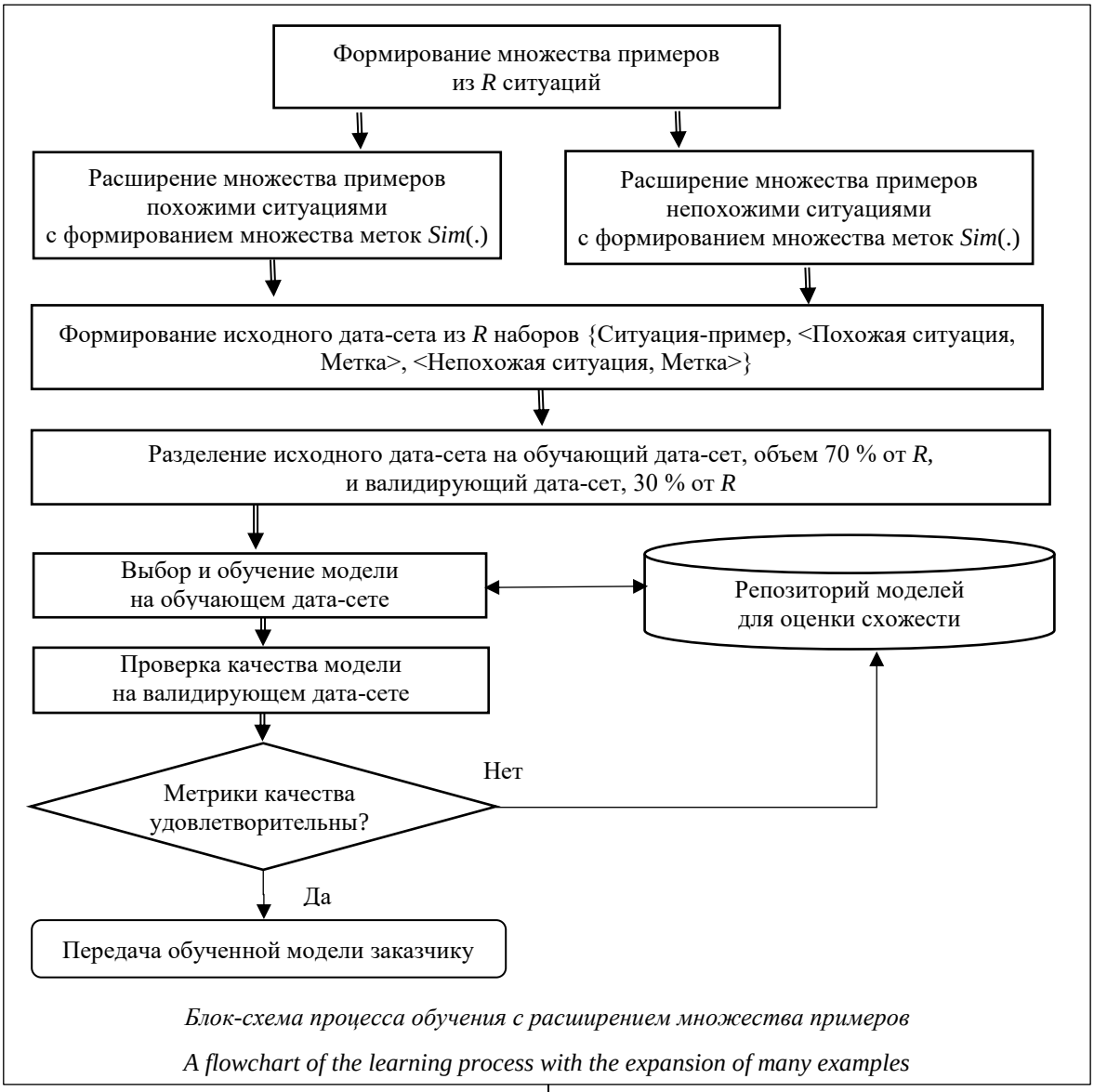
при $\alpha_i \in [0, 1]$ и $\sum \alpha_i = 1$.

Трудность состоит в том, что в сложном объекте весовые коэффициенты формулы (3) не являются постоянными, так как важность одних элементов может зависеть от состояний других. Это учитывается при формировании исходного множества примеров $SIT = \{Sit_r | r = 1, \ldots, R\}$, где полагается, что для каждой ситуации экспертным путем сопоставлен свой вектор весовых коэффициентов, который будет использован в формуле (3). Модификация способа агрегирования частных оценок сходства с помощью (3) использует дополнительную функцию активации: $H(Sit') = f\left(\sum_{i=1}^{N} \alpha_i Sim_i\right)$, где $f(.)$ – функция активации, учитывающая дополнительные условия.

В частности, $f\left(\sum_{i=1}^{N} \alpha_i Sim_i\right) = \sum_{i=1}^{N} \alpha_i Sim_i$ при условии выполнения всех ограничений и $f(.) = 0$ при невыполнении хотя бы одного требования относительно сходства векторов состояний для элементов, входящих в контекст ситуации.

Описанные алгоритмы 2 и 4 служат для формирования множеств ситуаций, являющихся похожими на исходные ситуации относительно порога сходства Th. Для формирования обучающего дата-сета, который также будет включать в себя непохожие ситуации, используются эти же алгоритмы, в которых условия $h(Sit_{out}) > Th$ и $H(Sit_{out}) > Th$ заменяются в соответствующих шагах алгоритма на противоположные $h(Sit_{out}) \le |Th - d|$ и $H(Sit_{out}) \le |Th - d|$, где d – дополнительный коэффициент для более надежного разделения похожих и непохожих ситуаций. Итоговый обучающий дата-сет составляется из полученных расширенных множеств вместе с сохраняемыми оценками сходства Sim их элементов, которые выступают в качестве меток обучающих данных.

Общая блок-схема процесса подготовки дата-сетов и обучения моделей оценки схожести ситуаций, использующая разработанные алгоритмы, приведена на рисунке. Здесь репозиторий предполагает наличие заранее реализованных архитектур (многослойная нейросеть, регрессионная модель, ансамбль моделей и т.п.), которые в процессе обучения настраиваются на вычисление сходства ситуаций и, таким образом, подготавливаются для участия в



Блок-схема процесса обучения с расширением множества примеров

A flowchart of the learning process with the expansion of many examples

процессах отбора ситуаций в предметно-ориентированной CBR-системе. Для оценки качества обученных моделей используются известные метрики из арсенала машинного обучения: MAPE (Mean Absolute Percentage Error) – абсолютная средняя процентная ошибка значения *Sim* [15]; nDCG@k (Normalized Discounted Commulative Gain для лучших *k*) – метрика для оценки верности ранжирования *k* ситуаций лучших по величине *Sim* [16]; Accuracy – метрика для оценки верности определения класса [17], в рассматриваемом случае класс пар ситуаций похож/непохож при заданном пороге *Sim*.

Экспериментальная часть

Описанный подход апробирован при подготовке дата-сетов и проведении исследований моделей оценки схожести ситуаций в работе [8]. Первоначально с помощью экспертов был сформирован набор примеров ситуаций, который далее расширен с помощью предложенных алгоритмов. Программная реализация алгоритмов выполнена посредством языка VBA в среде MS Excel, реализация нейросетевых моделей и их обучение для оценки схожести ситуаций – средствами Python с библиотеками Keras, Scikit-learn.

Для проведения экспериментов был рассмотрен сложный технологический объект – тепловой пункт здания. Технологическая схема представляет собой независимую двухконтурную систему отопления, где внешний теплоноситель через теплообменник передает тепловую энергию теплоносителю системы отопления дома.

Элементы сложного объекта сформированы в группы:

– технологическое оборудование (насос, теплообменник, внутренняя труба);

– обеспечение (ПО, электричество, прочее оборудование);

– персонал (электрик, сантехник, аварийная служба);

– окружение (помещение теплового пункта, соседние здания, природные объекты и явления).

Элементы групп «персонал» и «окружение» не связаны напрямую с объектом, но считаются его частью, поскольку могут влиять на него. Например, снегопад может усложнить доступ персонала к объекту и повлиять на программу действий (то есть решение) при устранении проблемной ситуации.

Представим некоторый пример ситуации на объекте «тепловой пункт здания» через матрицу состояний, отражающую ситуацию на сложном техническом объекте, где единица в столбце соответствует данному состоянию элемента (см. таблицу).

Запишем представление данной ситуации в виде вектора *Sit*, длина которого составляет 112 позиций:
(1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,
1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,
1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,
0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,
0,0,0,0,0,0,0,0,1).

Для обучения нейросетевой модели, которая будет отбирать в базе похожие ситуации, требуется сформировать множество *SIT*, в котором каждая из ситуаций является похожей на исходную *Sit*. С помощью экспертов подобран пример Sit_1, отвечающий критерию сходства с *Sit*. Далее целевое множество получается путем преобразований этого примера с помощью введенных алгоритмов.

Изначально имеется пара векторов *Sit* и Sit_1, второй из которых выглядит следующим образом: (1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,
0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,
0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,
0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,
0,0,1,0,0,0,0,0,0,1,0).

Ситуации отличаются состоянием последнего элемента (Помещение). Схожесть ситуаций *Sim* = 0.952, оценка схожести определялась по методике, описанной в исследовании [8] с учетом нормированного фактора важности каждого элемента.

В результате работы алгоритма 1 получаем дополнительную ситуацию Sit_2 для сравнения с исходной: (1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,
0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,
0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,
0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,
0,0,0,0,1,0,0,0,0,0,0,0,0,0,1).

Новая полученная ситуация имеет с изначальной ситуацией *Sit* такую же схожесть 0.952, однако отличается состоянием другого элемента, который обладает таким же весовым коэффициентом (Природные явления). Для получения большего числа ситуаций использован алгоритм 2, где был принят порог $h(Sit_{out}) > Th = 0{,}85$. В результате применения этого алгоритма получено расширенное мно-

Матрица состояний

The state matrix

Sit_{act}	Исправно	Сломано	В работе	Остановлен	Недоступно	Доступно	Влияет	Не влияет
Ввод	1	0	0	0	0	0	0	0
Внутренняя труба	1	0	0	0	0	0	0	0
Теплообменник	0	1	0	0	0	0	0	0
Насос	0	0	1	0	0	0	0	0
Прочее оборудование	0	0	0	1	0	0	0	0
IT	0	1	0	0	0	0	0	0
Электричество	0	0	0	0	1	0	0	0
Аварийная служба	0	0	0	0	0	1	0	0
Сантехник	0	0	0	0	0	1	0	0
Электрик	0	0	0	0	0	1	0	0
Здания	0	0	0	0	0	0	0	1
Природные объекты	0	0	0	0	0	0	1	0
Природные явления	0	0	0	0	0	0	0	1
Помещение	0	0	0	0	0	0	0	1

жество *SIT*, куда вошли, в частности, такие дополнительные ситуации, схожие с *Sit*:

Sit_3: (1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1); оценка схожести 0.903;

Sit_4: (1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1), оценка схожести 0.879;

Sit_5: (1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1), оценка схожести 0.952.

Таким образом, выполнено расширение набора обучающих векторов для ситуации *Sit*. Далее, задавая с помощью экспертов новые примеры ситуаций и применяя их преобразования, были сформированы обучающий и валидирующий дата-сеты общей размерностью 150 пар похожих и непохожих ситуаций. Результаты работы подтвердили возможность автоматического формирования больших объемов обучающих данных из относительно небольшого числа примеров, для подготовки которых могут привлекаться эксперты.

Заключение

В исследовании предложены алгоритмы, позволяющие из исходного множества ситуаций-примеров, заданных с помощью формальных представлений, получить расширенное множество, которое будет включать в себя ситуации, отвечающие критериям сходства с данными примерами. Применение этих алгоритмов позволяет расширять исходные данные недостаточного объема и создавать дата-сеты, которые будут использоваться для подготовки моделей машинного обучения в задачах оценки схожести ситуаций и выбора решений в СППР с выводом на прецедентах.

Подход к определению схожих ситуаций на основе машинного обучения актуален для сложных, комплексных ситуаций, когда оценка сходства между ними сталкивается с необходимостью определения локальных метрик сходства и трудностями их агрегирования. Результативность и применимость на практике моделей машинного обучения во многом определяются исходными обучающими данными. Предлагаемые результаты позволяют автоматизировать создание дата-сетов, используя для этого подготовленные и одобренные примеры характерных ситуаций и решая задачу преобразований как задачу поиска оптимума целевой функции схожести.

Разработанные алгоритмы не оптимизированы с точки зрения объема вычислений. Их совершенствование для сокращения числа переборов, а также создание новых алгоритмов на основе других методов оптимизации может стать предметом дополнительного исследования предложенных задач.

Исследование выполнено при финансовой поддержке РФФИ и Тюменской области в рамках научного проекта № 20-47-720004.

Литература

1. Башлыков А.А. Применение методов теории прецедентов в системах поддержки принятия решений при управлении трубопроводными системами // Автоматизация, телемеханизация и связь в нефтяной промышленности. 2016. № 1. С. 23–33.

2. Kuzyakov O.N., Andreeva M.A. Applying case-based reasoning method for decision making in IIoT system. Proc. FarEastCon, 2020, pp. 1–5. DOI: 10.1109/FarEastCon50210.2020.9271301.

3. Eremeev A., Varshavskiy P., Alekhin R. Case-based reasoning module for intelligent decision support systems. Proc. I Int. Sci. Conf. IITI, 2016, vol. 1, pp. 207–216. DOI: 10.1007/978-3-319-33609-1_18.

4. Feuillâtre H., Auffret V., Castro M., Lalys F., Le Breton H., Garreau M. Similarity measures and attribute selection for case-based reasoning in transcatheter aortic valve implantation. PLoS ONE, 2020, vol. 15, no. 9, art. e0238463. DOI: 10.1371/journal.pone.0238463.

5. Еремеев А.П., Кожухов А.А., Голенков В.В., Гулякина Н.А. О реализации средств машинного обучения в интеллектуальных системах реального времени // Программные продукты и системы. 2018. Т. 31. № 2. С. 239–245. DOI: 10.15827/0236-235X.122.239-245.

6. Еремеев А.П., Кожухов А.А. Реализация методов обучения с подкреплением на основе темпоральных различий и мультиагентного подхода для интеллектуальных систем реального времени // Программные продукты и системы. 2017. Т. 30. № 1. С. 28–33. DOI: 10.15827/0236-235X.117.028-033.

7. Gabel T., Godehardt E. Top-down induction of similarity measures using similarity clouds. In: Case-Based Reasoning Research and Development, 2015, pp. 149–164. DOI: 10.1007/978-3-319-24586-7_11.

8. Glukhikh I., Glukhikh D. Case-based reasoning with an artificial neural network for decision support in situations at complex technological objects of urban infrastructure. Applied System Innovation, 2021, vol. 4, no. 4, art. 73. DOI: 10.3390/asi4040073.

9. Aamodt A., Plaza E. Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Communications, 1994, vol. 7, no. 1, pp. 39–59. DOI: 10.3233/AIC-1994-7104.

10. Chen H., Birkelund Y., Zhang Q. Data-augmented sequential deep learning for wind power forecasting. Energy Conversion and Management, 2021, vol. 248, art. 114790. DOI: 10.1016/j.enconman.2021.114790.

11. Glukhikh I., Glukhikh D. Case based reasoning for managing urban infrastructure complex technological objects. CEUR Workshop Proceedings, 2021, vol. 2843, no. 038. URL: http://ceur-ws.org/Vol-2843/paper038.pdf (дата обращения: 20.04.2022).

12. Глухих И.Н., Глухих Д.И., Карякин Ю.Е. Представление и отбор ситуаций на сложном технологическом объекте в условиях неопределенности // Вестн. РосНОУ. Сер. Сложные системы модели, анализ и управление. 2021. № 2. С. 65–73.

13. Глухих И.Н., Глухих Д.И., Карякин Ю.Е. Нейросетевая архитектура вывода решений в опасных ситуациях на сложном технологическом объекте // Прикладная информатика. 2021. Т. 16. № 5. DOI: 10.37791/2687-0649-2021-16-5-99-107.

14. Wright S.J. Coordinate descent algorithms. Mathematical Programming, 2015, vol. 151, no. 1, pp. 3–34. DOI: 10.1007/s10107-015-0892-3.

15. de Myttenaere A., Golden B., Le Grand D., Rossi F. Mean absolute percentage error for regression models. Neurocomputing, 2016, vol. 192, pp. 38–48. DOI: 10.1016/j.neucom.2015.12.114.

16. Wang Y., Wang L., Li Y., Di H., Tie-Yan L., Wei Ch. A Theoretical analysis of NDCG type ranking measures. Proc. PMLR, 2013, no. 30, pp. 25–54.

17. Taylor J.R. An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements. University Science Books Publ., Sausalito, CA, 1997, 327 p.

Software & Systems
DOI: 10.15827/0236-235X.140.660-669

Received 11.05.22, Revised 22.07.22
2022, vol. 35, no. 4, pp. 660–669

Algorithms for generating training sets in a system with case-based inference based on example situations

I.N. Glukhikh [1], *Dr.Sc. (Engineering), Professor*
D.I. Glukhikh [1], *Postgraduate Student, d.i.glukhikh@utmn.ru*

[1] *Department of Information Systems, University of Tyumen, Tyumen, 625004, Russian Federation*

Abstract. The paper considers the issue of creating training sets and their scaling in machine learning problems. The subject of the study is the process of generating training sets based on examples in order to augment them.

To implement the idea of expansion, it is proposed to use the transformation of existing examples of situations. The transformation of examples is based on a well-known optimization method - the method of coordinate descent.

The paper describes the statement of the problem of transformations of example situations in terms of the introduced representation model. There are proposed algorithms that make it possible to obtain an extended set from the initial set of example situations specified using formal representations, which will include situations that meet the similarity criteria with these examples.

The paper presents the testing of the proposed algorithms for expanding a set of example situations, carried out in order to form a data set for the studying artificial neural networks. The obtained results are of practical importance for training artificial neural networks used in intelligent decision support systems. The proposed algorithms make it possible to automate the formation of datasets using the available prepared and approved examples of typical situations and solving the transformation problem as the problem of finding the optimum of the similarity objective function.

Keywords: neural network training, training data, case-based reasoning, artificial intelligence, coordinate descent.

Acknowledgements. *The research was funded by RFBR and Tyumen Region, Russia, project no. 20-47-720004.*

References

1. Bashlykov A.A. Precedent theory methods applyed in the systems of decision-making when managing pipeline systems. *Automation, Telemechanization and Communication in Oil Industry*, 2016, no. 1, pp. 23–33 (in Russ.).
2. Kuzyakov O.N., Andreeva M.A. Applying case-based reasoning method for decision making in IIoT system. *Proc. FarEastCon*, 2020, pp. 1–5. DOI: 10.1109/FarEastCon50210.2020.9271301.
3. Eremeev A., Varshavskiy P., Alekhin R. Case-based reasoning module for intelligent decision support systems. *Proc. I Int. Sci. Conf. IITI*, 2016, vol. 1, pp. 207–216. DOI: 10.1007/978-3-319-33609-1_18.
4. Feuillâtre H., Auffret V., Castro M., Lalys F., Le Breton H., Garreau M. Similarity measures and attribute selection for case-based reasoning in transcatheter aortic valve implantation. *PLoS ONE*, 2020, vol. 15, no. 9, art. e0238463. DOI: 10.1371/journal.pone.0238463.
5. Eremeev A.P., Kozhukhov A.A., Golenkov V.V., Gulyakina N.A. On the implementation of machine learning tools in real-time intelligent systems. *Software & Systems*, 2018, vol. 31, no. 2, pp. 239–245. DOI: 10.15827/0236-235X.122.239-245 (in Russ.).
6. Eremeev A.P., Kozhukhov A.A. Implementation of reinforcement learning methods based on temporal differences and a multi-agent approach for real-time intelligent systems. *Software & Systems*, 2017, vol. 30, no. 1, pp. 28–33. DOI: 10.15827/0236-235X.117.028-033 (in Russ.).
7. Gabel T., Godehardt E. Top-down induction of similarity measures using similarity clouds. In: *Case-Based Reasoning Research and Development*, 2015, pp. 149–164. DOI: 10.1007/978-3-319-24586-7_11.
8. Glukhikh I., Glukhikh D. Case-based reasoning with an artificial neural network for decision support in situations at complex technological objects of urban infrastructure. *Applied System Innovation*, 2021, vol. 4, no. 4, art. 73. DOI: 10.3390/asi4040073.
9. Aamodt A., Plaza E. Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Communications*, 1994, vol. 7, no. 1, pp. 39–59. DOI: 10.3233/AIC-1994-7104.
10. Chen H., Birkelund Y., Zhang Q. Data-augmented sequential deep learning for wind power forecasting. *Energy Conversion and Management*, 2021, vol. 248, art. 114790. DOI: 10.1016/j.enconman.2021.114790.
11. Glukhikh I., Glukhikh D. Case based reasoning for managing urban infrastructure complex technological objects. *CEUR Workshop Proceedings*, 2021, vol. 2843, no. 038. Available at: http://ceur-ws.org/Vol-2843/paper038.pdf (accessed April 20, 2022).
12. Glukhikh I.N., Glukhikh D.I., Karyakin Yu.E. Representation and retrieve of the situation on a complex technological object in the uncertainty conditions. *Bull. of RosNOU. Ser. Complex Systems: Models, Analysis, Management*, 2021, no. 2, pp. 65–73 (in Russ.).
13. Glukhikh I.N., Glukhikh D.I., Karyakin Yu.E. Neural network architecture for outputting solutions in dangerous situations at a complex technological facility. *J. of Applied Informatics*, 2021, vol. 16, no. 5, pp. 99–107. DOI: 10.37791/2687-0649-2021-16-5-99-107 (in Russ.).
14. Wright S.J. Coordinate descent algorithms. *Mathematical Programming*, 2015, vol. 151, no. 1, pp. 3–34. DOI: 10.1007/s10107-015-0892-3.
15. de Myttenaere A., Golden B., Le Grand D., Rossi F. Mean absolute percentage error for regression models. *Neurocomputing*, 2016, vol. 192, pp. 38–48. DOI: 10.1016/j.neucom.2015.12.114.
16. Wang Y., Wang L., Li Y., Di H., Tie-Yan L., Wei Ch. A Theoretical analysis of NDCG type ranking measures. *Proc. PMLR*, 2013, no. 30, pp. 25–54.
17. Taylor J.R. *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*. University Science Books Publ., Sausalito, CA, 1997, 327 p.

Для цитирования

Глухих И.Н., Глухих Д.И. Алгоритмы генерации обучающих множеств в системе с прецедентным выводом на основе ситуаций-примеров // Программные продукты и системы. 2022. Т. 35. № 4. С. 660–669. DOI: 10.15827/0236-235X.140.660-669.

For citation

Glukhikh I.N., Glukhikh D.I. Algorithms for generating training sets in a system with case-based inference based on example situations. *Software & Systems*, 2022, vol. 35, no. 4, pp. 660–669 (in Russ.). DOI: 10.15827/0236-235X.140.660-669.