Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем
Технология программирования

Харитонов Никита Алексеевич

Алгебраические байесовские сети: представление данных, алгоритмы обработки и реинжиниринг комплекса программ

Бакалаврская работа

Научный руководитель:
проф. кафедры информатики, д. ф.-м. н., доц., Тулупьев А. Л.

Рецензент:
доц. кафедры компьютерных технологий, к. ф.-м. н., Фильченков А. А.

Санкт-Петербург
2017

SAINT-PETERSBURG STATE UNIVERSITY

Software and Administration of Information Systems
Technology of Programming

Nikita Kharitonov

Algebraic Bayesian Networks: data representation, processing algorithms and programme complex reengineering (joint project)

Bachelor's Thesis

Scientific supervisor:
Prof. Computer Science Department, Dc. Sc. in Math, Assoc. Prof. Alexander Tulupyev

Reviewer:
As. Prof. Computer Technology Department, Ph. D. Sc. in Math Andrey Filchenkov

Saint-Petersburg
2017

Оглавление

Введение

Актуальность темы. Верятностные графические модели являются одним из направалений современной информатики [6, 15, 16]. К ним, в частности, относятся марковские сети, байесовские сети доверия и алгебрические байесовские сети [2, 3, 11]. Последние представлют собой ненаправленные графы с идеалами конъюнктов в узлах. Им приписывается скалярные или интервальные оценки вероятности. Также в алгебраических байесовских сетях есть понятие фрагмента знаний — более мелкой структурной единицы, которая введена для более удобной и оптимальной работы с сетью.

Впервые понятие алгебраических байесовских сетей было введено в 1993 году. С тех пор идёт развитие данной теории. В 2009 году была разработана java-библиотека AlgBN Modeller j.v.01[9] для работы с алгебраическими байесовскими сетями, позволяющая хранить фрагменты знаний и переходить от одних к другим. Дальнейшим развитием данного проекта были надстройки Algebraic Bayesian Networks Inferrer и Algebraic Bayesian Networks Propagator[7, 8], в которых реализована поддержка непротиворечивости фрагмента знаний, локальный априорный и апостериорный выводы, некоторые виды глобального логико-вероятностного вывода. В 2011 была реализована C++ библиотека AlgBN KPB Reconciler cpp.v.01[14], также реализующая некоторую функциональность алгебраических байесовских сетей.

В 2016 году была разработана библиотека на языке C#[1, 10], использующая новый матрично-векторный подход. Были реализованы представления фрагментов классов, поддержка локальной непротиворечивости, локальный априорный и апостериорный вывод. Данная работа является прподолжением развития этой библиотеки.

Целью данной работы является доработка и автоматизация алгоритмов поддержки непротиворечивости в алгебраических байесовских сетях. Для достижения поставленной цели были сформулированы следующие **задачи**:

- автоматизация разбора входной строки, содержащей пропозициональную формулу;

- доработка алгоритмы разбора строки и поддержания непротиворечивости;
- реализация программного представления алгебраической байесовской сети и данных алгоритмов;
- разработка тестов и сопроводающей документации.

Апробация результатов исследования. Результаты исследования были представлены на XX Международной конференции по мягким вычислениям и измерениям (SCM'2017) и ещё трёх конференциях.

Публикации. По теме выпускной квалификационной работы было подготовлено 6 публикаций, на момент написания 2 из них опубликованы[12, 17] и 3 приняты к публикации.

Данная выпускная квалификационная работа бакалавра содержит материалы исследований, частично поддержанных грантом РФФИ 15-01-09001 — «Комбинированный логико-вероятностный графический подход к представлению и обработке систем знаний с неопределенностью: алгебраические байесовские сети и родственные модели».

1. Теоретические основы

1.1. Введение

В данной части текста будут введены основные определения и утверждения, необходимые для описания проделанной работы.

1.2. Фрагмент знаний

Алгебраические байесовские сети структурно делятся на *фрагменты знаний*. Более формальное их определение будет дано ниже. И фрагмент знаний, и алгебраическая байесовская сеть строятся над *алфавитом*, состоящим из *атомов*.

Определение 1.1 *[5] Атом — переменная, имеющая некоторое означивание. В теории байесовских сетей атомами считаются элементарные неделимые высказывания о предметной области.*

Определение 1.2 *[5] Алфавит — это множество, содержащее n различных атомов:*

$$A = \{x_0, \dots, x_{n-1}\}.$$

Определение 1.3 *[5] Набор всех возможных пропозициональных формул над A обозначим*

$$F^\circ = F^\circ(A).$$

Далее при описании вместо F используется фактор-множество

$$F = F(A) = F^\circ(A)/\equiv,$$

состоящее из классов эквивалентности пропозициональных формул, построенных над A.

Одними из основных понятий теории алгебраических байесовских сетей являются *конъюнкты* и *кванты*. Именно им приписывается вероятность, о чём речь пойдёт позднее.

Определение 1.4 *[5] Конъюнкт — это цепочка конъюнкций атомов из заданного алфавита.*

Определение 1.5 *[5] Квантом* $\{\tilde{x}_0, \tilde{x}_1, \dots, \tilde{x}_{n-1}\}$ *называется цепочка конъюнкций всех атомов над заданным алфавитом, где атом входит либо сам, либо с отрицательным означиванием.*

Определение 1.6 *[5] Набор квантов над алфавитом A обозначается*

$$Q = Q(A) = \{\tilde{x}_0, \tilde{x}_1, \dots, \tilde{x}_{n-1}\}.$$

После введения данных определений появляется возможность описания вероятностного пространства, работа с которым будет вестись в дальнейшем [5].

Пусть множество Q — множество элементарных событий. На нём задается распределение веротяностей $p^\circ : Q \to [0;1]$, удовлетворяющее условиям:

$$(\forall q \in Q) p^\circ(q) \geq 0; \tag{1}$$

$$\sum_{q \in Q} p^\circ(q) = 1. \tag{2}$$

Для получения вероятностного пространства необходимо определить вероятность $p : 2^Q \to [0;1]$:

$$(\forall S \subset Q) p(S) = \sum_{q \in Q} p^\circ(q).$$

Таким образом, можно говорить о вероятностном пространстве

$$\langle Q, 2^Q, p \rangle.$$

Далее, чтобы определить вероятностное пространство

$$\langle Q, F, p \rangle.$$

необходимо ввести вероятностную меру на множестве F:

$$(\forall f \in F) p(f) = p(S(f)).$$

Здесь S — это функция следующего вида:

$$S : F \to 2^Q, S(f) = S_f,$$

где S_f — множество квантов, учавствующих в совершенной дизъюнктивной нормальной форме (СДНФ) формулы f.

Таким образом, была задана вероятность на пропозициональных формулах $F(A)$, что позволяет ввести ряд определений.

Определение 1.7 *[5] Интервальная оценка вероятности пропозициональной формулы f обозначается следующим образом:*

$$p(f) = [p^-(f); p^+(f)],$$

где $p^-(f), p^+(f)$ — нижняя и верхняя границы соответственно.

Определение 1.8 *[5] Скалярная оценка вероятности может быть определена как интервальная с совпадающими верхней и нижней границами.*

Определение 1.9 *[5] Идеалом конъюнктов над заданным множеством атомов*

$$S = \{y_0, \dots, y_{m-1}\}$$

называются все положительно-означенные цепочки конъюнкций атомарных пропозициональных формул, включая пустую и односимвольную

$$C(S) = \{y_{i_1} \dots y_{i_k} : (i_1, \dots, i_k) \in 2^{0(1)(m-1)}, k \in 0(1)(m-1)\}.$$

Здесь $k \in 0(1)(m-1)$ означает, что k проходит значения от 0 до $m-1$ с шагом 1.

В дальнейшем нередко будет употребляться обозначение C вместо $C(S)$.

Для нумерации квантов и конъюнктов используются схожие механизмы: если переменная входит в конъюнкт или входит в квант без отрицания, ей сопоставляется индекс 1. В противном случае, если переменная не входит в конъюнкт или входит в квант с отрицанием, ей сопоставляется индекс 0.

То есть номер кванта или конъюнкта $b_{m-1} \ldots b_1 b_0$ определяется следующими правилами:

- Для кванта $\tilde{y}_0, \tilde{y}_1, \ldots, \tilde{y}_{n-1}$:

$$b_i = \begin{cases} 0 & \text{если } \tilde{y}_i = \bar{y}_i, \\ 1 & \text{если } \tilde{y}_i = y_i. \end{cases}$$

- Для конъюнкта Z:

$$b_i = \begin{cases} 0 & \text{если } z_i \notin Z, \\ 1 & \text{если } z_i \in Z. \end{cases}$$

Таким образом, каждому кванту или конъюнкту сопоставляется уникальный индекс, равный числу, двоичная запись которого описана выше. Это позволяет упорядочивать данные структуры.

Пусть $\mathbf{Q}^{(n)}$ — вектор скалярных оценок вероятностей истинности квантов над атомарными пропозициональными формулами, $\mathbf{P}^{(n)}$ — вектор скалярных оценок вероятностей конъюнктов из идеала цепочек конъюнкций, построенного над атомарными пропозициональными формулами. Например (здесь и в дальнейшем $e_\wedge$ — пустой конъюнкт):

$$\mathbf{Q}^{(2)} = \begin{pmatrix} p(\bar{x}_2, \bar{x}_1) \\ p(\bar{x}_2, x_1) \\ p(x_2, \bar{x}_1) \\ p(x_2, x_1) \end{pmatrix};$$

$$\mathbf{P}^{(2)} = \begin{pmatrix} p(e_\wedge) \\ p(x_1) \\ p(x_2) \\ p(x_2, x_1) \end{pmatrix}.$$

Данные обозначения будут использоваться при описании непротиворечивости.

Определение 1.10 *[5] Фрагмент знаний со скалярными оценками — пара вида $\langle C, p\rangle$, где C — идеал конъюнктов, p — функция из C в интервал $[0; 1]$. Обозначение — $\mathcal{C}$.*

p является функцией вероятности. То есть, каждому конъюнкту из идеала сопостовляется его скалярная оценка вероятности.

Определение 1.11 *[5] Фрагмент знаний с интервальными оценками — структура вида $\langle C, p\rangle$, где C — идеал конъюнктов, p — функция из C в множество интервалов вида*

$$\{[a; b] : a, b \in [0; 1], a \leq b\}.$$

Стоит отметить, что фрагмент знаний со скалярынми оценками является частным случаем фрагмента знаний с интервальными оценками. Чтобы показать это, функция p определяется следующим образом:

$$\forall c \in C : p(c) = [p(c); p(c)].$$

То есть фрагмент знаний со скалярными оценками идентичен фрагменту знаний с интервальными оценками, нижние и верхние границы которых совпадают.

1.3. Непротиворечивость фрагмента знаний

При присваивании квантам вероятностей последние должны удовлетворять $(1-2)$.

Для наглядности будут рассмотрены примеры:

- Для скалярных оценок.

  Пусть имеются следующие оценки вероятности для кванта $\tilde{x}$: $p(x) = 0.6, p(\bar{x}) = 0.4$. В этом случае противоречия с аксиоматикой верятностной логики отсутсвуют: оценки положительны, их сумма равна 1.

Если же назначить вероятности $p(x) = -0.2, p(\bar{x}) = 1.2$ или $p(x) = 0.6, p(\bar{x}) = 0.5$, то возникнет противоречие: в первом случае сумма равна 1, однако одна из оценок отрицательна, другая — больше 1; во втором — сумма оценок не равна 1.

- Для интервальных оценок.

  Пусть имеются следующие оценки вероятности для кванта $\tilde{x}$: $0.6 \leq p(x) \leq 0.7, 0.3 \leq p(\bar{x}) \leq 0.4$. Данные оценки непртиворечивы, поскольку для любой оценки $p(x)$ из первого нтервала можно выбрать оценку $p(\bar{x})$, которая будет удовлетворять условиям $(1-2)$, например, $p(x) = 0.65$, тогда $p(\bar{x}) = 0.35$.

  В случае, когда сделать подобный выбор возможно только для некоторых оценок, последние можно *согласовать*, убрав те, для которых нет пары. Например: $0.5 \leq p(x) \leq 0.7, 0.2 \leq p(\bar{x}) \leq 0.4$ при согласовании преобразуются в оценки, указанные выше.

  В случае, когда сделать подобный выбор невозможно, оценки *противоречивы*. Например: $0.5 \leq p(x) \leq 0.7, 0.1 \leq p(\bar{x}) \leq 0.2$.

Оценки вероятностей над идеалом конъюнктов также имеют ограничения. Ниже сформулировано определение непротиворечивости для фрагмента знаний со скалярными оценкаи:

Определение 1.12 *[5] Пусть $S \subset A$ — некоторое подмножество алфавита, над S построен идеал конъюнктов $C = C(S)$. Над C построен фрагмент знаний со скалярными оценками вероятности $\langle C, p^{\triangle} \rangle$. Фрагмент знаний $\mathcal{C}$ со скалярными оценками $p^{\triangle}$ непротиворечив тогда и только тогда, когда существует такая дискретная плотность вероятности на квантах Q*

$$p^{\circ} : Q \to [0; 1],$$

которая удовлетворяет требованиям $(1-2)$, вытекающим из аксиоматики вероятностей, и индуцирует вероятность p на пропозициональных формулах $F(S)$, сужение которой на кванты совпадает с

p:
$$(\forall f \in C) p^{\triangle}(f) = p(f).$$

Определение 1.13 *[5] Введем предикат* $\mathtt{Consistent}[\mathcal{C}]$*, который истинен тогда и только тогда, когда его аргумент - фрагмент знаний со скалярными оценками* $\mathcal{C}$ *непротиворечив.*

Фрагмент знаний с интервальными оценками непротиворечив тогда, когда для любой скалярной оценки одного из конъюнктов можно выделить скалярные оценки из других таким образом, что построенный с этими оценками скалярный фрагмент знаний будет непротиворечив. Для введения более формального определения необходимо определить функцию $\mathbf{p}^{\triangle}$ с интервальными значениями над элементами фрагмента знаний $\mathcal{C}$ такую, что:

$$\mathbf{p}^{\triangle}(f) : C \longleftarrow \{[a; b] : a, b \in [0; 1], a \leq b\}.$$

Таким образом формализуется определение интервальных оценок во фрагменте знаний.

Определение 1.14 *[5] Пусть из алфавита атомов выбрано подмножество* S*, над которым построен идеал конъюнктов* $C = C(S)$*, над которым, в свою очередь, построен фрагмент знаний с интервальными оценками вероятностей* $\mathcal{C} = \langle C, \mathbf{p}^{\triangle} \rangle$*. Фрагмент знаний* $\mathcal{C}$ *с интервальными оценками* $\mathbf{p}^{\triangle}$ *непротиворечив тогда и только тогда, когда выполнено следующее условие:*

$$\forall f \in C \forall \rho \in \mathbf{p}^{\triangle}(f)$$

$$\exists p^{\triangle} \in \{\varphi | \varphi : C \longrightarrow [0; 1]\} :$$

$$(p^{\triangle}(f) = \rho) \&$$

$$\& (\forall g \in C : p^{\triangle}(g) \in \mathbf{p}^{\triangle}(g)) \&$$

$$\& \mathtt{Consistent}[\langle C, p^{\triangle} \rangle].$$

Аналогично скалярному случаю:

Определение 1.15 *[5] Введем предикат* $\mathtt{Consistent}[\mathcal{C}]$, *который истинен тогда и только тогда, когда его аргумент — фрагмент знаний с интервальными оценками* $\mathcal{C}$ *непротиворечив.*

В определении ниже будет формализовано понятие *согласуемого* фрагмента знаний, речь о котором шла в примерах.

Определение 1.16 *[5] Фрагмент знаний* $\langle C, \mathbf{p}^{\triangle} \rangle$ *с интервальными оценками* $\mathbf{p}^{\triangle}$ *является согласуемым, если существует такой непротиворечивый набор скалярных оценок* p', *что*

$$\forall f \in \mathcal{C} p'(f) \in \mathbf{p}^{\triangle}(f).$$

Для более удобной реализации алгоритмов на компъютере приведенные выше определения переводятся на матрично-векторный язык. Для этого вводятся следующие матрицы:

$$\mathbf{I}_1 = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix};$$

$$\mathbf{J}_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix};$$

$$\mathbf{1}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix};$$

$$\mathbf{0}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Матрицы более высоких степеней вводятся следующим образом:

$$\mathbf{I}_m = \mathbf{I}_1^{[m]},$$

$$\mathbf{1}_m = \mathbf{1}_1^{[m]},$$

$$\mathbf{0}_m = \mathbf{0}_1^{[m]},$$

где под степенью подразумевается степень Кронекера матриц.

Ниже приведён пример матрицы **I** второй и третей степеней:

$$\mathbf{I}_2 = \begin{pmatrix} 1 & -1 & -1 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix};$$

$$\mathbf{I}_3 = \begin{pmatrix} 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 0 & 1 & 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Используя данные матрицы, можно сформулировать следующую теорему, позволяющие выразить вероятности конъюнктов через вероятности квантов:

Теорема 1.1 *[5]*

$$\mathbf{I}_m \times \mathbf{P}^{(m)} = \mathbf{Q}^{(m)},$$

$$\mathbf{J}_m \times \mathbf{Q}^{(m)} = \mathbf{P}^{(m)}.$$

Далее приведены две теоремы, позволяющие перейти к формулировке матрично-векторного условия непротиворечивости фрагмента знаний со скалярными оценками.

Теорема 1.2 *[5]*

$$(\mathbf{I}_m \times \mathbf{P}^{(m)})\mathbf{1}_m = \mathbf{1}.$$

Теорема 1.3 *[5] Требования аксиоматики вероятностной логики* (1–2) *к значениям дискретной плотности вероятностей на квантах в матрично-векторном виде выразятся как*

$$\mathbf{Q}^{(m)} \geq \mathbf{0}_m,$$

$$\mathbf{Q}^{(m)}\mathbf{1}_m = 1.$$

Теперь на основании теорем $(2.1 - 2.3)$ и определения 2.12 можно сформулировать следующую теорему:

Теорема 1.4 *[5] Для того, чтобы фрагмент знаний $\mathcal{C} = \langle C, \mathbf{P}^{(m)} \rangle$ со скалярными оценками вероятностей был непротиворечивым, необходимо и достаточно, чтобы выполнялось векторное неравенство*

$$\mathbf{I}_m \mathbf{P}^{(m)} \geq \mathbf{0}_m. \qquad (3)$$

Проверка непротиворечивости во фрагменте знаний с интервальными оценками происходит путём решения задач линейного программирования. Для описания данной задачи необходимо ввести ряд определений:

Определение 1.17 *[5] $\mathcal{E}^{(m)}$ — множество линейных ограничений, состоящее из линейных неравенств, задаваемых условием* 3.

Определение 1.18 *[5] $\mathcal{D}^{(m)}$ — множество линейных ограничений на основе сведений из предметной области, состоящее из двойных линейных неравенств над вероятностями элементов идеала и представимое в векторном виде как*

$$\mathbf{P}_{\circ}^{-,(m)} \leq \mathbf{P}^{(m)} \leq \mathbf{P}_{\circ}^{+,(m)},$$

где $\mathbf{P}_{\circ}^{-,(m)}$ — вектор, составленный из констант — исходных нижних границ вероятностей истинности соответствующих конъюнктов, а $\mathbf{P}_{\circ}^{+,(m)}$ — вектор, но содержащий верхние границы. Компоненты обоих векторов с нулевым индексом всегда равны единице.

Определение 1.19 *[5] Объединение двух множеств ограничений, исходящих из аксиоматики вероятностной логики и из предметной области, имеет следующее обозначение:*

$$\mathcal{R}^{(m)} = \mathcal{D}^{(m)} \cup \mathcal{E}^{(m)}.$$

Ниже приведён пример подобных ограничений для фрагмента знаний, построенного над алфавитом $A = \{x_1, x_2\}$:

$$\mathcal{E}^{(2)} = \left\{ \begin{matrix} p(x_1, x_2) \geq 0, \\ p(x_1) - p(x_1, x_2) \geq 0, \\ p(x_2) - p(x_1, x_2) \geq 0, \\ 1 - p(x_1) - p(x_2) + p(x_1, x_2) \geq 0 \end{matrix} \right\};$$

$$\mathcal{D}^{(2)} = \left\{ \begin{matrix} 1 \leq 1 \leq 1, \\ p_\circ^-(x_1) \leq p(x_1) \leq p_\circ^+(x_1), \\ p_\circ^-(x_2) \leq p(x_2) \leq p_\circ^+(x_2), \\ p_\circ^-(x_1, x_2) \leq p(x_1, x_2) \leq p_\circ^+(x_1, x_2) \end{matrix} \right\}.$$

Для проверки фрагмента знаний с интервальными оценками на непротиворечивость для каждой формулы f из идеала конъюнктов предлагается решать следующие задачи линейного программирования:

$$p^-(f) = \min_{\mathcal{R}^{(m)}}\{p(f)\},$$

$$p^+(f) = \min_{\mathcal{R}^{(m)}}\{p(f)\}.$$

Для удобства определения ограничений в более общей форме вводится обозначения для векторов, содержащих уточненные верхние и нижние оценки вероятностей, обозначающиеся $\mathbf{P}^-$ и $\mathbf{P}^+$ соответсвенно.

Тогда ограничения имеют следующий вид:

$$\mathbf{P}^- = \min_{\mathcal{R}}\{\mathbf{P}\}, \tag{4}$$

$$\mathbf{P}^+ = \min_{\mathcal{R}}\{\mathbf{P}\}. \tag{5}$$

Ниже формулируется теорема, позволяющая приводить фрагмент знаний к непротиворечивому виду.

Теорема 1.5 *[5] Если задачи линейного программирования $4-5$ имеют решения, то фрагмент знаний $\mathcal{C} = \langle C, \mathbf{P}_\circ^-, \mathbf{P}_\circ^+ \rangle$ с, возможно, уточненными интервальными оценками непротиворечив, а оценки представляют собой замкнутые промежутки.*

Также необходимо уделить внимание априорному выводу, поскольку описанный ниже парсер строки используется в первую очередь в нем.

Априорный вывод в алгебраических байесовских сетях заключается в том, чтобы по известным вероятностным оценкам истинности заданных пропозициональных формул построить вероятностную оценку пропозициональной формулы, не вошедшей в число заданных.

По теореме о совершенной дизъюнктивной нормальной форме (СДНФ) любая пропозициональная формула представима в виде дизъюнкции некоторого набора квантов. Для определения этого набора вводится характерестический вектор χ, который равен 1, если квант входит в разложение формулы, и 0 в противоположном случае.

Тогда вероятность формулы равна сумме вероятностей квантов этого набора:

$$p(f) = (\chi_f, \mathbf{Q}_n).$$

Вероятность формулы выражается через вероятности конъюнктов следующим образом:

$$\forall f \exists! \, \mathbf{L}_f : p(f) = \mathbf{L}_f \mathbf{P}_n,$$

где $\mathbf{L}_f$ — вектор вещественных констант, совпадающий по размерности с $\mathbf{P}_n$.

1.4. Алгебраическая байесовская сеть

Алгебраическая байесовская сеть, образованная из идеалов конъюнктов и их оценок истинности, является одной из возможных математических моделей представления совокупности фразментов знаний. Более формальное определение звучит следующим образом:

Определение 1.20 *[5] Определим алгебраическую байесовскую сеть $\mathcal{N}$ как набор $\mathcal{N}^\circ$ фрагментов знаний:*

$$\mathcal{N}^\circ = \{\mathcal{C}_i\}_{i=1}^{i=n} = \{\langle C_i, \mathbf{p}_i \rangle\}_{i=1}^{i=n},$$

или, в случае скалярных оценок —

$$\mathcal{N}^\circ = \{\mathcal{C}_i\}_{i=1}^{i=n} = \{\langle C_i, p_i \rangle\}_{i=1}^{i=n},$$

Также необходимо определить *носитель* алгебраической байесовской сети:

Определение 1.21 *[5] Носителем* $\mathtt{N} = supp(\mathcal{N})$ *алгебраической байесовской сети* $\mathcal{N}$ *будет объединение идеалов конъюнктов, лежащих в основе фрагментов знаний, вошедших в сеть:* $\mathtt{N} = \bigcup_{i=1}^{i=n} C_i$.

Теперь, когда дано определение алгебраической байесовской сети, можно поговорить о её структуре. Существует несколько уровней представления структуры, ниже даны определения первых двух:

Определение 1.22 *[5] Первичная структура алгебраической байесовской сети — это набор фрагментов знаний, которые в неё вошли.*

Определение 1.23 *[5] Вторичная структура алгебраической байесовской сети — это граф смежности, узлы которого находятся во взаимооднозначном соответствии с набором фрагментов знаний, вошедших в первичную структуру сети, причем весом узла является идеал конъюнктов (без пустого элемента), являющийся носителем соответствующего фрагмента знаний.*

Вторичная структура может быть представлена в виде графа смежности или дерева смежности, что обозначается соответствующим образом:

$$\mathcal{N}_G = \langle \mathcal{N}^\circ, G \rangle, \mathcal{N}_T = \langle \mathcal{N}^\circ, T \rangle.$$

В проекте работа ведётся с ациклическими алгебраическими байесовскими сетями.

1.5. Виды непротиворечивости в алгебраической байесовской сети

1.5.1. Локальная непротиворечивость

Определение 1.24 *АБС является локально непротиворечивой, если каждый отдельно взятый ФЗ в сети непротиворечив.*

Сформулируем данное утверждение на матрично-векторном языке. Условие непротиворечивости для одного фрагмента знаний записывается следующим образом: $\mathbf{PI}_{\mathrm{n}} \geq 0$. Тогда для множества фрагментов знаний, составляющих АБС, условие локальной непротиворечивости запишется как

$$\mathrm{Local}(\mathcal{N}^{\circ}) \equiv \forall \langle C_i, \mathbf{P}^i \rangle \in \mathcal{N}^{\circ}\, \mathbf{P}^i \mathbf{I}_{\mathrm{n}} \geq 0$$

Локальная непротиворечивость накладывает самые ”слабые” условия на оценки вероятностей и тем самым является самое легкодостижимой, но при этом обеспечивает наименьшую степень согласованности данных в базе знаний.

1.5.2. Экстернальная непротиворечивость

Следующим рассматриваемым видом непротиворечивости является экстернальная непротиворечивость. Определение экстернально непротиворечивой АБС расширяет локальную непротиворечивость, накладывая условия равенства оценок вероятностей конъюкнтов, принадлежащих двум ФЗ.

Определение 1.25 *АБС является экстернально непротиворечивой, если она локально непротиворечива и оценки истинности конъюнкта, входящего в два и более ФЗ совпадают:*

$$\begin{aligned}\mathrm{External}(\mathcal{N}^{\circ}) \equiv & \mathrm{Local}(\mathcal{N}^{\circ}) \\ & \& \quad \forall C_i, C_j \in \mathcal{N}^{\circ} \quad \forall \mathbf{P}^i[k], \mathbf{P}^j[m] \in C_i \cap C_j \quad \mathbf{P}^i[k] = \mathbf{P}^j[m]\end{aligned}$$

В случае интервальных оценок вероятностей в ФЗ равенство оценок вероятностей подразумевает равенство нижней и верхней границ интервалов оценок. Данный вид непротиворечивости автоматически поддерживается в программных комплексах, где оценки совпадающих конъюкнтов хранятся в единственном экземпляре, а ФЗ лишь ссылаются на список собственных конъюкнтов.

1.5.3. Интернальная непротиворечивость

Определение 1.26 *АБС является интерально непротиворечивой, если она локально непротиворечива и для любого конъюнкта для любого скалярного значения из интервала оценки его истинности можно выбрать такие оценки во всех ФЗ, что получившаяся АБС будет экстернально непротиворечивой.*

$$\begin{aligned}\text{Internal}(\mathcal{N}^\circ) \equiv & Local(\mathcal{N}^\circ) \\ & \& \quad \forall c_i \in N \, \forall \mathbf{P}_i \in [\mathbf{P}_i^-, \mathbf{P}_i^+] \quad \text{External}(\mathcal{N}^\circ|\mathbf{P}_i)\end{aligned}$$

1.5.4. Глобальная непротиворечивость

Определение 1.27 *АБС является глобально непротиворечивой, если ее с имеющимися оценками можно погрузить в непротиворечивый объемлющий фрагмент знаний C и при этом оценки вероятностей $\mathbf{P}^i$ не изменятся.*

$$\text{Global}(\mathcal{N}^\circ) \equiv \exists \langle C, \mathbf{P} \rangle : \, \mathbf{PI}_\text{n} \geq 0 \, \& \, \forall f \in N \, \mathbf{P}_f = \mathbf{P}_{N_f}$$

1.6. Выводы по главе

Описанный в данной главе теоретический материал позволяет перейти к описанию алгебраической баейсовской сети и алгоритмов непротиворечивости в ней.

2. Парсер строки содержащей пропозициональную формулу

2.1. Введение

При разработке программного обеспечения многие процессы необходимо автоматизировать. Комплекс программ для работы с алгебраическими байесовскими сетями не является исключением. В ходе разработки возникла необходимость приведения строки, содержащей пропозициональную формулу, к характеристическому вектору квантов. Данный раздел работы описывает решение этой задачи при помощи построения совершенной дизъюнктивной нормальной формы, для чего использовалось дерево разбора строки и последующий обход его в глубину.

2.2. Постановка задачи

На вход подаётся строка, содержащая пропозициональную формулу. Необходимо проверить правильность введённой строки и преобразовать её в характеристический вектор квантов.

2.3. Алгоритм решения

Так как из строки получается характеристический вектор квантов, то задача сводится к нахождению СДНФ.

Для получения СДНФ строится таблица истинности. Тем наборам значений переменных, для которых формула истинна, сопоставляется квант, в который переменные входят по следующему принципу: если значение 1, то входит сама переменная; в противном случае — её отрицание.

В данной работе для получения положительноозначенных наборов используется дерево разбора и рекурсивный спуск по нему.

Пример

Пусть имеется фрагмент знаний над $\{x_1, x_2, x_3\}$.

Рассматривается логическое выражение $f \sim \overline{x}_1 \equiv x_2 \wedge x_3$. Для него строится таблица истинности (таблица 1).

x_1	x_2	x_3	$\overline{x}_1 \equiv x_2 \wedge x_3$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Таблица 1: Таблица истинности для выражения $\overline{x}_1 \equiv x_2 \wedge x_3$

Тогда характеристический вектор квантов выглядит следующим образом:

$$\chi_f^T = (0, 0, 0, 1, 1, 1, 1, 0).$$

Пусть известен вектор вероятностей квантов:

$$\mathbf{Q}_3 = \begin{pmatrix} p(\overline{x}_1\overline{x}_2\overline{x}_3) \\ p(\overline{x}_1\overline{x}_2x_3) \\ p(\overline{x}_1x_2\overline{x}_3) \\ p(\overline{x}_1x_2x_3) \\ p(x_1\overline{x}_2\overline{x}_3) \\ p(x_1\overline{x}_2x_3) \\ p(x_1x_2\overline{x}_3) \\ p(x_1x_2x_3) \end{pmatrix}.$$

Тогда вероятность формулы считается:

$$p(f) = (\chi_f^T, \mathbf{Q}_3) = p(\overline{x}_1x_2x_3) + p(x_1\overline{x}_2\overline{x}_3) + p(x_1\overline{x}_2x_3) + p(x_1x_2\overline{x}_3).$$

2.4. Построение дерева разбора

Для построения дерева разбора использовалась модификация алгоритма получения обратной польской записи [13].

В строке встречаются следующие типы элементов: переменные, символы операций, открывающая или закрывающая скобки.

При разборе строка разбивается на эти элементы. В её конец добавляется специальный символ \$.

Создаётся два стека:

- **E**, в котором хранятся вершины поддеревьев графа; изначально пуст;
- **T**, в котором хранятся операции; изначально содержит символ \$.

Основная часть алгоритма состоит в следующем: сравнивается текущий элемент строки и элемент на вершине T. При этом переменные в рассмотрении не участвуют, а сразу добавляются в E. В зависимости от результатов сравнения происходит одна из 6 операций:

1. Поместить символ из исходной строки в T;

2. Извлечь из T операцию. В зависимости от её арности извлечь 1 или 2 элемента, сформировать новый узел и поместить в E. Записать текущую операцию в стек T;

3. Удалить символ из T;

4. Извлечь из T операцию. В зависимости от её арности извлечь 1 или 2 элемента, сформировать новый узел и поместить в E. Совершить операцию согласно таблице 2;

5. Ошибка, завершить разбор;

6. Заверить разбор.

Выбор операции производится при помощи таблицы 2, в которой по горизонтали выбирается текущий символ строки, а по вертикали - символ на вершине T.

	$\$$	(	$\equiv$	$\Rightarrow$	$\vee$	$\wedge$	$\overline{x}$	)
$\$$	6	1	1	1	1	1	1	5
(	5	1	1	1	1	1	1	3
$\equiv$	4	1	2	1	1	1	1	4
$\Rightarrow$	4	1	4	2	1	1	1	4
$\vee$	4	1	4	4	2	1	1	4
$\wedge$	4	1	4	4	4	2	1	4
$\overline{x}$	4	1	4	4	4	4	2	4

Таблица 2: Выбор действия

Пример

В таблице 3 приведена последовательность действий при разборе строки, содержащей $f \sim \overline{x}_1 \equiv x_2 \wedge x_3$.

Пояснение шагов:

1. На вершине T находится \$, текущий символ строки — отрицание. Согласно таблице, выбирается действие 1: отрицание добавляется в стек T.

Строка:	E :	T :	Действие:	шаг
$\overline{x}_1 \equiv (x_2 \wedge x_3)\$$		$\$$	1	1
$\equiv (x_2 \wedge x_3)\$$	x_1	$\$, !$	4	2
$\equiv (x_2 \wedge x_3)\$$	$\overline{x}_1$	$\$$	1	3
$(x_2 \wedge x_3)\$$	$\overline{x}_1$	$\$, \equiv$	1	4
$\wedge x_3)\$$	$\overline{x}_1, x_2$	$\$, \equiv, ($	1	5
$)\$$	$\overline{x}_1, x_2, x_3$	$\$, \equiv, (, \wedge$	4	6
$)\$$	$\overline{x}_1, x_2 \wedge x_3$	$\$, \equiv, ($	3	7
$\$$	$\overline{x}_1, x_2 \wedge x_3$	$\$, \equiv$	4	8
$\$$	$\overline{x}_1 \equiv x_2 \wedge x_3$	$\$$	6	9

Таблица 3: Пример разбора строки $\overline{x}_1 \equiv (x_2 \wedge x_3)$

2. x_1 как переменная добавляется в E. На вершине T находится !, текущий символ строки — эквиваленция. Её приоритет меньше, чем у отрицания, и, согласно таблице, выбирается действие 4: отрицание применяется для последнего элемента E. При этом рассматриваемым символом строки остаётся эквиваленция.

3. Аналогично п.1, эквивалентность добавляется в T.

4. x_2 добавляется в E. Согласно таблице, номер действия 1, поэтому открывающая скобка переносится в T.

5. x_3 добавляется в E, $\wedge$ переносится в T.

6. Текущий символ строки — закрывающая скобка. Поэтому операции из T будут поочерёдно переноситься в E, пока на вершине T не окажется открывающая скобка. В данном случае действие происходит один раз — в E создаётся вершина с $\wedge$, её потомки — переменные, над которыми происходит операция: x_1 и x_2.

7. На вершине T оказывается открывающая скобка. Она удаляется из T, берётся следующий символ строки.

8. В E создаётся вершина с эквивалентностью.

9. Разбор строки окончен. В T не осталось операций, в E только один элемент — дерево разбора.

2.5. Построение СДНФ по дереву разбора

Для получения СДНФ из дерева разбора используется метод рекурсивного спуска. Вершине сопостовляется значение 1. Это объясняется тем, что в ней стоит операция, выполняемая в последнюю очередь, то есть её результат будет результатом всего логического выражения. А для получения СДНФ необходимо, чтобы пропозициональная формула была истинна.

Далее оценивается, при каких значениях потомков вершина может иметь такое значение. Операция повторяется для потомков при этих значениях. Таким образом обходится всё дерево.

В итоге получается набор значений переменных. При совпадении некоторых наборов оставляется только один из них. Набор, в котором одна переменная имеет противоположные значения, изымается из рассмотрения.

Пример

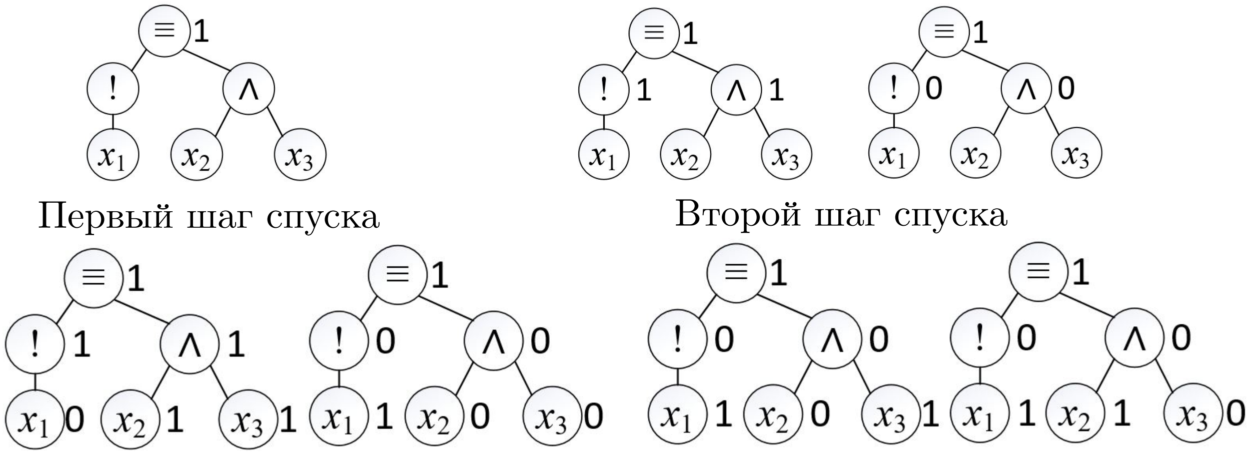
Рассмаривается всё то же логическое выражение $\overline{x}_1 \equiv x_2 \wedge x_3$. На рисунке 1 проиллюстрированы шаги спуска по дереву.

Пояснение шагов:

1. Вершине сопоставляется положительное означивание.

2. Так как в вершине находится эквивалентность, то её потомки должны быть одновременно либо 1, либо 0. Спуск продолжается для каждого из этих вариантов в отдельности.

3. Операция повторяется для следующих вершин, в итоге получается 4 набора значений переменных x_1, x_2, x_3: $(0, 1, 1), (1, 0, 0), (1, 1, 0)$, $(1, 0, 1)$.



Первый шаг спуска

Второй шаг спуска

Третий шаг спуска

Рис. 1: Спуск по дереву

2.6. Выводы по главе

Реализованный функционал в дальнейшем позволит облегчить работу пользователя, которому удобнее вводить пропозициональную формулу, а не характеристический вектор.

3. Структура разработанной программной части

3.1. Введение

В данной части приведена реализованная структура классов программного представления АБС, псевдокод алгоритмов поддержания экстернальной и интернальной непротиворечивости, указаны проведенные доработки программного комплека.

3.2. Структура классов

Для реализации алгоритмов поддержки непротиворечивости была разработана система классов, описывающих АБС, структура которых изображена на рис. 2.

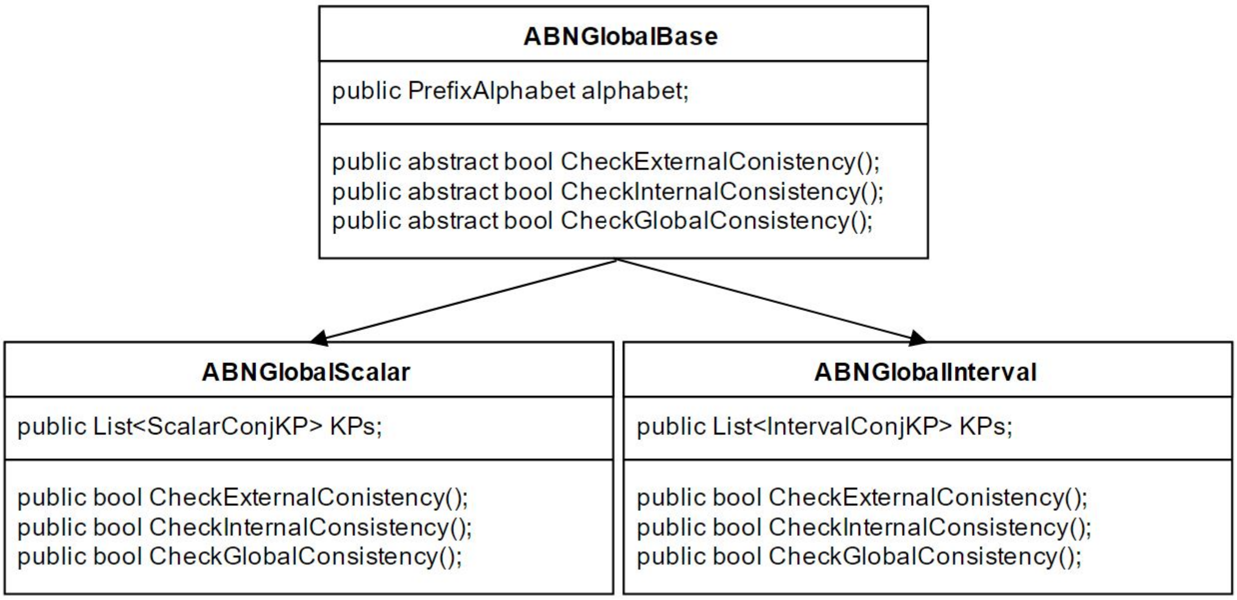


Рис. 2: Структура классов АБС.

Класс ABNGlobalBase является абстракцией, от которой наследуются реализации АБС со скалярными и интервальными оценками. В нём же описаны основные методы поддержки непротиворечивости, которые будут реализованы в наследниках. Поскольку АБС обоих типов строятся над идентичными алфавитами, то алфавит также объявлен в ABNGlobalBase. В то же время списки фрагментов знаний не описаны в базовом классе. Это объясняется тем, что для создания этих списков используются различные по своей сути структуры: IntervalConjKP и ScalarConjKP, для которых нет возможности создать единого “родителя”, имеющего необходимый функционал.

Класс ABNGlobaLScalar является представлением АБС со скалярными оценками и содержит реализацию алгоритма поддержки непроиворечивости. В АБС данного типа из экстернальной следуют интернальная и глобальная непротиворечивости, поэтому функциональность методов совпадает. При вызове проверки любого типа происходит обращение к внутренней процедуре CheckConistensy, которая реализует поддержку экстернальной непротиворечивости.

Для реализации непротиворечивости в АБС с интервальными оценками создан класс ABNGlobalInterval. На данный момент АБС считается ациклической, то есть из интернальной непротиворечивости следует глобальная, вследствие чего алгоритм поддержки последней не имеет должной реализации. Как временный вариант, до перехода проекта к обработке циклических АБС, метод CheckGlobalConsistency возвращает результат работы метода CheckIntervalConsistency.

Конструкторы АБС как со скалярными, так и с интервальными оценками, получают на вход алфавит, над которым будет построена сеть, и список ФЗ.

В обоих классах также реализованы внутренние методы, упрощающающие проверку непротиворечивости. В некоторых случаях они совпадают для обоих типов АБС, поэтому содержатся и в ABNGlobalBase. Среди них такие методы, как:

- protected List<int> GetLocalByGlobal(long knowlegePatternGlobalIndex, int localIndex) — метод, позволяющий узнать глобальный индекс символа алфавита из ФЗ. Возвращает указанный индекс в виде списка 0 и 1.

- protected string getElementNameByIndex(long knowlegePatternGlobalIndex, int localIndex) — благодаря этому методу можно узнать имя конъюнкта из ФЗ, в том числе если он содержит несколько символов алфавита. Необходимость данного метода заключается в том, что ФЗ не позволяет получить доступ к последнему. При этом имена переменных активно используются в различных видах поддержания непротиворечивости.

Иные вспомогательные методы позволяют получать пересечение двух ФЗ (сепаратор), обновлять оценки вероятностей конъюнктов и выполнять другие элементарные алгоритмы.

3.3. Скалярные оценки вероятностей: алгоритмы поддержания непротиворечивости

Так как проверка непротиворечивости в скалярной АБС происходит по одному алгоритму, приведенному ниже, он вынесен в отдельный метод, который вызывается в функциях CheckExternalConistency(), CheckIntervalConsistency и CheckGlobalConsistency. В отличие от проверки непротиворечивости в интервальной АБС по результатами работы данных функций оценки не могут быть изменены. Алгоритм выглядит следующим образом:

Algorithm 1 Алгоритм проверки экстернальной непротиворечивости в скалярной алгебраической байесовской сети

while есть ФЗ в KPs **do**
    if ФЗ непротиворечив **then**
        Перейти к следующему ФЗ;
    else
        return false;
    end if
    Удалить ФЗ из KPs;
end while

while есть конъюнкты в АБС **do**
  if оценки данного конъюнкта иден-тичны во всех ФЗ **then**
    Перейти к следующему конъюнкту;
  else
    return false;
  end if
end while
return true.

3.4. Интервальные оценки вероятностей: алгоритмы поддержания непротиворечивости

В АБС с интервальными оценками вероятностей реализована проверка как экстернальной, так и интернальной непро-тиворечивости. Однако, в отличие от скалярного случая, при поддержании непротиворечивости в случае интервальных оценок может происходить их изменение. При этом методы CheckExternalConsistency и CheckIntervalConsistency возвращают лишь информацию о том непротиворечива АБС или нет, то есть информация о возможном изменении оценок скрыта от пользователя.

Алгоритмы, используемые в модуле поддержания глобальной непротиворечивости, уже были описаны в [15], однако в ходе разработки удалось произвести их доработку и улучшение.

В ходе создания ЗЛП при проверке интернальной непротиворечивости также происходит запоминание того какие переменные (конъюнкты) в ней присутствуют. Принимая этот факт во внимание нам удается избежать добавления новых ограничений в ЗЛП. Это позволяет сделать задачу менее объёмной и тратить меньше ресурсов на её решение.

Для проверки интернальной непротиворечивости используется библиотека lp_solve55 [4], которая позволяет решать задачи линейного программирования. Псевдокоды алгоритмов проверки экстернальной и интернальной непротиворечивости представлены ниже:

Algorithm 2 Алгоритм поддержания экстернальной непротиворечивости в интервальной алгебраической байесовской сети

Create stack;

while есть ФЗ в KPs **do**

  Взять ФЗ из АБС, имеющий наименьшее число пересечений с другими ФЗ;

  if ФЗ непротиворечив **then**

    Добавить в stack;

  else

    return false;

  end if

  Удалить ФЗ из KPs;

end while

while есть конъюнкты в stack **do**

  Найти наиболее узкую оценку конъюнкта;

  Присвоить её всем вхождениям конъюнкта в ФЗ из stack;

  Перейти к следующему конъюнкту;

end while

while есть ФЗ в stack **do**

  if ФЗ непротиворечив **then**

    Добавить в KPs;

  else

    return false;

  end if

end while

return true.

Algorithm 3 Алгоритм поддержания интернальной непротиворечивости в интервальной алгебраической байесовской сети

```
Создать ЗЛП;
while есть ФЗ в KPs do
    Добавить условия из ФЗ в ЗЛП;
end while
while есть конъюнкты в АБС do
    Решить ЗЛП относительно данного конъюнкта на max;
    if есть решение then
        Изменить оценку;
    else
        return false;
    end if
    Решить ЗЛП относительно данного конъюнкта на min;
    if есть решение then
        Изменить оценку;
    else
        return false;
    end if
end while
return true;
```

3.5. Выводы по главе

Разработанная структура классов и алгоритмы поддержки непротиворечивости позволяют перейти к реализации алгоритмов глоабльного априорного, апостериорного вывода.

4. Программная реализация

4.1. Введение

В данной части приведены примеры работы описанных выше алгоритмов.

4.2. Пример работы проверки разных степеней непротиворечивости

Для проверки корректной работы модуля проверки непротиворечивости был разработан обширный спектр тестов, от анализа простейших вспомогательных методов до проверки интернальной непротиворечивости в АБС с интервальными оценками. В работе приведены два примера работы алгоритмов поддержания непротиворечивости интервальной АБС, однако тесты для проверки непротивоечивости в АБС со скалярными оценками в данной статье не рассматриваются, что объясняется простотой проверки: методы поддержания непротиворечивости в них выдают только результат true или false и не возвращают уточненные оценки вероятностей элементов АБС, в отличие от интервальной.

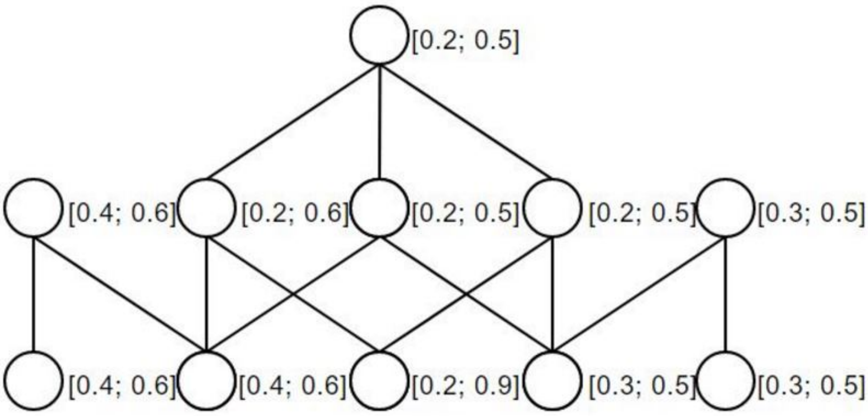
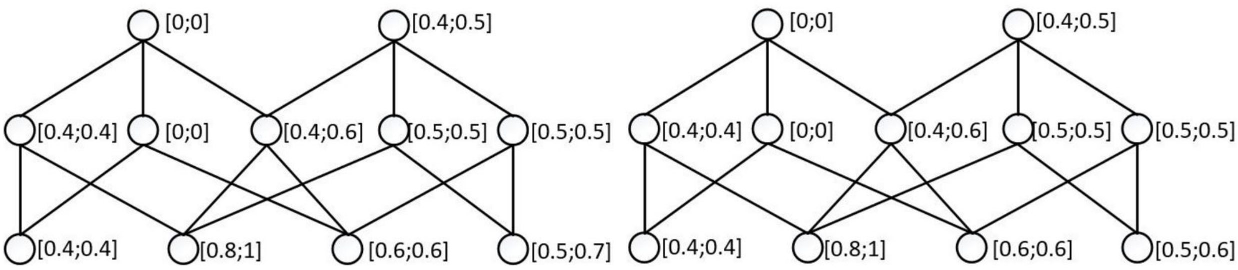


Рис. 3: Результат проверки экстернальной непротиворечивости.

Для примера поддержании экстернальной непротиворечивосты была взята АБС, состоящая из трех ФЗ: первый построен над атомами $x1, x2$, второй — над $x2, x3, x4$, третий — над $x4, x5$. В первом ФЗ все конъюнкты имеют оценки $[0.4; 0.6]$, во втором — $[0.2; 0.9]$, в третьем — $[0.3; 0.5]$. В результате поддержки экстернальной непротиворечивости оценки в АБС некоторые изменились. Поскольку метод возвращает только bool значение, показывающе, непротиворечива ли АБС, и не даёт информации об изменении оценок, была произведена пошаговая отладка теста. Она позволила определить, как именно изменились оценки. Результат работы изображён на рисунке 3 и соответствует ожидаемому результату, полученному на основании теоретических выкладок.

Для проверки интернальной непротиворечивости использовалась АБС, представленная на рисунке 4. Данный пример интересен тем фактом, что в результате выполнения алгоритма должны измениться оценки в элементе $[0.5; 0.7]$ на $[0.5; 0.6]$. Как и в первом примере, была произведена пошаговая отладка, в результате которой было обнаружено, что оценки действительно меняются и в итоге получается АБС, изображённая на рис 4.



До проверки После проверки

Рис. 4: Пример проверки интернальной непротиворечивости.

5. Заключение

В ходе данной работы достигнута основная цель — реализованы алгоритмы поддержки непротиворечивости в алгебраических байесовских сетях.

Выполнены поставленные задачи:

- разработан парсер строки;
- реализованы алгоритмы поддержания экстернальной и интернальной непротиворечивости;
- реализована структура классов алгебраической байесовской сети;
- написаны охватывающие весь объём работы тесты, каждый метод имеет документацию.

Результаты вошли в 5 публикаций:

- Харитонов Н.А. Представление пропозициональной формулы в форме СДНФ для оценки вероятности ее истинности. // Материалы 6-й всероссийской научной конференции по проблемам информатики СПИСОК-2016. 26–29 апреля 2016 г. Санкт-Петербург — СПб.: ВВМ, 2016. сс. 574-580
- Харитонов Н.А., Мальчевская Е.А. Локальный априорный вывод в АБС: автоматизация анализа пропозициональной формулы. //Региональная информатика (РИ-2016). Юбилейная XV Санкт-Петербургская международная конференция «Региональная информатика (РИ-2016)». Санкт-Петербург, 26-28 октября 2016 г.: Материалы конференции. СПОИСУ.– СПб, 2016. с. 522
- E. Malchevskaya, N. Kharitonov, A. Zolotin, A. Birillo, Algebraic Bayesian Networks: Probabilistic-Logic Inference Algorithms and Storage Structures //Finnish-Russian University Cooperation in Telecommunications 3 - 7 april 2017 (in press)

- Харитонов Н.А. Алгоритмы непротиворечивости в алгебраических байесовских сетях Всероссийская научная конференция по проблемам информатики СПИСОК-2017 (в печати)

- N. Kharitonov, A. Tulupyev, A Zolotin Software Implementation of Reconcilation Algorithms in Algebraic Bayesian Networks // International Conference on Soft Computing and Measurements (in press)

Также результаты были представлены на следующих конференциях: СПИСОК-2016, СПИСОК-2017, Региональная информатика, SCM-2017.

Была подана заявка на регистрацию программы для ЭВМ: Харитонов Н.А, Тулупьев А.Л.Algebraic Bayesian Networks String Parser Version 01 for CSharp (AlgBN StringParser cs.v.01)

Данная выпускная квалификационная работа бакалавра содержит материалы исследований, частично поддержанных грантом РФФИ 15-01-09001 — «Комбинированный логико-вероятностный графический подход к представлению и обработке систем знаний с неопределенностью: алгебраические байесовские сети и родственные модели».

Полученные результаты позволяют продолжить разивитие функциональности библиотеки, реализовать априорный и апостериорный логико-вероятностный выводы в алгебраических байесовских сетях.

Список литературы

[1] Mal'chevskaya E.A. Berezin A.I. Zolotin A.A. Tulupyev A.L. Algebraic Bayesian Networks: Local Probabilistic-Logic Inference Machine Architecture and Set of Minimal Joint Graphs //Proceedings of the First International Scientific Conference "Intelligent Information Technologies for Industry"(IITI'16). — Springer International Publishing, 2016. – С. 69-79.

[2] Nielsen T.D. Jensen F.V. Bayesian networks and decision graphs. — Springer Science Business Media, 2009.

[3] V. Jensen F. An introduction to Bayesian networks. — London : UCL press, 1996. — Т. 210.

[4] lp_solve 5.5. Sourceforge lp_solve reference guide. — URL: `http://lpsolve.sourceforge.net/` (дата обращения: 07.04.2017).

[5] А.Л. Тулупьев. Алгебраические байесовские сети: логико-вероятностная графическая модель баз фрагментов знаний с неопределенностью. — Наук по спец. – 2009. – Т. 5.

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

[7] А.Л. Тулупьев. Система для апостериорного вывода в алгебраических байесовских сетях и их фрагментах Algebraic Bayesian Networks Propagator, Version 01 for Java (AlgBN Propagator j.v.01) (Свидетельство). Свид. о гос. рег. прогр. для ЭВМ. Рег. № 2009613804 (16.07.2009). — Роспатент.. Бюлл. «Прогр. для ЭВМ, БД, топол. инт. микросх.». 2009. № 4. С. 65.

[8] А.Л. Тулупьев. Система для синтеза непротиворечивых алгебраических байесовских сетей и их фрагментов Algebraic BayesianNetworks Inferrer, Version 01 for Java (AlgBN Inferrer j.v.01)

(Свидетельство). Свид. о гос. рег. прогр. для ЭВМ. Рег. № 2009613803 (16.07.2009). — Роспатент.. Бюлл. «Прогр. для ЭВМ, БД, топол. инт. микросх.». 2009. № 4. С. 65.

[9] А.Л. Тулупьев. Система представления алгебраических байесовских сетей и их фрагментов Algebraic Bayesian Networks Modeler, Version 01 for Java (AlgBN Modeler j.v.01) (Свидетельство). Свид. о гос. рег. прогр. для ЭВМ. Рег. № 2009613802 (16.07.2009). — Роспатент.. Бюлл. «Прогр. для ЭВМ, БД, топол. инт. микросх.». 2009. №4. С. 64–65.

[10] Мальчевская Е.А. Золотин А.А. Логико-вероятностный вывод в АБС: архитектура и примеры использования программного комплекса на языке C#. — Гибридные и синергетические интеллектуальные системы. – 2016. – С. 181-187.

[11] Маталыцкий М.А. Колузаева Е.В. Марковские сети массового обслуживания произвольной топологии с доходами. — Академия наук Беларуси. — 2009. — Т. 53. — №. 4.

[12] Н.А. Харитонов. Представление пропозициональной формулы в форме СДНФ для оценки вероятности ее истинности. — Материалы 6-й всероссийской научной конференции по проблемам информатики СПИСОК-2016. 26–29 апреля 2016 г. Санкт-Петербург — СПб.: ВВМ, 2016. сс. 574-580.

[13] Сироткин А.В. Тулупьев А.Л. Локальный априорный вывод в алгебраических байесовских сетях: комплекс основных алгоритмов. — Труды СПИИРАН. Вып. 5. СПб.: Наука, 2007. С. 100-111.

[14] Тулупьев А.Л. Сироткин А.В. Программа для моделирования фрагмента знаний алгебраической байесовской сети, поддержания его непротиворечивости и апостериорного вывода в нем Algebraic Bayesian Network Knowledge Pattern Modeler, Reconciler and Propagator Version 01 for C++ (AlgBN KP MRP cpp.v.01) (Сви-

детельство). Свид. о гос. регистрации программы для ЭВМ. Рег. №2010615242(13.08.2010). — Роспатент.

[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] Харитонов Н.А. Мальчевская Е.А. Локальный априорный вывод в АБС: автоматизация анализа пропозициональной формулы. — Региональная информатика (РИ-2016). Юбилейная XV Санкт-Петербургская международная конференция «Региональная информатика (РИ-2016)». Санкт-Петербург, 26-28 октября 2016 г.: Материалы конференции. СПОИСУ.– СПб, 2016. с. 522.