



Лабораторная работа №4

Тема: “Лабораторная работа: Bayesian Networks на примере датасета Mushroom Classification”

Таланкин Кирилл Александрович
М80-309Б-23

Bayesian Networks

Bayesian Network - это вероятностная графовая модель, у которой узлы это переменные, причинно-следственные зависимости обозначаются стрелками. Количественные зависимости задаются таблицей условных вероятностей

Bayesian Network используется для:

- Классификации
- Прогнозирования
- Моделирования неопределенности
- Объяснения зависимости между признаками

Описание датасета

- ▶ Датасет: Mushroom Classification. Он содержит 8124 строки, 23 столбца, из которых class и 22 признака
- ▶ Признаки: все категориальные
- ▶ Классы: съедобные 52%, ядовитые 48%

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment
0	p	x	s	n	t	p	f
1	e	x	s	y	t	a	f
2	e	b	s	w	t	l	f

gill-spacing	gill-size	gill-color	stalk-shape	stalk-root	stalk-surface-above-ri...
c	n	k	e	e	s
c	b	k	e	c	s
c	b	n	e	c	s

stalk-surface-below-ri...	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color
s	w	w	p	w
s	w	w	p	w
s	w	w	p	w

veil-type	veil-color	ring-number	ring-type	spore-print-color	population	habitat
p	w	o	p	k	s	u
p	w	o	p	n	n	g
p	w	o	p	n	n	m

Загрузка датасета

```
1 import pandas as pd
2
3 # Загрузка датасета
4 data = pd.read_csv('mushrooms.csv')
✓ [9] 13ms
```

Посмотрим на его структуру

```
1 data.head(3)
✓ [10] 17ms
```

	class	cap-shape	cap-surface	cap-color	bruise
0	p	x	s	n	t
1	e	x	s	y	t
2	e	b	s	w	t

```
1 data.shape
✓ [11] < 10 ms
(8124, 23)
```

Обработка датасета

2. Обработка датасета

можно закодировать метки с помощью Label Encoding

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
for col in data.columns:
    data[col] = le.fit_transform(data[col])

data.head(3)
```

✓ [12] 32ms

	class	cap-shape	cap-surface	cap-color	bruise
0	1	5	2	4	
1	0	5	2	9	
2	0	0	2	8	

можно сохранить как категориальный, но удаляем дубликаты

```
data = data.drop_duplicates()
print(data.shape)
```

✓ [13] 11ms

(8124, 23)

Построение Bayesian Network

```
1 from pgmpy.models import DiscreteBayesianNetwork
2
3 # Задаем направление между вершинами
4 network = [
5     ('cap-shape', 'class'),
6     ('cap-surface', 'class'),
7     ('cap-color', 'class'),
8     ('bruises', 'class'),
9     ('odor', 'class'),
10    ('gill-attachment', 'class'),
11    ('gill-spacing', 'class'),
12    ('gill-size', 'class'),
13    ('gill-color', 'class'),
14    ('stalk-shape', 'class'),
15    ('stalk-root', 'class'),
16    ('stalk-surface-above-ring', 'class'),
17    ('stalk-surface-below-ring', 'class'),
18    ('stalk-color-above-ring', 'class'),
19    ('stalk-color-below-ring', 'class'),
20    ('veil-type', 'class'),
21    ('veil-color', 'class'),
22    ('ring-number', 'class'),
23    ('ring-type', 'class'),
24    ('spore-print-color', 'class'),
25    ('population', 'class'),
26    ('habitat', 'class')
27 ]
28
29 # Строим Дискретную Байесовскую сеть
30 model = DiscreteBayesianNetwork(network)
31 model.edges() # Просмотр ребер
✓ [14] 1s 582ms
```

```
OutEdgeView([('cap-shape', 'class'), ('cap-surface', 'class'),
             ('cap-color', 'class'), ('bruises', 'class'), ('odor', 'class'),
             ('gill-attachment', 'class'), ('gill-spacing', 'class'), ('gill-size',
             'class'), ('gill-color', 'class'), ('stalk-shape', 'class'),
             ('stalk-root', 'class'), ('stalk-surface-above-ring', 'class'),
             ('stalk-surface-below-ring', 'class'), ('stalk-color-above-ring',
             'class'), ('stalk-color-below-ring', 'class'), ('veil-type', 'class'),
             ('veil-color', 'class'), ('ring-number', 'class'), ('ring-type',
             'class'), ('spore-print-color', 'class'), ('population', 'class'),
             ('habitat', 'class')])
```

Построение Bayesian Network

```
1 from pgmpy.estimators import HillClimbSearch, BIC
2
3 hc = HillClimbSearch(data)
4 best_model = hc.estimate(scoring_method=BIC(data))
5 model = DiscreteBayesianNetwork(best_model.edges())
6 model.edges() # Автоматическая структура
✓ [15] 2s 341ms
```

INFO:pgmpy: Datatype (N=numerical, C=Categorical Unordered, O=Categorical Ordered) inferred from data:

```
{'class': 'N', 'cap-shape': 'N', 'cap-surface': 'N', 'cap-color': 'N', 'bruises': 'N', 'odor': 'N', 'gill-attachment': 'N',
  'gill-spacing': 'N', 'gill-size': 'N', 'gill-color': 'N', 'stalk-shape': 'N', 'stalk-root': 'N',
  'stalk-surface-above-ring': 'N', 'stalk-surface-below-ring': 'N', 'stalk-color-above-ring': 'N', 'stalk-color-below-ring':
  'N', 'veil-type': 'N', 'veil-color': 'N', 'ring-number': 'N', 'ring-type': 'N', 'spore-print-color': 'N', 'population':
  'N', 'habitat': 'N'}
```

INFO:pgmpy: Datatype (N=numerical, C=Categorical Unordered, O=Categorical Ordered) inferred from data:

```
{'class': 'N', 'cap-shape': 'N', 'cap-surface': 'N', 'cap-color': 'N', 'bruises': 'N', 'odor': 'N', 'gill-attachment': 'N',
  'gill-spacing': 'N', 'gill-size': 'N', 'gill-color': 'N', 'stalk-shape': 'N', 'stalk-root': 'N',
  'stalk-surface-above-ring': 'N', 'stalk-surface-below-ring': 'N', 'stalk-color-above-ring': 'N', 'stalk-color-below-ring':
  'N', 'veil-type': 'N', 'veil-color': 'N', 'ring-number': 'N', 'ring-type': 'N', 'spore-print-color': 'N', 'population':
  'N', 'habitat': 'N'}
```

0% | 53/1000000 [00:02<10:57:10, 25.36it/s]

```
OutEdgeView([('class', 'habitat'), ('class', 'stalk-surface-above-ring'), ('class', 'population'), ('class', 'bruises'),
  ('class', 'stalk-surface-below-ring'), ('habitat', 'stalk-color-below-ring'), ('habitat', 'ring-number'), ('population',
  'gill-spacing'), ('population', 'cap-surface'), ('bruises', 'habitat'), ('bruises', 'cap-color'), ('bruises', 'cap-shape'),
  ('odor', 'class'), ('odor', 'cap-color'), ('odor', 'gill-spacing'), ('gill-spacing', 'ring-type'), ('gill-spacing',
  'stalk-surface-above-ring'), ('gill-spacing', 'habitat'), ('ring-type', 'stalk-color-below-ring'), ('ring-type',
  'stalk-surface-below-ring'), ('ring-type', 'bruises'), ('ring-type', 'stalk-surface-above-ring'), ('gill-size',
```

Оценка параметров и CPT

```
1 # Байесовский оценщик
2 from pgmpy.estimators import BayesianEstimator
3
4 model.fit(data, estimator=BayesianEstimator, prior_type='BDeu', equivalent_sample_size=10)
✓ [17] 97ms

INFO:pgmpy: Datatype (N=numerical, C=Categorical Unordered, O=Categorical Ordered) inferred from data:
{'class': 'N', 'cap-shape': 'N', 'cap-surface': 'N', 'cap-color': 'N', 'bruises': 'N', 'odor': 'N', 'gill-attachment': 'N',
 'gill-spacing': 'N', 'gill-size': 'N', 'gill-color': 'N', 'stalk-shape': 'N', 'stalk-root': 'N',
 'stalk-surface-above-ring': 'N', 'stalk-surface-below-ring': 'N', 'stalk-color-above-ring': 'N', 'stalk-color-below-ring':
 'N', 'veil-type': 'N', 'veil-color': 'N', 'ring-number': 'N', 'ring-type': 'N', 'spore-print-color': 'N', 'population':
 'N', 'habitat': 'N'}

WARNING:pgmpy:Replacing existing CPD for class
WARNING:pgmpy:Replacing existing CPD for habitat
WARNING:pgmpy:Replacing existing CPD for stalk-surface-above-ring
WARNING:pgmpy:Replacing existing CPD for population
WARNING:pgmpy:Replacing existing CPD for bruises
WARNING:pgmpy:Replacing existing CPD for stalk-surface-below-ring
WARNING:pgmpy:Replacing existing CPD for stalk-color-below-ring

<pgmpy.models.DiscreteBayesianNetwork.DiscreteBayesianNetwork at 0x1a54d2520d0>
```


Оценка параметров и СРТ

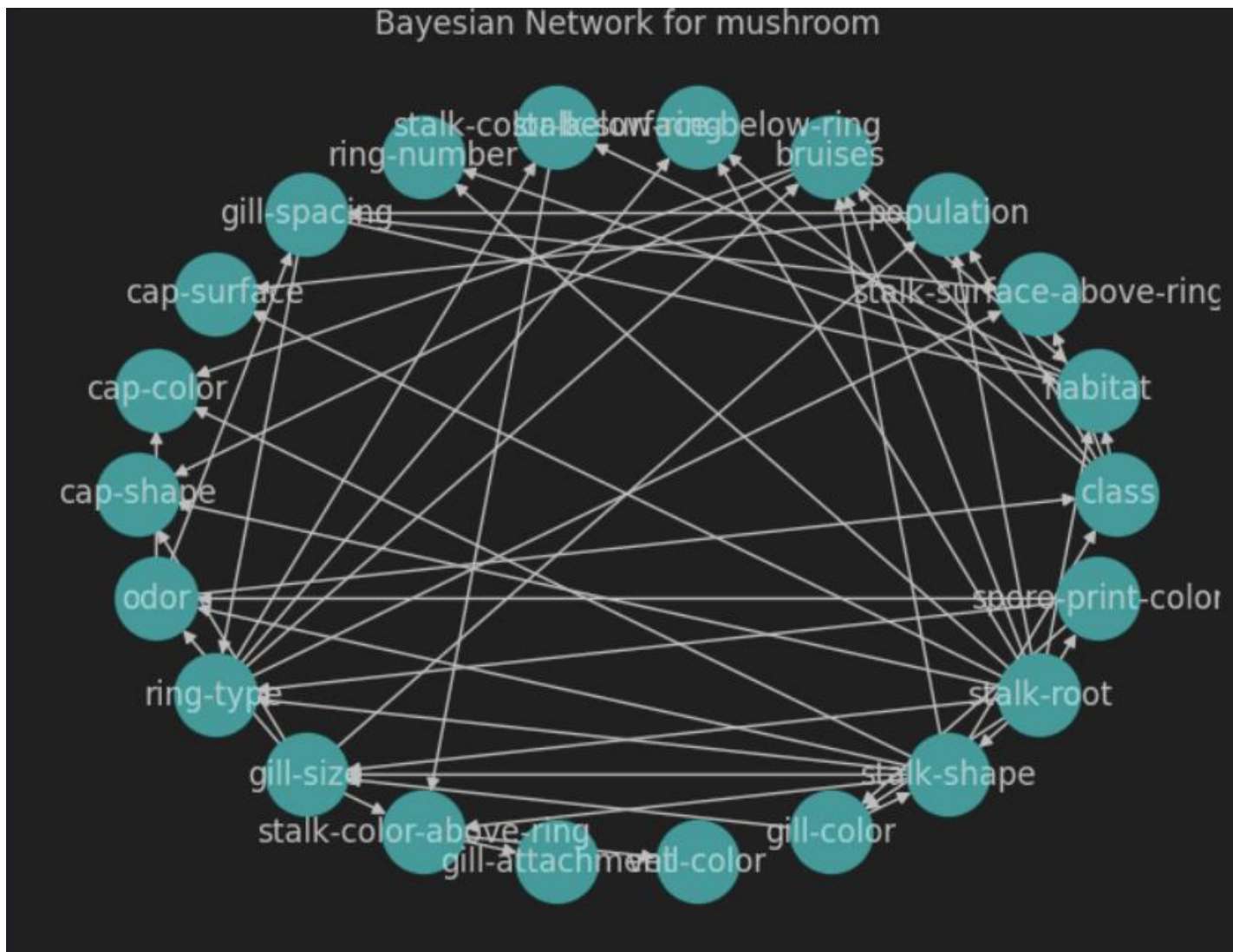
```
1 for node in ['odor', 'bruises', 'veil-color']:
2     cpt = model.get_cpds(node)
3     print(f"CPT for {node}:\n{cpt}")
```

✓ [20] 10ms

CPT for odor:

```
+-----+-----+-----+
| gill-size      | ... | gill-size(1)      |
+-----+-----+-----+
| spore-print-color | ... | spore-print-color(8) |
+-----+-----+-----+
| stalk-shape     | ... | stalk-shape(1)      |
+-----+-----+-----+
| odor(0)         | ... | 0.1111111111111111 |
+-----+-----+-----+
| odor(1)         | ... | 0.1111111111111111 |
+-----+-----+-----+
| odor(2)         | ... | 0.1111111111111111 |
```

Визуализация сети



Пример интерфейса

5. Посмотрим пример инференса

```
1 from pgmpy.inference import VariableElimination
2
3 infer = VariableElimination(model)
4 query = infer.query(variables=['class'], evidence={'odor': 2})
5 print(query) # Вероятности классов
✓ [36] < 10 ms
```

3 rows ▾ 3 rows × 2 cols		Static Output
class ▾	phi(class) ▾	
class(0)	0.0003	
-----	-----	
class(1)	0.9997	

Оценка с Baseline

```
61
62 print(f"Bayesian Network – accuracy: {acc_bn}")
63 print(f"Bayesian Network – log-loss: {ll_bn}")
64
65 nb = CategoricalNB()
66 nb.fit(X_train, y_train)
67
68 y_pred_nb = nb.predict(X_test)
69 y_prob_nb = nb.predict_proba(X_test)
70
71 acc_nb = accuracy_score(y_test, y_pred_nb)
72 ll_nb = log_loss(y_test, y_prob_nb, labels=nb.classes_)
73
74 print(f"CategoricalNB – accuracy: {acc_nb}")
75 print(f"CategoricalNB – log-loss: {ll_nb}")
✓ [37] 966ms
```

```
Bayesian Network – accuracy: 1.0
Bayesian Network – log-loss: 0.00036036407835434483
CategoricalNB – accuracy: 0.9527326440177253
CategoricalNB – log-loss: 0.1305532135783994
```