ПРОБЛЕМЫ РАЗРЕШИМОСТИ И ВЫБОРА АЛГОРИТМОВ ДЛЯ ЗАДАЧИ ПРИНЯТИЯ РЕШЕНИЙ ПО ПРЕЦЕДЕНТНОСТИ

В.В. Краснопрошин, В.А. Образцов

Белорусский государственный университет, Минск, Беларусь

ВВЕДЕНИЕ

К принятию решений сводится широкий круг практических и теоретических задач. В постановочном смысле эта задача трактуется довольно неоднозначно. Связано это главным образом с невозможностью формализации понятия *решений*. В методологическом смысле принятие решений – это процесс, результатом которого является решение. Даже если не определять понятие *решения*, то такой методологический взгляд на проблему позволяет просто иначе расставить акценты. При этом проблема трансформируется в последовательность более ясных и однозначно понимаемых проблем. В этой последовательности обязательно присутствует задача вычисления свойства объектов, а само решение может быть просто ассоциировано с таким свойством [21]. В этом случае задачу принятия решений можно свести к вычислению (определению) свойств анализируемой информации. Это, безусловно, один из вариантов задачи принятия решений. В основе этого варианта лежит задача вычисления свойств.

Принятие решений позволяет легко обнаружить во всех задачах вычисления свойств единую природу, привести их на формальном уровне к одной постановке. Какие задачи вычисления свойств имеются в виду? Это хорошо изученные задачи исчисления высказываний, предикатов [1, 2]. Менее изученные, но имеющие больше практических интерпретаций, задачи логической диагностики [3, 4]. И, наконец, просто плохо формализованные задачи теории распознавания образов (ТРО) [5, 9]. Нетрудно видеть, что во всех перечисленных задачах на формальном уровне имеются множества, разбитые на подмножества. Каждое из подмножеств характеризуется некоторым свойством. Проблема заключается в вычислении данного свойства для каждого элемента исходного множества.

Зачем необходимо объединять все перечисленные задачи? Во-первых, из-за их единой постановочной сути. А, во-вторых, из-за необходимости разработки методологии решения плохо формализованных задач (таких как логическая диагностика, ТРО). Дело в том, что в задачах исчисления высказываний (и некоторых других) естественно вводятся понятия *разрешимой задачи*, *обоснованного алгоритма*. Аналогичные понятия для задач ТРО только в рамках самой теории невозможны или требуют привлечения дополнительной информации. Между тем, для задач ТРО тоже необходимо иметь возможность показать разрешимость конкретной задачи или обоснованность выбора алгоритмов, т.к. к постановке ТРО сводится много важных в практическом смысле задач [9, 18].

Указанное объединение и является, по сути, целью данной работы. Оно является средством рассмотрения и исследования задач ТРО. Из всего спектра проблем ТРО в наибольшей степени нас интересуют алгоритмические аспекты *задачи распознавания с обучением* [5, 9]. В контексте сказанного выше, эта задача отнесена к классу проблем принятия решений по прецедентности. Хотя еще раз подчеркнем, что за таким названием стоит только попытка объединения однотипных задач, а терминология может быть иной.

В работе получены следующие основные результаты:

– введено понятие разрешимой задачи и получены некоторые условия разрешимости для случая задания информации по прецедентности;

- подробно исследован один из классов алгоритмов, обеспечивающих выполнение необходимых условий разрешимости, и показано, что полученные результаты являются не улучшаемыми;
- введен и исследован класс алгоритмов индуктивного вывода, который применим для решения всего спектра задач принятия решений и сравним с известными алгоритмами дедуктивного вывода.

Приведенные в работе результаты очень тесно тематически связаны с результатами, опубликованными в [6, 7]. Там же можно ознакомиться с доказательством основных утверждений и теорем.

1. ПОСТАНОВКА ЗАДАЧИ И ОБЩИЕ УСЛОВИЯ ЕЕ РАЗРЕШИМОСТИ

1.1. *Формальная постановка задачи*

Рассмотрим следующую задачу:
на множестве объектов $\boldsymbol{X}$ произвольной природы задано некоторое, возможно и бесконечное, число подмножеств (классов) $\boldsymbol{X}_1,...,\boldsymbol{X}_l$. Требуется указать алгоритм А *(может быть наилучший в некотором смысле), который определен на всем множестве $\boldsymbol{X}$, и результат работы которого для каждого $x \in \boldsymbol{X}$ можно интерпретировать в терминах принадлежности подмножествам $\boldsymbol{X}_i$.*

Сразу оговоримся, что данную задачу мы рассматриваем на формальном, а не содержательном уровне. Поэтому произвольность природы объектов из множества $\boldsymbol{X}$ означает, что некоторая функция кодирования уже выбрана и соответствующие пространства зафиксированы. Также фиксированным является пространство, в котором формируются результаты работы алгоритмов. Ясно, что при этом возникает проблема выбора функций кодирования и интерпретации. И эта проблема тесно связана с разрешимостью сформулированной задачи. Более подробно с характером этой проблемы и методологией ее решения можно ознакомиться в [6].

Обозначим сформулированную задачу через $\boldsymbol{Z}$. По своей постановке эта задача (или точнее – класс задач) находится на максимальном уровне общности. Нетрудно видеть, что к этой постановке сводится большое число задач. Различие между ними заключается в способах задания информации о множестве $\boldsymbol{X}$ и подмножествах $\boldsymbol{X}_1,...,\boldsymbol{X}_l$.

Даже в такой общей постановке можно заметить, что для решения задачи $\boldsymbol{Z}$ необходимо ответить на ряд вопросов. Главным среди них является вопрос о разрешимости задачи $\boldsymbol{Z}$ и возможности обоснования решения. Без удовлетворительного на него ответа решение любых прикладных задач, сводящихся к какому-либо из вариантов задачи $\boldsymbol{Z}$, всегда будет неполноценным. К сожалению, в общем случае возможность обоснования является сомнительной, даже когда речь идет о математическом формализме [2]. Хотя существуют задачи, для которых такая возможность доказана. Это, к примеру, задача классификации формул в исчислении высказываний с методом резолюций в качестве алгоритма А, а также множество технически связанных с ней задач распознавания свойств [8].

Что касается разрешимости задачи $\boldsymbol{Z}$, то возможность обоснования является одним из достаточных условий. Необходимые же условия определяются в основном выразительной силой языка формализации и постановкой конкретной задачи. Чтобы описать такие условия, необходимо провести классификацию задач, соответствующих данной постановке.

Классификация, в частности, может быть проведена по следующим основаниям:
- является ли число классов l и множество $\boldsymbol{X}$ *конечным* или *бесконечным*;
- *полностью* ли *описаны* классы $\boldsymbol{X}_1,...,\boldsymbol{X}_l$, и все ли множество $\boldsymbol{X}$ они *покрывают*;
- какой способ *описания объектов* и *функции принадлежности* выбран;

– что известно об объектах, образующих классы $\boldsymbol{X}_1,...,\boldsymbol{X}_l$.

Перечисленные основания носят “информационный” характер, что вполне соответствует логике постановки любой задачи, согласно которой алгоритмы должны быть следствием формализации и полностью определяться характером и объемом известной информации.

Рассмотрим примеры некоторых задач.

Задача $\boldsymbol{Z}_A$. *Классификация формул в исчислении высказываний* [2].

Число классов конечно (l = 3), а множество $\boldsymbol{X}$ *бесконечно. Способ описания классов – через аксиомы и правила вывода без кванторов. Классы* $\boldsymbol{X}_i$. *описаны полностью и покрывают все множество* $\boldsymbol{X}$. *И, наконец, для аксиом принадлежность задана в явном виде.*

Такая задача выбора решается методом резолюций [2]. Алгоритм решения обозначим через A_0.

Задача $\boldsymbol{Z}_B$. *Логическая диагностика* [3, 4].

Число классов l конечно, а множество $\boldsymbol{X}$ *может быть конечным либо бесконечным. Классы* $\boldsymbol{X}_i$ *описаны не полностью и/или не покрывают все множество* $\boldsymbol{X}$. *Способ описания классов – через предикаты (правила), которые одновременно используются для описания объектов и функции принадлежности. Информация о принадлежности объектов, удовлетворяющих правилам, считается заданной.*

В зависимости от языка формализации для решения данной задачи обычно используется один из вариантов алгоритма A_0 (параметризованный).

Задача $\boldsymbol{Z}_C$. *Распознавание образов с обучением* [5].

Число классов l конечно, а множество $\boldsymbol{X}$ *может быть конечным либо бесконечным. Классы* $\boldsymbol{X}_i$ *описаны не полностью и/или не покрывают все множество* $\boldsymbol{X}$. *Способ описания классов – по прецедентности (т.е. явным указанием объектов, принадлежащих каждому из классов* $\boldsymbol{X}_1,...,\boldsymbol{X}_l$*).*

В этой задаче наблюдается большое разнообразие алгоритмов. Техника их построения зависит от выбранного языка формализации, отношения к информации и некоторых дополнительных предположений.

В постановочном плане общность задач $\boldsymbol{Z}_{A\text{-}C}$ не вызывает сомнения – все они варианты сформулированной выше задачи $\boldsymbol{Z}$. Но в техническом (алгоритмическом) смысле они различны, что также относится и к проблеме разрешимости. В задаче $\boldsymbol{Z}_A$ разрешимость эквивалентна обоснованности алгоритма. В $\boldsymbol{Z}_B$ схема алгоритма аналогична. Но из-за параметризации, которая необходима для работы алгоритма вне классов $\boldsymbol{X}_1,...,\boldsymbol{X}_l$ или просто на объектах с априори неизвестной принадлежностью, обоснованной можно считать только схему. Сам же алгоритм, даже в случае редукции информации к условиям $\boldsymbol{Z}_A$, только в определенных случаях редуцируется в A_0. И, наконец, в $\boldsymbol{Z}_C$ соответствующий ей алгоритм вообще не имеет никакого отношения к $\mathbf{A}_0$. Поэтому говорить о какой-либо разрешимости задач $\boldsymbol{Z}_B$ и $\boldsymbol{Z}_C$ очень сложно.

Введем теперь понятие разрешимой задачи $\boldsymbol{Z}$. Для этого нам потребуются некоторые дополнительные обозначения и определения.

Исходя из постановки задачи, требуется механизм, позволяющий отличать объекты различных классов. Для этого введем систему предикатов $\mathrm{P}=(\mathrm{P}_1,...,\mathrm{P}_l)$ и определим ее следующим образом

$$\forall x \in \boldsymbol{\mathcal{X}} \quad (\mathrm{P}_i(x) \ \in \ \{0,1\} \ \wedge \ (\mathrm{P}_i(x) = 1 \quad \Leftrightarrow \quad x \in \boldsymbol{\mathcal{X}}_i)).$$

Если теперь обозначить через $I(\boldsymbol{\mathcal{X}})$ - некоторую информацию о множестве $\boldsymbol{\mathcal{X}}$ в задаче $\boldsymbol{Z}$, то любой алгоритм А, решающий задачи $\boldsymbol{Z}_{\boldsymbol{A}\text{ - }\boldsymbol{C}}$, можно представить в виде

$$\forall x \in \boldsymbol{\mathcal{X}} \quad (\mathrm{A}\colon\ x \times I(\boldsymbol{\mathcal{X}}) \ \to \ (\mathrm{P}_1^A(x),\ldots,\mathrm{P}_l^A(x))) \tag{1}$$

где $\mathrm{P}^A = (\mathrm{P}_1^A,\ldots,\mathrm{P}_l^A)$ также можно назвать предикатом, но *алгоритмическим*. В отличие от Р он вычисляется алгоритмом А и его значения выбирается из интервала [*0,1*].

Ясно, что на множестве всевозможных алгоритмов вида (1) может быть определен порядок. Для формализации требований к алгоритмам А, которые должны решать задачу $\boldsymbol{Z}$ определим l подмножеств:

$$\boldsymbol{\mathcal{X}}_i^{\mathrm{A}} = \left\{ x \in \boldsymbol{\mathcal{X}} : \mathrm{P}_i^{\mathrm{A}}(x) = 1 \right\}.$$

Очевидно, что $\boldsymbol{\mathcal{X}}_i^{\mathrm{A}}$ могут отличаться от $\boldsymbol{\mathcal{X}}_i$. Для характеризации такого отличия введем монотонные функции

$$\mu:\ \boldsymbol{\mathcal{X}}_i \times \boldsymbol{\mathcal{X}}_i^{\mathrm{A}} \ \to \ [0,1], \qquad \psi:[0,1]^l \ \to \ [0,1]$$

и потребуем, чтобы они удовлетворяли ограничениям

$$\mu(\boldsymbol{\mathcal{X}}_i,\boldsymbol{\mathcal{X}}_i^{\mathrm{A}}) = \begin{cases} 1, & if \quad \boldsymbol{\mathcal{X}}_i = \boldsymbol{\mathcal{X}}_i^{\mathrm{A}}, \\ 0, & if \quad \boldsymbol{\mathcal{X}}_i \cap \boldsymbol{\mathcal{X}}_i^{\mathrm{A}} = \varnothing, \end{cases} \qquad \psi(a_1,\ldots,a_l) = \begin{cases} 1, if \ a_1 = \ldots = a_l = 1, \\ 0, if \ a_1 = \ldots = a_l = 0. \end{cases}$$

Тогда $\Phi_{\mathrm{A}}(\boldsymbol{\mathcal{X}}) := \psi(\mu(\boldsymbol{\mathcal{X}}_1,\boldsymbol{\mathcal{X}}_1^{\mathrm{A}}),\ldots,\mu(\boldsymbol{\mathcal{X}}_l,\boldsymbol{\mathcal{X}}_l^{\mathrm{A}}))$ можно назвать *функционалом качества* алгоритма А, т.к. его значения легко интерпретируются в терминах несовпадения Р и P^A.

Задачу $\boldsymbol{Z}$, для которой $\Phi_{\mathrm{A}}(\boldsymbol{\mathcal{X}}) = 1$ назовем *разрешимой*. Соответствующий алгоритм, на котором достигается условие разрешимости, естественно назвать *обоснованным*.

Среди перечисленных выше задач $\boldsymbol{Z}_{\boldsymbol{A}\text{ - }\boldsymbol{C}}$, только задача $\boldsymbol{Z}_{\boldsymbol{A}}$ является разрешимой. Разрешима она, в частности, алгоритмом A_0, который в контексте введенных понятий является обоснованным. Тот же алгоритм A_0, но использованный для решения какой-то задачи $\boldsymbol{Z}_{\boldsymbol{B}}$, очевидно, становится уже не обоснованным. Так что понятия разрешимости и обоснованности очень тесно связаны.

В общем случае, можно сказать, что целью решения любой задачи из $\boldsymbol{Z}$ является доказательство ее разрешимости. А в качестве средства, при этом, могут быть использованы обоснованные алгоритмы.

1.2. *Разрешимость задачи* $\boldsymbol{Z}_{\boldsymbol{C}}$

Как уже было сказано во введении, основной целью данной работы является исследование задач $\boldsymbol{Z}_{\boldsymbol{C}}$. Исследование на предмет возможности обоснования и алгоритмизации. Для этого необходимо конкретизировать постановку задачи.

Чтобы исключить апелляцию к содержательным аспектам задачи (например, *множество объектов* $\boldsymbol{\mathcal{X}}$ *произвольной природы* в постановке задачи $\boldsymbol{Z}$), в последующем будем рассматривать только формальный уровень постановки. Исходя из этого, ниже мы будем обозначать множество объектов через X, а классы - $\mathrm{X}_1,\ldots,\mathrm{X}_l$. Ясно, что это изменение в обозначениях никак не влияет на постановку задачи и введенные в п. 1.1 определения. Далее, конкретизируем, что подразумевается в постановке задачи $\boldsymbol{Z}_{\boldsymbol{C}}$ под *явным указанием объектов,* *принадлежащих каждому из классов* $\mathrm{X}_1,...,\mathrm{X}_l$. Обычно под таким множеством объектов подразумевается конечная выборка объектов $x \in \mathrm{X}$, задающая информацию $I(\mathrm{X})$ об X. Обозначим через X^0 такую выборку. Называют такие выборки в литературе по-разному [5, 9, 20]. Но чаще всего - *обучающими*, т.к. их используют для построения алгоритмов вида:

$$\forall x \in \mathrm{X} \ \ (\mathrm{A} : x \times \mathrm{X}^0 \ \rightarrow \ (\mathrm{P}_1^A(x),...,\mathrm{P}_l^A(x))) \tag{2}$$

Нетрудно заметить, что функционал Φ_A в общем случае на всем множестве X для задачи $\boldsymbol{Z_C}$ не вычисляем, т.к. по постановке $I(\mathrm{X}) = \mathrm{X}^0$. В то же время его значение легко вычислить на обучающей выборке X^0 (по определению). Но этого недостаточно, т.к. разрешимость связана с условием

$$\Phi_\mathrm{A}(\mathrm{X}) = 1 \tag{3}$$

Иными словами, алгоритм A вида (2) в задаче $\boldsymbol{Z_C}$ должен удовлетворять условию (3) или быть в каком-то смысле наиболее близким к нему, если только нас интересует разрешимость конкретной задачи $\boldsymbol{Z_C}$. Но это нас приводит к рассмотрению следующего вопроса: можно ли решать задачу $\boldsymbol{Z_C}$ и требовать при этом от алгоритма (2), чтобы он удовлетворял, скажем, условию (3)?

Нетрудно заметить, что смысл ответа на последний вопрос заключается в определении соотношения между значениями функционалов $\Phi_\mathrm{A}(\mathrm{X})$ и $\Phi_\mathrm{A}(\mathrm{X}^0)$. Но такое соотношение очевидно

$$\forall \mathrm{A} \ \ (\Phi_\mathrm{A}(\mathrm{X}) < 1 \Leftrightarrow \exists \mathrm{X}^0 \ \Phi_\mathrm{A}(\mathrm{X}^0) < 1) \tag{4}$$

Соотношение (4) по статусу можно считать аксиомой (ее естественно было бы назвать *аксиомой редукции*).

Из данной аксиомы можно вывести условия, которым должен удовлетворять функционал качества Φ_A, чтобы аксиома была непротиворечивой, и могла использоваться для определения соответствующих условий на алгоритмы (2). Для этого мы сформулируем следующее

Утверждение 1. *Соотношение (4) имеет место только в том случае, если независимо от выбора алгоритма* A *функционал* Φ_A *удовлетворяет условиям:*

$$\forall \mathrm{X}^0 \subseteq \mathrm{X} \, \forall \widetilde{\mathrm{X}}^0 \subset \mathrm{X}^0 \, (\Phi_\mathrm{A}(\mathrm{X}^0) = 1 \Rightarrow \Phi_\mathrm{A}(\widetilde{\mathrm{X}}^0) = 1) \tag{5}$$

$$\forall \mathrm{X}^0, \widetilde{\mathrm{X}}^0 \subseteq \mathrm{X} \, (\Phi_\mathrm{A}(\mathrm{X}^0) = 1 \wedge \Phi_\mathrm{A}(\widetilde{\mathrm{X}}^0) = 1 \Rightarrow \Phi_\mathrm{A}(\mathrm{X}^0 \cup \widetilde{\mathrm{X}}^0) = 1) \tag{6}$$

В дальнейшем, без дополнительных оговорок, будем полагать, что Φ_A удовлетворяет условиям утверждения 1.

Фактически, условия (5), (6) на функционал Φ_A означают инвариантность последнего относительно некоторых операций на множестве X. В случае (5) - операции пересечения подмножеств, а в случае (6) - объединения. Они определяют условия на множество функционалов качества, при которых вообще можно ставить вопрос о разрешимости задачи.

Преобразуем теперь аксиому редукции с помощью эквивалентных преобразований к дизъюнктивному представлению

$$\forall \mathrm{A} \, ((\Phi_\mathrm{A}(\mathrm{X}) = 1 \wedge \forall \mathrm{X}^0 \, \Phi_\mathrm{A}(\mathrm{X}^0) = 1) \vee (\Phi_\mathrm{A}(\mathrm{X}) < 1 \wedge \exists \mathrm{X}^0 \, \Phi_\mathrm{A}(\mathrm{X}^0) < 1)) \tag{7}$$

Из данного представления нетрудно получить двойственную форму аксиомы

$$\forall \mathrm{A}\ (\Phi_{\mathrm{A}}(\mathrm{X})=1 \Leftrightarrow \forall \mathrm{X}^{0}\ \Phi_{\mathrm{A}}(\mathrm{X}^{0})=1) \tag{8}$$

Дизъюнктивное представление показывает, что в процессе решения задачи $\boldsymbol{Z_C}$ у нас есть всего две возможности - может иметь место только один из конъюнктов в (7). А каждый из алгоритмов, который решает сформулированную выше задачу, удовлетворяет одному и только одному из двух условий - либо (3), либо, к примеру, следующему

$$\sup_{\{\mathrm{A}\}} \Phi_{\mathrm{A}}(\mathrm{X}) \tag{9}$$

Причем, задачу построения алгоритма (3) нельзя рассматривать как предельный случай задачи (9). А, кроме того, близость алгоритмов может быть определена по-другому.

Для решения каждой из этих задач мы располагаем конечной выборкой X^{0}, с помощью которой определяются ограничения на выбор соответствующего алгоритма А. При решении (9) в большинстве случаев используются т.н. предельные схемы, которые основаны на доказательстве (а чаще всего - предположении), что имеет место

$$\Phi_{\mathrm{A}}(\mathrm{X})=\lim_{\mathrm{X}^{0}\to \mathrm{X}} \Phi_{\mathrm{A}}(\mathrm{X}^{0}).$$

Далее задача (9) подменяется задачей поиска алгоритма, экстремального на выборке X^{0} и полученный алгоритм принимается в качестве решения для (9). Возможность использования данного приема обусловлена тем, что отличие построенного и искомого алгоритмов, по крайней мере, контролируется и может быть сделано сколь угодно незначительным. А, кроме того, такая предельная зависимость между значениями функционалов $\Phi_{\mathrm{A}}(\mathrm{X})$ и $\Phi_{\mathrm{A}}(\mathrm{X}^{0})$ представляет собой типичные достаточные условия непротиворечивости аксиомы редукции в расширенном варианте. Следует отметить, что до определенного методологического завершения использование предельных схем доведено в рамках статистического подхода [5, 10], теории образов [11] (также статистической по своей сути и технике). Это вполне объяснимо, т.к. математическая статистика базируется и на частотной концепции, в рамках которой содержится аппарат для исследования всевозможных предельных схем.

Вернемся теперь к рассмотрению задачи (3). В отличие от (9), ее решение может быть получено путем использования аксиомы редукции в форме (8). Приведем некоторые достаточные условия непротиворечивости последней. Для этого нам потребуется следующее определение:

Выборку $\widetilde{\mathrm{X}}^{0}\subseteq \mathrm{X}$ назовем *представительной* относительно алгоритма А для $\boldsymbol{Z_C}$, если

$$\forall x_1\notin\widetilde{\mathrm{X}}^{0}\ \exists x_2\in\widetilde{\mathrm{X}}^{0}\ ((\mathrm{P}(x_1)=\mathrm{P}(x_2)) \Leftrightarrow (\mathrm{P}^{\mathrm{A}}(x_1)=\mathrm{P}^{\mathrm{A}}(x_2))).$$

Нетрудно убедиться, что имеет место

Утверждение 2. *Пусть* $\widetilde{\mathrm{X}}^{0}\subseteq \mathrm{X}$ *- произвольная конечная и представительная относительно некоторого алгоритма* А *выборка из* X. *Тогда*

$$\Phi_{\mathrm{A}}(\widetilde{\mathrm{X}}^{0})=1 \Rightarrow \Phi_{\mathrm{A}}(\mathrm{X})=1.$$

Подобное достаточное условие является не единственным. Рассмотрим еще одно, столь же очевидное.

Алгоритм А назовем *компетентным* на выборке $\mathrm{X}^{0}\subseteq \mathrm{X}$, если

$$\forall x\notin \mathrm{X}^{0}\ (\mathrm{P}_1^{\mathrm{A}}(x)=\ldots=\mathrm{P}_l^{\mathrm{A}}(x)=0).$$

Для таких алгоритмов может быть доказано следующее

Утверждение 3. *Пусть* $X_1^0, X_2^0, \ldots$ *- некоторая, может быть и бесконечная, последовательность выборок из* $\mathbf{X}$ *таких, что*

$$\forall i, j \ (i \neq j \Rightarrow X_i^0 \cap X_j^0 = \varnothing), \ \bigcup_i X_i^0 = X,$$

а $A_1, A_2, \ldots$ *- соответствующая им последовательность компетентных алгоритмов. Тогда для алгоритма* $A := (\sum_i P_1^{A_i}, \ldots, \sum_i P_l^{A_i})$ *справедливо*

$$\forall X_i^0 \ (\Phi_{A_i}(X_i^0) = 1) \Rightarrow \Phi_A(X) = 1.$$

Непосредственным обобщением и одновременно следствием утверждений 2, 3 является

Утверждение 4. *Пусть* $X_1^0, X_2^0, \ldots$ *- некоторая, может быть и бесконечная, последовательность выборок из* X *таких, что*

$$\forall i, j \ (i \neq j \Rightarrow X_i^0 \cap X_j^0 = \varnothing), \ \bigcup_i X_i^0 = X,$$

а $\tilde{X}_1^0, \tilde{X}_2^0, \ldots$ *- выборки, для которых имеют место:* $\forall i (\tilde{X}_i^0 \subseteq X_i^0)$ *и каждая* $\tilde{X}_i^0$ *является представительной относительно некоторого алгоритма* A_i *для* X_i^0 *. Тогда, если алгоритмы* $A_1, A_2, \ldots$ *являются компетентными на выборках* $X_1^0, X_2^0, \ldots$ *соответственно, то для алгоритма* $A := (\sum_i P_1^{A_i}, \ldots, \sum_i P_l^{A_i})$ *справедливо*

$$\forall \tilde{X}_i^0 \ (\Phi_{A_i}(\tilde{X}_i^0) = 1) \Rightarrow \Phi_A(X) = 1.$$

Обсудим теперь полученные результаты. По всей видимости, существуют и другие достаточные условия разрешимости задачи (3). В нашем же случае общим для всех результатов является следующее условие на алгоритм $A: \Phi_A(\tilde{X}^0) = 1$ (либо какая-то его разновидность). Такие алгоритмы введены в практику ТРО достаточно давно [9, 19, 20] и называются они *корректными*. Таким образом, свойство корректности на заданной выборке оказывается одним из достаточных условий. На данном условии может быть основан выбор алгоритма при решении задачи (3). Из (8) следует также, что корректность алгоритма безотносительно к выборке является и необходимым условием. Поэтому, какие бы в последующем достаточные условия ни были получены, они неизбежно будут связаны с данным свойством.

Полученные выше достаточные условия можно рассматривать также с точки зрения требований к информации. В этом смысле представительность выборки более “сильное“ требование, чем существование разбиения множества X. Наверное поэтому в последнем случае условие корректности алгоритма дополняется компетентностью, т.е. за счет “понижения” требований к информации происходит “повышение” требований к алгоритмам, но суммарная сложность решения задачи, по всей видимости, сохраняется.

Обратимся теперь к проблеме построения представительных выборок. Надо сказать, что это труднейшая проблема не только ТРО, но и индуктивного вывода в целом [8, 12-16]. За счет этого, кстати, задача (3) сохраняет свою индуктивную природу, даже, несмотря на то, что проблема выбора алгоритмов при фиксированной выборке решается дедуктивными средствами. Обсуждаемая же проблема может быть решена путем характеризации таких выборок. К примеру, без труда может быть доказано следующее

Свойство. *Выборка* $\tilde{X}^0 \subseteq X$ *является представительной относительно некоторого алгоритма* A *для* X**,** *только если она обладает свойством*

$$\forall i \in \{1,...,l\}(\widetilde{X}^0 \bigcap X_i \neq \varnothing).$$

Содержательно данное свойство означает: выборка может быть представительной только при условии, что она включает прецеденты из всех классов. Столь же легко могут быть получены и некоторые другие свойства таких выборок. Но полностью эта проблема может быть решена не раньше, чем будет хоть в каком-то смысле обоснован индуктивный вывод [12].

1.3. *Некоторые недостатки формализации*

Введенные выше понятия позволяют формализовать условия разрешимости задачи $\boldsymbol{Z_C}$ и подойти к пониманию обоснованности алгоритмов (2). Но, как это обычно бывает в математике, "излишняя" формализация, целью которой является точность в изложении, неизбежно приводит к неоднозначности.

Проанализируем введенный выше язык формализации. Для этого временно ограничимся случаем $l = 2$. Каждому алгоритму вида (2) поставим в соответствие алгоритм A':

$$\forall x \in X \;\; (A' : x \times X^0 \;\rightarrow\; (P_1^{A'}(x), P_2^{A'}(x))),$$

и его результаты определим следующим образом:

$$P_i^{A'}(x) = 1 - P_i^A(x).$$

Нетрудно заметить, что алгоритмы A и A' просто инверсные. Рассмотрим теперь соотношение между соответствующими значениями функционалов $\Phi_A(X)$ и $\Phi_{A'}(X)$. Очевидно, что независимо от множества X и условий, налагаемых на функционал Ф (кроме условий (5), (6)), соотношение имеет вид

$$\Phi_A(X) = 1 - \Phi_{A'}(X), \text{ при } \Phi_A(X) \in \{0,1\}.$$

А при аддитивности Ф (например, для таких функционалов как *доля правильных прогнозов* [9]), даже при условии $\Phi_A(X) \in [0,1]$. Это свидетельствует о том, что задачи (3), (9) могут решаться и в "инверсной" к ним постановке. С тем же результатом по поводу разрешимости задачи и обоснованности алгоритмов.

Пусть теперь A – произвольный алгоритм вида (2). Разобьем множество X на непересекающиеся подмножества X_1, X_2 такие, что $X_1 \bigcup X_2 = X$. Поставим в соответствие алгоритму A алгоритмы A_1, A_2 следующим образом:

$$\forall x \in X_i \; (A_i(x) = A(x)) \wedge \forall x \in X \setminus X_i \; (A_i(x) = (0,0)), i = 1,2.$$

Нетрудно заметить, что A_1, A_2 представляют собой "проекции" алгоритма A на соответствующие подмножества X_1, X_2 с доопределением на всем множестве X. Очевидно, что при этих условиях алгоритм A можно представить в виде $A = A_1 + A_2$. При дополнительном же условии аддитивности функционала Ф, справедливым будет следующее соотношение $\Phi_A(X) = \Phi_{A_1}(X) + \Phi_{A_2}(X)$.

Обратимся снова к задаче $\boldsymbol{Z_C}$. Допустим, что мы умеем вычислять значение функционала $\Phi_A(X) \in [0,1]$ в условиях задачи $\boldsymbol{Z_C}$. Допустим также, что это означает возможность непосредственного указания выборок X_1, X_2, где X_1 - выборка, на которой алгоритм A удовлетворяет условию $P(X_1) = P_A(X_1)$, а X_2 положим равным $X \setminus X_1$. Имея такое разбиение, нетрудно построить алгоритм A', для которого будет справедливо $\Phi_{A'}(X)=1$. Независимо от результата $\Phi_A(X)$.

В том же случае, когда мы не можем непосредственно указать выборки X_1, X_2, у нас все равно остается возможность построения процедуры (может быть и стохастической), в результате которой подходящий алгоритм A' будет построен. Отметим, правда, что в этом случае невозможно обойтись без привлечения дополнительных предположений о структуре информации X и/или соотношении множеств X и X^0.

Нам неизвестно, остаются ли справедливыми все проведенные рассуждения в случае, когда вместо вычисления значения $\Phi_A(X)\in[0,1]$ имеется способ его оценивания. Так, например, как это происходит при использовании предельных схем в задаче (9). Но даже если в постановке (9) все проведенные построения не работают, все равно они существенно снижают оптимизм по поводу возможности вычисления и/или оценивания $\Phi_A(X)\in[0,1]$.

Остается заметить, что все проведенные рассуждения обобщаются на случай $l>2$. Существование инверсного алгоритма A' вообще доказывается без труда. А вот его построение связано с определенными алгоритмическими сложностями. Наиболее простой путь – через дихотомическое разбиение множества X^A, представляющего собой объединение подмножеств $X_1^A,\dots,X_l^A$.

2. РЕАЛИЗАЦИЯ КОРРЕКТНЫХ АЛГОРИТМОВ

Результаты, приведенные выше при анализе разрешимости задачи $\boldsymbol{Z_C}$, базируются на условии корректности алгоритма A: $\Phi_A(\widetilde{X}^0)=1$. Ясно также, что обоснованные алгоритмы, также содержатся в множестве корректных. Поэтому, если формально подойти к решению задачи $\boldsymbol{Z_C}$, то необходимо для каждой такой задачи научиться строить все множество корректных алгоритмов. Желательно, чтобы технически такое построение осуществлялось с минимальными сложностями.

Как уже сказано во введении, большая часть приведенных в этом пункте результатов базируется или тематически связана с [6, 7, 17]. В связи с этим изложение носит в значительной степени схематичный и упрощенный характер. В указанных работах можно также уточнить многие детали, используемых в дальнейшем построений и обозначений.

2.1. *Общие условия корректности алгоритмов*

Опишем вначале множество алгоритмов распознавания, в котором ниже будут проведены все исследования. Для этого вернемся к сделанным при постановке задачи допущениям. Из них нетрудно заключить, что в наиболее общем виде алгоритм распознавания можно рассматривать как отображение $A: X \to \mathbf{B}_2^l$, где $\mathbf{B}_2^l$ - l-я декартова степень множества $\mathbf{B}_2=\{0,1\}$. В этом случае результат его работы на всех $x\in X$ без труда трактуется в системе предикатов P^A. В множестве таких отображений выделим подмножество (класс) алгоритмов, который порождается естественной суперпозицией вида: $A=c\circ B$, составленной из *распознающего оператора* $B: X \to \mathbf{R}^l$ (здесь $\mathbf{R}$ - пространство действительных чисел) и *решающего правила* $\mathbf{c}: \mathbf{R}^l \to \mathbf{B}_2^l$. Множество однотипных в каком-либо смысле распознающих операторов назовем моделью и обозначим $\mathfrak{M}$. Тогда $A_{\mathfrak{M}}=c\circ\mathfrak{M}$ представляет собой некоторую модель алгоритмов распознавания. Среди всех $A_{\mathfrak{M}}$, допускающих такое представление, мы ограничимся рассмотрением моделей с сюрьективными отображениями c, решающие правила которых удовлетворяют условию: $\forall b \in \mathbf{B}_2^l \ \exists r \in \mathbf{R}^l \ (\mathbf{c}(r) = b)$. Это связано с тем, что в исходной постановке задачи никаких ограничений на структуру классов, их соотношение не вводилось. И поэтому, как нетрудно убедиться, из (3) с необходимостью следует введенное ограничение на модель $\mathrm{A}_{\mathfrak{M}}$.

Рассмотрим теперь условия существования в модели $\mathrm{A}_{\mathfrak{M}}$ алгоритмов, корректных на фиксированной конечной выборке X^0 (или короче - корректности модели $\mathrm{A}_{\mathfrak{M}}$). Для того поставим в соответствие решающему правилу **c** на **X** следующее множество: $\mathrm{R_c(X)} = \{\mathrm{R} \subseteq \mathrm{R}^l \mid \mathrm{c(R)=P(X)}\}$, где $\mathrm{P(X)} \subseteq \mathbf{B}_2^l$ - совокупность всевозможных значений предиката P на X. Из сюрьективности **c** следует, что данное определение правомерно. Но по причинам, которые уже указывались ранее, вычислить множество $\mathrm{R_c(X)}$ не представляется возможным. На множестве же X^0 дело обстоит иначе, хотя и в этом случае имеются трудности, т.к. $\mathrm{R_c}(\mathrm{X}^0)$ может оказаться бесконечным. Как бы там ни было, структурируем последнее множество фиксацией определенного порядка в $\{1,...,l\}$ и $\{1,...,\mid \mathrm{X}^0 \mid\}$. Если теперь обозначить $\mid \mathrm{X}^0 \mid = \mathrm{q}$, то $\mathrm{R_c}(\mathrm{X}^0)$ можно рассматривать как матрицы пространства $\mathbf{R}^{\mathbf{q}l}$ размерности $q \times l$. Очевидно, что и $\mathrm{B}(\mathrm{X}^0)$ являются элементами того же матричного пространства для всех распознающих операторов $\mathrm{B} \in \mathfrak{M}$. Теперь без труда доказывается

Утверждение 5. *Модель* $\mathrm{A}_{\mathfrak{M}}$ *является корректной на выборке* X^0 *в том и только том случае, если*

$$\mathfrak{M}(\mathrm{X}^0) \cap \mathrm{R_c}(\mathrm{X}^0) \neq \varnothing \tag{10}$$

Действительно, при сделанных предположениях легко убедиться, что условие (10) эквивалентно следующему:

$$\exists \mathrm{A} \in \mathrm{A}_{\mathfrak{M}} \ (\Phi_{\mathrm{A}}(\mathrm{X}^0) = 1).$$

Полученное условие позволяет еще более конкретизировать множество рассматриваемых моделей $\mathrm{A}_{\mathfrak{M}}$, т.к. из него также с необходимостью следует требование сюрьективности решающего правила *c*, но на множестве, соответствующем выборке X^0 (такие *c* называют иногда *корректными* или *непротиворечивыми*). Это означает, что при построении и исследовании $\mathrm{A}_{\mathfrak{M}}$ можно ограничиться каким-то фиксированным сюрьективным решающим правилом, т.к. в этом случае будет иметь место: $\mathrm{R_c}(\mathrm{X}^0) \neq \varnothing$. Безусловно, отсюда не следует, что все множество возможных моделей $\mathrm{A}_{\mathfrak{M}}$ может быть порождено таким путем. Правильным будет заключение, что при этом все сложности реализации переносятся на уровень модели распознающих операторов $\mathfrak{M}$. Отметим также следующее немаловажное обстоятельство - исследование $\mathrm{A}_{\mathfrak{M}}$ существенно упрощается, т.к. пространство $\mathbf{R}^{\mathbf{q}l}$ по своим возможностям значительно "сильнее", чем $\mathbf{B}_2^{\mathbf{q}l}$ (где допустимы только логические операции) и заведомо не "слабее" любого множества X . Исходя из сказанного, в дальнейшем мы будем рассматривать модели $\mathrm{A}_{\mathfrak{M}}$ с т.н. *линейными пороговыми* решающими правилами, определенными и в пространстве $\mathbf{R}^{\mathbf{q}l}$ (наряду с $\mathbf{R}^l$):

$$\forall R \in \mathbf{R}^{\mathbf{q}l} \quad (\mathbf{c}(R) = \left\| \mathbf{c}(r_{ij}) \right\| \in \mathbf{B}_2^{\mathbf{q}l}),$$

и действующими по правилу (где $(i,j) \in \mathrm{I} = \{1,...,q\} \times \{1,...,l\}$, $c_0 \in \mathbf{R}$)

$$c(r_{ij})=\begin{cases} 1, \text{if } r_{ij}>c_0, \\ 0, \text{if } r_{ij}\le c_0. \end{cases}$$

Таким образом, нас будут интересовать условия корректности моделей $A_{\mathfrak{M}}$, порождаемых моделью $\mathfrak{M}$ и некоторым фиксированным линейным пороговым решающим правилом $c(c_0)$. Выберем порог из условия: $c_0 > 0$, что, как станет ясно из дальнейшего, ничуть не уменьшает общности полученных результатов. Для простоты формулировки результатов будем также считать, что все рассматриваемые ниже модели $\mathfrak{M}$ удовлетворяют следующему условию

$$\forall \mathrm{B} \in \mathfrak{M} \quad \forall x \in \mathrm{X} \quad (\mathrm{B}(x) = (b_1(x),...,b_l(x)),\ b_i(x) \ge 0).$$

Рассмотрим вначале самый общий случай, когда на модель $\mathfrak{M}$ никаких дополнительных условий не налагается. Нетрудно убедиться, что $\mathrm{R_c}(\mathrm{X}^0)$ образует выпуклое подмножество в пространстве $\mathbf{R}^{ql}$ и, в силу выбора решающего правила, является решением системы нестрогих линейных неравенств. Множество таких решений может быть охарактеризовано в терминах отделимости специальным образом построенных подпространств. Для их описания нам потребуются некоторые обозначения. Разобьем множество индексов $\mathbf{I}$ на подмножества

$$\mathrm{M}_t = \{(i,j) \,|\, (i,j) \in \mathbf{I}, \mathrm{P}_j(x_i) = t\}$$

и по аналогии с векторным случаем, введем в пространстве $\mathbf{R}^{ql}$ скалярное произведение матриц

$$\forall R_1, R_2 \in \mathbf{R}^{ql} \quad (< R_1, R_2 > := \sum_{(i,j)\in \mathbf{I}} r_{ij}^1 \cdot r_{ij}^2), \quad \text{где } R_k = \left\| r_{ij}^k \right\|,\ k = 1,2.$$

Обозначим через $\{E_{ij}\}$ - канонический базис (с единичным элементом $(i,j) \in \mathbf{I}$ и остальными равными нулю) пространства $\mathbf{R}^{ql}$. В этих обозначениях легко может быть сформулирован и доказан следующий критерий корректности модели $A_{\mathfrak{M}}$.

Теорема 1. *Модель* $A_{\mathfrak{M}}$ *является корректной на* $\mathbf{X}^0$ *в том и только том случае, если*

$$\exists\, \mathrm{B} \in \mathfrak{M} \ \Big(\min_{(i,j)\in \mathrm{M}_1} \{< \mathrm{B}(\mathrm{X}^0), E_{ij} >\} > c_0 \ge \max_{(i,j)\in M_0} \{< \mathrm{B}(\mathrm{X}^0), E_{ij} >\}\Big). \tag{11}$$

Доказательство этой теоремы не сложно и следует из эквивалентности (10) и (11).

Можно получить и другие условия, в том числе и достаточные для корректности. Однако на практике подобные условия оказываются малоэффективными, т.к. с их помощью подходящие алгоритмы могут быть получены только в результате решения систем матричных неравенств. Сделать это весьма сложно даже для простейших моделей $\mathfrak{M}$. Такая не конструктивность условия (11) объясняется его слишком большой общностью, что связано с отсутствием каких-либо требований к распознающим операторам из $\mathfrak{M}$.

2.2. *Корректные алгоритмы в алгебраических расширениях*

Рассмотрим теперь некоторые требования, которые могут предъявляться к модели $\mathfrak{M}$ с целью понижения сложности реализации корректных алгоритмов. Для этого воспользуемся техникой, основанной на следующем определении. Обозначим через F - некоторый класс однотипных функций вида

$$\forall f \in \mathbf{F} \quad (f: (\mathbf{R}^l)^m \ \rightarrow \ \mathbf{R}^l),\ m=1,2,...$$

Модель распознающих операторов $\mathfrak{M}$ назовем ***F** - нерасширяемой* на X и обозначим $\mathfrak{M}_{\mathrm{F}}$, если для суперпозиции $\mathrm{F} \circ \mathfrak{M}$ выполняется: $\mathbf{F} \circ \mathfrak{M}(\mathrm{X}) \subseteq \mathfrak{M}(\mathrm{X})$.

Нетрудно заметить, что данное условие является в некотором смысле "самодополнительным" - если по отношению к исследуемым классу функций F и модели $\mathfrak{M}$ не удается показать **F** - нерасширяемость последней, то дополнив $\mathfrak{M}$ стандартным образом функциями из **F** получим модель, которая заведомо таким свойством обладает. Построенная в итоге модель будет всегда превосходить исходную как по числу операторов, так и по мощности множества значений (на этом, кстати, основана идея корректировки или двухуровневых алгоритмов распознавания). Но важнее то, что от такой модели можно ожидать корректности при каких-то более "слабых" по сравнению с (10) условиях (т.е. достаточных для выполнения (10)). Очевидно, что то же можно сказать и в том случае, когда **F**- нерасширяемость модели $\mathfrak{M}$ доказана. Исходя из этого, введем еще одно определение.

Модель $\mathfrak{M}$ назовем *полной* на множестве X^{0}, если: $\mathfrak{M}(\mathrm{X}^0) = \mathbf{R}^{ql}$.

Легко видеть, что свойства полноты и корректности связаны именно так, как сказано выше, т.е. полнота влечет (10). И это является методологической основой для получения условий полноты.

Введем теперь три типа возможных функций F и рассмотрим условия корректности и полноты для соответствующих моделей с распознающими операторами $\mathfrak{M}_{\mathrm{F}}$. Надо заметить, что последующий выбор F обусловлен в основном традициями и является далеко не исчерпывающим. Вид функций будем определять сразу в пространстве $\mathbf{R}^{\mathbf{q}l}$, что вполне правомерно в силу предположения о фиксации X^{0}.

Класс $\mathbf{F}_1$ определим как множество отображений вида:

$$\mathbf{F}_1 := \{ f \,|\, f(R_1,...,R_m) = \sum_{i=1}^{m} \beta_i \cdot R_i \} \quad , \quad m=1,2,...$$

где $\beta_i \in \mathbf{R}$ $(\beta_i \geq 0)$, $R_i \in \mathbf{R}^{\mathbf{q}l}$. Класс $\mathbf{F}_2$ - как множество:

$$\mathbf{F}_2 := \{ f \,|\, f(R_1,...,R_m) = \sum_{i=1}^{m} \beta_i \cdot R_i^{t_i} \} \quad , \quad m=1,2,...$$

где $\beta_i \in \mathbf{R}$ $(\beta_i \geq 0)$, $R_i^{t_i} := \left\| (r_{uv}^{i})^{t_i} \right\|$ - t_i-я степень матрицы $R_i = \left\| (r_{uv}^{i}) \right\| \in \mathbf{R}^{ql}$ $(t_i \in \mathbf{N})$, определенная в коммутативной относительно операции произведения матричной алгебре (т.е. фактически в векторной алгебре).

Для описания следующего класса $\mathbf{F}_3$ нам потребуются некоторые дополнительные построения. Введем вначале множество отображений

$$\mathbf{G} := \{ \mathrm{g} \,|\, \mathrm{g} : (\mathbf{R}^{l})^{m} \to \mathbf{R}^{\mathrm{q}} \}.$$

Далее, если воспользоваться изоморфизмом пространства $\mathbf{R}^{\mathbf{q}l}$ и тензорного произведения $\mathbf{R}^{\mathbf{q}} \otimes \mathbf{R}^{l}$ векторных пространств $\mathbf{R}^{\mathbf{q}}$ и $\mathbf{R}^{l}$, то с помощью произвольного билинейного отображения

$$\mathbf{H} := \{ h \,|\, h : \mathbf{R}^{\mathbf{q}} \times \mathbf{R}^{l} \to \mathbf{R}^{\mathbf{q}l} \},$$

нетрудно построить искомое отображение **F**.

Более подробно со способами построения и свойствами таких отображений (включающими общие условия корректности и полноты соответствующих моделей алгоритмов) можно ознакомиться в [17]. В настоящей же работе мы зафиксируем класс $\mathbf{F}_3$, который был впервые введен и рассмотрен в [17]. С этой целью необходимо конкретизировать отображения $\mathbf{G}$. Введем еще два множества параметризованных функций $\mathbf{G}_1 : \mathbf{R}^l \rightarrow \mathbf{R}$, $\mathbf{G}_2 : \mathbf{R}^m \rightarrow \mathbf{R}$ и определим в пространстве $(\mathbf{R}^l)^m$ каким-то образом их суперпозицию $\mathbf{G}_2 \circ (\mathbf{G}_1)^m$. Тогда, очевидно, в качестве $\mathbf{G}$ можно выбрать множество всевозможных наборов мощности q (напомним, что q связано с размерностью выборки X^0) из построенной суперпозиции. В [9] показано, что при таком построении $\mathbf{G}$ в качестве $\mathbf{H}$ можно рассматривать обычные линейные операторы из $\mathbf{R}^\mathbf{q}$ в $\mathbf{R}^l$ (т.е. матрицы пространства $\mathbf{R}^{\mathbf{q}l}$). Если это матричное пространство обозначить через $\mathbf{L}(\mathbf{R}^\mathbf{q}, \mathbf{R}^l)$, то $\mathbf{F}$ можно представить в следующем виде: $\mathbf{F} = \mathbf{L}(\mathbf{R}^\mathbf{q}, \mathbf{R}^l) \circ \mathbf{G}$. Определим, наконец, $\mathbf{F}_3$ как такое подмножество построенных отображений F, в котором функции $\mathbf{G}_1, \mathbf{G}_2$ удовлетворяют условиям:

$$\forall \mathbf{g} \in \mathbf{G}_1 \quad \forall y \in \mathbf{R}^l \quad (\mathbf{g}(y) \ := \ \eta(y, y_\eta)),$$

где $y_\eta \in \mathbf{R}^l$ - параметр (зависящий от η), а η - некоторая метрика в $\mathbf{R}^l$;

$$\forall \mathbf{g} \in \mathbf{G}_2 \quad \forall y \in \mathbf{R}^m \quad (\mathbf{g}(y) \ := \ \mu(\sum_{i=1}^{m} y_i)),$$

где μ - монотонно убывающая функция такая, что: 1. $\mu(0) = c_1 > 0$ (c_1 - числовая постоянная), 2. $\lim\limits_{y\to\infty} \mu(y) = 0$.

Теперь можно сформулировать и доказать ряд следующих результатов для моделей $\mathfrak{M}_{\mathrm{F}_i}$ $(i=1,2,3)$.

Теорема 2. *Модель* $\mathfrak{M}_{\mathrm{F}_1}$ *тогда и только тогда*:

- *является полной на* $\mathbf{X}^0$, *когда можно указать* $q \cdot l$ *операторов* $\mathrm{B}_{11},...,\mathrm{B}_{\mathrm{q}l}$ *таких, что набор* $(\mathrm{B}_{11}(\mathrm{X}^0),...,\mathrm{B}_{\mathrm{q}l}(\mathrm{X}^0))$ *образует базис в* $\mathbf{R}^{\mathbf{q}l}$;
- *удовлетворяет условию* (10) *на* $\mathbf{X}^0$, *когда существует конечный набор операторов* $\mathrm{B}_1,...,\mathrm{B}_m$ *таких, для которых справедливо*

$$\begin{cases} \forall k \in \{1,...,m\} \, \exists (i_0, j_0) \in M_1 \, \forall (u,v) \in M_0 \, (b^k_{i_0 j_0} > m \cdot b^k_{uv}) \\ \forall (i,j) \in M_1 \, \forall (u,v) \in M_0 \, \exists k_0 \in \{1,...,m\} \, (b^{k_0}_{ij} > m \cdot b^{k_0}_{uv}). \end{cases} \tag{12}$$

Теорема 3. *Модель* $\mathfrak{M}_{\mathrm{F}_2}$ *тогда и только тогда*:

- *является полной на* X^0, *когда можно указать* $q \cdot l$ *операторов* $\mathrm{B}_{11},...,\mathrm{B}_{\mathrm{q}l}$ *таких, что для набора* $(\mathrm{B}_{11}(\mathrm{X}^0),...,\mathrm{B}_{\mathrm{q}l}(\mathrm{X}^0))$ *имеет место*

$$\begin{cases} \forall (i,j) \in \mathrm{I} \, \exists (i_0, j_0) \in \mathrm{I} \, \forall (u,v) \neq (i_0, j_0) \, (b^{ij}_{i_0 j_0} > b^{ij}_{uv}) \\ \forall (i,j) \in \mathrm{I} \, \forall (u,v) \neq (i,j) \, \exists (i_0, j_0) \in \mathrm{I} \, (b^{i_0 j_0}_{ij} > b^{i_0 j_0}_{uv}). \end{cases} \tag{13}$$

- *удовлетворяет условию* (10) *на* X^0, *когда существует конечный набор операторов* $\mathrm{B}_1,...,\mathrm{B}_m$ *таких, для которых справедливо*

$$\begin{cases} \forall k \in \{1,...,m\} \exists (i_0, j_0) \in M_1 \, \forall (u,v) \in M_0 \, (b^k_{i_0 j_0} > b^k_{uv}) \\ \forall (i,j) \in M_1 \, \forall (u,v) \in M_0 \, \exists k_0 \in \{1,...,m\} \, (b^{k_0}_{ij} > b^{k_0}_{uv}). \end{cases} \tag{14}$$

Теорема 4. *Модель* $\mathfrak{M}_{\mathrm{F}_3}$ *тогда и только тогда*:

- *является полной на* X^0, *когда можно указать конечный набор операторов* $\mathrm{B}_1,...,\mathrm{B}_m$ *таких, что для* $(\mathrm{B}_1(\mathrm{X}^0),...,\mathrm{B}_m(\mathrm{X}^0))$ *имеет место*

$$((b^1(x_i) \oplus ... \oplus b^m(x_i) \neq b^1(x_j) \oplus ... \oplus b^m(x_j)) \Leftrightarrow (i \neq j)) \tag{15}$$

- *удовлетворяет условию* (10) *на* X^0, *когда существует конечный набор операторов* $\mathrm{B}_1,...,\mathrm{B}_m$ *таких, для которых справедливо*

$$((b^1(x_i) \oplus ... \oplus b^m(x_i) \neq b^1(x_j) \oplus ... \oplus b^m(x_j)) \Leftrightarrow (\mathrm{P}(x_i) \neq \mathrm{P}(x_j))) \tag{16}$$

Сделаем несколько замечаний по поводу использованных в формулировках теорем обозначений. Для индексации наборов операторов использованы два типа множеств - $\mathbf{I}$, а также $\{1,...,m\}$. Первое множество - как для указания положения элемента в матрице $\mathrm{B}(\mathrm{X}^0) = \|b_{ij}\| \in \mathbf{R}^{ql}$, так и для указания номера оператора в соответствующем наборе, а второе - только в последнем смысле. Номер оператора во всех случаях указывается верхним индексом. Через $b^k(x_i)$ обозначена *i*-я строка (соответствующая $x_i \in \mathrm{X}^0$) матрицы значений распознающего оператора B_k (*k=1,...,m*). Символом $\oplus$ - стандартная операция прямого суммирования векторов. И, наконец, $\mathrm{P}(x_i)$ - значение предиката P на $x_i \in \mathrm{X}^0$, которое также определено и, согласно постановке задачи, известно.

Замечание. *Достаточность условий подобных тем, которые приведены в теоремах 2 и 3, впервые была показана в* [9, 19, 20]. *Введенные выше ограничения на модель* $\mathfrak{M}$ *и отображения* $\mathbf{F}_2$, $\mathbf{F}_1$ *существенно используются при доказательстве необходимости этих условий. Интересно, что в такой формулировке результатов наилучшим образом видна роль каждой операции из* множеств $\mathbf{F}_2$, $\mathbf{F}_1$. *По этому поводу отметим еще одну операцию - покомпонентного "обращения" матриц в* пространстве $\mathbf{R}^{ql}$, *которая может быть использована для построения* $\mathbf{F}$ и *понижения сложности реализации алгоритмов, удовлетворяющих* (10).

Обсудим вкратце приведенные результаты. Заметим вначале, что полученные условия можно использовать для определения требований к информации из $\mathbf{X}$ ($\mathbf{X}^0$), при выполнении которых в $\mathrm{A}_{\mathfrak{M}}$ можно указать корректный алгоритм. В нашем же случае такие требования определены косвенно - в множестве $\mathfrak{M}(\mathrm{X}^0)$. Для перенесения их на информацию необходимо конкретизировать модель $\mathfrak{M}$. Процесс определения таких требований в большинстве случаев довольно сложен в техническом смысле. Тем не менее, для ряда известных моделей $\mathfrak{M}$ показано [9], что при выполнении не слишком "сильных" условий на X^0, эти модели удовлетворяют (13) либо (12, 14). Что же касается условий, полученных в последней теореме, то здесь дело обстоит несколько иначе, можно даже сказать с точностью до наоборот. Нужно очень постараться, чтобы найти разумно поставленную задачу и некоторую модель $\mathfrak{M}$, для которых условия (15) или уж во всяком случае (16) не выполняются даже при *m=1*. Более того, нетрудно увидеть связь между условием (16) и следующим

$$\forall x_1, x_2 \in X \, (x_1 \neq x_2 \Leftrightarrow P(x_1) \neq P(x_2)) \tag{17}$$

Эту связь можно сформулировать в виде следующего содержательного утверждения:

некоторая модель $\mathrm{A}_{\mathfrak{M}}$ *удовлетворяет условию* $\Phi_A(\mathrm{X}^0)=1$ *тогда, а в случае* $\mathbf{F}_3$*-нерасширяемости* $\mathfrak{M}$ *и выполнении в последней* (16) *на* X^0, *и только тогда, когда имеет место* (17).

Данное утверждение, несмотря на его очевидную значимость, мы формализовать, и доказывать не будем, т.к. для этого необходимо конкретизировать $\mathfrak{M}$. Значимость же, по нашему мнению, заключается в неулучшаемости таких моделей $\mathrm{A}_{\mathfrak{M}}$ в том смысле, что вряд ли для какой-то другой можно получить более простые условия на информацию.

Неулучшаемости в контексте рассмотренных выше классов функций $\mathbf{F}_i$ ($i=1,2,3$) можно также придать сравнительную форму. Это нетрудно сделать путем формализации следующей известной гипотезы: *чем "сильнее" модель, тем шире множество получаемых решений и "слабее" условия разрешимости любой задачи.* В качестве следствия сформулированных результатов легко доказать:

Следствие. *Независимо от выбора* $\mathfrak{M}$ *справедливо*

$$\forall \mathrm{X}^0 \quad (\mathrm{F}_1 \circ \mathfrak{M}(\mathrm{X}^0) \subseteq \mathrm{F}_2 \circ \mathfrak{M}(\mathrm{X}^0) \subseteq \mathrm{F}_3 \circ \mathfrak{M}(\mathrm{X}^0)).$$

Рассмотрим еще вопрос о сложности реализации корректных алгоритмов. Его можно разделить на две взаимосвязанные части - сложность проверки соответствующих условий и, если последние выполняются, сложность реализации подходящих алгоритмов. Что касается первой части, то здесь получено не очень много результатов. Сошлемся, к примеру, на работу [9], в которой впервые рассмотрены условия типа (14) и показано, что такие операторы могут быть построены в результате решения подходящим образом сформулированной задачи о покрытиях, сложность которой очевидна. По всей видимости, на основе аналогичной техники можно построить и операторы, которые будут удовлетворять условиям (15), (16). После этого сложность реализации корректных алгоритмов не слишком высока. В [17] показано, что в рассмотренной выше $\mathbf{F}_3$-нерасширяемой модели такая сложность не превосходит сложности обращения матриц пространства $\mathbf{R}^{\mathbf{qq}}$, т.е. не выше $O(|\mathrm{X}^0|^3)$. В тоже время, во всех остальных случаях эта оценка значительно выше.

3. АЛГОРИТМЫ ДЛЯ РЕШЕНИЯ ЗАДАЧИ Z

Сравнивая описанные выше задачи $\boldsymbol{Z}_{\boldsymbol{A\text{-}C}}$ можно заметить, что первая из них носит теоретический характер, две последние имеют явный прикладной смысл. К ним сводится большое число практических задач из области искусственного интеллекта. Сравнивая эти задачи, можно прийти к следующим альтернативным суждениям относительно результатов их решения: *либо обоснованный результат, не имеющий никакой практической интерпретации, либо результат, относительно которого неправомерно вообще ставить вопрос о каком-то обосновании* (*по крайней мере, до тех пор, пока индуктивный вывод не будет обоснован* [8]). Ситуация достаточно сложная для практических задач, т.к. решать их все-таки нужно и при этом желательно иметь в максимальной степени обоснованные результаты. Лучше чтобы это достигалось исключительно постановкой, а не правдоподобными дополнительными предположениями. Возможно ли это? Ответ может быть положительным, по крайней мере, в следующем смысле. Задачи $\boldsymbol{Z}_{\boldsymbol{A\text{-}C}}$ могут решаться в рамках единого семейства алгоритмов. Это позволит разделить алгоритмические и информационные аспекты проблемы обоснования. Последняя при этом переходит в плоскость доказательства некоторой общности постановок задач и обоснования правомерности использования построенных алгоритмов для решения хотя бы одной из задач. Сделать это можно сравнением результатов, получаемых с помощью алгоритма A_0 и построенного алгоритма. Правомерность использования алгоритма A_0 для решения задачи $\boldsymbol{Z_A}$ доказана по всем канонам математической строгости. И если окажется, что результаты совпадут, то это будет означать обоснованность построенного алгоритма. Или, во всяком случае, правомерность его использования, в том числе, и для решения задач $\boldsymbol{Z_C}$.

Замечание. *Но даже в случае реализации описанного выше подхода, остается информационная сторона задачи, по которой приведенные постановки никогда не будут объединены. Поэтому алгоритмические аспекты проблемы обоснования не исчерпывают в целом проблемы. Кроме того, не все варианты задачи* **Z** *входят в рассматриваемый список. Исходя из сказанного, точно можно утверждать лишь следующее - развиваемая точка зрения на проблему только подтверждает правомерность постановки вопроса об обосновании алгоритмов и разрешимости задачи* $\boldsymbol{Z_C}$.

3.1. *Уточнение постановки и общие требования к алгоритмам*

Начнем с описания множества X, в котором будет осуществляться сравнение. В $\boldsymbol{Z_A}$ это множество, в отличие от случая задач $\boldsymbol{Z_B}$ и $\boldsymbol{Z_C}$, является всегда бесконечным. Кроме того, в $\boldsymbol{Z_A}$ и $\boldsymbol{Z_B}$ наряду с пространствами фиксированной размерности допускаются конструкции над- и подпространств, что соответствует избыточности и недостаточности информации. Для того чтобы задачи $\boldsymbol{Z_{A-C}}$ сделать сравнимыми, необходимо обеспечить корректность работы алгоритма на всей возможной информации. Для упрощения изложения и наглядности ограничимся случаем булевых пространств. Иначе говоря, множество X будет отождествляться со следующей конструкцией:

$$\bigcup_{i=1}^{\infty} \boldsymbol{B}_2^i, \quad \boldsymbol{B}_2 = \{0, 1\} \tag{18}$$

При этом вектор из $\boldsymbol{B}_2^i$ по отношению к пространству $\boldsymbol{B}_2^{i+1}$ будет трактоваться как вектор, у которого не определено значение в соответствующей координате i+1. И наоборот.

Теперь о способах получения объектов из пространства (18). В задаче $\boldsymbol{Z_C}$ описание классов X_i задается обучающей выборкой X^O. В задачах $\boldsymbol{Z_A}$, $\boldsymbol{Z_B}$ известным является предикат. Причем в $\boldsymbol{Z_B}$ конечное число объектов из X доставляет истинностное значение предиката Р. А в $\boldsymbol{Z_A}$ число таких объектов бесконечно и покрывает все множество X. Предположим, что существует конструктивный способ, который каждому классу X_i ставит в соответствие наборы объектов с истинным значением предиката P_i. Объединим такие объекты и получим аналог обучающей выборки X^O. Таким образом, для всех задач обучающую выборку X^O будем считать заданной.

И, наконец, о требованиях к алгоритмам. Сформулируем два принципа, которые непосредственно следуют из условий работы соответствующих алгоритмов на задачах $\boldsymbol{Z_{A-C}}$. Первый из них:

1. *Принцип корректности А.*

$$\forall x \in X^O \quad (A(x) = (\mathrm{P}_1(x), \ldots, \mathrm{P}_l(x))) \tag{19}$$

Этот принцип является следствием роли условия корректности в разрешимости задачи **Z**, хотя с формальной точки зрения он не совпадает с определением корректности выше.

Для формулировки второго введем определение. Пусть заданы объекты $x_1 \in \mathbf{B}_2^i$ и $x_2 \in \mathbf{B}_2^{i+1}$ (i - произвольное натуральное число). Зафиксируем координату $j \in [1, i+1]$, по которой пространства $\mathbf{B}_2^i$ и $\mathbf{B}_2^{i+1}$ отличаются друг от друга. Скажем, что x_1 является *подобъектом* x_2, если при удалении координаты j оставшаяся часть вектора x_2 совпадает с x_1. Очевидно, что имеет место инверсия определения (т.е. x_2 можно назвать *надобъектом* x_1) и элементарное его расширение (на случай $i+2,\ldots$). Теперь получаем

2. *Принцип объектной монотонности А.*

Предположим, что для некоторого класса X_i соответствующая обучающая выборка X_i^0 состоит из единственного объекта x. Пусть, кроме того, заданы некоторые объекты $x_1, x_2 \in X$. Тогда, если x_1 является подобъектом x, а x_2 - подобъектом x_1, то должно выполняться условие

$$\mathrm{P}_i^{A}(x_1) \ \geq \ \mathrm{P}_i^{A}(x_2) \tag{20}$$

Сформулированные принципы являются следствием общих идей, связанных с решением задач $Z_{A\text{ - }C}$. Так, принцип корректности имеет место для задачи Z_A и предпочтителен для Z_B. Принцип объектной монотонности характерен для всех задач аппроксимации: *чем меньше информации известно, тем меньше должны быть любые сравнительные количественные оценки*. Этот принцип очевиден для Z_C. Реализуется он и в Z_B, но путем параметризации предикатов либо аксиоматизации вывода. Это не хуже – просто так устроен индуктивный вывод [12,13, 15]. Принцип объектной монотонности выглядит частным случаем, так как $|X_i^0| = 1$, однако такой формулировки вполне достаточно для обобщения на $|X_i^0| > 1$. Характер обобщения зависит от схемы алгоритма A.

3.2. *Описание алгоритмов решения задачи* Z

Опишем параметрическое семейство алгоритмов $\mathfrak{A}$. Предварительно введем некоторые соглашения и обозначения. Будем считать:

1. Признаки пространства X, число которых может быть и бесконечным, занумерованы и порядок их зафиксирован. Множество таких номеров обозначим через I.
2. Множество X^0 разбитым на подмножества X_i^0 ($i=1,\ldots,l$). Объекты в X_i^0 также занумерованными и порядок их зафиксирован. Множества таких номеров обозначим через $\boldsymbol{J}_k$ ($k=1,\ldots,l$).
3. Множеству $\{1,\ldots,l\} \times I$ поставлена в соответствие матрица действительных чисел $\| a_{ij} \|$, где $i \in \{1,\ldots,l\}, j \in \{1,\ldots,|\boldsymbol{I}|\}$, $a_{ij} \in \mathbf{R}$, для которой выполняется

$$\forall i, j \ \ (a_{ij} \geq 0), \ \forall i \ \ (\sum_j a_{ij} \ > 0) \tag{21}$$

Теперь алгоритм A можно описать как последовательность следующих шагов:

Шаг 1. Фиксируем некоторый объект $x \in X$ и переходим к шагу 2.

Шаг 2. Для каждого $i \in \{1,\ldots,l\}$ и для всех $x_j \in X_i^0$ ($j \in \boldsymbol{J}_i$) вычисляем

$$\mu_{A}^{i,j}(x, x_j) = max\ \{0, (\sum_{u \in \boldsymbol{I}} (-1)^t \cdot a_{iu}) \cdot (\sum_{u \in \boldsymbol{I}} a_{iu})^{-1}\},$$

где

$$t = \begin{cases} 1, \ if \ x_u \neq x_{ju}, \\ 2, \ otherwise. \end{cases}$$

Шаг 3. Если все $x_j \in X_i^0$ исчерпаны, то вычисляем

$$\mathrm{P}_i^{\boldsymbol{A}}(x) = \max_{x_j \in X_i^0} \{\mu_{\boldsymbol{A}}^{i,j}(x, x_j)\}.$$

Если номера классов $i \in \{1,...,l\}$ не исчерпаны, то возвращаемся на шаг 2. В противном случае переходим к следующему шагу.

Шаг 4. Если множество объектов x не исчерпано, то возвращаемся к шагу 1. В противном случае алгоритм заканчивает работу.

Конец работы алгоритма.

Нетрудно заметить, что описанный алгоритм A однозначно связан с выбором конкретных параметров $\| a_{ij} \|$. Это определяется в основном неалгоритмическими соображениями: выбором пространства описания объектов из множества X, желанием придать числам $a_{ij} \in \mathbf{R}$ и величинам $\mathrm{P}_i^{\boldsymbol{A}}(x)$ некоторую содержательную интерпретацию, способом формализации понятия “похожести” объектов и т.п. Для лучшего понимания задачи и анализа исходной информации величины $\mathrm{P}_i^{\boldsymbol{A}}(x)$ желательно интерпретировать в терминах предметной области. Поэтому способ выбора параметров $\| a_{ij} \|$ имеет существенное значение. Хотя работа алгоритма A, как будет показано ниже, от этого не зависит.

Опишем одну из возможных схем определения параметров $\| a_{ij} \|$.

Шаг 1. Для заданных $\boldsymbol{I}$ и X_i^0 выполняем следующую последовательность шагов

Шаг 2. Фиксируем номер признака $j \in \{1,...,|\boldsymbol{I}|\}$ и для каждого $i \in \{1,...,l\}$ вычисляем

$$b_{ij} = (\sum_{x_u \in X_i^0} x_{uj}) \cdot (|X_i^0|)^{-1}.$$

Шаг 3. Если все признаки $j \in \{1,...,|\boldsymbol{I}|\}$ не исчерпаны, то возвращаемся на шаг 2. В противном случае вычисляем

$$b_j = (\sum_{i=1}^{l} b_{ij}) \cdot l^{-1}, \quad a_{ij} = | \ b_{ij} - b_j \ |.$$

Конец работы алгоритма.

Нетрудно убедиться, что построенные таким образом параметры a_{ij} удовлетворяют условию (21), если $l > 1$ и для всех i выполняется $X_i^0 \neq \varnothing$. Более того, можно указать границы параметров: $a_{ij} \in [0,1)$. Параметризованное таким образом семейство алгоритмов обозначим через $\boldsymbol{\mathcal{A}}(a)$.

Рассмотрим еще один вопрос, связанный с выбором пространства X в виде (18). Здесь допустимыми являются объекты как бесконечной, так и любой конечной размерности. В последнем случае возможны варианты:

1. Объекты в X^0 могут оказаться из различных подпространств и, следовательно, несравнимыми с точки зрения схемы вычисления параметров $\| a_{ij} \|$.
2. Исследуемый объект x может иметь размерность, несравнимую с объектами в любой ко-

нечной выборке X^0.

Оба случая имеют отношение к способу построения выборок X^0. В $\boldsymbol{Z_C}$ проблемы не возникают, так как предполагается безусловная сравнимость объектов. В $\boldsymbol{Z_A}$, $\boldsymbol{Z_B}$ вначале необходимо получить систему предикатов $\mathrm{P}=(\mathrm{P}_1,...,\mathrm{P}_l)$. Для $\boldsymbol{Z_B}$ это понятно, даже когда предикаты не заданы в явном виде. Поэтому можно считать, что для всех $i \in \{1,...,l\}$ имеются $\mathrm{P}_i(x)$, которые в худшем случае определены в различных подпространствах. Построить $X_i^0 = \{x \mid \mathrm{P}_i(x) = 1\}$, например, можно путем приведения $\mathrm{P}_i(x)$ к *дизъюнктивной нормальной форме* (ДНФ). Предикат может быть доопределен с учетом содержательного смысла информации и проблем с вычислением параметров не возникнет. Если же доопределить не возможно, то достаточно модифицировать шаг 3 схемы вычисления параметров. Например, при вычислении b_j суммирование выполняем только по тем $i \in \{1,...,l\}$, где значение b_{ij} (шаг 2) определено, для всех остальных (i,j) полагаем $a_{ij} = 0$. Аналогично можно поступить и в случаях, когда некоторый признак j не определен в предикатах $\mathrm{P}_i(x)$ для всех $i \in \{1,...,l\}$. В ситуации, когда исследуемый объект x имеет несравнимую размерность, то в предложенной модификации эта проблема исчезает. Достаточно при вычислении t рассматривать логическое, а не алгебраическое несовпадение значений признака u.

В задаче $\boldsymbol{Z_A}$ все обстоит иначе. Даже если можно было бы получить систему предикатов $\mathrm{P}=(\mathrm{P}_1,...,\mathrm{P}_l)$, то и тогда остается проблема бесконечной выборки X^0. Но мы рассматриваем алгоритмические аспекты задачи $\boldsymbol{Z}$. Поэтому по-прежнему будем считать, что и для задачи $\boldsymbol{Z_A}$ такая выборка существует. Техника построения параметров $\| a_{ij} \|$ ничем не отличается от изложенной выше техники для задачи $\boldsymbol{Z_B}$.

3.3 *Анализ семейства алгоритмов* $\mathfrak{A}(a)$

Покажем, что любой алгоритм из $\mathfrak{A}(a)$ является решением задачи $\boldsymbol{Z}$. Для этого сформулируем следующее

***Утверждение* 6**. *Для любого* $A \in \mathfrak{A}(a)$ *имеет место*: $\forall x \in X \;\; (\mathrm{P}_i^A(x) \in [0,1])$ *и выполняется условие корректности* (19).

Из утверждения 6 можно получить несколько очевидных следствий, имеющих отношение к алгоритмическим аспектам монотонности. Сформулируем их в виде свойств алгоритма *A*. Вначале напомним, что алгоритм A_0 по определению удовлетворяет условию:

$$\mathrm{P}_i^{A_0}(x) = \begin{cases} 1, \; if \;\; x \in X_i^0, \\ 0, \; otherwise. \end{cases}$$

***Свойство* 2**. *Для любого* $A \in \mathfrak{A}(a)$ *имеет место*:

$$\forall x \in X \;\; \forall i \in \{1,...,l\} \qquad (\, \mathrm{P}_i^A(x) \;\; \geq \;\; \mathrm{P}_i^{A_0}(x)) \,.$$

Данное свойство можно интерпретировать как отношение "мажоризации" между алгоритмами $\mathbf{A}$ и $\mathbf{A}_0$.

***Свойство* 3**. *Для любого* $A \in \mathfrak{A}(a)$ *имеет место*:

$$\forall i \in \{1,...,l\} \;\; \forall x \notin X_i^0 \;\; (\, 1 > \mathrm{P}_i^A(x)) \;\Leftrightarrow\; \forall \, j \in \{1,...,|\boldsymbol{I}|\} \;\; (a_{ij} > 0)\,.$$

Оно определяет условие, при котором естественное различие (отделимость) множеств X^0 и

$X \setminus X^{0}$ алгоритм A "трансформирует" в различие между значениями предиката P_i^{A}. Последнее также имеет характер отделимости и может трактоваться как категорическая монотонность на границе множества. Условие в свойстве 3 должно определяться содержательными соображениями (т.е. постановкой задачи) и схемой вычисления параметров $\left\| a_{ij} \right\|$.

Перейдем к принципу объектной монотонности. Расширим условие (20), выполнимость которого для $\mathfrak{A}(a)$ почти очевидна, и докажем следующее

***Утверждение* 7**. *Пусть задана выборка* X^{0}. *Предположим, что для некоторых* $x \in X$ *и* $i \in \{1,...,l\}$ x *является подобъектом всех объектов, входящих в соответствующую выборку* X_i^{0}. *Тогда для любого* $A \in \mathfrak{A}(a)$ *имеют место свойства*:

(a). *для любого* x', *являющегося подобъектом* x, *выполняется*

$$(\mathrm{P}_i^{A}(x) \geq \mathrm{P}_i^{A}(x')) \tag{22}$$

(b). *для любого* x', *являющегося надобъектом* x, *условие* (22) *выполняется только в том случае, когда* x' *остается подобъектом по крайней мере одного объекта из* X^{0}.

В рассмотренном утверждении несколько расширено условие (20). В общем случае анализ оценки $\mathrm{P}_i^{A}(x)$ представляет определенные сложности. Это в основном связано со свойствами отношения "объект-подобъект", которое является несимметричным, транзитивным и с неопределенной рефлексивностью. Сложности имеют технический характер, что не позволяет провести полный анализ принципа объектной монотонности в рамках настоящей работы.

Итак, почти показано (за исключением возможности построения для $\boldsymbol{Z}_A$ выборки X^{0}), что семейство алгоритмов $\mathfrak{A}(a)$ решает задачу $\boldsymbol{Z}$. Очевидно, что, преодолев это ограничение, мы получим искомое решение.

С этой целью обратимся к схеме обычного доказательства разрешимости формальных теорий. Вначале выбирается язык формализации, и записываются основные положения (аксиомы и правила вывода). Затем определяется функция кодирования и показывается, что любому утверждению теории можно поставить в соответствие уникальный номер (натуральное число). Номер связывается с доказательством (ходом вывода). При этом проблема сводится к анализу структуры множеств, которые можно получить некоторой характеризацией утверждений, исследованию их свойств. Если оказывается, что характеристическая функция подмножеств является вычислимой (или рекурсивной), то и исходная теория разрешима. Можно построить соответствующий алгоритм, который по исходной информации для каждого нового утверждения распознает его свойства.

Для $\boldsymbol{Z}_A$ структура очень простая - множество утверждений (формул) распадается на три непересекающихся подмножества: тавтологии, тождественно ложные и нейтральные. Соответствующий алгоритм - A_0. Остается выбрать подходящую кодировку, чтобы получить элементы пространства (18) и чтобы число таких элементов было конечным. Причем в контексте проведенных рассуждений достаточно показать существование такой кодировки и последующую применимость алгоритмов $A \in \mathfrak{A}(a)$.

Теперь можно сформулировать следующую:

***Теорема* 5**. *Для* $\boldsymbol{Z}_A$ *существует конечная выборка* X^{0} *такая, что для всех алгоритмов A из* $\mathfrak{A}(a)$ *справедливо*

$$\forall x \in X \quad (A(x) = A_0(x)) \tag{23}$$

Доказательство. Укажем схему построения искомой выборки X^0. При этом будет показано, что все элементы этой схемы, по крайней мере, существуют.

Итак, каждый объект (формула) в $\boldsymbol{Z}_A$ может быть закодирован. Допустим, что процедура геделевской нумерации объектов уже выполнена. В результате получаем непересекающиеся подмножества множества натуральных чисел **N**. Обозначим эти подмножества через $\mathbf{N}_i$ ($i = 1,2,3$). Для разрешимости $\boldsymbol{Z}_A$ существенным является обстоятельство, что объединение подмножеств $\mathbf{N}_i$ является подмножеством либо совпадает с множеством **N** и сами $\mathbf{N}_i$ замкнуты. Теперь продолжим кодировку таким образом, чтобы получить элементы пространства (18). Для этого выберем некоторую функцию $f:\mathbf{N} \to \mathbf{N}$, которая просто переупорядочивает элементы подмножеств $\mathbf{N}_i$ таким образом, чтобы, по крайней мере, для трех $n_i \in \mathbf{N}_i$ было справедливо

$$f(n_1) = f(n_2) - 1 = f(n_3) - 2 \geq 3 \tag{24}$$

И на следующем этапе кодировки переведем полученные элементы в бинарное представление. В выборку X_i^0 включим бинарное представление соответствующего элемента $f(n_i)$. По условию (24) эти элементы различаются только в двух последних разрядах и, следовательно, при вычислении параметров $\left\| a_{ij} \right\|$ будет выполняться $\forall\ i\ \forall\ j > 2\ \ (a_{ij} = 0)$. Подсчитать соответствующие значения для младших разрядов также несложно. Если теперь потребовать от *f*, чтобы оно вообще давало характеризацию подмножеств $\mathbf{N}_i$ в смысле (24), то выполнение (23) становится очевидным.

Покажем, что такая характеризация возможна. Для этого будем строить *f* с помощью функций f_1, f_2 так, чтобы $\forall\, n \in \mathbf{N}\ (f(n) = \ \ = f_1(n) + f_2(n))$. И потребуем, чтобы функции f_1, f_2 удовлетворяли следующим условиям соответственно:

для n_1, n_2, n_3, которые фигурируют в условии (24): $f_1(n_1) = f_1(n_2) = f_1(n_3)$;

$$\forall\, n \in \mathbf{N}\ \ f_2(n) = \begin{cases} 1,\ if\ \ n \in \mathbf{N}_1, \\ 2,\ if\ \ n \in \mathbf{N}_2, \\ 3,\ if\ \ n \in \mathbf{N}_3. \end{cases}$$

Существование подходящей функции f_1 очевидно. Что же касается f_2, то ее существование обеспечивается известной топологической леммой Урысона. Применимость ее обеспечивается условиями на подмножества $\mathbf{N}_i$.

Остается сказать, что при построении формул, соответствующих номерам n_1, n_2, n_3, можно воспользоваться, например, тем же алгоритмом A_0. Так как номера произвольны, то достаточно выбрать любые три и получить их геделевский код. ■

Замечания:

1. условие (24) является не единственным. Возможны и другие подобные условия при сохранении той же схемы доказательства;
2. в обучающую выборку X^0 могут быть включены еще некоторые объекты, коды которых удовлетворяют условию (24). Для алгоритмов из $\mathfrak{A}(a)$ и в этом случае будет выполняться (23);
3. функция f_2 эквивалентна алгоритму A_0 и, имея такую функцию, совсем не обязательно строить A_0. Это не совсем точно в контексте сделанных в утверждении предположений, так как f_2 является все-таки частью некоторой кодировки *f*. Конструктивно последняя может быть получена самым различным образом. Но сказать что-либо об A_0 можно толь-

ко в том случае, если показано существование f_2. А это вопрос анализа f, точнее, структуры получающихся кодов. Ну и, кроме того, *решение любой задачи - это вопрос выбора подходящей кодировки*;

4. нетрудно привести примеры задач, которые могут быть решены по схеме, намеченной для $\boldsymbol{Z}_A$. Это, скажем, задача распознавания свойства четности на множестве натуральных чисел и некоторые другие.

ЗАКЛЮЧЕНИЕ

Из полученных выше результатов можно сделать следующие выводы. Первый касается проблемы, сформулированной в названии работы: *выбор алгоритмов при решении конкретной практической задачи должен быть обязательно обоснован, т.к. постановка задачи распознавания образов не вырожденна, по крайней мере, в алгоритмическом смысле.* Хотелось бы добавить, что и в информационном смысле также, но это будет скорее вывод из других (упомянутых в тексте) работ. Второй имеет отношение к условию корректности или точнее - его роли в ТРО: к*орректность хотя бы одной модели всегда свидетельствует о непротиворечивости задачи в постановочном плане, а при определенных условиях - и о полноте информации.* Надо сказать, что такая роль условия корректности не так уж и незначительна, как может показаться на первый взгляд. Чтобы убедиться в этом, достаточно обратиться, например, к проблеме обоснования аксиоматизируемых систем в теории множеств. Признаком обоснованности таких систем является непротиворечивость и полнота, но в значительно более категоричном по сравнению с задачей распознавания смысле. В таком контексте уместно также вспомнить, что в тех же аксиоматических системах для вывода используется метод резолюций, роль которого сравнима с ролью корректных алгоритмов для ТРО. Третий вывод мы сделаем по поводу применимости корректных алгоритмов: *для практического использования данное условие обязательно должно быть дополнено каким-то условием на информацию.* Имеется в виду, что корректность (как свойство алгоритмов) никогда не будет гарантировать разрешимости задачи на глобальном уровне. Для этого ее необходимо дополнить каким-то условием, обеспечивающим конечность выбора при бесконечном числе объектов. Это может быть, к примеру, конечная емкость множества алгоритмов, сходимость к пределу по информации, устойчивость и т.п. Всякое такое условие должно сужать множество корректных алгоритмов, которые окажутся применимыми для решения соответствующего класса задач.

Последующие выводы не столь бесспорны и непосредственно из полученных результатов вроде бы не следуют. Тем не менее, мы сочли возможным их сформулировать, т.к. они являются следствием опыта авторов по решению разнообразных практических задач в области распознавания и имеют в большей степени методологическое значение. Итак, четвертый вывод: *при нынешнем положении дел в ТРО разработка еще одной модели алгоритмов почти никак не отразится ни на теории, ни на практике.* Иначе говоря, настало время качественного осмысления задач распознавания на основе того опыта и чисто технических результатов, которые накоплены за последние десятилетия. Убедиться в этом совсем несложно - достаточно попытаться решить упомянутые во введении задачи. Почти без риска ошибиться, можно утверждать, что в данном случае для всякой модели и любых алгоритмов найдется такая информация, которая вне зависимости от используемой методики приведет к необходимости отказа от алгоритма. И в связи с этим, пятый вывод: *решение задачи распознавания - это процесс, конечной целью которого является установление точной аналитической характеризации классов.* По всей видимости, такая характеризация с помощью моделей и аппарата ТРО (как это сегодня понимается) в принципе невозможна. Если, допустим, как-нибудь можно преодолеть индуктивную природу задачи, то все еще останется слишком большое множество различных моделей и решений. А последнее всегда является признаком не очень хорошей постановки задачи. И, наконец, последний - <u>шестой</u> вывод, который, как нам кажется, не требует никаких комментариев: *конкретная задача и ее решение всегда важнее любых теоретических построений, а уж тем более - предпочтений, чем бы последние ни аргументировались.*

Итак, нами доказано, что алгоритмы $\mathfrak{A}(a)$ решают любую из задач $\boldsymbol{Z}_{A\text{ - }C}$. Причем на $\boldsymbol{Z}_A$ любой $\mathbf{A} \in \mathfrak{A}(a)$ эквивалентен $\mathbf{A}_0$. Следовательно, и на остальных задачах эти алгоритмы ведут себя в некотором смысле правильным образом: по крайней мере, удовлетворяют принципу корректности на $\boldsymbol{X}^o$ и монотонны на $\boldsymbol{X} \setminus \boldsymbol{X}^o$. В силу общности постановок применимость самого алгоритма (как схемы переработки исходной информации в результат) не вызывает сомнений. И этого вполне достаточно с позиции целей, поставленных в настоящей работе.

Но остаются информационные аспекты проблемы. Оказывается, что из приведенных выше результатов и в этом направлении можно сделать определенные выводы. Главные из них имеют отношение к способу построения и структуре $\boldsymbol{X}^o$. Вполне очевидно, что основные трудности при решении задач (и не только практических) связаны с выбором подходящего способа кодирования информации. Если задача имеет решение, то существует конечная кодировка и конечная выборка, для которых вопрос построения алгоритма типа $\mathbf{A}_0$ имеет исключительно технический характер. Можно определить параметры этой кодировки, структуру множества $\boldsymbol{X}^o$ и т.д. Сделать это возможно в результате экспериментов, которые являются неотъемлемой частью задач типа $\boldsymbol{Z}_B$, $\boldsymbol{Z}_C$.

ЛИТЕРАТУРА

1. Godel K. Uber formal unentscheidbare Satze der Principia Mathematica und vervandter System I. Monatshefte Math. Phys. 1931, 38, 173-198.
2. Barwise J. Handbook of Mathematical Logic, North-Holland, Amsterdam, 1977.
3. Thayse A. and Gribomont P. Approche logique de l'intelligence artificielle.1 De la logique classique a la programmation logique. Bordas, Paris ,1988.
4. Lauriere J.-L. Intelligence Artificielle. Resolution de Problemes par l'Homme et la Machine. Eyrolles, Paris, 1987
5. Nilsson N.J. Learning Machines. New York, McGraw-Hill1, 1965.
6. Krasnoproshin V., Obraztsov V.A. The Problem of Algorithms Choosing in Pattern Recognition // Pattern Recognition and Image Analysis. 1996. v. 6, no. 2. -p.p.188-199.
7. Краснопрошин В.В., Образцов В.А. Распознавание с обучением как проблема выбора // Цифровая обработка. Минск, 1998г., вып.2, стр.80-94.
8. Пятницын Б.Н. К проблеме соотношения индукции и дедукции // Методы логического анализа - М.: Наука, 1977.
9. Журавлев Ю. И. Об алгебраических методах в задачах распознавания и классификации // Распознавание, Классификация, Прогноз. Математические методы и их применение: ежегодник, под. ред. Журавлева Ю.И. М.: Наука, вып. 1, 1988, стр. 9-17.
10. Вапник В. Н. Индуктивные принципы поиска эмпирических закономерностей // Распознавание, Классификация, Прогноз. Математические методы и их применение: ежегодник, под. ред. Журавлева Ю.И. М.: Наука, вып. 1, 1988, стр. 17-82.
11. Гренандер У. Лекции по теории образов, том 1, М.: Мир, 1979
12. Polya G. Mathematics and Plausible Reasoning. Princeton University Press, New Jersey, 1954.
13. Кайберг Г. Вероятность и индуктивная логика, М.: Прогресс, 1978.
14. Hintikka J. Towards a Theory of Inductive Generalization. Reading in the philosophy of science, New York, 1970.

15. Popper K. The Logic of Scientific Discovery. London,1959.
16. Крейсел Г. Основания интуиционистской логики // в сб. Математическая логика и ее применения, под ред. Нагела Э. и др., М.: Мир, 1965.
17. Краснопрошин В. В., Образцов В.А. Двухуровневые модели алгоритмов распознавания // Журнал вычислительной математики и математической физики, 1985, том 25, №10, стр. 1534-1547.
18. Рудаков К.В. *Об алгебраической теории универсальных и локальных ограничений для задач классификации*, Распознавание, Классификация, Прогноз. Математические методы и их применение: ежегодник, под.ред. Журавлева Ю.И. (Москва: Наука), Вып. 1, 1988, стр. 176-201.
19. Журавлев Ю.И. *Экстремальные алгоритмы в алгебре над некорректными алгоритмами*, Доклады АН СССР, 1977, том 237, № 3, стр. 509-512.
20. Журавлев Ю.И., Исаев И.В. *Построение алгоритмов распознавания, корректных для заданной выборки*, Журнал вычислительной математики и математической физики, 1979, том 19, № 3, стр. 726-738.
21. Журавлев Ю.И. *Об алгебраическом подходе к решению задач распознавания или классификации //* Проблемы кибернетики, Москва: Наука, Вып. 33, 1978, стр. 5-68.