



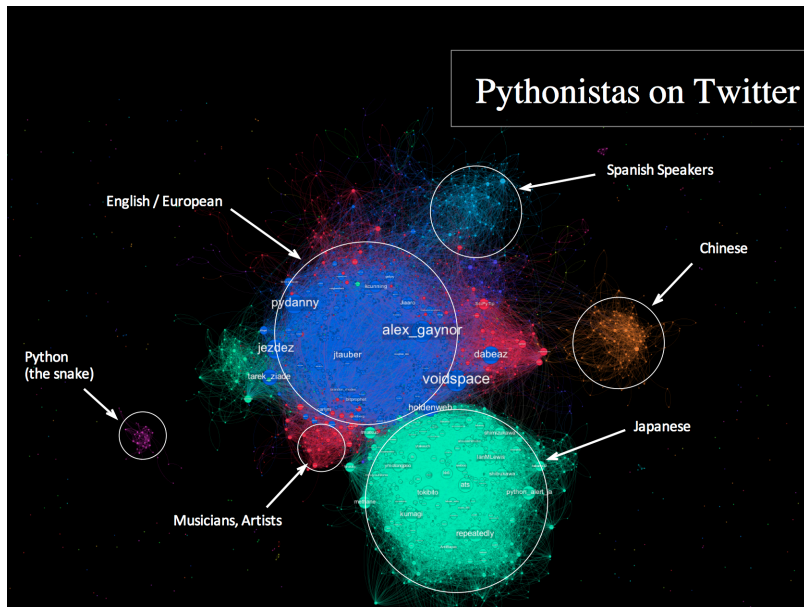
# ТЕХНОСФЕРА

## Анализ графов в социальных медиа

Ксения Стройкова

23 мая 2015 г.

# Python in Twitter



# Содержание

Постановка задачи

Исходные данные

Метрики в графе

Поиск сообществ

# Задачи анализа социальных графов

1. Социальный поиск - поиск объектов по социальным связям
2. Генерация рекомендаций - друзей или контента
3. Выявление настоящих связей
4. Лидеры мнений
5. Поиск сообществ
6. Уточнение интересов
7. Распространение новостей о событиях

# Warface

## Задачи

- ▶ Поиск игроков, которые могут покинуть игру при уходе данного игрока
- ▶ Оценка общей стоимости ухода такого игрока
- ▶ Социальные рекомендации для игроков

## Особенности

- ▶ Неявные связи
- ▶ Сессионность игры
- ▶ Данные меняются со временем

## Ближайшие цели

- ▶ Поиск сообществ
- ▶ Оценка влияния игрока на других

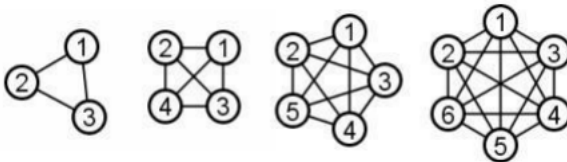
# Исходные данные

- ▶ Лог сессий игроков
- ▶ Лог сообщений игроков
- ▶ Общий лог событий (клановые события, отношения друзей)
- ▶ Лог покупок

# Граф

$$G = (V, E)$$

G- граф V - вершины E - ребра



# Метрики в графе

## Degree

$$C_d(v_i) = k_i = \text{number of neighbors}$$

## Betweenness

$$C_b(v) = \sum \frac{\sigma_{st}(v)}{\sigma_{st}} = \text{shortest paths/all paths}$$

## Closeness

$$C_c(v) = \frac{1}{\sum d(y, x)} = \text{distance to other nodes}$$



# Modularity

Метрика, оценивающая насколько внутренних связей в сообществе больше, чем внешних

$$Q = \frac{1}{2m} \sum \left( A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j)$$

- ▶  $A_{ij}$  - общее количество связей внутри сообщества
- ▶  $\frac{k_i k_j}{2m}$  - ожидаемое количество ребер между  $i$  и  $j$ , если ребра распределены случайно
- ▶  $m$  - количество ребер в графе
- ▶  $\delta = 1$ , если вершины в одном и том же сообществе
- ▶  $c$  - сообщество вершины
- ▶  $k$  - degree вершины

# Поиск сообществ

- ▶ Максимизация modularity - метрики качества структуры сообществ
- ▶ Иерархическая кластеризация
- ▶ Поиск за  $n \log n$

