УДК 004.8

АЛГОРИТМ РАВНОВЕРОЯТНОГО СИНТЕЗА МИНИМАЛЬНОГО ГРАФА СМЕЖНОСТИ[1]

Фильченков А.А., Столярова В.Ф., Тулупьев А.Л.
СПИИРАН, СПбГУ, г. Санкт-Петербург

Поступила в редакцию 07.05.2013, после переработки 18.02.2014.

Предложен алгоритм равновероятного синтеза минимального графа смежности, который по заданной первичной структуре алгебраической байесовской сети, представленной набором максимальных по включению подалфавитов некоторого алфавита, строит случайным образом минимальный по числу ребер граф смежности, причем все возможные реализации минимальных графов смежности генерирует с равной вероятностью. Оценена сложность предложенного алгоритма.

Ключевые слова: алгебраические байесовские сети, вторичная структура, минимальный граф смежности, машинное обучение, структурный синтез, случайные графы.

Вестник ТвГУ. Серия: Прикладная математика. 2014. №1. С. 73–89.

Введение

Алгебраические байесовские сети (АБС) относятся к классу логико-вероятностных графических моделей [22, 35] и используются для представления сложных систем знаний с неопределенностью. Необходимость компьютерной обработки знаний с неопределенностью возникает, в частности, при обращении к областям знаний, которые в значительной степени опираются на экспертную информацию в силу фундаментальной сложности процессов, встречающихся в таких областях знаний. Примером могут служить области медицинской диагностики и эпидемиологии, где при принятии решений лечащий врач сталкивается с неполнотой и неточностью информации о симптомах и результатах медицинских исследований, с неопределенностью протекания физиологических процессов индивида [3, 9, 32, 36, 39]. Другой областью знаний, где возможно использование вероятностных графических моделей, является область информационной безопасности [1, 2, 20]. В настоящее время экспертные знания интегрируются так же в модели, которые традиционно основаны на теоретико–вероятностном и статистическом анализе процессов, например, в области анализа финансовых рисков. Экспертная информация может быть представлена в терминах субъективных вероятностей [38] или же в рамках иных формализмов, таких как нечеткие случайные величины [40, 41].

[1]Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований, проекты №12-01-00945-а и №12-01-31202-мол_а.

Узлами алгебраической байесовской сети [10,13,15,16] выступают сложные случайные элементы [28] (и даже семейства таких случайных элементов, единых по структуре, но различающихся по значениям параметров распределений) — математические модели фрагментов знаний (фрагменты знаний) [6, 10, 15, 16]. Набор максимальных по включению фрагментов знаний (МФЗ) формирует *первичную структуру* алгебраической байесовской сети, а граф, построенный над ними — ее *вторичную структуру* [11, 12, 18].

Проблема обучения вторичной структуры АБС по ее первичной структуре сводится к задаче построения минимального графа смежности (МГС) по набору подалфавитов, над которыми построены фрагменты знаний [18, 19, 25]. Для исследования как свойств минимальных графов смежности, так и численных характеристик осуществления алгоритмов глобального логико-вероятностного вывода [7, 14, 17] можно было бы изучать все минимальные графы смежности, однако это неэффективно, поскольку множество минимальных графов смежности велико [26, 27] и алгоритм синтеза этого множества затратен [30]. В работе [24] было предложено исследовать случайные выборки из этого множества. В ней же была поставлена задача построения минимального графа смежности случайным образом и предложен алгоритм рандомизированного синтеза такого графа, причем каждый минимальный граф смежности может быть синтезирован таким алгоритмом с ненулевой вероятностью. Однако просто рандомизированного синтеза недостаточно: необходимо управлять распределением, которое задается работой алгоритма.

Интерес представляет, в первую очередь, тот случай, когда каждый минимальный граф смежности может быть построен с одинаковой вероятностью. Алгоритм, работа которого обеспечивает реализацию равномерного распределения, позволит оценивать частоту встречаемости тех или иных свойств у минимальных графов смежности, а также позволит привлечь генетические алгоритмы для синтеза минимального графа смежности, который минимизирует время работы алгоритмов пропагации или же уменьшает потерю точности оценок при пропагации в случае интервальных оценок вероятности.

Цель статьи — предложить алгоритм, который по заданному набору подалфавитов, соответствующему первичной структуре АБС, строит случайным образом минимальный граф смежности, причем все возможные реализации минимальных графов смежности строятся равновероятно.

Работа развивает идеи доклада, представленного на VII Международной научно-практической конференции «Интегрированные модели и мягкие вычисления в искусственном интеллекте» (20–22 мая 2013 г., Коломна) [4].

1. Графы и случайные графы.

Граф — пара $\langle V, E\rangle$, где V — множество вершин графа, а E — множество ребер. Через V и E будем обозначать функции от графа, возвращающие множество его вершин и множество его ребер соответственно.

Под случайным графом [33, 34] будем понимать случайный элемент [31], заданный на некотором множестве графов с фиксированным числом вершин, алгеброй событий на множестве графов выступает множество всех подмножеств множества графов с фиксированным числом вершин [5].

Будем опираться на алгоритм Прюфера [37] для построения алгоритма равновероятного синтеза минимального графа смежности. Рассмотрим ненаправленный граф из n вершин, которым произвольным образом присвоены номера с 0 по $n-1$.

Биркой размера n будем называть $(n-2)$-местный кортеж из чисел от 0 до $n-1$. *Кодом Прюфера* для дерева из n пронумерованных вершин называется бирка размера n, которая кодирует соответствующее дерево.

Код Прюфера строится последовательным выполнением $n-2$ этапов, каждый из которых состоит из следующих шагов:

1. Находится висячая вершина с наименьшим номером.
2. Номер вершины, с которой найденная соединена единственным ребром, записывается в самую левую свободную ячейку бирки.
3. Найденная вершина удаляется из дерева.

Алгоритм Прюфера строит по коду Прюфера дерево. Он состоит из $n-2$ этапов. Каждый этап состоит из следующих шагов:

1. Рассматривается крайнее слева число кода Прюфера.
2. Вершина с рассматриваемым номером соединяется с вершиной наименьшего номера, которая не встречается ни в коде Прюфера, ни во множестве уже рассмотренных вершин L_T.
3. Упомянутая вершина с наименьшим номером добавляется к множеству L_T.
4. Из кода Прюфера удаляется крайнее слева число.

После завершения этих $n-2$ этапов проводится ребро между двумя вершинами, не содержащимися в L_T.

Замечание 1. Алгоритм работает за линейное время от числа вершин (длины кода).

Через $T(r)$ будем обозначать функцию, восстаниваливающую дерево по коду Прюфера r. Через $d_G(v)$ будем обозначать степень вершины v в графе G. Через $\alpha_r(i)$ будем обозначать число вхождений числа i в код Прюфера r.

Утверждение 1. *[27].*

$$\alpha_r(i) + 1 = d_{T(r)}(v_i), \tag{1}$$

где v_i — вершина дерева, занумерованная индексом i.

2. Элементы глобальной структуры алгебраической байесовской сети

Будем следовать терминологии, принятой в работах [21, 25, 28–30].

Для краткости изложения зафиксируем первичную структуру АБС, и все вводимые далее объекты будем определять для нее, явно ее не указывая. *Алфавит* — множество атомарных пропозициональных формул: $A = \{x_1, \ldots, x_n\}$.

Нагруженный граф — тройка $\langle G, A, W\rangle$, где G — граф, A — алфавит, а W — функция нагрузки, заданная на вершинах и ребрах G и принимающая значения из множества подалфавитов алфавита A :

$$W : V(G) \cup E(G) \longrightarrow 2^A.$$

Сепаратором двух вершин называется пересечение нагрузок соответствующих вершин: $\mathrm{Sep}\,(v, u) = W(v) \cap W(u)$. Согласованным нагруженным графом называется нагруженный граф, в котором нагрузка ребра между вершинами равна их сепаратору: $\forall\{u, v\} \in G \quad W(\{u, v\}) = \mathrm{Sep}\,(u, v)$.

В введенном формализме A соответствует алфавиту, над которым задается первичная структура АБС, а нагрузка вершин соответствует подалфавитам, над которыми построены фрагменты знаний. При фиксированной первичной структуре для согласованного нагруженного графа A и W также фиксированы, поэтому для краткости изложения не будем явно упоминать их, ассоциируя согласованный нагруженный граф с первыми элементом тройки из определения. Множество непустых сепараторов (для заданной первичной структуры) будем обозначать как Sep.

Две вершины называются *сочлененными*, если их сепаратор непуст. *Граф максимальных фрагментов знаний* (граф МФЗ) — согласованный нагруженный граф, в котором ребра возможны только между сочлененными вершинами (т. е. нагрузка любого ребра непуста). *Магистральный путь* между двумя вершинами u и v в согласованном нагруженном графе — это такой путь $(u = w_0, w_1, \ldots, w_{n-1}, w_n = v)$, что нагрузка любой вершины содержит сепаратор u и v : $\forall w_i\ \mathrm{Sep}(u, v) \subset W(w_i)$. Согласованный нагруженный граф *магистрально связен*, если между каждой парой сочлененных вершин существует магистральный путь.

Граф смежности — магистрально связный граф МФЗ. В графе смежности возможны ребра только между сочлененными вершинами и между любой парой сочлененных вершин существует магистральный путь. *Минимальный граф смежности* (МГС) — граф смежности, число ребер которого минимально.

Сильное сужение $\downarrow U$ на сепаратор U — это граф, в который входят все вершины первичной структуры, нагрузка которых содержит U, и ребра проведены между каждой парой вершин, сепаратор которых строго содержит U (Рис. 1). Сильное сужение $\downarrow U$ разбивается на компоненты связности $\{c_1, c_2, \ldots, c_m\}$, которые называются *владениями*.

Оммажем H сепаратора U, называется дерево, множество вершин которого есть множество владений сильного сужения сепаратора U (Рис. 2). Так как над набором компонент связности можно построить множество деревьев, то сепаратору соответствует множество оммажей.

В свою очередь, каждому оммажу соответствует множество *жил*. Под *жилой* S оммажа H, понимается граф, множество вершин которого есть $V(S) = \{v_S^1, v_S^2, \ldots, v_S^m\} \subseteq V(G)$, причем $v_S^i \in c_i$, и две вершины которого v_S^i и v_S^j соединены ребром, если соответствующие владения c_i и c_j соединены ребром в оммаже H (Рис. 3). Здесь $c_1, c_2, \ldots, c_m$ есть владения сильного сужения $\downarrow U$. Таким образом, каждый оммаж задает структуру жилы, а каждая жила является реализацией оммажа. Каждая жила строится путем добавления (особым образом, соответствующим оммажу) ребер к сильному сужению $\downarrow U$, таким образом, согласно определению сильного сужения, ребра жилы имеют нагрузку U.

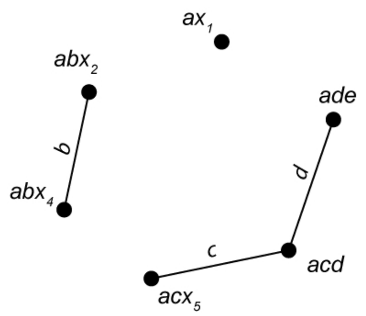

Рис. 1: Сильное сужение нагрузки a

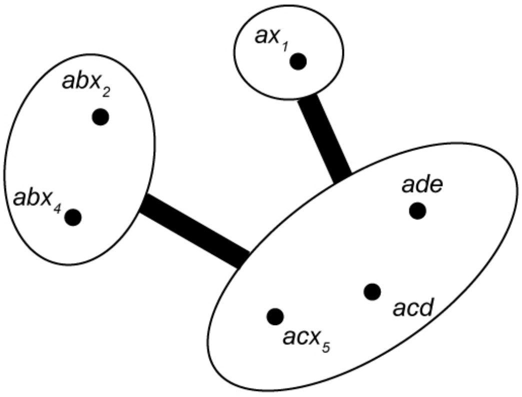

Рис. 2: Пример оммажа над владениями из Рис. 1

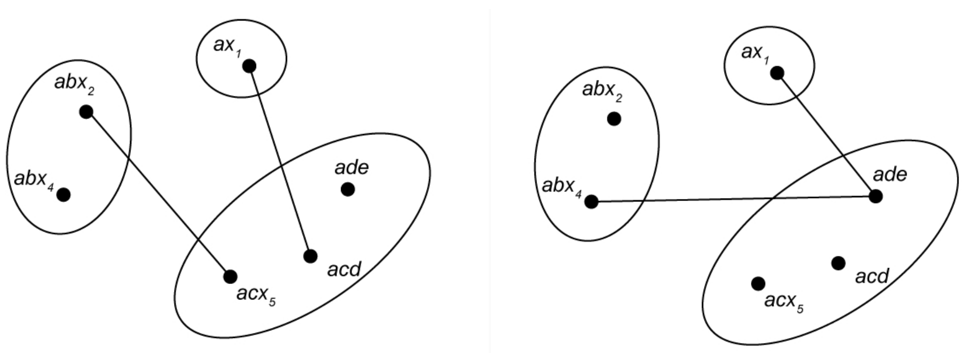

Рис. 3: Примеры жил, реализующих оммаж из Рис. 2

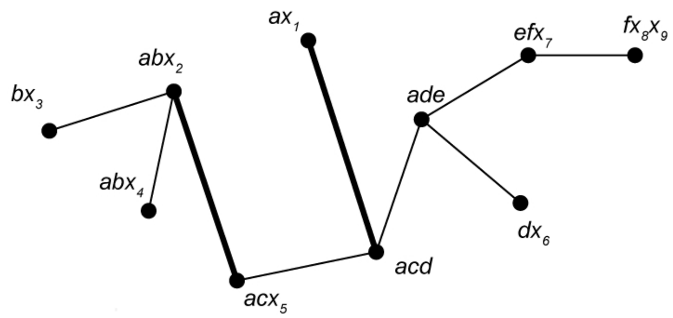

Рис. 4: Примеры пучка с жилой из правой части Рис. 3

Рассмотрим кортеж жил, где для каждого сепаратора встречается ровно одна жила с такой нагрузкой. Объединение элементов такого кортежа называется *пучком* (Рис. 4). Известна следующая теорема:

Теорема 1. *(о множестве МГС) [25, 28]. Множество минимальных графов смежности совпадает с множеством пучков.*

Теорема о множестве МГС используется для построения как множества МГС, так и определенных подмножеств этого множества [23,30]. В первом случае следует строить все возможные жилы каждого сепаратора, во втором случае — обладающие искомыми свойствами.

Пусть c — владение. Тогда $v(c)$, — число вершин, содержащихся во владении c, — будем называть *объемом владения*. Пусть H — дерево, вершинами которого являются владения. Тогда через $d_H(c)$ обозначим степень вершины в дереве H, соответствующей владению c.

Утверждение 2. *[27]. Каждому оммажу H соответствует*

$$\prod_{c \in V(H)} (v(c))^{d_H(c)}$$

жил.

Следствием из указанного утверждения является формула для мощности множества жил.

Следствие 1. *Мощность множества жил для каждого непустого сепаратора есть*

$$\left(\prod_{i=1}^{n} v(c_i)\right) \left(\sum_{i=1}^{n} v(c_i)\right)^{n-2},$$

где n — число компонент связности сильного сужения соответствующего непустого сепаратора.

3. Схема алгоритма равновероятного синтеза минимального графа смежности

Поскольку алгоритм синтеза множества минимальных графов смежности предполагает построение всех элементов этого множества, то сложность работы такого алгоритма снизу ограничена числом элементов этого множества, которое выражается как произведение мощностей множеств жил для каждого непустого сепаратора [27], которое, как видно из следствия 1, экспоненциальны (от числа владений сепаратора). Последняя величина в худшем случае может совпадать с числом фрагментов знаний (хотя в среднем значительно меньше). Таким образом, полный перебор всего множества оказывается достаточно неэффективным, что создает предпосылки для возникновения задачи формирования выборок минимальных графов смежности случайным образом.

Алгоритм равновероятного синтеза минимального графа смежности строит по первичной структуре АБС (множеству нагрузок) случайный минимальный граф смежности, причем каждый элемент множества МГС выбирается из множества МГС равновероятно. Он состоит из следующих шагов:

1. Построить множество $\mathrm{Sep} = \{U_1, U_2, \ldots, U_n\}$ всех непустых сепараторов первичной структуры АБС.

2. Для каждого $U_a \in \mathrm{Sep}$ построить сильное сужение $\downarrow\!\!\!\downarrow U_a$.

3. Разложить сужения $\downarrow\!\!\!\downarrow U_a$ на компоненты связности (владения) $\{c_1^a, c_2^a, \ldots, c_{h_a}^a\}$. Пронумеровать компоненты связности числами $\{1, 2, 3, \ldots, h_a\}$.

4. Случайным образом сгенерировать код Прюфера r размера h_a: $r = (p_1^a, p_2^a, \ldots, p_{h_a-2}^a)$, определяющий однозначно дерево $H_a^{b_a}$ на элементах $\{c_1^a, c_2^a, \ldots, c_{h_a}^a\}$, причем сам код Прюфера генерируется с вероятностью
$$P_{H,a}(r) := \frac{\prod_{i=1}^{h_a} (v(c_i^a))^{\alpha_r(i)+1}}{\left(\prod_{i=1}^{h_a} v(c_i^a)\right) \left(\sum_{i=1}^{h_a} v(c_i^a)\right)^{h_a-2}}.$$

5. Для каждого ребра дерева $H_a^{b_a}$, соединяющего владения c_i^a и c_j^a, построить ребро между вершиной $g \in c_i^a$ и вершиной $g' \in c_j^a$, причем в каждом владении c_k^a вершина выбирается равновероятно с вероятностью $\frac{1}{v(c_k^a)}$. Набор ребер и их концы образуют жилу $S_a^{l_a}$;

6. Объединить жилы $S_1^{l_1}, S_2^{l_2}, \ldots, S_n^{l_n}$.

Граф, полученный на последнем шаге работы алгоритма, является минимальным графом смежности по теореме о множестве минимальных графов смежности. Предложенный алгоритм генерирует минимальный граф смежности, согласно утверждению в работе [24]. Проверим, что любой граф из множества минимальных графов смежности может быть получен равновероятно при помощи предложенного алгоритма.

Теорема 2. *Каждый элемент множества минимальных графов смежности синтезируется равновероятно в результате работы алгоритма.*

Доказательство. Пусть G является элементом множества минимальных графов смежности. Результатом алгоритма является реализация случайного минимального графа смежности $\mathbf{G}$.

В дальнейшем верхним индексом $*$ будем обозначать конкретную реализацию случайного элемента.

Покажем, что конкретный граф G выбирается из множества всех минимальных графов смежности равновероятно в результате работы алгоритма. Согласно теореме 1, каждый минимальный граф смежности представляет собой пучок, т. е. объединение жил, выбранных по одной для каждого сепаратора. Обозначим через $\mathbf{S}_a$ случайный элемент, отвечающий выбору жилы для сильного сужения $\downarrow U_a$ при реализации алгоритма. Через P будем обозначать вероятность построения элемента в результате применения описанного алгоритма.

$$P(\mathbf{G}=G)=P\left(\bigcup_{a=1}^{n}\mathbf{S}_a=\bigcup_{a=1}^{n}S_a^*\right)=\prod_{a=1}^{n}P\left(\mathbf{S}_a=S_a^*\right).$$

Согласно предложенной схеме, жилы генерируются независимо. Таким образом, необходимо проверить, что каждая жила $S_a^*, a\in 1,\ldots,n$ выбирается из множества всех жил сильного сужения $\downarrow U_a$ равновероятно.

Выбор конкретной жилы зависит от построения конкретного оммажа. Обозначим через $\mathbf{H}_a$ случайный элемент, отвечающий выбору оммажа для сепаратора U_a при реализации алгоритма. Таким образом,

$$P\left(\mathbf{S}_a=S_a^*\right)=P\left(\mathbf{S}_a=S_a^*|\mathbf{H}_a=H_a^*\right)\cdot P\left(\mathbf{H}_a=H_a^*\right).$$

Вероятность получить конкретный оммаж при реализации алгоритма равна вероятности выбрать конкретный код Прюфера, т.е. $P\left(\mathbf{H}_a=H_a^*\right)=P_a$. При построении конкретной жилы мы обращаемся к владению c_i^a и выбираем из него вершину с вероятностью $\frac{1}{v(c_i^a)}$ столько раз, какова степень вершины в дереве H_a^*, таким образом вероятность построить конкретную жилу при условии уже выбранного дерева есть

$$P\left(\mathbf{S}_a=S_a^*|\mathbf{H}_a=H_a^*\right)=\frac{1}{\prod_{i=1}^{h_a}\left(v\left(c_i^a\right)\right)^{d_{H_a^*}(i)}}.$$

На шаге 4 схемы алгоритма выписан вид вероятности P_a :

$$P\left(\mathbf{S}_a=S_a^*|\mathbf{H}_a=H_a^*\right)\cdot P\left(\mathbf{H}_a=H_a^*\right)=$$
$$=\frac{\prod_{i=1}^{h_a}\left(v(c_i^a)\right)^{\alpha_{r^*}(i)+1}}{\left(\prod_{i=1}^{h_a}v(c_i^a)\right)\left(\sum_{i=1}^{h_a}v(c_i^a)\right)^{h_a-2}}\cdot\frac{1}{\prod_{i=1}^{h_a}\left(v\left(c_i^a\right)\right)^{d_{H_a^*}(i)}},$$

где r^* — код Прюфера для H_a^*.

Согласно свойствам кода Прюфера (1), получим

$$P\left(\mathbf{S_a}=S_a^*\right)=\frac{1}{\left(\prod_{i=1}^{h_a}v\left(c_i^a\right)\right)\left(\sum_{i=1}^{h_a}v\left(c_i^a\right)\right)^{h_a-2}}.$$

Таким образом, введенный алгоритм генерирует каждую жилу равновероятно из всего множества жил для сильного сужения $\downarrow U_a$. Доказательство окончено.

Для удобства дальнейших рассуждений обозначим $P\left(\mathbf{S_a} = S_a^*\right)$ как $P_{S,a}$.

4. Построение кода Прюфера с заданной вероятностью

Отдельно рассмотрим вопрос о том, как строить код Прюфера r с указанной в алгоритме вероятностью

$$P_{H,a}(r) = \frac{\prod_{i=1}^{h_a} \left(v(c_i^a)\right)^{\alpha_r(i)+1}}{\left(\prod_{i=1}^{h_a} v(c_i^a)\right)\left(\sum_{i=1}^{h_a} v(c_i^a)\right)^{h_a-2}}.$$

Несмотря на достаточно «увесистую» формулу, построение кода Прюфера с такой вероятностью технически сравнительно несложно. Чтобы сгенерировать код Прюфера, будем выбирать числа из набора $\{1, 2, \ldots, h_a\}$ на каждую из $h_a - 2$ позиций в коде.

Утверждение 3. *Если каждое число из набора $\{1, 2, \ldots, h_a\}$ может быть выбрано в код Прюфера с вероятностью $\frac{v(c_i^a)}{\sum_{j=1}^{h_a} v(c_j^a)}$ и выбор числа на каждую позицию в коде Прюфера производится независимо, то вероятность получить конкретный код r Прюфера есть $P_{H,a}(r)$.*

Доказательство. Пусть выбор числа на каждую позицию в коде Прюфера не зависим от выборов чисел на другие позиции. Тогда сам код Прюфера представляет собой $h_a - 2$ реализации случайного элемента $\mathbf{pruf}_a$, отвечающего выбору числа из набора $\{1, 2, \ldots, h_a\}$. Пусть каждое число i из упомянутого набора может быть выбрано в код Прюфера с вероятностью, пропорциональной объему соответствующего владения c_i^a:

$$P(\mathbf{pruf}_a = i) = \frac{v(c_i^a)}{\sum_{j=1}^{h_a} v(c_j^a)}.$$

Тогда вероятность сгенерировать конкретный код Прюфера r, в котором каждый индекс k встречается $\alpha_r(k)$ раз, есть

$$\frac{\prod_{i=1}^{h_a} (v(c_i^a))^{\alpha_r(i)}}{\left(\sum_{j=1}^{h_a} v(c_j^a)\right)^{h_a-2}} = P_{H,a}(r).$$

Доказательство окончено.

5. Система алгоритмов синтеза подмножеств минимальных графов смежности

В работе [30] была предложена двухэтапная система алгоритмов синтеза подмножеств минимальных графов смежности (подмножеством может выступать как множество минимальных графов смежности, так и одноэлементное множество —

т.е. один граф). На первом этапе такой системы строится ModelKit, состоящее из четырех множеств: множества нагрузок, множество стереосепараторов, множество владений стереосепараторов и множество обязательных ребер. *Стереосепаратором* называется сепаратор, у которого более одного владения. *Обязательным ребром* называет ребро, которое входит в каждый минимальный граф смежности.

В описании алгоритмов будем обозначать перечисленные множества соответственно:

$$\mathrm{ModelKit} = \langle \mathrm{Workloads}, \mathrm{Stereoseparators}, \mathrm{Stereoholding}, \mathrm{NecessaryEdge} \rangle.$$

На втором этапе по построенному ModelKit синтезируется желаемое подмножество минимальных графов смежности алгоритмом MJGAssembling за счет построения всевозможных жил заданного типа для каждого стереосепаратора с дальнейшим перебором кортежей всех таких жил по одной для каждого стереосепаратора и объединением элементов таких кортежей между собой и с множеством обязательных ребер. Построенное таким образом множество графов и будет искомым подмножеством [30].

Алгоритм MJGAssembling выглядит следующим образом [30]:

Листинг 1 Алгоритм сборки подмножества МГС MJGAssembling

Require: $\langle \mathrm{Workloads}, \mathrm{Stereoseparators}, \mathrm{Stereoholding}, \mathrm{NecessaryEdge} \rangle, T$
Ensure: $\mathrm{Stereoseparators} \cup \mathrm{NecessaryEdge} \neq \emptyset$

```
for all u ∈ Stereoseparators do
    Sinews [u] ← Sinews(T, Stereoholding[u])
end for
EdgeSets ←UnitingAlgebraicFolding(Sinews)
MJGSubset ← ∅
for all E ∈ EdgeSets do
    MJGSubset ← MJGSubset ∪ {⟨Workloads, UniteDSets(E, NecessaryEdge)⟩}
end for
if EdgeSets = ∅ then
    MJGSubset ← {Workloads, NecessaryEdge}
end if
return MJGSubset
```

UnitingAlgebraicFolding по заданному множеству множеств $\{\{s_{i,j}\}_j\}_i$ выдает множество объединений элементов всевозможных кортежей, в которых из каждого множества $\{s_{i,j}\}_j$ выбирается ровно один элемент.

UniteDSets обозначает операцию объединения двух заведомо непересекающихся множеств.

Sinews строит множество жил заданного типа T.

Фактически, тип синтезируемого подмножества определяется выбором алгоритма синтеза жил рассматриваемого типа по заданному стереосепаратору. Для равновероятного синтеза минимального графа смежности такой алгоритм должен строить случайную жилу с вероятностью $P_{S,a}$.

Утверждение 4. *[30] Сложность работы алгоритма сборки подмножества*

МГС равна

$$\sum_{u \in \mathrm{StSep}} S_u^T + O\left(s \prod_{u \in \mathrm{StSep}} M_u^T\right),$$

где StSep *— множество стереосепараторов,* S_u^T *— сложность построения множества жил указанного типа* T *для сепаратора* u, $s = |\mathrm{StSep}|$ *— число стереосепараторов, а* M_u^T *— мощность множества жил типа* T[2].

Таким образом, необходимо предложить реализацию алгоритма Sinews для типа жил, соответствующего одной реализации случаной жилы с равномерным распределением для заданного сепаратора, который может быть использован в этом алгоритме.

6. Алгоритм равновероятностного синтеза жилы

Перед тем, как описать алгоритм равновероятного синтеза жилы по набору владений, опишем используемые в ней алгоритмы.

Функция ChooseRandomProportionallyCDF(V, B) устроена следующим образом: первым ее элементом является массив V размера n из которого извлекает элемент с вероятностью, задаваемой функцией распределения, пропорциональной той, которая задается вторым аргументом — массивом чисел B размера n. Функция построена на том, чтобы сгенерировать случайное число y в промежутке от 1 до значения последнего элемента массива B, затем поиском в массиве B такого элемента i, что $B[i] \leqslant y \leq B[i+1]$. Значение $V[i]$ и будет является результатом работы этой функции. Поскольку B является упорядоченным, работа функции осуществляется на основе бинарного поиска, и, соответственно, сложность ее работы лежит в классе $O(\ln n)$.

Функция PrüferTree(r) по заданному коду Прюфера r строит соответствующее дерево $T(r)$. Ее описание приведено в разделе 1, сложность лежит в классе $O(n)$.

Функция ChooseRandom(V) выбирает элемент из заданного множества V равновероятно. Функция построена на том, чтобы сгенерировать случайный индекс массива, ее сложность можно считать лежащей в классе $O(1)$.

Утверждение 5. *Сложность работы алгоритма* SinewsEquiprobableOne *равна* $O(h \ln h)$, *где* h *— число владений соответствующего сепаратора.*

Доказательство.

Сложность работы цикла (2–5) равна $O(h)$.

Сложность работы цикла (6–8) равна $O(h \ln h)$, поскольку в нем h раз выполяется функция ChooseRandomProportionally для массивов из h элементов, имеющая сложность $O(\ln h)$.

Сложность работы шага (9) равна $O(h)$.

Сложность работы цикла (10–14) равна $O(h)$.

Следователь, сложность работы самого алгоритма равна $O(h \ln h)$. Доказательство окончено.

[2] Следует отметить, что в каждом случае мощность синтезируемого подмножества известна до начала работы алгоритма, поэтому ее можно отнести к исходным параметрам.

Листинг 2 Алгоритм равновероятного синтеза жилы для заданного сепаратора SinewsEquiprobableOne

Require: Holdings
Ensure: Holdings $= \{C_0, \ldots, C_{h-1}\}$; $h > 0$; $|c_i| > 1$

```
y ← 0
for all i ∈ {0, ..., |Holdings| − 1} do
    y ← y + |Holdings[i]|
    PrBound[i] ← y
end for
for all i ∈ {0, ..., |Holdings| − 1} do
    r[i] ← ChooseRandomProportionallyCDF(Holdings, PrBound)
end for
H ← PrüferTree(r)
for all (U, V) ∈ E(H) do
    g ← ChooseRandom(Holdings[U])
    g′ ← ChooseRandom(Holdings[V])
    Edges ← Edges ∪ {g, g′}
end for
return Edges
```

Утверждение 6. *Сложность работы алгоритма сборки подмножества минимальных графов смежности, состоящего из одной реализации случайного минимального графа смежности с равномерным распределением по* ModelKit *равна*

$$\sum_{u \in \mathrm{StSep}} h_u,$$

где h_u *— число владений стереосепаратора* u.

Доказательство. Это следует непосредственно из утверждений 4 и 5.

Реализация Sinews называется *эффективной*, если сложность ее работы лежит в классе $O(hM)$, где h — число владений соответствующего сепаратора, а M — мощность синтезируемого множества. Поскольку $M = 1$, то реализация не является эффективной. Впрочем, неэффективна реализация этой функции для рандомизированного алгоритма синтеза множества минимальных графов смежнотсти [8]. Сложность ее работы лежит в классе $O(hc)$, где c – максимальное число вершин во владении. К сожалению, c и $\ln h$ несравнимы, однако большое числа небольших владений встречаются чаще, чем небольшое число владений с большим числом вершин.

Заключение

Алгоритм равновероятностного синтеза минимального графа смежности, представленный в статье, позволяет получить реализацию случайного минимального графа смежности, выбранную равновероятно из множества всех минимальных графов смежности. Возможность равновероятного синтеза минимального графа смежности позволит оптимизировать подходы к изучению свойств вторичной структуры алгебраической байесовской сети и логико-вероятностного вывода на ней, а также будет полезна при генерации тестовых выборок.

Необходимо отметить, что реализованная в предложенном алгоритме случайность выбора минимального графа смежности зависит от используемых алгоритмов генерации случайных чисел.

С практической точки зрения, равновероятностного синтеза оказывается достаточно для удовлетворения всех исследовательских целей. Тем не менее, с теоретической точки зрения, продолжением исследования является развитие системы алгоритмов, работа которых задает заранее выбранное вероятностное распределение, что в совокупности с дальнейшим исследованием свойств минимальных графов смежности позволит разработать алгоритмы синтеза минимальных графов смежности, которые строят минимальные графы смежности с определенным свойствами (минимизация времени работы логико-вероятностного вывода и т.д.).

Список литературы

[1] Котенко И.В., Саенко И.Б., Юсупов Р.М. Защита информационных ресурсов в компьютерных сетях // Вестник Российской академии наук. 2011. Т. 81, № 8. С. 746–747.

[2] Котенко И.В., Саенко И.Б., Юсупов Р.М. Аналитический обзор докладов Международной конференции «Математические модели, методы и архитектуры для защиты компьютерных сетей» (MMM-ACNS-2010) // Труды СПИИРАН. 2010. №2 (13). С. 199–225.

[3] Мусина В.Ф. Байесовские сети доверия как вероятностная графическая модель для оценки медицинских рисков // Труды СПИИРАН. 2013. №24. С. 135—151.

[4] Мусина В.Ф., Фильченков А.А. Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов VII-й международной научно-практической конференции (ИММВ-2013, Коломна, 20–22 мая 2013 г.). М.: Физматлит, 2013. Т. 3. С. 1367—1375.

[5] Райгородский А.М. Модели случайных графов и их применение // Труды Московского физико-технического института. 2010. Т. 2, №4. С. 130–140.

[6] Сироткин А.В. Комплекс программ логико-вероятностного вывода в базах фрагментов знаний: реализация фрагмента знаний // Труды СПИИРАН. 2013. №2. С. 201–220.

[7] Сироткин А.В., Мусина В.Ф., Тулупьев А.Л. Алгебраические байесовские сети: нелинейная задача оптимизации в локальном апостериорном выводе при атомарном стохастическом свидетельстве // Труды СПИИРАН. 2012. №20. С. 200–215.

[8] Сироткин А.В., Тулупьев А.Л., Фильченков А.А., Фроленков К.В. Оценки сложности алгоритмов синтеза минимальных по числу ребер графов смежности // СПИСОК-2013: Материалы всероссийской научной конференции по

проблемам информатики (23–26 апреля 2013 г., Санкт-Петербург). СПб.: ВВМ, 2013. С. 776–783.

[9] Суворова А.В., Тулупьева Т.В., Тулупьев А.Л., Сироткин А.В., Пащенко А.Е. Вероятностные графические модели социально-значимого поведения индивида, учитывающие неполноту информации // Труды СПИИРАН. 2012. №22. С. 101–112.

[10] Тулупьев А.Л. Алгебраические байесовские сети. Логико-вероятностный подход к моделированию баз знаний с неопределенностью. СПб.: СПИИРАН, 2000. 292 с.

[11] Тулупьев А.Л. Ациклические алгебраические байесовские сети: логико-вероятностный вывод // Нечеткие системы и мягкие вычисления. 2006. Т. 1, №1. С. 57–93.

[12] Тулупьев А.Л. Байесовские сети: логико-вероятностный вывод в циклах. СПб.: Изд-во С.-Петербургского ун-та, 2008. 140 с. (Элементы мягких вычислений).

[13] Тулупьев А.Л. Согласованность данных и оценка вероятности альтернатив в цикле стохастических предпочтений // Известия высших учебных заведений. Приборостроение. 2009. Т. 52, №7. С. 3–7.

[14] Тулупьев А.Л. Апостериорные оценки вероятностей в алгебраических байесовских сетях // Вестник Санкт-Петербургского университета. Серия 10. Прикладная математика. Информатика. Процессы управления. 2012. №2. С. 51–59.

[15] Тулупьев А.Л., Николенко С.И., Сироткин А.В. Байесовские сети: логико-вероятностный подход. СПб.: Наука, 2006. 607 с.

[16] Тулупьев А.Л., Сироткин А.В., Николенко С.И. Байесовские сети доверия: логико-вероятностный вывод в ациклических направленных графах. СПб.: Изд-во С.-Петерб. ун-та, 2009. 400 с.

[17] Тулупьев А.Л., Сироткин А.В. Матричные уравнения локального логико-вероятностного вывода оценок истинности элементов в алгебраических байесовских сетях // Вестник Санкт-Петербургского университета. Серия 1: Математика. Механика. Астрономия. 2012. №3. С. 63–72.

[18] Тулупьев А.Л., Столяров Д.М., Ментюков М.В. Представление локальной и глобальной структуры алгебраической байесовской сети в Java-приложениях // Труды СПИИРАН. 2007. №5. С. 71–99.

[19] Тулупьев А.Л., Фильченков А.А., Вальтман Н.А. Алгебраические байесовские сети: задачи автоматического обучения // Информационно-измерительные и управляющие системы. 2011. Т. 9, №11. С. 57–61.

[20] Тулупьева Т.В., Тулупьев А.Л., Пащенко А.Е., Азаров А.А., Степашкин М.В. Социально-психологические факторы, влияющие на степень уязвимости пользователей автоматизированных информационных систем с точки зрения социоинженерных атак // Труды СПИИРАН. 2010. №12. С. 200—214.

[21] Фильченков А.А. Иерархия глобальных структур алгебраической байесовской сети как система графов и гиперграфов // Научно-технический вестник информационных технологий, механики и оптики. 2013. №1. С. 75–81.

[22] Фильченков А.А. Меры истинности и вероятностные графические модели для представления знаний с неопределенностью // Труды СПИИРАН. 2012. №4 (23). С. 254–295.

[23] Фильченков А.А. Субоптимальная звездчатая структура алгебраической байесовской сети // Информационно-управляющие системы. 2013. №2. С. 13–17.

[24] Фильченков А.А., Мусина В.Ф., Тулупьев А.Л. Алгоритм рандомизированного синтеза минимального графа смежности // Труды СПИИРАН. 2013. №2 (25). С. 221–234. URL: http://www.mathnet.ru/trspy592.

[25] Фильченков А.А., Тулупьев А.Л. Структурный анализ систем минимальных графов смежности // Труды СПИИРАН. 2009. №11. С. 104–129. URL: http://www.mathnet.ru/trspy50.

[26] Фильченков А.А., Тулупьев А.Л. Совпадение множеств минимальных и нередуцируемых графов смежности над первичной структурой алгебраической байесовской сети // Вестник Санкт-Петербургского государственного университета. Серия 1. Математика. Механика. Астрономия. 2012. №2. С. 65–74.

[27] Фильченков А.А., Тулупьев А.Л., Сироткин А.В. Мощность множества минимальных графов смежности // Труды СПИИРАН. 2010. №15. С. 136–161. URL: http://www.mathnet.ru/trspy404.

[28] Фильченков А.А., Тулупьев А.Л., Сироткин А.В. Особенности анализа вторичной структуры алгебраической байесовской сети // Труды СПИИРАН. 2010. №1 (12). С. 97–118. URL: http://www.mathnet.ru/trspy373.

[29] Фильченков А.А., Тулупьев А.Л., Сироткин А.В. Структурный анализ клик минимальных графов смежности // Вестник ТвГУ. Серия: Прикладная математика. 2011. №20. С. 139–151.

[30] Фильченков А.А, Фроленков К.В., Сироткин А.В., Тулупьев А.Л. Система алгоритмов синтеза подмножеств минимальных графов смежности // Труды СПИИРАН. 2013. №27. С. 200–244. URL: http://www.mathnet.ru/trspy656.

[31] Ширяев А.Н. Вероятность. М.: Наука, 1989. 640 с.

[32] Carayon P., Wood K.E. Patient safety: the role of human factors and systems engineering // Studies in health technology and informatics. 2010. Vol. 153. Pp. 23–46.

[33] Erdős P., Renyi A. On random graphs. I // Publicationes Mathematicae. 1959. Vol. 6. Pp. 290–297.

[34] Gilbert E.N. Random graphs // Annals of Mathematical Statistics. 1959. №30 (4). Pp. 1141–1144.

[35] Koller D., Friedman N. Probabilistic graphical models: principles and techniques. The MIT Press, 2009. 1208 p.

[36] Lucas P., van der Gaag L., Abu-Hanna A. Bayesian networks in biomedicine and healthcare // Artificial Intelligence in Medicine. 2004. №30. Pp. 201–214.

[37] Prüfer H. Neuer Beweis eines Satzes über Permutationen // Arch. Math. Phys. 1918. Vol. 27. Pp. 742–744.

[38] Shenoy C., Shenoy P.P. Bayesian network models of portfolio risk and return // Computational Finance. Ed. by W. Lo Y.S. Abu-Mostafa, B. LeBaron, A.S. Weigend. Cambridge: MIT Press, 1999. Pp. 85–104.

[39] Vincent C.A. Risk, safety and the dark side of quality // BMJ: British Medical Journal: International Edition. 1997. Vol. 314, №7096. Pp. 1775–1776.

[40] Yazenin A.V. Possibilistic–probabilistic models and methods of portfolio optimization // Perception-based Data Mining and Decision Making in Economics and Finance. Springer Berlin Heidelberg, 2007. Pp. 241–259.

[41] Yazenin A.V. Methods of possibilistic optimization with applications // Forging New Frontiers: Fuzzy Pioneers II. Springer Berlin Heidelberg, 2008. Pp. 385–390.

Библиографическая ссылка

Фильченков А.А., Столярова В.Ф., Тулупьев А.Л. Алгоритм равновероятного синтеза минимального графа смежности // Вестник ТвГУ. Серия: Прикладная математика. 2014. №1. С. 73–89.

Сведения об авторах

1. **Фильченков Андрей Александрович**
научный сотрудник лаборатории теоретических и междисциплинарных проблем информатики СПИИРАН.

Россия, 199178, г. Санкт-Петербург, 14-я линия В.О., д. 39, СПИИРАН. E-mail: aaafil@mail.ru.

2. **Столярова (Мусина) Валерия Фуатовна**
младший научный сотрудник лаборатории теоретических и междисциплинарных проблем информатики СПИИРАН.

Россия, 199178, г. Санкт-Петербург, 14-я линия В.О., д. 39, СПИИРАН. E-mail: valerie.stoliarova@gmail.com.

3. **Тулупьев Александр Львович**
заведующий лабораторией теоретических и междисциплинарных проблем информатики СПИИРАН, профессор кафедры информатики математико-механического факультета Санкт-Петербургского государственного университета (СПбГУ).

Россия, 199178, г. Санкт-Петербург, 14-я линия В.О., д. 39, СПИИРАН. E-mail: ALT@iias.spb.su.

ALGORITHM FOR MINIMAL JOIN GRAPH EQUIPROBABLE SYNTHESIS

Filchenkov Andrey Alexandrovich
Researcher at Theoretical and Interdisciplinary Computer Science Laboratory, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences
Russia, 199178, St. Petersburg, 14-th line V.O., 39, SPIIRAS.
E-mail: aaafil@mail.ru

Stoliarova Valeriya Fuatovna
Junior research fellow at Theoretical and Interdisciplinary Computer Science Laboratory, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences
Russia, 199178, St. Petersburg, 14-th line V.O., 39, SPIIRAS.
E-mail: valerie.stoliarova@gmail.com

Tulupyev Alexander Lvovich
Head of Theoretical and Interdisciplinary Computer Science Laboratory, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences; Professor of Computer Science Department, SPbSU
Russia, 199178, St. Petersburg, 14-th line V.O., 39, SPIIRAS.
E-mail: ALT@iias.spb.su

Received 07.05.2013, revised 18.02.2014.

The algorithm for minimal join graph equiprobable synthesis is proposed in the paper. This algorithm randomly synthesizes graph with minimal amount of edges on the base of given algebraic Bayesian network primary structure, which is represented by set of maximal by inclusion subalphabets of some alphabet. All probable minimal join graphs have equal probability of being synthesized by the algorithm. Algorithm complexity is estimated.

Keywords: algebraical Bayesian networks, secondary structure, minimal join graph, automated learning, machine learning, structure synthesis, random graphs.

Bibliographic citation

Filchenkov A.A., Stoliarova V.F., Tulupyev A.L. Algorithm for minimal join graph equiprobable synthesis. *Vestnik TvGU. Seriya: Prikladnaya matematika* [Herald of Tver State University. Series: Applied Mathematics], 2014, no. 1, pp. 73–89. (in Russian)