Санкт-Петербургский государственный университет

Направление математика и механика
Фундаментальная информатика и информационные технологии

Пак Святослав Тхон Иль Владимирович

Алгебраические байесовские сети: представление данных, алгоритмы обработки и реинжиниринг комплекса программ (проектная работа)

Выпускная квалификационная работа

Научный руководитель:
д. ф.-м. н., доцент Тулупьев А. Л.

Рецензент:
к. ф.-.м. н., доцент Фильченков А. А.

Санкт-Петербург
2017

SAINT-PETERSBURG STATE UNIVERSITY

Fundamental Informatics and Informational Technologies

Svyatoslav Tkhon Il Pak

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

Graduation Project

Scientific supervisor:
assistant professor Alexander Tulupyev

Reviewer:
assistant professor Andrey Filchenkov

Saint-Petersburg
2017

Оглавление

Введение

Одной из актуальных проблем информатики было и остаётся отсутствие полноценного искусственного интеллекта, способного заменить человека в некоторых сложных областях. Для решения данной задачи было создано множество разных подходов ([1, 2, 3]), например, популярные в последнее время нейронные сети. Результаты этих подходов используются повсеместно: от "умных" роботов и домов до систем принятия решений в бизнесе и разработки человекоподобных машин и синтезации речи.

Большое количество систем, где необходимо применение искусственного интеллекта, характеризуется "не-факторностью" ([4, 5]), то есть неточностью или неполнотой информации. Причин на то может быть несколько (что было выделено, в частности, в [5]):

- Неполнота самой предметной области. Зачастую закономерности в сложных или только начинающих развиваться областях настолько сложно (а зачастую даже неизвестно как) зависят от других параметров , что выводить их проще на основании эвристических методов (например, стохастически);

- Неполнота сведений об объекте исследования. Даже если закономерности могут быть довольно точно выведены, не всегда точность оборудования даёт возможность измерить все необходимые параметры;

- Чрезмерная сложность вычислений. Даже при наличии полного набора данных и точно известных закономерностей, при вычислении могут использоваться эвристики как для экономии ресурсов, так и для того, чтобы иметь возможность что-либо высчитать/вывести.

Поэтому часто речь идёт не о фактах и даже не о вероятностях некоторых событий, а лишь об их оценках.

В связи с вышесказанным одним из важных этапов создания искусственного интеллекта является разработка машинного обучения. Ни одна интеллектуальная система, не способная адаптироваться к меняющемуся окружению, не может считаться полноценной. Для реализации возможности интеллектуальных систем адаптироваться к новым условиям было разработано множество подходов. Более того, большинство современных разработок в области искусственного интеллекта фокусируются как раз на том, чтобы научить системы самообучаться. Одним из методов машинного обучения являются вероятностные графические модели — частный случай экспертных систем (то есть систем, основанных на неких экспертных оценках событий, [6]), ВГМ. Объединяет эти модели то, что они представляют собой графы с заданными вероятностями некоторых событий (как пример таких графов — марковские сети: узлами являются состояния, а рёбрами — вероятности перехода между этими состояниями), что даёт широкие возможности для реализации экспертных систем ([2, 3]).

Частным примером вероятностных графических моделей являются алгебраические байесовские сети — АБС. Данная вероятностная графическая модель является аналогом и конкурентом байесовских сетей доверия (БСД) — другой ВГМ со схожим принципом.

Алгебраические байесовские сети, как и любая вероятностная графическая модель, состоят из переменных и связей между ними. Эти связи можно декомпозировать на фрагменты знаний — пары-тройки знаний о предметной области, связи между которыми может детально описать эксперт.

АБС представляют собой графы смежности, где вершинами выступают фрагменты знаний (ФЗ), а рёбра показывают, что два ФЗ имеют общий непустой подалфавит (то есть, имеют вершины с одинаковыми именами в рамках данной сети). Вслед за В. И. Городецким (который ввёл само понятие АБС), под фрагментами знаний подразумевают идеалы конъюнктов над некоторым алфавитом. Такие АБС исследованы лучше остальных и потому их можно считать “классическими”.

Однако, как было показано ([4, 5, 10]), в качестве фрагмента знаний могут выступать не только идеалы конъюнктов, но и другие наборы пропозиций. Двумя таковыми наборами пропозиций являются набор всех квантов или идеал дизъюнктов над алфавитом данного ФЗ. Такие фрагменты знаний получили название “альтернативные фрагменты знаний”.

Такие ФЗ зачастую оказываются удобнее ФЗ над идеалами конъюнктов. Так, например, вся теория локального логико-вероятностного вывода построена на множествах квантов.

Данная работа рассматривает локальный логико-вероятностный вывод (ЛВВ) и поддержание непротиворечивости в альтернативных ФЗ на теоретическом и практическом уровне. Также она является частью проектной работы по разработке комплекса программ для работы с АБС AlgBNModeller.

Постановка задачи

Целью моей работы в рамках проектной деятельности была реализация алгоритмов локального логико-вероятностного вывода в альтернативных фрагментах знаний.

В ходе анализа цели были выявлены следующие задачи:

- В виду отсутствия матриц перехода для преобразования оценок над разными носителями, необходимо было реализовать матрицу перехода от вектора оценок вероятностей дизъюнктов к верктору оценок вероятностей соответствующих квантов.

- Разработать и реализовать классы для априорного вывода в фрагментах знаний над множествами квантов.

- Разработать и реализовать классы для априорного вывода в фрагментах знаний над идеалами дизъюнктов, в том числе реализовать функцию взятия дополнения к вектору оценок вероятностей.

- Разработать и реализовать классы апостериорного вывода, пользуясь классами априорного вывода.

Используемые технологии

Как было сказано ранее, данная работа — часть большого комплекса программ, и поэтому для реализации совместной работы над проектом было использовано множество технологий и программных продуктов.

В качестве платформы разработки была выбрана Microsoft Visual Studio, так как она, не обладая меньшей (в сравнении с той же Xamarin Studio) кроссплатформенностью, обладала большим функционалом.

Основным языком разработки был выбран C#, как один из нативных языков данной платформы.

Для обеспечения совместной работы был выбран BitBucket. Он проще в использовании, чем Microsoft Team Foundation, и обладает тем же функционалом, что и Git. Отдельным плюсом является то, что он хорошо сочетается с SourceTree — ещё одним программным обеспечением, используемым в разработке.

Важной для проекта библиотекой является LP. Данная библиотека предоставляет методы для решения задач линейного программирования, широко используется в проекте и потому была использована мною при реализации локального логико-вероятностного вывода.

Не менее важной библиотекой стала библиотека для юнит-тестирования NUnit, так как необходимо было проверить корректность работы разработанных классов.

1. Теоретическая часть

Введение

Прежде чем описывать программную часть работы, следует рассказать про АБС, ФЗ, альтернативных ФЗ, о том, что такое непротиворечивость и какой она бывает.

В данном разделе будет кратко пересказана теория, изложенная в [9]. Данная монография является наиболее объемлющей на данный момент в области изучения ФЗ. Также будет активно использоваться [4], рассматривающая альтернативные ФЗ на идеалах дизъюнктов. Помимо этого используется [7], так как она рассматривает альтернативные ФЗ на полных множествах квантов. Помимо всех этих книг стоит упомянуть ([10, 11]), как источники ценной информации для этой главы. Так, ([11]), например, будет служить нашей "базой" в теории конечных графов (ТКГ).

1.1. Некоторые сведения из теории конечных графов

В данном разделе будут введены основные понятия, необходимые для понимания теории АБС.

Начнём с определения графа:

Определение* 1. *Граф *G — это пара $\langle V, E\rangle$, где V — множество вершин графа, а E — множество пар вершин из V.*

Графы бывают связными и несвязными, направленными, ненаправленными и смешанными, с петлями и без петель, мультиграфами и гиперграфами. В данной работе под графом подразумевается ненаправленный граф без петель, не являющийся мультиграфом и не являющийся гиперграфом.

Также графы могут быть нагруженными: их узлам (либо рёбрам) задаются веса, с которыми можно оперировать. В отличие от, например, логистики, здесь они используются для обозначения оценки вероятности пропозиции данного узла (в случае фрагмента знаний) либо для обозначения алфавита, над которым построен ФЗ в данном узле. Рёбра в глобальных структурах часто нагружаются пересечением весов рёбер.

Наиболее важным для алгебраических байесовских сетей является следующий класс графов:

Определение 2. Граф смежности *— это такой граф, в котором:*

- *между каждой парой узлов, содержащих общие веса, существует путь;*
- *в веса каждого из узлов пути, указанного ранее, входят все элементы, общие для начального и конечного узлов;*
- *вес одного узла графа не входит в вес любого другого узла графа полностью.*

Также важными являются понятия дерева и дерева смежности:

Определение 3. Деревом *называется связный граф, в котором нет циклов.*

Дерево смежности — это, соответственно, дерево, являющееся графом смежности.

Фрагменты знаний также могут быть представлены графически. Для этого используются графы смежности, а весами вершин выступают множества элементов алфавита ФЗ, получивших положительное означивание.

1.2. Вероятностная логика

Ещё одним незаменимым инструментом для теории АБС является вероятностная логика. Что это такое подробнее излагается, например, в [10]. Здесь, как и в разделе выше, будут приведены лишь некоторые необходимые данные.

Определение* 4. *Пространство исходов *— это тройка вида $\langle \Omega, \chi, p \rangle$, где Ω — множество элементарных исходов (или же множество элементарных событий), χ — алгебра событий (то есть алгебра подмножеств Ω), а p — вероятностная мера.*

В связи с тем, что графы в алгебраических байесовских сетях конечные, то и пространства исходов, с этими графами связанные, должны быть конечными. То есть Ω должно быть конечным множеством.

Любое подмножество X множества элементарных событий Ω называется событием. Для таких множеств определены операции дополнения, объединения (дизъюнкции) и пересечения (конъюнкции).

Не менее важным является понятие полной группы событий:

Определение* 5. *Полная группа событий *(разбиение, полный дизъюнктный набор событий) — набор событий $B = \{X_k\}_{k=1}^{k=n}$, удовлетворяющий двум условиям:*

1. $\forall X_i, X_j \in B, i \neq j \Rightarrow X_i \cap X_j = \varnothing$;
2. $\cup_{t=1}^{n} X_t = \Omega$.

χ — алгебра событий — определяется следующим образом:

Определение* 6. *Алгебра событий *χ — набор множеств, замкнутый относительно операций объединения, дополнения и пересечения. Она может не совпадать с 2^{Ω} и аксиоматически задаётся следующим образом:*

1. $\Omega \in \chi$;
2. $X \in \chi, Y \in \chi \Rightarrow X \cup Y \in \chi, X \cap Y \in \chi$;
3. $X \in \chi \Rightarrow \overline{X} \in \chi$.

Вероятность $p : \chi \to [0, 1]$ является мерой χ.

Вероятностная логика связывает математическую логику и теорию вероятностей путём оценивания вероятностей событий, которые представляют собой некоторую пропозиционную формулу над алфавитом атомарных пропозиций.

Пусть имеется алфавит $A = \{x_1, \ldots, x_n\}$ — набор пропозиционных атомарных формул, являющихся суждениями о некоторой предметной области и поддающихся оценке. Аргументным местом назовём $\widetilde{x} = \{x, \overline{x}\}$. Набор всех возможных пропозиций над A обозначим $F^0(A)$, а множество всех квантов (цепочек конъюнкций агрументных мест максимальной длины) — $Q(A) = \{\widetilde{x}_1 \ldots \widetilde{x}_n\}$.Далее, заменив в определении вероятностного пространства Ω на $F^0(A)$, получаем вероятностное пространство над идеалом конъюнктов.

1.3. Алгебраические байесовские сети. Непротиворечивость

Алгебраическая байесовская сеть — это граф смежности, в узлах которого находятся фрагменты знаний. Каждый фрагмент знаний представляет собой пропозиции с заданным над ними вектором оценок вероятностей. В случае, если эти векторы удовлетворяют необходимым условиям, АБС считается непротиворечивой.

Непротиворечивость АБС бывает нескольких видов: локальной, интернальной, экстернальной, глобальной. Подробнее об этом можно прочитать, например, в [5].

Локальная непротиворечивость равносильна непротиворечивости каждого фрагмента знаний и считается самой простой. Остальные непротиворечивости требуют работы с глобальными структурами АБС.

Данная работа рассматривает только локальный логико-вероятностный вывод и поддержание локальной непротиворечивости.

Фрагмент знаний (C,p) со скалярными оценками называется непротиворечивым, если $\exists p_{_F}$ — вероятностное распределение, такое что:

$$\forall c \in C \; p_{_F}(c) = p(c).$$

В случае фрагмента знаний (C,$\mathbf{p}$) с интевальными оценками нужно убелиться, что $\forall c \in C$ и $\epsilon \in \mathbf{p}(c)$ найдётся $p : C \leftarrow [0;1]$ такое, что $p(c) = \epsilon$ и C, p — непротиворечивый скалярный ФЗ.

Если у интервального ФЗ только часть интервала удовлетворяет вышеописанному условию, то он называется согласуемым и может быть сужен (либо расширен) до согласованного путём уменьшения (либо увелиения) интервалов.

1.4. Логико-вероятностный вывод в АБС.

Поддержание непротиворечивости достаточно важно, но, к сожалению, оно работает только с теми данными, которые уже есть в сети, и не выводит из них ничего нового.

Для того, чтобы из сети можно было выводить оценки других формул, помимо конъюнктов, существует априорный вывод (также называемый инферренцией). Он по существующей сети и заданной пропозициональной формуле выводит её оценку.

Другим важным аспектом является то, что сеть должна уметь меняться, подстраиваться под изменения данных, выражающиеся в поступлении новых свидетельств из внешних источников. Мало того, что сеть должна корректировать свои оценки исходя из поступающих свидетельств, она должна уметь оценивать их вероятности в рамках текущей сети, чтобы отсеивать некорректные.

Для этого существует ещё один вид логико-вероятностного вывода — апостериорный вывод (часто называемый пропагацией свидетельства). Данный вид ЛВВ работает с данными извне (называемыми «обуславливающими»), либо оценивая их правдоподобность, либо меняя ФЗ под поступившие данные.

1.5. Матрицы в теории АБС

Так как теория АБС — это алгебраическая теория, то и представление данной модели может быть изменено для упрощения работы с ней. Так, оценки вероятностей пропозиций в фрагменте знаний можно представить в виде вектора значений, и все виды ЛВВ (в том числе и поддержание непротиворечивости) свести к операциям с данным верктором.

Дальнейшие наши действия будем производить с ФЗ над идеалом конъюнктов (следуя [10]). Пусть алфавит, над которым строится данный ФЗ — это $X = \{x_1, \dots, x_n\}$.

Мы нумеруем конъюнкты следующим образом: в случае вхождения элемента с номером n в конъюнкцию мы добавляем к номеру конъюнкта 2^{n-1}. Таким образом, итоговый номер конъюнкта можно вычислить по следующей формуле:

$$Num(x_{i_1} \dots x_{i_j}) = \sum_{k=1}^{k=j} 2^{i_k - 1}$$

Такая нумерация однозначно задаёт номер конъюнкта из идеала конъюнктов.

Далее можно составить вектор 1×2^n, где на i-том месте будет стоять вероятность конъюнкта с номером i. Такой вектор и называется вектором оценок вероятностей.

Такой вектор, как было показано в [10], можно умножить на матрицу $I_n = \begin{pmatrix} 1 & \text{-1} \\ 0 & 1 \end{pmatrix}^{[n]}$, чтобы получить верктор оценок вероятностей соответствующих квантов. Матрица I_n, в свою очередь,может быть задана через рекурсию:

$$I_1 = \begin{pmatrix} 1 & \text{-1} \\ 0 & 1 \end{pmatrix}, I_n = I_{n-1} \otimes I_1.$$

1.6. Альтернативные фрагменты знаний, их связь с классическими ФЗ и между собой

Схожим образом можно определить и альтернативные ФЗ. Так, следуя ([4]), ФЗ на идеале дизъюнктов — это такой фрагмент знаний, у которого носителем выступает идеал дизъюнктов (задаваемый схожим с идеалом конъюнктов образом). Аналогично можно задать фрагмент знаний над множеством квантов над данным алфавитом.

1.6.1. Матрицы перехода

В связи с вышесказанным встаёт вопрос: каким образом мы можем переходить между различными носителями? Для этого были введены матрицы перехода.

Как уже было сказано ранее, часто удобнее работать с вероятностями квантов, нежели с вероятностями конъюнктов и дизъюнктов. Для того, чтобы переходить между векторами оценок вероятностей множества квантов и идеала конъюнктов существуют матрицы I_n и J_n. Для работы с дизъюнктами существует матрица $L_n = \overline{E}_n I_n$. Работать с ней несколько сложнее, чем с аналогичной матрицей для конъюнктов: сначала вектор оценок вероятностей конъюнктов P_d нужно вычесть из единичного вектора, потом умножить слева на L_n. Полученный вектор и есть искомый вектор оценок вероятностей квантов.

Общее доказательство корректности данного утверждения может быть найдено, например, в [7]. Здесь же будет приведён простейший пример:

$$P_d = \begin{pmatrix} p(\epsilon_\vee) \\ p(x_1) \end{pmatrix}, \overline{P}_d = \begin{pmatrix} p(\epsilon_\wedge) \\ p(\overline{x}_1) \end{pmatrix},$$

$$L_n \overline{P}_d = \overline{E}_n I_n \overline{P}_d = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & \text{-1} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p(\epsilon_\wedge) \\ p(\overline{x}_1) \end{pmatrix} =$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} p(x_1) \\ p(\overline{x}_1) \end{pmatrix} = \begin{pmatrix} p(\overline{x}_1) \\ p(x_1) \end{pmatrix} = P_q.$$

Также L_n может быть представлена как $\begin{pmatrix} 0 & 1 \\ 1 & \text{-1} \end{pmatrix}^{[n]}$

2. Практическая часть

Вышеизложенная теория нашла приложение в комплексе программ AlgBNModeller. Моей задачей в рамках данного проекта стала реализация локального логико-вероятностного вывода для альтернативных фрагментов знаний.

Реализация базировалась во многом на разработках коллег по проекту, в частности на существующих наработках по ФЗ на идеалах конъюнктов. Для достижения поставленной цели требовалось выполнить пять задач:

- Добавление необходимых матриц перехода.
- Разработка классов для осуществления локального априорного вывода в альтернативных ФЗ.
- Разработка классов для осуществления локального апостериорного вывода в альтернативных ФЗ.
- Написание тестов для разработанных классов.
- Документация разработанных классов локального логико-вероятностного вывода.

Далее в работе будут разобраны все пять этапов разработки частей программного комплекса с примерами разработанных классов.

2.1. Добавление матриц перехода

В ходе работы была добавлена матрица L_m, которая хранится в виде функции, возвращающей элемент по заданным индексам. Реализовано это было через операции с бинарным представлением индексов (побитовое И, побитовое ИЛИ и проч.), сам код класса вы можете видеть ниже.

```
public class DQTransformationMatrix : TransformationMatrix
{
  public DQTransformationMatrix(int order) : base(order){}
  public override int getElementValue(int i, int j)
  {
    var k = getDimension() - i - 1;
    if ((k & j) != k) return 0;
    return (Util.BitCount(j^k)&1)==1?-1:1;
  }
}
```

Как можно понять из названия, матрица переводит оценки вероятностей идеала конъюнктов в оценки вероятностей множества квантов.

2.2. Поддержка априорного вывода

Все классы для осуществления априорного вывода, а также поддержания согласованности в ФЗ, находятся в пространстве имён Inferrer. Там находятся классы как для априорного вывода в альтернативных ФЗ, так и в классических.

Априорный вывод для ФЗ на множествах квантов достаточно прост, так как всё, что требуется проверить, — это сумма оценок вероятностей.

Так выглядит априорный вывод для скалярных ФЗ на идеалах дизъюнктов:

```
public bool processKP(ScalarDisjKP_I pattern)
{
  reset();
  if (pattern == null)
    return false;
  this.pattern = pattern;
  matrix = new
 DQTransformationMatrix(pattern.GetNumberOfAtoms());
  var quants =
 matrix.multiply(Util.Invert(pattern.GetPointEstimate()));
  quant = new ScalarQuantKP(pattern.GetGlobalIndex(), quants);
  for (var i = 0; i < pattern.GetNumberOfElements(); i++)
    if (KPSupp.compare(quants[i], 0) == -1)
    {
      consistent = false;
      return true;
    }
  consistent = true;
  result = new ScalarDisjKP(pattern.GetGlobalIndex(),
 pattern.GetPointEstimate());
  checkExtended();
  return true;
}
```

Из алгоритма можно увидеть, что сначала заводится матрица преобразования необходимого размера, после чего вектор оценок вероятностей переводится в вектор оценок вероятностей множества квантов по формуле

$$P_q = L_m(\mathbf{1} - P_d).$$

В этом же методе проводится проверка согласованности и сохраняется согласованный ФЗ.

Все классы априорного вывода имеют простые для понимания названия:. Сначала идёт префикс, показывающий, на каких ФЗ работате данный класс: скалярных (Scalar) или интервальных (Interval). Потом идёт префикс, указывающий тип носителя: множество квантов обозначается Quant, а идеал дизъюнктов — Disj. Дальше идёт LocalInferrer, так как данные классы реализуют локальный априорный вывод. Так, класс ScalarDisjLocalInferrer служит для априорного вывода на скалярных фрагментах знаний на идеалах дизъюнктов.

2.3. Поддержка апостериорного вывода

Все классы для осуществления апостериорного вывода, а также поддержания согласованности в ФЗ, находятся в пространстве имён Propagator. Там находятся как классы для апостериорного вывода в альтернативных ФЗ, так и в классических.

Все пропагаторы имеют три префикса, показывающих их предназначение: для каких свидетельств он работает и на каких фрагментах знаний. Так, ImpreciseIntervalDisjunctsLocalPropagator отбрабатывает неточные свидетельства для фрагментов знаний с интервальными оценками и идеалом дизъюнктов в качестве носителя.

В пропагаторах меньше открытых методов: установка фрагмента знаний и обработка свидетельства.

2.4. Написание документации

Для каждого открытого метода и класса была написана документация, поясняющая принципы работы методов данных классов.

Документация направлена в первую очередь на людей, интегрирующих эти части проекта в свои разработки, поэтому она выполнена прямо в коде на языке разметки XML. Так описания методов и классов будут видны из среды разработки, что позволит не отвлекаться на чтение сопроводительной документации.

2.5. Тесты для разработанных классов

Для проверки работоспособности написанных классов был написан ряд тестов, разделённый на три части: тесты матриц перехода, тесты классов априорного вывода и тесты пропагаторов.

Тесты писались при помощи библиотеки NUnit и направлены в первую очередь на то, чтобы проверить корректность написанных методов. В основном NUnit использовался для проверки априорного и апостериорного выводов, а также проверки поддержания непротиворечивости.

Также есть ряд тестов для проверки корректности реализованных матриц, работающий с выводом на экран.

Заключение

В данной работе были реализованы алгоритмы локального ввода-вывода в альтернативных фрагментах знаний. Были получены следующие результаты:

- написана матрица перехода от вектора оценок вероятностей ФЗ на идеале дизъюнктов к вектору оценок вероятностей ФЗ на множестве квантов;
- написаны классы для инферренции и пропагации альтернативных фрагментов знаний.

Список литературы

[1] Stuart J. Russel, Peter Norvig, Artificial Intelligence. A Modern Approach // PE, 3rd edition, 2015. 1150 p.

[2] Cowell G. Robert, Probabilistic Networks and expert systems // Springer, Berlin, Corrected edition, 2003. 324 p.

[3] Judea Pearl, Probabilistic Reasoning in Intelligent Systems // Morgan Kaufmann, 2nd edition revised, 1988. 552 p.

[4] Александр Львович Тулупьев, Алгебраические байесовские сети: логико-вероятностная графическа модель баз фрагментов знаний с неопределённостью // дисс. на соискание учёной степени д. ф.-м. н., Санкт-Петербург, 2009. 670 с.

[5] Андрей Александрович Фильченков, Синтез графов смежности в машинном обучении глобальных структур алгебраических байесовских сетей // дисс. на соискание учёной степени к. ф.-м. н., Санкт-Петербург, 2013. 339 с.

[6] Джозеф Джарратано, Гари Райли, Экспертные системы: принципы программирования // Вильямс, 4-е издание, 2006. 1152 с.

[7] Александр Владимирович Сироткин, Алгебраические байесовские сети: вычислительная сложность алгоритмов логико-вероятностного вывода в условиях неопределённости // дисс. на соискание учёной степени к. ф.-м. н., Санкт-Петербург, 2011. 218 с.

[8] Тулупьев А. Л., Алгебраические байесовские сети: глобальный логико-вероятностный вывод в деревьях смежности: Учеб. пособие // Санкт-Петербург,ООО Издательство «Анатолия», 2007. 40 с.

[9] Тулупьев А. Л., Сироткин А. В., Николенко С. И., Байесовские сети доверия: логико-вероятностный вывод в ациклических графах

// Санкт-Петербург, издательство Санкт-Петербургского университета, 2009. 400 с.

[10] Тулупьев А. Л., Алгебраические байесовские сети: локальный логико-вероятностный вывод: Учеб. пособие // Санкт-Петербург, ООО Издательство «Анатолия», 2007. 80 с.

[11] Frank Harary, Graph Theory // Basic Books, 1st edition, 1972. 274 p.