

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании
(КСУП)

Отчет по практической работе по дисциплине «Теория систем и
системный анализ»

ИЗМЕРЕНИЕ И ОЦЕНКА СВОЙСТВ СИСТЕМЫ. СВЁРТКА
ИЗМЕРЕНИЙ

Вариант 2

Выполнил

Студент гр. 513-2:

Заревич М.А.

Проверил

Ассистент каф. КСУП:

Гембух Л.А.

Томск 2025

Оглавление

Введение	3
Задание 1	3
Задание 2	6
Ответы на контрольные вопросы	14

Введение

В современной практике измерений понимание шкал и методов их применения играет ключевую роль в сборе и анализе данных. Шкала служит основой для систематизации и интерпретации измерений, что делает ее одной из важнейших составляющих в области научных исследований и технических приложений. В данной работе мы ставим перед собой несколько задач, целью которых является знакомство с понятием шкалы, понимание различных типов шкал и их свойств, а также изучение принципов представления измерений в разных шкалах. Мы будем работать с номинальными, ранговыми, интервалами и отношениями, чтобы продемонстрировать различные подходы к измерению свойств технической системы. Кроме того, мы рассмотрим методы свертки данных, которые позволят нам обобщить оценки альтернатив по различным критериям.

Задание 1

Техническая система - грузовик

Номинальные шкалы - Страна-производитель, компания-производитель, название модели.

Ранговые шкалы - комфортность, проходимость.

Шкалы интервалов и отношений - запас бензина, максимальная скорость, год выхода модели.

Объект	Страна-производитель	Компания-производитель	Название модели	Комфортность	Пройодимость	Запас бензина, л	Максимальная скорость, км\ч	Год выхода модели, год
Грузовик 1	Страна 1	Компания 1	Модель 1.5	1	1	500	150	2020
Грузовик 2	Страна 2	Компания 2	Модель 2.1	2	2	400	100	2015

Грузовик 3	Страна 3	Компания 3	Модель 3.5	3	3	300	130	2010
Грузовик 4	Страна 1	Компания 4	Модель 4.1	4	4	500	140	2018
Грузовик 5	Страна 3	Компания 3	Модель 3.6	5	5	600	120	2016

№1.1 Измерение свойств с помощью номинальной шкалы.

Матрица Кронекера

Свойство	Символ Кронекера									
	δ_{1-2}	δ_{1-3}	δ_{1-4}	δ_{1-5}	δ_{2-3}	δ_{2-4}	δ_{2-5}	δ_{3-4}	δ_{3-5}	δ_{4-5}
Страна-производитель	0	0	1	0	0	0	0	0	1	0
Компания-производитель	0	0	0	0	0	0	0	0	1	0
Название модели	0	0	0	0	0	0	0	0	0	0

Свойство	Частоты	Мода
Страна-производитель	$A_1=A_4=2/5, A_3=A_5=2/5$	Классы «Страна 1», «Страна 3».
Компания-производитель	$A_3=A_5=2/5$	Класс «Компания 3».
Название модели	$A_1=A_2=A_3=A_4=A_5=1/5$	Нет моды.

№1.2 Измерение свойств с помощью ранговой шкалы.

Объект	Название модели	Комфортность	Проходимость
A1	Модель 1.5	1	1
A2	Модель 2.1	2	2
A3	Модель 3.5	3	3
A4	Модель 4.1	4	4
A5	Модель 3.6	5	5

	Ранги	$A1 > A2 > A3 > A4 > A5$	$A1 > A2 > A3 > A4 > A5$
	Медиана	Медиана - 3	Медиана - 3

№1.3 Измерение свойств с помощью шкал интервалов и отношений.

Объект	Название модели	Запас бензина, л	Максимальная скорость, км\ч
A1	Модель 1.5	500	150
A2	Модель 2.1	400	100
A3	Модель 3.5	300	130
A4	Модель 4.1	500	140
A5	Модель 3.6	600	120
	Среднее	460	128
	Медиана	500	130
	СКО	102	17,2
	Модуль	40	2
	Симметричность	$40 < 102 * 3$, симметрична	$2 < 17,2 * 3$, симметрична

Задание 2

Задание было выполнено в Google Colaboratory, тут скриншоты программы.
Сам файл загрузил вместе с отчётом.

✓ Заревич Михаил 513-2 Вариант 2

✓ Подготовка данных

```
[ ] import math
import openpyxl

workbook = openpyxl.load_workbook("/content/TS_variants_Lab_2.xlsx")
```

```
[ ] sheet = workbook["TS_variants_Lab_4"] # Получение листа по имени
```

```
[ ] sheet = workbook["TS_variants_Lab_4"] # Получение листа по имени
```

▶ # Класс буду использовать в качестве структуры. т.е. пока обойдусь без инкапсуляции,
геттеров, сеттеров и т.д.

```
class Car():
    """
    Хранит данные о машинах
    """

    # Конструктор
    def __init__(self, name, carPrice, maintenancePrice, gasolinePer100, comfort):
        """
        Параметры
        name:
            Название машины
        """
        self.Name = name
        self.CarPrice = carPrice
        self.MaintenancePrice = maintenancePrice
        self.GasolinePer100 = gasolinePer100
        self.Comfort = comfort
```

```
def GetInfo(self):
    """
    Выводит информацию о машине.
    """
    print("Название: " + str(self.Name), "Цена: " + str(self.CarPrice), "Расходы на обслуживание: " + str(self.MaintenancePrice),
          "Бензин на 100 км: " + str(self.GasolinePer100), "Комфорт: " + str(self.Comfort))

    # можно и так
    #return self.Name, self.Car_price, self.Maintainence_price, self.Gasoline_Per_100, self.Comfort
```

```

▶ # список для хранения автомобилей
cars = []

# Получаю данные из моего варианта
for row in sheet.iter_rows(min_row=11, max_row=15, values_only=True):

    new_car = Car(row[0],row[1],row[2],row[3],row[4])
    cars.append(new_car)
|
    print(row)

```

```

⇨ ('A1', 608, 13, 12, 'средний', None, 'на выбор', None, None, None, None)
   ('A2', 689, 30, 11, 'низкий', None, None, None, None, None, None)
   ('A3', 847, 17, 12, 'средний', None, None, None, None, None, None)
   ('A4', 584, 24, 11, 'высокий', None, None, None, None, None, None)
   ('A5', 765, 29, 11, 'высокий', None, None, None, None, None, None)

```

```

[ ] for i in cars:
    i.GetInfo()

```

```

⇨ Название: A1 Цена: 608 Расходы на обслуживание: 13 Бензин на 100 км: 12 Комфорт: средний
  Название: A2 Цена: 689 Расходы на обслуживание: 30 Бензин на 100 км: 11 Комфорт: низкий
  Название: A3 Цена: 847 Расходы на обслуживание: 17 Бензин на 100 км: 12 Комфорт: средний
  Название: A4 Цена: 584 Расходы на обслуживание: 24 Бензин на 100 км: 11 Комфорт: высокий
  Название: A5 Цена: 765 Расходы на обслуживание: 29 Бензин на 100 км: 11 Комфорт: высокий

```


▼ Нормирование параметров

Чем выше значение, тем лучше.

$$q_j(x_i) = q_j^{ab}(x_i) / q_j^{et}.$$

q_j^{ab} - текущий элемент, q_j^{et} - максимальный элемент

Чем ниже значение, тем лучше.

$$q_j(x_i) = (q_j^{\max} - q_j^{ab}(x_i)) / (q_j^{\max} - q_j^{\min}),$$

где q_j^{\min} , q_j^{\max} – соответственно минимальное и максимальное значения j -го критерия.

Важно: q_j^{\min} должно быть меньше минимального значения в наборе данных, а q_j^{\max} долж быть больше.

```
[ ] # Задание минимальных и максимальных значений
    MinPrice = 500
    MaxPrice = 1000
    MinMaintenancePrice = 10
    MaxMaintenancePrice = 40
    MinGasolinePer100 = 5
    MaxGasolinePer100 = 15

    # Определение весовых коэффициентов критериев
    PriceCoefficient = 0.5
    MaintenanceCoefficient = 0.25
    GasolinePer100Coefficient = 0.25
```

```
[ ] # Все параметры: цену, расходы на обслуживание, расход бензина, нормируем по формуле 2, так как, чем ниже эти параметры, тем лучше.
for i in cars:
    i.CarPriceNorm = (MaxPrice - i.CarPrice) / (MaxPrice - MinPrice)
    i.MaintenancePriceNorm = (MaxMaintenancePrice - i.MaintenancePrice) / (MaxMaintenancePrice - MinMaintenancePrice)
    i.GasolinePer100Norm = (MaxGasolinePer100 - i.GasolinePer100) / (MaxGasolinePer100 - MinGasolinePer100)

for i in cars:
    print(i.CarPriceNorm, i.MaintenancePriceNorm, i.GasolinePer100Norm)
```

```
0.784 0.9 0.3
0.622 0.3333333333333333 0.4
0.306 0.7666666666666667 0.3
0.832 0.5333333333333333 0.4
0.47 0.3666666666666664 0.4
```

▼ Аддитивная свёртка

Значение интегрального критерия определяется как сумма значений частных критериев, поделенная на количество частных критериев:

$$\hat{q}(x_i) = \frac{1}{m} \sum_{j=1}^m q_j(x_i), i = \overline{1, n}.$$

В случае если частные критерии имеют различную важность (вес), вместо формулы среднеарифметического для определения значения интегрального критерия используют средневзвешенное арифметическое:

$$\hat{q}(x_i) = \sum_{j=1}^m v_j q_j(x_i), i = \overline{1, n},$$

где v_j – вес j -го критерия, отражающий вклад частного критерия в интегральный.

```
[ ] # число критериев
    number = 3

    sum = 0
    for i in cars:
        sum += i.CarPriceNorm
        sum += i.MaintenancePriceNorm
        sum += i.GasolinePer100Norm

    # Аддитивная свёртка, без коэффициентов
    i.AdConvWoCoef = sum/3

    sum = 0
```

Из полученных данных видно, что, согласно аддитивной свёртке, наилучшим вариантом является машина A1.

```
[ ] for i in cars:
    print(i.Name, i.AdConvWoCoef)
```

```
↩ A1 0.6613333333333334
  A2 0.4517777777777778
  A3 0.4575555555555555
  A4 0.5884444444444444
  A5 0.4122222222222227
```

▼ Разные весовые коэффициенты

```
[ ] sum = 0
    for i in cars:
        sum += PriceCoefficient*i.CarPriceNorm
        sum += MaintenanceCoefficient*i.MaintenancePriceNorm
        sum += GasolinePer100Coefficient*i.GasolinePer100Norm

        # Аддитивная свёртка с разными коэффициентами
        i.AdConvWCoef = sum

    sum = 0
```

Если учесть разный вес коэффициентов, то предпочтительным вариантом всё ещё будет A1

```
[ ] for i in cars:
    print(i.Name, i.AdConvWCoef)
```

```
↔ A1 0.692
   A2 0.4943333333333333
   A3 0.4196666666666667
   A4 0.6493333333333333
   A5 0.4266666666666664
```

▼ Мультипликативная свёртка

Если частные критерии имеют одинаковый вес, значение интегрального критерия определяется по формуле среднегеометрического

$$\hat{q}(x_i) = \sqrt[m]{\prod_{j=1}^m q_j(x_i)}, i = \overline{1, n}.$$

Если же частные критерии имеют различную важность, то определяется средневзвешенное геометрическое

$$\hat{q}(x_i) = \prod_{j=1}^m q_j(x_i)^{v_j}, i = \overline{1, n}$$

✓ Одинаковые весовые коэффициенты

```
sum = 0

for i in cars:
    sum = i.CarPriceNorm
    sum *= i.MaintenancePriceNorm
    sum *= i.GasolinePer100Norm

    #
    i.MultConvWoCoef = pow(sum, 1/3)

sum = 0
```

Согласно мультипликативной свёртке, наилучшим вариантом является машина A1.

```
[ ] for i in cars:
    print(i.Name, i.MultConvWoCoef)
```

```
→ A1 0.5959730330275612
   A2 0.4360902468219687
   A3 0.4128729409596421
   A4 0.56198839628247
   A5 0.410024454883458
```

✓ Разные весовые коэффициенты

```
sum = 0

for i in cars:

    sum = i.CarPriceNorm**PriceCoefficient
    sum *= i.MaintenancePriceNorm**MaintenanceCoefficient
    sum *= i.GasolinePer100Norm**GasolinePer100Coefficient

    i.MultConvWoCoef = sum

sum = 0
```

Мультипликативная свёртка с учётом коэффициентов также показывает, что A1 - наиболее предпочтительный

```
[ ] for i in cars:
    print(i.Name, i.MultConvWoCoef)
```

```
→ A1 0.6382619759473378
   A2 0.4765734856019676
   A3 0.38308281691217794
   A4 0.6199067181218597
   A5 0.42425970378789396
```

✓ Метод идеальной точки

Метод идеальной точки. Прежде всего, необходимо задать идеальную точку x_0 , т. е. объект с наилучшими значениями по всем критериям. Для этого по каждому из частных критериев q_j необходимо определить наилучшее значение $q_j(x_0)$. Как правило, $q_j(x_0) = \max q_j(x_j)$. В случае, если значения нормированы в диапазоне $[0...1]$, идеальной точкой будет элемент с оценками 1 по всем критериям. Значение интегрального критерия для объекта x_i определяется через евклидовое расстояние между ним и идеальной точкой x_0 по всем частным критериям:

$$\hat{q}(x_i) = \sqrt{\frac{1}{m} \sum_{j=1}^m (q_j(x_0) - q_j(x_i))^2}, i = \overline{1, n}.$$

В случае различной важности частных критериев используется взвешенная сумма расстояний

$$\hat{q}(x_i) = \sqrt{\sum_{j=1}^m v_j (q_j(x_0) - q_j(x_i))^2}, i = \overline{1, n}.$$

✓ Одинаковые весовые коэффициенты

```
sum = 0

for i in cars:

    sum += (1 - i.CarPriceNorm)**2
    sum += (1 - i.MaintenancePriceNorm)**2
    sum += (1 - i.GasolinePer100Norm)**2

    i.IdPointWoCoef = math.sqrt(sum/3)

sum = 0
```

Наименьшее расстояние от идеальной точки имеет машина A1

```
[ ] for i in cars:
    print(i.Name, i.IdPointWoCoef)
```

```
⇒ A1 0.426870784508224
   A2 0.5619396303413279
   A3 0.5848305864220977
   A4 0.4494447603350078
   A5 0.5893530665373999
```

▼ Разные весовые коэффициенты

```
[ ] sum = 0

for i in cars:

    sum += PriceCoefficient*(1 - i.CarPriceNorm)**2
    sum += MaintenanceCoefficient*(1 - i.MaintenancePriceNorm)**2
    sum += GasolinePer100Coefficient*(1 - i.GasolinePer100Norm)**2

    i.IdPointWCoef = math.sqrt(sum)

sum = 0
```

Если учесть разные весовые коэффициенты, минимальное расстояние также будет у A1

```
[ ] for i in cars:
    print(i.Name, i.IdPointWCoef)
```

```
⇒ A1 0.385133743003648
   A2 0.5220661941852883
   A3 0.6139455278044715
   A4 0.3981914670663404
   A5 0.5750893650362331
```

Ответы на контрольные вопросы

1. Что изучает теория измерений?

Теория измерений изучает различные виды измерений и погрешности, возникающие при них.

2. Что такое шкала?

Шкала — совокупность каких-либо критериев, с помощью которых можно охарактеризовать объект.

3. Какие существуют типы шкал по типам преобразований?

Количественные и качественные.

4. Какие шкалы относятся к группе количественных шкал?

Шкала интервалов, шкала отношений.

5. Какие способы нормирования показателей существуют?

Минимакс – приведение данных к диапазону $[0,1]$ /
Приведение данных к диапазону $[-1,1]$.
Стандартизация.

6. Опишите методы аддитивной, мультипликативной свертки и метод идеальной точки.

Аддитивная свёртка – значение критерия определяется суммой критериев, умноженных на свой весовой коэффициент.

Мультипликативная свёртка – значение критерия определяется произведением значений критериев в степени весового коэффициента.

Метод идеальной точки – выбирается гипотетический объект с наилучшими показателями. Определяется расстояние исследуемого объекта до него.

7. Как происходит свёртка измерений по критериям, выраженным в ранговой шкале?

Также как и по методу идеальной точки, но вместо нормированного значения критерия используют ранг.

8. Как можно подобрать весовые коэффициенты для критериев?

Весовые коэффициенты подбираются по степени важности критерия. Это можно сделать, например, опросив группу экспертов.

