

Интеллектуальные системы и технологии (практика)

Судаков Владимир Анатольевич
доктор технических наук, профессор кафедры КБ-9,
sudakov@ws-dss.com

ИСТОЧНИКИ

- Марк Лутц. Изучаем Python
 - Luciano Ramalho. Fluent Python
 - Joel Grus. Data Science from Scratch
 - Allen B. Downey. Think Complexity
 - PEP8
-
- https://github.com/sudakov/lab_it

Давайте решим задачу

- Дан список списков: $[[1, 2, 3, \dots], [4, 5, 7, \dots], \dots]$
- Найти сумму вторых элементов всех вложенных списков: $2 + 5 + \dots$
- Предложите разные решения на Python
- Может быть есть красивые решения на других языках?
- Сколько человек выбрали, то или иное решение?
- Какое решение лучше и почему?

Попробуем силы...

- Таблица: Дата, цена, товар
- Цены есть не на любую дату. Если цена не изменялась, то записи в таблице не будет
- Пока не появилась новая цена – действует старая
- Найдем цену на заданный товар и дату

select

value

from

prices

where

good_id = 100500 and

ddate = ?????????????? '2020-02-15'

.....

Python

- Интерпретация
- Байт-код
- Встроенные типы реализованы на C
- Динамический
- Мультипарадигменный
- Эталонной реализацией Python является интерпретатор CPython. Распространяется под свободной лицензией Python Software Foundation License. Есть реализация для JVM с возможностью компиляции, CLR. PyPy использует JIT-компиляцию, которая значительно увеличивает скорость выполнения Python-программ.

Где писать код?

- Текстовый редактор
- Atom
- Jupyter Notebook <https://colab.research.google.com>
- PyCharm

А где выполнять?

- консоль
- python.org
- см. выше

PEP8

Yes:

```
i = i + 1  
submitted += 1  
x = x*2 - 1  
hypot2 = x*x + y*y  
c = (a+b) * (a-b)
```

No:

```
i=i+1  
submitted +=1  
x = x * 2 - 1  
hypot2 = x * x + y * y  
c = (a + b) * (a - b)
```

PEP8. Как правильно?

```
foo = long_function_name(var_one, var_two,  
                          var_three, var_four)
```

```
def long_function_name(  
    var_one, var_two, var_three,  
    var_four):  
    print(var_one)
```

```
foo = long_function_name(  
    var_one, var_two,  
    var_three, var_four)
```

```
foo = long_function_name(var_one, var_two,  
                          var_three, var_four)
```

```
def long_function_name(  
    var_one, var_two, var_three,  
    var_four):  
    print(var_one)
```


PEP8

- Используйте пробелы, а не табуляции
- Строка не более 79 символов

```
with open('/path/to/some/file/you/want/to/read') as file_1, \
    open('/path/to/some/file/being/written', 'w') as file_2:
    file_2.write(file_1.read())
```

PER8. Как правильно?

```
income = (gross_wages +  
          taxable_interest +  
          (dividends - qualified_dividends) -  
          ira_deduction -  
          student_loan_interest)
```

```
income = (gross_wages  
          + taxable_interest  
          + (dividends - qualified_dividends)  
          - ira_deduction  
          - student_loan_interest)
```

import this

- Красивое лучше, чем уродливое.
- Явное лучше, чем неявное.
- Простое лучше, чем сложное.
- Сложное лучше, чем запутанное.
- Плоское лучше, чем вложенное.
- Разреженное лучше, чем плотное.
- Читаемость имеет значение.
- Особые случаи не настолько особые, чтобы нарушать правила.
- При этом практичность важнее безупречности.
- Ошибки никогда не должны замалчиваться.
- Если не замалчиваются явно.
- Встретив двусмысленность, отбрось искушение угадать.
- Должен существовать один — и, желательно, *только* один — очевидный способ сделать это.
- Хотя он поначалу может быть и не очевиден, если вы не голландец.
- Сейчас лучше, чем никогда.
- Хотя никогда зачастую лучше, чем *прямо* сейчас.
- Если реализацию сложно объяснить — идея плоха.
- Если реализацию легко объяснить — идея, *возможно*, хороша.
- Пространства имён — отличная вещь! Давайте будем делать их больше!

Метод k ближайших соседей

- Метрика близости
- Голосование k ближайших соседей
- Если результат равный – то убираем самого дальнего соседа или считаем средневзвешенный голос
- Для целей обучения именно программированию не используем **Scikit-learn** или аналоги, хотя можно использовать для сравнения...

Задача

- Давайте познакомимся:
 - Какая у Вас ближайшая станция метро?
 - Что Вы пьете по утрам? Чай или Кофе?
- Разбиваемся на команды 3-4 человека:
 - Распределение ролей
 - Парное программирование
 - Подготовка исходных данных
 - Тестирование
 - Анализ – какое к лучше?
 - Показ решения
- Обсуждение
 - Какое решение лучше и почему?

Центральность в графе

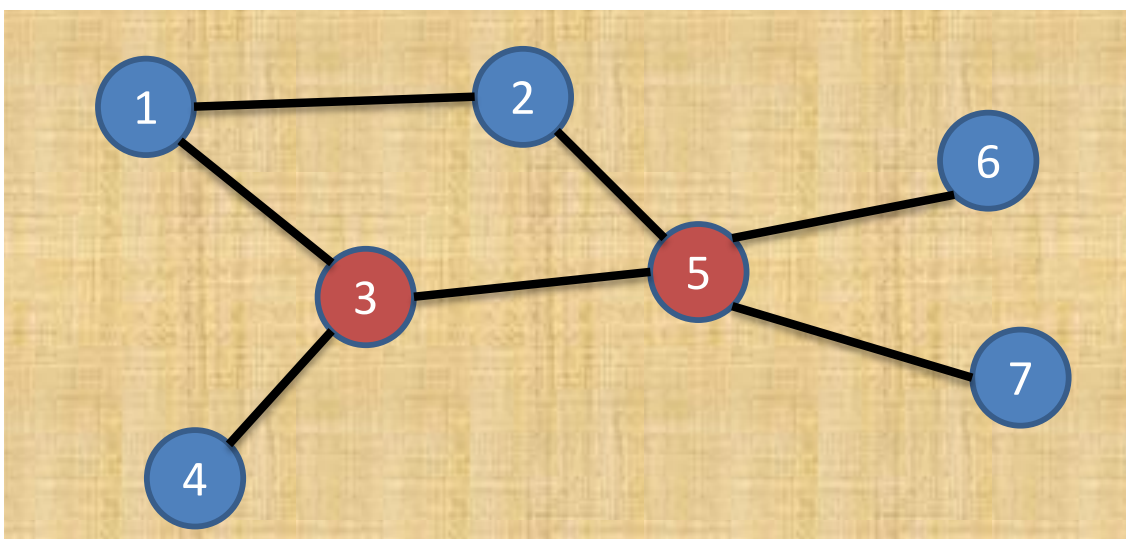
Центральность вершин в графе – это вектор, сопоставляющей каждой вершине графа некоторое число (индекс).

Наиболее распространенные индексы:

- Степенная центральность (degree centrality);
- Центральность по близости (closeness centrality);
- Центральность по посредничеству (betweenness centrality);
- Центральность по собственному вектору (eigenvector centrality);
- Центральность PageRank.

Центральность по близости

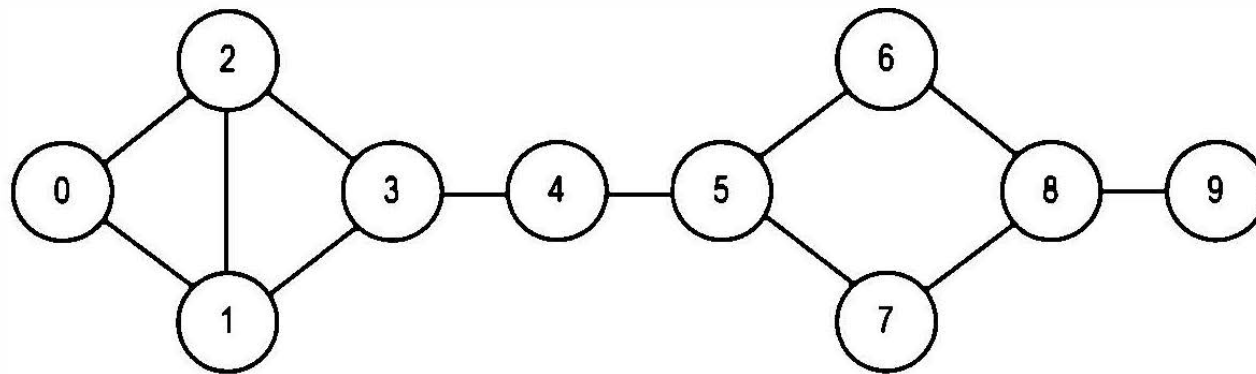
Вершина, находящаяся ближе всех к другим вершинам сети, является наиболее центральной



$$C_i = \frac{1}{\sum_j d_{ij}} \quad C_i = \sum_j \frac{1}{d_{ij}}$$

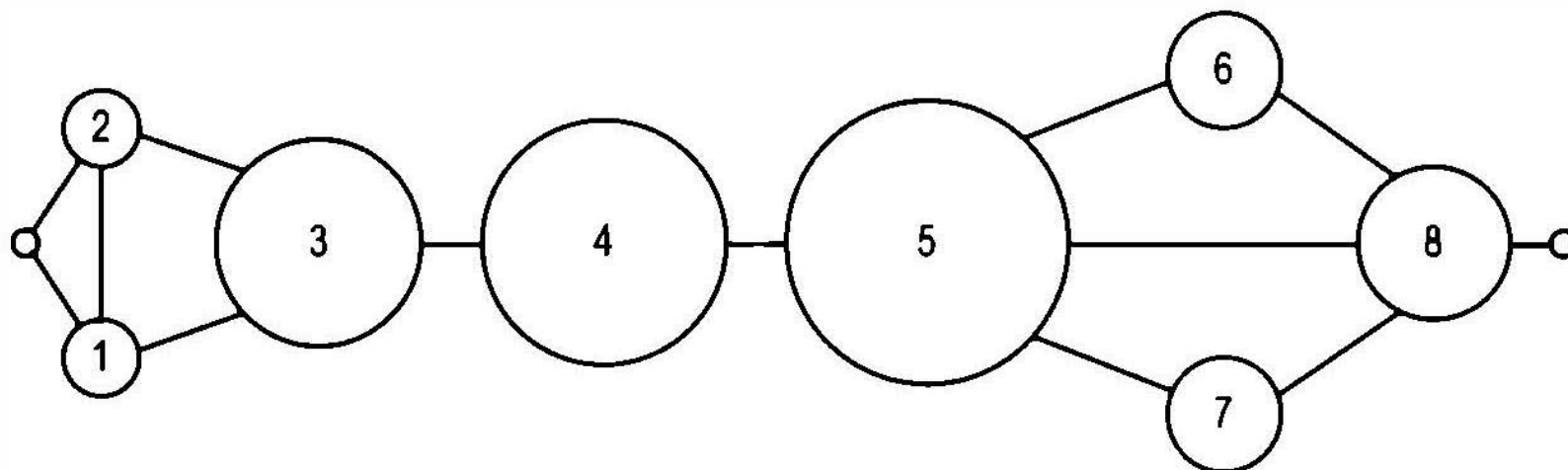
Центральность по посредничеству

Вершина, через которую проходит наибольшее число кратчайших путей, является наиболее центральной.



$$C_i = \sum_{jk} \frac{w_{jk}(i)}{w_{jk}}$$

Центральность по посредничеству



Центральность по собственному значению

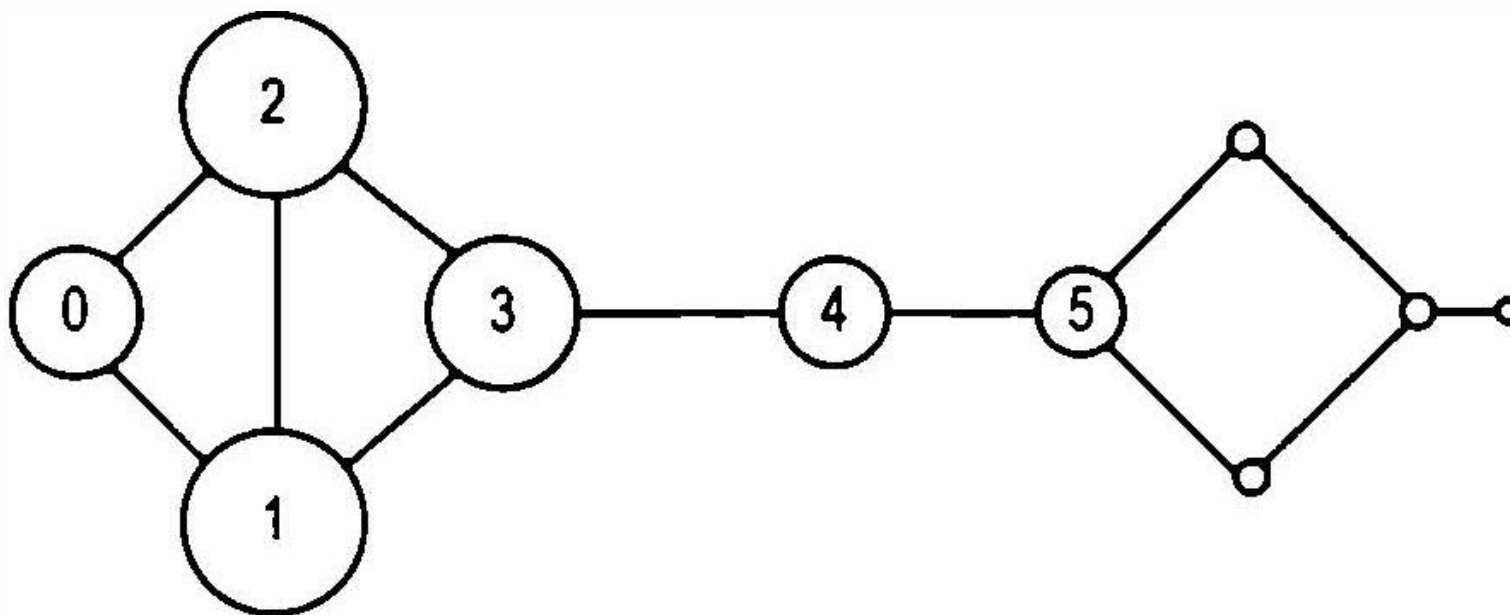
Центральность вершины i зависит от центральностей соседей вершины i .

$$x_i = \frac{1}{\lambda} \sum_{j \in F_i} x_j = \frac{1}{\lambda} \sum_j a_{ij} x_j$$

$$\lambda x = Ax$$

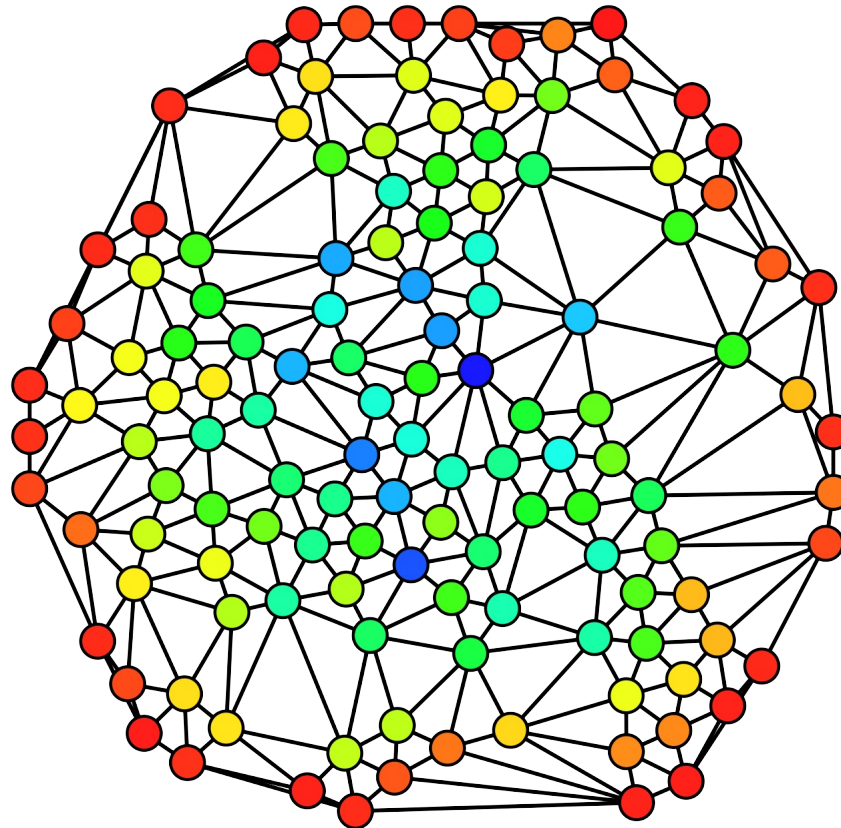
- Выбирается собственный вектор, соответствующий максимальному собственному значению.
- Данная центральность учитывает дальние взаимодействия.
- Наиболее центральными считаются вершины, которые сами указывают на сильные вершины.

Центральность по собственному значению



Задача

- Давайте соберем информацию о друзьях из VK
- Оценить центральность: по посредничеству, по близости, собственного вектора



Домашнее задание

- Реализовать модель Шеллинга (модель расовой сегрегации)
- Дан квадрат $n \times n$. 45% клеток синие, 45% клеток красные, 10% клеток пустые. Начальное заполнение в случайном порядке.
- Клетка «счастлива» если у нее 2 или более соседа одного с ней цвета. Соседи – это 8 клеток вокруг данной.
- Моделирование: выбрать случайным образом «несчастную» клетку и переместить ее в случайно выбранную пустую клетку.
- Вывести квадраты через данное некоторое количество шагов иллюстрирующее расовую сегрегацию.

В классе

- Реализовать конечный автомат

Начальное состояние автомата (вектор длины n):

0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

- **Вход:** *целое число* – определяет поведение автомата

При преобразовании в бинарный вид, например числа 1:

111	110	101	100	011	010	001	000
0	0	0	0	0	0	0	1

Нижняя строка показывает какое число нужно записать в среднюю ячейку при данном состоянии трех соседних элементов вектора.

- **Выход:** последовательность состояний вектора (одна строка – одно состояние). Выводится не более m состояний

ДЗ

- Постройте список уникальных типов самолетов зарегистрированных в России
- Какой тип самолета имеет самую раннюю дату выдачи сертификата?
- Постройте запрос: Владелец аэропорта, Аэропорт, Пассажиропоток суммарный за 2018 год, Грузопоток суммарный за 2018 год
- Перечислите аэропорты, где пассажиропоток меньше медианы, а грузопоток больше медианы
- Перечислите авиакомпании у которых нет типов воздушных судов зарегистрированных в России
- Выведите список: Месяц, суммарный пассажиропоток за данный месяц, аэропорт в котором пассажиропоток в данном месяце максимальный
- Выведите список: Тип аэропорта, средний грузопоток в месяц в аэропортах данного типа

Разместите квадраты

- Дано поле из клеточек $n * m$
- Даны квадраты. Их размеры клеточках перечислены в списке [2,3,4,1,7]
- Заполнить поле квадратами, так чтобы они не пересекались
- Чем больше квадратов поместится, тем лучше....