Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут»

Інститут Прикладного системного аналізу
Кафедра Системного проектування

Лабораторна робота №4

з дисципліни «Теорія прийняття рішень»

«Прийняття рішень за допомогою методів голосування»

Виконала:
студентка групи ДА-42
Балан Катерина

Київ - 2017

Мета роботи

Ознайомитись з методами прийняття рішень за допомогою методів голосування

Варіант №1

Варіант	Профіль			Методи
1.	**5**	**4**	**2**	- відносної більшості - Кондорсе - альтернативних голосів
	a	*c*	*b*	
	b	*a*	*a*	
	c	*b*	*c*	
	d	*d*	*d*	

Аналітичний рахунок

Відносної більшості a(n = 5) , c(n = 4), b(n = 2), d(n = 0).

Перемагає кандидат a.

Кондорсе a:b(9:2) a:c(7:4) a:d(11:0) – Перемагає клієнт a.

Альтернативних голосів – найгірший кандидат D(11) виключаємо його із таблиці

5	**4**	**2**
A	C	B
B	A	A
C	B	C
D	D	D

Найгірший кандидат C(7) виключаємо його із таблиці

5	**4**	**2**
A	C	B
B	A	A
C	B	C

Найгірший кандидат B(9) виключаємо його із таблиці – Перемагає кандидат A

5	**4**	**2**
A	A	B
B	B	A

Короткі теоретичні відомості

Таблиця 1

Кількість голосів	5	3	5	4
Впорядкування кандидатів	a	a	b	c
	d	d	c	d
	c	b	d	b
	b	c	a	a

Правило (метод) відносної більшості. На перше місце вісім виборців поставили кандидата a $(n_a = 8)$, п'ять виборців – b $(n_b = 5)$ і чотири виборці – c $(n_c = 4), n_d = 0$. . Перемагає той кандидат, за якого проголосувала більшість виборців (у даному випадку – а, випадок рівності голосів поки що не розглядаємо). Зрозуміло, що перемогти може й кандидат, за якого проголосували, наприклад, 1 % виборців (за інших – ще менше). Абсурд? Так, але ж за цим методом обираються "мери" в Україні. Тому назвемо це "парадоксом голосування".

Правило Кондорсе. За Кондорсе переможцем оголошується той ка- ндидат, що "перемагає" всіх інших у попарних порівняннях. Так, у попередньому профілі: вісім виборців поставило кандидата a вище за b, дев'ять виборців поставило a нижче за b (позначимо це $a:b=8:9$). Маємо $b:c=8:9$, $c:d=9:8$, $a:c=8:9$, $b:d=5:12$, $a:d=8:9$. Єдиний кандидат, який "перемагає" всіх інших – це кандидат c. Зауважимо, що правило Кондорсе видається вельми логічним – пере- можець перемагає всіх інших у єдиноборствах. Шахіст, що переміг усіх інших претендентів у мікро матчах (скажімо, із двох партій) – безумовно найкращий. Та й при формуванні індивідуальної переваги виборець по- парно порівнює (свідомо чи підсвідомо) кандидатів. Але в читача вже, мабуть, виникло запитання. А як бути, якщо кожен із кандидатів когось перемагає, а комусь програє? У цьому випадку за визначенням перемо- жця за Кондорсе (переможця Кондорсе) не існує. Це один із так званих "парадоксів голосування". Найпростіший випадок маємо при $n=m=3$: для першого виборця a кращий за b і c, b кращий за c (позначимо це $a \succ b \succ c$); для другого $b \succ c \succ a$; для третього $c \succ a \succ b$. Цей профіль на- зивається "Циклом Кондорсе". Маємо – $a:b=2:1$, $a:c=1:2$, $b:c=2:1$ (у кожного по одному виграшу і по одному програшу).

Метод альтернативних голосів. Виключаємо тих кандидатів, хто отримав найменшу кількість голосів. Потім знову підраховуємо голоси виборців для кандидатів, що залишились і знову виключаємо "найгірших" до того часу, поки не залишиться один кандидат (або декілька з рівною кількістю голосів).

Виконання роботи

Результати розрахунку

```
/usr/bin/python2.7 /home/katya/workspace/Decision_theory/lab_tpr4/lab4.py
Profiles:
['a', 'c', 'b']
['b', 'a', 'a']
['c', 'b', 'c']
['d', 'd', 'd']
Profiles weight: [5, 4, 2]
Winner (relative majority): a
Winner (Condorcet): a
Winner (method of alternative votes): a
```

```
Process finished with exit code 0
```

Лістинг програми

https://github.com/katebalan/Decision_theory/tree/master/lab_tpr4

Висновки : За правилом відносної більшості перемагає кандидат за якого проголосувала більшість виборців, але можливий так званий «парадокс голосування» коли перемагають кандидати з менш чи 1% більшості. За правилам Кондорсе перемагає кандидат який переміг всіх в попарних порівняннях, проте можливий випадок коли переможця за даним правилом знайти неможливо. За методом альтернативних голосів на кожному етапі ми виключаємо найгірших і залишається лиш найкращий кандидат, проте можливий варіант коли буде декілька переможців за даним методом.

Лістинг програми

```
import itertools
from collections import defaultdict
document1 = open('1.txt', 'r')
profiles = []
print "Profiles: "
for line in document1:
    profiles.append([])
    for symbol in line.split():
        profiles[len(profiles) - 1].append(symbol)
    print profiles[len(profiles) - 1]
document1.close()
document2 = open('2.txt', 'r')
profiles_weight = []
for number in document2.readline().split():
    profiles_weight.append(int(number))
document2.close()
print "Profiles weight: {}".format(profiles_weight)
def relative_majority(profiles, profiles_weight):
    result = defaultdict(int)
    for i in range(len(profiles[0])):
        result[profiles[0][i]] += profiles_weight[i]
    winner = max(result, key=result.get)
    return winner
def condorcet(profiles, profiles_weight):
    profiles_tranc = zip(* profiles)
    results = defaultdict(int)
    for iter in range(len(profiles_weight)):
        for c1, c2 in itertools.combinations(profiles_tranc[iter], 2): #
combinations('ABCD', 2) --> AB AC AD BC BD CD
            assert (c1 != c2)
            mini, maxi, res = (c1, c2, profiles_weight[iter]) if c1 < c2 else
(c2, c1, -profiles_weight[iter])
            results[mini, maxi] += res
    for c in {c for profile in profiles_tranc for c in profile}:
        # TO DO understand if statement
        if (all((res > 0 if c == mini else res < 0) for (mini, maxi), res in
results.items() if c in [mini, maxi])):
            return c
    return "can't be find"
def alternative_votes_method(profiles, profiles_weight):
    result = defaultdict(int)
    profiles_count = len(profiles[0])
    last = len(profiles[0]) - 1
    while( last > 1 ):
        for iter in range(profiles_count):
            result[profiles[last][iter]] += profiles_weight[iter]
        worst = max(result, key=result.get)
        profiles_tranc = [list(i) for i in zip(* profiles)]
        for profile in profiles_tranc:
            profile.remove(worst)
        profiles = [list(i) for i in zip(* profiles_tranc)]
        result.clear()
        last -= 1
    return relative_majority(profiles, profiles_weight)
print "Winner (relative majority): {}".format(relative_majority(profiles,
profiles_weight))
print "Winner (Condorcet): {}".format(condorcet(profiles, profiles_weight))
print "Winner (method of alternative votes):
{}".format(alternative_votes_method(profiles, profiles_weight))
```