

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт космических и информационных технологий  
Базовая кафедра «Интеллектуальные системы управления»

УТВЕРЖДАЮ  
Заведующий кафедрой

\_\_\_\_\_ Ю.Ю. Якунин

"\_\_" \_\_\_\_\_ 2018 г.

### МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Алгоритм пропагации свидетельств в байесовской сети с циклами

27.04.03 Системный анализ и управление

27.04.03.02 Системный анализ данных и технологий принятия решений

Научный руководитель \_\_\_\_\_ доцент, канд.техн. наук  
подпись, дата

А.А. Даничев

Выпускник \_\_\_\_\_  
подпись, дата

А.В. Зыков

Рецензент \_\_\_\_\_ доцент, канд. физ-мат. наук В.А.Красиков  
подпись, дата

Красноярск 2018

## РЕФЕРАТ

Магистерская диссертация по теме «Алгоритм пропагации свидетельств в байесовской сети с циклами» содержит 45 страниц текстового документа, 12 использованных источников, 25 иллюстраций, 5 таблиц, 35 формул.

**БАЙЕСОВСКИЕ СЕТИ, БАЙЕСОВСКАЯ СЕТЬ С ЦИКЛАМИ, АЛГОРИТМ ПРОПАГАЦИИ СВИДЕТЕЛЬСТВ, РАСПРОСТРАНЕНИЕ СВИДЕТЕЛЬСТВ.**

**Объектом** исследования являются Байесовские сети.

**Предметом** исследования являются алгоритмы пропагации в байесовских сетях с циклами.

**Целью** данной работы является разработка алгоритма распространения свидетельств в байесовской сети с циклами.

**Задачи исследования:**

- Изучить научно-методическую литературу по проблеме исследования;
- Проанализировать имеющиеся алгоритмы распространения свидетельств в байесовских сетях;
- Разработать собственный алгоритм распространения свидетельств в байесовских сетях;
- Реализовать объектную модель алгоритма.
- Апробировать алгоритм на искусственных циклических примерах

В ходе проведения исследования, были сделан вывод, о том, что для сетей со сложной структурой, известные алгоритмы распространения свидетельств не пригодны.

В разработанном алгоритме распространения свидетельств в байесовских сетях с циклами несогласованность потомков не влияет на априорные вероятности их предков. Возможно получать интервальные оценки вероятностей.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1 Основные понятия теории байесовских сетей .....	6
1.1 Событие и вероятность события.....	6
1.2 Условные вероятности и теорема Байеса .....	7
1.3 Условная независимость .....	8
1.4 Байесовские сети .....	9
1.4.1 Байесовские сети доверия .....	11
1.4.2 Связи между узлами сети и d-разделимость .....	13
1.5 Изолированные циклы БСД.....	15
1.6 Ограничения подхода изолированных циклов.....	21
2 Алгоритмы пропации свидетельств и объектная модель алгоритма .....	24
2.1 Общие формулы для распространения свидетельств.....	24
2.2 Алгоритмы пропации свидетельств.....	25
2.2.1 Инициализация вершин в байесовской сети .....	25
2.2.2 Алгоритм распространения свидетельств в байесовской сети без циклов	25
2.2.3 Алгоритм распространения свидетельств в байесовской сети с циклами .	27
2.3 Объектная модель для реализации программного обеспечения, для работы с Байесовскими сетями.....	30
3 Апробация алгоритма .....	34
3.1 Апробация алгоритма на сети без циклов .....	34
3.2 Апробация алгоритма на сети с циклами .....	40
3.2.1 Подсчет вероятностей по измененному алгоритму.....	43
ЗАКЛЮЧЕНИЕ .....	44
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	45

## ВВЕДЕНИЕ

Байесовские сети представляют собой удобный инструмент для описания достаточно сложных процессов и событий с неопределенностями. Для описания байесовской сети необходимо определить структуру графа и параметры каждого узла. Эта информация может быть получена непосредственно из данных или из экспертных оценок. Основные этапы этого процесса: идентификация переменных, определение структуры, определение параметров [1,2,3,4]. После того, как байесовская сеть сконструирована, она готова для того, чтобы с её помощью можно было проводить вычисления. После поступления некоторого количества свидетельств, могут быть вычислены апостериорные вероятности. Проблема использования БС заключается в том, что реальные модели могут содержать в себе циклы, но большинство алгоритмов требуют ациклической структуры сети. Отсюда можно выделить две группы алгоритмов [5].

Первая группа требует ациклическости первичной Байесовской сети, при этом прорабатывается их скорость работы, возможность интервальных оценок, совместная вероятность изменения, расширенное алгебраическое дополнение, множественное значение вершин [3].

Вторая группа рассматривает циклические первичные байесовские сети и приводит циклы к согласованному виду, при этом в результатах отсутствуют интервальные оценки, что видится неприемлемым, поскольку не удалось найти апробации данных алгоритмов на случай известных правильных “ответов”, а также отсутствуют оценки ошибок согласования. Большой вопрос также вызывают способы согласования вершин в цикле, например, в алгоритме «Николенко-Тулупьева-Сироткина» [2,3], ввод несогласованных свидетельств игнорируется, что так же является не совсем правильным. Использование подобных алгоритмов возможно только для простых сетей, где нет

пересекающихся циклов, а в самих циклах отсутствуют вершины со множеством потомков.

**Объектом** исследования являются Байесовские сети.

**Предметом** исследования являются алгоритмы пропагации в байесовских сетях с циклами.

**Целью** данной работы является разработка алгоритма распространения свидетельств в байесовской сети с циклами.

**Задачи исследования:**

- Изучить научно-методическую литературу по проблеме исследования;
- Проанализировать имеющиеся алгоритмы распространения свидетельств в байесовских сетях;
- Разработать собственный алгоритм распространения свидетельств в байесовских сетях;
- Реализовать объектную модель алгоритма.
- Апробировать алгоритм на искусственных циклических примерах.

# 1 Основные понятия теории байесовских сетей

## 1.1 Событие и вероятность события

Основополагающим аспектом теории вероятности является понятие элементарного события, представляющего собой набор допустимых результатов, получаемых в ходе эксперимента. Событие, в свою очередь, может состоять из множества элементарных событий. Согласно теории байесовских сетей элементарное событие – это квант, описываемый ключевыми понятиями предметной области, т.е. пропозиционными выражениями. В таком случае событием считается дизъюнкция элементарных событий[5].

Вероятность представляет собой не что иное, как меру неопределенности и меру истинности некоторого события. Классическая теория сравнивает вероятность с концепцией экспериментов, которые можно воспроизвести. Тогда данное понятие обозначает частоту появления определенного события. Самым известным примером является подбрасывание монеты, в результате которого экспериментатор получает два возможных результата: монета падает орлом или решкой вверх. Причем, если монета «честная», то вероятность каждого исхода стремится к 50%.

Формальное определение вероятности будет выглядеть следующим образом[6]: если имеется пространство элементарных событий  $\Psi$ , то вероятность – это числовая функция  $p: \Psi \rightarrow [0; 1]$ , такая, что:

$$\sum_{e \in \Psi} p(e) = 1.$$

Вероятность любого события  $Z \in \Psi$  может быть записана формулой:

$$p(Z) = \sum_{e \in Z} p(e).$$

## 1.2 Условные вероятности и теорема Байеса

Условной вероятностью события  $Z$  при условии события  $K$  с  $p(K) > 0$  называется величина

$$p(Z|K) = p(ZK) / p(K)$$

Вероятности  $p(Z)$  и  $p(K)$  носят название априорных, то есть тех вероятностей, которые задаются до эксперимента и могут быть получены от экспертов предметной области. Противопоставляются им апостериорные вероятности, например,  $p(Z|K)$ , то есть условная вероятность случайного события  $Z$  при получении свидетельства о наступлении события  $K$ . Справедливы также следующие утверждения[6,7]:

$$p(Z|Z) = 1; p(\emptyset|Z) = 0; p(Z|K) = 1, Z \supseteq K;$$
$$p(Z1 \cup Z2|K) = p(Z1|K) + p(Z2|K), Z1 \cap Z2 = \emptyset.$$

События  $Z1, \dots, Zn$  составляют разбиение  $\Psi$ , если  $\sum_{i=1}^n Zi = \Psi$ ,  $Zi \cap Zj = \emptyset$  и  $p(Zi) > 0$ ,  $i, j \in \overline{1, n}$ ,  $i \neq j$ .

Тогда полная вероятность может быть вычислена с помощью выражения:

$$p(K) = \sum_{i=1}^n p(K|Zi) p(Zi)$$

$$\text{Если } p(Z) > 0, \text{ то } p(K) = p(K|Z)p(Z) + p(K|\bar{Z})p(\bar{Z}).$$

В случае, если события  $Z1, \dots, Zn$  таковы, что  $p(Z1 \cap \dots \cap Zn) > 0$ , то  $p(Z1 \cap \dots \cap Zn \cap K) = p(Z1) p(Z2|Z1) p(Z3|Z1 \cap Z2) \dots p(K|Z1 \cap \dots \cap Zn)$ .

Если события  $Z1, \dots, Zn$  составляют разбиение  $\Psi$  и  $p(K) > 0$ , то

$$p(Zi|K) = p(Zi) p(K|Zi) / \sum_{j=1}^n p(Zj) p(K|Zj)$$

Данное выражение носит название теоремы Байеса, в упрощенном виде записываемой, как:

$$p(Z|K) = p(Z)p(K|Z) / p(K).$$

### 1.3 Условная независимость

В общем случае, события считаются независимыми, если наступление одного из событий не влияет на наступление другого. Иными словами, события  $Z$  и  $K$  называются независимыми относительно вероятности  $p$ , если

$$p(Z \cap K) = p(Z)p(K).$$

Тогда для таких событий справедливо:

$$p(K|Z) = p(K);$$

$$p(\bar{Z} \cap \bar{K}) = p(\bar{Z})p(\bar{K});$$

$$p(\bar{Z} \cap K) = p(K) - p(Z \cap K) = p(K) - p(Z)p(K) = p(\bar{Z})p(K).$$

В общем виде события  $Z_1, \dots, Z_n$  называются независимыми в совокупности [6,8], если для любых  $k \in \overline{1, n}$  и  $1 \leq i_1 < \dots < i_k \leq n$  верно, что

$$p(Z_{i_1}, \dots, Z_{i_k}) = p(Z_{i_1}) \dots p(Z_{i_k}).$$

Объединение понятий условной вероятности и независимости выражается в следующем определении: события  $Z$  и  $T$  условно независимы при условии события  $K$ , если

$$p(\bar{Z} \cap \bar{K} \cap \bar{T})p(K) = p(\bar{Z} \cap \bar{K})p(\bar{K} \cap \bar{T}).$$



Причем, если

$$p(\bar{K}) > 0, \text{ то } p(\bar{Z} \cap \bar{T} | \bar{K}) = p(\bar{Z} | \bar{K})p(\bar{T} | \bar{K}).$$

#### 1.4 Байесовские сети

Байесовские сети представляют собой графическую вероятностную модель, которая представляет собой множество переменных и их вероятностных зависимостей. БС представляют собой удобный инструмент для описания достаточно сложных процессов и событий с неопределенностями. БС оказалась особенно полезной при разработке и анализе машинных алгоритмов обучения. Основной идеей построения графической модели является понятие модульности, то есть разложение сложной системы на простые элементы. Для объединения отдельных элементов в систему используются результаты теории вероятностей.

Динамические байесовские сети являются обобщенной моделью в пространстве состояний. Название «динамические» указывает не на зависимость структуры от времени, а только на зависимость от моделирования процесса. СММ является упрощенной ДБС без причинно-следственных связей. Если пространство состояний непрерывное, то такая модель называется фильтром Кальмана, в дискретном случае — ДБС. Структура БС остается неизменной во всех временных срезах. Срезом называют текущее состояние ДБС в дискретный момент времени. Верши- 8 на в сети может иметь родителя только в своем временном срезе или в непосредственно предшествовавшем временном срезе. Другими словами, ДБС определяется как Марковский процесс первого порядка[4,9].

БС обычно используются в случае, когда имеется ряд связанных друг с другом событий (пропозиций), вероятности истинности которых могут быть получены от специалистов предметной области. БС должна сопоставить информацию (знания) от экспертов так, чтобы отдельные факты не

противоречили друг другу. Затем принимаются во внимание фактические данные (свидетельства) о произошедших событиях или возможные вариации априорных вероятностей.

В общем случае знания представляют собой информацию, описывающую определенные объекты предметной области. Если говорить о знаниях в искусственном интеллекте, то можно охарактеризовать их следующим образом:

- информация, представленная определенным образом, которая позволяет решать задачи в конкретной предметной области методом логического вывода;

- форма представления информации, отличающаяся от данных рядом особенностей:

- внутренняя интерпретируемость (каждое знание имеет уникальное наименование, по которому система может к нему обратиться);

- структурированность;

- связность (возможность определения отношений между знаниями).

Для представления знаний применяется целый ряд методов: текстовый (книги, документы), графический (иллюстрации, картины), формализованный (математические формулы и выражения) и другие. Также были разработаны специальные средства, такие как продукционные правила, семантические сети, логика предикатов, фреймы и другие.

Модель представления знаний (данных, сведений) в области отображения и обработки знаний с неопределенностью может быть описана следующими положениями[4]:

- предметная область представлена сведениями, являющимися набором переменных; переменные, в свою очередь, соответствуют утверждениям предметной области, описывают параметры ее объектов;

- каждой переменной сопоставляется ряд возможных значений, находящихся на дискретном или непрерывном интервале;

- некоторым переменным и математическим выражениям, в которых они используются, может соответствовать либо точечное значение, либо набор значений (ограниченное множество возможных значений);

- значения переменных ограничены определениями или свойствами объектов предметной области;

- множество переменных может быть разделено на отдельные подмножества, внутри которых значения переменных взаимосвязаны; переменные могут входить сразу в несколько подмножеств; все это делается для упрощения работы с имеющимися знаниями;

- необходимо учитывать возможность реализации математических операций над множеством неопределенных знаний исходя из существующих информационных технологий.

Приведенные выше требования применимы к реализации, теоретическому изучению и практическому применению байесовских сетей доверия (БСД), описанных Дж. Пиэрлом[7].

#### **1.4.1 Байесовские сети доверия**

Байесовские сети доверия (БСД, причинно- следственные сети) предназначены для создания вероятностных экспертных систем с использованием байесовского вывода и применяются при решении задач с неопределенностью, связанной с:

- многообразием причинно-следственных связей в исследуемой предметной области;

- наличием неполных знаний о параметрах;

- случайными факторами, влияющими на осуществление событий.

БСД применяются в задачах описания взаимосвязей различных переменных и могут быть изображены, как направленный граф причинно- следственных связей, являющийся ациклическим (направленные циклы

недопустимы). События являются вершинами графа, а причинно- следственные связи между ними обозначены ребрами. Такой подход позволяет производить оценку вероятностей событий, зависящих от других событий[2,3].

Рассмотрим подробнее пример БСД, представленный на рисунке 1. Изображенная модель представляет упрощенный вариант определения причины того, что лужайка во дворе стала мокрой. Наблюдаемое следствие (мокрая лужайка) позволяет сделать заключение о промежуточных причинах (был включен распылитель для полива газона, прошел дождь) и установить первопричину (облачная или ясная погода). При этом необходимо учитывать, что все события носят вероятностный характер. Например, возможно, что даже при включении распылителя трава может остаться сухой (в случае поломки или засора распылителя), и наоборот, нельзя однозначно утверждать, что лужайка стала мокрой из-за включения распылителя (возможно, прошел дождь)[1].



Рисунок 1– Пример графа байесовской сети доверия

Таким образом, каждый вывод может быть получен только с некоторой степенью уверенности, для оценивания которой и предназначены байесовские сети. Одним из центральных принципов применения БСД является пропация свидетельств (апостериорный, байесовский вывод), когда на вход поступают сведения о наступлении событий, относящихся к определенным узлам.

### 1.4.2 Связи между узлами сети и d-разделимость

Последовательная связь – это прямая связь между узлами сети (рисунок 2а). В этом случае 1 влияет на 2, а 2, в свою очередь, влияет на 3, и узлы 1 и 3 получают связанными. Однако, если в 2 поступило свидетельство, связь между 1 и 3 нарушается: если мы твердо знаем, что промежуточное событие произошло, связи между его причиной и следствием уже нет (ее обеспечивал тот факт, что вероятность следствия зависит от вероятности причины, но если следствие уже произошло, этот механизм больше не работает). На рисунок 2 такому виду связи соответствует соединение узлов  $c$ ,  $s$ ,  $w$ , при котором  $c$  воздействует на  $s$ , а  $s$  – на  $w$  (если погода была облачной, то оценка вероятности того, что лужайка стала мокрой, повышается). Но при поступлении сведений в  $s$  связь между  $c$  и  $w$  изменяется: если мы уверены в том, что распылитель был включен, то нам уже не важна причина этого события – была ли погода ясной, или произошло что-то другое – на вероятность следствия это уже не повлияет.

В случае расходящейся связи (diverging connection) у одного узла сети имеется несколько потомков (рисунок 2б). Информация об одном из потомков может повлиять на вероятность другого потомка одного и того же узла. Это связано с тем, что не только информация о произошедшей причине повышает вероятность (правильнее, конечно, было бы сказать «оценку вероятности», а еще правильнее и более нейтрально — «степень нашей уверенности в»), но для краткости мы будем говорить просто «вероятность») следствия, но и случившееся следствие повышает вероятность причины. Но если общий предок уже получил означивание, то связь нарушается. Действительно, если мы уже твердо знаем, что причина двух следствий имеет место, эти следствия становятся независимыми. Например, если мы знаем, что прошел дождь, то вероятность того, что была облачная погода, увеличивается. А если погода была облачной, то увеличивается вероятность дождя[4,7].

Сходящаяся связь (converging connection) возникает, когда два узла имеют общего потомка (рисунок 2в). Эта ситуация — самая интересная. Казалось бы, связи между узлами 1 и 3 нет: если произошло 1, то это повлияет на вероятность события 2, но вероятность 3 от этого измениться не должна. Однако ситуация меняется, если свидетельство 2 уже получено. Действительно, после получения свидетельства 2 вероятности обеих его причин повышаются: раз уж что-то произошло, значит, какая-то причина у этого была — неизвестно, какая именно, но обе причины становятся более вероятными. Но если теперь мы вдруг узнаем, что одна из причин произошла, это должно понизить вероятность другой причины: ведь следствие уже объяснено, и его влияние на вторую причину должно уменьшиться. Итак, в случае сходящейся связи зависимость должна появляться, только если общий потомок (или любой из его потомков — там ситуация будет аналогичной) уже получил какое-то означивание. На рисунке 1 таким образом соединены узлы  $s$ ,  $r$  и  $w$ . С одной стороны, непосредственная связь между узлами  $s$  и  $r$  отсутствует: если наступило событие  $s$ , то изменится вероятность события  $w$ , но это не повлияет на вероятность  $r$ . Лужайка стала мокрой, но мы не можем сказать, что включение распылителя как-то скажется на вероятности того, что пройдет дождь.

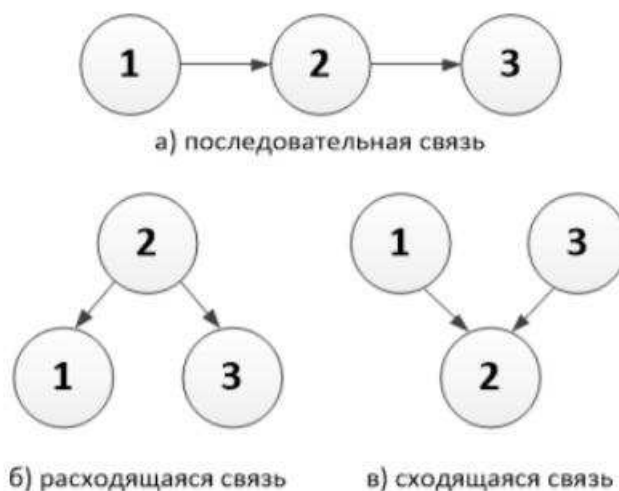


Рисунок 3 – Виды связей, используемых для d-разделимости

С другой стороны, получение сведений о наступлении события  $w$  увеличивает вероятность обеих причин: лужайка стала мокрой, но причина этого не установлена, однако обе причины становятся более вероятными. Если станет известно, что одна из причин имела место, это уменьшит вероятность другой причины, т.к. следствие уже определено, т.е. его воздействие на вторую причину должно снизиться. Тогда зависимость проявляется только тогда, когда имеет место означивание общего дочернего узла нескольких родительских вершин[5].

Дадим пояснение термину  $d$ -разделимости. Два узла направленного графа  $x$  и  $y$  называются  $d$ -разделенными, если для всякого пути из  $x$  в  $y$  (здесь не учитывается направление ребер) существует такой промежуточный узел  $z$  (не совпадающий ни с  $x$ , ни с  $y$ ), что-либо связь в пути в этом узле последовательная или расходящаяся, и узел  $z$  получил означивание, либо связь сходящаяся, и ни узел  $z$ , никакой-либо из его потомков означивания не получил. В противном случае узлы называются  $d$ -связанными[1,3].

### 1.5 Изолированные циклы БСД

Рассмотрим БСД, представляющую собой изолированный цикл из конечного числа вершин. Это означает, что в качестве начальных данных заданы тензоры условных вероятностей

$$p(x_2 | x_1), p(x_2 | \bar{x}_1), p(x_3 | x_2), p(x_3 | \bar{x}_2), \dots, p(x_n | x_{n-1}), p(x_n | \bar{x}_{n-1}).$$

Попытаемся выразить тензоры совместных вероятностей через начальные данные. Следуя основным свойствам условных вероятностей, мы можем записать систему линейных уравнений:

$$\begin{cases} p(x_1) = p(x_1 | x_n)p(x_n) + p(x_1 | \bar{x}_n)(1 - p(x_n)), \\ p(x_2) = p(x_2 | x_1)p(x_1) + p(x_2 | \bar{x}_1)(1 - p(x_1)), \\ \vdots \\ p(x_n) = p(x_n | x_{n-1})p(x_{n-1}) + p(x_n | \bar{x}_{n-1})(1 - p(x_{n-1})). \end{cases}$$

Эта система имеет очень простую структуру и может быть решена за линейное время (для этого можно просто последовательно подставлять неизвестные вероятности в последующие уравнения системы, элиминируя переменные и, вследствие того, что каждая из переменных встречается лишь дважды, получая в конце линейное уравнение линейного размера). Отметим, что все условные вероятности уже даны, и мы решаем систему относительно маргинальных вероятностей  $p(x_i)$  [1,2,10].

Обозначая

$$\xi_{ij} = p(x_i | \bar{x}_j) - p(x_i | x_j)$$

видим, что система имеет вид  $\mathbf{A}\mathbf{p} = \mathbf{B}$ , где

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & \xi_{1n} \\ \xi_{21} & 1 & 0 & \dots & 0 & 0 \\ 0 & \xi_{32} & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \xi_{n,n-1} & 1 \end{pmatrix},$$

$$\mathbf{p} = \begin{pmatrix} p(x_1) \\ p(x_2) \\ p(x_3) \\ \vdots \\ p(x_n) \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} p(x_1 | \bar{x}_n) \\ p(x_2 | \bar{x}_1) \\ p(x_3 | \bar{x}_2) \\ \vdots \\ p(x_n | \bar{x}_{n-1}) \end{pmatrix}.$$

Для того, чтобы исследовать систему на возможную вырожденность, вычислим определитель матрицы системы  $\det \mathbf{A}$ , разложив его по первой строке:

$$\det \mathbf{A} = 1 + (-1)^n \xi_{1n} \xi_{21} \xi_{32} \dots \xi_{n,n-1}$$



Мы видим, что система может быть вырожденной, только если  $\xi_{i,i-1} = \pm 1$  для всех  $i = 1..n$

и, более того, знаки подобраны так, чтобы общее произведение

$$(-1)^n \xi_{1n} \xi_{21} \xi_{32} \dots \xi_{n,n-1}$$

оказалось, равно минус единице, а не единице. Это соответствует случаю, когда все  $x_i$  семантически эквивалентны, т.е. описывают одно и то же утверждение  $x$ .

В этом случае каждая из переменных, соответствует либо  $x$ , либо  $\tilde{x}$ , таким образом,  $p(\tilde{x}_i, \tilde{x}_j)$  равна нулю или единице. Тогда, действительно, на  $p(x)$  не налагается никаких ограничений, и каждое значение даст решение системы. Однако этот случай на практике бесполезен, т.к. нет никакой пользы (а на самом деле есть вред) от использования нескольких вершин графа для обозначения одного и того же события. По этим причинам в дальнейшем мы рассматриваем невырожденный случай.

Итак, в невырожденном случае вышеописанная система дает точечные вероятности всех вершин цикла. Следующая формула позволяет нам вычислить также совместные вероятности соседних вершин цикла

$$p(\tilde{x}_i \tilde{x}_{i-1}) = p(\tilde{x}_{i-1})p(\tilde{x}_i | \tilde{x}_{i-1})$$

Сейчас мы сначала подробно рассмотрим простейшие случаи — циклы из двух и трех вершин, — а затем перейдем к более общим вопросам.

В простейшем случае циклических байесовских сетей доверия — случае цикла из двух вершин  $x_1$  и  $x_2$  — никакой неопределенности в распределении вероятностей не содержится. Маргинальные вероятности подчиняются следующей системе ограничений:

$$\begin{cases} p(x_1) + (p(x_1 | x_2) - p(x_1 | \bar{x}_2))p(x_2) = p(x_1 | \bar{x}_2), \\ (p(x_2 | x_1) - p(x_2 | \bar{x}_1))p(x_1) + p(x_2) = p(x_2 | \bar{x}_1). \end{cases}$$

Решая ее, получаем маргинальные вероятности в явном виде:

$$p(x_1) = \frac{p(x_1 | \bar{x}_2) - (p(x_1 | x_2) - p(x_1 | \bar{x}_2))p(x_2 | \bar{x}_1)}{1 - (p(x_1 | x_2) - p(x_1 | \bar{x}_2))(p(x_2 | x_1) - p(x_2 | \bar{x}_1))},$$

$$p(x_1) = \frac{p(x_2 | \bar{x}_1) - (p(x_2 | x_1) - p(x_2 | \bar{x}_1))p(x_1 | \bar{x}_2)}{1 - (p(x_1 | x_2) - p(x_1 | \bar{x}_2))(p(x_2 | x_1) - p(x_2 | \bar{x}_1))}.$$

Теперь легко получить все распределение целиком:

$$\begin{aligned} p(x_1 x_2) &= p(x_1) p(x_2 | x_1), \\ p(x_1 \bar{x}_2) &= p(x_1) (1 - p(x_2 | x_1)), \\ p(\bar{x}_1 x_2) &= (1 - p(x_1)) p(x_2 | \bar{x}_1), \\ p(\bar{x}_1 \bar{x}_2) &= (1 - p(x_1)) (1 - p(x_2 | \bar{x}_1)). \end{aligned}$$

Отметим, что, как и в общем случае, эти формулы имеют смысл только если  $(p(x_1 | x_2) - p(x_1 | \bar{x}_2)) (p(x_2 | x_1) - p(x_2 | \bar{x}_1)) \neq 1$ , что может произойти лишь в описанном выше вырожденном случае[2,3].

Случай цикла из трех вершин (рисунок 4) вносит элементы неопределенности в получаемый ответ, однако эта неопределенность «одномерна», и задача линейного программирования становится тривиальной. Стоит отдельно отметить, что задача эта может, несмотря на свою тривиальность, не решиться; цикл может оказаться противоречив не только в описанной выше вырожденной ситуации. В данном частном случае система линейных уравнений принимает следующий вид:

$$\begin{cases} p(x_1) = p(x_1 | x_3) p(x_3) + p(x_1 | \bar{x}_3) (1 - p(x_3)), \\ p(x_2) = p(x_2 | x_1) p(x_1) + p(x_2 | \bar{x}_1) (1 - p(x_1)), \\ p(x_3) = p(x_3 | x_2) p(x_2) + p(x_3 | \bar{x}_2) (1 - p(x_2)). \end{cases}$$

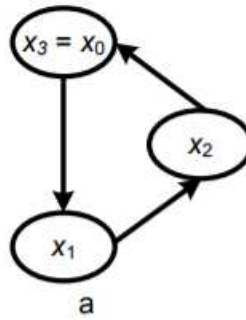


Рисунок 4 – Пример цикла из трех вершин

Так как в данном случае любая пара переменных является парой соседей в цикле, все вероятности конъюнкций длины два заданы однозначно. Единственным неизвестным является  $p(x_1, x_2, x_3)$ . Решив систему, получим маргинальные вероятности вершин цикла и попарные совокупные вероятности. Примем теперь за неизвестную величину  $p(x_1, x_2, x_3)$  и запишем систему ограничений, вытекающую из простейших свойств вероятности[3,4]:

$$\begin{cases} p(x_1x_2x_3) \geq 0, \\ p(x_1x_2) - p(x_1x_2x_3) \geq 0, \\ p(x_1x_3) - p(x_1x_2x_3) \geq 0, \\ p(x_2x_3) - p(x_1x_2x_3) \geq 0, \\ p(x_1) - p(x_1x_2) - p(x_1x_3) + p(x_1x_2x_3) \geq 0, \\ p(x_2) - p(x_1x_2) - p(x_2x_3) + p(x_1x_2x_3) \geq 0, \\ p(x_3) - p(x_1x_3) - p(x_2x_3) + p(x_1x_2x_3) \geq 0, \\ 1 - p(x_1) - p(x_2) - p(x_3) + p(x_1x_2) + p(x_2x_3) + p(x_1x_3) - p(x_1x_2x_3) \geq 0. \end{cases}$$

Это – тривиальный одномерный случай задачи линейного программирования. Для ее решения достаточно пересечь вышеуказанные интервалы. В явном виде ее решение запишется следующим образом:

$$\max \left\{ \begin{array}{l} 0, \\ p(x_1x_2) + p(x_1x_3) - p(x_1), \\ p(x_1x_2) + p(x_2x_3) - p(x_2), \\ p(x_1x_3) + p(x_2x_3) - p(x_3) \end{array} \right\} \leq p(x_1x_2x_3) \leq \min \left\{ \begin{array}{l} p(x_1x_2), \\ p(x_1x_3), \\ p(x_2x_3), \\ 1 - p(x_1) - p(x_2) - p(x_3) + \\ + p(x_1x_2) + p(x_1x_3) + p(x_2x_3) \end{array} \right\}.$$

Затем можно выразить через  $p(x_1, x_2, x_3)$  и все остальные вероятности, формирующие совокупное распределение:

$$\begin{aligned}
 p(\bar{x}_1 x_2 x_3) &= p(x_2 x_3) - p(x_1 x_2 x_3), \\
 p(x_1 \bar{x}_2 x_3) &= p(x_1 x_3) - p(x_1 x_2 x_3), \\
 p(x_1 x_2 \bar{x}_3) &= p(x_1 x_2) - p(x_1 x_2 x_3), \\
 p(\bar{x}_1 \bar{x}_2 x_3) &= p(x_3) - p(x_2 x_3) - p(x_1 x_3) + p(x_1 x_2 x_3), \\
 p(\bar{x}_1 x_2 \bar{x}_3) &= p(x_2) - p(x_1 x_2) - p(x_2 x_3) + p(x_1 x_2 x_3), \\
 p(x_1 \bar{x}_2 \bar{x}_3) &= p(x_1) - p(x_1 x_2) - p(x_1 x_3) + p(x_1 x_2 x_3), \\
 p(\bar{x}_1 \bar{x}_2 \bar{x}_3) &= 1 - p(x_1) - p(x_2) - p(x_3) + p(x_1 x_2) + p(x_1 x_3) + p(x_2 x_3) - p(x_1 x_2 x_3).
 \end{aligned}$$

Как наиболее эффективно решать вышеописанную систему уравнений? Мы опишем один из простейших, но вместе с тем достаточно удобных и эффективных подходов к вычислению  $p(x_n)$ , а значения остальных оценок вероятностей в таком случае можно будет просто, обойдя цикл, подсчитать по известным значениям условных вероятностей.

Итак, мы решаем систему  $A p = B$ . Мы хотим подсчитать  $p(x_n)$  по формуле Крамера, разделив определитель матрицы  $A'$ , которая получается из  $A$  заменой последнего столбца на столбец  $B$ , на определитель матрицы  $A$ . Определитель  $\det A$  мы уже вычисляли[7]:

$$\det \mathbf{A} = 1 + (-1)^n \xi_{1n} \xi_{21} \xi_{32} \dots \xi_{n,n-1}.$$

Вот как выглядит матрица  $A'$ :

$$\mathbf{A}' = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & p(x_1 | \bar{x}_n) \\ \xi_{21} & 1 & 0 & \dots & 0 & p(x_2 | \bar{x}_1) \\ 0 & \xi_{32} & 1 & \dots & 0 & p(x_3 | \bar{x}_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \xi_{n,n-1} & p(x_n | \bar{x}_{n-1}) \end{pmatrix}.$$

Изложенный способ вычислять ее определитель – это рекуррентно, один за другим, вычислять аналогичные определители меньшего порядка, раскладывая определители по последней строке. Эти формулы основаны на простой рекурсии, что позволяет легко их реализовать на любом языке программирования. В том числе, они удачны для вычисления более сложных моделей  $p(x_n)$ . Вот как это выглядит ( $\Delta_i$  определитель подматрицы порядка  $i$  (при этом положим  $\Delta_0=0$ ):

$$\Delta_0 = 0,$$

$$\Delta_1 = -\xi_{1n}\Delta_0 + p(x_1 | \bar{x}_n),$$

$$\Delta_2 = \begin{vmatrix} 1 & p(x_1 | \bar{x}_n) \\ \xi_{21} & p(x_2 | \bar{x}_1) \end{vmatrix} = -\xi_{21}\Delta_1 + p(x_2 | \bar{x}_1),$$

$$\Delta_3 = \begin{vmatrix} 1 & 0 & p(x_1 | \bar{x}_n) \\ \xi_{21} & 1 & p(x_2 | \bar{x}_1) \\ 0 & \xi_{32} & p(x_3 | \bar{x}_2) \end{vmatrix} = -\xi_{32}\Delta_2 + p(x_3 | \bar{x}_2),$$

$$\Delta_4 = \begin{vmatrix} 1 & 0 & 0 & p(x_1 | \bar{x}_n) \\ \xi_{21} & 1 & 0 & p(x_2 | \bar{x}_1) \\ 0 & \xi_{32} & 1 & p(x_3 | \bar{x}_2) \\ 0 & 0 & \xi_{43} & p(x_4 | \bar{x}_3) \end{vmatrix} = -\xi_{43}\Delta_3 + p(x_4 | \bar{x}_3),$$

$$\Delta_i = \begin{vmatrix} 1 & 0 & 0 & \dots & 0 & p(x_1 | \bar{x}_n) \\ \xi_{21} & 1 & 0 & \dots & 0 & p(x_2 | \bar{x}_1) \\ 0 & \xi_{32} & 1 & \dots & 0 & p(x_3 | \bar{x}_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & p(x_{i-1} | \bar{x}_{i-2}) \\ 0 & 0 & 0 & \dots & \xi_{i,i-1} & p(x_i | \bar{x}_{i-1}) \end{vmatrix} = -\xi_{i,i-1}\Delta_{i-1} + p(x_i | \bar{x}_{i-1}).$$

## 1.6 Ограничения подхода изолированных циклов

Подход, изложенный в разделе 1.5 позволяет эффективно согласовывать только простые циклы, где у всех вершин только один потомок.

Использование данного алгоритма для сетей, содержащих узлы с большим количеством предков и потомков сведется к тяжелым вычислениям.

Для примера рассмотрим сеть на рисунке 5. Система уравнений, описывающая условные вероятности в цикле будет не линейной, а 10 порядка:

$$Y \sim X1 * X2 * X3 \sim W \sim Z1 * Z2 * Z3 * X2 * X3 \sim W^5 \sim V1^5 * V2^5 \sim Z^{10} \sim Y^{10}$$

Таким образом при большем количестве узлов имеющих большее количество общих предков и потомков в худшем случае мы получим систему порядка

$$N = \prod N_{\text{потомков}}$$

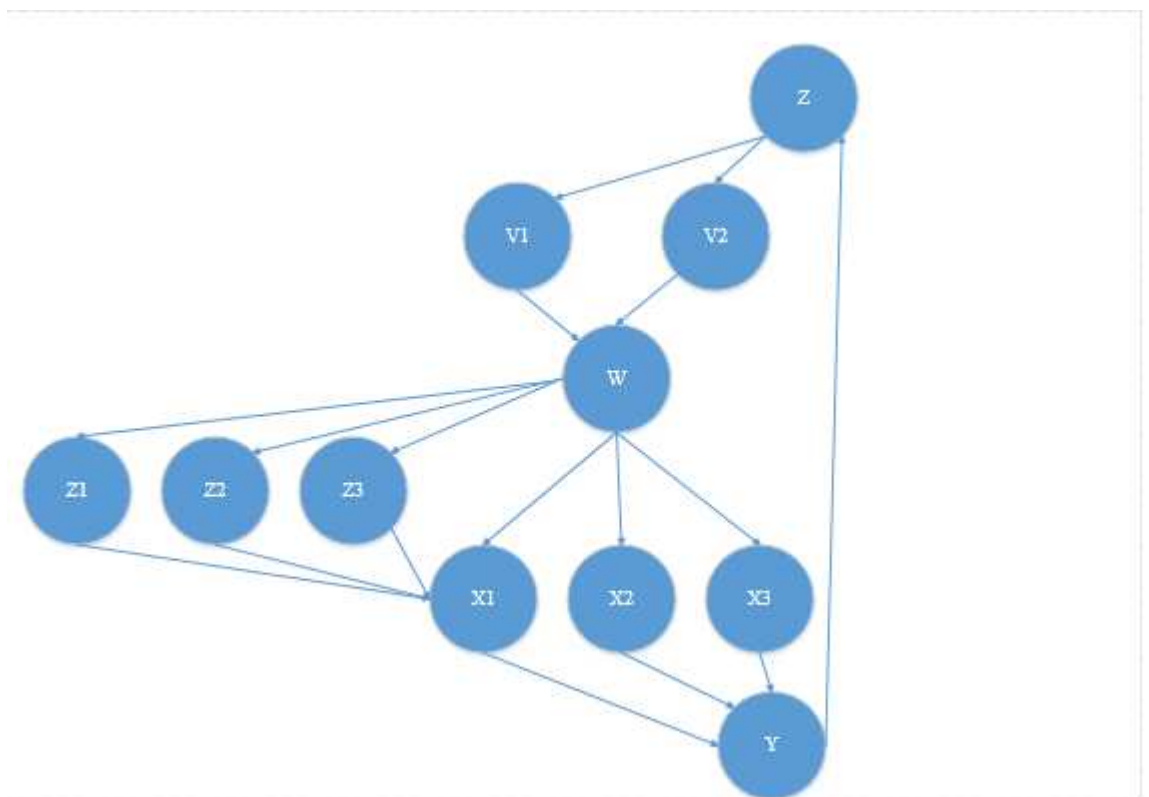


Рисунок 5 – пример байесовской сети

Вторая проблема такого подхода, это работа с сетями, имеющими более одного цикла, пример такой сети на рисунке 6.

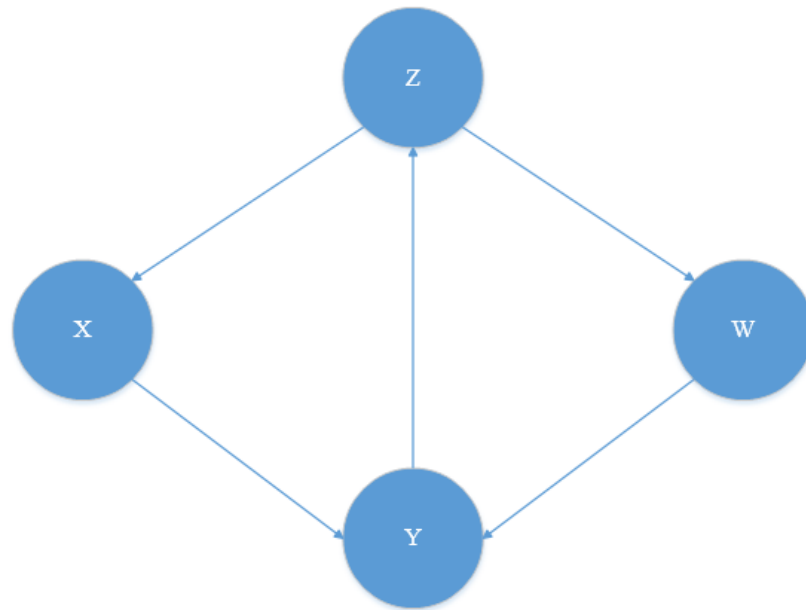


Рисунок 6 – Сеть с несколькими внутренними циклами

Получив решения для каждого цикла в отдельности, мы наверняка получим противоречие для общих вершин.

Третья проблема – это необходимость выполнять расчеты для всех комбинаций свидетельств для их согласования. Для сколько-нибудь сложной сети произойдет комбинаторный взрыв.

Четвертый вопрос, это способ согласования противоречий. Например, в алгоритме “Николенко-Тулупьева-Сироткина” [2,3], ввод несогласованных свидетельств игнорируется, что представляется искажением и потерей информации. При этом отсутствуют оценки ошибок согласования.

В качестве замечаний можно добавить отсутствие интервальных оценок. Так же не удалось найти апробации данных алгоритмов на случай известных правильных “ответов”.



## 2 Алгоритмы пропагации свидетельств и объектная модель алгоритма

### 2.1 Общие формулы для распространения свидетельств

В случае если распространение свидетельства идет от родительской вершины к дочерней используется формула (2.1) [6]:

$$P(E) = \sum_i (P(E|H_i) * P(H_i)) \quad (2.1)$$

Данный вид распространения свидетельств представлен на рисунке 7

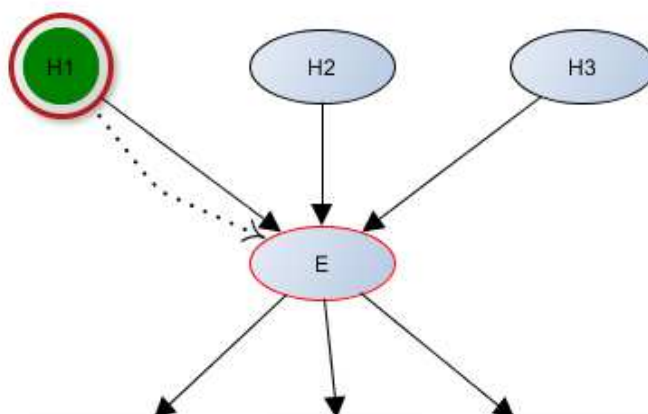


Рисунок 7 – Пример распределения свидетельств от дочерней к родительской

Если же свидетельство поступило от дочерней вершине к родительской используется формула (2.2):

$$P(H) = P(H|E) * P(E)^{**} + P(H|\bar{E}) * P(\bar{E})^{**} \quad (2.2)$$

где  $P(E)^{**}$  и  $P(\bar{E})^{**}$  это новые полные вероятности получившиеся при поступлении свидетельства из предыдущей вершины.



$$P(H|E) = \frac{\sum_{\{H_1 \dots H_k\}} (P(E|H \wedge \bar{H}_1 \dots \bar{H}_n) * P(H) * \prod_i^k P(\bar{H}_i))}{P(E)} \quad (2.3)$$

где  $P(E)$  - полная вероятность посчитанная при инициализации сети. А  $P^*$  - это вероятности родительских для них вершин, без учета нового изменения состояния  $E$  [5]. Переменные с  $\bar{H}$  обозначают, что данные события могут быть истинными  $H$ , или ложными  $\bar{H}$ .

Данный вид распределения свидетельств представлен на рисунке 8.

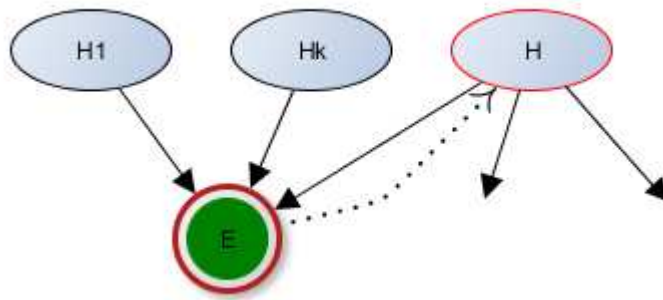


Рисунок 8 – Пример распределения свидетельств от дочерней к родительской

## 2.2 Алгоритмы пропагации свидетельств

### 2.2.1 Инициализация вершин в байесовской сети

Инициализация проходит от вершин не имеющих родительских, вниз по дереву по формуле (2.1).

### 2.2.2 Алгоритм распространения свидетельств в байесовской сети без циклов

Шаг 1: Проводим инициализацию вершин.

Шаг 2: Ввод свидетельства в вершину (начальная).

Шаг 2.1: Помечаем вершину как активную.

Шаг 3:Распространяем свидетельства от родительских вершин к дочерним.

Шаг 3.1:Пересчитываем активную вершину по формуле (2.1).

Шаг 3.2:Делаем вершину неактивной.

Шаг 3.3:Снимаем метку.

Шаг 3.4:Запоминаем родителя-источник свидетельства.

Шаг 3.5:Если изменение текущей вершины больше порога делаем активными все дочерние вершины.

Шаг 3.6: Если остались активные, возвращаемся к шагу 3.1

Шаг 4: начальная вершина активна.

Шаг 4.1:Распространяем свидетельства от дочерних вершин к родительским.

Шаг 4.2:Пересчитываем активную вершину по формуле (2.2).

Шаг 4.3:Делаем вершину неактивной.

Шаг 4.4:Снимаем метку,

Шаг 4.5:Запоминаем предка-источник свидетельства.

Шаг 4.6:Если изменение текущей вершины больше порога то Шаг 4.6.1,Шаг 4.6.2

Шаг 4.6.1:Помечаем дочерние вершины (кроме источника и вершин посчитанных на шаге 3).

Шаг 4.6.2:Делаем активными все родительские вершины.

Шаг 4.7: Если остались активные, возвращаемся к шагу 4.1

Шаг 5 :Все помеченные вершины делаем активными.

Шаг 6:Распространяем свидетельства от родительских вершин к дочерним.

Шаг 6.1:Пересчитываем активную вершину по формуле (2.1).

Шаг 6.2:Делаем вершину неактивной.

Шаг 6.3:Снимаем метку.

Шаг 6.4:Запоминаем родителя-источник свидетельства.

Шаг 6.5: Если изменение текущей вершины больше порога делаем активными все дочерние вершины, за исключением вершин посчитанных на шагах 3 и 4.

Шаг 6.6: Если остались активные, возвращаемся к шагу 6.1

### 2.2.3 Алгоритм распространения свидетельств в байесовской сети с циклами

Пример распространения свидетельства в байесовской сети с циклом, представлен на рисунке 9.

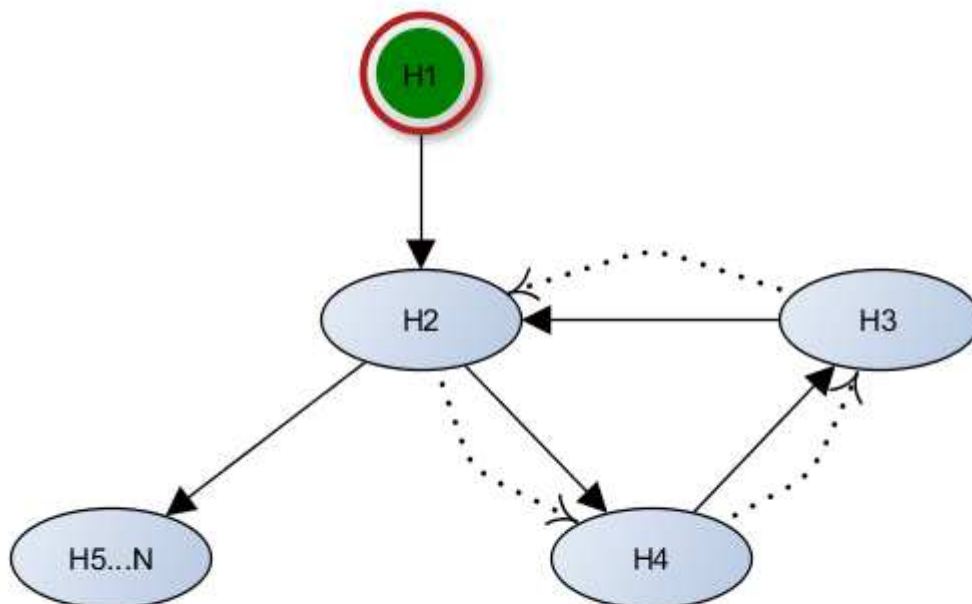


Рисунок 9 – Байесовская сеть с циклами

В нашем алгоритме, мы используем метод интервальных оценок, но для каждой вершины, мы будем хранить пороговых четыре значения вероятности, и изменение значения этих порогов, будет происходить из того, к какому интервальному промежутку отнесется текущее значение вершины. Графическое представление такого подхода, представлено на рисунке 10.

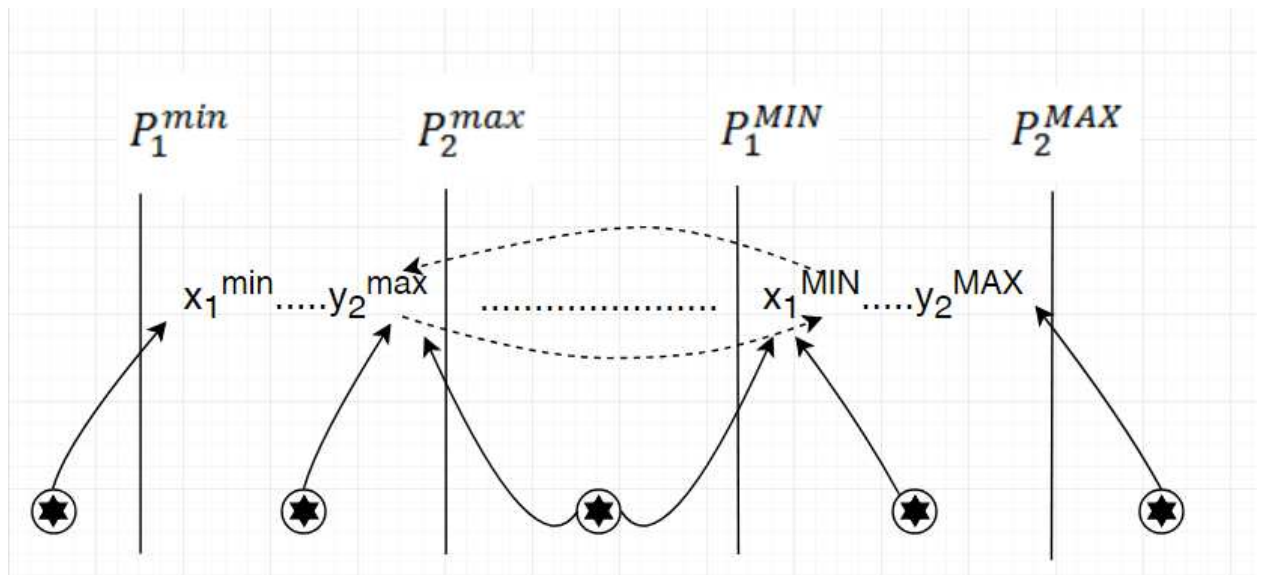


Рисунок 10 – Графическое представление метода интервальных оценок

На представленном выше рисунке мы видим графическое отображение некоторых шагов представленного ниже алгоритма, где звездой в круге, обозначено значение попавшее в какой либо интервал, пунктирной и сплошной линией обозначена замена значений.

Шаг 1: Вычисление начального значения  $X$  и переходим к шагу 2.

Шаг 2: Составление интервала исходя из имеющихся значений вероятности:  $|X_{i_1}^{min} \dots X_{i_2}^{max}| \dots |X_{i_1}^{MIN} \dots X_{i_2}^{MAX}|$  и переходим к шагу 3.

Шаг 3: Ввод следующего значения вероятности  $X^*i$ , и переход к шагу 4.

Шаг 4: Если значение  $X^*i \in |X_{i_1}^{min} \dots X_{i_2}^{max}|$ , то заменяем максимальное значение данного интервала, а предыдущее значение переносим на минимальное значение 2 интервала:  $|X_{i_1}^{min} \dots X^*i_2^{max}| \dots |X_2^{max} \dots X_2^{MAX}|$  (в случае если перенесенное значение совпадает с установленным на предыдущем шаге, останавливаем цикл и переходим к шагу 9), и возвращаемся к шагу 3 (так же необходимо учитывать в какой именно интервал попало значение и пометать его), иначе переходим к шагу 5.

Шаг 5: Если значение  $X^*i \in |X_{i_1}^{MIN} \dots X_{i_2}^{MAX}|$ , то заменяем минимальное значение данного интервала, а предыдущее значение переносим на минимальное значение 1 интервала:

$|Xi_1^{min} \dots Xi_1^{MIN} | \dots |X_1^{*MIN} \dots X_2^{MAX} |$ , (в случае если перенесенное значение совпадает с установленным на предыдущем шаге, останавливаем цикл и переходим к шагу 9) и возвращаемся к шагу 3 (так же необходимо учитывать в какой именно интервал попало значение и пометить его), иначе переходим к шагу 6.

Шаг 6: Если значение  $X^*I > |Xi_1^{MIN} \dots Xi_2^{MAX} |$ , то меняем максимальное значение данного интервала:  $|Xi_1^{MIN} \dots X^*i_2^{MAX} |$ , и возвращаемся к шагу 3

Так же необходимо учитывать в какой именно интервал попало значение и пометить его, иначе переходим к шагу 7.

Шаг 7: Если значение  $X^*I < |Xi_1^{min} \dots Xi_2^{max} |$ , то меняем минимальное значение данного интервала:  $|X^*i_1^{min} \dots Xi_2^{max} |$  и возвращаемся к шагу 3 (так же необходимо учитывать в какой именно интервал попало значение и пометить его), иначе переходим к шагу 8.

Шаг 8: Если значение  $X^*I$  находится между интервалами  $|Xi_1^{min} \dots Xi_2^{max} | > X^*I < |Xi_1^{MIN} \dots Xi_2^{MAX} |$ , заменяем максимальное значение 1 интервала и минимальное значение 2 интервала :

$|Xi_1^{min} \dots X^*i_2^{max} | > X^*I < |X^*i_1^{MIN} \dots Xi_2^{MAX} |$ , и возвращаемся к шагу 3.

Шаг 9: После остановки цикла необходимо выбрать интервал в котором мы будем искать конечный ответ, для этого проверяем в какие интервалы попадала  $Xi$ , на предыдущем шаге:

В случае, если интервал совпадает, то выбираем его же.

В случае интервалы различны, то :  $|Xi_2^{min} \dots Xi_1^{MAX} |$

И переходим к шагу 10.

Шаг 10: Ищем середину выбранного интервала, и получаем, что искомая вероятность равна  $P = \frac{X+Y}{2}$ , где  $x$ - начало интервала,  $y$ - конец интервала.

### **2.3 Объектная модель для реализации программного обеспечения, для работы с Байесовскими сетями**

На современном этапе развития информационных систем и технологий структуре и организации программного обеспечения уделяется особое внимание. При разработке структуры программного обеспечения был использован объектно-ориентированный подход [11]. В основе объектно-ориентированного подхода лежит объектная декомпозиция, т.е. представление разрабатываемого программного обеспечения в виде совокупностей объектов, в процессе взаимодействия, которых, через передачу сообщений, происходит выполнение необходимых функций [12]. Спецификация разрабатываемого программного обеспечения объединяет в себе следующие модели.

- Модель использования – представляет собой описание функциональности программного обеспечения с точки зрения пользователя.

- Концептуальная модель – модель, описывающая основные абстракции предметной области, которые обеспечивают требуемую функциональность ПО и их взаимодействие.

- Модель реализации – определяет реальную организацию программных модулей и файлов.

Для построения этих моделей был использован унифицированный язык моделирования (UML). В модели использования проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования. Вариант использования служит для описания сервисов, которые система предоставляет актеру. При этом модель не отображает, каким образом будет реализовано взаимодействие актеров с системой. Модель использования приведена на рисунке 11. В модели выделены варианты использования и 1 актер (эксперт), между которыми установлены отношения включения и расширения. Значения указанных в модели кратностей отражают общие правила получения экспертной оценки БСД [11].



Рисунок 11 – модель использования ПО

Таким образом, на основании описания данной модели представляется возможным построить концептуальную модель ПО. Концептуальная модель ПО приведена на рисунке 12.

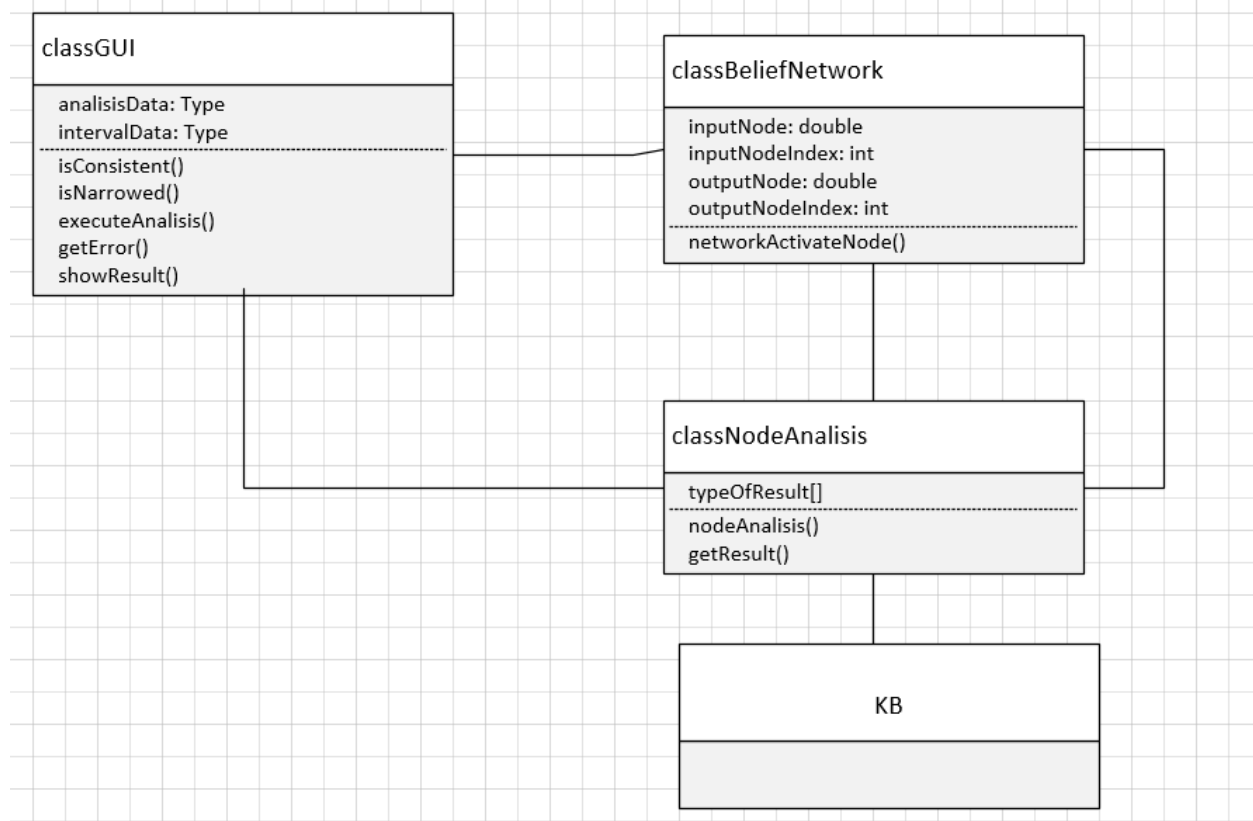


Рисунок 12 – концептуальная модель ПО

Модель представлена в виде совокупности классов. Для данной модели характерно отношение «целое – часть». В модели можно выделить класс «classGUI» как «целое», а все остальные классы являются его «частями». Задача проверки непротиворечивости и априорного вывода тесно связаны между собой и реализованы в классе classNodeAnalysis. Иерархия интерфейсов класса представлена на рисунке 13. За счет параметризации интерфейсов и унификации имен методов одинаковые методы были перенесены в интерфейс LocalInferfers. Классы quantLocalInferfers и IntervalLocalInferfers реализуют представленные выше интерфейсы. С их помощью можно:

Провести проверку непротиворечивости. После вызова метода processKP(pattern) метод isConsistent() вернет true или false;

В случае интервальных оценок знаний можно получить уточненные оценки вероятностей. Метод isNarrowed() вернет true, метод getResult() вернет результат с уточненными оценками;

Провести априорный вывод

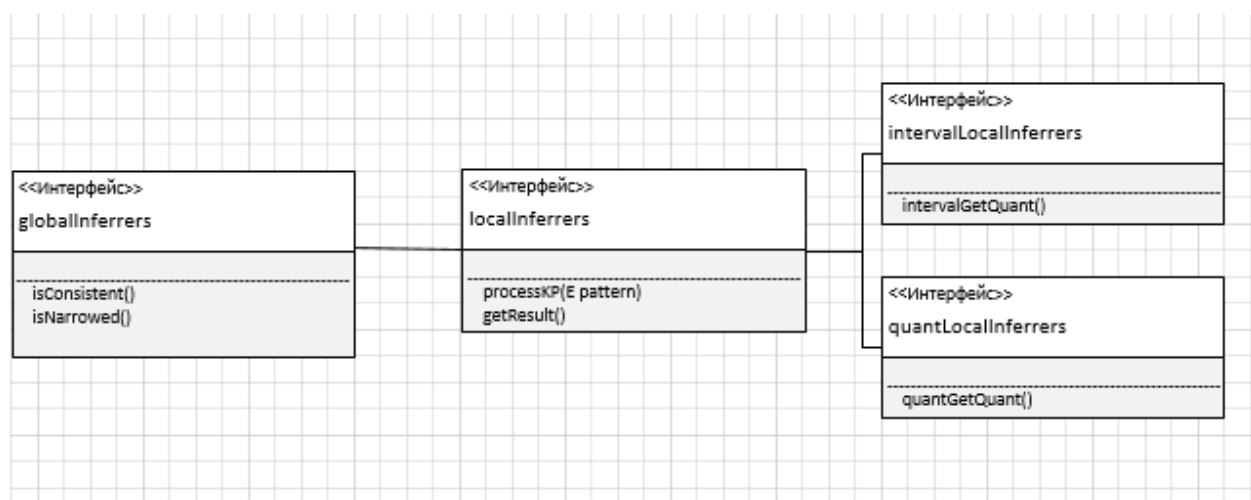


Рисунок 13 – Реализация интерфейсов класса «classNodeAnalysis»

В модели описаны особенности физического представления системы, установлены зависимости между программными компонентами. Компонент MetalNeuro.java реализует определенный набор интерфейсов для взаимосвязи с



базой данных и модулями, а также служит для общего обозначения элементов физического представления модели.

- Модуль класса `classGUI` реализует свойства и методы интерфейсной части, отвечает за проверку входящих данных и вывод результатов.

- Модуль класса `classBeliefNetwork` содержит информацию о текущем состоянии сети, ее узлах, наличии циклов и их количестве.

- Модуль класса `classNodeAnalysis` – описывает реализует свойства и методы анализа, отвечает за пересчет состояния узлов и изменение порогов интервалов при работе с сетями с циклами.

Таким образом, построенная объектная модель отображает основные абстракции предметной области, варианты использования программного обеспечения, его физическое представление, а также информационные потоки, функционирующие в программном обеспечении, необходимом для процесса анализа.

### 3 Апробация алгоритма

#### 3.1 Апробация алгоритма на сети без циклов

Для апробации алгоритма, мы используем байесовскую сеть, графическое представление которой изображено на рисунке 14.

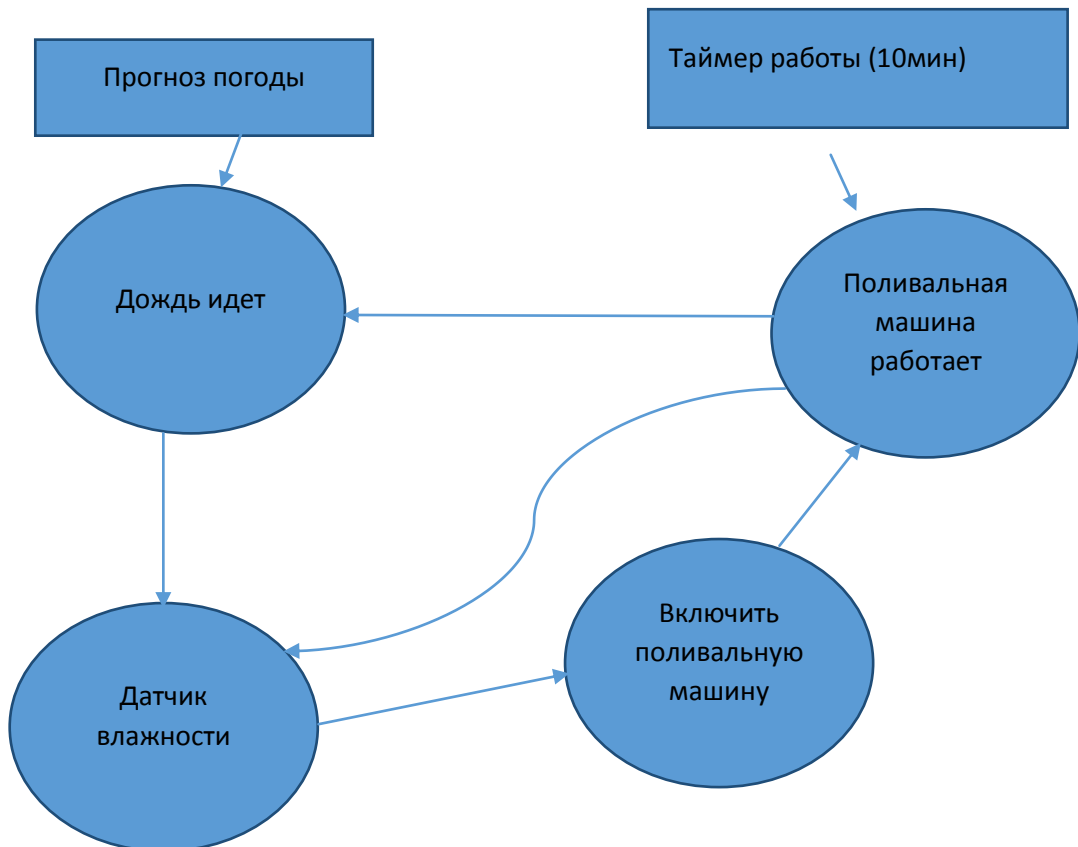


Рисунок 14 – Графическое представление Байесовской сети

В данной сети событие «Прогноз погоды» имеет априорную вероятность 0,8.

«Дождь идёт», «Датчик Влажности» «Включить поливальную машину» и «Поливальная машина работает» имеют условные вероятности 0 или 1 согласно логики рисунки(15-19):

► y		0.8
n		0.2

Рисунок 15 – Вероятность дождя согласно прогнозу погоды

Weather forec...		y	n
► y		0.7	0.3
n		0.3	0.7

Рисунок 16 – Условная вероятность события «Дождь идет»

Rain		y	n
► y		1	0
n		0	1

Рисунок 17 – Условная вероятность события «Датчик влажности»

Sensor		y	n
► y		0	1
n		1	0

Рисунок 18 – Условная вероятность события «Включить поливальную машину»

Turn on watering		y	n
► y		1	0
n		0	1

Рисунок 19 – Условная вероятность события «Поливальная машина работает»

Поскольку теория байесовских сетей доверия не предполагает работу с направленными циклами, такие циклы необходимо преобразовывать в семантически эквивалентные объекты из теории алгебраических байесовских сетей. При этом каждый граф, зависящий от результатов предыдущего. Также это необходимо, для апробации алгоритма на сети без циклов. Схема полученной декомпозиции представлена на рисунках 20-22.

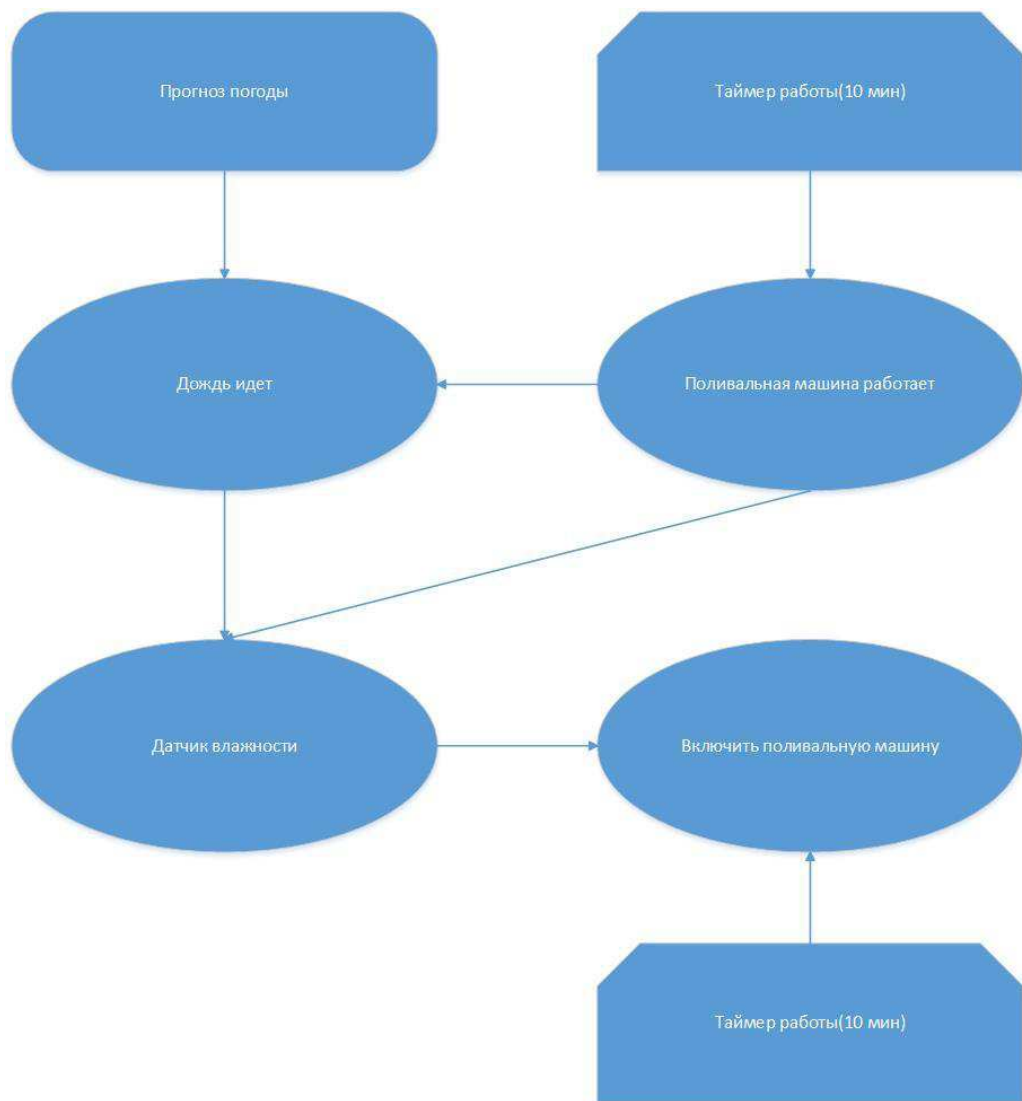


Рисунок 20 — Схема декомпозиции цикла в Байесовской сети

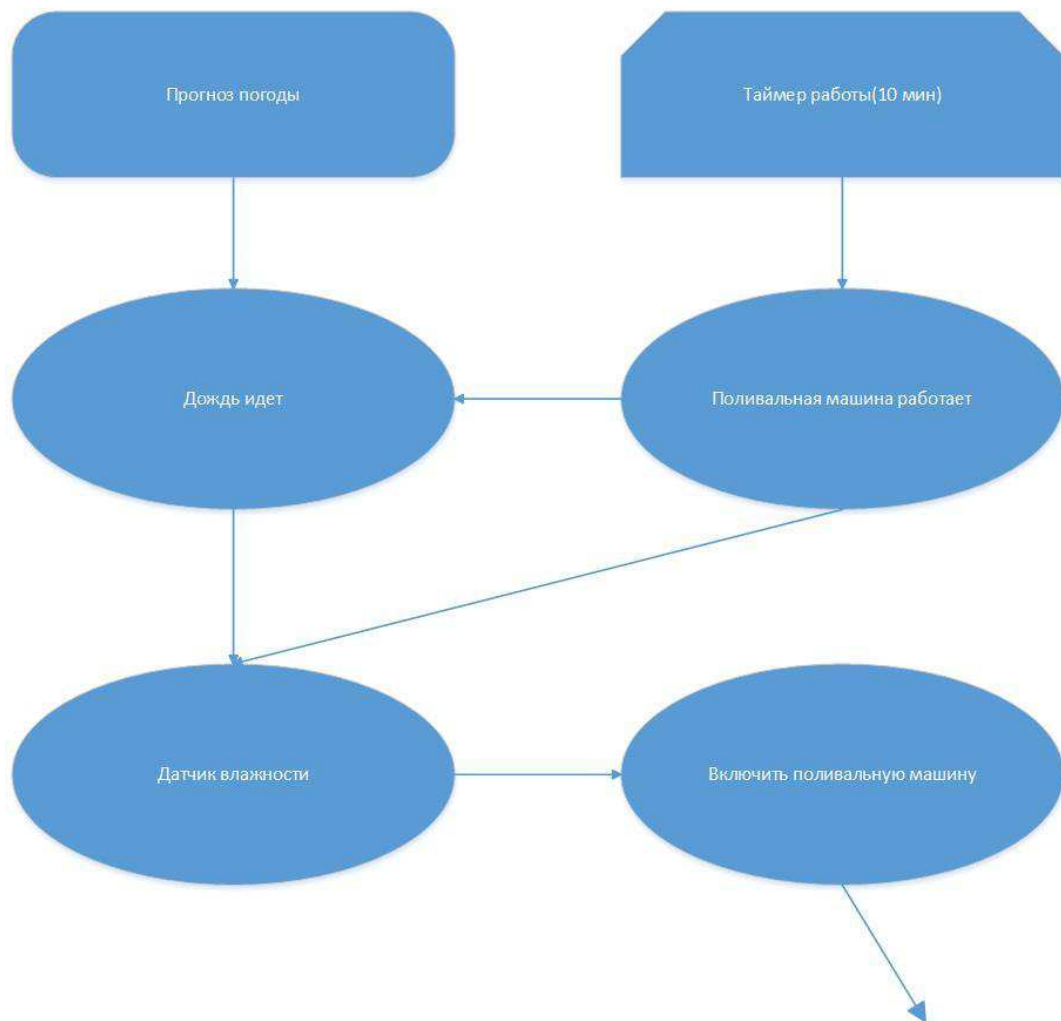


Рисунок 21 — Схема декомпозиции цикла в Байесовской сети(Продолжение)

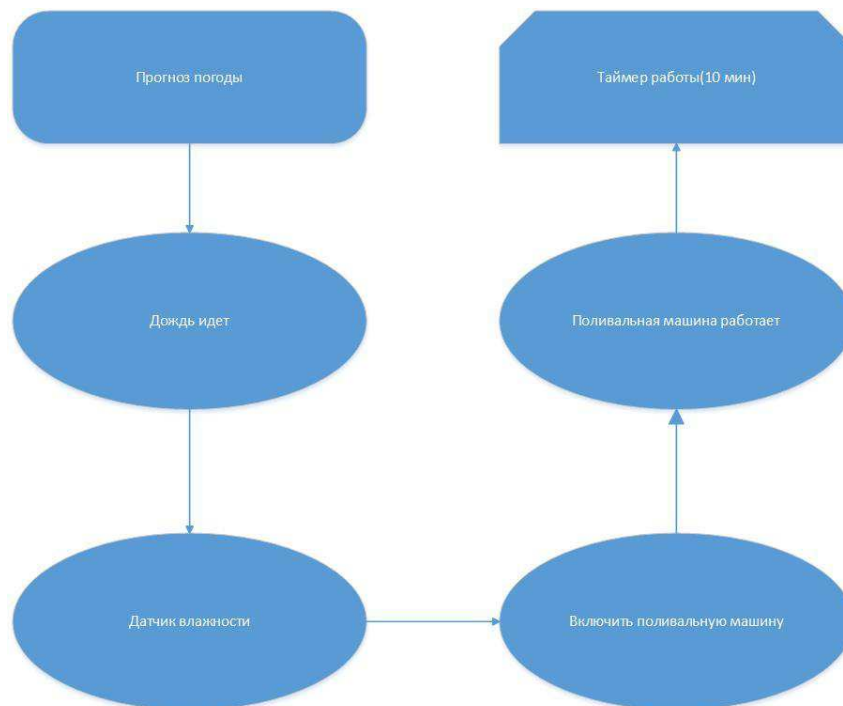


Рисунок 22 – Схема первого шага системы

Рассмотрим первый шаг нашей системы, и представим ее в виде схемы в Genie  
Рисунки (23-24)

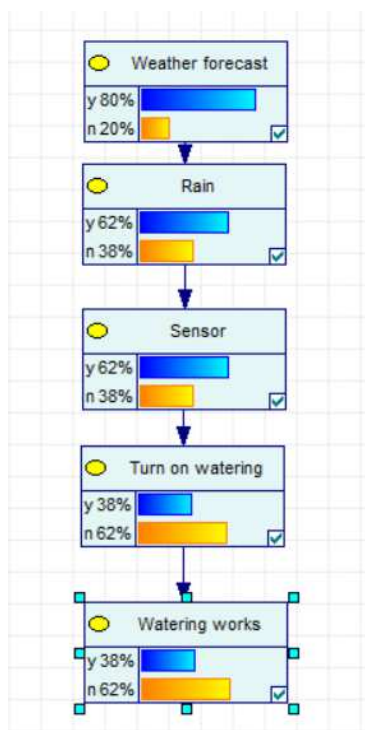


Рисунок 23 – Первый шаг системы представленный в Genie

Как видно из рисунка 23, вероятность события «Поливальная машина работает», составляет 38%. Используем полученный результат в шаге 2 нашей системы рисунок 24

Сравнение результатов апробации алгоритмов "Genie" и алгоритма разработанного в данной работе представлены в таблице 3.1

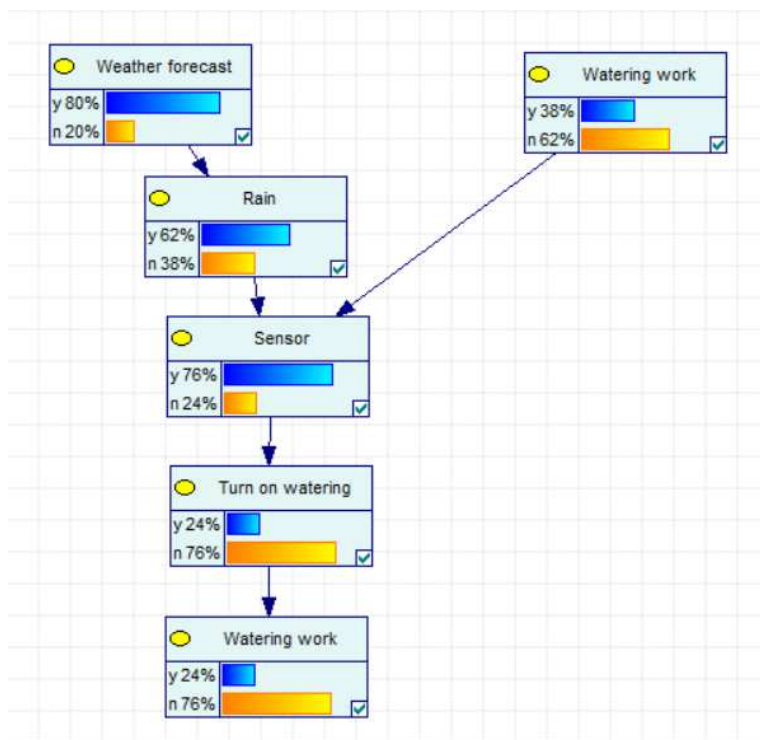


Рисунок 24– Второй шаг системы представленный в Genie

Таблица 3.1 – Сравнение результатов апробации алгоритмов

События	Вероятности полученные при инициализации первого системы		Вероятности полученные при инициализации второго шага системы при инициализации события «Watering work» = 0,38	
	алгоритм "Genie"	алгоритм пропагации	алгоритм "Genie"	алгоритм пропагации
Weather forecast	0,8	0,8	0,8	0,8
Rain	0,62	0,62	0,62	0,62

Sensor	0,62	0,62	0,76	0,76
Turn on watering	0,38	0,38	0,24	0,24
Watering work	0,38	0,38	0,24	0,24

### 3.2 Аprobация алгоритма на сети с циклами

В прошлом примере событие «Включить поливальную машину» можно удалить, так как оно связано с другими чёткой логикой.

Если исключить лишние звенья из сети, мы получим следующую сеть Байеса, с двумя направленным циклом рисунок 25, которую мы используем для аprobации алгоритма на сети с циклами.

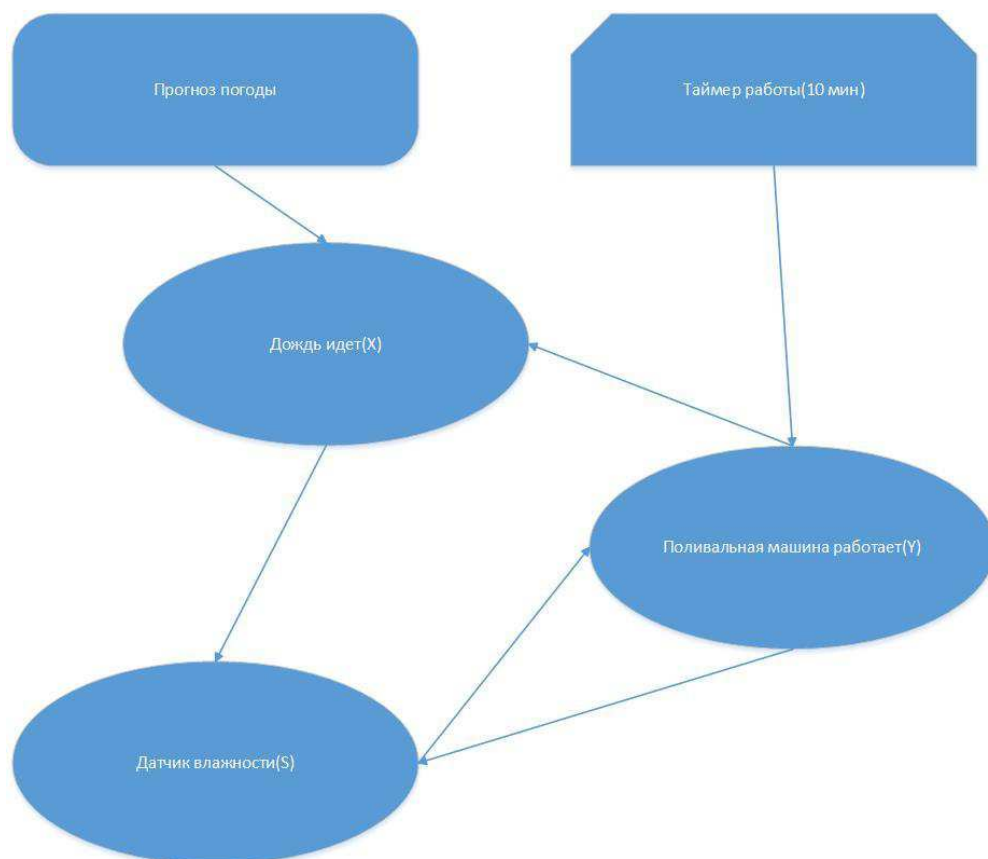


Рисунок 25 – Сеть Байеса с двумя направленными циклами



Априорные вероятности узлов сети представлены в таблицах 3.2-3.4 соответственно.

Таблица 3.2 – Априорная вероятность события «Прогноз погоды»

$P(Z)$	$P(\bar{Z})$
0,8	0,2

Таблица 3.3 – Вероятности вершины X

Условие	$P(X   \tilde{Y}\tilde{Z})$
$YZ$	0,8
$Y\bar{Z}$	0,9
$\bar{Y}Z$	0,1
$\bar{Y}\bar{Z}$	0,2

Таблица 3.4 – Вероятности вершины S

Условие	$P(S   \tilde{X}\tilde{Y})$
$XY$	1
$\bar{X}Y$	1
$X\bar{Y}$	1
$\bar{X}\bar{Y}$	0

Обработаем эти вероятности, как в случае изолированного цикла. Для этого исходя из формулы полной вероятности составим и решим систему линейных уравнений:

$$\begin{cases} P(X) = 0,7 * P(Y) + 0,12 \\ P(Y) = 1 - P(S) \\ P(S) = P(X) + P(Y) - P(X) * P(Y) \end{cases}$$

В нашем случае решением является квадратное уравнение  $0,7 * P(S)^2 - 2,36 * P(S) + 0,34 = 0$

Решив уравнение, мы получим единственное решение

$$P(S) = 0,15$$

$$P(Y) = 0,85$$

$$P(X) = 0,715$$

Теперь по формулам:

$$P(XS) = P(X|S)P(S),$$

$$P(XY) = P(Y|X)P(X),$$

$$P(YS) = P(S|Y)P(Y)$$

Получаем:

$$P(XS) = 0,71 \quad P(XY) = 0,85 \quad P(X_2X_3) = 0,15$$

Убедимся, что данное распределение непротиворечиво. Для этого проверим, что существует  $P(X_1X_2X_3)$ , удовлетворяющее следующему неравенству:

$$\begin{aligned} & \max \left\{ \begin{array}{l} 0, \\ P(X_1X_2) + P(X_1X_3) - P(X_1), \\ P(X_1X_2) + P(X_2X_3) - P(X_2), \\ P(X_1X_3) + P(X_2X_3) - P(X_3) \end{array} \right\} \leq \\ & \leq P(X_1X_2X_3) \leq \\ & \leq \min \left\{ \begin{array}{l} P(X_1X_2), \\ P(X_1X_3), \\ P(X_2X_3), \\ 1 - P(X_1) - P(X_2) - P(X_3) + P(X_1X_2) + P(X_1X_3) + P(X_2X_3) \end{array} \right\} \end{aligned}$$

Или в численном виде:

$$0,1 \leq P(X_1 X_2 X_3) \leq 0,75$$

Таким образом, мы получили, что означивание предков сети дает непротиворечивое распределение вероятностей, на элементах цикла.

### 3.2.1 Подсчет вероятностей по измененному алгоритму

При инициализации вершины  $X$  стартовые значения интервала в соответствии с шагом 1 алгоритма – отсутствуют, остальные шаги представлены в таблице 3.5.

Таблица 3.5 – Шаги алгоритма при изменении поступающего сигнала

Шаг	S	$P_1^{Min}$	$P_2^{Min}$	Изменился P	$P_1^{Min}$	$P_2^{Max}$	Изменился p
1	0,07	—	—		—	—	
2	0,1	0,07	0,07	*	0,1	0,1	*
3	0,25	0,07	0,07		0,1	0,25	*
4	0,09	0,07	0,09	*	0,09	0,25	*
5	0,14	0,07	0,14	*	0,14	0,25	*
6	0,16	0,07	0,16	*	0,16	0,25	*
7	0,16	/	/		/	/	

Так как поступивший сигнал повторился, распространение свидетельств завершено итоговые вероятности вершины равны:  $P = \frac{X+Y}{2} = 0,155$ . Что схоже с результатами, полученными при решении сети, как изолированной.

Имитация сильных противоречий данных в цикле приводит к значениям вершин в виде больших интервалов (например  $[0,1;0,8]$ ) со средним около 0,5. Подобные значения можно трактовать, как неопределенные состояния вершин, что при сильных противоречиях выглядит уместным.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы были изучены теоретические источники по байесовским сетям. Современные алгоритмы распространения свидетельств в сетях без циклов и с циклами. Изучены основные проблемы работы с сетями, содержащими циклы.

В ходе проведения исследования, были сделан вывод, о том, что для сетей со сложной структурой известные алгоритмы распространения свидетельств не пригодны.

В разработанном алгоритме распространения свидетельств в байесовских сетях с циклами несогласованность потомков не влияет на априорные вероятности их предков. Алгоритм апробирован на искусственных примерах.

В ходе работы была разработана объектная модель и реализовано программное обеспечение для работы с байесовскими сетями.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Тупалев, А.Л. Циклы в байесовских сетях, вероятностная семантика и отношения с соседними узлами / А. Л. Тупалев, С. И. Николенко, А. В. Сироткин. // Труды СПИИРАН. – 2006. – №3. – с. 240-263.
2. Jensen, F. V. Bayesian Networks and Decision Graphs / F. V. Jensen. — NY: SpringerVerlag, 2001. — 268p.
3. Probabilistic Networks and Expert Systems: tutorial / R. G. Cowell, P. Dawid, S. L. Lauritzen, D. J. Spiegelhalter.— New York: Springer-Verlag, 1999. — 205p.
4. Millán E. A Bayesian Diagnostic Algorithm for Student Modeling and its Evaluation / E. Millán, J. L. Pérez-de-la-Cruz // User Modeling and User-Adapted Interaction. — 2002. — N. 12. — P. 281–330.
5. Рыбина Г.В. Обучающие интегрированные экспертные системы: некоторые итоги и перспективы // Искусственный интеллект и принятие решений, 1/2008. – стр. 22-56.
6. Гмурман, В. Е. Теория вероятностей и математическая статистика: учебное пособие / В. Е. Гмурман. – Москва : 2003. – 479 с.
7. Лозв, М. Теория вероятностей: учебное пособие / М. Лозв; под. ред. Ю. В. Прохорова. – Москва : Издательство иностранной литературы, 1962. – 711 с.
8. Колмогоров, А. Н. Основные понятия теории вероятностей: учебное пособие / А. Н. Колмогоров. – Москва: Наука, 1974. – 412 с.
9. Reye, J. Student Modelling based on Belief Networks / J. Reye // International Journal of Artificial Intelligence in Education. — 2004. — N. 14. — P. 1–33.c.
10. Almond, R. G. Modeling diagnostic assessments with Bayesian networks / R. G. Almond, L. V. DiBello, B. Moulder, J. D. Zapata-Rivera // Journal of Educational Measurement. — 2007. — N. 44. — P. 341–359.

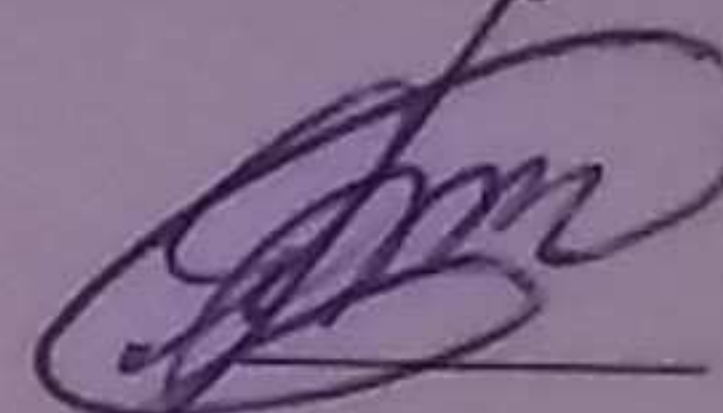
11. Комлев, Н.Ю. Объектно Ориентированное Программирование — СОЛОН-Пресс, 2014. — 298 с.

12. Якобсон И. Унифицированный процесс разработки программного обеспечения / И. Якобсон, Г. Буч, Дж. Рамбо. – СПб.: Питер, 2002. – 458 с.



Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт космических и информационных технологий  
Базовая кафедра «Интеллектуальные системы управления»

УТВЕРЖДАЮ  
Заведующий кафедрой

 Ю.Ю. Якунин

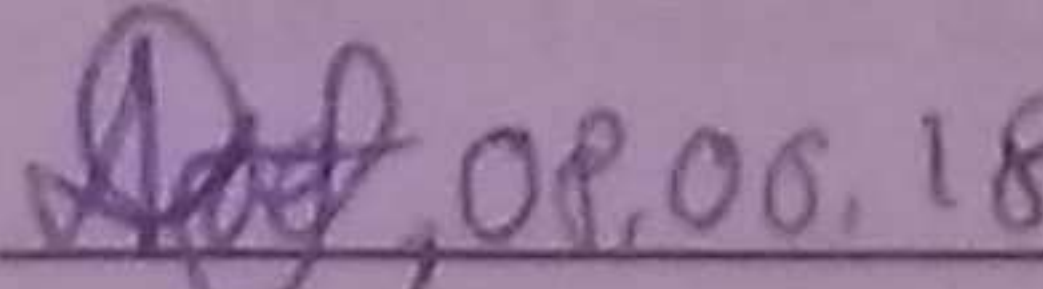
" \_\_\_ " \_\_\_\_\_ 2018 г.

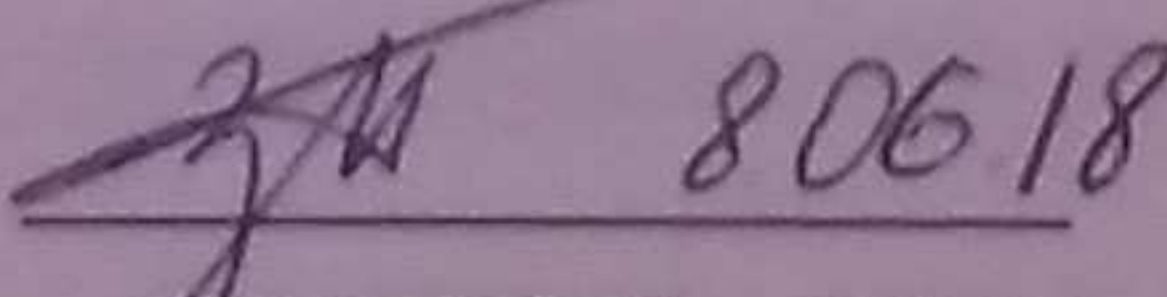
## МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

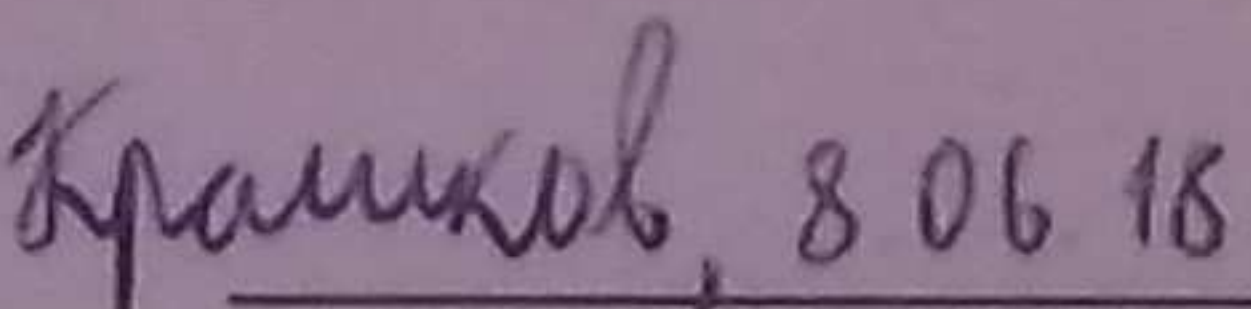
Алгоритм пропаганды свидетельств в байесовской сети с циклами

27.04.03 Системный анализ и управление

27.04.03.02 Системный анализ данных и технологий принятия решений

Научный руководитель  08.06.18 доцент, канд. техн. наук А.А. Даничев  
подпись, дата

Выпускник  8.06.18 А.В. Зыков  
подпись, дата

Рецензент  8.06.18 доцент, канд. физ-мат. наук В.А.Красиков  
подпись, дата

Красноярск 2018