ЛАБОРАТОРНАЯ РАБОТА #4

Байесовские сети

Ивченко Матвей Сергеевич
М8О-307Б-23

Направление лабораторной работы

- Лабораторная работа посвящена изучению байесовских сетей **Bayesian Networks** с помощью библиотеки pgmpy. В целях освоить загрузку и обработку данных, построение модели, оценку параметров, анализ вероятностных таблиц, визуализацию и дополнительные аспекты анализа. Работа выполняется на Python, с использованием библиотек *pandas, pgmpy, networkx и matplotlib*.

Введение

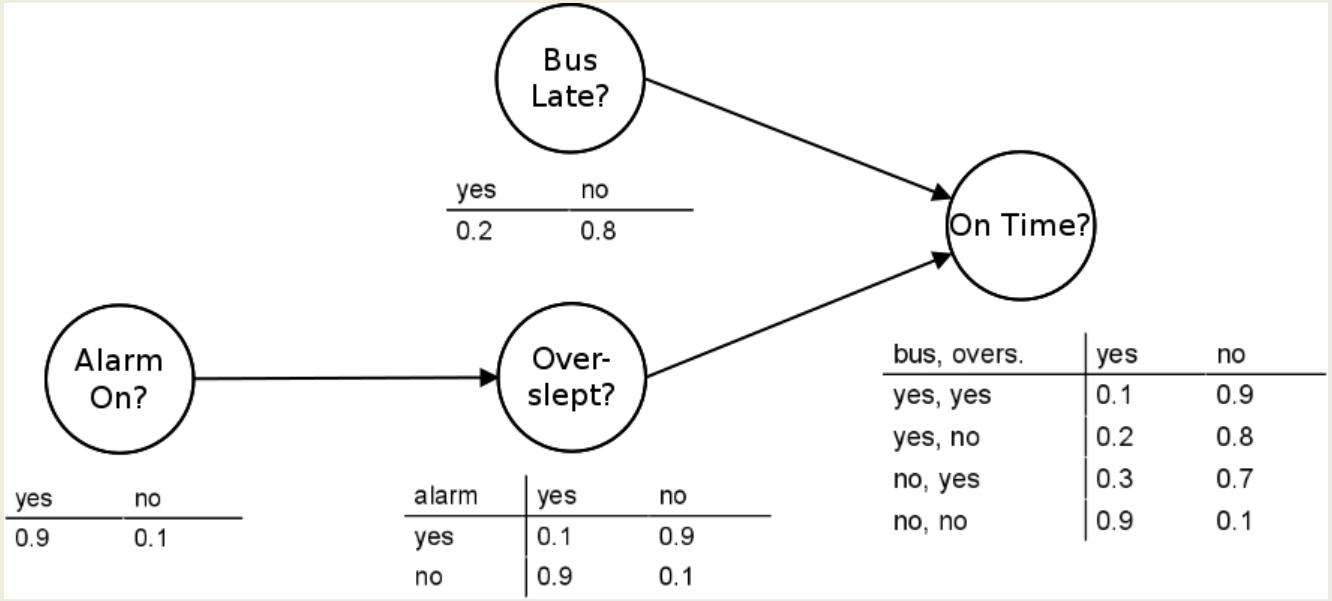
Байесовские сети (Bayesian Networks) — это графические вероятностные модели, представляющие множество случайных переменных и их условные зависимости с помощью направленного ациклического графа (DAG). В узлах графа находятся переменные, а рёбра отражают зависимости между ними. Каждая переменная имеет таблицу условных вероятностей, которая описывает её распределение при заданных значениях родительских узлов.

Ключевые особенности:

• Позволяют моделировать неопределённость и причины-следственные связи.

• Используются для вероятностного вывода и принятия решений в условиях неполной информации.

• Применяются в медицине, машинном обучении, обработке естественного языка и других областях.



Введение

Байесовские сети строятся на теореме Байеса:

$$P(H \mid E) = P(H) \times \frac{P(E \mid H)}{P(E)}$$

P – это вероятность, *probability*

H (*hypothesis*) – гипотеза, которую нам надо оценить

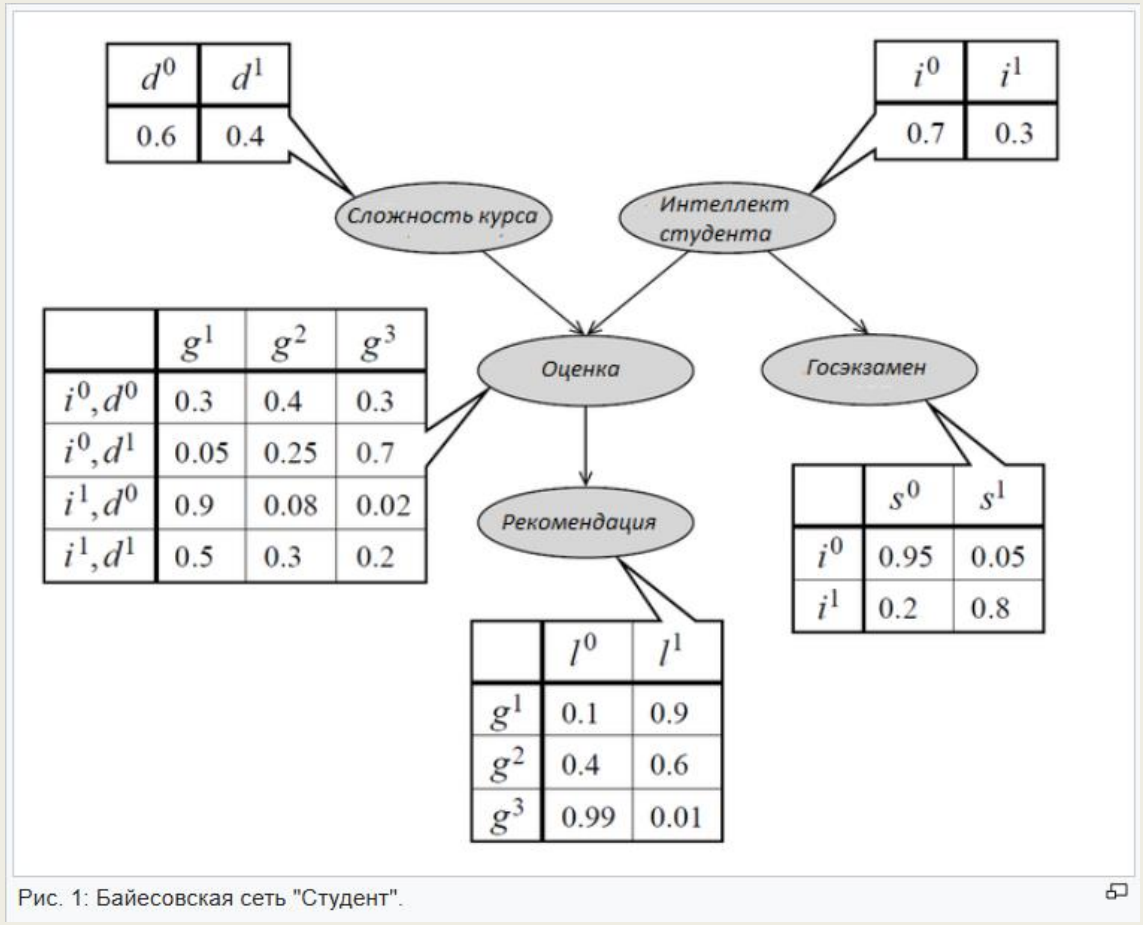
Вертикальная черта читается как «при условии», то есть P(H|E) можно расшифровать как вероятность, что произойдёт H при условии, что событие E произошло

E (*evidence*) – некоторое событие, которое произошло и, возможно, повлияло на оценку нашей гипотезы

И цепном правиле:

$$\mathrm{P}(X_1, \ldots, X_n) = \prod_{i=1}^{n} \mathrm{P}(X_i \mid \mathrm{parents}(X_i))$$

Цепное правило позволяет разложить (факторизовать) совместное распределение в произведение условных распределений.



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^1, d^0	0.9	0.08	0.02
i^1, d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

Рис. 1: Байесовская сеть "Студент".

Обработка датасета

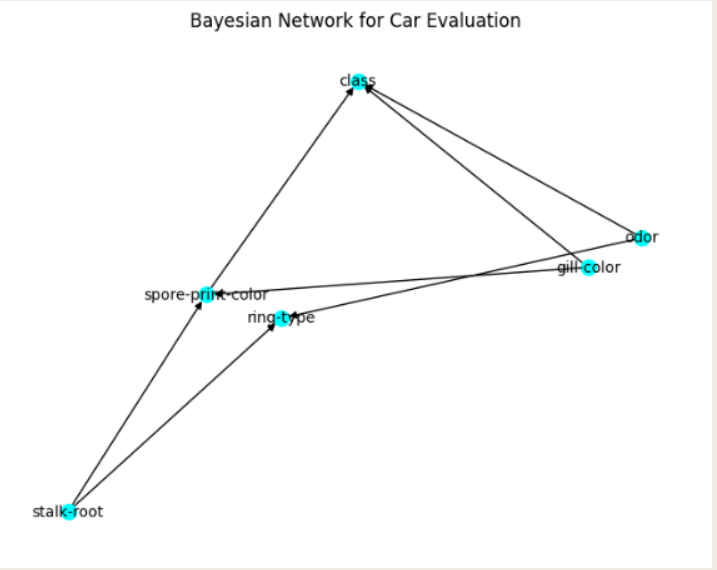
- Датасет mushrooms был загружен с OpenML с помощью функции fetch_openml из пакета scikit-learn.
- Датасет содержит исключительно дискретные признаки, а потому был закодирован с помощью col.cat.codes (встроенный в pandas аналог Label Encoding)
- Пропуски были только в одном признаки, они были заполнены модой.

	cap-shape	cap-surface	cap-color	bruises%3F	odor	gill-attachment	gill-spacing	gill-size
0	x	s	n	t	p	f	c	n
1	x	s	y	t	a	f	c	b
2	b	s	w	t	l	f	c	b
3	x	y	w	t	p	f	c	n
4	x	s	g	f	n	f	w	b

Часть датасета

Построение Bayesian Network

- Сеть строилась вручную (автоматические методы давали неинтерпретируемую сложную архитектуру)
- В ходе экспериментов выяснилось, что из-за обилия категориальных признаков, имеющих большое количество уникальных значений, сеть должна быть неглубокого размера, иначе таблицы распределений становятся слишком большими (и требуют умопомрачительных объёмов данных) и разреженными (а значит в датасете не будет необходимых примеров для обучения)
- Обучение происходило с помощью BayesianEstimator, так как он даёт менее разреженные таблицы распределений



Итоговая структура

Инференс и результаты

- Так как модель явно недообучена, качество у неё будет слабое. Однако для очевидных примеров результаты будут правильные

```
Edible:
+----------+--------------+
| class    |   phi(class) |
+==========+==============+
| class(e) |       0.9995 |
+----------+--------------+
| class(p) |       0.0005 |
+----------+--------------+
Poisonous:
+----------+--------------+
| class    |   phi(class) |
+==========+==============+
| class(e) |       0.3229 |
+----------+--------------+
| class(p) |       0.6771 |
+----------+--------------+
```

```
Accuracy: 0.0054
Log-Likelihood: -76000.3582
```

Bayesian Network

```
Accuracy: 0.9507692307692308
Log-likelihood: -225.43986722309594
```

Логистическая регрессия

Выводы

Хотя в этой работе Bayesian Networks показали себя не так хорошо, как даже линейная модель, не стоит сразу отказываться от этой идеи. Архитетура BN предполагает наличие экспертной оценки этого графа, а также данные, которые были бы собраны с учётом этой структуры. Такая модель будет прекрасно интерпретироваться, так как структура графа была определена изначально из логических и интерпретируемых выводов экспертов, а значения таблиц - подобраны статистически. Однако на практике нам хочется, чтобы модель умела хотя бы интерполировать зависимости, чем BN, к сожалению, похвастаться не может.