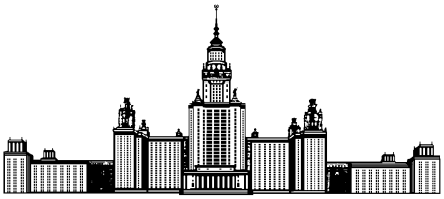


Московский государственный университет имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра математических методов прогнозирования

Лунин Дмитрий Игоревич

Методы структурного обучения байесовских сетей

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:

д.ф-м.н. О. В. Сенько

Москва, 2017

Содержание

Аннотация

В данной работе рассмотрены методы поиска структуры байесовской сети. Получена оценка дисперсии score структуры байесовской сети, рассчитанного по подвыборке данных, что позволяет получать ответ на вопрос, достаточно ли данных для обоснованного сравнения двух байесовских сетей.

Также была разработана библиотека на Python для работы с байесовскими сетями. Алгоритм поиска структуры, реализованный в ней, применен к набору медицинских данных.

1 Введение

1.1 Байесовские сети

Байесовская сеть – это направленный ациклический граф $\mathcal{G}$, каждой вершине которого соответствует случайная величина x_i. При этом выполняется следующее условие:

$$p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i \,|\, \mathrm{Pa}(x_i)) \tag{1}$$

где $\mathrm{Pa}(x_i)$ – множество родителей вершины x_i в графе $\mathcal{G}$. [1]

1.2 Задача поиска структуры

Формулировка Задача поиска структуры байесовской сети состоит в том, чтобы по данным (т.е. реализациям случайных величин $x_1, x_2, \ldots x_n$) восстановить граф $\mathcal{G}$.

Основные подходы Для решения этой задачи существуют различные подходы:

- Методы, основанные на применении статистических тестов для нахождения условных независимостей признаков в данных, и построении байесовской сети на их основе.
- Методы, основанные на введении некоторой функции $\mathrm{score}(\mathcal{G})$, оценивающей совместимость графа и данных, и дальнейшей ее оптимизации.

В данной работе рассматривается второй класс методов.

Выбор функции $\mathrm{score}(\mathcal{G})$ Одной из часто используемых функций score для байесовских сетей с дискретными переменнными является BIC [5]:

$$\mathrm{score}_{BIC}(\mathcal{G}) = l(\mathcal{G}|\mathcal{D}) - \frac{\log N}{2}\mathrm{Dim}[\mathcal{G}] \tag{2}$$

где $\mathrm{Dim}[\mathcal{G}]$ – количество параметров распределений в байесовской сети.

Функция $\mathrm{score}(\mathcal{G})$ называется разложимой, если

$$\mathrm{score}(\mathcal{G}) = \sum_{i=1}^{n} \mathrm{score}(X_i) \tag{3}$$

где score(X_i) – функция, зависящая только от значений признаков переменной, соответствующей вершине X_i и ее непосредственным родителям в графе $\mathcal{G}$. Как будет показано далее, разложимость позволяет быстро пересчитывать изменение score($\mathcal{G}$) при применении локальных операций.

NP-сложность Задача поиска оптимальной структуры является NP-сложной [6], даже в случае, когда данные были порождены байесовской сетью, количество родителей у вершин в графе ограничено числом $K \geq 3$ и в пределе количества объектов в данных $m \to \infty$ [7]

1.3 Обзор методов

Жадный поиск В этом алгоритме рассматриваются т.н. локальные операции – добавление ребра, удаление ребра и разворот ребра.

Определим score(op) операции op:

$$\text{score}(\text{op}) = \text{score}(\text{op}(\mathcal{G})) - \text{score}(\mathcal{G}) \tag{4}$$

где op($\mathcal{G}$) – граф, полученный применением операции op к графу $\mathcal{G}$.

Алгоритм жадного поиска работает следующим образом. В начале задан некоторый начальный граф G_0. На каждом шаге алгоритма к текущему графу применяется операция, максимально увеличивающая его score, то есть операция с максимальным score(op). Когда ни одна из доступных операций не увеличивает score графа, алгоритм завершает работу.

В случае разложмой функции score($\mathcal{G}$), можно эффективно вычислять значения score(op). Заметим, что для большинства операций score(op) не изменяется. Пересчитывать необходимо лишь операции, связанные с вершинами затронутыми примененной операцией.

GES Алгоритм GES (Greedy Equivalence Search) [8] основан на т.н. предположении Мика.

Предположение Мика 1. *Пусть $\mathcal{G}$ и $\mathcal{H}$ – графы, такие что $\mathcal{G}$ является I-отображением $\mathcal{H}$. Пусть r – количество ребер в $\mathcal{H}$, которые имеют противоположную ориентацию в $\mathcal{G}$, а m – количество ребер в $\mathcal{H}$, которые отсутствуют*

Algorithm 1 Алгоритм жадного поиска

procedure GREEDYLOCALSEARCH($\mathcal{G}_0, \text{score}(\cdot), \text{ops}(\cdot)$)
  $\mathcal{G} \leftarrow \mathcal{G}_0$
  while $\exists o \in \text{ops}(\mathcal{G}) : \text{score}(o) > 0$ **do**
    $o \leftarrow \underset{o \in \text{ops}(\mathcal{G})}{\text{argmax}}\, \text{score}(o)$
    $\mathcal{G} \leftarrow o(\mathcal{G})$
  end while
  return $\mathcal{G}$
end procedure

в $\mathcal{G}$. Тогда существует последовательность из не более чем $r + 2m$ добавлений и разворотов ребер в $\mathcal{G}$, обладающих следующими свойствами:

- *Каждый разворот ребра не выводит граф из его класса эквивалентности*
- *После каждого разворота или добавления граф $\mathcal{G}$ остается ациклическим и является I-отображением $\mathcal{H}$*
- *После всех разворотов и добавлений ребер $\mathcal{G} = \mathcal{H}$*

Используя этот результат, можно построить алгоритм жадного поиска в пространстве классов I-эквивалентности графов. Этот алгоритм состоит из двух последовательных фаз, отличающихся применяемыми к текущему классу эквивалентности операциями.

Алгоритм GES интересен тем, что если данные были сгенерированы байесовской сетью, а также при выполнении некоторых условий, накладываемых на score, в пределе размера выборки $m \to \infty$ он сходится к оптимальной структуре байесовской сети. На практике эти условия как правило не выполняются, и алгоритм GES работает чуть хуже жадного поиска

2 Оптимизации жадного поиска

Рассмотрим некоторые способы улучшить работу алгоритма жадного поиска.

Использование кучи На каждом шаге алгоритма необходимо определять операцию с наибольшим значением *score*. При этом на каждом шаге изменяются значения *score* не всех, а только $3(n-1)$ операций. Это позволяет использовать для поиска наилучшей операции использовать кучу, затрачивая таким образом $\mathcal{O}(n \log n)$ действий, а не $\mathcal{O}(n^2)$ как при наивной реализации.

Возмущение данных (data perturbation) В процессе работы алгоритма, может возникнуть ситуация, когда у всех графов, отличающихся от текущего на одну операцию такой же (или хуже) score, не являющийся при этом глобальным максимумом – т.н. "плато". Так происходит из-за существования больших классов эквивалентности байесовских сетей. Одним из способов борьбы с этим является возмущение данных – если сделать ресемплинг выборки, то значения score изменятся слабо, но структура классов эквивалентности будет нарушена, что позволит выйти из "плато"при продолжении работы жадного алгоритма.

Оценка score **по части данных** В данной работе предлагается новый способ ускорения работы жадного алгоритма, основанный на использовании лишь части выборки для оценки score. В процессе работы жадного алгоритма, только небольшая часть операций приводит к большому увеличению score. Предлагаемый метод состоит в том, чтобы точно вычислять score только для таких операций, а для остальных оценивать его по небольшой части данных.

Возникает вопрос, как определить по небольшой части данных, какие операции не являются перспективными, при этом по возможности не пропустив операцию с большим score. Для этого был разработан способ оценки дисперсии величины score. Он позволяет построить доверительные интервалы для истинного значения score по небольшой подвыборке. Далее используя их можно отсеять те операции, которые с заданной вероятностью не являются оптимальными.

3 Оценка дисперсии score

3.1 Вывод оценок

Будем рассматривать оценки параметров дискретного распределения в вершине X_i по данным как случайную величину: если данные – это набор реализаций случайных величин $X_1, X_2, \ldots, X_n$, то выборочные оценки p_i, как функции от случайных величин, также являются случайными величинами.

В таком случае, score, оцененный по выборке также можно рассматривать как случайную величину.

В этом разделе будет показано, как оценить дисперсию score. У данного результата могут быть следующие применения:

- Дисперсию можно использовать для оценки доверительного интервала, в который попадает score, посчитанный по всей выборке. Как обсуждалось в предыдущем разделе, с его помощью можно отбросить часть рассматриваемых на шаге жадного алгоритма операций. Однако, полученный способ вычисления дисперсии достаточно затратен, поэтому данный подход оправдан только в случае очень больших размеров данных.

- Вычисленную дисперсию можно использовать для проверки, является ли количество данных достаточным, чтобы сделать обоснованный выбор между двумя структурами байесовской сети. Если их доверительные интервалы score пересекаются, значит для обоснованного сравнения необходимо больше данных.

Многие варианты функции score, включая BIC, основаны на вычислении правдоподобия $\log p(\mathcal{G}|X)$. В случае байесовской сети с дискретными переменными, оно может быть выражено с помощью взаимной информации:

$$\log p(\mathcal{G}|X) = N \sum_{i=1}^{M} I(X_i; Pa(X_i)) - N \sum_{i=1}^{M} H(X_i) \tag{5}$$

Взаимная информация определяется как

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \, \log \frac{p(x,y)}{p(x)\, p(y)} \, dx \, dy \tag{6}$$

и может быть выражена через энтропию

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{7}$$

Приведем вспомогательные результаты, позволяющие построить оценку дисперсии $I(X;Y)$.

Теорема 1. *Если* $p_1, p_2, \ldots, p_m \sim Dirichlet(\alpha_1, \alpha_2, \ldots, \alpha_n)$, $x = p_i$, $\alpha_x = \alpha_i$, $y = p_j$, $\alpha_y = \alpha_j$ *и* k_x, k_y, m_x, m_y *являются произвольными положительными константами, то*

$$\mathbb{E}(x^{k_x} y^{k_y} \log^{m_x} x \log^{m_y} y) = \\ \frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} \sum_{i=0}^{m_y} C_{m_y}^i \frac{\partial^{m_y - i} B(\alpha_y + k_y, \alpha_z)}{(\partial \alpha_y)^{m_y - i}} \frac{\partial^{m_x + i}}{(\partial \alpha_x)^{m_x} (\partial \alpha_y)^i} B(\alpha_x + k_x, \alpha_y + \alpha_z + k_y) \tag{8}$$

где $\alpha_z = \alpha_0 - \alpha_x - \alpha_y$

Теорема 2. *Если* $x \sim Beta(\alpha_x, \alpha_y)$, $y = 1 - x$ *и* k_x, k_y, m_x, m_y *являются произвольными положительными константами, то*

$$\mathbb{E}(x^{k_x} y^{k_y} \log^{m_x} x \log^{m_y} y) = \frac{1}{B(\alpha_x, \alpha_y)} \frac{\partial^{m_x + m_y} B(\alpha_x + k_x, \alpha_y + k_y)}{(\partial \alpha_x)^{m_x} (\partial \alpha_y)^{m_y}} \tag{9}$$

Теорема 3. *Если* $p_1, p_2, \ldots, p_m \sim Dirichlet(\alpha_1, \alpha_2, \ldots, \alpha_n)$, $x = p_i$, $\alpha_x = \alpha_i$, $y = p_j$, $\alpha_y = \alpha_j$, *то*

$$\mathbb{E}(x^{k_x}\, y^{k_y} \log^{m_x} x\, \log^{m_y} y\, \log(x+y)) = \\ - \frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} \sum_{n=1}^{\infty} \frac{1}{n} \sum_{i=0}^{m_y} C_{m_y}^i \frac{\partial^{m_y - i} B(\alpha_y + k_y, \alpha_z + n)}{(\partial \alpha_y)^{m_y - i}} \cdot \\ \cdot \frac{\partial^{m_x + i}}{(\partial \alpha_x)^{m_x} (\partial \alpha_y)^i} B(\alpha_x + k_x, \alpha_y + \alpha_z + n + k_y)$$

Найдем дисперсию энтропии, оцененной по выборке:

$$H(p_1, p_2, \ldots, p_m) = \sum_{i=1}^{m} p_i \log p_i \tag{10}$$

$$p_1, p_2, \ldots, p_m \sim \text{Dirichlet}(\alpha_1 + n_1, \ldots, \alpha_m + n_m) \tag{11}$$

$$\mathbb{D}(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n} \mathbb{D}X_i + 2 \sum_{i=1}^{n} \sum_{j=1, j<i}^{n} \text{cov}(X_i, X_j) \tag{12}$$

$$\mathbb{D}(\sum_{i=1}^{n} p_i \log p_i) = \sum_{i=1}^{n} \mathbb{D}(p_i \log p_i) + 2\sum_{i=1}^{n} \sum_{j=1, j<i}^{n} \operatorname{cov}(p_i \log p_i, p_j \log p_j) \tag{13}$$

$$\mathbb{D}X = \mathbb{E}(X - \mathbb{E}X)^2 = \mathbb{E}(X^2 - 2X\mathbb{E}X - (\mathbb{E}X)^2) = \mathbb{E}(X^2) - (\mathbb{E}X)^2 \tag{14}$$

$$\operatorname{cov}(X, Y) = \mathbb{E}(X - \mathbb{E}X)(Y - \mathbb{E}Y) = \mathbb{E}(XY - X\mathbb{E}Y - Y\mathbb{E}X + \mathbb{E}X\mathbb{E}Y) = \mathbb{E}XY - \mathbb{E}X\mathbb{E}Y \tag{15}$$

$$\mathbb{D}(p_i \log p_i) = \mathbb{E}(p_i^2 \log^2 p_i) - (\mathbb{E}\, p_i \log p_i)^2 \tag{16}$$

$$\operatorname{cov}(p_i \log p_i, p_j \log p_j) = \mathbb{E}(p_i p_j \log p_i \log p_j) - (\mathbb{E}\, p_i \log p_i)(\mathbb{E}\, p_j \log p_j) \tag{17}$$

Значения $\mathbb{E}(p_i^2 \log^2 p_i)$, $\mathbb{E}\, p_i \log p_i$, $\mathbb{E}(p_i p_j \log p_i \log p_j)$, $\mathbb{E}\, p_i \log p_i$ и $\mathbb{E}\, p_i \log p_i$ могут быть посчитаны с помощью приведенных выше теорем. Таким образом, мы получили способ посчитать дисперсию энтропии, оцененной по выборке.

Перейдем к задаче оценки дисперсии взаимной информации.

$$MI(X, Pa(X)) = H(X) + H(Pa(X)) - H(X, Pa(X)) \tag{18}$$

$$\mathbb{D}(MI(X, Pa(X))) \approx \mathbb{D}(H(Pa(X)) - H(X, Pa(X))) \tag{19}$$

$$H(Pa(X)) - H(X, Pa(X)) = -\sum_{j} (\sum_{i} p_{ij}) \log(\sum_{i} p_{ij}) + \sum_{i,j} p_{ij} \log p_{ij} \tag{20}$$

$$\mathbb{D}(H(Pa(X)) - H(X, Pa(X))) = \mathbb{D}H(Pa(X)) + \mathbb{D}H(X, Pa(X)) - 2\operatorname{cov}(H(Pa(X)), H(X, Pa(X))) \tag{21}$$

$$\operatorname{cov}(H(Pa(X)), H(X, Pa(X))) = \operatorname{cov}(\sum_{j} (\sum_{i} p_{ij}) \log(\sum_{i} p_{ij}), \sum_{i,j} p_{ij} \log p_{ij}) = \sum_{a} \sum_{b,c} \operatorname{cov}((\sum_{i} p_{ia}) \log(\sum_{i} p_{ia}), p_{bc} \log p_{bc}) \tag{22}$$

$$\operatorname{cov}((\sum_{i} p_{ia}) \log(\sum_{i} p_{ia}), p_{bc} \log p_{bc}) = \mathbb{E}((\sum_{i} p_{ia}) \log(\sum_{i} p_{ia})\, p_{bc} \log p_{bc}) - \mathbb{E}((\sum_{i} p_{ia}) \log(\sum_{i} p_{ia}))\mathbb{E}(p_{bc} \log p_{bc}) \tag{23}$$

Если $a = c$,

$$\mathbb{E}((\sum_i p_{ia}) \log(\sum_i p_{ia})\, p_{ba} \log p_{ba}) = \mathbb{E}((p_{ba} + \sum_{i \neq b} p_{ia}) \log(p_{ba} + \sum_{i \neq b} p_{ia})\, p_{ba} \log p_{ba}) =$$
$$\mathbb{E}(p_{ba}^2 \log(p_{ba} + \sum_{i \neq b} p_{ia}) \log p_{ba}) + \mathbb{E}((\sum_{i \neq b} p_{ia}) \log(p_{ba} + \sum_{i \neq b} p_{ia})\, p_{ba} \log p_{ba}) \quad (24)$$

Эту величину можно вычислить применив теорему 3. В противном случае (если $a \neq c$) необходимо применить теорему 1 или 2.

3.2 Эксперименты

Была проведена экспериментальная оценка скорости сходимости ряда из Теоремы 3, на вычислении которого основаны полученные оценки для дисперсии *score*. Для этого значение матожидания $\mathbb{E}(x^{k_x}\, y^{k_y} \log^{m_x} x \, \log^{m_y} y \, \log(x+y))$ было оценено с помощью сэмплирования с большим количеством сэмплов (1000000).

В Теореме 3 значение $\mathbb{E}(x^{k_x}\, y^{k_y} \log^{m_x} x \, \log^{m_y} y \, \log(x+y))$ получено в форме бесконечной суммы ряда. На практике оно оценивается конечным числом слагаемых. Экспериментально была оценена скорость сходимости частичных сумм в зависимости от значений параметров α_x, α_y и α_z. Полученные результаты представлены на графиках (Рис. 1).

На графиках видно, что скорость сходимости зависит от относительной величины α_z по сравнению с α_x и α_y. При этом, когда эти значения близки, достаточно всего нескольких членов ряда. Графики построены для случая $m_x = k_x = m_y = k_y = 1$, однако при других значениях этих параметров результаты аналогичные. В применении к задаче оценки дисперсии, большие значения α_z по сравнению с α_x, α_y как правило возникают, когда у условных распределений большое число параметров.

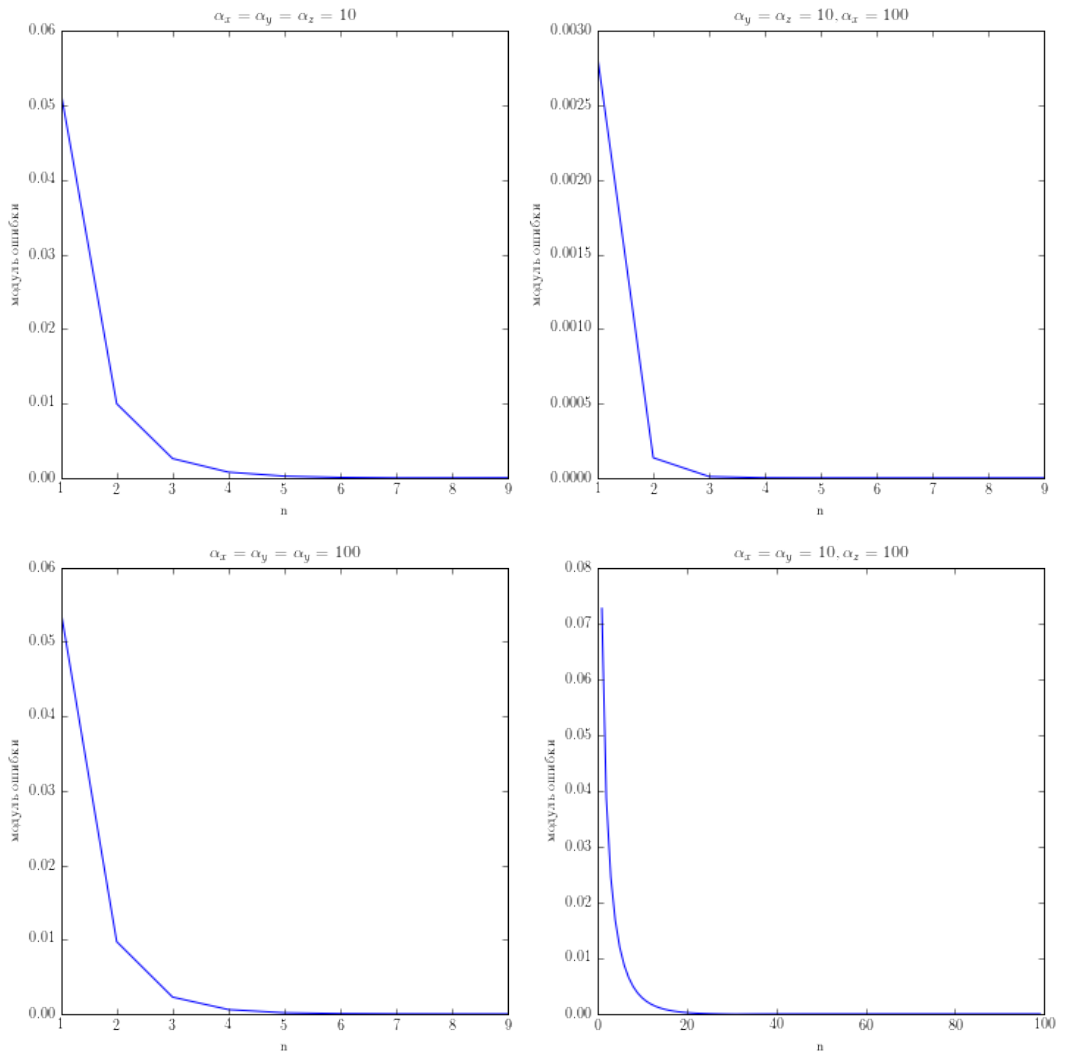


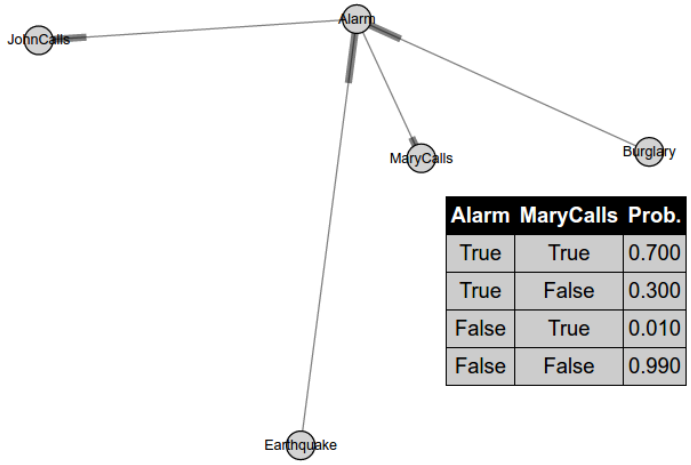
Рис. 1: Модуль ошибки оценки матожидания $\mathbb{E}(x^{k_x}\, y^{k_y} \log^{m_x} x \, \log^{m_y} y \, \log(x+y))$ в зависимости от количества вычисленных слагаемых ряда. $m_x = k_x = m_y = k_y = 1$

4 Библиотека для работы с байесовскими сетями

В ходе работы была реализована библиотека на Python, включающая в себя:

- Алгоритмы точного вывода в байесовских сетях.
- Оптимизированный (см. предыдущий раздел) алгоритм жадного поиска оптимальной структуры байесовской сети.
- Эффективное сэпмлирование переменных за счет реализации критичных частей кода на C++.

```
In [12]: dgm.draw() # you can move the cursor on a node to see it's CPD
Out[12]:
```



Alarm	MaryCalls	Prob.
True	True	0.700
True	False	0.300
False	True	0.010
False	False	0.990

Рис. 2: Визуализация байесовской сети Alarm с помощью библиотеки pygraphmodels

- Возможность интерактивной визуализации байесовской сети (в том числе встраиваемую в ipython notebook).

5 Применение алгоритма поиска структуры к медицинским данным

Результат работы алгоритма приведен на Рис. 3. При этом показаны только те переменные, для которых в найденном графе нашлось хотя бы одно инцидентное ребро. Приведем описание этих переменных, а так же интерпретацию найденных связей в терминах предметной области:

- ОНМК – острое нарушение мозгового кровообращения (инсульт)
- ТА (ТИА) – транзиторная ишемическая атака

Связь ОНМК и ТИА в данном случае является следствием того, как устроена процедура сбора данных. ОНМК и ТИА – болезни с очень похожими симптомами. Пациенты с ними попадали на обследование, далее им диагностировали либо одно,

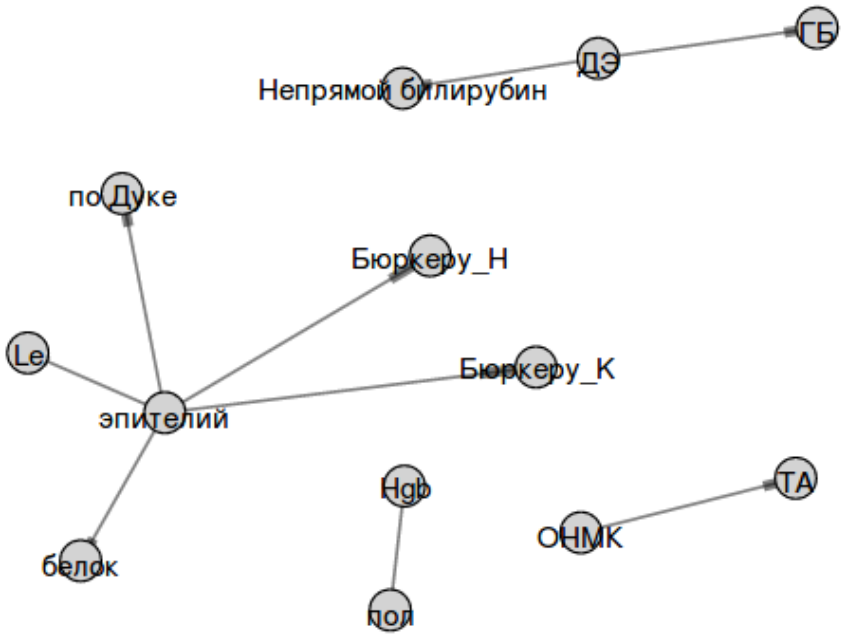


Рис. 3: Результат работы алгоритма структурного обучения

либо другое заболевание. Поэтому в данных эти переменные являются взаимоисключающими.

- пол пациента
- Hgb – содержание гемоглобина в крови

Средний уровень гемоглобина в крови различается в зависимости от пола, что объясняет найденную зависимость.

- Непрямой билирубин – связан со скоростью разрушения эритроцитов
- ГБ – Гипертоническая болезнь
- ДЭ – Дисциркуляторная энцефалопатия

Дисциркуляторная энцефалопатия – тяжелое сосудистое заболевание головного мозга. Может возникать как осложнение гипертонической болезни, т.н. гипертоническая дисциркуляторная энцефалопатия. Это объясняет связь ГБ и ДЭ. Также при дисциркуляторной энцефалопатии изменяется размер и форма эритроцитов, что может приводить к изменению показателя непрямого билирубина.

- по Дуке, по Бюркеру – показатели свертываемости крови
- Le – содержание лейкоцитов в моче
- эпителий – содержание эпителия в моче

По мнению экспертов, связь между этими переменными может быть обусловлена тем, что они все изменяются при наличии почечных заболеваний.

6 Заключение

В ходе работы были достигнуты следующие результаты:

- Получен способ оценки дисперсии score, что позволяет определять, достаточно ли данных для выбора между заданными структурами байесовской сети.
- Разработана библиотека pygraphmodels для работы с байесовскими сетями, и, в частности, для решения задачи структурного обучения.
- Алгоритм поиска структуры, реализованный в ней, применен к реальным медицинским данным, а полученные результаты проинтерпретированы.

Список литературы

[1] Daphne Koller, Nir Friedman *Probabilistic Graphical Models: Principles and Techniques*, The MIT Press, Cambridge, Massachusetts, 2009.

[2] David Heckerman, Dan Geiger, David M. Chickering *Learning Bayesian Networks: The Combination of Knowledge and Statistical Data*, 1995

[3] Edward Herskovits, Gregory Cooper *Kutato: An Entropy-Driven System for Construction of Probabilistic Expert Systems from Databases*

[4] Lam and F. Bacchus *Learning Bayesian belief networks: An approach based on the MDL principle*

[5] Barron, Rissanen, Yu *The minimum description length principle in coding and modeling*, 1998

[6] David M. Chickering. *Learning Bayesian networks is NP-complete*

[7] David M. Chickering, Christopher Meek, David Heckerman. *Large-sample learning of Bayesian networks is NP-hard*

[8] David M. Chickering *Optimal Structure Identification With Greedy Search*, Journal of Machine Learning Research, 2002

[9] David M. Chickering, Cristopher Meek *Finding Optimal Bayesian Networks*, UAI, 2002

A Доказательства вспомогательных теорем

A.1 Теорема 1

Теорема 1. *Если* $p_1, p_2, \ldots, p_m \sim Dirichlet(\alpha_1, \alpha_2, \ldots, \alpha_n)$, $x = p_i$, $\alpha_x = \alpha_i$, $y = p_j$, $\alpha_y = \alpha_j$ *и* k_x, k_y, m_x, m_y *являются произвольными положительными константами,*

то

$$\mathbb{E}(x^{k_x} y^{k_y} \log^{m_x} x \log^{m_y} y) =$$
$$\frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} \sum_{i=0}^{m_y} C_{m_y}^{i} \frac{\partial^{m_y - i} B(\alpha_y + k_y, \alpha_z)}{(\partial \alpha_y)^{m_y - i}} \frac{\partial^{m_x + i}}{(\partial \alpha_x)^{m_x} (\partial \alpha_y)^{i}} B(\alpha_x + k_x, \alpha_y + \alpha_z + k_y) \tag{25}$$

где $\alpha_z = \alpha_0 - \alpha_x - \alpha_y$

Доказательство. Пусть z – вероятность реализации любого значения, кроме v_i и v_j. Тогда:

$$x, y, z \sim \text{Dirichlet}(\alpha_x, \alpha_y, \alpha_z) \tag{26}$$

$$p(x, y, z) = \frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} x^{\alpha_x - 1} y^{\alpha_y - 1} z^{\alpha_z - 1} \tag{27}$$

$x + y + z = 1$, поэтому z – детерминированная функция от x и y:

$$p(x, y) = \frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} x^{\alpha_x - 1} y^{\alpha_y - 1} (1 - x - y)^{\alpha_z - 1} \tag{28}$$

$$\mathbb{E}(x^{k_x} y^{k_y} \log^{m_x} x \log^{m_y} y) = \int\limits_{0<x+y<1} p(x, y)\, x^{k_x} y^{k_y} \log^{m_x} x \log^{m_y} y \, dx \, dy =$$
$$\int\limits_{0<x+y<1} \frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} x^{\alpha_x - 1} y^{\alpha_y - 1} (1 - x - y)^{\alpha_z - 1} \, x^{k_x} y^{k_y} \log^{m_x} x \log^{m_y} y \, dx \, dy =$$
$$\frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} \int\limits_{0<x+y<1} x^{\alpha_x + k_x - 1} y^{\alpha_y + k_y - 1} (1 - x - y)^{\alpha_z - 1} \, \log^{m_x} x \log^{m_y} y \, dx \, dy =$$
$$\frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} \int\limits_0^1 x^{\alpha_x + k_x - 1} \log^{m_x} x \, dx \int\limits_0^{1-x} y^{\alpha_y + k_y - 1} (1 - x - y)^{\alpha_z - 1} \, \log^{m_y} y \, dy =$$
$$\frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} \int\limits_0^1 x^{\alpha_x + k_x - 1} \log^{m_x} x \, dx \int\limits_0^{1-x} \frac{\partial^{m_y}}{(\partial \alpha_y)^{m_y}} (y^{\alpha_y + k_y - 1} (1 - x - y)^{\alpha_z - 1}) \, dy =$$
$$\frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} \int\limits_0^1 x^{\alpha_x + k_x - 1} \log^{m_x} x \, dx \, \frac{\partial^{m_y}}{(\partial \alpha_y)^{m_y}} \int\limits_0^{1-x} (y^{\alpha_y + k_y - 1} (1 - x - y)^{\alpha_z - 1}) \, dy$$

Теперь посчитаем $I(x) = \int\limits_0^{1-x} (y^{\alpha_y + k_y - 1} (1 - x - y)^{\alpha_z - 1}) \, dy$.

Пусть $y = (1-x)t$, тогда

$$I(x) = \int_0^1 ((1-x)t)^{\alpha_y+k_y-1}(1-x-(1-x)t)^{\alpha_z-1}\,d((1-x)t) =$$

$$\int_0^1 (1-x)^{\alpha_y+\alpha_z+k_y-1}\,t^{\alpha_y+k_y-1}(1-t)^{\alpha_z-1}\,dt =$$

$$(1-x)^{\alpha_y+\alpha_z+k_y-1}\int_0^1 t^{\alpha_y+k_y-1}(1-t)^{\alpha_z-1}\,dt = (1-x)^{\alpha_y+\alpha_z+k_y-1}B(\alpha_y+k_y,\alpha_z)$$

$$\mathbb{E}(x^{k_x} y^{k_y} \log^{m_x} x \log^{m_y} y) =$$

$$\frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} \int\limits_0^1 x^{\alpha_x + k_x - 1} \log^{m_y} x \, \frac{\partial^{m_y} (1-x)^{\alpha_y + \alpha_z + k_y - 1} B(\alpha_y + k_y, \alpha_z)}{(\partial \alpha_y)^{m_y}} \, dx =$$

$$\sum_{i=0}^{m_y} C_{m_y}^i \frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} \frac{\partial^{m_y - i} B(\alpha_y + k_y, \alpha_z)}{(\partial \alpha_y)^{m_y - i}} \cdot$$

$$\cdot \int\limits_0^1 x^{\alpha_x + k_x - 1} \log^{m_x} x \, (1-x)^{\alpha_y + \alpha_z + k_y - 1} \, \log^i (1-x) dx =$$

$$\sum_{i=0}^{m_y} \frac{C_{m_y}^i}{B(\alpha_x, \alpha_y, \alpha_z)} \frac{\partial^{m_y - i} B(\alpha_y + k_y, \alpha_z)}{(\partial \alpha_y)^{m_y - i}} \cdot$$

$$\cdot \int\limits_0^1 x^{\alpha_x + k_x - 1} \log^{m_x} x \, (1-x)^{\alpha_y + \alpha_z + k_y - 1} \, \log^i (1-x) dx =$$

$$\sum_{i=0}^{m_y} \frac{C_{m_y}^i}{B(\alpha_x, \alpha_y, \alpha_z)} \frac{\partial^{m_y - i} B(\alpha_y + k_y, \alpha_z)}{(\partial \alpha_y)^{m_y - i}} \cdot$$

$$\cdot \int\limits_0^1 \frac{\partial^{m_x}}{(\partial \alpha_x)^{m_x}} (x^{\alpha_x + k_x - 1} \, (1-x)^{\alpha_y + \alpha_z + k_y - 1} \, \log^i (1-x)) \, dx =$$

$$\sum_{i=0}^{m_y} \frac{C_{m_y}^i}{B(\alpha_x, \alpha_y, \alpha_z)} \frac{\partial^{m_y - i} B(\alpha_y + k_y, \alpha_z)}{(\partial \alpha_y)^{m_y - i}} \frac{\partial^{m_x}}{(\partial \alpha_x)^{m_x}} \cdot$$

$$\cdot \int\limits_0^1 x^{\alpha_x + k_x - 1} \, (1-x)^{\alpha_y + \alpha_z + k_y - 1} \, \log^i (1-x) \, dx =$$

$$\sum_{i=0}^{m_y} \frac{C_{m_y}^i}{B(\alpha_x, \alpha_y, \alpha_z)} \frac{\partial^{m_y - i} B(\alpha_y + k_y, \alpha_z)}{(\partial \alpha_y)^{m_y - i}} \frac{\partial^{m_x}}{(\partial \alpha_x)^{m_x}} \frac{\partial^i}{(\partial \alpha_y)^i} \int\limits_0^1 x^{\alpha_x + k_x - 1} \, (1-x)^{\alpha_y + \alpha_z + k_y - 1} \, dx =$$

$$\frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} \sum_{i=0}^{m_y} C_{m_y}^i \frac{\partial^{m_y - i} B(\alpha_y + k_y, \alpha_z)}{(\partial \alpha_y)^{m_y - i}} \frac{\partial^{m_x + i}}{(\partial \alpha_x)^{m_x} (\partial \alpha_y)^i} B(\alpha_x + k_x, \alpha_y + \alpha_z + k_y)$$

□

A.2 Теорема 2

Теорема 2. *Если* $x \sim Beta(\alpha_x, \alpha_y)$, $y = 1 - x$ *и* k_x, k_y, m_x, m_y *являются произвольными положительными константами, то*

$$\mathbb{E}(x^{k_x} y^{k_y} \log^{m_x} x \log^{m_y} y) = \frac{1}{B(\alpha_x, \alpha_y)} \frac{\partial^{m_x+m_y} B(\alpha_x + k_x, \alpha_y + k_y)}{(\partial\alpha_x)^{m_x}(\partial\alpha_y)^{m_y}} \tag{29}$$

Доказательство.

$$\mathbb{E}(x^{k_x} y^{k_y} \log^{m_x} x \log^{m_y} y) = \frac{1}{B(\alpha_x, \alpha_y)} \int_0^1 p(x) x^{k_x} y^{k_y} \log^{m_x} x \log^{m_y} y dx =$$

$$\int_0^1 x^{\alpha_x+k_x-1} y^{\alpha_y+k_y-1} \log^{m_x} x \log^{m_y} y \, dx =$$

$$\frac{1}{B(\alpha_x, \alpha_y)} \int_0^1 x^{\alpha_x+k_x-1} (1-x)^{\alpha_y+k_y-1} \log^{m_x} x \log^{m_y} (1-x) \, dx =$$

$$\frac{1}{B(\alpha_x, \alpha_y)} \int_0^1 \frac{\partial^{m_x}}{(\partial\alpha_x)^{m_x}} \left(x^{\alpha_x+k_x-1} (1-x)^{\alpha_y+k_y-1} \log^{m_y} (1-x)\right) dx =$$

$$\frac{1}{B(\alpha_x, \alpha_y)} \int_0^1 \frac{\partial^{m_x+m_y}}{(\partial\alpha_x)^{m_x}(\partial\alpha_y)^{m_y}} \left(x^{\alpha_x+k_x-1} (1-x)^{\alpha_y+k_y-1}\right) dx =$$

$$\frac{1}{B(\alpha_x, \alpha_y)} \frac{\partial^{m_x+m_y}}{(\partial\alpha_x)^{m_x}(\partial\alpha_y)^{m_y}} \int_0^1 \left(x^{\alpha_x+k_x-1} (1-x)^{\alpha_y+k_y-1}\right) dx =$$

$$\frac{1}{B(\alpha_x, \alpha_y)} \frac{\partial^{m_x+m_y} B(\alpha_x + k_x, \alpha_y + k_y)}{(\partial\alpha_x)^{m_x}(\partial\alpha_y)^{m_y}}$$

□

A.3 Теорема 3

Теорема 3. *Если* $p_1, p_2, \ldots, p_m \sim Dirichlet(\alpha_1, \alpha_2, \ldots, \alpha_n)$, $x = p_i$, $\alpha_x = \alpha_i$, $y = p_j$, $\alpha_y = \alpha_j$, *то*

$$\mathbb{E}(x^{k_x}\, y^{k_y} \log^{m_x} x \, \log^{m_y} y \, \log(x+y)) =$$
$$-\frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} \sum_{n=1}^{\infty} \frac{1}{n} \sum_{i=0}^{m_y} C_{m_y}^{i} \frac{\partial^{m_y - i} B(\alpha_y + k_y, \alpha_z + n)}{(\partial \alpha_y)^{m_y - i}} \cdot$$
$$\cdot \frac{\partial^{m_x + i}}{(\partial \alpha_x)^{m_x} (\partial \alpha_y)^{i}} B(\alpha_x + k_x, \alpha_y + \alpha_z + n + k_y)$$

Доказательство.

$$\mathbb{E}(x^{k_x}\, y^{k_y} \log^{m_x} x \, \log^{m_y} y \, \log(x+y)) =$$
$$\int\limits_{0<x+y<1} \frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} x^{k_x + \alpha_x - 1} y^{k_y + \alpha_y - 1} (1-x-y)^{\alpha_z - 1} \log(x+y) \log^{m_x} x \, \log^{m_y} y \, dx\, dy =$$
$$\frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} \int\limits_{0<x+y<1} x^{k_x + \alpha_x - 1} y^{k_y + \alpha_y - 1} (1-x-y)^{\alpha_z - 1} \log(1-(1-x-y)) \cdot$$
$$\cdot \log^{m_x} x \, \log^{m_y} y \, dx\, dy = \frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} \int\limits_{0<x+y<1} x^{k_x + \alpha_x - 1} y^{k_y + \alpha_y - 1} (1-x-y)^{\alpha_z - 1} \cdot$$
$$\cdot \log(-\sum_{n=1}^{\infty} \frac{(1-x-y)^n}{n}) \log^{m_x} x \, \log^{m_y} y \, dx\, dy =$$
$$-\frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} \sum_{n=1}^{\infty} \frac{1}{n} \int\limits_{0<x+y<1} x^{k_x + \alpha_x - 1} y^{k_y + \alpha_y - 1} (1-x-y)^{n + \alpha_z - 1} \log^{m_x} x \, \log^{m_y} y \, dx\, dy \tag{30}$$

Заметим, что интеграл имеет такую же форму, как и в доказательстве теоремы 1, поэтому получаем:

$$\mathbb{E}(x^{k_x}\, y^{k_y} \log^{m_x} x \, \log^{m_y} y \, \log(x+y)) = -\frac{1}{B(\alpha_x, \alpha_y, \alpha_z)} \sum_{n=1}^{\infty} \frac{1}{n} \sum_{i=0}^{m_y} C_{m_y}^{i} \cdot$$
$$\cdot \frac{\partial^{m_y - i} B(\alpha_y + k_y, \alpha_z + n)}{(\partial \alpha_y)^{m_y - i}} \frac{\partial^{m_x + i}}{(\partial \alpha_x)^{m_x} (\partial \alpha_y)^{i}} B(\alpha_x + k_x, \alpha_y + \alpha_z + n + k_y)$$

□