

Антипов С.Г., Фомина М.В.
МЭИ(ТУ), Москва

Метод формирования обобщенных понятий
с использованием
темпоральных деревьев решений

Антипов С.Г., Фомина М.В.
МЭИ(ТУ), Москва



Цель работы

Целью работы является исследование и разработка методов и алгоритмов диагностики технического объекта с использованием темпоральных деревьев решений; моделирование процесса диагностики.

Актуальность работы обуславливается усложнением технических систем и их повсеместным применением. Для обеспечения корректности и надежности их работы необходимо исследование и разработка методов диагностики.

Проблема индуктивного формирования понятий

Задачами *индуктивного формирования понятий* назовем задачи, которые моделируют возможность человека давать описания, охватывающие множество примеров некоторого понятия. В основе процесса индуктивного формирования понятий лежит умение человека выделять некоторые наиболее общие или характерные фрагменты описаний среди описаний отдельных примеров понятия, избавляясь от мелких, незначительных характеристик, присущих конкретным примерам понятия.

Люди пытаются понимать свое окружение, упрощая его. Создание подобной упрощенной модели называется индуктивным обучением. В процессе обучения человек наблюдает свое окружение и определяет взаимосвязи между объектами и событиями в этом окружении. Он группирует сходные объекты в классы и конструирует правила, предсказывающие поведение объектов такого класса.

Подобным образом может обучаться и компьютер. Изучение и компьютерное моделирование процесса обучения является предметом исследования в области искусственного интеллекта, называемой машинным обучением (Machine Learning). Как правило, система машинного обучения пользуется не единичными наблюдениями, а целыми (конечными) множествами наблюдений сразу. Такое множество называется *обучающим множеством* или *обучающей выборкой*.

Диагностика в искусственном интеллекте

Актуальной проблемой современных интеллектуальных систем является обнаружение знаний в массивах данных. Это связано с большим объемом современных баз данных, которые так велики, что практически невозможно вручную проанализировать их для извлечения ценной информации, помогающей принимать важные решения. Отсюда следует, что люди нуждаются в помощи интеллектуальных систем для повышения своих аналитических возможностей.

Системы автоматического обнаружения знаний могут анализировать "сырые" данные и предоставлять извлеченную информацию скорее и успешнее, чем аналитик мог бы найти её самостоятельно. Во многих случаях для описания поведения сложных систем приходится использовать сотни независимых атрибутов, которые необходимо анализировать, чтобы наиболее точно смоделировать поведение системы. В такой ситуации крайне важно решать задачу обобщения для получения компактных описаний классов ситуаций на объекте управления.

Диагностика как раздел искусственного интеллекта занимается разработкой методов и алгоритмов, способных определить корректность работы изучаемого объекта (системы). Если система работает некорректно, нужно как можно более точно определить, в какой части системы произошел отказ, и какая ошибка произошла. Определение ошибки происходит на основе наблюдений, которые дают информацию о поведении системы.

Постановка задачи обобщения

Дадим формулировку задачи обобщения понятий по признакам. Пусть имеется O - множество объектов, которые могут быть представлены в интеллектуальной системе S . Каждый объект характеризуется r признаками. Обозначим через $Dom(A_1), Dom(A_2), \dots, Dom(A_r)$ множества допустимых значений признаков, где $Dom(A_k) = \{x_{k1}, x_{k2}, \dots, x_{kq}\}$, $1 \leq k \leq r$, и q - количество различных значений признака A_k . Каждый объект $o \in O$, представляется как множество значений признаков, т.е. $o = \langle x_{i1}, x_{i2}, \dots, x_{ir} \rangle$, где $x_{ik} \in Dom(A_k)$. Такое описание объекта называется *признаковым описанием*. В качестве признаков объектов могут использоваться количественные, качественные либо шкалированные признаки.

В основе процесса обобщения лежит сравнение описаний исходных объектов, заданных совокупностью значений признаков, и выделение наиболее характерных фрагментов этих описаний. В зависимости от того, входит или не входит объект в объем некоторого понятия, назовем его *положительным* или *отрицательным* объектом для этого понятия.

Пусть O - множество всех объектов, представленных в некоторой системе, V - множество положительных объектов и W - множество отрицательных объектов. Будем рассматривать случай, когда $O = V \cup W$, $V \cap W = \emptyset$. Пусть K - непустое множество объектов такое, что $K = K^+ \cup K^-$, где $K^+ \subset V$ и $K^- \subset W$. Будем называть K *обучающей выборкой*. На основании обучающей выборки надо построить правило, разделяющее положительные и отрицательные объекты обучающей выборки.

Таким образом, понятие сформировано, если удалось построить решающее правило, которое для любого примера из обучающей выборки указывает, принадлежит ли этот пример понятию, или нет. Решающее правило является корректным, если оно в дальнейшем успешно распознаёт объекты, не вошедшие первоначально в обучающую выборку.

Методы решения задачи обобщения

Известен целый ряд методов, пригодных для обнаружения знаний. Каждый предложенный метод имеет свои преимущества. Методы, обычно используемые для извлечения полезной информации из данных, можно разделить на следующие группы:

Статистические методы. Статистический подход обычно основывается на вероятностных моделях. Предполагается, что такие методы будут использованы специалистами по статистике и, следовательно, требуется человеческое вмешательство при порождении гипотез и моделей.

Вывод, основанный на прецедентах. Это технология решения поставленной проблемы путём непосредственного использования предыдущего опыта и ранее принятых решений. Прецедентом (случаем) считается проблема, которая возникала ранее и которая была решена. При возникновении новой проблемы вывод, основанный на прецедентах, проверяет множество ситуаций, хранящихся в информационной базе, и находит схожую.

Нейронные сети. Нейронные сети состоят из большого числа нейроподобных элементов, соединенных между собой наподобие нейронов человеческого мозга. Подобно тому, как это происходит в человеческом мозге, стремление нейронов к взаимодействию может изменяться в ответ на воздействие или полученный выходной сигнал, что делает возможным обучение в сети.

Дерева решений. Дерево решений – это дерево, в котором каждый узел, не являющийся конечным, представляет собой проверку условия, или, если узел конечный, предъявляется решение для рассматриваемого элемента. Чтобы выполнить классификацию предъявленного примера, необходимо начать с корневого узла. Затем мы следуем по дереву решений вниз, пока не будет достигнут конечный узел (или лист). В каждом узле, не являющемся конечным, проверяется одно из условий. В зависимости от результата проверки условия выбирается соответствующая ветвь для дальнейшего движения по дереву. Решение получено, если мы достигли конечного узла.

Байесовские доверительные сети. Байесовская доверительная сеть представляет собой направленный ациклический граф, узлы которого представляют значения атрибутов, а веса дуг представляют вероятностные зависимости между атрибутами. Каждому узлу сопоставляются условные распределения вероятностей, описывающие отношения между узлом и предками этого узла.

Генетические алгоритмы \ эволюционное программирование. Генетические алгоритмы и эволюция – это алгоритмически оптимизированные стратегии, базирующиеся на принципах естественной эволюции. Генетические алгоритмы и эволюционное программирование используют методы обобщения, чтобы сформулировать гипотезы относительно зависимостей между переменными в форме ассоциативных правил, либо используя какой-то иной формализм.

Приближенные множества. Приближенное множество определяется посредством задания верхней и нижней границ некоторого множества, называемых приближениями этого множества. Каждый элемент нижнего приближения наверняка является элементом множества. Каждый элемент, не вошедший в верхнее приближение, наверняка не является элементом множества. Верхнее приближение приближенного множества - это объединение нижнего приближения и так называемой граничной области. Элемент граничной области вероятно (но, не наверняка) является элементом множества. Подобно нечетким множествам, приближенные множества являются математической концепцией для работы с нечеткостями в данных.

Модель для описания объекта

Для имитации поведения объекта и выявления неисправностей может быть использована модель специального вида, которая описывает структуру и поведение сложного технического объекта.

Данная модель представляет собой четверку $\langle O, E, S, B \rangle$, где:

O – множество компонент сложного технического объекта;

E – функциональные связи между компонентами;

S – множество переменных, описывающих состояние системы;

B – множество управляющих действий, допустимых в системе.

Модель для описания компонентов объекта

Для описания отдельного компонента o из O также используется модель, представляющая из себя тройку $\langle S', M, R \rangle$, где:

- S' - подмножество переменных из S , описывающих состояние данного компонента;
- M – набор режимов работы, включающих в себя состояние «норма» (корректное поведение) и состояния «неисправность» (некорректное поведение) ;
- R – набор отношений, связывающих множество переменных S , описывающих состояние системы, и набор режимов работы M .

Общая схема процедуры диагностики и выбора управляющего воздействия



Формирование обобщенных понятий для решения задачи диагностики

Решение задачи обобщения на основе анализа набора признаков предполагает первоначально построение обобщенного понятия, а затем использование его для классификации ситуаций на объекте. Рассмотрим возможные ситуации на объекте. Введем понятия:

C_n – множество состояний сложного технического объекта, которые диагностируются как нормальные;

C_f – множество состояний, в которых наблюдаются неисправности на объекте.

Необходимо сформировать описание понятий C_n и C_f в рамках введенной модели. На основе полученных обобщенных описаний классов C_n и C_f необходимо выработать рекомендации по выбору восстанавливающего действия на сложном техническом объекте; такое действие должно переводить систему из состояния «неисправно» в состояние «норма».

Недостатки классических деревьев решений

В настоящее время известен целый ряд алгоритмов для построения деревьев решений. Такие алгоритмы, как ID3, ДРЕВ и различные модификации этих алгоритмов - C4.5, ID5R и др. - обрели широкое распространение и зарекомендовали себя в широком спектре приложений.

Однако деревья решений в их классическом виде имеют некоторые ограничения — в частности, невозможность работы с поведением объекта или системы во времени, невозможность принятия решений с течением времени.

Классические деревья решений позволяют рассматривать только отдельные состояния объекта;

- невозможно проследить динамику изменения состояний;
- невозможность учитывать *фактор времени*.

Без учета фактора времени не удастся проследить динамику изменения состояния системы. Предлагается расширить признаковое описание объектов: введем понятие «время» как один из атрибутов, используемых при построении дерева решений. Будем использовать дискретное время: $t = 0, 1, 2, \dots$

Описание состояния объекта (системы) с учетом фактора времени

s1	s2	...	sn	t
z	l	...	l	0
z	h	...	z	1
h	n	...	n	2
n	n	...	z	3
n	v	...	z	4
h	v	...	l	5
l	z	...	n	6
l	h	...	z	7
z	v	...	z	8
l	h	...	v	9
...

Ситуация

Пусть для слежения за состоянием сложного технического объекта или системы используется некоторое количество датчиков. Обозначим эти датчики s_1, s_2, \dots, s_n . Пусть показания с этих датчиков снимаются через некоторый временной интервал t_0 - в каждый момент времени.

Каждую ситуацию из S_n и S_f предлагается рассматривать на некотором временном интервале t^* , включающем определенное количество дискретных значений t . При этом ситуацией будем называть последовательность из t^* значений датчиков с соответствующими им временными метками.

В данной работе под конкретной ситуацией понимается последовательность показаний датчиков, снятых за интервал времени t^* . Показания датчиков сопровождаются соответствующими им временными метками.

Пример ситуации(1)

s1	s2	...	sn	t
z	l	...	l	0
z	h	...	z	1
h	n	...	n	2
n	n	...	z	3
n	v	...	z	4
h	v	...	l	5
l	z	...	n	6
l	h	...	z	7
z	v	...	z	8
l	h	...	v	9
...

$t^*=5$

Пример ситуации (2)

s1	s2	...	sn	t
z	l	...	l	0
z	h	...	z	1
h	n	...	n	2
n	n	...	z	3
n	v	...	z	4
h	v	...	l	5
l	z	...	n	6
l	h	...	z	7
z	v	...	z	8
l	h	...	v	9
...

$t^*=5$

Пример ситуации (3)

s1	s2	...	sn	t
z	l	...	l	0
z	h	...	z	1
h	n	...	n	2
n	n	...	z	3
n	v	...	z	4
h	v	...	l	5
l	z	...	n	6
l	h	...	z	7
z	v	...	z	8
l	h	...	v	9
...

$t^*=5$

Темпоральные (временные) деревья решений

Темпоральное дерево решений — это взвешенный ориентированный граф

$$T_{temp} = (V_{temp}, E_{temp}).$$

Во множестве вершин V_{temp} выделим вершину v_0 — корень дерева. Все вершины разделим на два класса: V_i — множество внутренних вершины дерева; V_l — множество внешних вершин дерева (листьев)

Внутренние вершины V_i дерева взвешены (помечены) наблюдением, то есть парой $\langle a, t_c \rangle$, где: a — имя атрибута; t_c — временная метка.

Вершины-листья V_l взвешены (помечены) названием или номером ситуации и предлагаемым восстановительным действием, если ситуация относится к классу

C_f

Каждая дуга e дерева решений взвешена условием «атрибут[t_c] = значение_атрибута». где «атрибут» - имя атрибута в вершине, из которой исходит дуга e , «значение_атрибута» - одно из возможных значений признака «атрибут»; t_c — момент времени, в который необходимо проводить эту проверку, $0 \leq t_c < t^*$.

Исходные данные для построения темпоральных деревьев решений

Исходными данными является таблица наблюдений и модель восстановительных действий.

Каждой строке в таблице соответствует некоторая ситуация s_{it} , которая определяется:

- показаниями датчиков $Датчик_j$ в моменты времени $t=0, 1, \dots, t^*$
- восстановительным действием D_j для каждой ситуации;
- крайним сроком K_j выполнения соответствующего действия.

Такое представление позволяет не вводить каких-либо ограничений на способ представления времени в системе.

О восстановительных действиях

Восстановительные действия должны переводить сложный технический объект или систему из состояния «неисправность» в состояние «норма», при этом надо учесть, что они:

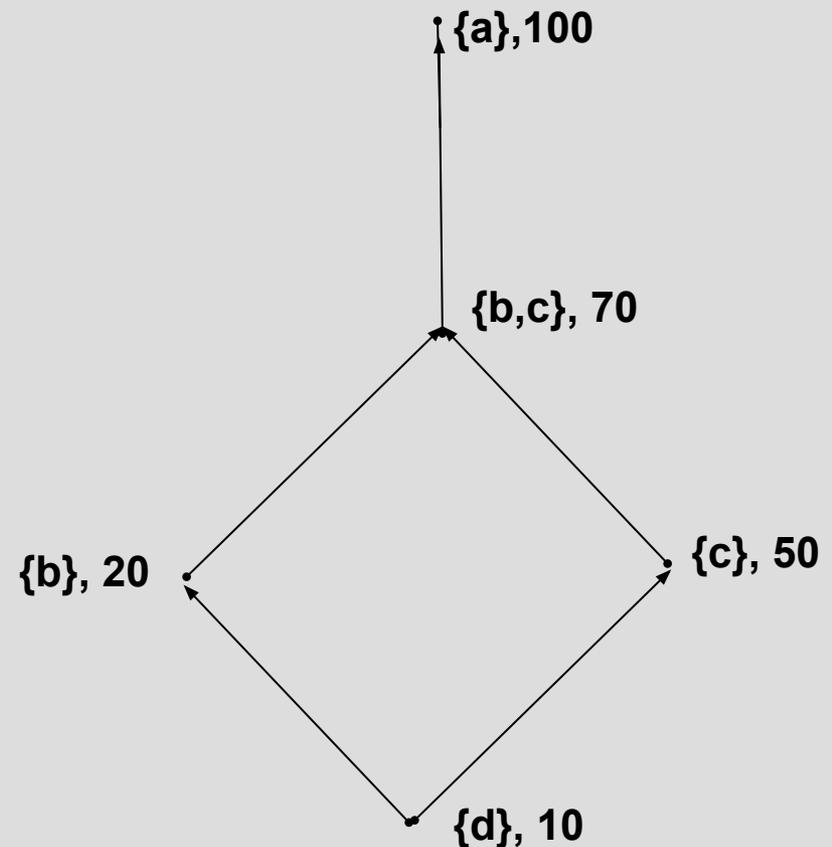
- могут негативно сказываться на функциональности системы (например, отключение процессора в компьютере)

- могут быть связанными между собой: операции, выполняемые восстановительным действием $d1$ в ситуации $f1$, могут включать в себя некоторые (или даже все) операции, которые выполняются восстановительным действием $d2$ в ситуации $f2$

Стоимость восстановительных действий

Восстановительные действия характеризуются названием и некоторой условной величиной — стоимостью (учитывается возможное негативное влияние).

Дополнительно можно ввести порядок над действиями (учитывается возможная связь между ними).



Пример исходных данных для алгоритмов построения темпоральных деревьев решений

t	Датчик1								Датчик2								Датчик3								Д	К
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7		
СИТ ₁	п	п	п	п	h	h			l	v	v	v	v	v			п	п	п	l	l	l			a	5
СИТ ₂	h	h	h						h	п	п						п	п	п						b	2
СИТ ₃	п	п	п	п	h	h	h	h	l	l	l	l	v	v	v	v	п	п	п	l	l	l	v	v	b	7
СИТ ₄	п	п	п	h	h	h	h		l	l	l	l	l	v			п	п	h	h	h	h	h		c	6
СИТ ₅	h	h	h	h					h	п	п	п					п	п	п	l					c	3
СИТ ₆	п	п	п	h	h	h			l	v	v	z	z	z			п	п	п	l	l	v			d	5
СИТ ₇	h	h	h	h	h	h			l	l	п	п	l	v			п	п	п	l	l	v			b	5
СИТ ₈	h	h	h	h	h	h			h	h	п	п	l	l			п	п	п	l	v	z			c	5

(z, п, l, v, h соответствуют качественным показаниям датчиков)

Общая схема алгоритма построения темпоральных деревьев решений

Алгоритм Построение_темпорального_дерева

(S: Таблица с ситуациями,

O: Наблюдения,

RM: Модель восстановительных действий)

Результат: Темпоральное дерево решений \mathcal{T}

начало

Если для всех ситуаций из S восстановительные действия совпадают,
то вернуть Лист(S, M)

Пусть D — минимальный крайний срок для ситуаций из S.

Если ситуации из S неразличимы на основе показаний датчиков с
меткой времени $t \leq D$, то вернуть Лист(S, M).

Выбрать наблюдение $\langle s^*, t' \rangle$, которое будет проверяться в данном
узле дерева.

Пусть $s^*_1, s^*_2, \dots, s^*_n$ — различающиеся показания датчика s^* в момент
времени t' , а $S^*_j, j=1, 2, \dots, n$ — подмножества ситуаций из S, состоящие
из ситуаций с показанием s^*_j датчика s^* в момент времени t' .

Вернуть дерево с корнем, помеченным выбранным наблюдением
 $\langle s^*, t' \rangle$, и дугами, помеченными $s^*_1, s^*_2, \dots, s^*_n$, соединяющими
корень соответственно с деревьями

Построение_темпорального_дерева($S^*_1, O \setminus \{ \langle s^*, t' \rangle \}, M$)

Построение_темпорального_дерева($S^*_2, O \setminus \{ \langle s^*, t' \rangle \}, M$)

...

Построение_темпорального_дерева($S^*_n, O \setminus \{ \langle s^*, t' \rangle \}, M$)

конец

Алгоритм CPD

В работе был исследован алгоритм для построения темпоральных деревьев решений, изложенный в работе *Console L., Picardi C., Dupré D. Temporal decision trees: model-based diagnosis of dynamic systems on-board // Journal of artificial intelligence research. 2003. №19* (назовем его CPD как сокращение от фамилий авторов - Console-Picardi-Dupre). Этот алгоритм предлагается использовать для бортовой диагностики, так как исходным ограничением на темпоральное дерево решений было неумножение временных меток при движении от корня дерева к листьям. В связи с этим отпадала необходимость хранить предшествующие значения датчиков для их возможного дальнейшего использования (вычислительные ресурсы, используемые при бортовой диагностике, сильно ограничены). Такое требование, наложенное на темпоральное дерево решений, нашло свое отражение и в алгоритме его построения. Более того, определение некорректной работы и неисправностей при бортовой диагностике часто включает в себя выбор некоторых управляющих действий, которые должны по возможности вернуть объект или систему в корректное состояние. В общем случае, как было показано выше, восстановительное действие имеет некоторую стоимость, которая выражается в том, что функциональность объекта или системы уменьшается - например, уменьшение скорости или полная остановка автомобиля, отключение каких-либо модулей системы и т. п. Поэтому хотелось бы минимизировать негативный эффект от восстановительных действий.

Минимизируется величина — ожидаемая стоимость дерева:

$$\chi(\text{node}) = \begin{cases} \text{стоимость действий в } \text{node}, & \text{если } \text{node} \text{ – лист дерева;} \\ \sum_{c \in \text{ребра дерева}} (\text{перейти из } \text{node} \text{ в } c) * \chi(c), & \text{если } \text{node} \text{ – внутренний узел дерева.} \end{cases}$$

Алгоритм CPD – выбор наблюдения для разбиения

Алгоритм Выбор_наблюдения_для_разбиения_CPД

(S: Таблица с ситуациями,

O: Наблюдения,

M: Модель восстановительных действий)

Результат: o^* - наиболее информативное наблюдение

начало

O1 = Выбрать_безопасные_наблюдения(S,O,M)

Вернуть Выбор_наблюдения_для_разбиения(S, O1)

конец

Алгоритм CPD – выбор безопасного наблюдения

Алгоритм Выбрать_безопасные_наблюдения

(S: Таблица с ситуациями,

O: Наблюдения,

M: Модель восстановительных действий)

Результат: O1- безопасные наблюдения

начало

Пусть cost – сумма стоимостей всех действий из S

Обозначим t_up - минимальный крайний срок для ситуаций из S;

t_low – минимальная временная метка для наблюдений O

t_max = t_up

Обозначим также разбиение part как пару $\langle S, t_up \rangle$

Для каждой временной метки tx от t_up до t_low

part = Произвести_разбиение(part, O, tx)

Посчитать стоимость newcost для получившегося разбиения part

Если newcost < cost, то {cost=newcost; t_max=tx}

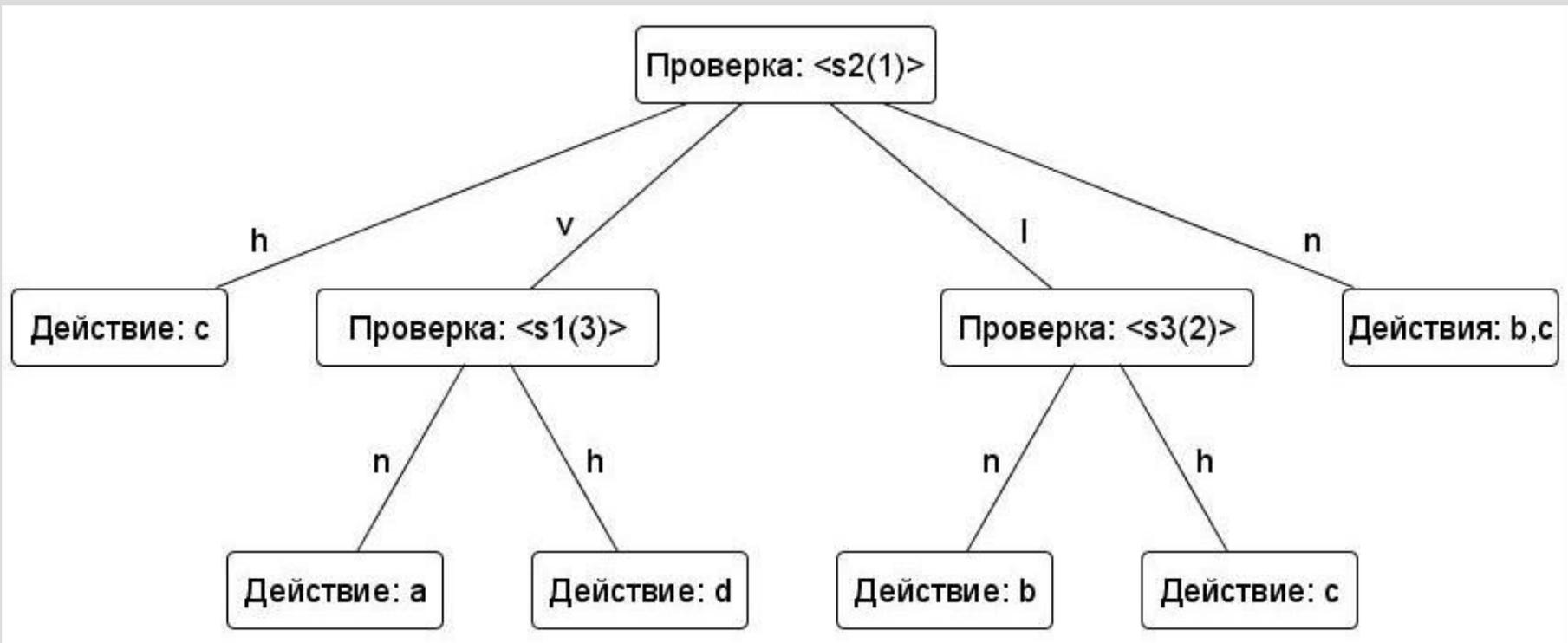
O1 = наблюдения из O с временными метками, не превосходящими

t_max

Вернуть O1

конец

Темпоральное дерево решений, построенное с помощью алгоритма CPD



Алгоритм temporal ID3

В работе предлагается оригинальный алгоритм (назовем его Temporal ID3), который является расширением алгоритма ID3, учитывающим фактор времени. Данный алгоритм представлен ниже. По сравнению с алгоритмом CPD, на темпоральное дерево решений не накладывается никаких ограничений по временным меткам в узлах. Однако не учитывается стоимость восстановительных действий при выборе наблюдения на каждом шаге. Кроме того, снимается ограничение на неумножение временных меток при движении от корня дерева к листьям, что приводит к необходимости сохранять некоторые значения датчиков для дальнейшего использования при проведении диагностики.

Наблюдения

(пара <датчик, временная метка>) рассматриваются как атрибуты (признаки).

Алгоритм temporal ID3

Алгоритм Temporal_ID3

(S: таблица с ситуациями,

O: наблюдения,

M: модель восстановительных действий)

Результат: Темпоральное дерево решений \hat{T}

начало

Если для всех ситуаций из S восстановительные действия совпадают,
то вернуть Лист(S, M)

Если множество наблюдений O пусто, то вернуть Лист(S, M)

Пусть D – минимальный крайний срок для ситуаций из S

Если ситуации из S неразличимы на основе показаний датчиков
с меткой времени $t \leq D$, то вернуть Лист(S, M)

$\langle s^*, t' \rangle = \text{Выбор_наблюдения_для_разбиения_temporal_ID3}(S, O)$

Пусть $s^*_1, s^*_2, \dots, s^*_n$ – различающиеся показания датчика s^* в момент
времени t' , а $S^*_j, j=1, 2, \dots, n$ – подмножества ситуаций из S, состоящие
из ситуаций с показанием s^*_j датчика s^* в момент времени t' .

Вернуть дерево с корнем, помеченным выбранным наблюдением

$\langle s^*, t' \rangle$, и дугами, помеченными $s^*_1, s^*_2, \dots, s^*_n$, соединяющими

корень соответственно с деревьями

Temporal_ID3($S^*_1, O \setminus \{ \langle s^*, t' \rangle \}$, M)

Temporal_ID3($S^*_2, O \setminus \{ \langle s^*, t' \rangle \}$, M)

...

Temporal_ID3 ($S^*_n, O \setminus \{ \langle s^*, t' \rangle \}$, M)

конец

Алгоритм temporal ID3 – выбор наиболее информативного наблюдения

Алгоритм Выбор_наблюдения_для_разбиения_temporal_ID3

(S: Таблица с ситуациями,

O: Наблюдения)

Результат: o^* - наиболее информативное наблюдение

начало

Для всех наблюдений $\langle s, t \rangle$ из O вычисляем количество информации, которое получаем благодаря этому наблюдению:

$\text{Gain}(\langle s, t \rangle, S) = \text{Info}(S) - \text{Info}(\langle s, t \rangle, S)$, где

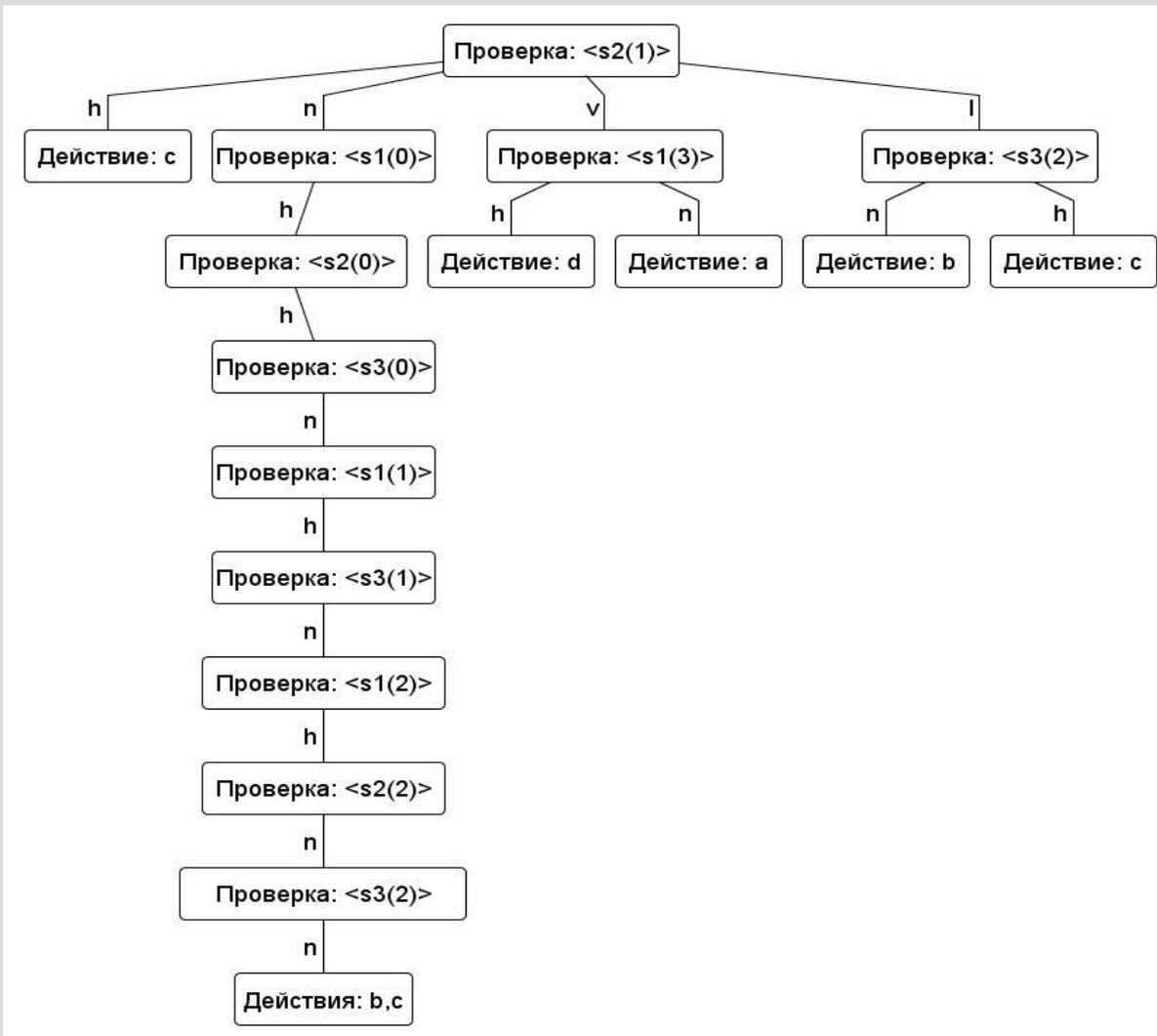
$\text{Info}(S)$ – энтропия для ситуаций из S (распределение восстановительных действий)

$\text{Info}(\langle s, t \rangle, S)$ – взвешенное среднее информации, необходимой для идентификации класса ситуации в каждом подмножестве, полученном при разбиении множества ситуаций из S на основе значений $\langle s, t \rangle$

Вернуть $\langle s^*, t^* \rangle$ - наблюдение с наибольшим значением $\text{Gain}(\langle s, t \rangle, S)$

конец

Темпоральное дерево решений (temporal ID3)



Моделирование процесса диагностики

Апостериорная диагностика: известны ситуации (все показания датчиков), требуется определить некорректность ситуации.

Диагностика в псевдореальном времени: последовательно поступают значения датчиков, требуется выявить некорректные ситуации.

Апостериорная диагностика

Для апостериорной диагностики может быть использовано одно темпоральное дерево решений. Ситуация классифицируется при прохождении последовательно от корня дерева к листьям. При достижении листа выбирается соответствующее восстановительное действие.

Диагностика в псевдореальном времени

При диагностике в псевдореальном времени последовательно поступают значения датчиков.

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	..
Датчик ₁	z	v	l	z	v	n	z	h	z	z	z	n	z	h	h	z	n	h	..
Датчик ₂	z	z	z	n	h	h	h	v	z	v	n	n	n	n	h	h	h	n	..
Датчик ₃	h	v	z	v	n	h	l	n	z	h	v	h	h	v	l	l	z	h	..

Диагностика в псевдореальном времени

При диагностике в псевдореальном времени одним темпоральным деревом не обойтись. Потребуется t^* рабочих агентов, которые будут следить за временными метками и хранить не потребовавшиеся на очередном шаге значения датчиков для их возможного дальнейшего использования. Первый такой агент начинает работу в момент времени $t=0$, второй — в момент времени $t=1$, ..., последний, t^* -ый агент начинает работу в момент времени $t=t^*-1$. Для организации совместной работы этих агентов потребуется агент-координатор, который будет получать информацию с датчиков, рассылать ее рабочим агентам и получать от них сведения о некорректной работе объекта или системы

Диагностика в псевдореальном времени



Точность диагностики

Моделирование процесса диагностики на основе изложенных алгоритмов состоит из нескольких этапов: первый — построение темпорального дерева решений на основе таблицы с исходными некорректными ситуациями; второй — создание многоагентной системы для диагностики в псевдореальном времени; третий — непосредственно процесс диагностики. Результаты программного моделирования представлены в таблице 2. Исходные данные для таблицы, на основе которой проводилось построение темпорального дерева решений, генерировались случайным образом по описанию предметной области. Каждая строка таблицы соответствует одному тесту. Пример № 0 (строка в таблице 2) соответствует тесту, представленному в таблице 1. Описание предметной области (метаданные) включает в себя:

- описание датчиков и их возможных значений;
- временной интервал t^* , в течение которого может возникнуть некорректная ситуация;
- описание восстановительных действий, их стоимостей и порядка над ними.

Для каждого теста выбранным алгоритмом строится темпоральное дерево решений. Полученное дерево решений далее используется многоагентной системой для отнесения ситуации, возникшей на объекте, к одному из классов ситуаций, для которых известны необходимые управляющие воздействия.

Второй этап моделирования заключается в генерации ситуаций, которые могут происходить на объекте, с последующей классификацией сгенерированных ситуаций. Представленные тесты содержат в среднем 40% некорректных ситуаций, остальные ситуации относятся к корректным.

Так как листья темпоральных деревьев решений помечены ситуациями, можно вычислить степень отличия таких ситуаций от ситуаций, которые были диагностированы деревом как некорректные. Предлагается следующая метрика:

$$Diff(sitList, sitCurrent) = \sum_{sens} \sum_{t=0}^{Deadl} sens(sitList, t) \neq sens(sitCurrent, t)$$

РЕЗУЛЬТАТЫ ДИАГНОСТИКИ

№	Ситуации для построения темпорального дерева		t*	Число ситуаций для апостериорной диагностики	Число временных тактов для диагностики в псевдореальном времени	Число некорректных ситуаций, выявленных алгоритмом			
	количество	Мин. крайний срок				Апостериорная диагностика		Диагностика в псевдореальном времени	
						T ID3	CPD	T ID3	CPD
(0)	8	8	8	20	20	3	7	6	12
(1)	8	3	8	200	200	68	99	69	111
(2)	8	3	8	200	200	64	105	79	113
(3)	8	3	8	200	200	64	112	73	116
(4)	20	3	8	200	200	139	143	147	150
(5)	20	5	8	200	200	126	128	112	142
(6)	20	6	16	200	200	178	147	173	143
(7)	20	8	16	200	200	180	134	181	122
(8)	20	10	16	200	200	161	115	157	126
(9)	20	9	24	200	200	133	162	115	165
(10)	20	12	24	200	200	168	132	164	138
(11)	20	15	24	200	200	171	152	149	147

Выводы по результатам эксперимента

Были рассмотрены два варианта диагностирования. В первом случае выполняется классификация ситуации целиком – это возможно, если показания датчиков сняты на временном интервале, не превышающем временной интервал, указанный в исходной таблице. При этом для классификации требуется единственное дерево.

Другой вариант - классификация на большом временном интервале: на вход последовательно поступают показания датчиков. В этом случае для классификации потребуется несколько деревьев решений (точнее, столько, какова величина временного интервала, использованного при построении деревьев). Помимо этого потребуются агенты, работающие с каждым из этих деревьев (выполняют функции "получение данных", "контроль времени") и агент-координатор, который получает данные и рассылает их рабочим агентам. Такая многоагентная система позволяет проводить бортовую диагностику в псевдореальном времени.

При проведении эксперимента было выявлено, что темпоральные деревья решений, построенные с использованием алгоритмов CPD и Temporal ID3, правильно определяют значительное количество некорректных ситуаций и выбирают нужные восстановительные действия. При этом следует отметить, что одним из недостатков алгоритма CPD является большое число ложных срабатываний — так называемых ошибок первого рода. Поэтому минимизация размера дерева решений не должна быть единственным критерием при его построении.

Заключение

В работе были решены следующие задачи:

- рассмотрен подход к диагностике с использованием деревьев решений;
- введен фактор времени как один из важных параметров при проведении диагностики;
- приведено описание исходных данных для задачи диагностики с учетом временного фактора;
- предложено формальное описание темпоральных деревьев решений;
- рассмотрен алгоритм CPD для построения темпоральных деревьев решений;
- предложен алгоритм temporal ID3, позволяющий строить темпоральные деревья решений без ограничения на неубывание временных меток в узлах дерева;
- рассмотрен процесс диагностики с использованием темпоральных деревьев решений;
- проведено моделирование апостериорной диагностики с использованием темпоральных деревьев решений;
- проведено моделирование диагностики в псевдореальном времени;
- на основе проведенного моделирования сделаны выводы о возможности использования темпоральных деревьев решений для задач диагностики, указаны недостатки рассмотренных алгоритмов и предложены варианты их совершенствования.

Спасибо за внимание!