HW1 + HW2

Гаврилов В.Р. 142М группа.

27 сентября 2022 г.

1. Представление канонической информации

Итак, на вход поступает последовательность векторов x_i. Необходимо вычислить вектор выборочного среднего

$$X = \frac{1}{n}\sum_{i=1}^{n} x_i,$$

а также выборочную ковариационную матрицу

$$V = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - X)(x_i - X)^T.$$

В качестве канонической информации предлагается следующая тройка: (S_1, S_2, n), где:

$$S_1 = \sum_{i=1}^{n} x_i,$$

$$S_2 = \sum_{i=1}^{n} x_i x_i^T$$

2. Проверка свойств канонической информации

2.1. Существование и единственность

Очевидна.

2.2. Полнота

Действительно, что X, что V могут быть представлены через тройку (S_1, S_2, n) следующим образом:

$$X = \frac{1}{n}S_1,$$

$$V = \frac{1}{n-1}(\sum_{i=1}^{n} x_i x_i^T - X(\sum_{i=1}^{n} x_i^T) - (\sum_{i=1}^{n} x_i)X^T + nXX^T) = \frac{1}{n-1}(S_2 - \frac{1}{n}S_1 S_1^T)$$

2.3. Элементарная каноническая информация

Каноническая информация для единственного наблюдения не существует, так как требуется рассчитать несмещенную оценку ковариационной матрицы и данное значение не определено для $n = 1$.

2.4. Пустая каноническая информация

Существует, в виде тройки нулей $(0, 0, 0)$.

2.5. Композиция

Пусть два набора векторов представлены через канонические информации (S_1^1, S_2^1, n^1) и (S_1^2, S_2^2, n^2). Тогда их композиция определяется следующим образом:

$$(S_1^1, S_2^1, n^1) \oplus (S_1^2, S_2^2, n^2) = (S_1^1 + S_1^2, S_2^1 + S_2^2, n^1 + n^2).$$

Коммутативность, ассоциативность и свойство нулевого элемента выполенены в силу соответствующих свойств матриц и вещественных чисел.

2.6. Обновление

Пусть на вход поступает новый вектор x. Тогда каноническая информация нового набора будет вычисляться следующим оразом:

$$(S_1 + x, S_2 + xx^T, n + 1)$$

2.7. Минимальное число наблюдений

X определен при $n \geq 1$, а V при $n \geq 2$. Таким образом, каноническая информация определена при $n \geq 2$.

2.8. Компактность и эффективность

Пусть n - кол-во векторов, а m - размерность векторов, причем, вероятнее всего $m \ll n$. Явная информация потребует $O(n * m)$ памяти. В свою очередь, для хранения канонической информации потребуется $O(m * m)$ памяти. Обновление, как и композиция, потребуют выполнения $O(m * m)$ операций.

Учитывая вышесказанное, при $n \to \infty$, что хранение, что обработка канонической информации потребуют константных затрат. Таким образом, каноническую информацию вполне можно назвать компактной и эффективной.

3. Программная реализация

```
import numpy as np
import scipy.linalg
from functools import reduce

def canInf(x):
  m, n = x.shape
  S1 = np.reshape(np.sum(x, axis=1), (m, 1))
  S2 = np.zeros((m, m))
  for i in range(n):
    a = x[:, i:i + 1]
    S2 += a @ np.transpose(a)
  return (S1, S2, n)

def reduce_(x, y):
    return (x[0] + y[0], x[1] + y[1], x[2] + y[2])

def result(r):
    return (r[0] / r[2], (r[1] - (r[0] @ np.transpose(r[0])) / r[2]) / (r[2] - 1))

m = 3
n = 20

X0 = np.array([[1.],[2.],[3.]])
V0 = np.array([[1., .5, .5],[.5, 1., .5],[.5, .5, 1.]])

Vs = scipy.linalg.sqrtm(V0)

X = X0 @ np.ones((1,n)) + Vs @ np.random.normal(size=(m, n))

nodes = [X[:,:5], X[:,5:10], X[:,10:15], X[:,15:20]]

print(result(reduce(reduce_, map(canInf, nodes)))[0])
print(result(reduce(reduce_, map(canInf, nodes)))[1])

#output
[[0.9457166 ]
 [2.22476709]
 [2.73598639]]
[[0.83747578 0.46275489 0.71241183]
 [0.46275489 1.31909954 0.86327879]
 [0.71241183 0.86327879 1.48644159]]

print(np.reshape(np.mean(X,axis=1), (m, 1)))
print(np.cov(X))

#output
[[0.9457166 ]
 [2.22476709]
 [2.73598639]]
[[0.83747578 0.46275489 0.71241183]
 [0.46275489 1.31909954 0.86327879]
 [0.71241183 0.86327879 1.48644159]]
```