Алгебраические байесовские сети: подход к ленивой генерации множества вторичных структур [1]

Березин А. И., студент мат.-мех. фак-та кафедры информатики СПбГУ, стажер-исследователь, лаб. ТиМПИ СПИИРАН, beraliv.spb@gmail.com

Аннотация

В настоящей работе представлено схематичное описание алгоритма ленивой генерации множества минимальных графов смежности. Приведены псевдокоды отличительных от инкрементальной реализации частей алгоритма.

Введение

Алгебраические байесовские сети (АБС) — экземпляры класса вероятностных графических моделей (ВГМ) [13, 14, 7]. С помощью таких моделей можно декомпозировать данные на отдельные фрагменты, или фрагменты знаний (ФЗ) [10, 6], что позволяет обрабатывать большие данные, тесно связанные между собой.

В структурной части [8] алгебраических байесовских сетей центральным объектом является граф смежности [4, 11], а также его частный случай — минимальный граф смежности. Такие графы хранят в себе ФЗ, а также их связи и зависимости.

На данный момент разработаны следующие виды алгоритмов множества минимальных графов смежности (ММГС):

- прямой,
- жадный,
- инкрементальный и
- декрементальный.

Первые два алгоритма проводят синтез структур с нуля, то есть не используются никакие промежуточные результаты для вычисления следующих структур при добавлении прибывшего фрагмента знания. Алгоритмы

[1] Статья содержит материалы исследований, частично поддержанных грантами РФФИ 15-01-09001-а — «Комбинированный логико-вероятностный графический подход к представлению и обработке систем знаний с неопределенностью: алгебраические байесовские сети и родственные модели».

были разработаны Фильченковым А. А. и представлены в следующих работах: [11, 12].

Последние два алгоритма, напротив, генерируют множества инкрементально в общем смысле этого слова, то есть задействуется некоторый промежуточный результат, который ускоряет вычисления. Данные алгоритмы были предложены автором данной работы и описаны в [2, 3].

Однако все выше перечисленные алгоритмы имеют общий недостаток: они вычисляют все возможные графы смежности, даже если они нам могут не понадобиться прямо в данный момент. Казалось бы, если вычисления достаточно быстры, то проще было бы сгенерировать необходимый граф в момент запроса. Данный подход и предлагает такой способ вычисления минимального графа смежности.

Таким образом, целью настоящей работы является представление схемы ленивой генерации множества минимальных графов смежности.

Определения и обозначения

Система терминов и обозначения подробно изложены в [4, 5, 9, 1]. Так как алгоритм ленивой генерации множества незначительно отличается от реализации инкрементальных алгоритмов, опустим все схожие моменты, и рассмотрим только изменения, которые произошли при вычислении жил. Для ознакомления рекомендуется для прочтения [1]. Здесь ограничимся лишь определением графа смежности, минимального графа смежности и жилы.

Пусть задан конечный алфавит символов A, а непустые множества символов (без повторов) — слова — рассматриваются как возможные значения нагрузок вершин графов и их ребер. Пусть имеется набор вершин $V = \{v_1, \ldots, v_n\}$ и нагрузки $W = \{w_1, \ldots, w_n\}$, причем W_u является нагрузкой для вершины u. Мощностью графа G будем называть число вершин в этом графе.

Назовем неориентированный граф $G = \langle V, E\rangle$ графом смежности, если он удовлетворяет следующим условиям:

1. $\forall u, v \in V : \exists$ путь P в графе $G : \forall s \in P \Rightarrow W_e = W_u \cap W_v \neq \varnothing$,
2. $\forall e = \{u, v\} \in E \Rightarrow W_e = W_u \cap W_v \neq \varnothing$,
3. $\nexists u, v \in V : W_u \subseteq W_v$, — нагрузка одной вершины графа не входит полностью в нагрузку любой другой вершины.

Граф смежности с минимальным числом ребер мы будем называть минимальным графом смежности G_{min}. Следует отметить, что такой граф в общем случае не единственный, что и порождает ряд проблем, однако это обсуждение выходит за рамки настоящей статьи.

Жила $s_{i,j}^{S}$ — граф, построенный на вершинах сужения $G \downarrow S$ и соответствующий оммажу $H_i^S : \sigma_U(s_{i,j}^S) = H_i^S$. Множество жил сепаратора S будем обозначать как $\mathrm{SinewSet}_S$

Постановка проблемы

Алгоритм ленивого синтеза множества минимальных графов смежности, как и любые ММГС алгоритмы, состоит из двух этапов: формирование МК-пары и сборка такой пары. Как отмечалось выше, основное отличие инкрементального алгоритма от ленивого происходит при генерации жил, то есть влияет на второй этап алгоритма, который мы подробно и рассмотрим.

Ниже представлен алгоритм[2] 1 вычисления всех жил, взятый из [12] и реализованный в рамках ВКР(б) [1].

Algorithm 1 SinewSet.All

Input: ComponentSet
Output: SinewSet

Homages = **PruferTrees**(SinewSet)
for each homage **in** Homages
    LLEdges = ∅
    for each edge **in** homage
        LEdges = ∅
        for each $\langle \mathrm{v}_1, \mathrm{v}_2 \rangle$ **in** ComponentSet[edge]
            LEdges $\cup =$ $\langle \mathrm{v}_1, \mathrm{v}_2 \rangle$
        LLEdges $\cup =$ LEdges
    SinewSet $\cup =$ **UAF**(LLEdges)

Алгоритм **PruferTrees** использует только 2 трудоемких операции: вычисление всевозможных кодов Прюфера и соответствующих им деревьев.

Объединяющей алгебраической сверткой, или **UAF**, набора множеств $S = \{S_i\}_{i=1,\dots,n}$, где $\forall i \; S_i = \{S_i^j\}_{j=1,\dots,n_i}$ называется

$$\mathbf{UAF}(S) = \bigotimes_{i=1}^{n} S_i, \text{ где } A \otimes B = \bigotimes_{i=1,j=1}^{m,n} A_i \cup B_j$$

[2]Обозначения LLEdges и LEdges означают список списков ребер и список ребер соответственно

Если рост **PruferTrees** и является экспоненциальным, то вычисление чисел по сравнению с формированием объектов, таких как графов, ребер и фрагментов знаний в **UAF**, является незначительным. Получается, что **UAF** является узким местом для всего алгоритма генерации МГС.

Изменение алгоритма генерации жил

Чтобы избежать подобной выше ситуации, попробуем переписать алгоритм и приведем его ниже 2.

Algorithm 2 SinewSet.Lazy

Input: ComponentSet
Output: LazySinewSet

1: Homages = **PruferTrees**(SinewSet)
2: **for each** homage **in** Homages
3:     LLEdges = $\varnothing$
4:     **for each** edge **in** homage
5:         LEdges = $\varnothing$
6:         **for each** $\langle \mathrm{v}_1, \mathrm{v}_2 \rangle$ **in** ComponentSet[edge]
7:             LEdges $\cup =$ $\langle \mathrm{v}_1, \mathrm{v}_2 \rangle$
8:         LLEdges $\cup =$ LEdges
9:     SinewSet $\cup =$ LLEdges

На строке 9 теперь формируется множество, состоящее из списка списков ребер. То есть результирующий объект увеличился на одно измерение. Для того, чтобы понимать, как это скажется на вычислении множества МГС, представим еще один алгоритм 3, получающий нужный список ребер для одного минимального графа смежности.

На строках 2 и 5 определена функция **Count**, которая помогает пропускать нужное кол-во подмножеств для определения требуемого LazyEdges.

Вспомогательная функция **SubCount** по индексу index возвращает массив чисел из соответствующего RelativeIndex, в котором и будет требуемое LazyEdges.

Таким образом, мы получаем нужные ребра МГС.

Заключение

Разработана схема алгоритма ленивого синтеза множества минимальных графов смежности, приведены псевдокоды основных измененных частей.

Algorithm 3 GetLazyEdges

Input: LazySinewSet, index
Output: LazyEdges

```
RelativeIndex = 0
RelativeCount = Count(0)
while index >= RelativeIndex
    RelativeIndex += 1
    RelativeCount += Count(RelativeIndex)
Indices = SubCount(index, RelativeIndex)
LazyEdges = ∅
for each position in Indices
    LazyEdges ∪= LazySinewSet[RelativeIndex][index][position]
```

Данные разработки дают возможнсоть к последующему проведению статистических экспериментов. Представленные результаты являются очередным шагом для дальнейших исследований множества вторичных структур в теории алгебраических байесовских сетей.

Литература

[1] Березин А. И. Алгебраические байесовские сети: система анализа и синтеза вторичной структуры. // ВКР(б) СПбГУ. — 2016. — С. 46.

[2] Березин А. И., Зотов М. А., Иванова А. В. Синтез множества минимальных графов смежности: статистическая оценка сложности инкрементального алгоритма // Всероссийская научная конференция по проблемам информатики СПИСОК. — 2016. — С. 443–452.

[3] Березин А. И., Зотов М. А., Иванова А. В. Синтез множества минимальных графов смежности: статистическая оценка сложности декрементального алгоритма // Нечеткие системы и мягкие вычисления. — 25 мая 2016. — Т. 1. С. 94—97.

[4] Опарин В. В., Тулупьев А. Л. Синтез графа смежности с минимальным числом ребер: формализация алгоритма и анализ его корректности. // Тр. СПИИРАН. — 2009. — Вып. 11. С. 142–157.

[5] Тулупьев А. Л. Алгебраические байесовские сети: глобальный логико-вероятностный вывод в деревьях смежности: Учеб. пособие. // СПб.: СПбГУ; ООО Издательство «Анатолия». — 2007. — С. 40. (Сер. Элементы мягких вычислений).

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

[7] Тулупьев А. Л., Сироткин А. В., Николенко С. И. Байесовские сети доверия. // СПб.: Изд-во СПбГУ. — 2009. — С. 400.

[8] Тулупьев А. Л., Столяров Д. М., Ментюков М. В. Представление локальной и глобальной структуры алгебраической байесовской сети в Java-приложениях // Тр. СПИИРАН. — 2007. — Т. 5. С. 71-–99.

[9] Фильченков А. А. Синтез графов смежности в машинном обучении глобальных структур алгебраических байесовских сетей // Дисс. к-та физ.-мат. н. Самара. (Самарск. гос. аэрокосм. ун-т им. ак. С.П. Королева (нац. исслед.)). — 2013. — С. 339.

[10] Фильченков А. А, Тулупьев А. Л. Связность и ацикличность первичной структуры алгебраической байесовской сети // Вестник СПбГУ. Серия 1: Математика, Механика, Астрономия. — 2013. — № 1. С. 110–118.

[11] Фильченков А. А., Тулупьев А. Л., Сироткин А. В. Минимальные графы смежности алгебраической байесовской сети: формализация основ синтеза и автоматического обучения // Нечеткие системы и мягкие вычисления. — 2011. — Т. 6. № 2. С. 145–163.

[12] Фильченков А. А., Фроленков К. В., Сироткин А. В., Тулупьев А. Л. Система алгоритмов синтеза подмножеств минимальных графов смежности // Труды СПИИРАН. — 2013. — Вып. 4 № 27.

[13] Razib R. H., Nikolaev A. Probabilistic Graphical Modeling method for inferring hydraulic conductivity maps from hydraulic head maps. // Journal of Hydrologic Engineering. — 2016. — Vol. 21. no. 2.

[14] Yodo N., Wang P. Resilience Modeling and Quantification for Engineered Systems Using Bayesian Networks. // Journal of Mechanical Design. — 2016. — Vol. 138. no. 3.