**Uniwersytet Wrocławski**
**Wydział Matematyki i Informatyki**
**Instytut Matematyczny**
*specjalność: ogólna*

***Rafał Płoszka***

# Wykrywanie społeczności w grafach: metoda Louvain

Praca licencjacka
napisana pod kierunkiem
dr. Pawła Lorka

Wrocław 2019

# Spis treści

# 1 Wstęp

Metoda Louvain to algorytm służący do wykrywania społeczności w sieciach złożonych zaprezentowany przez Vincenta D. Blondela wraz ze współautorami w 2008 roku na Uniwersytecie w Louvain. Dzięki swojej intuicyjności i łatwości w implementacji jest ona szeroko stosowana w analizie sieci modelujących rzeczywiste zjawiska. W mojej pracy skupię się na dokładnym opisie metody, a ponadto zastosowaniu jej w połączeniu z systemem rekomendacji opartym na użytkowniku.

W pierwszej części zaprezentuję podstawowe definicje, tzn. sieci złożonych oraz społeczności, a następnie przejdę do sposobów ich wykrywania. Jedna z grup tychże sposobów opiera sie na optymalizacji modularności, dlatego też poświęcę temu pojęciu dużą część rozdziału, gdyż jest to kluczowa definicja do zrozumienia algorytmu.

W kolejnym rozdziale przejdę do kompleksowego opisu samej metody Louvain. Opis wzbogacę grafikami, a ponadto porównam algorytm z innymi znanymi do tej pory algorytmami służącymi do wykrywania społeczności oraz wyróżnię jego zalety i ograniczenia. Przybliżę również jego możliwe zastowania, przedstawiając szczegółowo, jak użyli tej metody sami autorzy oraz nakreślając inne drogi jego zaaplikowania.

Następnie przejdę do zaprezentowania innowacyjnego podejścia do systemów rekomendujących. W celu usprawnienia działania jednego z wariantów opartego na tzw. *Collaborative Filtering*, zbadamy strukturę społeczności danych wejściowych przy pomocy metody Louvain. Wprowadzę podstawowe pojęcia koniecznie do zrozumienia narzędzi rekomendujących oraz zbadam wzrost efektywności wynikający ze skorzystania z metody Louvain. Wynikiem pracy będzie również program napisany w języku Python.

# 2 Sieci złożone i społeczności

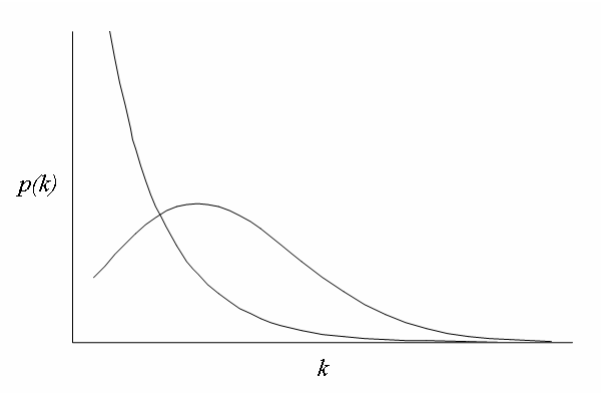
## 2.1 Sieci złożone

W ujęciu formalnym sieć złożona (*complex network*) to graf o nietrywialnych właściwościach topolgicznych, to znaczy takich, które nie występują na przykład w kratach czy grafach losowych. Grafy losowe zakładały, że połączenia w sieciach powstają losowo, dzięki czemu można było korzystać z rachunku prawodpobieństwa. Model ten został wyparty przez dwie klasy sieci złożonych: sieci bezskalowe (*scale-free networks*) oraz sieci *smallworld*.

Sieć bezskalowa to sieć, której część wierzchołków $P(k)$ mających $k$ połączeń z innymi wierzchołkami, dla dużych wartości $k$ wyraża się jako:

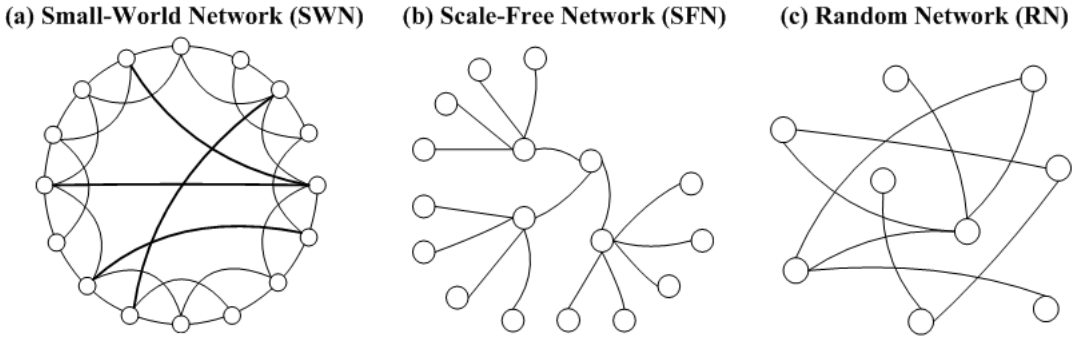
$$P(k) \sim k^{-\gamma}$$

$\gamma$ to parametr zazwyczaj z przedziału $(2, 3)$ [1]. Rozkład potęgowy liczby powiązań sprawia, że sieć nabiera własności fraktalnych, stąd nazwa sieć bezskalowa.




Rysunek 1: Powyższy wykres obrazuje różnicę rozkładów stopnia wierzchołków (*degree distribution*) w modelu bezskalowym (rozkład potęgowy) i modelu losowym (rozkład dwumianowy).

Sieć typu *smallworld network* to graf, w którym większość wierzchołków nie jest w bezpośrednim sąsiedztwie z innymi, ale sąsiedzi dowolnego wierzchołka są prawdopodobnie wzajemnymi sąsiadami oraz większość wierzchołków może być osiągnięta z każdego innego wierzchołka niewielką liczbą kroków.




Rysunek 2: Rysunki obrazujące różnice pomiędzy trzema omówionymi modelami. [2]

Badanie sieci złożonych to stosunkowo młoda nauka (około rok 2000), jednak z uwagi na to, że opisuje rzeczywiste zjawiska i ma szerokie zastosowanie, rozwija się bardzo dynamicznie.
Oto kilka przykładów sieci złożonych:

- sieć WWW; wierzchołkami są strony internetowe, a krawędzami hiperłącza. Pomimo swojego rozmiaru sieć ta posiada własność *smallworld*. Badanie z roku 2000 na 50 milionach stron internetowych pokazała, że średnia długość ścieżki tzn. liczba hiperłączy łączących dwa dokumenty wynosiła zaledwie 16. [3]

- sieć interakcji białkowych (*Protein–protein interaction networks, PPIs*); wierzchołkami są białka, a krawędzi symbolizują interakcje pomiędzy nimi. Jest to najintensywniej badany typ sieci biologicznej.

- sieci społeczne (*social networks*); składają się z osób (*social actors*) oraz relacji społecznych pomiędzy nimi, są szczególnie użyteczne w socjologii i psychologii.

Jako, że sieci złożone to rozbudowane obiekty, możemy badać je pod różnymi kątami, kilka spośród istotnych właściwości to wspomniany już rozkład stopnia wierzchołków (*degree distribution*), współczynnik grupowania (*clustering coefficient*), struktura hierarchiczna (*hierarchical structure*) oraz struktura społeczności (*community structure*), której poświęcę kolejny podrozdział.

## 2.2 Społeczności

Mówimy, że w sieci złożonej występuje struktura społeczności, jeśli można naturalnie podzielić wierzchołki grafu na grupy nazywane społeczności. Generalnie dopuszcza się, aby społeczności nakładały się na siebie, w naszych rozważaniach wykluczymy jednak ten przypadek i będziemy rozpatrywać jedynie *non-overlapping communities*.

Możemy zatem doprezycować naszą definicję: społeczności są powiązane wewnętrznie gęściej aniżeli pomiędzy sobą. Pod roboczym pojęciem gęstości mam na myśli sumę wag krawędzi lub liczbę krawędzi w grafach odpowiednio: ważonych i nieważonych.

Wykrycie społeczności w grafie daje szereg korzyści, przede wszystkim, dzięki temu jesteśmy w stanie poznać własności występujące jedynie w pewnej grupie wierzchołków, a nie globalnie. I tak na przykład w sieci interakcji białkowych społeczności odpowiadają białkom o podobnych funkcjach w komórce biologicznej, a w sieci cytatów grupowanie następuje według tematu badań.

Istnienie społeczności wpływa ponadto istotnie na takie procesy jak rozprzestrzenianie się informacji (sieci społeczne) czy epidemii (sieci biologiczne). Dodatkowo, efektywne wyszukanie społeczności może pomóc w odnalezieniu brakujących połączeń czy też identyfikacji fałszywych. [4]

Widzimy zatem, że dzięki tego typu algorytmom, możemy lepiej zrozumieć działanie poszczególnych fragmentów sieci złożonych, ale także tego jak dane grupy na siebie wpływają i jakie przełożenie ma to na funkcjonowanie całego grafu. To wszystko sprawia, że narzędzia pozwalające na odnalezienie społeczności w grafach są bardzo cenne. Samo zagadnienie poszukiwania często

jest trudne obliczeniowo, ze względu na rozmiar sieci złożonych oraz fakt, iż zazwyczaj nie znamy z góry liczby, wielkości czy gęstości społeczności.

W ostatnich latach odkryto jeszcze jeden aspekt struktury społeczności. Otóż okazuje się, że gdy gęstości połączeń wewnątrz społeczności oraz pomiędzy nimi stają się coraz bliższe sobie (tzn. gdy struktura społeczności staje się zbyt słaba lub sieć staję się zbyt luźna), w pewnym momecie społeczności przestają być wykrywalne. [5] W pewnym sensie dalej istnieją, jako że obecność lub brak krawędzi wciąż są skorelowane z przynależnością ich wierzchołków do poszczególnych społeczności. Staje się jednak teorytycznie niemożliwe, aby pogrupować wierzchołki lepiej niż w sposób losowy czy nawet odróżnić sieć od grafu losowego pozbawionego struktury społeczności. To przejście jest niezależne od typu algorytmu używanego do wykrywnia społeczności, co oznacza, że istnieje pewna granica naszej zdolności do wykrywania społeczności w sieciach.

## 2.3 Modularność

Modularność (*modularity*) jest miarą struktury sieci złożonej. Sieć o wysokiej modularności posiada gęste połączenia między wierzchołkami wewnątrz modułów i rzadkie pomiędzy modułami. Wyraźne podobieństwo do definicji społeczności wynika z faktu, iż główną motywacją do wprowadzenia pojęcia modularności było stworzenie miary, której maksymalizacja posłuży do wykrywania społeczności w grafach.

Modularność przyjmuje wartość z zakresu $[-1, 1]$. Jest dodatnia, gdy suma wag krawędzi wewnątrz grup jest wyższa w modelu rzeczywistym niż w modelu losowym, tj. gdyby rozłożenie krawędzi byo dobierane losowo przy założeniu niezmienności stopni wierzchołków. Dla zadanego podziału sieci modularność odzwierciedla zagęszczenie krawędzi wewnątrz modułów tegoż podziału w porównaniu do sytuacji, gdyby krawędzie były rozmieszczone losow pomiędzy wszystkie wierzchołki grafu nie zważając na moduły.

Rozważmy nieskierowany graf ważony $G = (V, E)$ z zadanym podziałem na społeczności. Wzór na modularność takiego grafu to: [6]

$$Q = \frac{1}{2m} \sum_{i,j} \left[ A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

gdzie:

- $A$ jest macierzą sąsiedztwa grafu $G$, tzn. $A_{ij}$ wyraża wagę krawędzi pomiędzy wierzchołkami $i$ i $j$,
- $k_i = \sum_j A_{ij}$ jest sumą wag krawędzi dołączonych do wierzchołka $i$,
- $m = \frac{1}{2} \sum_{i,j} A_{ij}$ jest sumą wag wszyskich krawędzi w grafie,
- $\delta(c_i, c_j)$ wynosi 1, gdy wierzchołki $i$ i $j$ należą do tej samej społeczności lub 0 w przeciwnym wypadku.

Zastanówmy się nad znaczeniem wyrażenia pod znakiem sumy. Pamiętając, iż $k_i$ to suma wag krawędzi wierzchołka $i$, zauważmy, że $\frac{k_j}{2m}$ jest średnią częścią

tej wagi, która byłaby przypisana do wierzchołka $j$, gdyby wierzchołek $i$ przypisał sumę wag swoich krawędzi innym wierzchołkom w sposób losowy, przy zachowaniu proporcji do wag ich własnych połączeń.
Zatem rożnica $A_{ij} - \frac{k_i k_j}{2m}$ pokazuje, jak mocno powiązane są wierzchołki $i$ i $j$ w rzeczywistej sieci w porównaniu do tego jak mocne byłoby to połączenie w sieci losowej opisanej powyżej.

Alternatywnie modularność można wyrazić następująco: [7]

$$Q = \sum_{i=1}^{c} \left(e_{ii} - a_i^2\right)$$

gdzie:

- $e_{ij} = \sum_{v \in c_i w \in c_j} \frac{A_{vw}}{2m}$ - część krawędzi, których jeden wierzchołek należy do społeczności $i$, a drugi do społeczności $j$

- $a_i = \frac{k_i}{2m}$ - suma wag krawędzi, które połączone są z wierzchołkami należącymi do społeczności $i$

Modularność ma pewne ograniczenie. Przypomnijmy, że modularność porównuje liczbę krawędzi wewnątrz pewnej grupy wierzchołków z oczekiwaną liczbą krawędzi, jeśli sieć byłaby siecią losową o tej samej liczbie wierzchołków i gdzie każdy wierzchołek zachowuje swój stopień, ale krawędzie są w inny sposób przypadkowo dołączone. Ten losowy model zerowy zakłada, że każdy wierzchołek może zostać przyłączony do dowolnego innego wierzchołka w sieci.
Założenie to jest jednak nieuzasadnione, jeśli sieć jest bardzo duża, ponieważ zasięg wierzchołka obejmuje niewielką część sieci, ignorując większość innych wierzchołków. Co więcej, oznacza to, że oczekiwana liczba krawędzi między dwiema grupami węzłów zmniejsza się, jeśli zwiększa się rozmiar sieci. Jeśli więc sieć jest wystarczająco duża, oczekiwana liczba krawędzi między dwiema grupami wierzchołków w modelu zerowym modularności może być mniejsza niż jeden. Gdyby tak się stało, pojedyncza krawędź między dwoma grupami byłaby interpretowana przez modularność jako znak silnej korelacji między dwoma grupami, a optymalizacja modularności prowadziłaby do połączenia dwóch grup, niezależnie od ich cech. Tak więc nawet słabo połączone grafy pełne, które mają najwyższą możliwą gęstość wewnętrznych połączeń i

reprezentują najlepiej rozpoznawalne społeczności, zostałyby połączone poprzez optymalizację modularności, gdyby sieć była wystarczająco duża. Powrócę do tego problemu przy samej charakterystyce metody Louvain.

# 3 Metoda Louvain

## 3.1 Opis

Wracając do pierwszej definicji modularności, problem optymalizacji modularności można traktować jako grupowanie w społeczności taki sposób, że elementy sumy były jak największe (pamiętając, że funkcja delta zeruje elementy w sumie odpowiadające parom wierzchołków należących do różnych społeczności). Generuje to jednak pewien problem. Rozpatrzmy prosty przykład: na drodze obliczeń dochodzimy do wniosków, że wierzchołki 1 i 2 powinny należeć do jednej społeczności, a ponadto wierzchołki 2 i 3 również powinny być w jednej społeczności. Jednakże może zdarzyć się tak, że wartość $A_{13} - \frac{k_1 k_3}{2m}$ jest ujemna, co oznacza, że 1 i 3 nie powinny znajdować się w jednej społeczności. Nawet tak mało złożony przykład pokazuje, że optymalizacja modularności jest trudnym zagadnieniem, a uściślając NP-trudnym [8]. W związku z tym stosuje się heurestyki. Jedną z nich jest metoda Louvain. Jest to to tzw. algorytm zachłanny (*greedy algorithm*), tzn. podejmujący decyzję optymalną lokalnie, wydającą się najlepszą w danej w chwili, bez skupiania się na dalszych krokach. Przejdźmy zatem do opisu samej metody.

Metoda Louvain składa się z dwóch kroków powtarzanych iteracyjnie:

1. Początkowo zakładamy, iż każdy wierzchołek grafu ważonego jest osobną społecznością. Następnie dla każdego wierzchołka $i$ rozważamy jego sąsiadów $j$ i obliczamy, czy przeniesienie wierzchołka $i$ do społeczności wierzchołka $j$ spowoduje wzrost modularności rozważanej sieci złożonej. Umieszczamy wtedy wierzchołek $i$ w tej społeczności, przy której nastąpi największy wzrost modularności. Jeżeli żadne z możliwych przeniesień wierzchołka $i$ nie zwiększy modularności, $i$ zostaje w swojej społeczności.
   Takim sposobem rozważamy każdy wierzchołek w sieci, dopóki żaden wzrost modularności nie jest możliwy. Warto w tym miejscu zaznaczyć, ze podczas kroku pierwszego naszego algorytmu zazwyczaj będziemy

rozważać jeden wierzchołek więcej niż jeden raz. Nasuwa się tutaj pytanie, jak zdecydować o kolejności rozważania wierzchołków. Według autorów metody, wstępne badania wykazały, iż kolejność doboru wierzchołków nie wpływa znacząco na ostatecznie uzyskaną modularność. Jednak optymalizacja kolejności może wpłynąć na czas obliczeń. Jest to zatem jeden z aspektów algorytmu, który warto byłoby zgłębić.
Wspomniany powyżej wzrost modularności przy przesuwaniu pojedynczego wierzchołka $i$ do społeczności $C$ może zostać w prosty sposób obliczony następującym wzorem:

$$\Delta Q = \left[\frac{\sum_{in}+k_{i,in}}{2m} - \left(\frac{\sum_{tot}+k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - (\frac{k_i}{2m})^2\right] \tag{1}$$

gdzie:

- $\sum_{in} = \sum_{i,j\in C} A_{ij}$ oznacza sumę wag wewnątrz społeczności $C$
- $\sum_{tot} = \sum_{i,j} A_{ij} 1_{i\in C}$ to suma wag krawędzi przyległych do wierzchołków w $C$
- $k_i = \sum_{j} A_{ij}$ suma wag krawędzi przyległych do wierzchołka $i$
- $k_{i,in} = \sum_{j\in C} A_{ij}$ suma wag krawędzi łaczących wierzchołek $i$ z wierzchołkami wewnątrz społeczności $C$
- $m = \frac{1}{2}\sum_{i,j} A_{ij}$ jest sumą wag wszystkich krawędzi w sieci

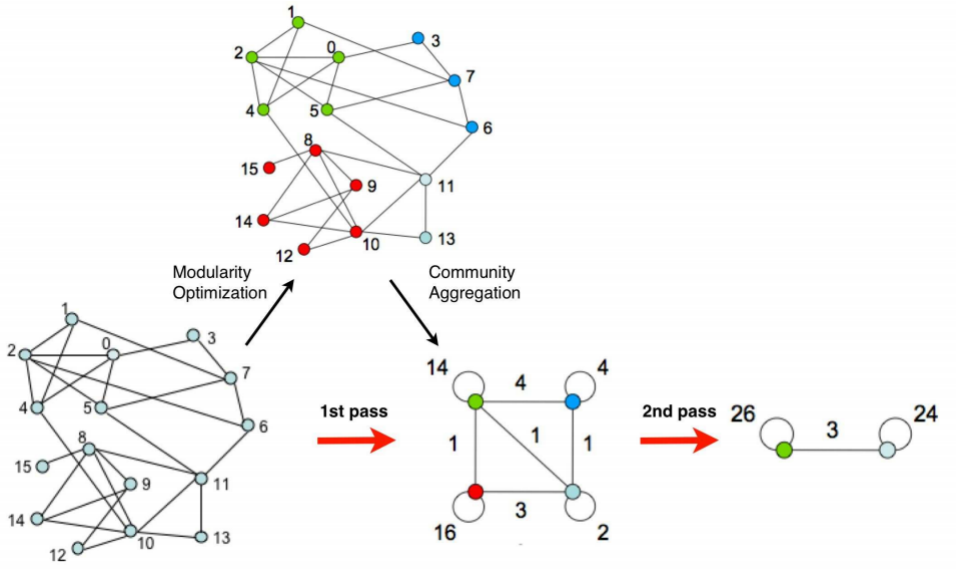
2. W drugim kroku za wierzchołki przyjmujemy społeczności powstałe w kroku pierwszym. W związku z tym waga krawędzi pomiędzy nowymi wierzchołkami to suma wag krawędzi pomiędzy wierzchołkami należącymi do społeczności z pierwszego kroku. Natomiast krawędzie pomiędzy wierzchołkami wewnątrz jednej społeczności zamieniane są na pętle o odpowiedniej wadze. Taka transformacja sieci umożliwia nam ponowne przeprowadzenie pierwszego kroku.

Jednokrotne wykonanie obu kroków nazywać będziemy przejściem. Przejścia powtarzamy iteracyjnie do momentu, aż żadne modyfikacje z kroku pierwszego nie zwiększają modularności. Takie podejście umożliwa nam oprócz optymalizacji modularności, odkrycie hierarchii społeczności. Z każdym przejściem ujawniany jest kolejny poziom hierarchii. Jak zobaczymy w później-

szych rozważaniach, liczba przejść nawet w sieciach o bardzo dużych rozmiarach jest niewielka.

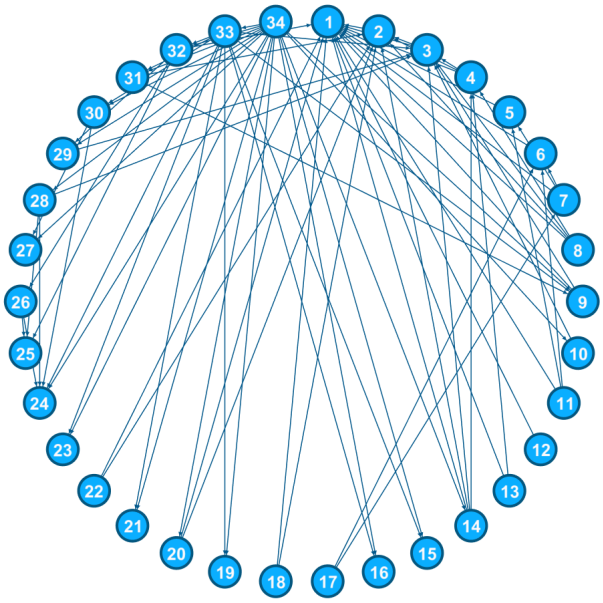
## 3.2 Przykłady

1. Prosty przykład autorów metody [6]
   Na poniższym rysunku widzimy zastosowanie metody Louvain do nieważonego grafu o 15 wierzchołkach. W pierwszej fazie pierwszego przejścia grupujemy wierzchołki w społeczności, co zostało zaznaczone kolorami. Następnie tworzymy nowy graf, gdzie wierzchołki reprezentują społeczności, a krawędzie powiązania wewnątrz i pomiędzy społecznościami. I tak na przykład 3 ma jedno połączenie z 7, 7 ma po 1 połączenie z 3 i 6, a 6 ma jedno połączenie z 7, co łącznie daje 4, stąd waga pętli przy niebieskim wierzchołku w kroku 2 wynosi 4. Natomiast wierzchołki 3, 7, 6 łączy tylko jedna krawędź z wierzchołkami 11 i 13 i dlatego krawędź pomiędzy niebieskim a błęktinym wierzchołkiem w kroku drugim ma wagę 1.




2. *Zachary's karate club*
   Jest to przykład prostej sieci społecznej, powszechnie używany do badania struktury społeczności w grafach.
   Składa się z 34 wierzchołków rezprezentujących członków akademickiego klubu karate oraz 76 krawędzi odpowiadających interakcjom między

parami członków poza klubem.



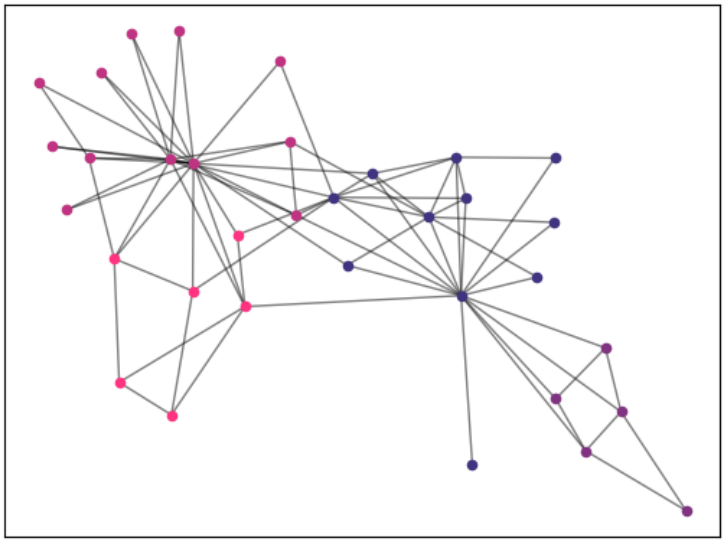

Rysunek 3: Graf przedstawiający *Zachary's karate club* [9]

Korzystając z modułu Python *igraph* otrzymujemy 2 poziomy hierarchii struktury społeczności:

```
Clustering with 34 elements and 6 clusters
[0] 4, 10
[1] 2, 3, 7, 9, 12, 13
[2] 5, 6, 16
[3] 0, 1, 11, 17, 19, 21
[4] 23, 24, 25, 27, 28, 31
[5] 8, 14, 15, 18, 20, 22, 26, 29, 30, 32, 33

Clustering with 34 elements and 4 clusters
[0] 4, 5, 6, 10, 16
[1] 0, 1, 2, 3, 7, 9, 11, 12, 13, 17, 19, 21
[2] 23, 24, 25, 27, 28, 31
[3] 8, 14, 15, 18, 20, 22, 26, 29, 30, 32, 33
```

Zatem widzimy, iż po pierwszym przejściu graf zostaje podzielony na 6 społeczności. W drugim przejściu liczba społeczności spada do 4. Przy trzeciej próbie algorytm zatrzymuje się i otrzymujemy graf z 4 społecznościami oraz modularnością 0,42.
Oczywiście z uwagi na niewielki stopień rozbudowania grafu ciężko precyzyjnie badać przy jego pomocy czas pracy i skuteczność algorytmów do wykrywania społeczności. Pozwala on jednak przystępnie zwizualizować istotę działania metody.



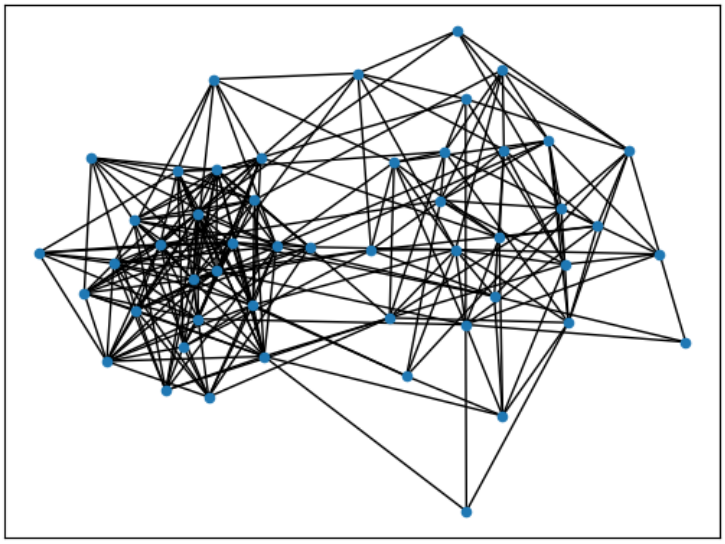
Rysunek 4: Przy pomocy modułu NeworkX w Pythonie możemy przedstawić podział na społeczności na grafie (kolory odpowiadają społecznościom, rozmieszczenie wierzchołków jest losowe)

3. Stochastyczny model blokowy
   Stochastyczny model blokowy (*stochastic block model, SBM*) to model generujący graf z zadanym podziałem na grupy, co umożliwia m.in. testowanie algorytmów wykrywających społeczności [10]. SBM przyjmuje następujące parametry: $n$ wierzchołków pogrupowanych rozłącznie na $r$ społeczności oraz symetryczną macierz $P$ o wymiarach $r \times r$. Krawędzie w grafie tworzone są wg reguły: 2 dowolne wierzchołki $u$ i $v$ z

różnych społeczności $C_i$ i $C_j$ ($u \in C_i$ i $v \in C_j$) są połączone krawędzią z prawdopodobieństwem $P_{ij}$. Rozważmy zatem przykładowy stochastyczny model blokowy zadany macierzą:

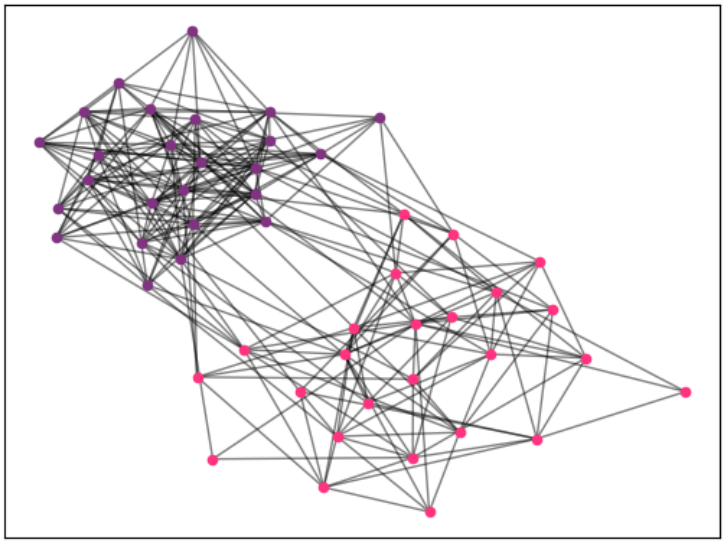
$$P = \begin{bmatrix} 0.5 & 0.05 \\ 0.05 & 0.25 \end{bmatrix}$$

składający się z 2 klastrów, po 25 wierzchołków każdy. Z macierzy możemy odczytać, że między wierzchołkami wewnątrz pierwszej grupy krawędzie występują z prawdopodobieńswem 0.5, zaś między wierzchołkami w grupie drugiej z prawdopodobieństwem 0.25. Wyraźnie widać to na poniższym wykresie: wierzchołki po lewej stronie są znacznie bardziej zagęszczone niż te po prawej. Ponadto prawdopodobieństwo istnienia krawędzi pomiędzy wierzchołkami różnych społeczności wynosi 0.05.



Rysunek 5: Wykres SBM stworzony za pomocą modułu NetworkX

Po zastosowaniu metody Louvain otrzymujemy graf podzielony na 2 społeczności, co widoczne jest na rysunku nr 6.



Rysunek 6: Podział grafu utworzonego modelem SBM na 2 społeczności oznaczone kolorami.

> Warto w tym miejscu zastanowić się nad szczególnym przypadkiem, gdy wszystkie wartości w macierzy są równe. Otrzymujemy wtedy graf losowy (model Erdős–Rényi) i rozważanie w nim społeczności traci sens, ponieważ prawdopodobieństwo, że wierzchołki wewnątrz grup i pomiędzy grupami są połączone jest równe, zatem nie można mówić w takiej sytuacji o strukturze społeczności.

## 3.3 Zalety i ograniczenia

Po dogłębnym zrozumieniu samej metody możemy przejść do omówienia jej największych plusów i minusów.
Przede wszystkim sam algorytm jest intuicyjny i prosty w implementacji. Ponadto cechuje się nadzwyczajną szybkością. Wynika to z faktu, że wzrost

modularności można łatwo obliczyć za pomocą wzoru (1). Ponadto liczba społeczności drastycznie spada po zaledwie kilku przejściach, a zatem większość czasu obliczeń przypada na pierwsze iteracje.

Metoda Louvain napotyka na dwa ograniczenia.

1. Tzw. ograniczenie rozdzielczości (*resolution limit*), jak wspomniałem w podrozdziale poświęconemu modularności jest to problem dotyczący każdego sposobu optymalizacji modularności. Sprowadza się do tego, że w sieciach o bardzo dużych rozmiarach algorytmy tego typu mają trudności z wykrywaniem małych społeczności. Dzięki wielpoziomowemu podejściu metody Louvain problem ten jest częściowo omijany. W pierwszej fazie algorytmu następuje przemieszczenie pojedynczych węzłów z jednej społeczności do innej, zatem prawdopodobieństwo, że dwie społeczności zostaną połączone w jedną jest stosunkowo niskie. Społeczności te mogą być połączone w późniejszych przejściach, jednak wstępne badania wskazują, że rozwiązania pośrednie znalezione przez algorytm mogą również mieć znaczenie oraz że odkryta hierarchiczna struktura daje możliwość powiększenia sieci i obserwowania jej struktury z pożądaną rozdzielczością.

2. Problem „degeneracji" (*degeneracy problem*). Zazwyczaj istnieje duża liczba przypisań społeczności z modularnością bliską maksimum. Może to stanowić problem, ponieważ w obecności dużej liczby rozwiązań o wysokiej modularności trudno jest znaleźć globalne maksimum oraz trudno określić, czy globalne maksimum jest faktycznie bardziej istotne niż lokalne maksima z podobną modularnością [11].

Widzimy zatem, iż z powodu oparcia metody o maksymalizację modularności nie sposób uniknąć pewnych ograniczeń, natomiast algorytm zdecydowanie przewyższa inne znane metody pod względem szybkości działania, co pokażę w kolejnym podrozdziale.

## 3.4 2.4 Porównanie

Początkowo optymalizacja modularności opierała się na metodzie podziału

autorstwa Newmana i Girvana (2002) [12], gdzie stopniowo usuwa się krawędzie z sieci, a tak zmodyfikowana sieć ujawnia strukturę społeczności. Jednak z czasem rozwinięto szybsze metody, tzw. metody aglomeracyjne (iteracyjne grupowanie wierzchołków). Są to Clauset, Newman and Moore (CNM) [13], Pons and Latapy (PL) [14], Wakita and Tsurumi (WT) [15] oraz metoda Louvain (our algorithm).

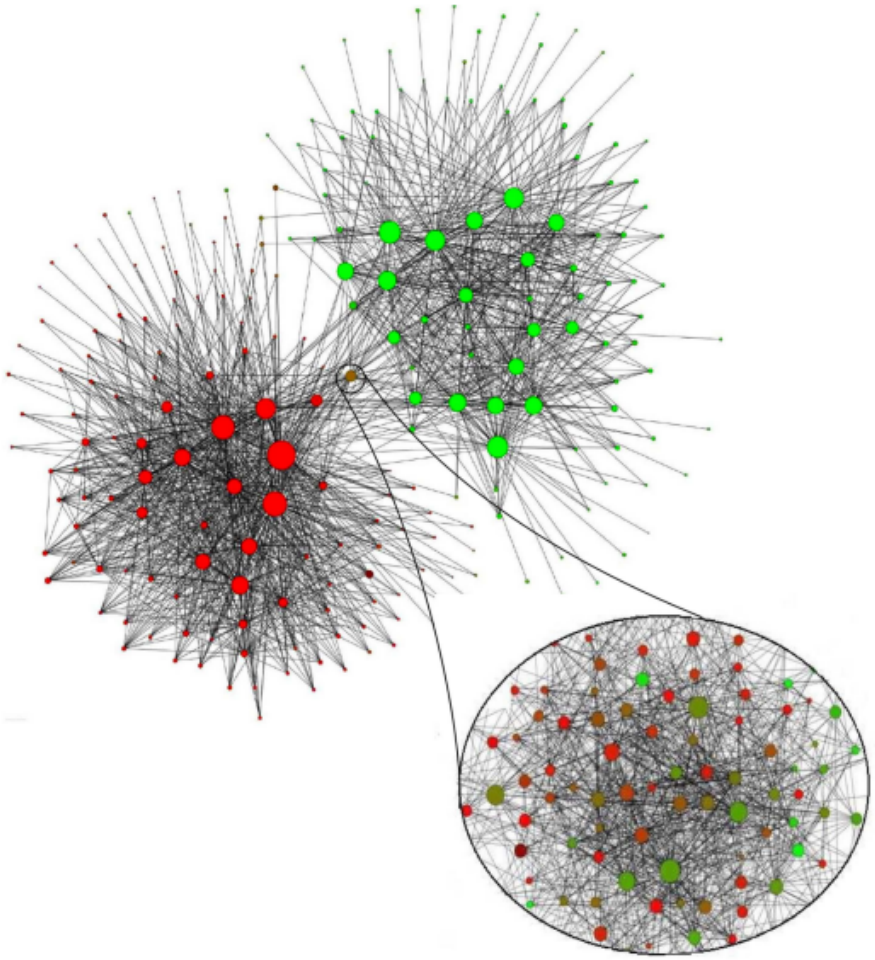
| | Karate | Arxiv | Internet | Web nd.edu | Phone | Web uk-2005 | Web WebBase 2001 |
|---|---|---|---|---|---|---|---|
| Nodes/links | 34/77 | 9k/24k | 70k/351k | 325k/1M | 2.6M/6.3M | 39M/783M | 118M/1B |
| CNM | .38/0s | .772/3.6s | .692/799s | .927/5034s | -/- | -/- | -/- |
| PL | .42/0s | .757/3.3s | .729/575s | .895/6666s | -/- | -/- | -/- |
| WT | .42/0s | .761/0.7s | .667/62s | .898/248s | .56/464s | -/- | -/- |
| Our algorithm | .42/0s | .813/0s | .781/1s | .935/3s | .769/134s | .979/738s | .984/152mn |

W powyższej tabeli [6] zebrano wyniki porównania tych metod na kilku przykładów sieci złożonych. W wierszu *Nodes/links* przedstawione zostały liczby odpowiedno wierzchołków i krawędzi w grafach. W komórkach wewnątrz tabeli zaprezentowana obliczoną modularność oraz czas działania ('-' odpowiada czasowi przekraczającemu 24 godziny).
Karate to opisywany już klub karate. Z uwagi na niską złożoność modelu trudno tutaj rozpatrywać czas działania. Modularność także jest taka sama, nie licząc metody CNM.
Arxiv to elektroniczne archiwum naukowych preprintów. Sieć ta w dalszym ciągu nie jest imponujących rozmiarów, ale już tutaj widać wyraźną przewagę w czasie działania i wykrywanej modularności. Tendencja ta pogłębia się przy dwóch kolejnych badanych obiektach.
*Phone* oznacza sieć telefonów komórkowych, której poświęcony będze kolejny rozdział. Jest to już sieć bardzo złożona, dwie spośród omawianych metod potrzebują ponad 24 godzin na analizę tego grafu, dlatego trudno mówić o ich użyteczności w takich przypadkach. Inną ciekawą obserwacją jest istotna różnica w obliczonej modularności pomiędzy metodą Louvain a WT. Może ona wynikać z tendencji algorytmu WT do tworzeniu zrównoważonych społeczności, wady, której pozbawiona jest metoda Louvain.
Przy dwóch ostatnich modelach porównywane metody nie są wystarczające, natomiast algorytm Louvain zwraca wysoką modularność oraz działa w bardzo dobrym czasie, biorąc pod uwagę wielomilionowy rozmiar sieci.

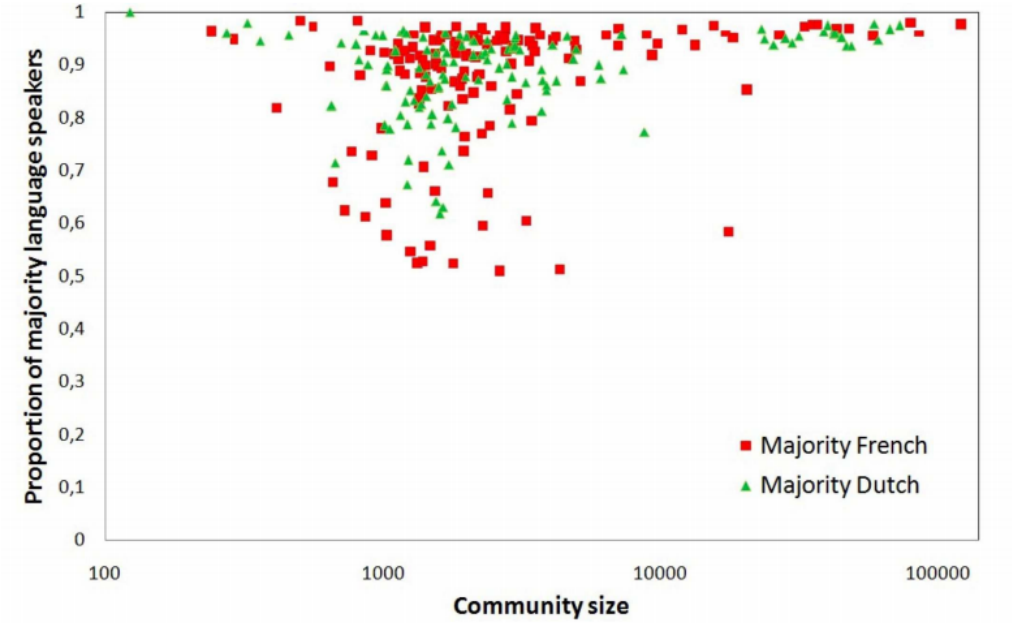
## 3.5 Zastosowanie do sieci komórkowej

Autorzy metody zastosowali ją do zbadania belgijskiej sieci komórkówej. Sieć składała się z 2 milionów wierzchołków reprezentujących użytykowników sieci komórkowej oraz krawędzi, których wagi odpowiadały liczbie połączęń pomiędzy danymi klientami przez okres 6 miesięcy. O każdym kliencie znanych było kilka informacji, takich jak wiek, płeć, kod pocztowy, oraz język, którym sie posługiwał. Właśnie ta ostatnia informacja będzie jednym z badanych aspektów. Warto tutaj zaznaczyć, że Belgia jest krajem dwujęzycznym (język francuski i holenderski), dzięki czemu jest to doskonały przykład na przetestowanie metody wykrywania społeczności mając na uwadze ich jednorodność językową.

Algorytm wykrył 6 poziomów hierarchii. Początkowo każdy z 2 milionów wierzchołków jest osobną społecznością, a finalnie otrzymujemy 261 społeczności. Odrzucamy przy tym społeczności mniejsze niz 100 wierzchołków. Jest to uzasadnione założenie z uwagi na rozmiar sieci, a po odrzuceniu małych społeczności rozważamy około 75% użytkowników sieci komórkowej. Metoda Louvain ujawnia silny podział sieci na społeczności ze względu na język. Spośród 37 społeczności składających się z co najmniej 10000 członków, aż 36 to społeczności niemal jednojęzyczne, uściślając 85% (lub więcej) ich użytkowników posługuję się jednym językiem. Szczegółowo dane ta zostały zaprezentowane na rysunku nr 8. Ponadto wykres nr 7 wyraźnie ukazuje, że społeczności francuskojęzyczne są znacznie bardziej powiązane niż społeczności posługujące się głównie językiem holenderskim, różnica ta to średnio 54%. Może to wynikać to z różnic zachowań społecznych i sugeruje istnienie innych powiązań między topologicznymi właściowościami sieci, a aspektami socjologicznymi.



Rysunek 7: Rysunek obrazujący podział sieci komórkowej na społeczności. [6] Wierzchołki odpowiadają społecznościom uzyskanym po zastsowaniu metody Louvain. Rozmiar wierzchołków jest proporcjonalny do liczebności społeczności (zaprezentowane zostały społeczności, które zawierały przynajmniej 100 członków). Kolory wierzchołków odpowiadają dominującemu językowi, którym posługiwali się członkowie społeczności. Kolor czerwony oznacza język francuski, kolor zielony- holenderski. Przybliżenie pokazuje część społeczności, w których podział ze względu na język nie jest tak wyraźny.




Rysunek 8: Wykres przedstawiający jednorodność językową w społecznościach przynajmniej stuosobowych. [6] Oś pionowa oznacza jaka część członków danej społeczności mówi w dominującym języku. Oś pozioma to liczebność społeczności. Kolory ponownie symbolizują język, czerwony- francuski, zielony-holenderski. Oprócz wspomnianego już powiązania między struturą społeczności a jednorodnością językową, na wykresie możemy zauważyć, iż zdecydowane większość społeczności składa się z 1000-10000 użytkowników. Ponadto społeczności holenderskojęzyczne są w większym stopniu monojęzyczne, we wszystkich za wyjątkiem czterech, przynajmniej 70% ich członków mówi w jednym języku.

# 4 Zastosowanie w systemach rekomendacji

## 4.1 Systemy rekomendacji

Systemy rekomendacji (*recommender systems*) to narzędzia programistyczne służące do przewidywania preferencji użytkowników (*users*) co do pewnych elementów (*items*). Firmy takie jak Amazon, Netflix czy Youtube wykorzystują systemy rekomendujące, aby pomóc użytkownikom odkrywać nowe i istotne produkty (takie jak filmy, pracę, muzykę), tworząc udogodnienia dla użytkownika przy jednoczesnym zwiększaniu przychodów [16].

Ciekawostką jest, że jednym z czynników, który spowodował wzrost zainteresowania systemami rekomendacji był konkurs ogłoszony przez Netflixa. Był to otwarty konkurs trwający miedzy rokiem 2006 a 2009, w którym zadaniem było stworzenie systemu rekomendacji na bazie 100 milionów filmów o 10% efektywniejszego, niż ten używany przez Netflixa. Nagroda w wysokości miliona dolarów została wręczona jednemu zespołowi, jednak wiele pozostałych projektów okazało się być przełomowymi i znalazło zastosowanie w innych dziedzinach.

Przejdę teraz do dokładniejszego opisu systemów rekomendacji, ich funkcji oraz rodzajów. Trzy podstawowe pojęcia to:

- elementy (*items*) to obiekty podlegające rekomendacjom. Elementy mogą być scharakteryzowane przez ich złożoność i ich wartość lub użyteczność. Wartość przedmiotu może być dodatnia, jeśli przedmiot jest użyteczny dla użytkownika lub ujemna, jeśli przedmiot nie jest odpowiedni i użytkownik podjął niewłaściwą decyzję. Zauważamy, że gdy użytkownik nabywa przedmiot, zawsze ponosi koszt, który obejmuje poznawczy koszt wyszukiwania i rzeczywisty koszt pieniężny.
- użytkownicy (*users*) mogą mieć bardzo zróżnicowane cele i cechy. W celu spersonalizowania rekomendacji i interakcji komputer-człowiek, systemy rekomendacji wykorzystują szereg informacji o użytkownikach. Te

informacje mogą być skonstruowane na różne sposoby, a wybór informacji zależy od techniki rekomendacji.

- ponadto stosuje się jeszcze termin transakcji (*transaction*), który odnosi się do zarejestrowanej interakcji pomiędzy użytkownikiem a systemem. Na przykład, dziennik transakcji może zawierać odniesienie do elementu wybranego przez użytkownika lub rekomendacji. Bardzo często transakcja uwzględnia również informację otrzymaną od użytkownika, jak na przykład ocena elementu.

Systemy rekomendacji są cenione ze względu na wiele swoich funkcji, główne z nich to:

- zwiększenie liczby sprzedawanych produktów. Nie ulega wątpliwości, że wspomniane firmy korzystające z sytemów rekomenduacji są nastawione na maksymalizację zysków, w związku z czym można uznać tę funkcję za kluczową w kwestii praktycznego zastosowania. Ponadto dobrze skonstruowany system rekomendacji potrafi poszerzyć grono sprzedawanych produktów. Dzięki takiemu dopasowaniu mniej popularne produkty mogą być skutecznie sprzedawane.

- satysfakcja użytkownika. Dobrze zaprojektowany system rekomendacji może również poprawić doświadczenia użytkownika ze stroną lub aplikacją. prawdopodobieństwo, że zalecenia zostaną przyjęte

- większe „przywiązanie" użytkownika do strony/aplikacji. Jest to obupulna korzyść, gdyż im dłużej użytkownik pozostaje w interakcji z systemem, tym bardziej trafne stają się sugestie systemu, na czym korzysta także użytkownik. Ukazuje to konieczność jednej z cech systemu, tj. rozpoznawanie i zapamiętywanie starych klientów.

- bardziej efektywna praca i mniejsze obciążenie. Dzięki trafnym rekomendacjom można wyświetlać czy analizować mniejszą ilość danych, co jest szczególnie wartościowe, gdy mamy do czynienia ze zbiorami dużych danych.

Systemy rekomendacji opierają się głównie na dwóch podejściach: filtrowaniu opartym na treści (*content-based filtering*), filtrowaniu opartym na współpracy (*collaborative filtering*). Dodatkowo występują sytemy łaczące cechy tych

podejść, nazywane są hydbrydowymi.

Systemy *content-based filtering* są zależne są od danych wprowadzonych przez użytkownika. Szczególnie przydatne są w sytuacjach, gdy znamy dane na temat elementu, natomiast nie mamy informacji dotyczących użytkownika i stosują je na np. Wikipedia czy Google. System uczy się rekomendować elementy podobne do tych, które użytkownik wybierał w przeszłości. Podobieństwo elementów jest obliczane na podstawie cech związanych z porównywanymi obiektami. Temat ten jest bardziej złożony, jednak w naszych rozważaniach skupimy się na drugim podejściu.

Systemy rekomendacji *collaborative filtering* mają tę przewagę, że są o wiele bardziej spersonalizowane [16]. Możemy podzielić je na dwa typy: oparte na użytkowniku (*user-based*) oraz oparte na elementach (*item-based*).

*User-based collaborative filtering (UBCF)* opiera się na założeniu, iż użytkownicy, którzy zgadzali się w przeszłości, zgodzą się także w przyszłości i w dalszym ciągu ich preferencje co do elementów będą się pokrywać. Kluczową zaletą tego typu podejścia jest fakt, że system nie musi skupiać się na „zrozumieniu" natury badanych obiektów, w związku z czym, mogą być one wysoce złożone. Skrótowy opis działania takich algorytmów prezentuje się następująco:

1. Porównanie preferencji obsługiwanego użytkownika z preferencjami innych użytkowników. Podobieństwo to może być obliczane róznymi metodami, np. współczynnikiem korelacji Pearsona czy też algorytmem $k$ najbliższych sąsiadów. Innymi słowy sytem UBCF tworzy macierz *user-item*.

2. System rekomenduje użytkownikowi elementy, które zostały wysoko ocenione przez użytkowników podobnych do obsługiwanego.

*Item-based collaborative filtering (IBCF)* skupia się natomiast na porównywaniu elementów. Nie porównuje się jednak cech elementów i ich dosłownego podobieństwa, ale podobieństwo w tym kontekście określa się na podstawie preferencji elementów przez użytkowników. Postępowanie jest analogiczne jak przy UBCF tzn.:

1. Dokonuje się porównania (tym razem elementów) i tutaj często stosowanymi metodami są: korelacja Pearsona i podobieństwo cosinusów. Tworzymy zatem macierz *item-item*,

2. Po uzyskaniu podobieństwa między elementami następuje przewidywanie preferencji: przyjmując średnią ważoną ocen obsługiwanego użytkownika lub korzystając z aproksymacji ocen opartych na modelu regresji.

Porównując oba te podejścia warto zwrócić uwagę na dwa aspekty: czas działania oraz problem rzadkości danych (*data sparsity*). *UBCF* w najgorszym wypadku działa w czasie $O(NM)$ przy $N$ użytkownikach i $M$ elementach. Zazwyczaj jednak złożoność obliczeniowa wynosi $O(N+M)$, ponieważ większość użytkowników ocenia małą liczbę elementów (stąd $O(N)$), a dla części użytkowników wystawiających dużo ocen złożoność bliska jest $O(M)$. W przypadku *IBCF* złożoność obliczeniowa budowy macierzy *item-item* to w najgorszym wypadku $O(N^2M)$, zaś w praktyce zazwyczaj jest to $O(NM)$. Natomiast samo przewidywanie zależy tylko od liczby ocenionych elementów przez obsługiwanego użytkownika, a z racji, że to jest to zazwyczaj stosunkowo niewielka liczba, złożoność obliczeniowa tej części systemu rekomendacji również jest niska. Zatem pozornie wydaje się, że system *IBCF* jest bardziej czasochłonny, jednak samo tworzenie macierzy *item-item* zachodzi offline, a a przewidywanie działające online jest niskiej złożoności. W wypadku *UBCF* cały proces zachodzi online, co powoduje, iż w rzeczywistości jego działanie jest o wiele wolniejsze.
Problem *data sparsity* dotyka systemy *UBCF*, ponieważ, jak zostało wspomniane, większość użytkowników ocenia niewielką liczbę elementów, w związku z tym, może zdarzyć się, że liczba użytkowników, których preferencje można porównać z obsługiwanym użytkownikiem jest zbyt niska, w związku z czym rekomendacja staje się mało wydajna. *IBCF* jest pozbawiony tej wady [17].

## 4.2 Połączenie z metodą Louvain

Przejdę teraz do opisu implementacji metody Louvain w połączeniu z systemami rekomendacji, opiszę dane oraz przebieg dwóch części eksperymentu, a następnie sforumułuję wnioski i dalsze możliwe udoskonalenia.

Dane, na których będę działał to baza filmów MovieLens [18]. Jest to bardzo często używany obiekt w badań do badania algorytmów operujących na dużych zbiorach danych. Dostępne są dwie jego wersje: pełna z 27 000 000 ocen oraz mniejsza (której fragmentem się posłużę) zawierająca 100 000 ocen wystawionych przez 600 użytkowników, dotyczą one 9 000 filmów. Dane zawierają: ID użytkownika, ID filmu, ocenę (od 0 do 5) oraz znacznik czasu (nie będzie on istotny w dalszych rozważaniach).

Naszym pierwszym celem będzie zbadanie powiązań pomiędzy użytkownikami na podstawie ich preferencji co do ocenianych filmów, a następnie próba przewidzenia konkretnej oceny. Dlatego też stworzymy macierz *user-user*, która posłuży nam do wyznaczenia grafu, który będzie można przeanalizować metodą Louvain.

Eksperyment miał następujący przebieg:

1. Wczytujemy oceny 200 użytkowników usuwając kolumnę znacznika czasu.

2. Wybieramy losowo 8 wierszy i usuwamy z nich oceny (to właśnie je będziemy próbowali przewidzieć).

3. Dla każdej pary użytkowników $A$ i $B$ rozważamy oceny (odpowiednio: $rate_A$ i $rate_B$) filmów, obejrzanych przez obu z nich. Dla pojedynczego filmu $x$ stosujemy wzór:

$$w(A, B, x) = \frac{1}{|rate_A(x) - rate_B(x)| + 0.5}$$

   Zatem $w(A, B, x)$ osiąga maksimum dla $rate_A = rate_B$, a minimum przy skrajnie różnych ocenach (tzn. $rate_A = 5$ i $rate_B = 0$ lub odwrotnie).

4. Tworzymy symetryczną macierz *user-user*, w której wartości są sumami ocen wyliczonych ze wzoru z punktu powyżej. To znaczy, że jeśli $n$

filmów zostało ocenione przez użytkowników $A$ i $B$ to posługujemy się formułą:

$$W_{AB} = \sum_{i=1}^{n} w(A, B, x_i)$$

Uwaga: aby ograniczyć mało istotne powiązania, do macierzy wpisujemy wartość obliczoną z powyższego wzoru jedynie, gdy jest ona wyższa niż 40.

5. Mając macierz wag tworzymy graf ważony użytkowników bazy filmowej i badamy w nim strukturę społeczności za pomocą metody Louvain.

Rozważana sieć to graf składający się z 80 społeczności i o modularności równej 0.18. Odkrycie struktury społeczności pozwala nam na bardziej efektywne przeprowadzenie części drugiej, tj. próby przewidzenia oceny użytkownika. Zatem dla każdej z usuniętych ocen rozważamy jej autora. Sprawdzamy w jakiej znajduje się społeczności, a następnie badamy jak inni członkowie danej społeczności ocenili rozważany (usunięty) przez nas film. Przewidujemy, że „brakująca“ ocena tego filmu to średnia jego ocen wystawionych przez użytkowników z tej samej społeczności. Korzystamy zatem z założenia, będącego podstawą *User-based collaborative filtering*. Tak prezentują się wyniki przedstawionego powyżej rozumowania:

| ID użytkownika | ID filmu | Rzeczywista ocena | Przewidziana ocena |
|---|---|---|---|
| 19 | 338 | 2.0 | 2.0 |
| 19 | 490 | 2.0 | 2.5 |
| 38 | 551 | 4.0 | 3.57 |
| 38 | 500 | 3.0 | 3.62 |
| 139 | 8360 | 2.0 | 3.24 |
| 200 | 51255 | 5.0 | 3.95 |
| 163 | 1722 | 2.0 | 2.0 |
| 179 | 110 | 5.0 | 4.32 |

Średnia różnica pomiędzy rzeczywistą a przewidzianą oceną to 0.56, co jest całkiem dobrym wynikiem i pozwala przypuszczać, iż oparcie przewidywania na strukturze społeczności jest trafną strategią.

Przeprowadzony eksperyment nie jest pozbawiony niedoskonałości. Przede wszystkim niepokoi niska wartość modularności, co sprawia, że odkrycie struktury społeczności traci na miarodajności. Być może problem ten dałoby się zniwelować dobierając inną formułę obliczania podobieństwa między

użytkownikami. Niewątpliwie najwięcej czasu pochłania stworzenie macierzy *user-user* i to właśnie spowodowało dobranie sieci stosunkowo niewielkiej, przez co metoda Louvain nie może ukazać całego swojego potencjału. Jednakże pomimo tych ograniczeń można zauważyć przydatność metody Louvain w systemach rekomendacji. Szczególnie w przypadku sieci olbrzymich rozmiarów, niebywale istotną oszczędnością jest rozważanie jedynie społeczności, nie zaś całej sieci. Jest to uzasadnione zarówno w przypadku systemów *user-based* jak i *item-based*. W obu z nich odnotowano szybszy czas działania, a ponadto systemy *user-based* stają się bardziej spersonalizowane [19].

# 5 Podsumowanie

Sieci złożone to niezwykle złożone obiekty o szerokiem zastosowaniu w wielu dziedzinach, jedną z ich istotnych cech jest niewątpliwie struktura społeczności. Zaprezentowana metoda Louvain jest intuicyjnym i bardzo szybko działającym algorytmem, który znacznie przewyższa inne znany metody optymalizacji modularności, zarówno pod względem efektywności jak i szybkości działania. Jest to szczególnie widoczne w przypadku sieci złożonych o bardzo dużych rozmiarach. Jak zostało pokazane, metoda ta może posłużyć do lepszego zrozumienia sieci, ale także jako dodatek do systemów rekomendujących. Narzędzi niezbędnych do obsługi dużych zbiorów danych oraz znajdujących szerokie zastosowanie komercyjne. Dzięki zrozumieniu podstaw ich działania, w szczególności typu *collaborative filtering* oraz implementacji metody Louvain mogliśmy zobaczyć, jak wykrycie struktury społeczności wspomaga proces rekomendacji. Pomimo, iż dane, na których operowaliśmy nie były typową siecią złożoną to jednak przewidywane oceny niewiele odbiegały od rzeczywistych. Ponadto oszczędność czasu wynikająca z rozważania jedynie jednej społeczności, a nie całej sieci uwidacznia się tym bardziej, im większa jest sieć złożona. Zatem niewątpliwie badanie struktury społeczności jest jedną z możliwych dróg rozwoju systemów rekomendacji.

# 6 Bibliografia

## Literatura


[1] Dorogovtsev, S.N., Mendes, J.F.F. (2003) Evolution of Networks: From Biological Nets to the Internet and WWW. Oxford University Press

[2] Chung-Yuan Huang, Chuen-Tsai Sun, Hsun-Cheng Lin (2005) Influence of Local Information on Social Simulations in Small-World Network Models

[3] Andrei Broder et al. (2000) Graph structure in the Web

[4] Aaron Clauset; Cristopher Moore; M.E.J. Newman (2008). Hierarchical structure and the prediction of missing links in networks

[5] Reichardt, J.; Leone, M. (2008). (Un)detectable Cluster Structure in Sparse Networks

[6] V.D. Blondel et al. (2008). Fast unfolding of communities in large networks. J. Stat. Mech. P10008

[7] M.E.J. Newman, M. Girvan (2004) Finding and evaluating community structure in networks.

[8] S. Fortunato, M. Barthelemy (2008) Resolution limit in community detection

[9] Nikos Koufos, Brendan Martin (2018) K-Means & Other Clustering Algorithms: A Quick Intro with Python

[10] Abbe, Emmanuel; Sandon, Colin (2015) Recovering communities in the general stochastic block model without knowing the parameters

[11] B.H. Good, Y-A de Montjove, A. Clauset. (2009) The performance of modularity maximization in practical contexts

[12] Girvan M., Newman M. E. J. (2002) Community structure in social and biological networks

[13] A. Clauset, M.E.J. Newman, C. Moore. (2004) Finding community structure in very large networks

[14] P. Pons, M. Latapy. (2006) Computing communities in large networks using random walks

[15] K. Wakita, T. Tsurami. (2007) Finding Community Structure in Mega-scale Social Networks

[16] Francesco Ricci and Lior Rokach and Bracha Shapira (2011) Introduction to Recommender Systems Handbook

[17] Wang Bin (2018) Comparison of User-Based and Item-Based Collaborative Filtering

[18] https://grouplens.org/datasets/movielens/latest

[19] Vishnu Sundaresan, Irving Hsu, Daryl Chang (2014) Subreddit Recommendations within Reddit Communities