

# МЕТОД ФОРМИРОВАНИЯ СВЕРТКИ МНОЖЕСТВА АДАПТАЦИОННЫХ ПРАВИЛ В ДИНАМИЧЕСКОЙ СЦЕНАРНО-ПРЕЦЕДЕНТНОЙ СИСТЕМЕ

УДК 004.986

## ШЕРСТЮК Владимир Григорьевич

д.т.н., профессор кафедры информационных технологий Херсонского национального технического университета

**Научные интересы:** методы и модели поддержки принятия решений реального времени, принятие решений на основе прецедентов, мультиагентные системы, комбинированные логические системы представления знаний.

**e-mail:** v\_sherstyuk@bigmir.net

### ВВЕДЕНИЕ

Динамические сценарно-прецедентные системы (ДСПС) предназначены для решения трудноформализуемых задач в слабоструктурированных предметных областях, вследствие чего функционируют в условиях неопределенности исходной информации, а также дефицита времени [1]. Соответственно, к ДСПС предъявляются особые требования по обработке неполной и неточной исходной информации (событий) от нескольких независимых источников в условиях воздействия шумов и искажений, при обеспечении быстрого действия, достаточного для функционирования ДСПС в реальном времени.

В [2] показано, что функционирование ДСПС в реальном времени является процессом правдоподобных рассуждений в соответствии со «здравым смыслом». В качестве основы представления знаний для ДСПС предложена ситуационно-событийная модель, над которой построен формализм правдоподобных древовидных сетей событий (ПДСС) [3]. ДСПС реализуют цикл правдоподобного вывода по прецедентам путем инкрементного формирования заключений для множества активных гипотез, построенных в соответствии с динамической оценкой подобия наблюдаемого потока событий и эталонных потоков событий, содержащихся в прецедентах.

Вместе с тем, вследствие накопления значительного числа прецедентов при наблюдении за реальной предметной областью объем хранилища прецедентов  $M$ , представляемого с помощью ПДСС  $W$ , интенсивно и

неограниченно растет. Существенно сократить число прецедентов, накапливаемых в хранилище  $M$ , возможно, используя методы адаптации [4], сводящиеся к обобщению возможных решений для определенных классов проблемных ситуаций в виде множества адапционных правил (АП)  $\{AR_j\}_{j=1}^n$ , хранимых совместно с описанием проблемы  $\Delta(s)$  и ее возможного решения  $\Delta(r)$ . Каждое АП позволяет сопоставить выявленным отклонениям описания проблемы  $\Delta(s)$  необходимые коррекции (отклонения) решений  $\Delta(r)$ . Однако, в процессе функционирования ДСПС число АП также существенно возрастает, что замедляет вывод по прецедентам и снижает быстродействие ДСПС.

Используя методы машинного обучения, можно сформировать множество АП, содержащее наиболее общие правила, применимые на определенных подмножествах конкретных ситуаций. В то же время, в отличие от известных подходов к организации машинного обучения [5], в которых задачи отбора и верификации наиболее полезных результатов обучения возложены на пользователя, процесс обучения ДСПС вследствие дефицита времени должен быть автоматическим.

В процессе обучения ДСПС можно использовать методы индуктивного обучения [6] и выявления скрытых знаний (*knowledge discovery*) [7]. Однако, использование машинного обучения в ДСПС сопряжено с рядом проблем, к основным из которых относится чрезмерно высоко-

кая вычислительная сложность алгоритмов, так как алгоритмы машинного обучения работают на основе полного перебора обучающей выборки, а также значительное число правил на выходе, большинство из которых являются нерепрезентативными, а значит, и бесполезными для принятия решений.

С первым из указанных недостатков можно бороться, сокращая различными способами объем обучающей выборки, со вторым – используя различные механизмы отсека ненепрезентативных правил. Радикальным решением первой проблемы является также вынос фазы машинного обучения за пределы основного цикла функционирования ДСПС.

**Цель** статьи состоит в разработке метода формирования свертки адаптационных правил, основанного на использовании механизмов машинного обучения и использующего формализм представления знаний в виде ПДСС, который был бы пригоден для использования в ДСПС при осуществлении вывода по прецедентам в реальном времени.

### ПОСТАНОВКА ЗАДАЧИ

Рассмотрим вопросы противодействия неограниченному росту объема хранилища прецедентов (как в смысле накапливаемого числа прецедентов, так и в смысле числа адаптационных правил), негативно влияющего на производительность ДСПС, с этой целью поставим задачу обучения как задачу обобщения на пространстве уточнения решений.

Целью процесса автоматического обучения ДСПС будем считать получение методом обобщения свертки наиболее общих АП, описывающих способы адаптации решений для типовых ситуаций.

Пусть хранилище прецедентов представлено как  $M = \{e_1, \dots, e_N\}$ , где  $N = |M|$ ,  $e_i \in M$  – прецеденты. Всякий прецедент  $e_i \in M$  может быть представлен вектором  $e_i = (x_{i1}, \dots, x_{in}, g_i, \bar{e}_i)$ , где  $\bar{e}_i$  – ссылка на решение прецедента  $r_i \in W$  в форме куста ПДСС  $Y_{g_i}$ ,  $g_i \in C$  – целевое состояние, а  $(x_{i1}, \dots, x_{in})$  – вектор оценки значений параметров контекста  $\{v_{e_i}^1, v_{e_i}^2, \dots, v_{e_i}^n\} \in \text{Ctx}(e_i)$ , причем  $(x_{i1}, \dots, x_{in}) \in W$ .

Пусть для каждого параметра контекста  $v_{e_i}^j$  задан его вес  $w_j$ , такой что  $w_j \in [0, 1] \quad \forall j \in [1 \dots n]$ . Зададим также predetermined на этапе синтеза ДСПС линейный порядок  $\leq_w$  на множестве параметров контекста.

Тогда *задача обучения* ДСПС сводится к задаче обобщения на пространстве уточнения решений в соответствии с [8].

Для заданных:

- множества пар различающихся прецедентов  $\{(e_k, e_l), \dots\}$  (экземпляров), образующих обучающую выборку  $TS$ ;

- языка описания экземпляров обучающей выборки  $\Lambda = \Lambda(s) \times \Lambda(r)$ ;

- языка описания отклонений  $\Lambda(\Delta) = \Lambda(\Delta(s)) \times \Lambda(\Delta(r))$ ;

- иерархии отклонений  $(\Lambda(\Delta), \leq_\Delta)$ , индуцирующей операции включения ( $\subseteq$ ) / вложения ( $\supseteq$ );

- множества зависимостей между экземплярами  $TS$  и их обобщениями, заданных в виде АП, таких что  $AR = (\Delta(s) \Rightarrow \Delta(r))$  покрывает

$AR_{kl} = (\Delta(s)^{kl} \Rightarrow \Delta(r)^{kl})$ , если

$(\Delta(s), \Delta(r)) \subseteq (\Delta(s)^{kl}, \Delta(r)^{kl})$ ,

задача обучения состоит в нахождении множества обобщенных адаптационных правил  $AR_j$ , каждое из которых совместимо с  $TS$  и покрывает минимум один из экземпляров  $TS : \exists AR_{kl} \in TS \mid AR_{kl} \models AR_j$ .

### РЕШЕНИЕ ЗАДАЧИ

Процесс обучения ДСПС в соответствии с [7] может быть представлен следующей последовательностью операций (рис. 1):

- загрузки данных, при этом производится выборка некоторого подмножества прецедентов  $\{e_i, \dots, e_q\} \in M$  в векторном формате представления;

- нормализации выборки, связанной с выполнением атрибутной кластеризации хранилища прецедентов для сокращения размера обучающей выборки  $TS$ ;

- предобработки выборки, заключающейся в построении отклонений  $\Delta(s)^{kl} \Rightarrow \Delta(r)^{kl}$  для каждой пары прецедентов  $(e_k, e_l)$  на основе отклонений значений

параметров  $(\Delta_{x_{kl1}} = (x_{k1} - x_{l1}), \dots, \Delta_{x_{klm}} = (x_{km} - x_{lm}))$ ,  $\Delta_s^{kl} = (\Delta_{x_{kl1}}, \dots, \Delta_{x_{klm}})$  (в зависимости от особенностей предметной области операция может потребовать также очистки значений параметров от шумов, сглаживания, фильтрации и т.д. [9]).

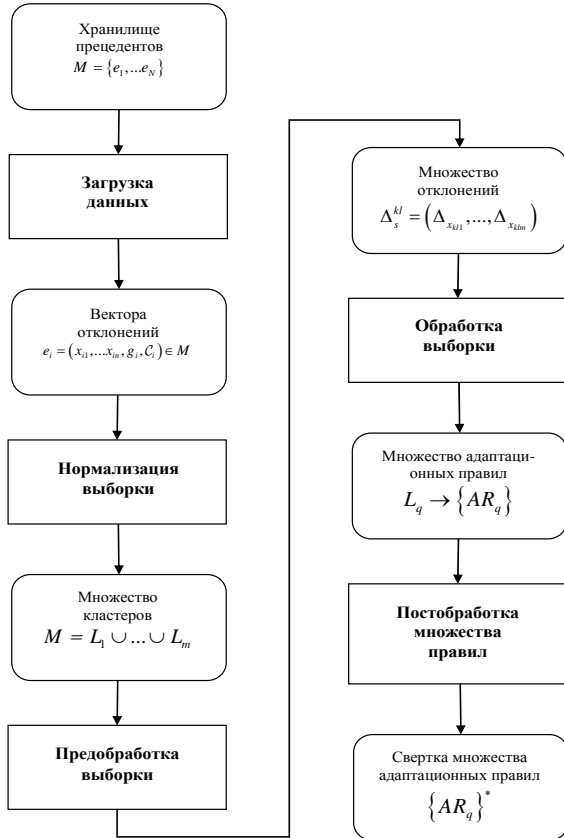


Рис. 1. Процесс обучения ДСПС

- обработки обучающей выборки  $TS$ , связанной с использованием одного или нескольких алгоритмов *data mining* для отыскания АП, являющихся обобщением некоторого подмножества обучающей выборки  $TS$ ;
- постобработки результатов.

Кластеризация хранилища прецедентов используется для ускорения операций выборки сходных прецедентов, при этом хранилище  $M$  разбивается на кластеры  $L_1, \dots, L_m$  на основе взвешенной меры близости  $d_{kl}^{(w)}$  согласно весам  $w_j$  относительно  $\leq_w$ .

Для заданного набора весов признаков  $w_j$  и пары прецедентов  $(e_k, e_l)$  определим взвешенную меру близости  $d_{kl}^{(w)}$

$$d_{kl}^{(w)} = d^{(w)}(e_k, e_l) = \sqrt{\sum_{j=1}^n w_j^2 (x_{kj} - x_{lj})^2}, \quad (1)$$

и, на ее основе, меру сходства  $SIM_{kl}^{(w)}$  прецедентов  $e_k$  и  $e_l$ :

$$SIM_{kl}^{(w)} = 1 / (1 + \alpha \cdot d_{kl}^{(w)}), \quad (2)$$

где  $\alpha$  – ненулевой положительный коэффициент.

Если все веса  $w_j$  равны 1, то мера близости является евклидовой мерой и обозначается как  $d_{kl}$ , а соответствующая ей мера сходства –  $SIM_{kl}$ .

В основу метода кластеризации положено понятие матрицы сходства, определяемой на множестве пар прецедентов  $\{(e_k, e_l), \dots\}$  в соответствии с (1), (2) [10]. Модифицированная матрица сходства заполняется оценками  $sim_{kl} = \max_j [\min(SIM_{kj}^{(w)}, SIM_{jl}^{(w)})]$ . Далее матрицу сходства можно преобразовать в матрицу эквивалентности, используя значимый порог неразличимости  $\beta \in [0..1]$ .

**Определение 2.** Прецеденты  $e_k$  и  $e_l$  принадлежат одному и тому же кластеру  $L_k$ ,  $e_k, e_l \in L_k$ , если и только если  $e_k \cong e_l$  и  $sim_{kl} \geq \beta$ .

Опред. 2 предоставляет возможность группировать подмножества сходных друг другу с оценкой  $sim_{kl} \geq \beta$  прецедентов в кластеры, причем результат кластеризации зависит исключительно от упорядоченности оценок весов параметров  $w_j$ , используемых при вычислении матрицы сходства. Если хранилище прецедентов  $M$  разбито на  $m$  кластеров,  $\mathbb{L} = \{L_1, \dots, L_m\}$ , то для любых  $i \in [1..m]$  и прецедента  $e_k \in L_i$

$$\min_{i \in L_i, l \neq k} SIM_{kl}^{(w)} > \max_{i \leq j \leq m, j \neq i} (\max_{l \in L_j} SIM_{kl}^{(w)}), \quad (3)$$

т.е. сходство между любыми прецедентами, принадлежащими одному и тому же кластеру, всегда больше, чем между любым из прецедентов данного кластера и любым из прецедентов любого другого кластера.

С помощью кластера  $L_q$  в хранилище прецедентов представлены множества конкретных ситуаций, соответствующие некоторым типовым ситуациям  $s_q$ . Для вы-

полнения кластеризации можно использовать алгоритмы атрибутно-ориентированной индукции [11].

На этапе предобработки выборки для каждого кластера  $L_q = \{e_1, e_2, \dots, e_m\}$ , содержащего прецеденты в форме  $e_i = (x_{i1}, \dots, x_{in}, g_i, \bar{e}_i)$ , на каждой паре  $(e_k, e_l) \in L_q$  можно сформировать множество векторов отличия.

**Определение 3.** Всякий прецедент  $e_k \in L_q, k = [1..m]$  может быть представлен как *вектор отличия*  $\bar{V}_{lk}$  относительно другого прецедента  $e_l \in L_q, l = [1..m]$ :

$$\begin{aligned} \bar{V}_{lk} &= \Delta(e_l - e_k) = \\ &= ((x_{l1} - x_{k1}), (x_{l2} - x_{k2}), \dots \\ &\dots, (x_{ln} - x_{kn}), (g_l \odot g_k), (\bar{e}_l \odot \bar{e}_k)) = \\ &= \{\Delta_{xlk1}, \Delta_{xlk2}, \dots, \Delta_{xlkn}, \Delta_{glk}, \Delta_{clk}\}, \end{aligned} \quad (4)$$

где  $|\bar{V}_{lk}| = n + 2$ .

Вектор отличия является экземпляром соответствующего отклонения прецедентов  $\Delta^{lk} = (\Delta_s^{lk} = (\Delta_{xlk1}, \dots, \Delta_{xlkn}), \Delta_r^{lk} = (\Delta_{glk}, \Delta_{clk}))$ , что может быть использовано далее в процессе *обработки* кластера  $L_q$ . Пусть  $X = \{X_1, \dots, X_n\}$  – множество параметров.

Назовем *набором признаков* подмножество  $\mathfrak{Z} \in 2^X$  множества параметров  $X$ .

На вход алгоритма *data mining* подаются т.наз. транзакции, которые представляют собой множества наборов признаков [9]. На выходе требуется получить множество обобщенных АП  $AR_j$ .

*Значимость* правила  $AR_j$  оценивается относительным охватом обучающей выборки

$$s(AR_j) = \left| \{AR_{kl} \in TS \mid AR_{kl} \models AR_j\} \right| / |TS|, \quad (5)$$

Нетрудно заметить, что  $AR_i \models AR_j \Leftrightarrow s(AR_i) \leq s(AR_j)$ .

Значимость  $s(\mathfrak{Z})$  набора признаков  $\mathfrak{Z}$  есть отношение числа транзакций  $m_{\mathfrak{Z}} \supseteq \mathfrak{Z}$ , содержащих набор признаков  $\mathfrak{Z}$ , к общему числу транзакций  $m$  [12].

Введем *порог значимости*  $\sigma_{\min} \in [0, 1]$ .

**Определение 4.** Набор признаков  $\mathfrak{Z}$  называется *часто встречающимся* для заданного порога  $\sigma_{\min}$  если  $s(\mathfrak{Z}) \geq \sigma_{\min}$ .

**Определение 5.** Набор признаков  $\mathfrak{Z}$  называется *замкнутым*, если добавление нового признака изменяет его значимость:

$$s(\mathfrak{Z}) > s(\mathfrak{Z} \cup \{X_q\}) \mid \forall X_q \in X, X_q \notin \mathfrak{Z}, \quad (6)$$

Если существует  $n$  АП  $AR_{kl} \in TS$ , таких что  $AR_{kl} \models AR_1 \wedge AR_2 \wedge AR_3$  и  $\mathfrak{Z} = \{AR_1, AR_2, AR_3\}$ , набор признаков  $\mathfrak{Z}$  является часто встречающимся в случае  $n \geq \sigma_{\min} \times |TS|$ .

Если  $\mathfrak{Z}$  незамкнут, он представляет собой чрезмерное (бесосновательное) обобщение [13], основанное на нерепрезентативных экземплярах выборки.

Задачей переработки обучающей выборки является выявление всех часто встречающихся замкнутых наборов признаков (*frequent closed itemset*, FCI). Решение данной задачи может быть получено использованием ряда алгоритмов, нацеленных на получение FCI, как по отдельности, так и в различных комбинациях, например A-Priori, Charm, GenMAX, Eclat и др. [14].

Порог значимости  $\sigma_{\min}$  позволяет отрегулировать число АП, составляющих результат: чем выше  $\sigma_{\min}$ , тем меньше число обобщенных правил. Завышая значение  $\sigma_{\min}$ , можно ограничить число генерируемых правил, но существует риск «пропустить» полезные правила меньшей общности. Занижение  $\sigma_{\min}$  приведет к увеличению числа результирующих правил, что приведет к резкому увеличению («экспоненциальному взрыву») вычислительной сложности адаптации сформированных решений.

Несмотря на то, что большинство известных алгоритмов поиска FCI способны сокращать число генерируемых правил путем отсеивания (*pruning*) тех правил, оценка интересности или полезности для которых не превышает установленный порог, проблема адекватного подбора значения  $\sigma_{\min}$  существенна, т.к. адаптационные знания по определению сильно зависимы от контекста ситуации, поэтому полезные правила специфичны (конкретны) и могут затеряться среди значительного множества сгенерированных более конкретных правил, либо быть отсеиваны при генерации компактного множества правил боль-

шей общности. Углубленный обзор существующих алгоритмов *data mining* в приложении к обработке больших хранилищ прецедентов с анализом способов генерации компактных множеств правил дан в [15].

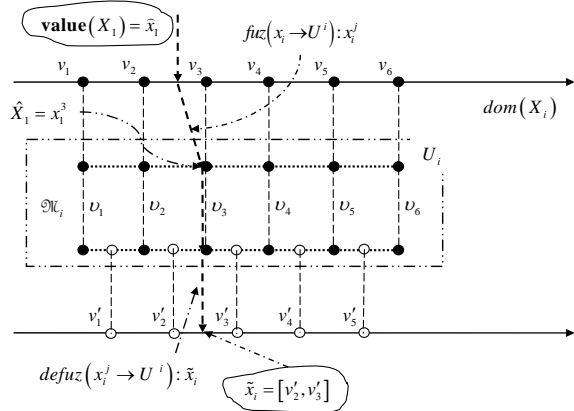
В результате выполнения нормализации, предобработки и собственно обработки обучающую выборку разбивают на кластеры  $L_1, \dots, L_m$ , и для каждого кластера  $L_q$  строят множество АП  $\{AR_q\}$ , которое естественным образом ранжировано по значимости  $s(AR_q)$  составляющих его правил.

Целесообразным способом постобработки множества АП является его индуктивное обобщение с использованием алгоритма С4.5 [16], который обычно используется для построения деревьев решений (иерархической классификационной модели). В нашем случае дерево решений уже построено на фазе обработки выборки, поэтому алгоритм С4.5 будем использовать для его обобщения путем обучения на примерах. К достоинствам алгоритма следует отнести высокую скорость процесса обучения и простоту интерпретации, возможность обработки пропущенных значений.

Исходя из опред. 3, в кластере  $L_q$  все прецеденты  $\{e_1, e_p\} \in L_q$  представлены в виде вектора отличия  $\bar{V}_{ik} = \{\Delta_{xk1}, \Delta_{xk2}, \dots, \Delta_{xkn}, \Delta_{gk}, \Delta_{clk}\}$ .

Сформируем набор признаков (атрибутов), каждый из которых имеет дискретное или числовое значение, при этом  $P^i = \Delta_{xki} \forall i \in [1, n]$ , и введем классификационный признак  $P^{n+1}$ . Введем для каждого атрибута  $P^i$  множество-носитель значений  $Q^i = \{v_1, \dots, v_m\}$ , сформируем решетку  $U^i = \langle Q^i, \leq \rangle$ , и определим функцию  $Range(dom(X_i) \xrightarrow{Q^i} U^i)$ , разбивающую область значений параметра  $X_i$ , соответствующего атрибуту  $P^i$ , на  $m$  интервалов.

Зададим для каждого атрибута  $P^i$  функцию фаззификации  $fuz(x_i \rightarrow U^i): x_i^j$  и обратную ей функцию дефаззификации  $defuz(x_i^j \rightarrow U^i): \tilde{x}_i$  (рис. 2).



$$Q_i = \{v_1, v_2, v_3, v_4, v_5, v_6\} \quad U_i = (Q_i, \leq) \quad m = 6$$

Рис. 2. Фаззификация и дефаззификация атрибутов

Выполненное отображение значений каждого атрибута  $P^i \in \bar{V}$  на решетку  $U^i$  позволяет для числовых параметров  $X_i$  перейти от бесконечного множества значений, определенных на  $\mathbb{R}$ , к компактному  $m$ -ограниченному множеству значений на  $Q^i$ . В качестве множества примеров используем множество векторов различия  $\{\bar{V}_{kl}\}$ ,  $e_k, e_l \in L_q$ , имеющее размерность  $(p-1)!$ .

Если рассматривать каждый вектор отклонения как пример, можно на его основе сформировать  $(n+1)$  столбцов таблицы решений  $\nabla$ , а множество векторов различия даст  $(p-1)!$  строк, причем каждая ячейка таблицы может иметь не более  $m$  различных значений (на практике для каждого атрибута требуется различная мощность множества-носителя, поэтому  $m$  представляет собой максимальное значение мощности носителя для всех  $n$  атрибутов).

Так как всякое АП  $AR_j$  может быть представлено в виде конъюнктивной нормальной формы  $(P^1 = V_j^1) \wedge \dots \wedge (P^n = V_j^n) \Rightarrow (U = V_j^{n+1})$ , где  $V_j^i$  – значение атрибута  $P^i$ , АП может быть преобразовано в строку таблицы, а всякий путь от корня дерева решений к конечному его листу однозначно конвертируется в АП, и наоборот [17].

Разобьем кластер  $L_q$  на множество подкластеров  $L_q = \{L_q^1, \dots, L_q^m\}$ .

Поскольку дерево решений уже построено на фазе обработки выборки, относительная частота некоторого узла  $V_N$  дерева решений в кластере  $L_q^j$  составляет значимость соответствующего АП  $AR_N$ :

$$P_q(V_N) = s(AR_N). \quad (7)$$

Энтропия ( $E$ ) узла  $V_N$  по кластерам  $L_q^j$  определяется как:

$$E(V_N) = \sum_{j=1}^m s(AR_N) [1 - s(AR_N)] \quad (8)$$

Таким образом, для каждого атрибута  $P^i$  существует  $m$  дочерних узлов представляющего атрибут узла  $V_S$ . Тогда информационная значимость атрибута  $P^i$  в узле  $V_S$  определяется как:

$$Info(P^i, V_S) = E(V_S) - \sum_{i=1}^m \frac{s(V_S \cap A_j^i)}{\sum_{j=1}^m s(V_S \cap A_j^i)} \cdot E(V_S \cap P_j^i). \quad (9)$$

Постобработка дерева решений алгоритмом С4.5 принципиально проста [18]. Имеется корень и ассоциированная с ним таблица  $\nabla$ , которую необходимо разбить на подмножества. Для этого случайно выбирается один из узлов  $V_S$ , и один из атрибутов  $P^i$  в узле  $V_S$  в качестве проверки. Вычисляем величину  $Info(P^i, V_S) > 0$ , и выбираем  $i$  такое, что  $E(V_S) = \arg \max_{1 \leq i \leq m} Info(P^i, V_S)$ .

Выбранный атрибут имеет  $m$  значений, что даёт разбиение на  $m$  подмножеств. Далее создаются  $m$  потомков корня, каждому из которых поставлено в соответствие своё подмножество, полученное при разбиении  $\nabla$ . Процедура выбора атрибута и разбиения по нему рекурсивно применяется ко всем  $m$  потомкам, для которых  $Info(P^i, V_S) > 0$  и останавливается в двух случаях:

- после очередного ветвления в вершине оказываются примеры из одного класса, для которых  $s(V_N)$  превышает заданный порог  $\eta$  (тогда она становится листом, а класс, которому принадлежат её примеры, будет решением листа),

- вершина оказалась ассоциированной с пустым множеством (тогда она становится листом, а в качестве решения выбирается наиболее часто встречающийся класс у непосредственного предка этой вершины).

Таким образом, алгоритм обучения на примерах становится правдоподобным эвристическим [616]. В резуль-

тате его работы получаем свертку множества правил  $\{AR_q\}^*$  по  $m$ , являющуюся индуктивным обобщением множества  $\{AR_q\}$  (рис. 3).

АП в обобщенном дереве решений соответствуют всем возможным путям от его корня к листьям, т.е. число таких правил равно числу листьев.

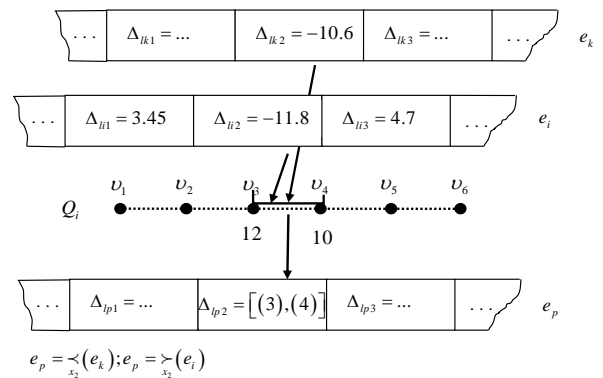
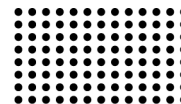
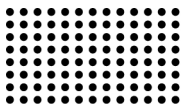


Рис. 3. Принцип обобщения адапционных правил

### ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ

В статье предложен метод формирования свертки множества адапционных правил, основанный на использовании моделей и методов машинного обучения. Метод реализуется правдоподобным эвристическим процессом обучения, интегрирующим фазы концептной кластеризации, переработки обучающей выборки с помощью алгоритмов *data mining* и постобработки множеств АП с использованием индуктивного обучения на примерах, и основан на представлении знаний правдоподобными деревьями решений. В результате выполнения процесса обучения формируется индуктивное обобщение в виде свертки множества адапционных правил.

Предложенный метод позволяет добиться сбалансированности кроны правдоподобных многоуровневых древовидных сетей событий, лежащих в основе хранилища прецедентов в динамической сценарно-прецедентной системе, а также оптимизировать размер хранилища прецедентов, объединяя в кластеры близкие по содержанию прецеденты и обеспечивая необходимые корректировки путем выполнения трансформационных операций. Метод может быть использован в контуре обратной связи в динамических сценарно-прецедентных системах, осуществляющих вывод по прецедентам в реальном времени.

**ЛИТЕРАТУРА**

1. Sherstjuk, V. Scenarno-precedentnoe upravlenie jergaticheskimi dinamicheskimi ob#ektami / V.G. Sherstjuk. – Saarbrücken, Deutschland: Lambert Academic Publishing, 2013. – 407 p.
2. Sherstjuk, V. Osnovy teorii dinamicheskikh scenarno-precedentnyh intellektual'nyh sistem / V.G. Sherstjuk. – Herson: Feniks, 2012. – 476 s.
3. Sherstjuk, V. Ispol'zovanie derev'ev sobytij dlja predstavlenija znaniy v dinamicheskikh precedentnyh intellektual'nyh sistemah / V.G. Sherstjuk // Vestnik Hersonskogo nacional'nogo tehničeskogo universiteta. – 2011. – №2(41). – S.100-111.
4. Sherstjuk V.G. Metod avtomatičeskoj adaptacii rešenij v gibridnoj dinamičeskoj scenarno-precedentnoj intellektual'noj sisteme / V.G. Sherstjuk // Problemy informacionnyh tehnologij. – 2011. – №2(10). – S.64-75.
5. Craw, S. Learning adaptation knowledge to improve case-based reasoning / S. Craw, N. Wiratunga, R. Rowe // Artificial Intelligence. – 2006. – Vol.170. – №16-17. – Pp.1175-1192.
6. Holland, J. Induction: Processes of Inference, Learning, and Discovery / J. H. Holland, K. J. Holyoak, R. E. Nisbett, P. R. Thagard. – Cambridge: MIT Press, 1986. – 416 p.
7. Han, J. Data Mining: Concepts and Techniques / J. Han, M. Kamber. – N.Y.: Morgan Kaufmann, 2011 – 744 p.
8. Mitchell, T. Generalization as search / T. M. Mitchell // Artificial Intelligence. – 1982. – Vol.18. – №2. – Pp.203-226.
9. Witten, I. Data Mining: Practical Machine Learning Tools and Techniques / I. H. Witten, E. Frank, M. A. Hall. – N.Y.: Morgan Kaufmann, 2011. – 664 p.
10. Shiu, S. Transferring case knowledge to adaptation knowledge: An approach for case-base maintenance / S. C. K. Shiu, D. S. Yeung, C. H. Sun, X. Z. Wang // Computational Intelligence. – 2001. – Vol.17. – №2. – Pp.295-314.
11. Cheung, D. Efficient rule-based attribute-oriented induction for data mining / D. W. Cheung, H. Y. Hwang, W. A. Fu, J. Han // Journal of Intelligent Information Systems. – 2000. – Vol.15. – №2. – Pp.175-200.
12. Adamo, J.-M. Data Mining for Association Rules and Sequential Patterns: Sequential and Parallel Algorithms / J.-M. Adamo. – Berlin: Springer-Verlag, 2000. – 264 p.
13. Dan, L. The over-generalization problem: predicates rigidly signifying the “unnatural” / L. de Sa Dan // Synthese. – 2008. – Vol.163. – №2. – Pp.263-272.
14. Zaki, M. Fundamentals of Data Mining Algorithms / M. J. Zaki, W. Meira Jr. – Cambridge: Cambridge University Press, 2009. – 555 p.
15. Pan, R. Mining competent case bases for case-based reasoning / R. Pan, Q. Yang, S. J. Pan // Artificial Intelligence. – 2007. – Vol.171. – №16-17. – Pp.1039-1068.
16. Quinlan, J. C4.5: Programs for Machine Learning / J. R. Quinlan. – San Mateo: Morgan Kaufmann Publishers Inc., 1993. – 302 p.
17. Levitin, A. Algoritmy: vvedenie v razrabotku i analiz / A. V. Levitin. – M.: Vil'jams, 2006. – 576 s.
18. Mitra, S. Fuzzy decision tree, linguistic rules and fuzzy knowledge-based network: generation and evaluation / S. Mitra, K. M. Konwar, S. K. Pal // IEEE Transactions on Systems, Man, and Cybernetics. Part C: Applications and Reviews. – 2002. – Vol.32. – №4. – Pp.328-339.