УДК 004.832.38

Стохастический поиск прообразов на основе кластеризации знаний в нечетком LP-выводе

А.Н. Шмарин – *аспирант, Воронежский государственный университет*
E-mail: tim-shr@mail.ru

Рассмотрена реализация способа стохастического поиска начальных прообразов, основанная на кластеризации представления нечетких знаний. Сделан вывод о возможности использования полученных результатов для разработки интеллектуальных систем продукционного типа, работающих с неполными знаниями, а также минимизирующих количество ресурсоемких запросов о значениях признаков классифицируемых объектов предметной области.

Ключевые слова: *LP-вывод, бинарное отношение, фактор-множество, кластеризация, стохастический алгоритм, машинное обучение, Python.*

Fuzzy LP-inference is the classification method based on the search the initial preimages – algebraic lattice elements, that composed of the initial atoms and are logically connected with some atom corresponding to feature class. This paper presents the implementation of the stochastic initial preimages search, that is based on the clustering of the fuzzy knowledge representation.

Keywords: *LP-inference, binary relation, quotient set, clustering, stochastic algorithm, machine learning, Python.*

С увеличением объемов и сложности обрабатываемой информации все более актуальным становится использование систем искусственного интеллекта. При этом одна из распространенных моделей представления знаний – продукционная [1]. Алгоритмы поиска решений в таких системах основаны на механизме логического вывода. На практике существует большое число задач принятия решений в условиях неопределенности, для которых получение информации о значениях признаков классифицируемых объектов некоторой предметной области является ресурсоемкой операцией. Такие задачи возникают, например, при исследовании роботом поверхности планеты. Пусть цель робота – достижение определенной скалы, однако будущий маршрут наблюдается лишь отчасти. Тогда робот должен искать оптимальное решение, позволяющее достичь цели с учетом ограниченности ресурсов, доступных для дополнительной разведки местности. Другим примером, где стоимость получения новой информации может быть высокой, является коммерческая медицина. Для минимизации издержек целесообразно находить приемлемый способ лечения с помощью минимального числа анализов, выполненных за минимальное время. Задержка в предоставлении лечения, вызванная проведением дополнительных анализов, приводит к увеличению затрат. Более того, состояние пациента, не дождавшегося помощи, может существенно ухудшиться. Стоимость дополнительных исследований существенна и в сфере разведки полезных ископаемых. Может оказаться, что более выгодное решение состоит в том, чтобы приступить к бурению скважины при уверенности в успехе 95%, чем потратить еще значительные средства для достижения 98%-ной уверенности.

С фактором ресурсоемкости операций получения данных для принятия решения связана задача минимизации числа запросов о признаках классифицируемого объекта (из некоторой предметной области) в ходе логического вывода. Данная задача является NP-трудной [2]. Для достижения глобальной минимизации числа внешних запросов предложен общий метод *LP*-вывода [3]. К сожалению, он обладает экспоненциальной вычислительной сложностью относительно числа атомарных фактов в базе знаний. Идея кластерно-релевантного LP-вывода [4] основана на вычислении специфических оценок (показателей релевантности) для ограниченного подмножества продукций и на обобщении полученных оценок на все множество продукций. Исследования и эксперименты показывают [5], что при использовании метода LP-вывода, по сравнению с обычным обратным выводом, снижение числа внешних запросов составляет в среднем 15–20%.

Ранее [6] обсуждались вопросы реализации вычисления приближенной оценки числа слоев без циклов в задаче нечеткого LP-вывода.

В настоящей работе рассматривается математическая формулировка задачи нечеткого LP-вывода, исследуется способ кластеризации нечетких знаний и предлагается основанная на нем программная реализация стохастического поиска начальных прообразов. Представленные результаты предназначены для итеративного анализа таких подмножеств продукций, которые наиболее существенно влияют на показатель релевантности.

Метод нечеткого LP-вывода

Обозначим через X множество объектов (ситуаций, прецедентов) некоторой предметной области. Например, в задачах машинного обучения, встречающихся в медицине, объектами могут являться пациенты, в сфере кредитования – заемщики, в задаче фильтрации спама – отдельные сообщения.

Признак – результат измерения некоторой характеристики объекта, то есть отображение $\text{feature}: X \to D_{\text{feature}}$, где D_{feature} – множество допустимых значений признака. Значениями признаков в прикладных задачах могут быть числовые последовательности, изображения, тексты, функции, графы, результаты запросов к базе данных и т.д.

Пусть имеется набор признаков $\text{feature}_1,\ldots,\text{feature}_n$ которые соответствуют характеристикам объектов $x \in X$ некоторой предметной области. Вектор $\left(\text{feature}_1(x),\ldots,\text{feature}_n(x)\right) \in D_{\text{feature}_1} \times \ldots \times D_{\text{feature}_n}$ называется признаковым описанием объекта $x \in X$. Признаковое описание можно отождествлять с самими объектами, то есть $X = D_{\text{feature}_1} \times \ldots \times D_{\text{feature}_n}$.

Предположим, что объекты $x \in X$ разделены некоторым образом на классы, которым соответствует конечное множество их номеров (имен, меток) Y. Пусть также задана продукционная база знаний, отражающая целевую зависимость – отображение $y^*(x) \in Y$.

Задача нечеткого LP-вывода состоит в том, чтобы как можно достовернее классифицировать произвольный объект $x \in X$, то есть найти для него класс $y^*(x) \in Y$, используя для этого как можно меньшую выборку значений из признакового описания этого объекта.

Далее рассмотрим более формализованную постановку задачи в терминах алгебраических решеток и бинарных отношений.

Пусть задано конечное множество $F = \{f\}$. На его основе определим атомно-порожденную ограниченную алгебраическую решетку $L = \lambda(F)$, где λ – функция-булеан [7]. На решетке зададим дополнительное бинарное отношение $R = \{r = (\tau, f) : \tau \in L, f \in F\}$, являющееся каноническим [5].

Применительно к продукционным логическим системам такое отношение R соответствует множеству продукций, F (множество атомов решетки) соответствует элементарным фактам продукционной системы, а сама решетка L соответствует множеству предпосылок и заключений продукций. Атом x решетки L называется начальным при отношении R, если в R нет ни одной пары вида (τ, x).

Обозначим множество всех начальных атомов решетки как F_{init}: $F_{\text{init}} = \left\{ f \in F : \mathsf{y}(\tau, f) \in R, \tau \in L \right\}$. Также обозначим множество всех атомов решетки, не являющихся начальными: $F_{\text{notinit}} = \left\{ f \in F : \exists(\tau, f) \in R, \tau \in L \right\}$.

Упорядоченная пара $(\alpha, \beta) \in L, \alpha \in L, \beta \in L$ называется дистрибутивно связанной отношением R, если существует такое множество $\left\{(\alpha_1, \beta_1),\ldots,(\alpha_p, \beta_p)\right\}$, что $\forall i = \overline{1, p}$, выполняется следующее условие: $\alpha_i \subseteq \alpha \wedge \beta \subseteq \beta_i \wedge \left[(\alpha_i = \beta_i \in L) \vee (\alpha_i, \beta_i) \in R\right]$.

Упорядоченная пара (α, β), $\alpha \in L$, $\beta \in L$ называется логически связанной отношением R, если она дистрибутивно связана отношением R, либо существует упорядоченный набор $(\alpha, \gamma_1,\ldots,\gamma_l, \beta)$, $\gamma_i \in L$, $i = \overline{1, l}$ такой, что каждая пара в последовательности $(\alpha, \gamma_1), (\gamma_1, \gamma_2),\ldots,(\gamma_{l-1}, \gamma_l), (\gamma_l, \beta)$ дистрибутивно

связана отношением *R*. Если пара (α, β) логически связана отношением *R*, то β будем называть образом α при отношении *R*, а α – прообразом β при отношении *R*. Прообраз в атомно-порожденной решетке называется начальным, если все его атомы являются начальными (при отношении *R*).

Введем отображение $M_R : R \to [0,1]$, которое каждому элементу отношения *R* ставит в соответствие степень истинности: $M_R(r) = \mu$, $r \in R$, $\mu \in [0,1]$.

Обозначим также $M_{\text{init}} : F_{\text{init}} \to [0,1]$ – отображение, сопоставляющее каждому начальному атому решетки некоторую степень истинности.

Применение отображения M_{init} соответствует выполнению запроса о неизвестном значении некоторого признака классифицируемого объекта и может оказаться ресурсоемкой операцией.

В приложении к продукционным логическим системам отображение M_R соответствует степеням истинности продукций, а M_{init} – степеням истинности значений признаков. Степень истинности – это числовой коэффициент $\mu \in [0,1]$, характеризующий достоверность утверждения. Так, если продукция вида «**если** $x_1 \wedge \ldots \wedge x_n$ **то** x_{n+1}» соответствует некоторому элементу отношения $r \in R$, то значение $M_R(r) \in [0,1]$ будет означать степень, с которой истинно утверждение данной продукции. Аналогично, если для объекта предметной области $x \in X$ значение некоторого его признака $\text{feature}(x) \in D_{\text{feature}}$ соответствует атому $f \in F_{\text{init}}$, то значение $M_{\text{init}}(f) \in [0,1]$ будет представлять степень, с которой истинно утверждение «признак feature объекта *x* равен значению $\text{feature}(x)$».

В введенных обозначениях задача нечеткого LP-вывода заключается в том, чтобы в некотором подмножестве $F_C \subseteq F_{\text{notinit}}$ неначальных атомов решетки найти элемент, обладающий наибольшей степенью истинности $M_R(X)$, применив при этом отображение M_{init} как можно меньшее число раз. Данный элемент представляет решение, имеющее наибольшую степень истинности.

Множество F_C будет соответствовать некоторому множеству классов, на которые разделены объекты предметной области.

Процесс поиска прообраза с наибольшей степенью истинности $f_c \in F_C$ состоит из нескольких шагов. В первую очередь для каждого атома решетки, принадлежащего множеству $F_C \subseteq F_{\text{notinit}}$, выполняется поиск начальных прообразов при отношении *R* (то есть поиск для атома логически связанных с ним отношением *R* подмножеств начальных атомов). Затем на основе результатов поиска вычисляются степени истинности соответствующих атомов решетки. В заключение выбирается атом с наибольшей степенью истинности.

Предположим, что заданы некоторые бинарные операции *T* и T_{conorm} такие, что *T* является операцией *t*-нормы, а T_{conorm} – соответствующей ей операцией *t*-конормы [8]. Вычисление степеней истинности выполняется для всевозможных атомов $f_c \in F_C$ и соответствующих им начальных прообразов. При этом используются отображения M_R, M_{init}, операции *t*-нормы и *t*-конормы, а также обобщенное правило вывода modus ponens.

По каждому атому $f_c \in F_C$ формируется подмножество $R_1 \subseteq R$, которое логически связывает выбранные атом *x* и начальный прообраз α. Нахождение степени истинности некоторого неначального атома *f*, участвующего в логической связи α и *x*, начинается с выбора подмножества элементов, содержащих атом *f* в своей правой части: $R_f = \{(\tau, f) \in R_1, \tau \in L\}$. Обозначим мощность данного подмножества символом $n = |R_f|$. Тогда, $R_f = \{r_1, \ldots, r_n\} = \{(\tau_1, f), \ldots, (\tau_n, f)\}$, где $\tau_i \in L$, $1 \le i \le n$. По определению, элементы τ_i решетки *L* состоят из ее атомов. То есть, $\tau_i = \{g_{i,1}, \ldots, g_{i,m_i}\}$, где $g_{i,j} \in F$ – атомы решетки, m_i – число таких атомов в τ_i, $1 \le j \le m_i$, $1 \le i \le n$.

Используя введенные обозначения, можно выразить формулы для вычисления $\mu(f)$ – степени истинности некоторого атома *f*. Если $f \in F_{\text{notinit}}$, то атом является неначальным, а его степень истинности задается рекурсивно:

$$\mu(f) = T_{\text{conorm}}\left(T\left\{T\left[\mu(g_{1,1}),\ldots,\mu(g_{1,m_1})\right], M_R(r_1)\right\},\ldots,T\left\{T\left[\mu(g_{n,1}),\ldots,\mu(g_{n,m_n})\right], M_R(r_n)\right\}\right),$$

где $r_i = (\tau_i, f) \in R_f$, $\tau_i = \{g_{i,1},\ldots,g_{i,m_i}\}$, $g_{i,j} \in F$, $1 \le j \le m_i$, $1 \le i \le n$, T и T_{conorm} – операции *t*-нормы и *t*-конормы соответственно.

Если $f \in F_{\text{init}}$, то есть атом является начальным, то $\mu(f) = M_{\text{init}}(f)$. Применение отображения M_{init} к некоторому начальному атому означает нахождение для этого атома соответствующей степени истинности. Поскольку атом является начальным, его степень истинности невозможно вывести на основе отношения *R* или степеней истинности других атомов решетки.

Для нахождение такой степени истинности потребуется внешний запрос о значении неизвестного признака объекта. Минимизация числа подобных запросов достигается за счет специального порядка применения отображения M_{init} к начальным атомам решетки. В первую очередь данное отображение следует применять к тем начальным атомам, которые входят в наибольшее число найденных прообразов. Кроме того, соответствующие прообразы должны содержать как можно меньшее число элементов.

Порядок применения отображения M_{init} к начальным атомам решетки определяется на основе их так называемых показателей релевантности [9]. Нахождение их точных значений является вычислительно трудной задачей [2]. Таким образом, для практического использования целесообразно вычислять показатели релевантности для ограниченных «кластеров» LP-структуры.

Обозначим символом *A* фактор-множество – разбиение бинарного отношения *R* на классы эквивалентности относительно уникальных правых частей его пар. То есть если

$$r,t \in R,\ r = (\tau_r, f_r), t = (\tau_t, f_t),\ \text{то}\ r \square t \Leftrightarrow f_r = f_t,\ A = R/\square \Leftrightarrow A = \left\{\alpha_f = \{r = (\tau, f) : r \in R\} : f \in F\right\}.$$

Данное фактор-множество представляет основу структурного расслоения [3] исходного отношения *R* на частичные отношения. Следует отметить, что в силу своего построения число классов эквивалентности фактор-множества *A* равно числу всех неначальных атомов решетки, то есть $|A| = |F_{\text{notinit}}|$. *Слоем* называется выборка по одному элементу из каждого класса эквивалентности $\alpha \in A : l = \left\{r_1 \in \alpha_1,\ldots,r_{|F_{notinit}|} \in \alpha_{|F_{notinit}|}\right\}$.

Нахождение показателей релевантности основано на поиске прообразов в таких слоях. Слой либо содержит прообраз, либо его элементы образуют цикл и не могут содержать прообраз [9].

Обозначим $S(A) : A \to \lambda(R)$ отображение, переводящее фактор-множество $A = \left\{\alpha_1,\ldots,\alpha_{|F_{\text{notinit}}|}\right\}$ в множество всех слоев, которые *A* описывает.

$$S(A) = \left\{r_1,\ldots,r_{|A|} \mid r_i \in \alpha_i, i = \overline{1,|A|}\right\} = \bigcup_{(r_1,\ldots,r_{|A|}) \in \alpha_1 \times \ldots \times \alpha_{|A|}} \left\{r_1,\ldots,r_{|A|}\right\}.$$

Если фактор-множество *A* состоит из классов $\alpha_1 \ldots \alpha_n$, то число слоев на основе *A* равно числу всевозможных выборок из каждого класса эквивалентности по одному элементу, то есть $|S(A)| = |\alpha_1| \cdot \ldots \cdot |\alpha_n|$.

Для кластеризации множества слоев могут быть использованы свойства логической связанности пар бинарного отношения, ассоциированные с этими парами значения степеней истинности, а также оценки числа слоев без циклов, включающих в себя заданный атом решетки.

Оценка числа слоев без циклов, включающих в себя заданный атом решетки

Ранее [10] были рассмотрены вопросы оценки числа слоев без циклов, включающих в себя заданный атом решетки. Приведем основные определения и результаты из работы [10], а затем рассмотрим свойства монотонности для функции оценки числа слоев без циклов, необходимые для нахождения ее приближенных значений.

Отображения $\mathrm{in}(\alpha, f) = \{(\tau, g) \in \alpha : f \in \tau\}$ и $\overline{\mathrm{in}}(\alpha, f) = \{(\tau, g) \in \alpha : f \notin \tau\}$ переводят класс эквивалентности α в множество пар, которые соответственно содержат либо не содержат в своих левых частях атом *f*.

Отображение $\lambda(X, k, i) : X \to \lambda(X)$, $k = \overline{1, |X|}$, $i = \overline{1, \binom{|X|}{k}}$ переводит множество X в i-й элемент множества всех его k-элементных подмножеств.

Отображение $A_f = \{\alpha \in A \mid \exists(\tau, g) \in \alpha : f \in \tau\} = \{\alpha \in A : in(\alpha, f) \neq \varnothing\}$ переводит фактор-множество A в такое подмножество его классов эквивалентности, что каждый элемент подмножества включает в себя хотя бы одну пару, содержащую f в левой части.

На основе бинарного отношения R построим граф $G = \{v\}$:

$v \in G$, $v = (f, g)$, $f, g \in F \Leftrightarrow \exists(\tau, g) \in R : f \in \tau$.

По графу G, используя некоторый из известных математических методов, построим фундаментальную систему циклов $H = \{h : h \subseteq G\}$. На основе отношения R и системы циклов H определим E – *фундаментальную систему циклов канонического бинарного отношения*. То есть на основе дуг $v = (\tau, g)$, $f, g \in F$, образующих элементы фундаментальной системы циклов $H = \{h\}$, построим множество $E = \{e_1, \ldots e_{|H|}\}$, состоящее из подмножеств отношения R таких, что

$\forall h_i = \{v = (f, g) \in G\} : e_i = \{(\tau, g) \in R : f \in \tau\}$.

Пусть $\varepsilon \subseteq E, \varepsilon = \{e \in E\}$ – подмножество фундаментальных циклов канонического бинарного отношения R, $\dot{\varepsilon}(\varepsilon) = \bigcup_{e \in \varepsilon} \bigcup_{r \in e} r$ – объединение всех пар бинарного отношения, формирующих элементы данного подмножества. Сужение фактор-множества A, описывающее всевозможные слои, которые содержат только фундаментальные циклы из множества ε, определяется следующим отображением:

$$\theta(\alpha, \varepsilon) = \begin{cases} \alpha \cap \dot{\varepsilon}(\varepsilon), \text{ если } \alpha \cap \dot{\varepsilon}(\varepsilon) \neq \varnothing, \\ \alpha, \text{ иначе} \end{cases} \quad \varepsilon \subseteq E,$$

$$\Theta(A, \varepsilon) = \{\theta(\alpha_1, \varepsilon), \ldots, \theta(\alpha_{|A|}, \varepsilon)\}, \varepsilon \subseteq E.$$

Определим специальные отображения $S_{\text{all}}(A, \Theta, \varepsilon)$ и $S_{\text{subsets}}(A, \Theta, \varepsilon)$.

$S_{\text{all}}(A, \Theta, \varepsilon)$ – множество всевозможных слоев, включающих в себя все циклы из множества $\varepsilon \subseteq E$.

$S_{\text{all}}(A, \Theta, \varepsilon) = S(\Theta(A, \varepsilon))$, $\varepsilon \subseteq E$, $\left|S_{\text{all}}(A, \Theta, \varepsilon t)\right| = \left|\theta(\alpha_1, \varepsilon)\right| \cdot \ldots \cdot \left|\theta(\alpha_{|A|}, \varepsilon)\right|$.

$S_{\text{subsets}}(A, \Theta, \varepsilon)$ – множество всевозможных слоев, включающих в себя всевозможные подмножества циклов из множества $\varepsilon \subseteq E$.

В [10] показано, что для фундаментальной системы циклов $E = \{x_1, \ldots, x_l\}$ канонического бинарного отношения $S_{subsets}$ может быть выражено в виде

$$\left|S_{\text{subsets}}(A, \Theta, E)\right| = \sum_{x_1 \in E} \left|S_{\text{all}}(A, \Theta, \{x_1\})\right| - \sum_{\substack{\{x_1, x_2\} \subset E \\ x_1 \neq x_2}} \left|S_{\text{all}}(A, \Theta, \{x_1, x_2\})\right| +$$

$$+ \ldots + (-1)^{l+1} \left|S_{\text{all}}\left(A, \Theta, \{x_1, \ldots, x_l\}\right)\right| = \sum_{k=1}^{l} (-1)^{k+1} \sum_{i=1}^{\binom{l}{k}} \left|S_{\text{all}}(A, \Theta, \{\lambda(E, k, i)\})\right|$$

Определим вспомогательные отображения:

$F(A_1,\Theta,\varepsilon)=\{e\in\varepsilon\mid\forall\alpha_e\in e/\sim\exists\alpha_{A_1}\in A_1/\sim:\alpha_e\subseteq\alpha_{A_1}\},\varepsilon\subseteq E$ – результат фильтрации множества циклов ε, представляющий только те циклы, которые включены в $S(A_1)$ – множество слоев, построенное на основе фактор-множества A_1;

$\bar{A}(i,s,f)=\mathrm{in}\left(\lambda(A_f,i,s),f\right)\cup\overline{\mathrm{in}}(A\ \lambda(A_f,i,s),f),\ f\in F$ – результат фильтрации фактор-множества A, представляющий (i,s)-ю выборку классов эквивалентности из A_f, объединенную с множеством классов эквивалентности, принадлежащих A, из которого исключены соответствующие элементам (i,s)-й выборки из A_f. Причем из принадлежащих фактор-множеству A классов эквивалентности исключаются пары, содержащие в левой части атом f.

Общее число слоев для атома f вычисляется как сумма [10]:

$$W(A,f)=\sum_{i=1}^{|A_f|}\sum_{s=1}^{\binom{|A_f|}{i}}\prod_{\beta\in\bar{A}(i,s,f)}|\beta|.$$

Число слоев с циклами, включающих в себя заданный атом F, вычисляется следующим образом:

$$W_{\text{loops}}=\sum_{i=1}^{|A_f|}\sum_{s=1}^{\binom{|A_f|}{i}}\left|S_{\text{subsets}}\left[\bar{A}(i,s,f),\Theta,F\left(\bar{A}(i,s,f),\Theta,E\right)\right]\right|.$$

Следует отметить, что число элементов в приведенных суммах растет экспоненциально относительно значения $|A_f|$.

Вычисление приближенных оценок числа слоев без циклов и кластеризация множества слоев

Рассмотрим этапы приближенного вычисления оценок числа слоев без циклов, продемонстрируем процесс кластеризации виртуального расслоения бинарного отношения и представим стохастический поиск начальных прообразов.

Реализация приближенного вычисления оценок числа слоев без циклов выполнена на языке Python и является частью библиотеки машинного обучения Fuzzy LPExpert [11].

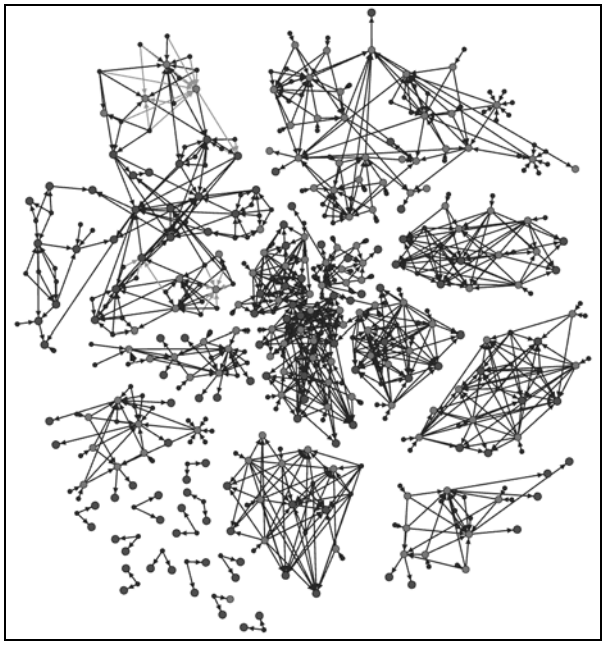

Рис. 1. Граф тестовой базы знаний

Работа программы начинается с чтения базы знаний в формате JSON. Считанная база знаний преобразуется в нормализованное представление: значения признаков отображаются на индексы атомов, правила преобразуются в кортеж («список индексов атомов предпосылки», «единственный индекс атома заключения», «степень истинности»). Кроме того, удаляются атомы, соответствующие номерам классов, но являющиеся начальными. Такое представление является программной абстракцией бинарного отношения R, на основе которого строится граф $G=\{v\}:\ v\in G$, $v=(f,g),\quad f,g\in F\Leftrightarrow\exists(\tau,g)\in R:f\in\tau$. По данному графу с помощью библиотеки graph-tool [12] выполняется построение простых циклов [13], на основе которого строится E – фундамен-

тальная система циклов канонического бинарного отношения, определение которой приведено ранее. Затем, также с помощью библиотеки graph-tool, строится разбиение вершин графа базы знаний на кластеры [14] $G_i = \{ f \in F \}$, $i = \overline{1,m}$. Так, на рис. 1 представлен пример графа тестовой базы знаний, а на рис. 2 – его кластеризация.

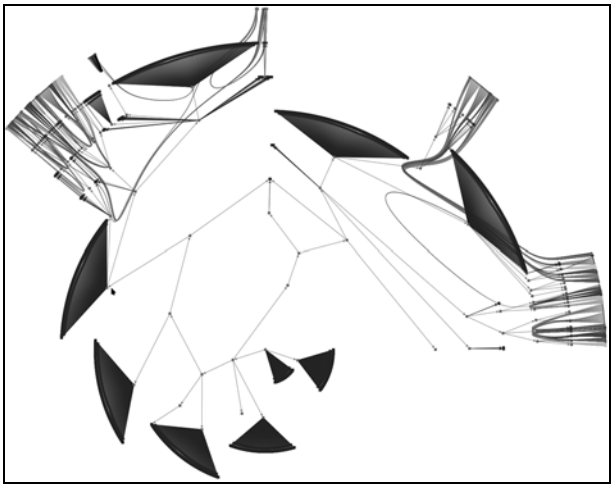

Рис. 2. Кластеризация графа базы знаний

Следующий этап – вычисление для каждого атома решетки медианы степеней истинности правил:

$$\mathrm{med}_{\mu}(f) = \begin{cases} \mathrm{median}\left(\left\{ M_R(r) \mid r = (\tau, f) \in R \right\}\right) \textbf{ if} \\ \mathrm{median}\left(M_{\mathrm{init}}(f)\right), \mathrm{else}. \end{cases}$$

.

Затем рассчитывается приближенная оценка отношения числа слоев без циклов к общему числу слоев:

$$\omega(f) \approx \frac{\overline{W}(A, f) - \overline{W}_{\mathrm{loops}}(A, f)}{\max\limits_{f \in F}\left(\overline{W}(A, f)\right)}, \quad f \in F,$$

где $\overline{W}(A, f)$ – приближенное общее число слоев, включающих атом *f*.

Данное значение нормируется к общему числу слоев в *A*; $\overline{W}_{\mathrm{loops}}(A, f)$ – приближенное число слоев, включающих атом *f* и содержащих циклы. Данное значение также нормируется к общему числу слоев в *A*.

Значения $\overline{W}(A, f)$ и $\overline{W}_{\mathrm{loops}}(A, f)$ вычисляются следующим образом. В конфигурации программы определяются значения C_1, C_2, C_3 и C_4. Если значение $\left|A_f\right| < C_1$, то вычисляется точное значение нормированной оценки $\overline{W}(A, f)$:

$$\overline{W}(A, f) = \frac{1}{2^{|A|}} \sum_{i=1}^{|A_f|} \sum_{s=1}^{\binom{|A_f|}{i}} \prod_{\beta \in \overline{A}(i,s,f)} |\beta|.$$

Если при этом $|E| < C_2$, то вычисляется и точное значение нормированной оценки $\overline{W}_{\mathrm{loops}}(A, f)$:

$$\overline{W}_{\mathrm{loops}} = \frac{1}{2^{|A_f|}} \sum_{i=1}^{|A_f|} \sum_{s=1}^{\binom{|A_f|}{i}} \left| S_{\mathrm{subsets}}\left[\overline{A}(i,s,f), \Theta, F\left(\overline{A}(i,s,f), \Theta, E \right) \right] \right|.$$

Если же $\left|A_f\right| \geq C_1$, то значение нормированной оценки общего числа слоев $\overline{W}(A, f)$ находится с помощью интегрирования методом Монте-Карло [15]:

$$\overline{W}(A, f) = \frac{1}{C_3} \sum_{m=1}^{C_3} \left(\prod_{\beta \in \mathrm{in}(\Upsilon, f) \cup \overline{\mathrm{in}}(A \setminus \Upsilon, f)} |\beta| \right) \Bigg|_{\Upsilon = \mathrm{random\_subset}\left(A_f\right)}.$$

Аналогично при $\left|A_f\right| \geq C_1 \vee |E| \geq C_2$ значение нормированной оценки числа слоев без циклов $\overline{W}_{\mathrm{loops}}(A, f)$ также вычисляется с помощью интегрирования методом Монте-Карло:

$$\overline{W}_{\mathrm{loops}}(A, f) = \frac{1}{C_4} \sum_{j=1}^{|A_f|} \sum_{m=1}^{C_4} \left| S_{\mathrm{subsets}}\left[\overline{A}(i,s,f), \Theta, F\left(\overline{A}(i,s,f), \Theta, \varepsilon \right) \right] \right\|_{\varepsilon = \mathrm{random\_subset}(E):\, |\varepsilon| < C_2}.$$

Таблица. ***Пример списка значений оценок для атомов решетки***

Индекс атома	$\omega(f)$	Номер кластера в графе базы знаний	$\mathrm{med}_{\mu}(f)$
7	0.014	5	0.448
9	0.2	4	0.471
10	0.07	5	0.321
…	…	…	…

В приведенных обозначениях $\mathrm{random\_subset}(X): X \to \lambda(X)$ – отображение, генерирующее по равномерному распределению случайную выборку из множества X. В Python оно реализуется с помощью функции random.sample стандартной библиотеки.

Фактор-множество A строится как словарь, ключами которого являются индексы атомов, а значениями – классы эквивалентности, представляющие собой списки пар отношения R и дополнительные данные, необходимые для быстрого вычисления значений $\hat{A}(i,s,f)$ и $F\left(\hat{A}(i,s,f),\Theta,\varepsilon\right)$.

Особенностью вычисления значений $\hat{W}(A,f)$ и $\hat{W}_{\mathrm{loops}}(A,f)$ является то, что полученные значения произведений выходят за пределы 64 разрядов. Для того, чтобы обойтись без трудоемкой реализации механизма динамической нормализации больших чисел и при этом сохранить достаточную точность вычислений, используются библиотека работы с числами произвольной точности GNU Multiprecision Library и ее Python-интерфейс в виде модуля mpz.

Следующим этапом вычислений является формирование для каждого атома $f \in F$ вектора из трех числовых значений: $v = \left\{\omega(f), i, \mathrm{med}_{\mu}(f)\right\}$, где i – номер кластера графа G. Обозначим множество таких векторов как $V = \{v_1,\ldots,v_{|F|}\}$. В программной реализации данное множество строится как список кортежей, имеющий показанный в таблице вид.

Далее выполняется кластеризация [16, 17] построенного вектора V значений оценок. Этот шаг реализуется с помощью библиотеки scikit-learn [18] и ее функции cluster.DBSCAN, реализующей одноименный алгоритм кластеризации [19]. На рис. 3 приведен пример кластеризации вектора оценок для тестовой базы знаний. Обозначим полученные кластеры символами V_j, $j=\overline{1,n}$, где n – число кластеров. Также, обозначим $f_{j,k}$ атом, соответствующий k-му элементу кластера V_j. Для каждого кластера определим значение вероятности:

$$l_j = \sum_{k=1}^{|V_j|} \sqrt{\omega(f_{j,k})^2 + \mathrm{med}_\mu(f_k)^2}\ ,\ p_j = \frac{l_j}{\sum_{k=1}^{n} l_k}.$$

На заключительном этапе вычислений выполняется стохастический поиск начальных прообразов в виртуальном расслоении бинарного отношения. Для этого в конфигурации программы задаются четыре ограничения: C_t – максимальное время работы, C_i – максимальное число итераций, C_p – максимальное количество найденных уникальных прообразов, C_W – максимальное число слоев в кластере для анализа.

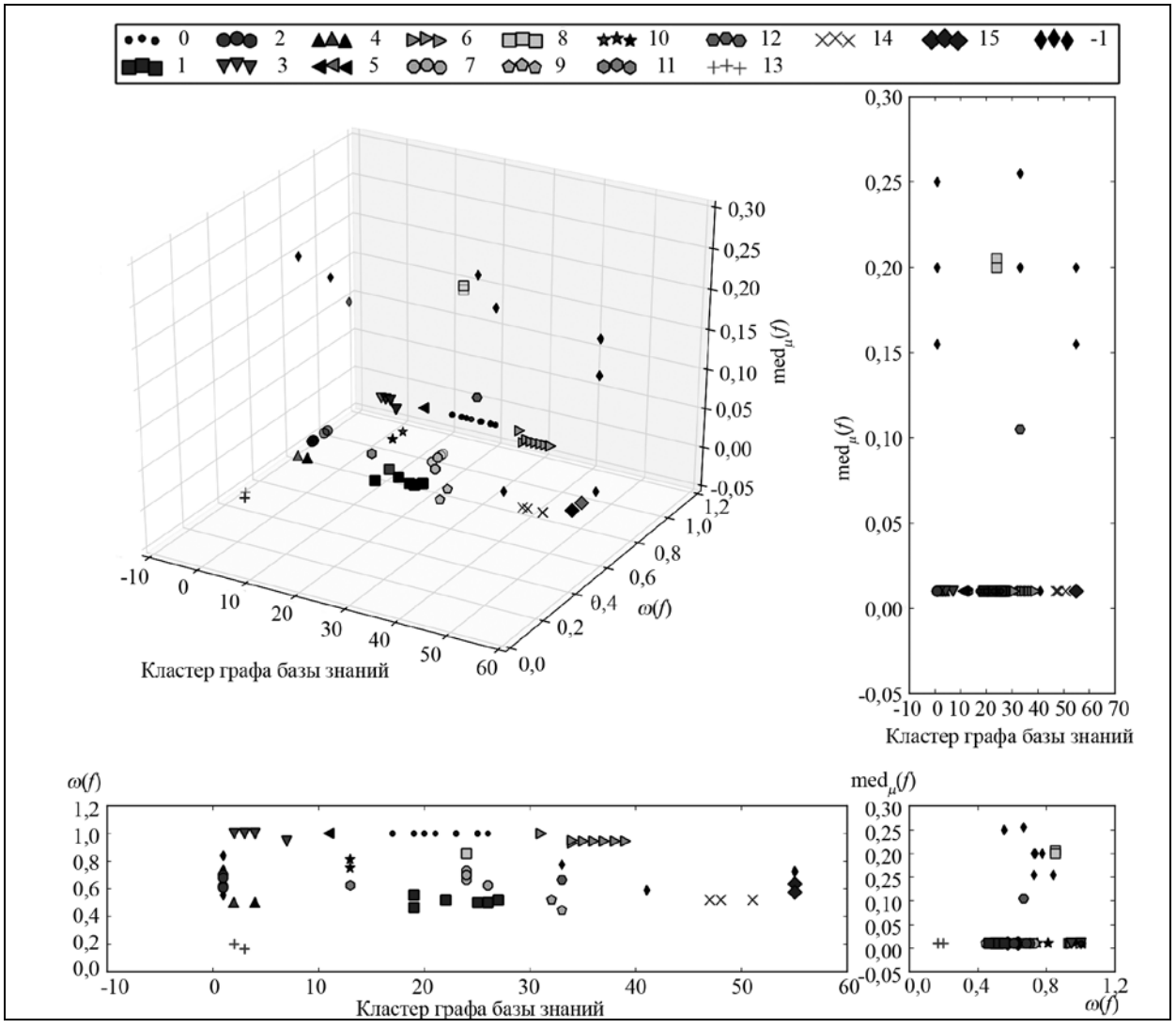

Рис. 3. Пример кластеризации тестовой базы знаний

Для выполнения стохастического поиска используется ряд специальных функций:

$\text{rand}(X) = x \in X$ – генератор случайного элемента x множества $X = \{x_1, \ldots, x_m\}$ по равномерному распределению вероятностей. В Python, данный генератор реализован с помощью функции randrange модуля random стандартной библиотеки.

$\text{rand}(X, P) = x \in X$ – генератор случайного элемента x множества $X = \{x_1, \ldots, x_m\}$ по распределению вероятностей $P = \{p_1, \ldots, p_m\}$. В Python, данный генератор реализован с помощью функции choice модуля random библиотеки NumPy [20];

minPreImages(v) – функция, выполняющая поиск прообразов на множестве слоев $S_{all}(A, v)$. Спецификация ее работы аналогична одноименной функции в методе LP-вывода [4].

Далее приводится псевдокод алгоритма, выполняющего стохастический поиск прообразов:

input: C_t, C_i, C_p, C_W

output: *preimages*

$i \leftarrow 0,\ t \leftarrow 0,\ preimages \leftarrow \varnothing$

while $t < C_t \wedge i < C_i \wedge \left|preimages\right| < C_p$ **do**

  $v \leftarrow \varnothing$

  while $v = \varnothing \vee \left|S_{all}(A, v)\right| > C_W$ **do**

    $j \leftarrow \text{rand}\left(\{1, \ldots, n\}, \{p_1, \ldots, p_n\}\right)$

  $\gamma \leftarrow \left\{f_{j,1}, \ldots, f_{j,|v_j|}\right\}$

    while $|\gamma| > 0 \vee \left|S_{all}(A, v)\right| > C_W$ **do**

      $f \leftarrow rand(\gamma)$

      $\gamma \leftarrow \gamma \setminus \{f\}$

      $v \leftarrow v \cup \{f\}$

    end

  end

  $preimages \leftarrow preimages \cup \text{minPreImages}(v)$

  $t \leftarrow \Delta t$

  $i \leftarrow i + 1$

end.

- Рассмотрена математическая формулировка задачи нечеткого LP-вывода. Представлена реализация кластеризации и стохастического поиска прообразов в виртуальном расслоении бинарного отношения. Полученные результаты могут быть применены для ускорения обратного вывода в интеллектуальных системах продукционного типа и распространены и на более сложные логические системы в информатике [21].

Литература

1. *Джарратано Д., Райли Г.* Экспертные системы: принципы разработки и программирование: Пер. с англ. М.: Вильямс. 2007. 1152 с.
2. *Махортов С.Д., Шмарин А.Н.* Оптимизация метода LP-вывода // Нейрокомпьютеры. Разработка, применение. 2013. № 9. С. 59–63.
3. *Махортов С.Д.* Основанный на решетках подход к исследованию и оптимизации множества правил условной системы переписывания термов // Интеллектуальные системы. Теория и приложения. 2009. Т. 13. С. 51–68.
4. *Махортов С.Д.* Интегрированная среда логического программирования LPExpert // Информационные технологии. 2009. № 12. С. 65–66.
5. *Болотова С.Ю., Махортов С.Д.* Алгоритмы релевантного обратного вывода, основанные на решении продукционно-логических уравнений // Искусственный интеллект и принятие решений. 2011. № 2. С. 40–50.
6. *Шмарин А.Н., Махортов С.Д.* О приближенных оценках количества слоев без циклов в задаче нечеткого LP-вывода // Нейрокомпьютеры. Разработка, применение. 2015. № 12. С. 44–51.

7. *Салий В.Н., Богомолов В.А.* Алгебраические основы теории дискретных систем. М.: Физматлит. 1997. 368 с.
8. *Пегат А.* Нечеткое моделирование и управление. М.: Бином. Лаборатория знаний. 2013. 798 с.
9. *Махортов С.Д.* LP-структуры для обоснования и автоматизации рефакторинга в объектно-ориентированном программировании // Программная инженерия. 2010. № 2. С. 15–21.
10. *Шмарин А.Н.* О реализации приближения числа слоев без циклов в задаче нечеткого LP-вывода // Программная инженерия. 2016. № 7. С. 330–336.
11. Свидетельство о государственной регистрации программы для ЭВМ № 2017613295 / *А.Н. Шмарин.* Fuzzy LPExpert; № 2017610737; заявл. 24.01.2017; зарегистр. 14.03.2017, Бюл. № 3.
12. *Peixoto T.P.* Nonparametric Bayesian inference of the microcanonical stochastic block model // Phys. Rev., 2017. № E 95. P. 24. doi:10.1103/PhysRevE.95.012317
13. *Оре О.* Теория графов: Пер. с англ. М.: ЛИБРОКОМ. 2009. 354 с.
14. *Schaeffer S.E.* Graph clustering // Computer Science Review. 2007. V. I. P. 27–64.
15. *Ермаков С.М.* Метод Монте-Карло в вычислительной математике. СПб.: Невский Диалект. 2009. 192 с.
16. *Kanunfo T., Mount D.M., Netanyahu N.S.* An Efficient k-Means Clustering Algorithm: Analysis and Implementation // IEEE Transactions on pattern analysis and machine intelligence, 2002. V. 24. № 7. P. 881–892.
17. *Флах П.* Машинное обучение: Наука и искусство построения алгоритмов, которые извлекают знания из данных: Пер. с англ. М.: ДМК. 2015. 400 с.
18. *Abraham A., Pedregosa F., Eickenberg M.* Machine Learning for Neuroimaging with Scikit-Learn. // Frontiers in Neuroinformatics. 2014. № 8. P. 11. doi:10.3389/fninf.2014.00014
19. *Patwary M.M., Palsetia D.A.,* New Scalable Parallel DBSCAN Algorithm Using the Disjoint-Set Data Structure // IEEE. 2012. V. 12. P. 10–16.
20. *Bressert E.,* SciPy and NumPy. USA Sebastopol: O'Reilly. 2013. 57 pp.
21. *Чечкин А.В.* Нейрокомпьютерная парадигма информатики // Нейрокомпьютеры: разработка, применение. 2011. № 7. С. 3–9.

Поступила 12 апреля 2017 г.

Stochastic search of preimages based on the clustering of knowledge in the fuzzy LP-inference

A. N. Shmarin – *Post-graduate Student, Voronezh State University*
E-mail: tim-shr@mail.ru

The increase in the volumes and complexity of processed information makes the use of artificial intelligence systems more and more actual. In such systems, one of the most widespread knowledge representation models is the production model and search algorithms are based on the inference engine. In practice, the values measurement of the features of the classified objects from some data domain is time-consuming operation for many problems. These tasks appear, for example, when the robot explores the surface of another planet. If the goal of the robot is to get to some rock, but the future route is observed only partially, then the robot should try to make the best decision on how to achieve the goal, taking into account limitation of the resources available to additional exploration of terrain. Another example, in which the cost of obtaining new information may be high, is the commercial medicine. To minimize costs, it is necessary to find the acceptable treatment method using the minimum number of analyses performed in minimal time. The delay in providing treatment, caused by the additional analyses, leads to increased costs. Moreover, the patient's condition, which was not provided timely assistance, can significantly deteriorate. Also, the cost of additional research is essential in the field of mineral exploration. It can turn out that more cost-effective solution is to begin drilling, if confidence of the success is 95%, than to spend the considerable resources to achieve the 98% confidence.

The complexity of the operations of data acquisition necessary for decision-making, leads to the problem of minimizing the number of requests for information about values of the features of the classified object during inference. This problem is NP-hard. To achieve global minimization there is the general method of LP-inference with exponential computational complexity relative to the number of atomic facts in the knowledge base. However, some of its heuristic modifications have polynomial complexity. The goal of the provided research is development of the approximating method to better minimize the number of feature values requests performed in the inference.

Earlier, the questions of implementation of computing approximate estimates of the number of layers without cycles in the fuzzy LP-inference task were considered. This paper presents the mathematical statement of the fuzzy LP-inference task, researches the fuzzy knowledge clustering method, and presents the implementation of the stochastic initial preimages search. This algorithm is designed for the iterative analysis of such subsets of productions, which most significantly influence to the relevancy. Presented approach can be applied to accelerate the reverse inference in production type systems of artificial intelligence.

REFERENCES

1. Dzharratano D., Rajli G. E'kspertny'e sistemy': princzipy' razrabotki i programmirovanie: Per. s angl. M.: Vil'yams. 2007. 1152 s.
2. Maxortov S.D., Shmarin A.N. Optimizacziya metoda LP-vy'voda // Nejrokomp'yutery'. Razrabotka, primenenie. 2013. № 9. S. 59–63.

3. Maxortov S.D. Osnovanny'j na reshetkax podxod k issledovaniyu i optimizaczii mnozhestva pravil uslovnoj sistemy' perepisy'vaniya termov // Intellektual'ny'e sistemy'. Teoriya i prilozheniya. 2009. T. 13. S. 51–68.
4. Maxortov S.D. Integrirovannaya sreda logicheskogo programmirovaniya LPExpert // Informaczionny'e texnologii. 2009. № 12. C. 65–66.
5. Bolotova S.Ju., Maxortov S.D. Algoritmy' relevantnogo obratnogo vy'voda, osnovanny'e na reshenii produkczionno-logicheskix uravnenij // Iskusstvenny'j intellekt i prinyatie reshenij. 2011. № 2. S. 40–50.
6. Shmarin A.N., Maxortov S.D. O priblizhenny'x oczenkax kolichestva sloev bez cziklov v zadache nechetkogo LP-vy'voda // Nejrokomp'yutery'. Razrabotka, primenenie. 2015. № 12. S. 44–51.
7. Salij V.N., Bogomolov V.A. Algebraicheskie osnovy' teorii diskretny'x sistem. M.: Fizmatlit. 1997. 368 s.
8. Pegat A. Nechetkoe modelirovanie i upravlenie. M.: Binom. Laboratoriya znanij. 2013. 798 s.
9. Maxortov S.D. LP-struktury' dlya obosnovaniya i avtomatizaczii refaktoringa v ob''ektno-orientirovannom programmirovanii // Programmnaya inzheneriya. 2010. № 2. S. 15–21.
10. Shmarin A.N. O realizaczii priblizheniya chisla sloev bez cziklov v zadache nechetkogo LP-vy'voda // Programmnaya inzheneriya. 2016. № 7. S. 330–336.
11. Svidetel'stvo o gosudarstvennoj registraczii programmy' dlya E'VM № 2017613295 / A.N. Shmarin. Fuzzy LPExpert; № 2017610737; zayavl. 24.01.2017; zaregistr. 14.03.2017, Byul. № 3.
12. Peixoto T.P. Nonparametric Bayesian inference of the microcanonical stochastic block model // Phys. Rev., 2017. № E 95. P. 24. doi:10.1103/PhysRevE.95.012317
13. Ore O. Teoriya grafov: Per. s angl. M.: LIBROKOM. 2009. 354 s.
14. Schaeffer S.E. Graph clustering // Computer Science Review. 2007. V. I. P. 27–64.
15. Ermakov S.M. Metod Monte-Karlo v vy'chislitel'noj matematike. SPb.: Nevskij Dialekt. 2009. 192 s.
16. Kanunfo T., Mount D.M., Netanyahu N.S. An Efficient k-Means Clustering Algorithm: Analysis and Implementation // IEEE Transactions on pattern analysis and machine intelligence, 2002. V. 24. № 7. P. 881–892.
17. Flax P. Mashinnoe obuchenie: Nauka i iskusstvo postroeniya algoritmov, kotory'e izvlekayut znaniya iz danny'x: Per. s angl. M.: DMK. 2015. 400 s.
18. Abraham A., Pedregosa F., Eickenberg M. Machine Learning for Neuroimaging with Scikit-Learn. // Frontiers in Neuroinformatics. 2014. № 8. P. 11. doi:10.3389/fninf.2014.00014
19. Patwary M.M., Palsetia D.A., New Scalable Parallel DBSCAN Algorithm Using the Disjoint-Set Data Structure // IEEE. 2012. V. 12. P. 10–16.
20. Bressert E., SciPy and NumPy. USA Sebastopol: O'Reilly. 2013. 57 pp.
21. Chechkin A.V. Nejrokomp'yuternaya paradigma informatiki // Nejrokomp'yutery': razrabotka, primenenie. 2011. № 7. S. 3–9.