А.О. Снарський, Д.В. Ланде, І.Ю. Субач

ОСНОВИ ТЕОРІЇ СКЛАДНИХ МЕРЕЖ

Навчальний посібник

Київ – 2023

ІСЗЗІ КПІ ім. Ігоря Сікорського, 2023

МІНИСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
ІНСТИТУТ СПЕЦІАЛЬНОГО ЗВ’ЯЗКУ ТА ЗАХИСТУ ІНФОРМАЦІЇ

А.О. Снарський, Д.В. Ланде, І.Ю. Субач

ОСНОВИ ТЕОРІЇ СКЛАДНИХ МЕРЕЖ

Навчальний посібник

Рекомендовано Вченою радою ІСЗЗІ КПІ ім. Ігоря Сікорського для використання у навчальному процесі з підготовки фахівців другого (магістерського) та третього (освітньо-наукового) рівня вищої освіти за спеціальностями: 122 Комп’ютерні науки, 125 Кібербезпека та захист інформації, 104 Фізика та астрономія

Київ
ІСЗЗІ КПІ ім. Ігоря Сікорського
2023

УДК 004.942 (075.8)

Рекомендовано до видання Вченою радою ІСЗЗІ КПІ ім. Ігоря Сікорського (Протокол № 3 від 30.09.2023 р.) і Вченою радою фізико-математичного факультету КПІ ім. Ігоря Сікорського (Протокол № 9 від 20.09.2023 р.)

Рецензенти: Б.І. Лев, доктор фізико-математичних наук, професор, академік НАН України
Д.І. Могилевич, доктор технічних наук, професор

Снарський А.О., Ланде Д.В., Субач І.Ю.

Л 12 Основи теорії складних мереж: навч. пос. / Снарський А.О., Ланде Д.В., Субач І.Ю.; ІСЗЗІ КПІ ім. Ігоря Сікорського. – Київ: ТОВ "Інжиніринг", 2023. – 225 с. ISBN 978-966-2344-95-0.

У навчальному посібнику розглядаються базові питання теорії складних мереж: характеристики, алгоритми, моделі завдання пошуку, ранжування, а також наводяться відомості, необхідні для математичного та комп'ютерного моделювання, аналізу та візуалізації складних мереж.

Видання призначене для курсантів, студентів та аспірантів закладів вищої освіти, а також інженерів та наукових співробітників, які працюють у галузях знань інформаційні технології та природничі науки та електроніка.

УДК 004.942 (075.8)

ISBN 978-966-2344-95-0

Зміст

Вступ

Мережі оточують нас усюди – це соціальні мережі, мережі дружби, співавторства у наукових публікаціях, бізнес-зв'язків, публікацій у ЗМІ, спільного вживання слів у текстах, метаболізмів (обмінних процесів), кровоносних судин, транспорту, нарешті (див. рис. 1), Інтернет та вебпростір. Новий напрямок – теорія складних мереж (Complex Networks), охоплює величезне поле діяльності, отримано багато результатів, виникли нові розділи у наукових журналах та нові журнали.

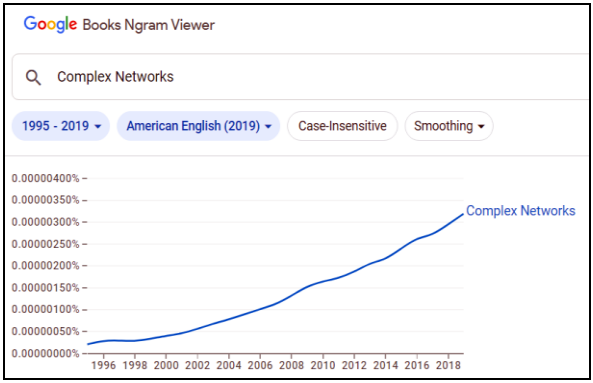


Рисунок 1 – Динаміка вживання словосполучення Complex Networks, отримана за допомогою сервісу Google Ngram Viewer (https://books.google.com/ngrams)

При першому знайомстві з оглядами теорії складних мереж виникають питання: – Чи не просто модна назва ця теорія складних мереж? - Чим відрізняється мережа, хай і складна, від графа? Адже теорія складних мереж походить з теорії графів, початок якої поклав ще Ейлер своїм знаменитим завданням про мости Кенігсбергу.

Формально, з математичного погляду, будь-яка мережа це граф.

З давніх-давен серед жителів Кенігсберга була поширена така загадка: як пройти по всіх семи мостах

(через річку Преголя, див. рис. 2), не проходячи по жодному одному з них двічі. Довести чи спростувати можливість існування такого маршруту ніхто не міг.

Рисунок 2 – Схема мостів Кенігсберга

Леонард Ейлер (1707-1783)

Леонард Ейлер 13 березня 1736 р. у листі до італійського математика та інженера Маріоні написав, що знайшов правило, користуючись яким легко визначити, чи можна пройти по всіх мостах, не проходячи двічі по одному з них:

Ейлер довів, що можливість пройти через граф, проходячи кожне ребро рівно один раз, залежить від того, скільки ребер торкаються кожної вершини. Степінь вершини визначається кількістю таких ребер. Ейлер показав, що необхідною умовою для можливості такого проходу через граф є зв'язність графу і відсутність або наявність рівно двох вершин непарної ступені. Ця умова також виявилась достатньою, як пізніше це було доведено Карлом Гьєхолзером.

Граф Кенігсберзьких мостів має п'ять непарних вершин, тобто мости не можна обійти, не проходячи двічі якимсь із них.

У 1905 році за наказом Кайзера Вільгельма було збудовано Імператорський міст, який був згодом зруйнований під час бомбардування під час Другої світової війни. У даний час у Калінінграді сім мостів, і граф, як і раніше, не має ейлерового шляху.

Але все ж таки теорія складних мереж, не є теорією графів. Наведемо аналогію. Класична механіка вивчає, зокрема, рух матеріальних точок. Ідеальний газ – це якраз рухомі матеріальні точки. Проте властивості газу вивчає інший розділ фізики – статистична механіка, що має свої методи, терміни, прийоми. А «звичайна» класична механіка із задачею опису газу впоратися не може, надто багато частинок навіть у невеликому «шматочку» газу. Тому Максвеллу, Больцману, Гіббсу та багатьом іншим ученим довелося створити нову науку та запровадити нові поняття, наприклад, температуру та ентропію, які не потрібні і не вводяться принципово у класичній механіці.

Аналогічні взаємини теорій графів та складних мереж. Як основою класичної статистичної фізики є класична механіка, так основою теорії складних мереж є теорія графів. При цьому теорія графів може отримувати змістовні твердження як правило для графів (мереж) невеликого розміру або спеціальної структури. У той самий час, теорія складних мереж має справу з великою кількістю по-різному з'єднаних вузлів. У зв'язку з цим, з одного боку, багато поширених питань теорії графів не становлять інтересу для теорії складних мереж. Наприклад, таке питання – чи планарний цей граф? З іншого боку, багато дуже важливих понять в теорії складних мереж для графа з невеликою кількістю вузлів і зв'язків або не становлять інтересу, або їх просто складно змістовно сформулювати. Наприклад, така важлива характеристика теорії складних мереж як функція розподілу за ступенями вузлів може бути підрахована для будь-якого графу, в тому числі і малої кількості вузлів. Однак у силу свого ймовірнісного змісту це поняття

виявляється корисним лише у разі великої кількості вузлів та зв'язків.

Чим займається теорія складних мереж? Перелічимо деякі проблеми та завдання цієї теорії.

По-перше, дослідженням стандартних характеристик графів для складних мереж різної природи – випадкових графів, безмасштабних мереж, мереж малого світу, динамічних мереж тощо.

По-друге, визначенням та вивченням нових характеристик складних мереж, таких, наприклад, як посередництво, коефіцієнт кластеризації, модулярності тощо.

По-третє, вивченням різних «фізичних» процесів на складних мережах – дифузії, перколяції, епідемічних процесів, різних потоків (інформації, електричного струму тощо).

По-четверте, пошук і блукання зв'язків на складних мережах (алгоритм PageRank для веб, пошук у мережах типу P2P і т.і.).

По-п'яте, є дуже важливий у прикладному відношенні, напрямок – методи відновлення, захисту та руйнування мереж. Таке питання – скільки вузлів (зв'язків) потрібно видалити для того, щоб, наприклад, зруйнувався «гігантський кластер» або щоб значно збільшився мінімальний середній шлях? Сюди примикають і питання оптимізації мереж.

По-шосте, пошук неявних зв'язків, тих, що штучно приховуються. Важливе застосування цього завдання – пошук зв'язків терористів. І, звичайно, бізнес-розвідка.

Методи, які використовуються для вирішення цих та багатьох інших завдань можна, умовно, поділити на три типи:

1. Методи теорії графів, значною мірою комбінаторні.

2. Чисельне моделювання, нині добре розвинене, випробуване і пристосоване з метою дослідника складних мереж. Наприклад, для мови програмування Phyton розроблено спеціальний пакет Networkx[1] – інструментарій для створення, маніпулювання та вивчення складних мереж, що дозволяє знайти чисельно практично всі можливі характеристики складних мереж.

3. Третій тип методів, який дозволив встановити основні закономірності у складних мережах – методи теоретичної фізики. Від теорії середнього поля до ренорм-групи та діаграмної техніки. Недарма багато провідних дослідників складних мереж – фізики теоретики.

Таким чином, наше видання призначене, в першу чергу, тим, хто тільки чув термін «складна мережа» або зустрічався з ним у статті за своєю предметною сферою, наприклад, сферами захисту інформації, кібербезпеки, аналізу даних тощо. Читачеві, що зацікавився, досить складно відразу ж освоїти великий науковий огляд, вивчити основні алгоритми і методи. Тут і має допомогти пропонований навчальний посібник. У його першій частині описані основні характеристики та моделі побудови кількох мереж, що найчастіше зустрічаються. Друга частина книги призначена тим читачам, які хотіли б ознайомитися з аналізом складних мереж. Третя частина видання знайомить читача з корисними та популярними програмними системами аналізу та візуалізації мереж.

---

[1] *Eric Ma and Mridul Seth.* Network Analysis Made Simple An introduction to network analysis and applied graph theory using Python and NetworkX. Leanpub, 2021. – 191 p.

1. Складні мережі

Незважаючи на те, що до теорії складних мереж потрапляють різні мережі – електричні, транспортні, інформаційні, найбільший внесок у розвиток цієї теорії зробили дослідження саме соціальних мереж.

Термін «соціальна мережа» означає зосередження соціальних об'єктів, які можна розглядати як мережу (чи граф), вузли якої – об'єкти, а зв'язки – соціальні відносини. Цей термін було запроваджено у 1954 році соціологом з «Манчестерської школи» Дж. Барнсом (J. Barnes) у роботі «Класи та збори в норвезькому острівному приході»[2]. У другій половині XX століття поняття "соціальна мережа" стало популярним у західних дослідників. З точки зору теорії соціальних мереж, набув широкого розвитку такий напрямок, як аналіз соціальних мереж (Social Network Analysis, SNA). Сьогодні термін «соціальна мережа» означає поняття, що виявилося ширшим за свій соціальний аспект, воно включає, наприклад, багато інформаційних мереж, у тому числі й вебпростір.

Теоретично у складних мережах виділяють три основних напрями: дослідження статистичних властивостей, які характеризують поведінку мереж; створення моделі мереж; передбачення поведінки мереж за зміни структурних властивостей. У прикладних дослідженнях зазвичай застосовують такі типові для мережевого аналізу характеристики, як розмір мережі, мережна щільність, рівень центральності тощо.

При аналізі складних мереж як і теорії графів досліджуються параметри окремих вузлів; параметри мережі у цілому; мережеві підструктури.

Нова парадигма – «складні мережі» охоплює мережі, які мають властивості:

[2] *Barnes, J.A.* «Class and Committees in a Norwegian Island Parish», Human Relations 7:39-58.

1) великі розміри;
2) елементи випадковості при формуванні;
3) зростання (зміна) у часі;
4) деякі вузли можуть утворювати компактні групи – ансамблі.

Вивчення значної кількості складних артефактних (штучно створених) мереж, деякі з яких описуються у цьому посібнику, було ініційовано бажанням зрозуміти та описати численні реальні мережі – від мереж комунікації до екологічних мереж. Ось деякі з них:

1. World Wide Web: Кількість веб-сайтів – 1 101 218 364 (станом на липень 2023 року за даними служби Netcraft, див. рис. 1).
2. Internet – «Фізична мережа» (станом на січень 2019 див. рис. 2, 3);
3. Протеїнові мережі (див. рис. 4);
4. Мережа метаболізму;
5. Екологічні мережі;
6. Мережа телефонних дзвінків;
7. Мережа зв'язків терористів;
8. Мережа цитування (ациклічна);
9. Лінгвістична (мережа пов'язаних слів);
10. Нейронні мережі.

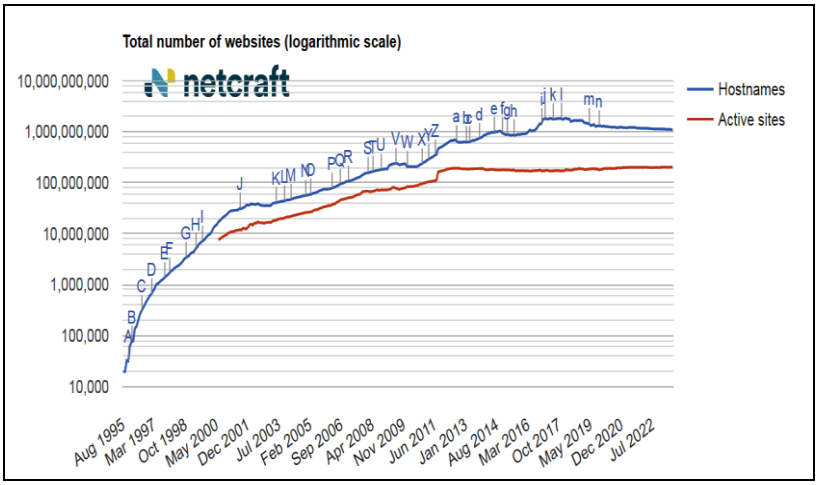


Рисунок 3 – Динаміка розвитку мережі World Wide Web за даними служби Netcraft (http://netcraft.com)

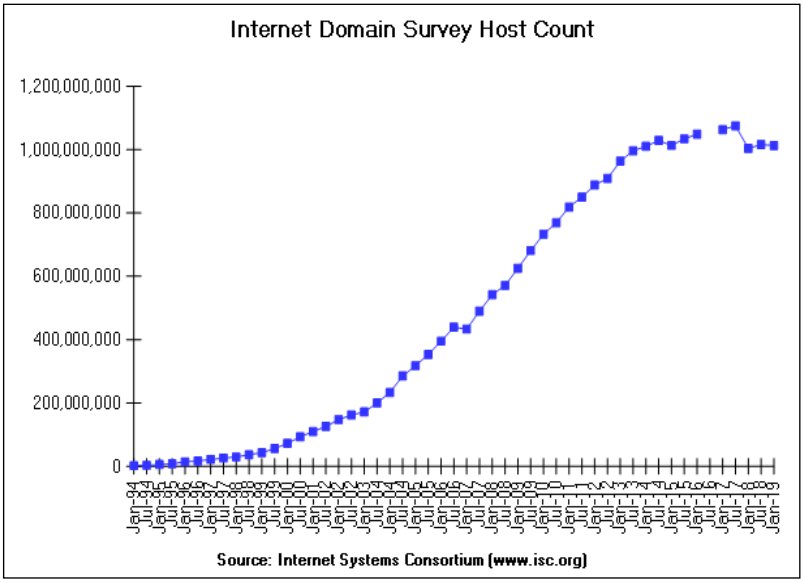


Рисунок 4 – Динаміка зростання кількості Інтернет-доменів за інформацією служби isc.org

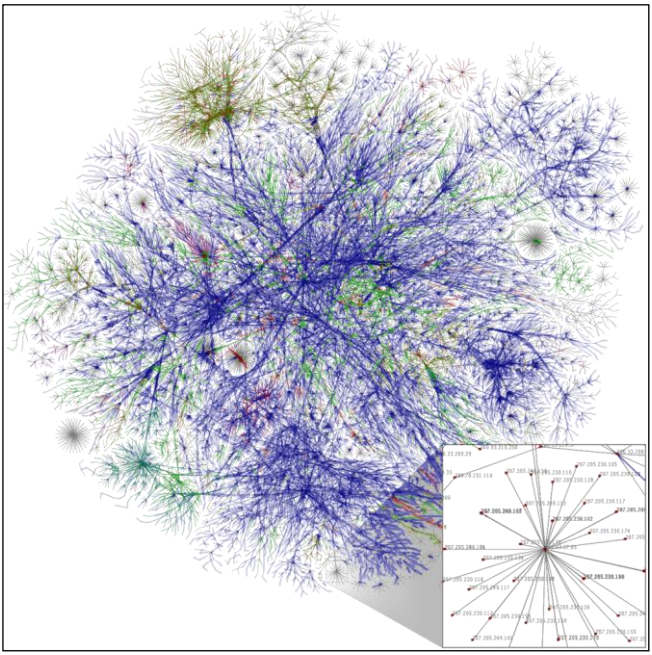


Рисунок 5 – Карта зв'язків Інтернет-серверів як складна мережа (за даними wikipedia.org)

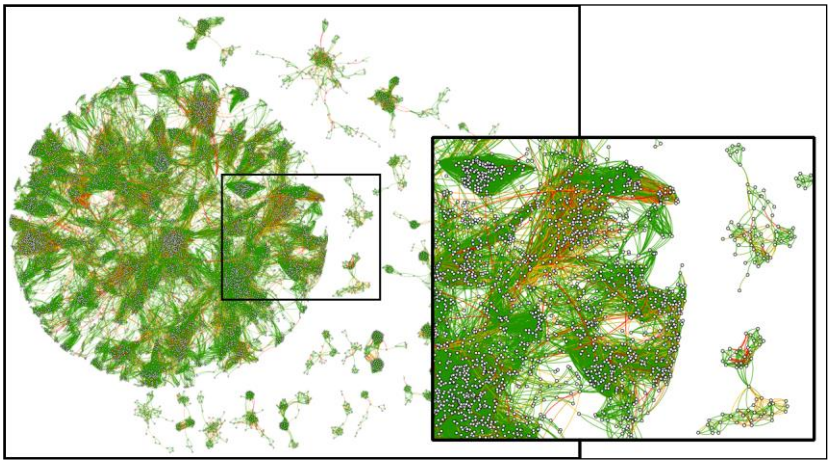


Рисунок 6 – Протеїнова мережа (проєкт Protein Structure Initiative, сайт www.helixscript.com)

При аналізі мереж найчастіше використовуються декілька типових характеристик, які допомагають зрозуміти структуру та властивості цих мереж.

1. Розмір мережі: це проста та важлива характеристика, яка відображає кількість вузлів (вершин) у мережі. Розмір мережі визначає загальну кількість об'єктів чи елементів, пов'язаних між собою.
2. Щільність мережі: ця характеристика показує, наскільки мережа щільно пов'язана. Мережева щільність вимірює відношення кількості реальних зв'язків до максимально можливої кількості зв'язків у мережі. Висока щільність може вказувати на наявність близьких зв'язків між об'єктами, тоді як низька щільність може вказувати більш розріджені взаємодії.
3. Міра центральності: центральність вузла відбиває його важливість чи впливовість у мережі. Існують різні типи заходів центральності, такі як центральність за ступенем (кількість зв'язків вузла), близькості (близькість вузла до інших вузлів), та посередництва (роль вузла у пересиланні інформації між іншими вузлами). Заходи центральності дозволяють виявити ключові вузли, які відіграють важливу роль у функціонуванні мережі.

Крім названих характеристик, при аналізі складних мереж, аналогічно теорії графів, також досліджуються параметри окремих вузлів, щоб зрозуміти їх ролі та взаємодії, параметри мережі в цілому для загального уявлення про структуру та динаміку мережі, а також мережеві підструктури, такі як спільноти або кластери, щоб виявити угруповання об'єктів зі схожими характеристиками або функціональними зв'язками.

Для аналізу мережі загалом також використовують такі параметри:

1. Кількість вузлів: це проста характеристика, що відображає загальну кількість вузлів (вершин) у мережі.
2. Кількість ребер: це кількість зв'язків між вузлами в мережі.
3. Середня відстань між вузлами: це середнє значення геодезичних відстаней (найкоротших шляхів) між усіма парами вузлів у мережі. Це дозволяє зрозуміти середню віддаленість між вузлами.
4. Щільність мережі: це співвідношення наявних і можливих зв'язків:

   $\Delta = \frac{2L}{n(n-1)}$, де L – кількість спостережуваних зв'язків, n – кількість вузлів у мережі. Щільність відбиває, наскільки мережа щільно пов'язана.
5. Кількість симетричних, транзитивних та циклічних трійок: ці параметри відображають характерні особливості трійок вузлів у мережі, які можуть бути симетричними (всі зв'язки між вузлами прямі та у зворотному напрямку), транзитивними (коли вузли A та C пов'язані через вузол B), або утворювати замкнуті цикли.
6. Діаметр мережі: це найбільше значення геодезичних відстаней між усіма парами вузлів у мережі. Діаметр відображає максимальну віддаленість між вузлами в мережі.

Мережі в цілому характеризуються такими параметрами, як кількість вузлів, кількість зв'язків, мережна щільність, відстань між вузлами, середня відстань від одного вузла до інших, мережна щільність, кількість симетричних, транзитивних та циклічних тріад, діаметр мережі – найбільша відстань між вузлами в мережі та і т.д.

Існує кілька актуальних завдань дослідження складних мереж, серед яких можна виділити такі основні:

- визначення кліків у мережі. Кліки – це підгрупи чи кластери, у яких вузли пов'язані між собою сильніше, ніж із членами інших клік;
- виділення компонентів (частин мережі), які не пов'язані між собою, вузли яких пов'язані всередині цих компонентів;
- знаходження блоків та перемичок. Вузол називається перемичкою, якщо його при вилученні мережа розпадається на незв'язані частини;
- виділення угруповань – груп еквівалентних вузлів (які мають максимально схожі профілі зв'язків).

1.1 Параметри вузлів мережі

Для окремих вузлів у мережі виділяють такі параметри, які допомагають зрозуміти їх роль та вплив на загальну структуру та функціонування графа:

- Вхідний напівступінь вузла: цей параметр показує кількість ребер, спрямованих у цей вузол. Він відбиває важливість вузла як приймача чи одержувача зв'язків.
- Вихідний напівступінь вузла: цей параметр показує кількість ребер, які спрямовані з цього вузла. Він відбиває важливість вузла як джерела чи відправника зв'язків.
- Середня відстань від цього вузла до інших: це середня кількість ребер, які необхідно пройти від цього вузла до кожного іншого вузла в мережі. Цей параметр характеризує, як швидко і ефективно вузол може обмінюватися інформацією з іншими вузлами.
- Ексцентриситет: цей параметр визначає максимальне значення геодезичних відстаней від цього вузла до всіх інших вузлів у мережі. Він дозволяє виявити, наскільки далеко від інших вузлів

знаходиться даний вузол і наскільки може бути центром поширення інформації.

- Посередництво (betweenness): цей параметр показує, скільки найкоротших шляхів проходить через цей вузол. Вузол з високим значенням посередництва відіграє важливу роль передачі інформації між іншими вузлами і може бути ключовим посередником у мережі.
- Параметри центральності: це різні показники, які визначають важливість вузла в мережі з погляду його зв'язків з іншими вузлами. Наприклад, центральність за рівнем вузла відображає кількість зв'язків даного вузла з іншими вузлами, а центральність за посередництвом - його роль у проходженні найкоротших шляхів між іншими вузлами.

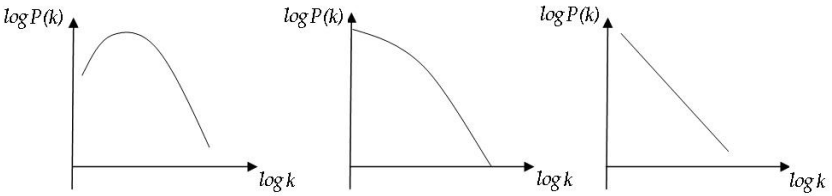
1.2 Розподіл ступенів вузлів

Класифікація мереж за типами може бути здійснена різними способами. Можна розрізняти мережі у тому, що є вузли і зв'язку.

Для великих (N>>1) мереж із випадковою структурою однією з найважливіших характеристик є функція розподілу за ступенями вузлів $P(k)$. Багато реальних складних мереж схожі на наступні:

1. Випадкова мережа чи мережа Ердеша-Реньї (ER), так званий випадковий граф $P(k) \sim e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!}$. Таким чином, у випадку мережі ER, функція розподілу є функція Пуассона.
2. Мережа з експоненційним розподілом $P(k) \sim e^{-k/\langle k \rangle}$.
3. Мережа зі степеневим розподілом (Scale-Free) $P(k) \sim k^{-\gamma}$.

У подвійному логарифмічному масштабі ці розподіли мають вигляд, наведений на рис. 7.



а б в

Рисунок 7 – Щільності розподілу $P(k)$ у подвійному логарифмічному масштабі: а – розподіл Пуассона (мережа ER); б - експоненційний розподіл; в – степеневий розподіл

Мережу Ердеша-Реньї (ER) можна побудувати, розподіливши випадковим чином M зв'язків між N вузлами. Тоді з одного боку $\langle k \rangle = 2M / N$, а з іншого $\langle k \rangle = mN$, де m – ймовірність з'єднання вузлів. При $N \to \infty$ і $m \to 0$ розподіл ступеня вузлів є Пуассоновим.

До мереж із степеневим розподілом відносяться мережі Барабаші-Альберта (BA)[3], для побудови яких застосовується спеціальна процедура, яка полягає у тому, що до спочатку невеликого числа вузлів N_0 поступово додаються нові вузли, зв'язки від яких з більшою ймовірністю приєднуються до тих вузлів, у яких більше зв'язків.

Альберт-Ласло Барабаши

[3] *Barabási A.L., Albert R.* Emergence of scaling in random networks. *Science, 1999.* – Vol. 286 (5439): 509–512.

Існують мережі іншого типу, так звані мережі малого світу (Small World – SW), у яких більшість вузлів є сусідами одне одного, але сусіди будь-якого заданого вузла, мабуть, будуть сусідами один одного. Завдяки цьому до більшості сусідніх вузлів можна дістатися з іншого вузла за невелику кількість переходів або кроків. Такі мережі можна моделювати, наприклад, таким чином: до впорядкованої структури додаються випадкові зв'язки. Найбільш відомий приклад такої мережі мережа Ваттса-Строгатца[4].

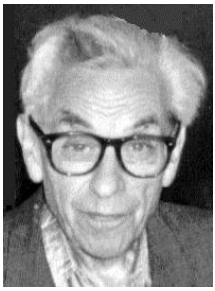
1.3 Найкоротший шлях між вузлами

Відстань між вузлами визначається як кількість кроків, які необхідно зробити, щоб існуючими ребрами дістатися від одного вузла до іншого. Природньо, вузли можуть бути з'єднані безпосередньо або опосередковано, через інші вузли.

Найкоротшим шляхом (SP, shortest path) між вузлами назвемо мінімальну відстань між ними. Для всієї мережі можна ввести поняття середнього найкоротшого шляху, як середня по всіх парах вузлів мінімальна відстань між ними:

$$l = \frac{2}{n(n+1)} \sum_{i \geq j} l_{ij},$$

де n – кількість вузлів, l_{ij} – найкоротший шлях між вузлами i та j.



Пауль Эрдеш

(1913-1996)

[4] *Watts DJ, Strogatz SH* (June 1998). "Collective dynamics of 'small-world' networks". Nature. 393 (6684): 440–2.

П. Ердешем (P. Erdős) та А. Реньї (A. Rényi) було показано, що середній найкоротший шлях у випадковому графі зростає повільно – як логарифм від числа вузлів[5],[6].

З ім'ям П. Ердеша пов'язані не лише дослідження складних мереж, а й популярне число Ердеша, яке використовується як один із критеріїв визначення рівня математиків у відповідному соціумі, що базується на так званій мережі співавторства.

Відомо, що Ердеш написав близько півтори тисячі статей, а також те, що кількість його співавторів перевищувала 500. Така велика кількість співавторів і породила таке поняття, як число Ердеша, яке визначається таким чином: у самого Ердеша це число дорівнює нулю; у співавторів Ердеша це число дорівнює одиниці; співавтори людей з числом Ердеша, що дорівнює одиниці, мають число Ердеша два; і т.д.

Таким чином, число Ердеша це довжина шляху від деякого автора до самого Ердеша із спільних робіт. Відомий факт, що 90% математиків мають число Ердеша не вище 8, що відповідає мережам «малих світів», про які піде нижче.

1.4 Коефіцієнт глобальної ефективності

Мережа може виявитися не зв’язною, тобто знайдуться вузли, відстань між якими виявиться нескінченною. Отже, середній шлях, згідно з зазначеною вище формулою, також буде нескінченним. Для урахування таких випадків запроваджується поняття середнього зворотного шляху між вузлами (його також називають "глобальною ефективністю мережі"), яка розраховується за формулою:

---

[5] *Erdős, P., Rényi A.* On Random Graphs. Publicationes Mathematicae, 1959. – № 6. – pp. 290-297.
[6] *Erdős P., Rényi A.* On the evolution of random graphs. Publ. Math. Inst. Hungar. Acad. Sci., 1960. – № 5. – pp. 17-61.

$$ge = \frac{2}{n(n-1)} \sum_{i>j} \frac{1}{l_{ij}}.$$

Зворотна величина глобальної ефективності – це середня гармонійна геодезичних відстаней: $h = 1/ge.$

Один із способів знайти критичні компоненти мережі – це пошук найбільш уразливих вузлів. Вразливість мережі щодо вузла може бути визначена як зменшення глобальної ефективності мережі при видаленні вузла та всіх суміжних з ним ребер із мережі:

$$V_i = \frac{(ge - ge_i)}{ge},$$

де ge – глобальна ефективність вихідної мережі, а ge_i – глобальна ефективність після видалення вузла і всіх суміжних йому ребер.

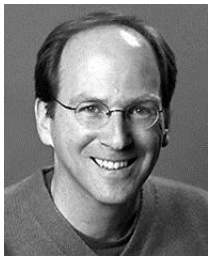
Упорядкований розподіл вузлів за цією величиною пов'язаний із структурою всієї мережі. Таким чином, вузол, який найбільше впливає на вразливість мережі, займає найвищу позицію в мережній ієрархії. Міра вразливості мережі – це максимальна вразливість серед усіх вузлів:

$$V = \max V_i.$$

1.5 Коефіцієнт кластеризації

Д. Уаттс (D. Watts) та С. Строгатц (S. Strogatz) у 1998 році визначили такий параметр мереж, як коефіцієнт кластеризації. Цей коефіцієнт характеризує тенденцію до утворення сильно зв'язаних груп вузлів, клік (Clique). Для конкретного вузла коефіцієнт кластеризації показує, скільки найближчих сусідів даного вузла є також найближчими сусідами один для одного[7].

Д.Уаттс



С. Строгатц

Нехай із вузла виходить k зв'язків, які з'єднують його з k іншими вузлами, найближчими сусідами. Якщо припустити, що це найближчі сусіди з'єднані безпосередньо один з одним, кількість зв'язків з-поміж них становила б $k(k-1)/2$. Тобто це число, яке відповідає максимально можливій кількості зв'язків, якими могли б поєднуватися найближчі сусіди обраного вузла.

Відношення реальної кількості зв'язків, які з'єднують найближчих сусідів даного вузла i до максимально можливого (такого, при якому всі найближчі сусіди даного вузла були б з'єднані безпосередньо один з одним) називається коефіцієнтом кластеризації вузла C_i. Природно, ця величина не перевищує одиниці.

Коефіцієнт кластеризації може визначатися як для кожного вузла, так і для всієї мережі:

$$C = \frac{1}{n}\sum_{i=1}^{n} C_i$$

[7] *Watts D.J., Strogatz S.H.* Collective dynamics of "small-world" networks. Nature, 1998. – Vol. 393. – pp. 440-442.

У соціальних мережах можна говорити про «структуру спільноти», коли існують групи вузлів, які мають високу густину зв'язків між собою, при тому, що густина ребер між окремими групами – низька. Для великих соціальних мереж наявність структури угруповань виявилася невід'ємною властивістю. Традиційний метод виявлення структури співтовариств – кластерний аналіз. Існують десятки прийнятних для цього методів, що базуються на різних заходах відстаней між вузлами, зважених колійних індексах між вузлами тощо.

1.6 Посередництво

Посередництво (betweenness) – це параметр, який показує, скільки найкоротших шляхів проходить через вузол. Ця характеристика відбиває роль даного вузла у встановленні зв'язків у мережі. Вузли з найбільшим посередництвом грають головну роль у встановленні зв'язків між іншими вузлами у мережі. Посередництво b_m вузла m визначається за формулою:

$$b_m = \sum_{i \neq j} \frac{B(i,m,j)}{B(i,j)},$$

де $B(i,j)$ – загальна кількість найкоротших шляхів між вузлами i nf j, $B(i,m,j)$ – кількість найкоротших шляхів між вузлами i та j, що проходять через вузол m.

1.7 Модулярність

Модулярність - один із мережевих параметрів, який був введений для вимірювання ступеня розбиття мережі на модулі (кластери, кліки). Він обчислюється як різниця між часткою ребер всередині кластера в мережі, що розглядається, і очікуваною часткою ребер всередині кластера в мережі, в якій вершини мають той же ступінь, що і у вихідній, але ребра розподілені випадковим чином.

Для розрахунку модулярності використовують поняття матриці суміжності. Матриця суміжності A складається з елементів a_{vw}, значення яких дорівнює 0 якщо вузол v не пов'язаний з вузлом w, і ваги зв'язку між v і w, якщо ці вузли з'єднані між собою.

Модулярність мережі можна виразити формулою:

$$Q = \frac{1}{2m} \sum_{v,w} \left[a_{vw} - \frac{k_v k_w}{2m} \right] \delta(c_v, c_w),$$

де a_{vw} – елемент матриці суміжності A, m – кількість ребер у графі, k_v, k_w – ступені вузлів v і w, відповідно, δ – дельта-функція Кронекера (показує, чи знаходяться вузли v та w в одному модулі).

Отже, модулярність – це міра якості кластеризації, на основі якої будується широкий клас алгоритмів виявлення груп у мережах.

1.8 Еластичність мережі

Властивість еластичності мереж відноситься до розподілу відстаней між вузлами при вилученні окремих вузлів (наприклад, толерантність до атак)[8].

Еластичність мережі залежить від її зв'язності, тобто. існування шляхів між парами вузлів. Якщо вузол буде вилучений із мережі, типова довжина цих шляхів збільшиться.

При дослідженні атак на вебсервери вивчався ефект вилучення вузлів мережі, які є підмножиною вебпростору з 326000 сторінок і близько 1,5 млрд. гіперпосилань. Для цієї мережі було визначено параметри вхідного та вихідного розподілу ступенів: $P(k) \sim k^{-\gamma}$, де $\gamma_{in} = 2{,}1$ і $\gamma_{out} = 2{,}45$.

[8] *Albert R., Jeong H., Barabasi A.* Error and attack tolerance of complex networks. Nature, 2000. – Vol. 406. – pp. 378-382.

Середня відстань між двома вузлами як функція від кількості вилучених вузлів майже не змінилася при випадковому видаленні вузлів (висока еластичність при атаці на мережі зі статичним розподілом). Разом з тим, цілеспрямоване видалення вузлів з найбільшою кількістю зв'язків призводить до руйнування мережі. Таким чином, вебпростір є високоеластичною мережею по відношенню до випадкового вилучення вузла в мережі, але високочутливою до навмисної атаки на вузли з високими ступенями зв'язків з іншими вузлами.

1.9 Феномени мереж

Про "структуру спільноти" можна говорити тоді, коли існують групи вузлів (кластери), що мають високу густину ребер між собою тому, що густина ребер між окремими групами – низька. Традиційний спосіб виявлення структури спільноти – кластерний аналіз. Існують численні методи кластерного аналізу, які базуються різних вимірах відстаней між вузлами. Для великих лінгвістичних мереж наявність структури угруповань виявилася невід'ємною властивістю.

1.9.1 Слабкі зв'язки

Серед характеристик реальних соціальних мереж є, так звані, "слабкі" зв'язки. Аналогом слабких соціальних зв'язків можуть бути, наприклад, відносини з далекими знайомими та колегами. У деяких випадках ці зв'язки виявляються ефективнішими, ніж "сильні" зв'язки. Було отримано концептуальний висновок щодо мобільних комунікацій, суть якого полягає у тому, що " слабкі " соціальні зв'язки між індивідуумами виявляються найважливішими для існування соціальної мережі.

Виявили, що саме слабкі соціальні зв'язки об'єднують у єдине ціле велику соціальну мережу. Якщо ці зв'язки ігнорувати, то мережа розіб'ється на окремі фрагменти, тобто зв'язок мережі порушиться. Виявилося, що саме

слабкі зв'язки є тим феноменом, який поєднує мережу в єдине ціле.

Очевидно, що у лінгвістичних мережах цей ефект також має місце.

1.9.2 Small World

С. Мілгрем

У 70-ті роки минулого століття американський психолог Мілграм (Milgram) задався питанням, якою є «відстань» між двома випадково обраними людьми. Під відстанню розуміється кількість знайомств, необхідна для встановлення зв'язку між даними людьми. Мілграм вчинив так – оскільки він жив у Бостоні, то було обрано далеке від Бостона місто – Небраска, і випадково обраним людям було роздано конверти, які треба було передати у Бостон конкретній людині. Усього було роздано 300 конвертів, які можна було передавати лише через своїх знайомих. Усього до адресата дійшло 60 конвертів. Мілграм отримав дуже несподіваний результат: у середньому кожен конверт пройшов через п'ять осіб. Так і народилася теорія «шести рукостискань». У цьому сенсі про наш світ говорять, як про малий світ – “Small World”.

1.9.3 Феномен клубу багатих

WWW (World Wide Web) є мережею, для якої також підтверджено феномен малих світів. Аналіз топології Інтернету, проведений S. Zhou та R.J. Mondragon[9] з Лондонського університету, показав, що вузли з великою кількістю вихідних гіперпосилань мають більше зв'язків

[9] Shi Zhou, Mondragon R.J. The rich-club phenomenon in the Internet topology. IEEE Communications Letters, 2004. Vol. 8. Iss. 3. – pp. 180 – 182. DOI: 10.1109/LCOMM.2004.823426

між собою, ніж з вузлами з малим числом посилань, у той час як останні мають більше зв'язків з вузлами з більшим числом посилань, ніж між собою. Цей феномен був названий "клубом багатих" (Rich-Club Phenomenon). Дослідження показало, що 27% всіх зв'язків відбуваються лише між 5% найбільших вузлів, 60% припадає на зв'язки між рештою 95% вузлів і цими 5% великих, а лише 13% зв'язків утворюють вузли, що не входять до цих основних 5%.

Ці дослідження дають підстави вважати, що залежність WWW від великих вузлів є значно суттєвішою, ніж передбачалося раніше, тобто вона ще чутливіша до зловмисних кібератак. З концепцією "малих світів" пов'язаний також практичний підхід, який називається "мережевою мобілізацією", що реалізується над структурою "малих світів". Зокрема, швидкість поширення інформації завдяки ефекту "малих світів" у реальних мережах зростає на порядки порівняно з випадковими мережами, адже більшість пар вузлів реальних мереж з'єднані короткими шляхами.

Питання для самоконтролю

1. Що означає термін "соціальна мережа" за сучасними визначеннями?
2. Які три основні напрями аналізу соціальних мереж виділяються в теорії складних мереж? Дайте короткий опис кожного напряму.
3. Які ключові характеристики мережі використовуються для аналізу її структури та властивостей? Поясніть, що вони відображають.
4. Які основні завдання дослідження складних мереж ви визначаєте? Дайте короткий опис кожного завдання.
5. Яке значення має параметр "мережева щільність"? Як вона визначається?
6. Що означає вхідна та вихідна напівступінь вузла в мережі? Які аспекти ролі вузла вони відображають?

7. Що означає поняття посередництва (betweenness) в мережі? Як воно пов'язане з проходженням інформації між вузлами?
8. Які параметри центральності використовуються для оцінки важливості вузла в мережі? Надайте приклади таких параметрів та поясніть їхню суть.
9. Які основні характеристики мережі зв'язані з розподілом ступенів вузлів? Які типи розподілів ступенів можна спостерігати у реальних складних мережах?
10. Як виглядає розподіл ступенів вузлів у випадковій мережі Ердеша-Реньї (ER)? Яка характеристика цього розподілу?
11. Яким чином можна побудувати мережу Барабаші-Альберта (BA)? Які основні різниці між цією мережею та випадковою мережею ER?
12. Які характеристики мають мережі малого світу (SW)? Як вони допомагають забезпечити швидкий обмін інформацією в мережі?
13. Яким чином моделюється мережа Ваттса-Строгатца? Яку структуру вона має та які властивості?
14. Як визначається середній найкоротший шлях для всієї мережі? Як цей параметр пов'язаний із зростанням мережі та логарифмом числа вузлів?
15. Як визначається число Ердеша та яку роль воно відіграє в оцінці рівня математиків?
16. Як визначається глобальна ефективність мережі? Які ситуації можуть викликати нескінченність середнього шляху, і як це враховується в розрахунках?
17. Як визначається вразливість мережі та яким чином можна знайти найбільш уразливі вузли?
18. Що таке коефіцієнт кластеризації в мережі? Як він визначається для окремих вузлів та для всієї мережі? Як він пов'язаний з утворенням груп взаємозалежних вузлів?
19. Що таке посередництво в мережі та як воно визначається?
20. Які ролі грають вузли з найбільшим посередництвом у встановленні зв'язків у мережі?
21. Як визначається модулярність мережі? Яка її суть та яку інформацію вона надає про структуру мережі?
22. Як використовується матриця суміжності для обчислення модулярності мережі?

23. Як впливає еластичність мережі на розподіл відстаней між вузлами після видалення окремих вузлів? Які ситуації можуть спричинити зміни у середній відстані?

2. Моделі артефактних мереж

2.1. Імовірнісні розподіли

Найбільш частими (як зазвичай вважається), універсальними законами розподілу випадкових величин, що зустрічаються в різних природничих дослідженнях, є нормальний закон – розподіл Гауса і так званий логнормальний розподіл (див. рис. 8):

$$f\ x\ = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}},$$

$$f\ x\ = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{\ln x\ ^2}{2\sigma^2}}, \qquad x > 0$$

Часта зустрічальність нормального закону пояснюється тим, що коли розподіл випадкової величини пов'язаний із сумою незалежних процесів, розподіл наближається до нормального. Саме це твердження є змістом центральної граничної теореми теорії ймовірності.

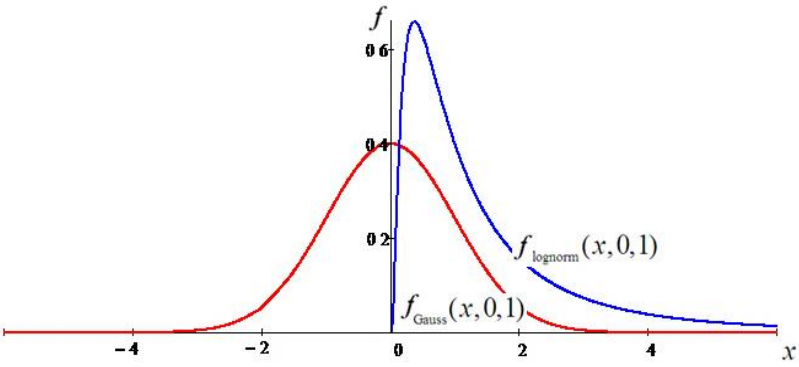


Рисунок 8 – Графіки нормального та логнормального розподілу. Середнє значення для нормального розподілу вибрано рівним нулю

Зауважимо, що часто в конкретних дослідженнях розподіл Гауса випадкової величини приймається через звичку або зручність.

Не менш універсальним, часто зустрічається законом розподілу випадкової величини є статечний (часто говорять гіперболічний) розподіл із щільністю ймовірності:

$$f\ x = \frac{B}{x^{\beta}},$$

або

$$P\ X \geq x = \frac{A}{x^{\alpha}}, \quad x > 0, \quad \alpha = \beta - 1,$$

де $P\ X \geq x$ – ймовірність того, що $X \geq x$, а A та α – деякі позитивні константи, параметри розподілу.

Відповідно,

$$P\ X \geq x = \int_{x}^{\infty} \frac{B}{x^{\beta}} dx,$$

$$\frac{dP\ x}{dx} = -f\ x\ .$$

Слід зазначити, що наведений вище розподіл розглядався Мандельбротом, як уточнення закону Ципфа та його часто називають розподілом Ципфа-Мандельброта, який відомий у лінгвістиці – це розподіл слів тексту за рангом. При цьому виявилося, що α – близька до одиниці величина, яка може змінюватись в залежності від властивостей тексту та мови.

Заради справедливості, слід зазначити, що статечні функції розподілу розглядалися ще Коші. Як наочний приклад розподілу Коші можна навести модель стрільби з обертового, з постійною кутовою швидкістю ω у горизонтальній площині, кулемету (див. рис. 9).

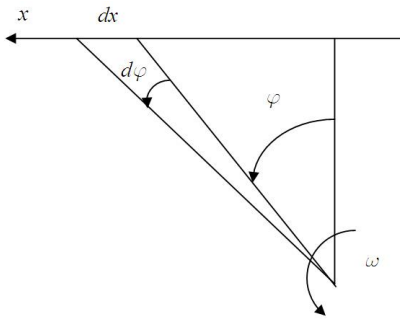


Рисунок 9 – Приклад розподілу Коші

Якщо, починаючи одиночні постріли, натискати на курок рівноймовірно за будь-якого його положення, то функція розподілу пострілів по кутку φ буде величиною постійною: $F(\varphi) = const$. З іншого боку, ймовірність попадання в нескінченно малу ділянку dx нескінченної плоскої мішені дорівнює $f(x)dx = F(\varphi)d\varphi$. Звідки, з урахуванням $x = a \cdot tg(\varphi)$, після перетворень знаходимо розподіл Коші:

$$f(x) = \frac{1}{\pi}\frac{a}{a^2 + x^2}, \quad -\infty < x < \infty.$$

Оскільки для цієї функції середнє $\langle x^\alpha \rangle$ від x^α ($\langle x^\alpha \rangle = \int_{-\infty}^{\infty} x^\alpha f(x)dx$) не визначено для $\alpha \geq 1$, то ні математичне очікування (тобто середнє від x^α), ні дисперсія $\langle x^2 \rangle$, жодних моментів старших порядків цього розподілу не визначені. У цьому випадку кажуть, що математичне очікування не визначене, а дисперсія - нескінченна.

Частковий випадок ступеневого розподілу – гіперболічний розподіл A/x названо на честь В. Парето, а

дискретний закон розподілу з ранжованою змінною був названий на честь Дж. Ципфа, який сформулював його для опису частоти вживання слів.

2.2 Мережа Ердеша-Реньї

Мережа (граф) Ердеша-Реньї (ER-мережа) це така мережа, в якій кожна пара вузлів з'єднана з ймовірністю p. У граничному випадку (при збільшенні кількості вузлів $N \to \infty$) функція розподілу ступенів вузлів має вигляд:

$$P_k = e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!}.$$

У реальному випадку для кінцевого значення числа вузлів слід розрізняти дві моделі мережі ER – модель Гільберта (G_{np} -модель) і власне модель Ердеша-Реньї (G_{nm}). У моделі G_{nm} фіксується ймовірність p. Для мережі з кінцевим значенням вузлів N Це означає, що $\langle k \rangle = pN$, причому кількість зв'язків M визначено тільки в середньому $\langle M \rangle = pN(N-1)/2$. У моделі Гільберта G_{np} задана не ймовірність p, а кількість зв'язків M, тепер ймовірність p визначена як $\langle M \rangle = pN(N-1)/2$, а середній ступінь вузла дорівнює $\langle k \rangle = 2M/N$. Відмінність між цими моделями аналогічна різниці між канонічним та мікроканонічним ансамблями у статистичній фізиці. У мікроканонічному ансамблі задана енергія системи, а в канонічному температура, при цьому енергія флуктує навколо середнього значення.

У граничному випадку $N \to \infty$, $p \to 0$, pN – кінцеве число не дорівнює нулю, обидва визначення мережі Ердеша-Реньї та Гільберта збігаються.

Мережа Ердеша-Реньї є «добре з'єднаною» – середня мінімальна відстань між вузлами порядку $\ln N << N$ ($N >> 1$).Середню мінімальну відстань між вузлами легко оцінити з

таких міркувань. У кожного сусіда якогось вузла є ще в середньому по $\langle k \rangle$ сусідів, до кожного з яких можна дійти за два кроки із цього вузла. За l кроків можна в середньому дійти до $\langle k \rangle^l$ вузлів.

Тоді для середньої мінімальної відстані l між вузлами мережі з N вузлів отримуємо:

$$l = \frac{\ln N}{\ln \langle k \rangle}.$$

Дуже невелике (логарифмічно мале в порівнянні з числом вузлів) значення мінімальної відстані робить випадкову ER-мережу так званим «малим світом». До кожного вузла мережі, що складається, наприклад, з $N = 10^9$ вузлів (порядок кількості людей Землі) із середнім числом зв'язків $\langle k \rangle = 100$ (приблизно стільки людей ми особисто знаємо) мінімальне середнє відстані дорівнює $l = \ln 10^9 / \ln 10^2 \approx 4.5$, тобто не більше п'яти кроків.

Зауважимо, що для регулярної мережі, наприклад, для квадратних ґрат це відстань значно більша $l \sim N^{1/2} >> \ln N$, для наведеного прикладу $l \sim \sqrt{10^9} \approx 30000 >> 4{,}5.$

Критичне значення p_c, у якому в мережі ER народжується гігантський кластер відразу ж із критерію Моллоя-Рида $p_c = \langle k \rangle / \left(\langle k^2 \rangle - \langle k \rangle \right)$. Так як $\langle k^2 \rangle = \langle k \rangle \left(\langle k \rangle + 1 \right)$, то для p_c отримуємо:

$$p_c = \frac{1}{\langle k \rangle}.$$

2.3 Модель мережі Барабаші-Альберт

Сценарій побудови мережі Барабаші-Альберт базується на двох механізмах – зростанні та переважному приєднанні (preferential attachment). Модель використовує алгоритм: зростання мережі відбувається починаючи з невеликої кількості вузлів n_0, до яких на кожному

часовому кроці додається новий вузол з $n < n_0$ зв'язками, що приєднуються до існуючих вузлів; переважне приєднання полягає в тому, що ймовірність приєднання $P(k_i)$ нового вузла до існуючого вузла i залежить від ступеня k_i вузла i:

$$P(k_i) = \frac{k_i}{\sum_j k_j}.$$

У знаменнику підсумовування ведеться по всім вузлам. Як комп'ютерні моделі, так і аналітичні рішення дають ступінчасту асимптотику розподілу щаблів вузлів з показником γ, близьким за значенням до 3.

Один із важливих напрямів аналізу мереж – це їхня візуалізація, яка відіграє значну роль, оскільки часто дозволяє зробити важливі висновки про характер взаємодії між вузлами мережі, не вдаючись до точних методів аналізу даних. Візуалізація є потужним інструментом для представлення складної структури мереж і допомогає дослідникам краще зрозуміти їх особливості та властивості.

При візуалізації моделі мережі часто виникає необхідність уявити вузли у двох вимірах, щоб побачити зв'язки та взаємодії між ними наочно. Це дозволяє виявити групи суб'єктів, які пов'язані один з одним, а також виявити особливості структури мережі, такі як центральні вузли або окремі групи.

Додатково можна застосовувати просторове впорядкування об'єктів в одному вимірі, ґрунтуючись на певних кількісних характеристиках. Наприклад, вузли можуть бути впорядковані за ступенем важливості або за кількістю зв'язків, що дозволяє краще візуалізувати мережну динаміку і зрозуміти, які вузли мають найбільший вплив на всю мережу.

З іншого боку, при візуалізації мереж застосовуються загальні для всіх мережевих діаграм методи для відображення кількісних і якісних характеристик об'єктів

та їхніх відносин. Наприклад, різні форми та кольори вузлів та зв'язків можуть представляти різні характеристики мережі, такі як типи вузлів або силу зв'язків між ними.

Таким чином, візуалізація є інструментом для вивчення та аналізу мережевих структур, дозволяючи дослідникам виявити цікаві патерни та взаємозв'язки між вузлами, що сприяє глибшому розумінню їх функціонування та динаміки.

Функція розподілу ступеня вузлів P_k для масштабно-інваріантної мережі (Scale-Free, SF) має вигляд:

$$P_k = \frac{1}{k^\gamma}, \qquad k \geq 0.$$

Такий розподіл добре відомий у теорії ймовірностей, як розподіл Парето. Зазвичай для реальних мереж показник γ лежить у межах від 2 до 3.

У разі, коли k вважатимуться безперервною змінною, позначимо її – x ($x > 0$), то необхідно врахувати, що у всіх реальних випадках існує мінімальне значення цієї змінної – $x_{\min}$. Нормувальна константа C для безперервного випадку визначається, при цьому, з умови $\int_{x_{\min}}^{\infty} p(x)dx = 1$ і дорівнює

$$p(x) = Cx^{-\gamma}, \qquad C = (\gamma - 1) / x_{\min}^{\gamma - 1},$$

звідки ясно, що цей розподіл має сенс лише за $\gamma > 1$.

Додаткове обмеження $\gamma > 2$ на показник накладає вимога мати кінцеве середнє значення $\langle x \rangle$:

$$\langle x \rangle = \int_{x_{\min}}^{\infty} x p(x)\, dx = \frac{\gamma - 1}{\gamma - 2} x_{\min}, \qquad \gamma > 2,$$

а для існування m-ого моменту:

$$\left\langle x^m \right\rangle = \frac{\gamma - 1}{\gamma - 1 - m} x_{\min}^m,$$

потрібно виконання умови $\gamma > 1 + m$.

З розподілом (1.2.5) пов'язано, так зване, правило в жартівливому варіанті – «20% людей випиває 80% пива», і передбачається, що такого роду співвідношення має місце для багатьох інших родів діяльності людини.

Справді, частка вузлів $S(x)$, що мають значення ступеня, більше ніж x, складає:

$$S(x) = \int_x^\infty p(x)dx = \left(\frac{x}{x_{\min}} \right)^{-\gamma+1}.$$

Ці вузли у сумі містять долю $W(x)$ від усіх зв'язків

$$W(x) = \frac{\int_x^\infty xp(x)dx}{\int_{x_{\min}}^\infty xp(x)dx} = \left(\frac{x}{x_{\min}} \right)^{-\gamma+2}.$$

Звідки одразу ж випливає

$$W = S^{\frac{\gamma-2}{\gamma-1}} = S^{1-\frac{1}{\gamma-1}}$$

На рисунку 10 наведено залежність W від S різних значень γ. Очевидно, чим більше значення показника γ, тим залежність $W = W(S)$ ближче до лінійної.

При показнику $\gamma = 2.161$ для $S = 0,2$ значення $W = 0.8$ відповідає правилу Парето (80/20). Чим більше значення показника γ, тим залежність $W = W(S)$ ближче до лінійної

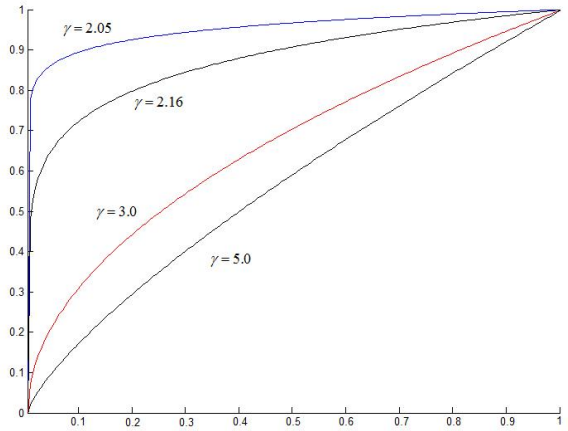


Рисунок 10 – Залежність $W = W(S)$

Пояснимо тепер, що означає термін «безмасштабна» по відношенню до мережі SF. Будемо, для наочності, вважати, що ступінь вузла (значення величини) - це багатство, яке має людина (даний вузол). Тоді можна обчислити відносну частку $S(x_M)$ людей з багатством більшим x_M, які володіють половиною всіх грошей.

З одного боку:

$$S(x_M) = \int_{x_M}^{\infty} p(x)dx = \left(\frac{x_M}{x_{\min}}\right)^{-\gamma+1},$$

з іншого боку, частка їхнього багатства $W(x_M) = 1/2$, тобто

$$W(x_M) = \frac{\int_{x_M}^{\infty} xp(x)dx}{\int_{x_{\min}}^{\infty} xp(x)d} = \left(\frac{x_M}{x_{\min}}\right)^{-\gamma+2} = \frac{1}{2}.$$

Підставляючи значення x_M з попереднього рівняння, знаходимо:

$$S(x_M) = 2^{-\frac{\gamma-1}{\gamma-2}}, \qquad S(x_M)\big|_{\gamma=2.161} = 6.7 \cdot 10^{-3},$$

тобто для значення показника $\gamma = 2.161$ половиною всіх грошей має менше 1% (0.67%) людей.

Назвемо цих людей «багатими». Безмасштабність означає, що серед «багатих» розподіл на «багатих» і «бідних» такий самий. Тобто, як легко підрахувати, що 0.67% серед «багатих» володіють половиною всього «багатства» «багатих» (тобто 1/4 від усього багатства).

Для будь-якого значення для безмасштабного розподілу

$$P(bx) = g(b)P(x),$$

тобто зміна масштабу ($x \to bx$) призводить лише до множення розподілу на константу.

У дискретному варіанті SF – розподіл нормовано трохи складніше:

$$1 = \sum_{k=1}^{\infty} P_k = C\sum_{k=1}^{\infty} \frac{1}{k^{\gamma}} = C\zeta(\gamma),$$

де $\zeta(\gamma) = \sum_{k=1}^{\infty} k^{-\gamma}$ – ζ -функція Римана.

Таким чином, отримуємо:

$$P_k = \frac{k^{-\gamma}}{\zeta(\gamma)}$$

У разі, коли обрізається «бідний хвіст», тобто. не розглядаються вузли зі ступенями меншими $k_{\min}$, розподіл p_k записується так[10]:

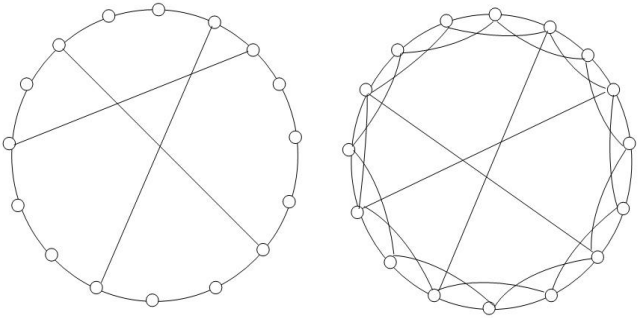
[10] *Dorogovtsev, A.V. Goltsev, J.F.F. Mendes.* Critical phenomena in complex networks, arXiv:0705.0010

$$p_k = \frac{k^{-\gamma}}{\zeta(\gamma, k_{\min})}, \qquad k \geq k_{\min},$$

$$\zeta(\gamma, k_{\min}) = \sum_{k=k_{\min}}^{\infty} k^{-\gamma}.$$

2.4 Мережа малого світу Уаттса – Строгатца

Ця модель є одновимірною регулярною решіткою, що складається з N вузлів, де кожен вузол з'єднаний тільки з k своїми найближчими сусідами і накладені періодичні граничні умови, тобто ґрати згорнули у кільце (див. рис. 11). Після чого, кожен зв'язок з ймовірністю $\phi << 1$ перекидали на інший випадково обраний вузол. Щоправда, за такої процедури є ймовірність появи ізольованих вузлів.



а б

Рисунок 11 – Приклад малого світу з трьома перекидами ($N = 16$): а – кожен вузол з'єднаний зі своїми найближчими сусідами ($k = 2$), б – кожен вузол з'єднаний із чотирма сусідами ($k = 4$)

Формально відстань до них від будь-якого вузла буде нескінченною. Щоб уникнути цього, Ньюман (Newman) і Уаттс запропонували зв'язки не перекидати, а просто

додавати[11]. Зупинимося на цьому варіанті моделі докладніше.

Середня відстань між кінцями доданих зв'язків є: $N/(2\cdot\phi\cdot N)=1/(2\cdot\phi)$. Для зручності опустимо двійку в знаменнику та визначимо ξ як:

$$\xi=\frac{1}{\phi}.$$

Для $k\geq 2$ природне узагальнення дає:

$$\xi=\frac{1}{k\cdot\phi}.$$

Так як існує тільки один характерний розмір системи ξ, то і безрозмірне відношення середньої відстані між вузлами графа до всіх вузлів графа l/N може залежати тільки від безрозмірної величини N/ξ. Тобто можна написати:

$$l=N\cdot f\left(\frac{N}{\xi}\right),$$

де $f(x)$ – скейлінгова функція з наступними асимптотиками:

$$f(x)\sim\begin{cases}cons, x<<1\\ \dfrac{\log(x)}{x}, x>>1\end{cases}$$

Існує багато способів визначення кореляційного радіуса. Припустимо, що $\xi\sim\phi^{-\tau}$. Покажемо за допомогою ренорм-

[11] Newman, M. E. J. & Watts, D. J. Renormalization group analysis of the small-world network model. Physics Letters A 263, 341 (1999).

групового перетворення (для $k=2$), що $\tau=1$[12]. Тож нехай маємо:

$$l = N \cdot f\left(N \cdot \phi^{\tau}\right)$$

Виконаємо ренорм-групове перетворення, показане на рисунку 12, а саме: об'єднаємо в пари сусідні вузли у графі, при цьому в новому графі вузли з'єднані доданим зв'язком, якщо у вихідному графі такий зв'язок був хоча б у однієї з пар.

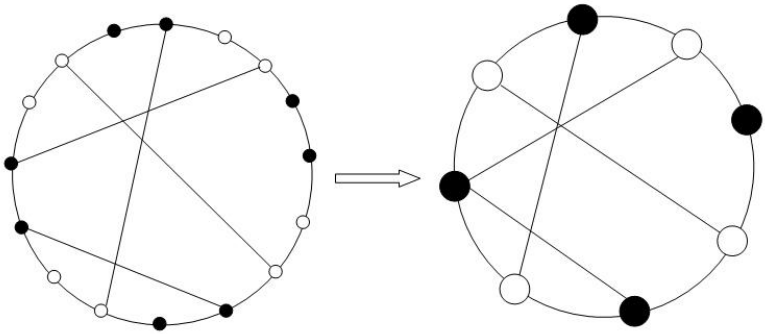


Рисунок 12 – Ренорм-групове перетворення. Два сусідні чорні вузли об'єднуються в один великий чорний вузол у новому графі та аналогічно для білих вузлів

За такого перетворення з очевидністю можна записати, що:

$$N' = \frac{1}{2}N\,, \quad \phi' = 2 \cdot \phi\,,$$

де штриховані величини відносяться до правого графа на рисунку 12.

Також зрозуміло, що і середня мінімальна відстань у новому графі l' відрізнятиметься вдвічі:

[12] *M.E.J. Newman and D.J. Watts.* Scaling and percolation in the small-world network model. Phys. Rev. E 60, 7332–7342 (1999)

$$l' = \frac{1}{2} l.$$

Виходячи з цього, отримуємо:

$$\tau = \frac{\log\left(N/N'\right)}{\log\left(\phi'/\phi\right)} = 1$$

Для $k > 2$ можна провести аналогічне перетворення, тільки тепер необхідно групувати не по два вузли, а за $k/2$ вузлами. Результат природно залишається тим самим – $\tau = 1$.

Вище описаною моделлю малого світу можна узагальнити великі розмірності. Так, наприклад, у двовимірному випадку це може бути регулярні квадратні грати з додатковими зв'язками, як показано на рисунку 13.

Тут далі під k розуміється ступінь вузла, тобто число найближчих сусідів. Але треба мати на увазі, що в літературі при описі малого світу в якості k часто розуміють кількість сусідів в одному напрямку.

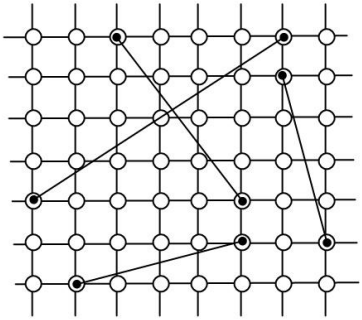


Рисунок 13 – Приклад двовимірного ($d = 2$) малого світу, $k = 4$

Тоді для значення ξ матимемо:

$$\xi = \frac{1}{\left(k \cdot \phi \cdot d\right)^{1/d}},$$

де – d розмірність малого світу. Тоді для l вираз набуде вигляду:

$$l = \frac{N}{k} \cdot f\left(\left(\phi \cdot k\right)^{1/d} \cdot N\right).$$

Для одномірного малого світу можна знайти в явному вигляді кластерність C і середньо мінімальну відстань l для малого світу, наведемо кінцеві вирази:

$$c(\phi) = \frac{3 \cdot (k-2)}{4 \cdot (k-1)} \cdot (1-\phi)^3,$$

$$l(\phi) = \frac{1}{\phi \cdot k \cdot \sqrt{1 + \frac{2}{N \cdot \phi}}} \cdot \operatorname{arth}\left(\frac{1}{\sqrt{1 + \frac{2}{N \cdot \phi}}}\right).$$

Як можна побачити, функція $f(x)$ має наступні асимптотики:

$$f(x) \sim \begin{cases} \frac{1}{4}, x << 1 \\ \frac{\log(2 \cdot x)}{4 \cdot x}, x >> 1 \end{cases}$$

Детальне виведення цієї формули засноване на наближенні теорії середнього поля описане у роботі Ньюмана та його співавторів[13].

[13] *M. E. J. Newman, C. Moore and D. J. Watts.* Mean-field solution of the small-world network model (2000). Phys. Rev. Lett. 84, 3201–3204.

На рисунку 14 наведено графіки нормованих залежностей кластерності $\overline{C}$ та середньої відстані $\overline{l}$ від концентрації перебросів p.

Для звичайного регулярного графа (наприклад, сітки) характерна велика середня мінімальна відстань та велика (близька до одиниці) кластерність. А для цілком випадкового графа обидві ці величини падають. Тому, зазвичай велике l асоціюється з великим C і, навпаки, мале l з малим C. Тут ми бачимо (див. рис. 14) що є велика область значень, у якій щодо велике C і мале l.

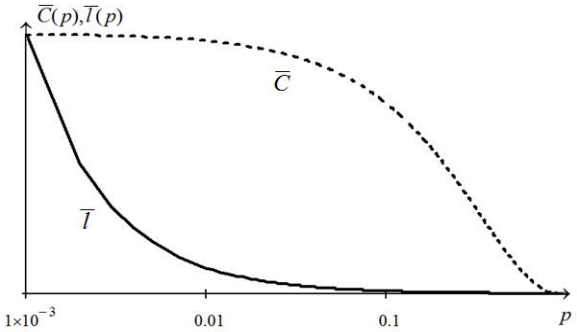


Рисунок 14 – Пунктирна лінія – нормована кластерність. Суцільна лінія – нормована середньо мінімальна відстань. Нормування відбувається на регулярний граф (без перекидів) – $\overline{C}(0)=1,\ \overline{l}(0)=1$

2.5 Мережа (u, v) – квіток і дерев

Мережа (граф) із SF-розподілом ступеня вузлів може бути як випадковою, так і детермінованою. Існує багато артефактних прикладів детермінованих мереж з SF розподілом, наприклад, так звані (u,v)-квіти і (u,v)-дерева. Побудова (u,v)-квітки починається з ланцюжка із $w=u+v$ зв'язків, після чого, на кожному наступному

кроці, кожен зв'язок замінюється на ланцюжок з двох частин u і v, як показано на рисунку 15.

На рисунку 15 показано кілька кроків побудови (1,2), (1,3), та (2,2) квіток.

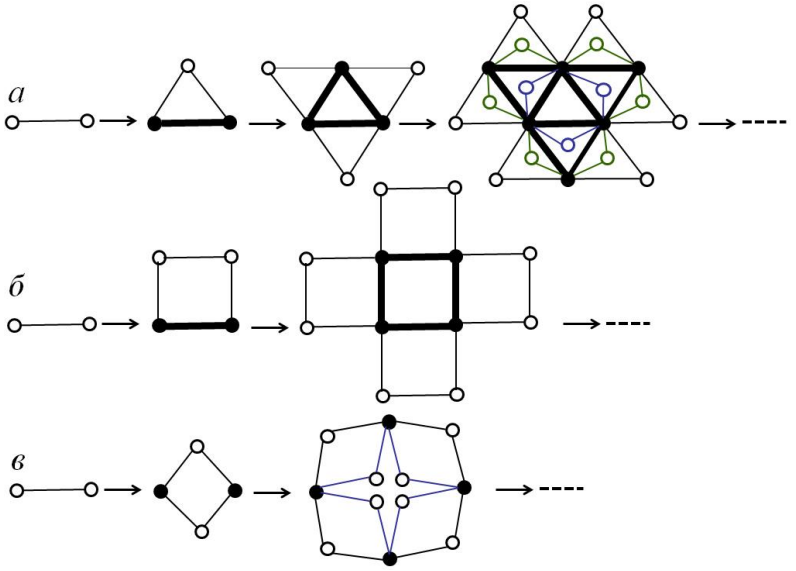


Рисунок 15 – Приклади побудови (u,v)–квіток: а – (1,2)-квіток; б – (1,3)-квіток; в – (2,2)-квіток:
○ – вузли, що з'являються на цьому кроці побудови;
● – "старі вузли"; жирні лінії – зв'язки, що з'являються на цьому кроці побудови

На n-ому кроці побудови кількість зв'язків у такій мережі $M_n = (u+v)^n$.

У той же час кількість вузлів на n-му кроці – N_n підпорядковується наступному ітераційному співвідношенню

$$N_n = wN_{n-1} - w, \quad w = u+v,$$

звідки

$$N_n = \left(\frac{w-2}{w-1}\right) w^n + \frac{w}{w-1}$$

Відповідно до правила побудови (u,v)-квітів, на n-ому кроці зустрічаються вузли тільки зі ступенями

$$k = 2^m,\ m = 1,\ 2,\ ...,\ n,$$

позначаючи N_n – кількість вузлів на кроці n зі ступенем 2^m можна записати наступне ітеративне співвідношення:

$$N_n(m) = N_{n-1}(m-1) + (w-2) w^{n-1} \delta_{m,1}.$$

Звідки:

$$N_n(m) = \begin{cases} (w-2) w^{n-m}, & m < n, \\ w, & m = n. \end{cases}$$

Так як при $n >> 1$ $N_n(m) \sim p(k)$ з $N_n(m) \sim w^{-m}$ і $k = 2^m$ випливає, що

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

де

$$\gamma = 1 + \frac{\ln w}{\ln 2} = 1 + \frac{\ln(u+v)}{\ln 2}.$$

І, таким чином, (u,v)-квіти дійсно є *SF*-мережами.

Для мережі $p_k \sim k^{-\gamma}$ є вузли з максимальним значенням ступеня $k_{\max}$. Значення $k_{\max}$ залежить від повного числа вузлів N. Ця залежність степенева і в такий спосіб залежить від показника γ:

$$k_{\max} \sim N^{-\frac{1}{\gamma-1}}.$$

Коефіцієнт кластеризації для SF для випадкового графа визначається як:

$$C = \frac{\langle k \rangle}{N} \left(\frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle^2} \right)^2,$$

де для $\gamma < 3$ маємо $\langle k^2 \rangle \sim k_{\max}^{3-\gamma}$ і, таким чином:

$$C \sim N^{-\beta},\ \beta = \frac{3\gamma - 7}{\gamma - 1}.$$

2.6 Моделювання живучості мереж

До основних ключових характеристик системи належить живучість, яку розуміють як здатність системи відповідати новим умовам функціонування та протистояти негативним впливам під час виконання основної цілі.

Структурна живучість розглядається як характеристика системи зберігати свою функціональність під час пасивного протистояння пошкодженням окремих елементів. Оцінка структурної живучості полягає у визначенні кількості відмовлених елементів, при цьому забезпечуючи незмінність працездатності системи. Можна зобразити схему функціонування складної системи за допомогою мережі, яка складається з вузлів та зв'язків, і це визначає фізичну структуру цієї системи.

У загальноприйних моделях вважають, що коли всі зв'язки, що виходять з певного вузла, видаляються, це відокремлює його, розриваючи усі шляхи до інших вузлів. У такому випадку мережа стає роз'єднаною, і її живучість дорівнює нулю.

Проте можливо розглядати інший критерій, який називається пороговим. Цей підхід передбачає оцінку за розміром найбільшої зв'язної компоненти в мережі. Хоча зв'язність всієї мережі може бути порушена, система все ж залишатиметься функціонально придатною, якщо розмір найбільшої зв'язаної компоненти (виміряної кількістю

вузлів) не впаде нижче певного заздалегідь визначеного порогу.

У роботах[14,15] запропоновано стандартне визначення показника живучості, при якому будь-яке видалення зв'язків (ребер графа) не призводить до втрати зв'язності.

Живучість мережі $R(G, p)$ визначається як ймовірність того, що граф мережі G залишиться зв'язним після видалення кожного зв'язку (ребра) з однаковою ймовірністю p. Обчислення $R(G, p)$ може бути здійснено за допомогою підрахунку остовних графів G. На практиці живучість тісно пов'язана з поліномом Татта-Уїтні. (Tutte-Whitney)[16], який є інваріантом графа та описує відповідні комбінаторні властивості.

Точний розрахунок живучості системи є NP-складною задачею, вартість вирішення якої росте експоненційно зі збільшенням кількості вузлів і зв'язків, оскільки для обчислення живучості полінома графа, що містить n ребер, потрібно пройти через всі можливі остовні підграфи графа G.

Розглядається підхід, який ґрунтується на використанні імітаційного моделювання:

1. Нехай модель системи – мережа, що складається з вузлів і зв'язків (ненаправлених) $S = (V, E)$. Вузли – це деякі однорідні функціональні компоненти.
2. Розглянемо «потужність» системи як кількість вузлів в найбільшій зв'язній компоненті V_s.

[14] Oxley, J.G.: Matroid Theory. Oxford Science Publications (1992).
[15] Sekine K., Imai H., Tani S.: Computing the Tutte Polynomial of a Graph of Moderate Size. In: 6th International Symposium on Algorithms and Computation (ISAAC'95). Lecture Notes in Computer Science. 1004. pp. 224–233 (1995).
[16] Tutte, W.T.: A Contribution to the Theory of Chromatic Polynomials. Canadian Journal of Mathematics, 6. 80-91 (1954).

3. Система перебуває у стані «життя», функціонально здатна, якщо питома «потужність» системи не менше деякого порогу τ, тобто $|V_s|/|V_o| \geq \tau$, де V_o – первинний розмір мережі.

4. На зв'язки мережі здійснюється деструктивний вплив. Кожний зв'язок може бути видалений з ймовірністю p.

Для кожної конкретної системи можна визначити міру живучості системи при заданому порозі τ, тобто ймовірність видалення окремих елементів (зв'язків) $p*$, при якій система перестає бутину «живою», тобто $|V_s|/|V_o| < \tau$.

Для моделювання досліджуються три артефактних мережі, а саме, Барабаші-Альберт, Ердеша-Рен`ї та Уаттса-Строгатца. Ці мережі можуть самі по собі є моделями багатьох реальних мереж.

Мережа Барабаші-Альберт

Багатьом артефактним мережам притаманний степеневий закон розподілу. Такий розподіл пояснюється ефектом переважного приєднання (preferential attachment). До таких мереж відносяться мережі Барабаші-Альберта (A.L. Barabási, R. Albert). Для побудови цих мереж використовується спеціальна процедура, яка полягає в тому, що з початку малій кількості вузлів поступово додаються нові, зв'язки від яких з більшою ймовірністю приєднуються до тих вузлів, у яких зв'язків більше[17]. Модель переважного приєднання Барабаші-Альберт реалізована, зокрема, мовою R в пакеті igraph[18].

Мережа Ердеша–Рен`ї

[17] Albert-László Barabási, Réka Albert: Emergence of scaling in random networks. Science, 286, 5439. 509-512 (1999).
[18] Douglas A. Luke. A User's Guide to Network Analysis in R. Springer; 1st ed. 2015 edition (December 21, 2015). 250 p. ISBN: 3319238825

Мережу Ердеша–Ренї, ER (P. Erdős, A. Rnyi) можна побудувати різними шляхами, наприклад, розподіливши випадковим чином M зв'язків між N вузлами. Її іноді називають моделлю пуассоновского випадкового графа (Poisson random graph model) через пуассонівський розподіл ступенів при $N \to \infty$. Модель випадкового графа ER реалізована мовою R в пакеті igraph.

Мережа «малого світу»

Д. Уаттс і С. Строгатц (D.J. Watts, S. Strogatz) формалізували феномен, характерний для багатьох реальних мереж, названий ефектом малих світів (Small Worlds). Ними була запропонована процедура побудови наочної моделі мережі, для якої притаманний цей феномен. Модель являє собою одномірну регулярну решітку, що складається з N вузлів, де кожен вузол з'єднаний тільки зі своїми чотирма найближчими сусідами, а також накладені граничні умови – решітка згорнута в кільце. В рамках моделі виконується така процедура: з імовірністю p відбувається перемикання (rewiring) невеликої кількості зв'язків (ребер), у ході чого вони видаляються і замінюються іншими зв'язками, які з'єднують випадково обрані вузли. У початковому стану ця мережа є регулярною, якщо кожний її вузол з'єднаний з чотирма сусідніми. Потім в мережі деякі «ближні» зв'язки випадковим чином замінюються «далекими» – саме в цьому стані виникає феномен «малих світів» (при $p \in (0.01,\ 0.1)$). При подальшому збільшенні p утворюється мережа, яка за властивостями близька до випадкової мережі Ердеша-Ренї.

Було проведено імітаційне моделювання процесу руйнування трьох мереж, а саме, Барабаші-Альберт, Ердеша-Ренї, Уаттса-Строгатца. Моделювання проводилося мовою програмування R з використанням бібліотеки igraph. Результати моделювання наведені у таблиці 1 та на рисунках 16-18.

Таблиця 1 - Результати моделювання

Назва моделі	**Параметри**	**Наближена формула**	**Точність R2**
Ердеша-Рен`ї	N=200, M=500	–3*10-8x3 + 10-5x2 – 0,0011x + 1,0091	0,99
Уаттса-Строгатца	N=200	10-7x3 – 6*10-5x2 + 0,0061x + 0,8768	0,98
Барабаші-Альберт	N=200	–4*10-7 x3 + 0,0001 x2 – 0,0187x + 1,0029	0,97

Якщо задати поріг руйнування, наприклад, наступним чином, мережа функціональна, якщо розмір найбільшої зв'язної компоненти V_s становить 0.2 від початкового розміру мережі V_0, тобто: $|V_s|/|V_o| \geq \tau$, $\tau = 0.2$ то, відповідно, отримуємо значення порогової ймовірності p^* для мереж:

Ердеша-Рен`ї: ≈ 0.8;
Уаттса-Строгатца: ≈ 0.7;
Барабаші-Альберт: ≈ 0.5.

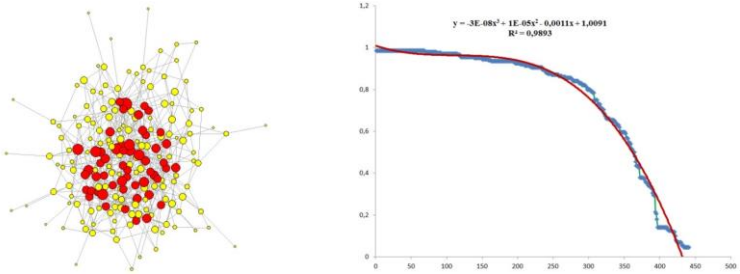


Рисунок 16 – Мережа Ердеша-Реньї та графік залежності «потужності» мережі (вертикальна вісь) від кількості вилучених зв'язків (горизонтальна вісь)

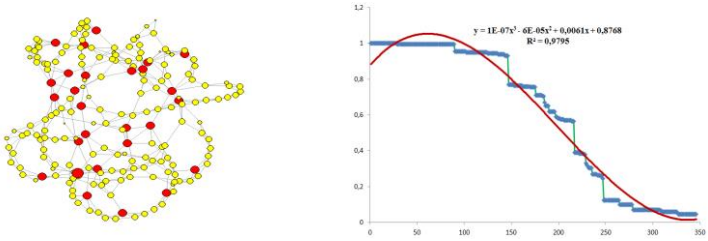


Рисунок 17 – Мережа Уаттса-Строгатца і графік залежності «потужності» мережі (вертикальна вісь) від кількості вилучених зв'язків (горизонтальна вісь)

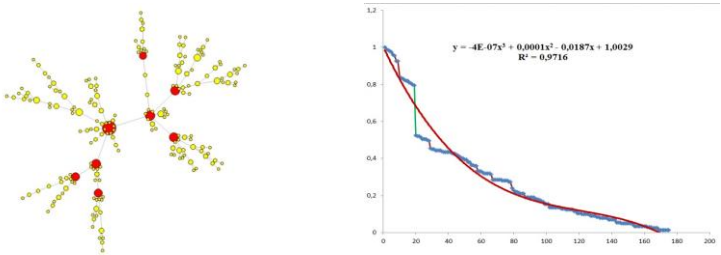


Рисунок 18 – Мережа Барабаші-Альберт і графік залежності «потужності» мережі (вертикальна вісь) від кількості вилучених зв'язків (горизонтальна вісь)

Як видно, в кожному разі, криві з високою точністю апроксимуються кубічними многочленами, тобто для досить точного наближення досить трьох ступенів многочлену Татте-Уїтні. З огляду на топологію мереж і наведені раніше аналітичні оцінки, можна зробити висновок, що найбільша структурна живучість серед трьох розглянутих мереж властива випадковій мережі Ердеша-Рен`ї (ця мережа має найбільшу кількість ребер). На другому місці – мережа малого світу, ця мережа, в якій вузли мають середню ступінь 2 з розподілом, близьким до Пуассонівського. І найгірші показники живучості у мережі Барабаші-Альберт. Слід звернути увагу на те, що в останньому випадку, на відміну від інших, спостерігається

«опукла вниз» функція живучості. Це свідчить про те, що розглянута в цій роботі «потужність» як міра живучості, різко знижується вже навіть при невеликих значеннях ймовірності деструктивних впливів.

2.7 Перколяційні мережі

Коротко зупинимося ще на одному типі мереж – перколяційних. У найпростішому варіанті перколяційна мережа будується з регулярних, наприклад, квадратних ґрат шляхом виривання (руйнування) випадковим чином обраних зв'язків. На рисунку 19 зв'язки, що залишилися, позначені жирною лінією, вирвані тонкою.

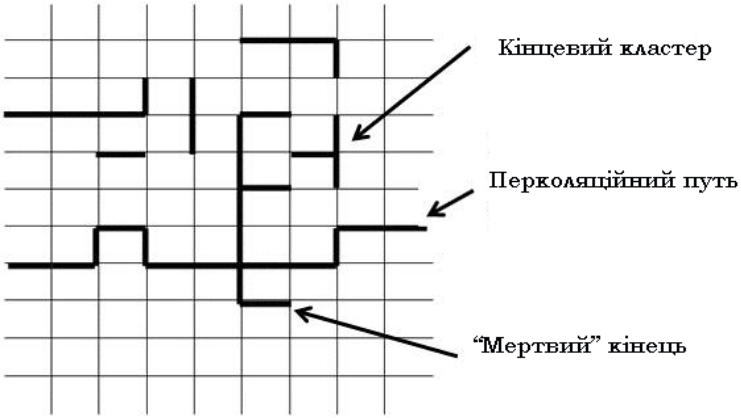


Рисунок 19 – Квадратні грати з випадково вирваними зв'язками (тонкі лінії)

Нехай при створенні перколяційної мережі з ймовірністю 1 – p вириваються зв'язки (вузли), тоді вона складатиметься з $p \cdot N$ (N – ціла кількість зв'язків (вузлів)), так званих, чорних зв'язків (вузлів).

Для кожної такої мережі (квадратної, трикутної, шестикутної, кубічної, …) існує таке значення p_c (поріг

протікання), що при $p > p_c$ можна пройти по чорних зв'язках через всю мережу, а при $p < p_c$ – ні.

На рисунку 20 показана перколяційна мережа великого розміру для двох випадків a – нижче за поріг протікання $(p < p_c)$ та – вище $(p > p_c)$.

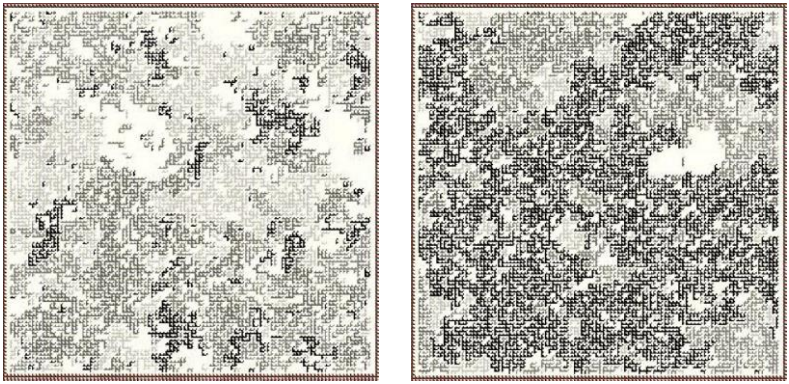


Рисунок 20 – Перколяційна мережа великого розміру. Зліва, випадок, коли концентрація зв'язків менша за пороговий, справа – більше

Для різних решіток – трикутної, квадратної, кубічної тощо, свій поріг протікання, чисельне значення якого можна отримати (за рідкісним винятком) лише шляхом чисельного моделювання. У таблиці 2 наведено значення порогу протікання для різних решіток.

Підкреслені значення p_c, які визначені точно. Перший стовпець відповідає вузлам, другий – зв'язкам.

При обчисленні p_c розмір решітки вибирається досить великим (теоретично – нескінченним) так, що значення p_c перестає бути залежним від повного числа вузлів N або зв'язків.

Таблиця 2 - Чисельне значення порогу протікання p_c

Решітка	Вузли	Зв'язки
Шестикутна	0.69	0.65
Квадратна	0.59	$\frac{1}{2}$
Трикутна	$\frac{1}{2}$	0.35
Кубічна	0.31	0.25
$d=4$ гіперкубічна	0.197	0.16
$d=5$ гіперкубічна	0.14	0.12
$d=6$ гіперкубічна	0.11	0.09

Головне питання в перколяції – це утворення, так званого, нескінченного кластеру, що дозволяє пройти по зв'язках через всю мережу (говорять з нескінченності в нескінченність, маючи на увазі нескінченний розмір мережі). Число зв'язків у нескінченному кластері відповідає порядку всіх зв'язків мережі.

Щодо цього, нескінченний кластер аналогічний Giant Cluster у складних мережах, наприклад у мережі ER. Існує як загальне так і відмінне у властивостях нескінченного кластеру в теорії протікання та Giant Cluster в теорії складних мереж.

Позначаючи для стислості Giant Cluster як GC і нескінченний кластер як PC відзначимо наступне.

Для перколяції на ґратах Келлі $p_c = 1/(z-1)$, де z – координатне число, для GC ER-мережі координатне число із N зв'язків дорівнює $N-1$, а для $N >> 1$ $p_c = 1/N$. Таким чином, при збільшенні N зменшується p_c, що аналогічно збільшенню просторової розмірності перколяційної мережі. І тут теорія складних мереж (ER) при $N \to \infty$ є аналогічною теорії перколяції в нескінченно вимірному просторі.

Як у задачі про GC, так і в задачі про PC, при $p < p_c$ ймовірність появи GC і PC дорівнює нулю.

Вище порога протікання, при $p > p_c$ розмір GC дорівнює $\left(f\left(p_c N\right) - f\left(pN\right)\right)N$, де f – функція, що експоненційно зменшується з $f\left(1\right) = 1$, в той час як розмір PC дорівнює $\left(p - p_c\right)N$. Ще одна відмінність полягає в структурі GC – це дерева, у той час як PC має фрактальну структуру.

Більш детально перколяційні мережі описані у 8-му розділі цього навчального посібника.

2.8 Обчислення характеристик мереж

Нижче будуть перераховані основні характеристики мереж і одночасно вони будуть продемонстровані на шести простих прикладах. Для того, щоб «відчути», що саме описують ті чи інші характеристики.

Далі шість прикладів мереж будемо позначати номерами 1–6 (див. рис. 21).

Матриці суміжності мереж, зображених на рисунку 6 наведені нижче:

$$\boldsymbol{A}_1 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \quad \boldsymbol{A}_2 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix},$$

$$\boldsymbol{A}_3 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{A}_4 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix},$$

$$\boldsymbol{A}_5 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{A}_6 = \begin{pmatrix} 0 & 2 & 0 & 0 & 0 \\ 2 & 0 & 8 & 0 & 7 \\ 0 & 8 & 0 & 9 & 1 \\ 0 & 0 & 9 & 0 & 4 \\ 0 & 7 & 1 & 4 & 0 \end{pmatrix}.$$

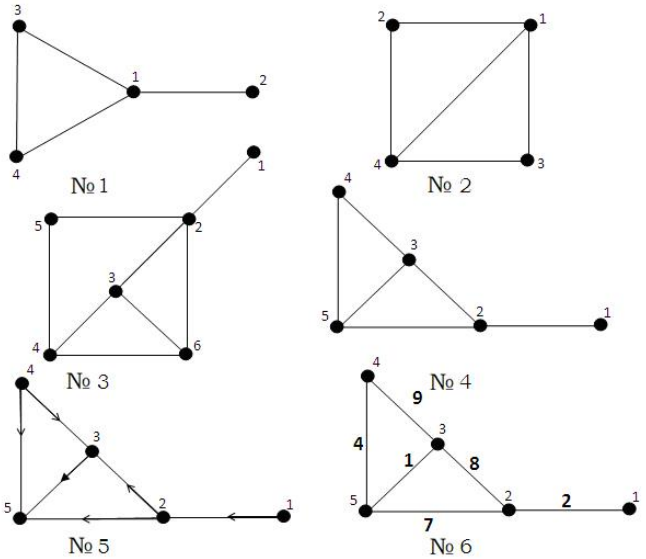


Рисунок 21 – Приклади найпростіших мереж: мережа № 5 – так званий орграф (направлений граф), мережа № 6 – мережа з вагами зв'язків (зазначені цифри на зв'язках)

Слід звернути увагу, що для мережі № 5 матриця суміжності A_5 не симетрична, а елементи матриці суміжності A_6 – це ваги зв'язків.

Розподіл вузлів за їхніми ступенями $P(k)$ наведено на рисунку 6, де також наведена така характеристика мереж, як середня кількість зв'язків на один вузол:

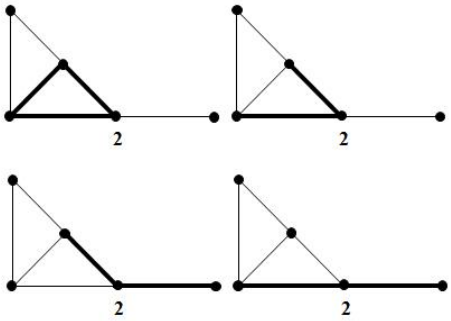
$$\langle k \rangle = \frac{1}{N}\sum_{i=1}^{N} k_i = \sum_{i=1}^{N} k_i P(k_i),$$

яка обчислюється як відношення всіх зв'язків у мережі до вузлів і, одночасно, як середнє дискретної змінної k_i з функцією розподілу $P(k_i)$.

Важливою характеристикою вузла є його коефіцієнт кластеризації – C_i, що характеризує зв'язність між собою сусідів цього вузла i. Коефіцієнт кластеризації C_i може бути записаний як відношення числа трикутників з вершиною до вилок (два зв'язки, що виходять з вузла) з основою в цьому вузлі:

$$C_i = \frac{\textit{кількість трикутників з вершиною } i}{\textit{число вилок з основою в } i}.$$

Розглянемо, наприклад, вузли та мережі № 4. Для вузла № 2 жирними лініями позначені вилки (їх три) та трикутники (він один).



Як можна переконатися, для другого вузла є три вилки та один трикутник, тому $C_2 = 1/3$.

Для третього вузла три вилки та два трикутники – $C_2 = 2/3$.

Ще одна мережа представлена на рисунку 22.

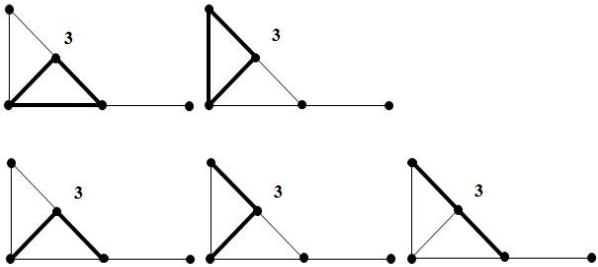


Рисунок 22 – Варіанти мережі

Коефіцієнт кластеризації всієї мережі обчислюється за такою формулою:

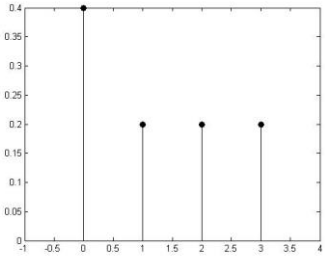
$$C = \langle C_i \rangle = \frac{1}{N} \sum_{i=1}^{N} C_i.$$

У таблиці 3 наведені коефіцієнти кластеризації всіх вузлів мереж №1, № 2, № 3 і № 4, і навіть коефіцієнт кластеризації всієї мережі *C*. На Рис. 21. Подано розподіл вузлів за ступенями для мереж, зображених на Рис. 23.

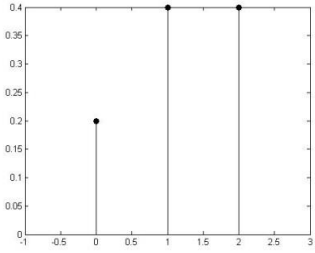
Як видно, найбільший коефіцієнт кластеризації має мережу № 2, у неї, у середньому, найбільше сусідів кожного вузла з'єднані між собою. Найменший коефіцієнт кластеризації має мережа №3.

Таблиця 3 - Коефіцієнти кластеризації

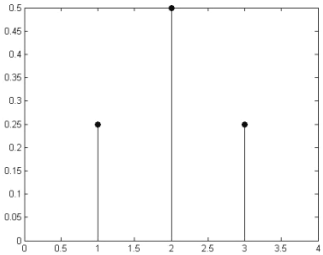
Мережа	C_1	C_2	C_3	C_4	C_5		C_6	C
№ 1	1/3	0	1	1	1			7/12
№ 2	2/3	1	1	1	2/3			10/12
№ 3	0	1/6	2/3	2/3	1/3	0	2/3	11/36
№ 4	0	1/3	2/3	2/3	1	2/3		8/15



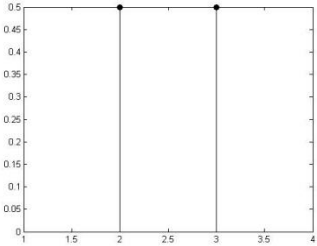
Мережа № 1, $\langle k \rangle = 2$



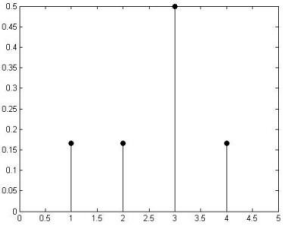
Мережа № 2, $\langle k \rangle = 2{,}5$



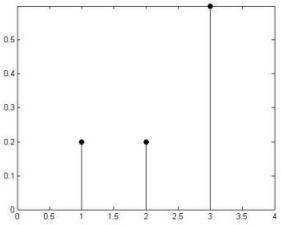
Мережа № 3,
$\langle k \rangle = 8/3 \approx 2.67$



Мережа № 4,
$\langle k \rangle = 12/5 = 2.4$



Мережа № 5, вхідні зв'язки
$\langle k \rangle = 1.2$



Мережа № 6, вхідні зв'язки
$\langle k \rangle = 1.2$

Рисунок 23 – Розподіл вузлів за ступенями для мереж, що розглядаються

Коефіцієнт кластеризації вузла можна обчислити і не вдаючись до перерахування трикутників і вилок безпосередньо з матриці суміжності:

$$C_i = \frac{\sum_{j,m} A_{ij} A_{jm} A_{mi}}{k_i (k_i - 1)}, \qquad k_i = \sum_j A_{ij},$$

де підсумовування ведеться по всіх вузлах.

На прикладах простих мереж легко безпосередньо переконатися, що різні методи визначення кластеризації дають тіж самі значення C_i.

Крім визначення коефіцієнта кластеризації всієї мережі в літературі зустрічається ще одне, близьке, але не тотожне визначення, що іноді називається транзитивністю – T:

$$T = \frac{\sum_{i=1}^{N} \left(\boldsymbol{A}^3 \right)_{ii}}{\sum_{i=1}^{N} k_i (k_i - 1)} \equiv \frac{Tr \boldsymbol{A}^3}{\sum_{i=1}^{N} k_i (k_i - 1)},$$

яке наступним чином виражається через кількість трикутників та вилок у всій мережі:

$$T = 3 \frac{\textit{число трикутників в мережі}}{\textit{число вилок в мережі}}.$$

Порівняння нижче наведено значення коефіцієнта кластеризації та транзитивності для мереж № 1, № 2, № 3 и № 4 наведено у таблиці 4.

Таблиця 4 - Коефіцієнти кластеризації

Мережа	№ 1	№ 2	№ 3	№ 4

C	$\frac{7}{12} \approx 0.58$	$\frac{10}{12} \approx 0.83$	$\frac{11}{36} \approx 0.31$	$\frac{8}{15} \approx 0.53$
T	$\frac{3}{5} = 0.6$	$\frac{3}{4} = 0.75$	$\frac{3}{35} \approx 0.086$	$\frac{3}{5} = 0.6$

Ще однією важливою характеристикою мереж є середня мінімальна відстань між їхніми вузлами:

$$l = \langle l_{ij} \rangle = \frac{1}{N(N-1)} \sum_{i \neq j} l_{ij},$$

де l_{ij} – найменша кількість кроків від вузла i до вузла j. При цьому кожен крок відповідає поодинокій відстані.

Аналогічне визначення можна використовувати і в навантажених мережах, при цьому кожен крок може відповідати як пропорційній відстані, так і зворотно пропорційній вазі:

$$\tilde{l} = \left(\frac{1}{N(N-1)} \sum_{i \neq j} \frac{1}{l_{ij}} \right)^{-1}.$$

У разі середнього арифметичного найбільший внесок будуть давати найбільш довгі шляхи (з найбільш коротких), у разі гармонійного середнього – найбільш короткі. Якщо в мережі є ізольовані вузли, дістатися яких неможливо і для яких природно вважати $l_{ij} = \infty$, то визначення мінімальної середньої відстані як арифметичної середньої перестає бути інформативним, оскільки навіть за наявності одного такого ізольованого вузла: $l = \langle l_{ij} \rangle = \infty$. У такій ситуації зручно користуватися визначенням середньої мінімальної відстані, як середньої гармонічної – $\tilde{l}$. Величину зворотну $\tilde{l}$ позначають $ge = 1/\tilde{l}$ і називають глобальною ефективністю мережі (efficiency).

Найменше кроків l_{ij} від вузла i до вузла j може бути записано у вигляді матриці $\boldsymbol{SP}$ (short path). Чисельні значення цієї матриці для мереж № 1 – № 6 дорівнюють:

$$\mathbf{SP}(1)=\begin{pmatrix}0&1&1&1\\1&0&2&2\\1&2&0&1\\1&2&1&0\end{pmatrix},\qquad \mathbf{SP}(2)=\begin{pmatrix}0&1&1&1\\1&0&2&1\\1&2&0&1\\1&1&1&0\end{pmatrix},$$

$$\mathbf{SP}(3)=\begin{pmatrix}0&1&2&3&2&2\\1&0&1&2&1&1\\2&1&0&1&2&1\\3&2&1&0&1&1\\2&1&2&1&0&2\\2&1&1&1&2&0\end{pmatrix},\qquad \mathbf{SP}(4)=\begin{pmatrix}0&1&2&3&2\\1&0&1&2&1\\2&1&0&1&1\\3&2&1&0&1\\2&1&1&1&0\end{pmatrix},$$

$$\mathbf{SP}(5)=\begin{pmatrix}0&1&2&\infty&2\\\infty&0&1&\infty&1\\\infty&\infty&0&\infty&1\\\infty&\infty&1&0&1\\\infty&\infty&\infty&\infty&0\end{pmatrix},\qquad \mathbf{SP}(6)=\begin{pmatrix}0&2&10&13&9\\2&0&8&11&7\\10&8&0&5&1\\13&11&5&0&4\\9&7&1&4&0\end{pmatrix}.$$

Зазначимо, що матриця $\mathbf{SP}(5)$ (для спрямованої мережі) містить нескінченні елементи, наприклад $l_{21}(5)=\infty$ (порівняйте з $l_{21}(4)=1$). Це, звичайно, означає, що від вузла 2 неможливо дійти до вузла 1.

У мережі з вагами при підрахунку l_{ij} кожен крок по зв'язку множиться на її вагу, тому підсумовується не просто кількість кроків (зв'язків), а їх ваги. Саме з цим пов'язано те, що найкоротший шлях у мережі № 6 між вузлами 3 і 4 включає два кроки з вагами 4 і 1, а не один крок з вагою 1.

Середні значення найкоротших шляхів (1.1.11) для мереж, що розглядаються, дорівнюють:

$$\langle l(1)\rangle = \frac{4}{3} \approx 1.3, \qquad \langle l(2)\rangle = \frac{7}{6} \approx 1.16, \qquad \langle l(3)\rangle = \frac{23}{15} \approx 1.53,$$

$$\langle l(4)\rangle = \frac{3}{2} = 1.5, \qquad \langle l(5)\rangle = \infty, \qquad \langle l(6)\rangle = 7.$$

Гармонічне середнє:

$$\langle \tilde{l}(1)\rangle = 1.2, \qquad \langle \tilde{l}(2)\rangle \approx 1.09, \qquad \langle \tilde{l}(3)\rangle \approx 1.32,$$

$$\langle \tilde{l}(4)\rangle \approx 1.28, \qquad \langle \tilde{l}(5)\rangle \approx 2.86, \qquad \langle \tilde{l}(6)\rangle \approx 3.85.$$

Ще однією важливою характеристикою мережі є так зване посередництво (betweenness). У таблиці 1.1.3 наведено чисельні значення посередництва всім вузлів мереж № 1 – № 6.

Таблиця. 1.1.3. Значення посередництва для мереж № 1 – № 6.

Вузол / № мережі	B_1	B_2	B_3	B_4	B_5	B_6
1	4	0	0	0		
2	1	0	0	1		
3	0	10	4/3	2	4/3	4/3
4	0	6	2	0	2	
5	0	2	0	0	0	
6	0	6	0	0	8	

Нагадаємо, що при обчисленні посередництва в мережі з вагою, найкоротший шлях розраховується з урахуванням ваги зв'язків.

Крім основної характеристики мережі – розподілу ступеня вузлів по зв'язкам – $P(k)$, вводиться також розподіл кінців зв'язків – $P_{nn}(k)$:

$$P_{nn}(k) = \frac{kP(k)}{\langle k \rangle}.$$

Цей розподіл називають також розподілом ступенів найближчих сусідів (degree distribution of the nearest neighbor), звідки й береться позначення $P_{nn}(k)$. На простому прикладі мережі № 1 легко переконатися, що наведене вище рівняння дійсно дає розподіл кінців зв'язків. Для мережі №1 маємо вісім кінців, тобто ймовірність попадання одного кінця зв'язку на вузол буде дорівнювати: у вузла один зв'язок, тому $P_{nn}(1) = 1/8$; у кожного з вузлів 3 і 4 – $2/8$, тобто $2 \cdot 2/8 = 1/2$, таким чином $P_{nn}(3) = P_{nn}(4) = 1/2$, $P_{nn}(2) = 1/2$ і для вузла 1 – $3 \cdot 1/8$, тобто $P_{nn}(3) = 3/8$.

З іншого боку, ці ж значення відразу можна розрахувати з наведеного вище рівняння, наприклад, для першого вузла:

$$P_{nn}(1) = \frac{1P(1)}{\langle k \rangle} = \frac{1 \cdot 1/4}{2} = \frac{1}{8},$$

та аналогічно для інших.

Якісно вираз для $P_{nn}(k)$ (1.1.13) можна пояснити так. Імовірність потрапляння випадково обраного зв'язку на вузли з k зв'язками пропорційно числу всіх зв'язків таких вузлів $P_{nn}(k) \sim NkP(k)$, де N – повне число вузлів у мережі. З огляду на те, що $\sum P_{nn}(k) = 1$, нормувальна константа дорівнює $1/\langle k \rangle$, звідки випливає наведене вище рівняння.

Розподіл $P_{nn}(k)$ для простих мереж збігається з $P(k)$ тільки у виняткових випадках (коли для кожного вузла

виконується рівність $k = \langle k \rangle$), наприклад, для нескінченної квадратної решітоки.

Маючи розподіл кінців зв'язків $P_{nn}(k)$ можна запровадити середню за цим розподілом ступінь вузлів:

$$\langle k \rangle_{nn} = \sum_{k=1}^{N} k P_{nn}(k) = \sum_{k=1}^{N} k \frac{kP(k)}{\langle k \rangle} = \frac{\langle k^2 \rangle}{\langle k \rangle}.$$

Для некорельованої мережі знання розподілу кінців $P_{nn}(k)$ дозволяє знайти можливість з'єднання вузлів зі ступенями k і k' – $P(k,k')$. Так як ймовірність того, що зв'язок «устромиться» у вузол зі ступенем k дорівнює $P_{nn}(k) = kP(k)/\langle k \rangle$, ймовірність того, що одночасно він «устромиться» і у зв'язок зі ступенем дорівнює добутку k'.

$$P(k,k') = P_{nn}(k) P_{nn}(k') = \frac{kP(k)k'P(k')}{\langle k \rangle^2}.$$

Введемо тепер поняття середнього числа перших, других та наступних сусідів. Якщо вузол має ступінь k, то він має k усідів, тому середнє число найближчих (перших) сусідів так само z_1, як і має бути:

$$z_1 = \sum k \cdot p_k = \langle k \rangle.$$

Перші сусіди вузла, своєю чергою, мають своїх сусідів, яких логічно назвати іншими найближчими до вихідного вузла сусідами. Якщо перший сусід має ступінь k, то у нього сусідів $k-1$ (один із зв'язків «пішла» до вихідного вузла).

Число перших сусідів зі ступенем k для всіх вузлів дорівнює $N \cdot k \cdot p_k$, тому число всіх інших сусідів дорівнює:

$$z_2 = \frac{1}{N} \sum (k-1) k \cdot p_k = \sum \left(k^2 - k\right) p_k = \langle k^2 \rangle - \langle k \rangle.$$

Відношення середньої кількості других сусідів до перших:

$$B = \frac{z_2}{z_1} = \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle}.$$

називають коефіцієнтом розгалуження (branching coefficient), він характеризує ступінь «розмноження» зв'язків у міру віддалення від вузла.

Коефіцієнт розгалуження можна отримати і з таких міркувань. Імовірність $P_{nn}(k)$ можна трактувати як ймовірність $Q(k-1)$ того, що зв'язок від деякого обраного вузла A увійде у вузол B з $k-1$ зв'язками, так що вузол матиме у сумі зв'язків:

$$Q(k-1) = P_{nn}(k) = \frac{kP(k)}{\langle k \rangle}.$$

Тоді середня кількість зв'язків, що виходять з вузла B (без урахування зв'язку з вузлом A) дорівнює:

$$\sum_{k=0}^{N} kQ(k) = \sum_{k=0}^{N} \frac{k(k+1)P(k+1)}{\langle k \rangle} =$$

$$= \sum_{k=1}^{N} \frac{(k-1)kP(k)}{\langle k \rangle} = \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle}.$$

Остання рівність справедлива при $N >> 1$.

Знаючи коефіцієнт розгалуження B легко обчислити середню кількість наступних (третіх, четвертих тощо) сусідів:

$$z_m = Bz_{m-1} = \frac{z_2}{z_1} z_{m-1} = \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} z_{m-1},$$

$$z_m = \left(\frac{z_2}{z_1} \right)^{m-1} z_1.$$

Вираз для середньої кількості $m-$них сусідів дозволяє отримати відповідь на дуже важливе питання – чи існує в мережі так званий гігантський кластер (Giant Connected Cluster або Giant Connected Component, GC).

При яких умовах існує у мережі кластер (пов'язана між собою частина вузлів), що включає число вузлів порядку всього числа вузлів в мережі? При обчисленні дедалі більш далеких і далеких сусідів ($m >> 1$) можливі два сценарії – при $z_2 > z_1, \quad z_m >> 1$ і при $z_2 < z_1, \quad z_m << 1$.

У другому випадку (передбачається, що $N >> 1$) випливає:

$$\sum_m z_m = z_1 \sum_m \left(\frac{z_2}{z_1} \right)^{m-1} \approx \frac{z_1^2}{z_1 - z_2},$$

а сувора рівність, звичайно, має місце у межі $N \to \infty$.

Звідсіи відразу ж слідує, що при $z_2 \geq z_1$, тобто при

$$\left\langle k^2 \right\rangle - \left\langle k \right\rangle \geq \left\langle k \right\rangle,$$

у мережі існує нескінченний кластер.

Ця нерівність називається критерієм Молоя-Ріда, її часто записують у такому вигляді:

$$\left\langle k^2 \right\rangle - 2\left\langle k \right\rangle \geq 0.$$

Таким чином, згідно цієї нерівності гігантський кластер існує, якщо число інших сусідів більше ніж середня кількість зв'язків, що виходять з вузла.

Існування гігантського кластера означає також, що вибираючи як завгодно віддалені один від одного вузли, ми

з ненульовою ймовірністю зможемо пройти по мережі від одного вузла до іншого.

Саме тому про ті значення параметрів, при яких $\langle k^2 \rangle$ досягає значення $2\langle k \rangle$, говорять як про поріг протікання. Значення порога протікання p_c при цьому дорівнює:

$$p_c = \frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle}.$$

Питання для самоконтролю

1. Які два найбільш універсальних закони розподілу випадкових величин зазвичай використовуються в природничих дослідженнях?
2. Чому частота вживання нормального закону випадкових величин пояснюється в природничих дослідженнях?
3. Яка центральна гранична теорема теорії ймовірності пов'язана з нормальним розподілом?
4. Як зазвичай називають розподіл, який часто зустрічається із статечною (гіперболічною) щільністю ймовірності? Яка основна характеристика цього розподілу?
5. Як Мандельброт пояснює появу гіперболічного розподілу у лінгвістиці?
6. Яким чином розподіл Коші пов'язаний з моделлю стрільби з обертового?
7. Чому математичне очікування та дисперсія розподілу Коші не визначені для певного діапазону значень?
8. Які змінні названо на честь В. Парето та Дж. Ципфа, і як вони використовуються в дослідженнях?
9. Як відбувається зростання мережі за моделлю Барабаші-Альберт?
10. Що таке переважне приєднання (preferential attachment) у контексті моделі Барабаші-Альберт?
11. Що означає безмасштабна структура мережі та яким чином ця властивість виявляється у розподілі ступенів вузлів?
12. Яка особливість моделі малого світу Ваттса-Строгатца у вимірах та зв'язках між вузлами?

13. Як ви описуєте відстань між кінцями доданих зв'язків у моделі малого світу? Як вона змінюється при переході до великих розмірностей?
14. Що таке (u,v)-"квіти" і "дерева" у контексті мережі зі SF-розподілом ступеня вузлів? Як вони будуються та які особливості мають?
15. Як визначається живучість системи? Які аспекти входять до цього поняття?
16. Як визначається канонічна живучість мережі R(G, p)? Як вона пов'язана з поліномом Татта-Уїтні?
17. Що таке поріг протікання у перколяційних мережах? Як він пов'язаний із здатністю пройти по чорних зв'язках через мережу?
18. Як змінюється значення порогу протікання зі збільшенням розмірності перколяційної мережі?
19. Які схожості і відмінності існують між поняттям "бескінечного кластера" у теорії перколяції та "Giant Cluster" у теорії складних мереж?

3. Пошук в мережах

3.1 Векторно-просторова модель пошуку

Герхард Солтон

(*1927-1995*)

Більшість відомих інформаційно-пошукових систем базується на використанні векторно-просторової моделі опису даних (Vector Space Model), запропонованої Г. Солтоном у 1975 р. та застосованої ним у системі SMART. Ця модель є однією з класичних – алгебраїчною. У рамках цієї моделі документ описується вектором в евклідовому просторі, в якому кожному терму, що використовується в документі, ставиться у відповідність його вагове значення, яке визначається на основі статистичної інформації про його появу як в окремому документі, так і в усьому документальному масиві. Опис запиту, відповідного необхідної користувачеві тематики, також є вектором у тому ж евклидовом просторі термів. Для оцінки близькості запиту та документа використовується скалярний добуток відповідних векторів запиту та документа.

У рамках цієї моделі кожному терму t_i в документі d_j відповідає деяка невід'ємна вага w_{ij}. Запиту q, який є множиною термів, не з'єднаних між собою ніякими логічними операторами, також відповідає вектор вагових значень w_{iq}.

Таким чином, кожен документ та запит можуть бути представлені у вигляді $n-$мірного вектору, де n – загальна кількість термів у словнику моделі. Відповідно до розглянутої моделі, близькість документа d_j до запиту q, які, як і в попередніх моделях, розглядаються як інформаційні вектори $d_j = (w_{1j},\ w_{2j}, ..., w_{nj})$ і оцінюється як їх

скалярний твір. При цьому вагу окремих термів можна обчислювати різними способами. Один із можливих найпростіших підходів – використовувати як вагу терму w_{ij} в документі нормалізовану частоту $freq_{ij}$ його находження в даному документі, тобто:

$$w_{ij} = tf_{ij} = freq_{ij} / \max_{1 \le i \le n}(freq_{ij}).$$

Однак цей підхід не враховує, наскільки часто цей терм використовується у всьому масиві документів, так звану дискримінаційну силу терму. Тому у випадку, коли доступна статистика використання термів у всьому документальному масиві, ефективніше наступне правило обчислення ваги:

$$w_{ij} = tf_{ij} \cdot \log \frac{N}{n_i},$$

де n_i – кількість документів, в яких використовується терм t_j, а N – загальна кількість документів у масиві.

Слід зазначити, що наведена вище формула багаторазово уточнювалася з метою найбільш точної відповідності запитам користувачів, що видаються документальними системами. У 1988 Солтоном був запропонований такий варіант для обчислення ваги терму t_i із запиту:

$$w_{iq} = \left(0.5 + \frac{freq_{iq}}{\max\limits_{1 \le l \le n} freq_{lq}} \right) \cdot \log \frac{N}{n_i},$$

де $freq_{iq}$ – частота терму t_i із запиту в тексті документу, що складається з n термів.

Зазвичай вагові значення w_{ij} унормуються, що дозволяє розглядати документ як ортонормований вектор. Такий метод зважування термів має стандартне позначення

$TF \cdot IDF$, де TF вказує на частоту появи терму в документі (term frequency), а IDF – на величину, зворотну числу документів масиву, що містять даний терм (inverse document frequency).

Коли виникає завдання визначення тематичної близькості двох документів або документа та запиту, у цій моделі використовується простий скалярний добуток $sim(d_1, d_2)$, двох відповідних векторів вагових значень $(w_{i1}, ..., w_{n1})$ і $(w_{i2}, ..., w_{n2})$, який, очевидно, відповідає косинусу кута між векторами – образами документів d_1 і d_2. Очевидно, $sim(d_1, d_2)$ належить діапазону [0, 1]. Чим більша величина тим більш близькі документи d_1 і d_2. Для будь-якого документа d маємо $sim(d,d) = 1$. Аналогічно мірою близькості документа d_j та запиту q є величина:

$$sim(d_j, q) = \frac{d_j \cdot q}{|d_j| \cdot |q|} = \frac{\sum_{i=1}^{n} w_{ij} w_{iq}}{\sqrt{\sum_{i=1}^{n} w_{ij}^2} \sqrt{\sum_{i=1}^{n} w_{iq}^2}}.$$

Векторно-просторова модель представлення даних забезпечує системам, побудованим на її основі, такі можливості:

- обробку запитів без обмежень їхньої довжини;
- простоту реалізації режиму пошуку подібних документів (кожен документ може розглядатися як запит);
- збереження результатів пошуку з можливістю виконання уточнюючого пошуку.

У векторно-просторовій моделі відсутня можливість виконання запитів, які містять логічні операції. Це суттєво обмежує її придатність для деяких завдань. Крім того, ця

модель є основою для розробки інших моделей пошуку, включаючи мережеві, і спрямована на пошук інформації у великих наборах даних, які не мають явно вираженої мережевої структури.

3.2 Моделі пошуку в піринговых мережах

Моделі пошуку у мережному середовищі розглянемо з прикладу пошуку у про піринговых мережах. Пірингові мережі (Peer-to-peer, P2P – рівний з рівним) – це комп'ютерні мережі, що ґрунтуються на рівноправності учасників. У таких мережах відсутні виділені сервери, а кожен вузол (peer) є як клієнтом, так і сервером[19]. Вперше фраза "peer-to-peer" була використана в 1984 Парбауеллом Йохнухуйтсманом (Parbawell Yohnuhuitsman) при розробці архітектури Advanced Peer to Peer Networking фірми IBM.

P2P – це мережевий протокол, що забезпечує можливість створення та функціонування мережі рівноправних вузлів та їх взаємодії. У багатьох випадках P2P є накладеними мережами, що використовують існуючі транспортні протоколи стеку TCP/IP – TCP або UDP. Слід зазначити, що на практиці пірингові мережі складаються з вузлів, кожен з яких взаємодіє лише з деякою підмножиною інших вузлів мережі (через обмеженість ресурсів). Для реалізації протоколу P2P використовуються клієнтські програми, що забезпечують функціональність як окремих вузлів, так і всієї пірінгової мережі.

Процедури пошуку в піринговых мережах враховують облік їхньої різноманітної топології, найчастіше децентралізованої. Сьогодні немає уніфікованих підходів до організації пошукових процедур, тому застосовуються найрізноманітніші методики. Саме завдяки піринговим мережам було розроблено багато методів пошуку в

[19] Masinde, N., Graffi, K. Peer-to-Peer-Based Social Networks: A Comprehensive Survey. SN COMPUT. SCI. 1, 299 (2020). https://doi.org/10.1007/s42979-020-00315-8

мережевому середовищі, докладний опис яких розглядатиметься нижче.

Досить часто пірингові мережі доповнюються виділеними серверами, що несуть організаційні функції, наприклад, авторизацію. Зокрема, відомі бібліотечні пірингові мережі, в яких використовуються виділені сервери, які відіграють роль центрів авторизації, хешування та реплікації бібліографічних даних.

Централізована архітектура «клієнт-сервер» має на увазі, що мережа залежить від центральних вузлів (серверів), що забезпечують підключені до мережі термінали (клієнти) необхідними сервісами. У цій архітектурі ключова роль приділяється серверам, які визначають мережу незалежно від наявності клієнтів. Незважаючи на те, що всі вузли P2P мають однаковий статус, реальні можливості їх можуть суттєво відрізнятися.

Очевидно, що зростання кількості клієнтів мережі типу «клієнт-сервер» призводить до зростання навантажень на серверну частину, внаслідок чого вона може бути перевантаженою.

Децентралізована пірингова мережа, навпаки, стає більш продуктивною зі збільшенням кількості вузлів, підключених до неї. Справді, кожен вузол додає до мережі P2P свої ресурси (дисковий простір і обчислювальні можливості), у результаті сумарні ресурси мережі збільшуються.

Порівняно з клієнт-серверною архітектурою P2P має такі переваги, як самоорганізованість, відмовостійкість при втраті зв'язку з вузлами мережі (висока живучість), можливість поділу ресурсів без прив'язки до конкретних адрес, збільшення швидкості копіювання інформації за рахунок використання відразу декількох джерел, широка смуга пропускання, гнучке балансування навантаження.

Крім названих вище переваг піринговых мереж, їм властивий також ряд недоліків.

Перша група недоліків пов'язана зі складністю управління такими мережами, порівняно з клієнт-серверними системами. Доводиться витрачати значні зусилля на підтримку стабільного рівня їхньої продуктивності, резервне копіювання даних, антивірусний захист, захист від інформаційного шуму та інших зловмисних дій користувачів.

Велика проблема – це легітимність контенту, що передається у таких P2P-мережах. Незадовільне вирішення цієї проблеми призвело до скандального закриття багатьох таких мереж (наприклад, Napster у липні 2001 року). Є й інші проблеми, що мають соціальну природу. Наприклад, у системі Gnutella, 70% користувачів не додають взагалі жодних файлів у мережу. Більше половини ресурсів у мережі надається одним відсотком користувачів, тобто мережа еволюціонує у напрямі клієнт-серверної архітектури.

Ще одна проблема P2P-мереж пов'язана з якістю та достовірністю контенту, що надається. Серйозною проблемою є фальсифікація файлів та поширення фальшивих ресурсів. Захист розподіленої мережі від хакерських атак, ботнетів, вірусів та «троянських коней» є дуже складним завданням. Найчастіше інформація з даними про учасників P2P-мереж зберігається у відкритому вигляді, доступному для перехоплення. Ще однією проблемою є можливість фальшування ID вузлів.

Головне завдання інформаційного пошуку в піринговых мережах – швидке та ефективне знаходження найбільш релевантних відгуків на запит, що передається від вузла до всієї мережі. У цьому, природно, пошук реалізується без участі центрального сервера, тобто децентралізовано. Зокрема, за такої організації пошуку актуальне завдання

отримання якісного результату при загальному зменшенні мережного трафіку.

Розглянемо докладно такі методи (алгоритми) децентралізованого пошуку[20]:

- алгоритм пошуку ресурсів за ключами;
- метод широкого первинного пошуку (Breadth First Search);
- метод випадкового широкого первинного пошуку (Random Breadth First Search);
- Інтелектуальний пошуковий механізм (Intelligent Search Mechanism);
- метод «більшості результатів з минулої евристики» (RES);
- алгоритм «випадкових блукань» (Random Walkers Algorithm).

3.2.1 Алгоритм пошуку ресурсів по ключам

У більшості піринговых мереж, орієнтованих на обмін файлами, використовується два виду сутностей, яким приписуються відповідні ідентифікатори (ID): вузли і ресурси, що характеризуються ключами (Key), тобто мережа може бути представлена двовимірною матрицею розмірності MN, де M – кількість вузлів, N – кількість ресурсів. У цьому разі завдання пошуку зводиться до знаходження ID вузла, у якому зберігається ключ ресурсу. На рисунку 24 представлений процес пошуку ресурсу, що запускається з вузла з ID 0.

[20] *Zeinalipour-Yazti D., Kalogeraki V., Gunopulos D.* Information Retrieval in Peer-to-Peer Networks. IEEE CiSE Magazine, Special Issue on Web Engineering, 2004. – pp. 1-13.

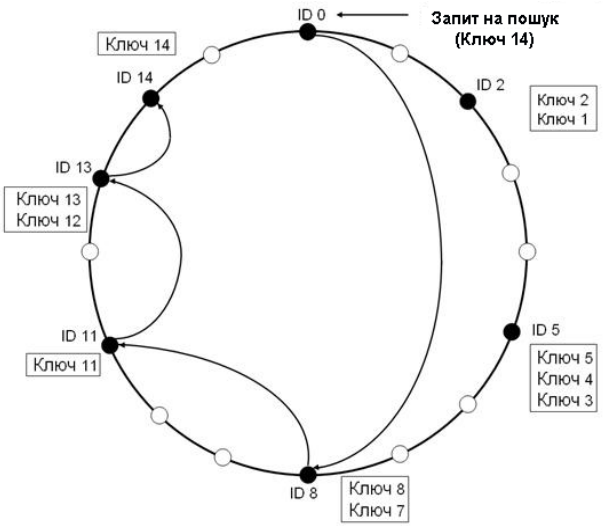


Рисунок 24 – Модель пошуку ресурсу за ключом (у чорних вузлах містяться документи з ключами – у білих – не містяться)

У даному випадку з вузла з ID 0 запускається пошук ресурсу з ключем 14. Запит проходить певний маршрут і досягає вузла, на якому знаходиться ключ 14.

Розглянемо алгоритми пошуку у піринговых мережах, обмежившись основними методами пошуку за ключовими словами.

3.2.2 Метод широкого первинного пошуку

Метод широкого первинного пошуку (Breadth First Search, BFS) широко використовують у реальних файлообмінних мережах P2P, як-от, наприклад, Gnutella (www.gnutella.com). Метод BFS (див. рис. 9) у мережі P2P розмірності реалізується у такий спосіб. Вузол q генерує запит, який адресується всім сусідам (найближчим за деякими критеріями вузлам). Коли вузол p отримує запит, виконується пошук у його локальному індексі. Якщо деякий вузол r приймає запит (Query) і обробляє його, він генерує повідомлення-відгук (QueryHit), щоб повернути

результат. Повідомлення QueryHit включає інформацію про релевантні документи, яка доставляється по мережі вузлу, що запитує (див. рис. 25).

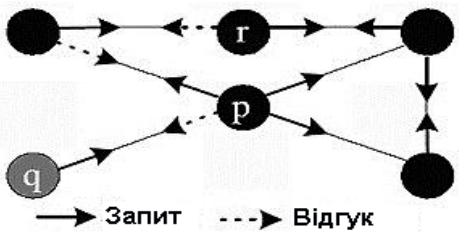


Рисунок 25 – Метод BFS

Коли вузол q отримує QueryHits з більш ніж одного вузла, він може завантажити файл з найбільш доступного ресурсу. Повідомлення QueryHit повертаються тим самим шляхом, що й первинний запит. У BFS кожен запит викликає надмірне навантаження мережі, оскільки він передається по всіх зв'язках (у тому числі й вузлів із високим часом очікування). Тому вузол із низькою пропускною здатністю може стати вузьким місцем. Один метод, що дозволяє уникнути навантаження всієї мережі повідомленнями полягає у приписуванні кожному запиту параметру час життя (Time-to-level, TTL). Параметр TTL визначає максимальну кількість переходів, якими можна пересилати запит. При типовому пошуку початкове значення для TTL становить зазвичай 5-7, яке зменшується щоразу, коли запит надсилається. Коли TTL дорівнює 0, повідомлення більше не передається. BFS гарантує високий рівень якості збігів за рахунок великої кількості повідомлень.

3.2.3 Метод випадкового широкого первинного пошуку

Метод випадкового широкого первинного пошуку (Random Breadth First Search, RBFS) було запропоновано як покращення «наївного» підходу BFS. У методі RBFS (див. рис. 26) вузол q пересилає пошукове розпорядження лише

частини вузлів мережі, обраних у випадковому порядку. Яка саме частина вузлів – це параметр методу RBFS.

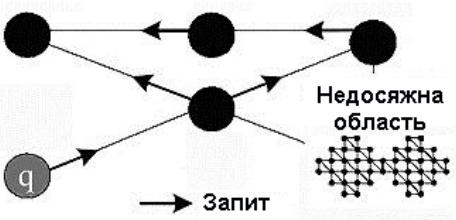


Рисунок 26 – Метод RBFS

Перевага RBFS полягає в тому, що не потрібна глобальна інформація про стан контенту мережі; вузол може отримувати локальні рішення так швидко, як це потрібно. З іншого боку, цей метод імовірнісний. Тому деякі великі сегменти мережі можуть виявитися недосяжними.

3.2.4 Інтелектуальний пошуковий механізм

Інтелектуальний пошуковий механізм (Intelligent Search Mechanism, ISM) – новий метод пошуку мереж P2P (див. рис. 27). Поліпшення швидкості та ефективності пошуку інформації за допомогою даного методу досягається за рахунок мінімізації витрат на зв'язки, тобто на кількість повідомлень, що передаються між вузлами, та мінімізації кількості вузлів, які опитуються для кожного пошукового запиту. Щоб досягти цього, для кожного запиту оцінюються лише ті вузли, які найбільше відповідають даному запиту.

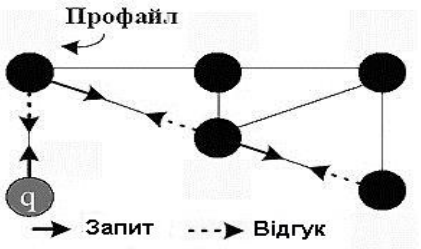


Рисунок 27 – Метод ISM

Інтелектуальний пошуковий механізм складається з двох компонентів – профайлу (profile) та способу його ранжування, так званого рангу релевантності. Кожен вузол мережі будує інформаційний профайл для кожного із сусідніх вузлів. Профайл містить останні відповіді кожного із вузлів. За допомогою рангу релевантності здійснюється ранжування профайлів вузлів для вибору тих сусідніх, які надаватимуть найбільш релевантні документи на запит.

Механізм профайлів використовується для того, щоб зберігати останні запити, а також кількісні характеристики результатів пошуку.

При реалізації моделі ISM використовується єдиний стек запитів розміром $O(TN)$, у якому зберігається по T запитів для N вузлів. Як тільки стек заповнюється, відбувається заміна «останнього найменш використовуваного» (Least Recently Used, LRU) для збереження останніх запитів. Функція «ранг релевантності» (Relevance Rank, RR) використовується вузлом P_l, щоб виконувати оперативну класифікацію його сусідів для визначення тих, які слід опитувати першими за запитом q. Для обчислення рангу релевантності кожного вузла P_i, порівнює q з усіма запитами у структурі профайлу, для якого відомий список відповідей на попередні запити, та обчислюється $RR(P_i, q)$:

$$RR(P_i, q) = \sum_{j \in Q} Sim(q_j, q)^{\alpha} \cdot S(P_i, q_j).$$

У цій формулі Q – множина запитів, на які була відповідь у вузла P_i, $S(P_i, q_j)$ – число результатів, що повертаються вузлом P_i за запитом q_j, а метрика Sim розраховується за правилом косинуса, аналогічно розглянутому у векторно-просторовій моделі пошуку:

$$Qsim(\boldsymbol{q}_j, \boldsymbol{q}) = \frac{\boldsymbol{q}_j \cdot \boldsymbol{q}}{|\boldsymbol{q}_j||\boldsymbol{q}|}.$$

RR забезпечує вищий ранг вузлу, який повертає більше результатів. Крім того, використовується параметр α, який дозволяє збільшувати вагу запитів, найбільш подібних до вихідного.

У разі коли α велике, запити з великою подібністю $Qsim(q_j, q)$ домінують у наведеній вище формулі. Розглянемо ситуацію, коли вузол P_1 відповідає запитам q_1 та q_2 значеннями подібності для запиту q: $Qsim(q_1, q) = 0.5$ і $Qsim(q_2, q) = 0.1$, а вузол P_2 відповідає запитам q_3 та q_4 значеннями $Qsim(q_3, q) = 0.4$ і $Qsim(q_4, q) = 0.3$. Якщо вибрати $\alpha = 10$, то $Qsim(q_1, q)$ домінує, тому що $0.5^{10} + 0.1^{10} > 0.4^{10} + 0.3^{10}$.

Однак для $\alpha = 1$ всі запити важать однаково, і P_2 дає більш високу релевантність. При $\alpha = 0$ враховується лише кількість результатів, повернутих кожним вузлом.

Метод ISM ефективно працює у мережах, де вузли містять деякі спеціалізовані відомості. Зокрема дослідження мережі Gnutella показує, що якість пошуку дуже залежить від «оточення» вузла, з якого задається запит. Ще одна проблема методу ISM полягає в тому, що пошукові повідомлення можуть зациклюватися, тому не в змозі досягти деяких частин мережі. Щоб вирішити цю проблему було запропоновано такий підхід. Вибиралася маленька випадкова підмножина вузлів (в експерименті додатково вибирався один випадковий вузол) і додавалася до набора релевантних вузлів для кожного запиту. В результаті механізм ISM став охоплювати більшу частину мережі.

2.3.5 Методи «більшості результатів з минулої евристики»

У рамках методу «більшості результатів з минулої евристики» (>RES) (див. рис. 28) кожен вузол пересилав запит підмножині своїх вузлів, утвореної на підставі деякої узагальненої статистики.

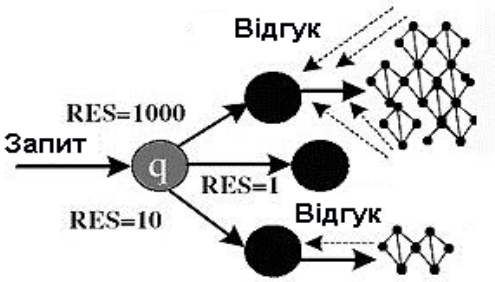


Рисунок 28 – Метод >RES

Запит у методі >RES є задовільним, якщо видається Z або більше результатів (Z – деяка константа). У методі >RES вузол q пересилає запити до k вузлів, які видали найбільші результати останніх m запитів. В проведених експериментах k змінювалося від 1 до 10 і таким шляхом метод >RES варіювався від BFS до підходу глибинного первинного пошуку (Depth-first-search). Метод >RES подібний до методу ISM, який розглядався, але використовує більш просту інформацію про вузли. Його головний недолік у порівнянні з ISM – відсутність аналізу параметрів вузлів, зміст яких пов'язаний із запитом. Тому метод >RES характеризується швидше як кількісний, а не якісний підхід. З досвіду відомо, що >RES хороший тим, що він маршрутизує запити у великі сегменти мережі (які можливо, також містять більш релевантні відповіді). Він також захоплює сусідів, які менш перевантажені, починаючи з тих, котрі зазвичай повертають більше результатів.

2.3.6 Метод «випадкових блукань»

Ключова ідея алгоритму «випадкових блукань» (Random Walkers Algorithm, RWA) полягає в тому, що кожен вузол випадково пересилає повідомлення із запитом, що називається «посиланням» одному зі своїх вузлів. Щоб скоротити час, необхідний для отримання результатів, ідея однієї посилки розширена до k посилок, де k – число незалежних посилок, послідовно запущених з пошукового вузлу. Очікується, що «k посилок» після T кроків досягне тих самих результатів, що й одна посилка за kT кроків. Цей алгоритм нагадує метод RBFS, але в RBFS кожен вузол пересилає повідомлення запиту частині з його сусідів. До того ж, у RBFS передбачається експоненційне збільшення повідомлень, що пересилаються, а в методі випадкових блукань – лінійне. Обидва методи – і RBFS, і RWA не використовують жодних явних правил для того, щоб адресувати пошуковий запит до найбільш релевантного змісту.

Ще однією методикою, подібною до RWA, є «адаптивний ймовірнісний пошук» (Adaptive Probabilistic Search, APS). В APS кожен вузол розгортає на своїх ресурсах локальний індекс, що містить значення умовних ймовірностей для кожного сусіда, який може бути обраний для наступного переходу майбутнього запиту. Головна відмінність від RWA у разі – те, що в APS вузол використовує зворотний зв'язок від попередніх пошуків (як умовних ймовірностей) замість цілком випадкових переходів. Тому метод APS часто дає кращі результати, ніж RWA.

Інший алгоритм, розроблений у Каліфорнійському університеті, побудований на методі випадкових блукань, використовує принцип порога перколяції зв'язків, тобто порогу протікання або просочування зв'язків між тісно пов'язаними вузлами мережі. На етапі перколяції зв'язків запит потрапляє на один із базових серверів, які з'єднані один з одним потужними каналами зв'язку. Виявилося, що

повноцінний процес пошуку можна проводити локально, тобто при опитуванні тільки сусідніх серверів. За такого підходу кожен запит генерує відносно невеликий трафік.

Існує багато областей, де успішно застосовується P2P-технологія, наприклад, паралельне програмування, кешування даних, резервне копіювання даних.

Завдяки таким характеристикам, як живучість, стійкість до відмов, здатність до саморозвитку, пірингові мережі знаходять все більше застосування в системах управління виробництвами та організаціями (наприклад, P2P-технологія сьогодні застосовується в Державному Департаменті США). У разі можливого виходу із ладу частини вузлів чи серверів, це істотно не впливає на керованість всієї системи. Система доменних імен (DNS) у мережі Інтернет також фактично є мережею обміну даними, побудованою за принципом P2P.

Однією з реалізацій технології P2P є також популярна нині система розподілених обчислень GRID. Ще одним прикладом розподілених обчислень може бути проєкт distributed.net, учасники якого займаються легальним зломом криптографічних шифрів, щоб перевірити їхню надійність.

3.3 Рангові характеристики

Ранжування – процес, у якому пошукова система вибудовує результати пошуку упорядковані за принципом максимальної відповідності конкретному запиту. Таким чином, представлення результатів пошуку залежить від алгоритму ранжирування, який використовується у пошуковій системі.

У результаті пошуку, користувач може отримати великий список релевантних документів. Сортування цього списку таким чином, щоб найбільш важливі для користувача документи були на його початку, в технологіях

інформаційного пошуку прийнято називати ранжуванням відгуків інформаційно-пошукових систем.

Ранжування результатів пошуку за рівнем релевантності можливе не для всіх моделей пошуку (наприклад, неможливе для булевої моделі).

Перспективний підхід до ранжування – використання багатопрофільних шкал, сформованих на основі метаданих, мережевих властивостей, даних про користувачів.

Наприклад, реалізація сюжетних ланцюжків у тематичних інформаційних масивах та його зважування розглядаються як один з алгоритмів ранжування. Ранжування текстових та гіпертекстових документів має суттєві відмінності. Ранжування текстових документів може здійснюватися за рівнем релевантності та іншими параметрами, зокрема екстрагованими з текстів.

Ранжування гіпертекстових документів можливе також за властивостями, що обумовлюються мережевою структурою, гіперпосиланнями.

В Інтернеті для визначення авторитетності вебсторінки як джерела інформації або посередника використовується аналіз топології мережі, утвореної документами та відповідними гіперпосиланнями. Два алгоритми ранжування вебсторінок, заснованих на зв'язках, HITS (hyperlink induced topic search) та PageRank, були розроблені у 1996 році в IBM Дж. Клейнбергом[21] і у Стенфордському Університеті С. Брином і Л. Пейджем[22].

Обидва алгоритми призначені для вирішення "проблеми надмірності", яка властива широким запитам, збільшення

---

[21] *Kleinberg J.M.* Authoritative sources in a hyperlink environment. In Processing of ACM-SIAM Symposium on Discrete Algorithms, 1998, 46(5): 604-632.

[22] *Brin S., Page L.* The Anatomy of a Large-Scale Hypertextual Web Search Engine. WWW7, 1998.

точності результатів пошуку на основі методів аналізу складних мереж.

3.3.1 Алгоритм HITS

Дж. Клейнберг

Алгоритм HITS (Hyperlink Induced Topic Search), запропонований Дж. Клейнбергом, забезпечує вибір з інформаційного масиву найкращих «авторів» (першоджерел, на які ведуть посилання) та «посередників» (документів, від яких йдуть посилання цитування). Зрозуміло, що сторінка є хорошим посередником, якщо вона містить посилання на цінні першоджерела, і навпаки, сторінка є хорошим автором, якщо вона згадується хорошими посередниками.

Для кожного документа d_j ($j = 1,...,N$) рекурсивно обчислюється його значущість як автора $a(d_j)$ та посередника $h(d_j)$ за формулами:

$$a(d_j) = \sum_{i \to j}^{N} h(d_i), \quad h(d_j) = \sum_{j \to i}^{N} a(d_i).$$

Якщо ввести поняття матриці інцидентів $\boldsymbol{A}$, елемент якої a_{ij} дорівнює одиниці, коли документ d_i містить посилання на документ d_j, і нулю в іншому випадку, то алгоритм HITS забезпечує вибір найбільш авторитетних документів (авторів або посередників). Ці документи відповідають власним векторам матриць $\boldsymbol{A}\boldsymbol{A}^T$ та $\boldsymbol{A}^T\boldsymbol{A}$ з найбільшими модулями власних значень (тут $\boldsymbol{A}^T$ – транспонована матриця $\boldsymbol{A}$).

Алгоритм обчислення рангів HITS призводить до зростання рангів документів зі збільшенням кількості та ступеня пов'язаності документів відповідної спільноти. У

цьому випадку у результати видачі інформаційно-пошукової системи, що використовує алгоритм HITS, можуть потрапити у великій кількості документи за темами, відмінними від інформаційної потреби користувача, тобто частина результатів, що видаються, може відхилитися від домінуючої тематики, відбувається, так званий, зсув тематики (topic drift).

Для вирішення цієї проблеми в якості альтернативи стандартному алгоритму HITS було запропоновано алгоритм PHITS. У межах цього алгоритму передбачається: D – множина цитуючих документів, C – множина посилань, Z – множина класів (близьких за якимось критерієм документів), куди поділяються документи. Передбачається також, що подія $d \in D$ (те, що обраний для розгляду навмання документ є документом d) відбувається з ймовірністю $P(d)$.

Умовні ймовірності $P(c \mid z)$ (імовірність того, що посилання з класу z – це c) і $P(z \mid d)$ (ймовірність того, що обраний документ d відноситься до класу z) використовуються для опису залежностей між наявністю посилання $c \in C$, фактором $z \in Z$ та документом $d \in D$.

Оцінюється функція правдоподібності:

$$L(D,C) = \prod_{c \in C, d \in D} P(d,c) =$$

$$= \prod_{c \in C, d \in D} P(d)P(c \mid d),$$

де

$$P(c \mid d) = \sum_{z \in Z} P(c \mid z)P(z \mid d).$$

Мета алгоритму PHITS полягає у тому, щоб підібрати $P(z)$, $P(c \mid z)$, $P(d \mid z)$, щоб максимізувати $L(D,C)$.

де:

$P(c|z)$ – ранги авторів;

$P(d|z)$ – ранги посередників.

Для обчислення рангів необхідно задати кількість чинників у Z, і тоді $P(c|z)$ характеризуватиме якість сторінки як автора в контексті тематики z. До недоліків методу слід віднести те, що ітеративний процес найчастіше зупиняється не на абсолютному, а на локальному максимумі функції правдоподібності L (функція правдоподібності показує, наскільки правдоподібна наявність посилань при вибраних документах). Разом з тим у ситуаціях, коли в множині знайдених вебсторінок немає явного домінування тематики запиту, PHITS перевершує алгоритм HITS.

3.3.2 Алгоритм PageRank

Алгоритм PageRank був винайдений засновниками компанії Google для ранжування веб-сторінок. Він отримав назву на прізвище одного з його винахідників Ларі Пейджа (Larry Page).

Головну ідею цього алгоритму можна описати такими словами: значимість (ранг) сторінки тим вища, що більше посилань її у інших значимих сторінок. Тобто. PageRank розраховує ймовірність того, що людина, яка випадково переходить за посиланнями, дістанеться до деякої сторінки. Чим більше посилань веде на цю сторінку з інших популярних сторінок, тим вища ймовірність, що експериментатор суто випадково натрапить на неї. У нашій термінології сторінка це вузол мережі, а лінк на сторінку – це спрямований зв'язок.

Алгоритм PageRank близький за ідеологією до літературного індексу цитування, який розраховується для довільного документа з урахуванням кількості посилань від інших документів на цей документ, але при цьому у

PageRank як і в HITS, на відміну від літературного індексу цитування, не всі посилання вважаються рівнозначними.

Принцип розрахунку рангу вебсторінки PageRank наступний: розглядається модель – процес, у якому деякий користувач Інтернет відкриває випадкову вебсторінку, з якою переходить до випадково вибраного гіперпосилання. Потім він переміщається на іншу вебсторінку і знову активізує випадкове гіперпосилання тощо, постійно переходячи від сторінки до сторінки, ніколи не повертаючись. Іноді йому таке блукання набридає, і він знову переходить на випадкову вебсторінку не за посиланням, а набравши вручну деяку URL-адресу. У цьому випадку ймовірність того, що користувач, що блукає в Мережі, перейде на деяку певну вебсторінку – це її ранг. Очевидно, PageRank вебсторінки тим вище, чим більше інших сторінок посилається на неї, і чим ці сторінки популярніші.

Автори PageRank

Ларі Пейдж

Сергій Брин

Нехай є n сторінок $D=\{d_1, ..., d_n\}$, , які посилаються на даний документ (вебсторінку A), а $C(A)$ – загальна кількість посилань з вебсторінки A на інші документи. Відповідно до наведеної моделі поведінки користувача визначається деяке фіксоване значення (damping factor) як ймовірність того, що користувач, переглядаючи якусь вебсторінку з множини D, перейде на сторінку A за посиланням, а не набираючи її URL у явному вигляді.

В рамках моделі ймовірність продовження цим користувачем вебсерфінгу по мережі з вебсторінок без

використання гіперпосилань шляхом ручного введення адреси (URL) зі випадкової сторінки становитиме $1-\delta$ (альтернатива переходу за гіперпосиланнями). Індекс PageRank $PR(A)$ для сторінки A розглядається як ймовірність того, що користувач опиниться в деякий випадковий час на цій сторінці:

$$PR(A) = (1-\delta)/N + \delta\sum_{i=1}^{n}\frac{PR(d_i)}{C(d_i)}.$$

За цією формулою індекс сторінки легко підраховується простим алгоритмом ітерації. Як правило застосовується до 30 кроків ітерації задля досягнення стійких результатів.

Незважаючи на відмінності HITS і PageRank, у цих алгоритмах загальне те, що авторитетність (вага) вузла залежить від ваги інших вузлів, а рівень "посередника" залежить від того, наскільки авторитетними є вузли, на які він посилається.

Розрахунок авторитетності окремих документів на цей час широко застосовується при визначенні порядку сканування документів у мережі роботами пошукових систем, ранжуванні результатів пошуку, формуванні тематичних оглядів тощо.

Проілюструємо вищесказане на прикладі. Розглянемо невелику частину мережі, що складається з вузлів A з рангом $r_A = 0.5$, B з рангом $r_B = 0.3$ та визначимо ранг вузла C (див. рис. 29).

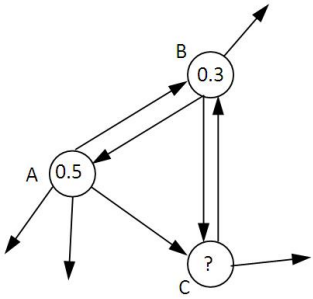


Рисунок 29 – Фрагмент мережі

Кожен із вузлів мережі A і B посилається на вузол C і їм ранг вже відомий. Вузол A при цьому посилається ще на три вузли, а вузол B ще на два вузли. Щоб обчислити ранг вузла C, ранги кожного вузла, що посилається на C, діляться на загальну кількість посилань для цього вузла, після чого отримані значення складаються.

$$r_C = r_A \cdot \frac{1}{4} + r_B \cdot \frac{1}{3} = 0.225.$$

У цьому прикладі всім вузлам, що посилаються на C, вже обчислений ранг. Але неможливо обчислити ранг вузла, поки невідомі ранги вузлів, що посилаються на нього, а ці ранги можна обчислити, тільки знаючи ранги вузлів, які посилаються на них. То як визначити значення рангу для множини вузлів, ранги яких ще невідомі? Рішення полягає в тому, щоб надати всім вузлам довільний початковий ранг і провести кілька ітерацій.

Загалом можна записати наступну формулу для $(n+1)$-ого кроку ітерації:

$$r_i^{(n+1)} = \sum_{j \in E(i)} \left(r_j^{(n)} \cdot \frac{1}{kout_j} \right),$$

де підсумовування по $j \in E(i)$ означає підсумовування по всіх вузлах, які мають посилання на i-й вузол, а $kout_j$ - число вихідних зв'язків для вузла j. У матричному вигляді ітераційне рівняння записується так:

$$\mathbf{r}^{(n+1)T} = \mathbf{r}^{(n)T} \cdot \mathbf{H},$$

де $H_{ij} = A_{ij} \Big/ \sum_j A_{ij}$ – нормалізована матриця інцидентності, а $\mathbf{A}$ – матриця інцидентності мережі.

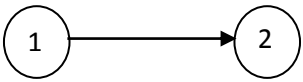
Для ітераційного процесу виникає низка питань, а саме:

– Чи збігається цей процес?

– Які властивості повинна мати матриця $\mathbf{H}$, щоб процес гарантовано сходився?

– Чи залежить кінцевий вектор $\boldsymbol{r}^{(\infty)}$ від початкових умов?

Розглянемо два простих приклади. Перший:

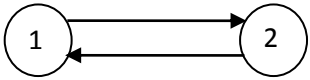


Після виконання ітерацій отримуємо:

Ітерація Ранг вузла	0	1	2	3
1	1	0	0	0
2	0	1	0	0

Результат явно відповідає очікуванням.

Розглянемо ще один, схожий приклад.



У даному випадку отримуємо наступне:

Ітерація Ранг вузла	0	1	2	3
1	1	0	1	0
2	0	1	0	1

З наведеного видно, що збіжності процесу немає.

Можна навести ще багато контрприкладів. З іншого боку, можна побачити, що наведене нагадує “power method” для обчислення власного вектора матриці $\mathbf{H}$ (відповідного власному значенню – одиниця) застосованого до Марківських ланцюгів з матрицею ймовірностей переходів $\mathbf{P} = \mathbf{H}$. У разі йдеться про «лівий» власний вектор. А оскільки теорія Марківських ланцюгів дуже добре вивчена, то можна відразу відповісти на запитання, яким властивостям повинна задовольняти матриця ймовірностей переходів $\mathbf{P}$, щоб процес сходився, не залежав від початкових умов і т.д.

Матриця ймовірностей переходів має бути стохастичною, ненаведеною та неперіодичною. Перша умова задовольняється шляхом переходу до матриці $\mathbf{S}$:

$$\mathbf{S} = \mathbf{H} + \mathbf{a} \cdot \left(\frac{1}{n} \cdot \mathbf{e}^T \right),$$

де $a_i = 1$ якщо з i-ого вузла не виходить жоден зв'язок (так званий "dangling node") і дорівнює нулю в іншому випадку, $\mathbf{e}$ – вектор складається n одиниць, n – число вузлів в мережі. І щоб задовольнити двом умовам, що залишилися, запишемо матрицю $\mathbf{G}$:

$$\mathbf{G} = \alpha \cdot \mathbf{S} + (1-\alpha) \cdot \frac{1}{n} \cdot \left(\mathbf{e} \cdot \mathbf{e}^T \right) =$$

$$= \alpha \cdot \mathbf{H} + \left(\alpha \cdot \mathbf{a} + (1-\alpha) \cdot \mathbf{e} \right) \cdot \frac{1}{n} \cdot \mathbf{e}^T,$$

де α – так званий коефіцієнт загасання. Він означає, що користувач продовжить переходити за посиланнями на

поточній сторінці з ймовірністю α, яке знаходиться в інтервалі від нуля до одиниці. Зазвичай приймають $\alpha = 0.85$. Отже, наше завдання зводиться до обчислення власного лівого вектору матриці $\mathbf{G}$.

Розглянемо перший контрприклад. Для нього:

$$\mathbf{H} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad \mathbf{a} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \mathbf{e} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Звідси отримуємо матрицю $\mathbf{G}$ (для визначеності покладемо $\alpha = 0.85$).

$$\mathbf{G} = \begin{pmatrix} 0.075 & 0.925 \\ 0.5 & 0.5 \end{pmatrix}.$$

Знаходимо ліві власні значення матриці $\mathbf{G}$.

$$\mathbf{r}^T = \begin{pmatrix} 0.351 & 0.649 \end{pmatrix}.$$

Видно, що другий вузол значніший, ніж перший, що цілком відповідає інтуїтивному уявленню, в наслідок того, що перший вузол посилається на другий. Абсолютно аналогічно для другого контрприкладу отримуємо наступні ранги:

$$\mathbf{r}^T = \begin{pmatrix} 05 & 0.5 \end{pmatrix}.$$

Отже, ми отримали рівнозначні значення для рангів.

Розглянемо складніший приклад мережі, зображений на рисунку 30.

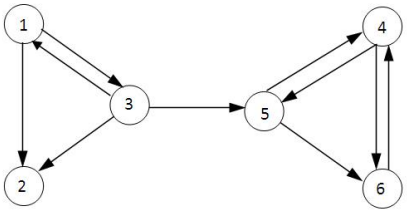


Рисунок 30 – Приклад мережі

Запишемо матрицю $\mathbf{H}$ та вектор $\mathbf{a}$ для цієї мережі.

$$\mathbf{H} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

$$\mathbf{a}^T = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Знайдемо матрицю $\mathbf{G}$, поклавши $\alpha = 0.85$:

$$\mathbf{G} = \begin{pmatrix} 0.025 & 0.45 & 0.45 & 0.025 & 0.025 & 0.025 \\ 0.167 & 0.167 & 0.167 & 0.167 & 0.167 & 0.167 \\ 0.308 & 0.308 & 0.025 & 0.025 & 0.308 & 0.025 \\ 0.025 & 0.025 & 0.025 & 0.025 & 0.45 & 0.45 \\ 0.025 & 0.025 & 0.025 & 0.45 & 0.025 & 0.45 \\ 0.025 & 0.025 & 0.025 & 0.875 & 0.025 & 0.025 \end{pmatrix}.$$

Знаходимо лівий власний вектор (для власного значення – одиниця):

$$\mathbf{r}^T = \begin{pmatrix} 0.052 & 0.074 & 0.057 & 0.349 & 0.2 & 0.269 \end{pmatrix}.$$

Для наочності перенумеруємо вузли відповідно до їхніх рангів – перший номер надамо вузлу з найбільшим рангом (див. рис. 31).

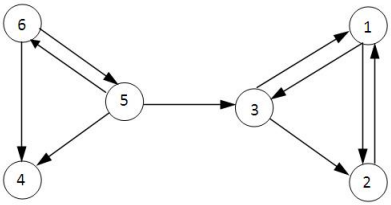


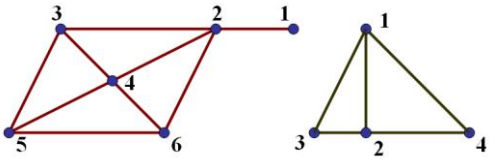
Рисунок 31 – Мережа з ранжованими вузлами

Можна відзначити, що без обчислень (інтуїтивно) навіть таку нескладну мережу практично неможливо ранжувати.

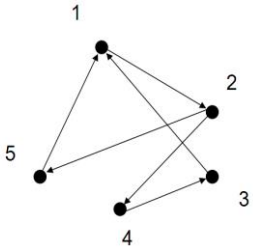
Питання для самоконтролю

1. Яким чином документ описується у векторно-просторовій моделі?
2. Як виглядає опис запиту у векторно-просторовій моделі?
3. Яким чином визначається близькість між документом та запитом у векторно-просторовій моделі?
4. Які існують підходи до обчислення ваги терму в документі, враховуючи статистику використання термів у всьому документальному масиві?
5. Які можливості забезпечує векторно-просторова модель для систем інформаційного пошуку, побудованих на її основі?
6. Яка основна концепція піринговых мереж (P2P)? Які переваги має децентралізована архітектура P2P порівняно з клієнт-серверною архітектурою?
7. Які методи децентралізованого пошуку ресурсів в піринговых мережах ви знаєте?
8. Як відбувається процес пошуку ресурсу в піринговых мережах за допомогою методу BFS?
9. Чим метод RBFS відрізняється від підходу BFS до пошуку в піринговых мережах?
10. Які переваги та недоліки методу інтелектуального пошукового механізму (ISM) в піринговых мережах?
11. Які компоненти входять до складу інтелектуального пошукового механізму (ISM) і як вони працюють разом?
12. Як ранг релевантності (RR) обчислюється для вузлів в інтелектуальному пошуковому механізмі (ISM)?
13. Яка ключова ідея алгоритму "випадкових блукань" (Random Walkers Algorithm, RWA)?
14. Яким чином алгоритм "випадкових блукань" відрізняється від методу RBFS? Які параметри впливають на ефективність алгоритму "випадкових блукань"?
15. Які обмеження та переваги має метод "адаптивний ймовірнісний пошук" (Adaptive Probabilistic Search, APS) порівняно з методом "випадкових блукань"?
16. Що означає термін "ранжування" в контексті пошукових систем? Яким чином ранжування впливає на спосіб відображення результатів пошуку для користувача?

17. Які відмінності між ранжуванням текстових документів та гіпертекстових документів? Які особливості ранжування гіпертекстових документів на основі властивостей мережевої структури та гіперпосилань?
18. Яким чином алгоритм HITS (Hyperlink Induced Topic Search) визначає значущість документів як авторів та посередників?
19. Що таке алгоритм PageRank і як він був винайдений? Яким чином PageRank визначає значимість (ранг) вебсторінок?
20. Як пояснити ідею алгоритму PageRank в термінах ймовірності випадкового переходу за посиланнями? Яким чином використовується ітераційний підхід для обчислення рангів вебсторінок за алгоритмом PageRank?
24. Знайти коефіцієнт кластеризації для вузлів зображених графів:



25. Написати систему рівнянь для рекурсивного підрахунку коефіцієнтів авторства та посередництва в алгоритмі HITS, а також провести розрахунок (4 ітерації) для мережі:



4. Семантичні мережі

Можливість уявлення знань у вигляді семантичних мереж зробила їх цінним інструментом для обробки природної мови та розуміння змісту текстів. Семантичні мережі являють собою структури, призначені для представлення семантичних зв'язків між поняттями або елементами у вигляді графу. Семантична мережа – це спосіб уявлення бази знань, де концепції зв'язуються між собою у формі мережі. Семантична мережа є інструментом, який використовується за наявності знань, які найкраще розуміти як набір пов'язаних один з одним концептів.

Структурою семантичної мережі є граф, де вершини представляють концепти, а ребра відображають семантичні зв'язки між ними [Sowa, 1987], утворюючи семантичні поля. Semantic Web може бути реалізована, наприклад, як графова база даних чи карта концептів. Типові семантичні мережі часто видаються у вигляді семантичних трійок.

Семантичні мережі застосовуються у додатках обробки природної мови, таких як семантичний розбір [Domingos, 2009] та дозвіл багатозначності слів [Sussna, 1993]. Також семантичні мережі можуть використовуватися для аналізу великих текстів з метою виявлення основних тем (наприклад, постів у соціальних медіа) або для виявлення упередженості (наприклад, в охопленні новин), а також для картографування, побудови моделі всієї предметної області [Segev, 2022].

Перші комп'ютерні семантичні мережі були детально розроблені Річардом Річенсом [Lehmann, 1992] в 1956 в рамках проекту Кембриджського центру вивчення мови з машинного перекладу. Процес машинного перекладу підрозділяється на 2 частини: переклад вихідного тексту в проміжну форму подання, та ця проміжна форма

транслюється потрібну мову. Такою проміжною формою якраз і були семантичні мережі.

Більшість семантичних мереж засновані на теорії пізнання та складаються з дуг та вузлів, які можуть бути організовані в таксономічну ієрархію. Ці мережі зробили внесок в ідеї поширення активації, успадкування та подання вузлів як прото-об'єктів.

Процес побудови семантичних мереж включає виявлення ключових слів у тексті, підрахунок частоти їхньої взаємної появи та аналіз мереж для виявлення центральних слів та кластерів тем у мережі [Segev, 2022].

Математика відома своєю здатністю описувати більшість явищ у навколишньому світі за допомогою логічних висловлювань. У свою чергу, семантичні мережі виникли як спроба візуалізації математичних формул, що становлять відносини між об'єктами. Основним уявленням для семантичної мережі є граф. Однак необхідно пам'ятати, що за графічним зображенням завжди стоїть суворий математичний запис.

Основною формою представлення семантичної мережі є граф, так як це форма, що найбільш зручно сприймається людиною. Коли на схемах семантичних мереж зазначені напрями відносин, їх називають картами знань, а сукупність таких карт, що дозволяє охопити великі ділянки семантичної мережі, називають атласом знання чи семантичної картою.

У математиці граф представляється як множина вершин та множина відносин (зв'язків) між ними. З погляду математичної логіки кожна вершина відповідає елементу предметної множини, а дуга – предикату.

У лінгвістиці відносини фіксуються у словниках та тезаурусах. У словниках у визначеннях через рід та видову відмінність родове поняття займає певне місце. У тезаурусах у статті кожного терміна можуть бути зазначені

всі можливі його зв'язки з іншими спорідненими на тему термінами.

У семантичних мережах можна виділити два важливі аспекти: поділ по арності та за кількістю типів відносин.

Щодо поділу по арності, то типовими є мережі з бінарними відносинами (що пов'язують рівно два поняття). На практиці, однак, можуть знадобитися відносини, які пов'язують більше двох об'єктів — *N*-арні.

За кількістю типів відносин мережі можуть бути однорідними або неоднорідними. Однорідні мережі містять лише один тип відносин, у неоднорідних мережах кількість типів відносин перевищує один. Такі мережі мають великий практичний інтерес, але й представляють складніші дослідницькі завдання. Неоднорідні мережі можуть бути представлені як переплетення деревоподібних багатошарових структур.

По арності мережі можуть бути бінарними або N-арними. Бінарні відносини пов'язують рівно два поняття і видаються на графі зручним чином у вигляді стрілки між двома концептами (саме такі мережі будуть досліджуватися у цій роботі). Але іноді виникає потреба у відносинах, які пов'язують більше двох об'єктів, таких відносин називають N-арними. Їхнє уявлення на графі може бути складнішим, і для цього можуть використовуватися концептуальні графи, де кожне відношення представляється окремим вузлом.

Семантичні мережі також можна класифікувати за розміром: галузеві мережі, які є базою для конкретних систем штучного інтелекту, та глобальні семантичні мережі, які прагнуть охопити всі взаємозв'язки у світі, що може стати можливим у майбутньому з розвитком технологій.

Починаючи з 1980-х років у зв'язку з розвитком веб-технологій, семантичні мережі та концепція Semantic Web

стали одним із ключових напрямків для більш ефективної організації та подання інформації в онлайн-середовищі. Semantic Web прагне надати більш структуровану та семантично багату інформацію, яка дозволить комп'ютерам та програмам глибше розуміти вміст та контекст вебресурсів.

Semantic Web – це концепція, запропонована Тімом Бернерсом-Лі, винахідником Всесвітнього павутиння та директором Консорціуму Всесвітньої павутини (W3C) (див. рис. 32). Він представив ідею про те, що вебсторінки мають бути не тільки для читання людьми, а й для розуміння та обробки комп'ютерами. Метою Семантичного Інтернету є створення структурованої інформації, яка може бути оброблена комп'ютерами, що дозволить автоматизувати пошук інформації, інтегрувати дані з різних джерел та створювати інтелектуальніші програми.

У мріях творця WWW Бернерса-Лі, Semantic Web повинен був вирішувати складні завдання, базуючись на знаннях, закладених в інформаційній надбудові над звичайним веб і системою інтелектуальних агентів, які обробляють знання, розподілені в мережі. Він писав: «Семантична Мережа надасть структуру значущому вмісту вебсторінок, створюючи середовище, де програмні агенти, які переміщаються зі сторінки на сторінку, зможуть легко виконувати складні завдання для користувачів. Такий агент, який переходить на вебсторінку клініки, буде знати не лише те, що сторінка містить ключові слова, такі як "лікування, медицина, фізична, терапія" (як це може бути закодовано сьогодні), а також те, що доктор Гартман працює у цій клініці по понеділках, середах та п'ятницях, і що сценарій приймає діапазон дат у форматі рррр-мм-дд та повертає часи призначень. І він буде "знати" всю цю інформацію, не потребуючи штучного інтелекту на масштабі "2001 року" або "Зоряних воєн". Замість цього ця семантика була закодована на вебсторінку, коли менеджер офісу клініки (який ніколи не брав участі у вступі до

Комп'ютерних наук, пристосував її за допомогою готового програмного забезпечення для написання сторінок Семантичної Мережі разом з ресурсами, переліченими на сайті Асоціації фізичної терапії.»[23]

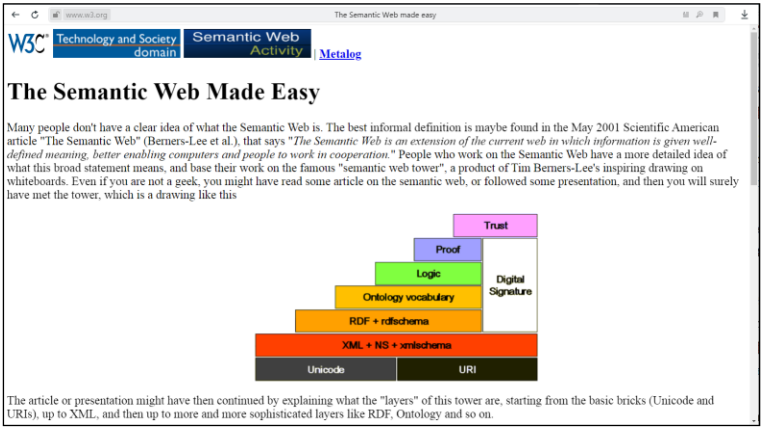


Рисунок 32. Фрагмент вебсторінки сайту W3C

У сучасному Інтернеті, коли застосовується концепція семантичної мережі, відбувається розширення мережі гіперпосилальних вебсторінок, зрозумілих людині, шляхом додавання машинно-читаних метаданих про сторінки та зв'язки між ними. Це дозволяє автоматизованим агентам більш інтелектуально взаємодіяти з Інтернетом та виконувати більш складні завдання від імені користувачів. Тімом Бернерсом-Лі, який відповідальний за розробку стандартів Semantic Web, визначено Semantic Web як "мережу даних, яку можуть обробляти машини безпосередньо і опосередковано".

Багато технологій, запропонованих W3C, вже існували до того, як вони були включені до стандартів W3C. Їх застосування різноманітне, особливо у галузях, де

[23] Berners-Lee, Tim; Hendler, James; Lassila, Ora (May 17, 2001). "The Semantic Web". Scientific American. Vol. 284, N. 5. – pp. 34–43.

інформація охоплює обмежену та певну сферу, і обмін даними є необхідністю, наприклад, у наукових дослідженнях чи обміні даними між компаніями. Крім того, з'явилися інші технології з аналогічними цілями, такі як мікроформати.

Одним із важливих досягнень Тіма Бернерса-Лі стало створення стандарту RDF (Resource Description Framework), який використовується для опису ресурсів та їх властивостей у семантичному Інтернеті.

Термін "Semantic Web" часто використовується більш конкретно для позначення форматів та технологій, які роблять її можливою. Збір, структурування та вилучення пов'язаних даних забезпечуються технологіями, які надають формальний опис концепцій, термінів та відносин у певній галузі знань. Ці технології визначені в стандартах W3C і включають:

- Resource Description Framework (RDF), загальний метод опису інформації;
- RDF Schema (RDFS) – Схема RDF;
- Simple Knowledge Organization System (SKOS) – Проста система організації знань;
- SPARQL, an RDF query language (мова запитів RDF);
- Notation3 (N3), розроблено з урахуванням зручності для читання людиною;
- N-Triples, формат для зберігання та передачі даних;
- Turtle (RDF Triple Language) – лаконічна мова для трійок RDF;
- Web Ontology Language (OWL), a family of knowledge representation languages – сукупність мов представлення знань;

- Rule Interchange Format (RIF), структура діалектів мови вебправил, що підтримує обмін правилами в Інтернеті;
- JavaScript Object Notation for Linked Data (JSON-LD), метод на основі JSON для опису даних;
- ActivityPub, загальний спосіб для клієнта та сервера спілкуватися один з одним.

Стек Semantic Web є структурою або архітектурою Semantic Web. Функції та взаємозв'язки його компонентів можна коротко описати так:

1. XML (Extensible Markup Language) – забезпечує синтаксис для структурування вмісту в документах, але не надає семантики для значення вмісту.
2. XML Schema – мова визначення структури та змісту елементів у XML-документах.
3. RDF (Resource Description Framework) – проста мова для вираження моделей даних та зв'язків між об'єктами.
4. RDF Schema (RDFS) – розширює RDF та надає словник для опису властивостей та класів ресурсів на основі RDF.
5. OWL (Web Ontology Language) – додає розширені словникові терміни для опису властивостей і класів з більш складними відносинами та характеристиками.
6. SPARQL (SPARQL Protocol and RDF Query Language) – протокол та мова запитів для доступу до даних Семантичної мережі.
7. RIF (Rule Interchange Format) – мова на основі XML для вираження правил вебу, які можуть виконувати комп'ютери.

Ці компоненти разом утворюють стек Semantic Web, що дозволяє організувати та обробляти дані семантично, збагачуючи інформацію та роблячи її більш зрозумілою та корисною для машинної обробки.

На даний момент немає широкодоступних засобів перегляду та безпосереднього використання інформації, що надається у Semantic Web. Незважаючи на рідкісні зразки та розрізнені спроби, програми-клієнти не перевищують рівня локальних дослідницьких проектів окремих ентузіастів. Це виклик для активного розвитку Semantic Web.

Коментатори вказують на різні причини, які перешкоджають прогресу Семантичного павутиння, починаючи від людського фактора (небажання людей займатися підтримкою документів з метаданими, проблеми достовірності метаданих та інші), і до проблеми поділу світу на окремі поняття, як вказував ще Аристотель. Важливим фактором стає питання реалізованості онтології верхнього рівня, яка є критичною для успішного розвитку Semantic Web. Всі ці фактори викликали сумніви щодо можливості повноцінного функціонування Semantic Web до теперішнього часу.

У відповідь на цю критику можна сказати, що зусилля, витрачені на розробку Semantic Web, не були марними. За цей час були створені стандарти, формати та програми, які є фундаментом для майбутнього розвитку. Особливо зараз, коли технології штучного інтелекту переживають революційне піднесення, зокрема у зв'язку з розвитком Великих лінгвістичних моделей, ми очікуємо на практичну реалізацію концепції Semantic Web у глобальному масштабі. Це особливо актуально у задачах пошуку, аналітики та підтримки прийняття рішень. У майбутньому Semantic Web може стати потужним інструментом для інтелектуальної взаємодії з інформацією, збагачення даних та покращення процесів роботи з інформацією на світовому рівні.

Питання для самоконтролю

1. Яка можливість представлення знань у вигляді семантичних мереж робить їх цінним інструментом для обробки природної мови та розуміння змісту текстів?
2. Які основні характеристики семантичних мереж і як вони відображають семантичні зв'язки між поняттями?
3. Як семантичні мережі використовуються у додатках обробки природної мови, наприклад, для семантичного розбору або розв'язання багатозначності слів?
4. Які були перші комп'ютерні семантичні мережі та як вони виникли?
5. Які основні аспекти структури семантичних мереж і як вони застосовуються у математиці, лінгвістиці та обробці природної мови?
6. Як визначаються типи семантичних мереж за кількістю типів відносин та по арності?
7. Яка основна ідея концепції Semantic Web?
8. Які основні цілі були пов'язані з розробкою Semantic Web?
9. Що таке Resource Description Framework (RDF)? Які компоненти входять до трійки RDF, та як вони взаємодіють між собою?
10. Які головні переваги застосування онтологій в рамках Semantic Web? Як вони допомагають у структуруванні та визначенні семантики даних?
11. Які можливості надає мова запитів SPARQL? Як вона допомагає виконувати пошук та витягування даних з ресурсів Semantic Web?
12. Яким чином Semantic Web сприяє автоматизованій обробці та взаємодії даних в Інтернеті? Які можливості він відкриває для розвитку інтелектуальних програм та агентів?

5. Мережі мови

Першим кроком у застосуванні теорії Complex Networks[24,25] до текстових документів було створення мережевої моделі цих документів у вигляді набору вузлів і зв'язків, тобто побудова мовних мереж[26], в яких виявляються найбільш значущі вузли, які іноді називаються опорними словами, або відповідними словосполученнями.

Поряд із послідовним аналізом текстових документів, побудова мереж, вузлами яких є такі елементи, як слова чи словосполучення, тобто фрагменти природної мови, дозволяє виявляти структурні елементи текстів, без яких тексти втрачають свою зв'язність. Відомо кілька підходів до побудови мереж з текстів та різні способи інтерпретації вузлів та зв'язків, що призводить, відповідно, до різних видів подання таких мереж. Вузли можуть бути з'єднані між собою, якщо відповідні їм слова стоять поруч у тексті[27,28], належать одному реченню або абзацу[29], синтаксично з'єднані[30], або з'єднані семантично[31,32].

---

[24] Strogatz S.H. Exploring Complex Networks. Nature, 2001. – 410. – pp. 268-276.

[25] Albert R., Barabasi A.-L. Statistical mechanics of complex networks. Reviews of Modern Physics, 2002. – 74. – P. 47.

[26] Головач Ю., Пальчиков В. Лис Микита і мережі мови. Журнал фізичних досліджень. – Т. 11, №. 1, 2007. – С. 22-33.

[27] Ferrer-i-Cancho, R., Sole R.V. The small world of human language. Proc. R. Soc. Lond. B 268, 2261 (2001)

[28] Dorogovtsev S.N., Mendes J. F. F. Language as an evolving word. Proc. R. Soc. Lond. B 268, 2603 (2001)

[29] Caldeira S.M.G., Petit Lobao T.C., Andrade R.F.S., Neme A., Miranda J.G.V. The network of concepts in written texts. Preprint ArXiv. physics/0508066 (2005)

[30] Ferrer-i-Cancho R., Sole R.V., Kohler R. Patterns in syntactic dependency networks. Phys. Rev. E 69, 051915 (2004)

[31] Motter A.E., de Moura A.P.S., Lai Y.-C., Dasgupta P. Topology of the conceptual network of language. Phys. Rev. Fr. E 65, 2002. – 065102(R)

Збереження синтаксичних зв'язків між словами призводить до подання тексту як Directed Network, де напрямок зв'язку відповідає підпорядкуванню слова.

Якщо поставити у відповідність кожному слову вузол мережі та з'єднати кожні два вузли зв'язком тоді, коли відповідні їм слова стоять поруч у реченні, таке представлення називають L-простором. У L-просторі, так само як і в інших наведених нижче мережевих моделях, у разі виникнення кратних зв'язків прийнято зберігати лише один з них.

Традиційно розрізняють чотири різновиди мереж мови (просторів):

1. L-простір. Зв'язуються сусідні слова, які стосуються одного речення. Кількість сусідів кожного слова (вікно слова) визначається радіусом взаємодії R, при цьому найчастіше розглядається випадок R=1.

2. B-простір. Розглядаються вузли двох видів, що відповідають реченням та належним до них словам.

3. P-простір. Усі слова, що стосуються одного речення, пов'язуються між собою.

4. C-простір. Речення зв'язуються між собою, якщо в них вживаються однакові слова.

У разі L-простору зв'язки можуть враховувати не лише «найближчих сусідів», а й групи за кількома словами, які знаходяться на певній відстані одна від одної. Для цього вводиться поняття «радіуса дії» R: при $R=1$ зв'язок існує тільки між найближчими сусідами, при $R=2$ – між найближчими та наступними близькими сусідами тощо. Змінна R може набувати значення від $R=1$ до $R_{\max}$, де $R_{\max}+1$ – загальна кількість слів у реченні.

---

[32] Sigman M., Cecchi G.A. Global Properties of the Wordnet Lexicon. Proc. Natl. Acad. Sci. USA, 99, 1742 (2002)

Зростання «радіусу взаємодії» R у цьому випадку призводить до зростання числа зв'язків, досягаючи насичення при = $R = R_{\max}$.

Ще один спосіб подати текст у вигляді мережі полягає у використанні дводольних (bipartite) графів. У такому поданні (B-простір) розглядаються вузли двох видів. Один вид відповідає реченням, інший – словам. Зв'язок між різними вузлами означає, що слово належить речення.

У P-просторі всі слова, що належать одному реченню, вважаються пов'язаними між собою.

У C-просторі вузли відповідають реченням, а зв'язок між вузлами-реченнями встановлюється у тому випадку, якщо у відповідних реченнях є загальні слова.

Для мережі, побудованої на підставі Британського національного корпусу (L-простір мови, R = 1) виявилося, що ця мережа англійської мови безмасштабна, а поведінка ступеня $P(k)$ характеризується двома режимами статечного розподілу зі значеннями відповідних степеневих показників $\gamma = 1,5$ для k < 2000 та $\gamma = 2,7$ для k > 2000.

Відповідно до визначення, якщо середня довжина найкоротшого шляху зростає з розміром (кількістю вузлів) мережі повільніше за будь-яку статечну функцію, то мережа є «малим світом». Мережі малого світу дуже компактні. Для вищезгаданої мережі англійської мови довжина найкоротшого шляху становить лише $< l >= 2,63$. Оскільки зростання R призводить тільки до додавання нових зв'язків, то $< l >$ зменшуються зі зростанням R.

Специфічною формою кореляції у мережах є утворення кластерів. Коефіцієнт кластеризації C характеризує схильність мережі до утворення з'єднаних трійок вузлів. Відомо, що для повного графа C = 1, а для мережі у формі дерева C = 0

Відношення середнього коефіцієнта кластеризації мереж, що вивчаються, до коефіцієнта кластеризації класичного випадкового графа свідчить про те, що мережі мови є добре корельованими структурами. Такі кореляції зростають із зростанням «радіусу взаємодії» R.

Для Британського національного корпусу на підставі аналізу текстів, що містять $\approx 10^7$ слів, отримано значення коефіцієнта кластеризації $< C >= 0,687$.

У разі розгляду P-простору, кожне слово-вузол пов'язане з усіма іншими словами, що належать до загального речення. Таким чином, кожне речення тексту входить до мережі як повний граф – кліки взаємопов'язаних вузлів. Різні речення-кліки поєднуються в мережу завдяки загальним словам. У L-просторі слова зв'язуються усередині вікна, розмір якого характеризуються величиною R. Коли розмір вікна R стає рівним розміру речення, то подання цього речення в L-і P-просторах збігаються. Відповідно, коли розмір вікна стає рівним розміру найбільшого речення тексту ($R = R_{\max}$), то представлення всього тексту в L- і P-просторах збігаються.

На практиці підтверджено, що мережа мови є дуже корельованим безмасштабним світом (Scale-Free Small World). Існує ряд робіт, в яких зроблено спробу пояснити властивості мереж мови за допомогою сценарію переважного приєднання (Preferential Attachment [Albert, 1999]), розглядаючи їх як результат процесу зростання, коли нові вузли-слова з більшою ймовірністю приєднуються до вузлів-хабів, що мають багато зв'язків.

Питання для самоконтролю

1. Як змінюється середня довжина найкоротшого шляху в мережі зі зростанням кількості вузлів? Як це співвідноситься з поняттям "малий світ"?
2. Яким чином визначається коефіцієнт кластеризації C у мережі? Як він характеризує схильність мережі до утворення з'єднаних трійок вузлів?

3. Які методи аналізу були застосовані до Британського національного корпусу для вивчення коефіцієнта кластеризації? Яке значення було отримано в результаті?
4. Які основні типи просторів (мереж) використовуються для моделювання мереж мови? Наведіть приклади кожного типу.
5. Що таке "радіус взаємодії" у мережах мови, і як він впливає на кількість та структуру зв'язків?
6. Які основні характеристики властиві мережам мови, побудованим за різними типами просторів (L, B, P, C)?
7. Які можливі застосування мереж мови у реальному житті, зокрема у вивченні мови, обробці текстів чи аналізі даних?
8. Яким чином виражається P-простір мови? Які взаємозв'язки встановлюються між словами в цьому просторі?
9. Які основні властивості L-простору мови? Які слова об'єднуються в цьому просторі та які зв'язки між ними враховуються?
10. Які результати свідчать про те, що мережа мови є малим світом? Які інші властивості цієї мережі було встановлено на практиці?
11. Як можна пояснити властивості мереж мови за допомогою сценарію переважного приєднання? Яким чином цей процес впливає на структуру мережі та її властивості?
12. Яким чином мережі мови відрізняються від класичних випадкових графів за показником кластеризації? Які кореляції можуть бути виявлені при зростанні "радіусу взаємодії"?
13. Які режими поведінки ступеня P(k) можуть бути спостережені в мережі мови? Які значення k визначають ці режими?
14. Як виглядає подання тексту у вигляді дводольного графа (B-простір)? Які зв'язки встановлюються між вузлами цього графа?

6. Графи видимості

6.1 Перетворення часових рядів у складні мережі

Стохастичні та хаотичні ряди є різними типами послідовностей, які виявляють певний ступінь непередбачуваності та випадковості. Ось основні відмінності між ними:

1. Статистична природа:
   - Стохастичний ряд: стохастичні ряди є послідовностями, які можна пояснити в термінах ймовірнісних закономірностей або випадкових процесів. Вони можуть мати якусь структуру чи тенденцію, але основний вплив з їхньої поведінки надають випадкові чинники.
   - Хаотичний ряд: хаотичні ряди також виявляють випадкові характеристики, але вони засновані на детермінованих системах, які чутливі до початкових умов та демонструють експоненційно зростаючу чутливість до малих змін. Це означає, що навіть невеликі варіації у початкових умовах можуть призвести до суттєво різних результатів.
2. Поведінка:
   - Стохастичний ряд: стохастичні ряди можуть мати різні рівні передбачуваності, їхня поведінка може бути описана з використанням імовірнісних моделей, таких як статистичні розподіли або авторегресійні моделі.
   - Хаотичний ряд: хаотичні ряди складніше передбачити через їх високу чутливість до початкових умов. Навіть якщо система детермінована і не містить випадкових елементів, найменші відхилення в початкових умовах можуть призвести до значних відмінностей у довгостроковій поведінці ряду.
3. Динамічні системи:

- Стохастичний ряд: багато стохастичних рядів можуть бути описані за допомогою стохастичних диференціальних рівнянь або інших ймовірнісних моделей, які враховують випадкові фактори.
- Хаотичний ряд: хаотичні ряди зазвичай виникають у детермінованих динамічних системах, які характеризуються складною поведінкою, але не мають стохастичних компонентів у моделях.

4. Відносна передбачуваність:
   - Стохастичний ряд: стохастичні ряди, хоч і непередбачувані в короткому терміні, проте можуть бути передбачувані в середньому або в довгостроковій перспективі при використанні статистичних методів і моделей.
   - Хаотичний ряд: хаотичні ряди зазвичай не мають періодичних компонентів або простих тенденцій і їх прогноз в довгостроковій перспективі обмежений високою чутливістю до початкових умов.

Важливо, що межа між стохастичними і хаотичними рядами може бути розмитою і деякі системи можуть виявляти змішані характеристики обох типів. Ці типи рядів та їхні властивості вивчаються у різних галузях науки, як-то математика, фізика, економіка, кібербезпека та інші.

Ідея вивчення складних часових рядів за допомогою відображення їх у складні мережі (графи) є дуже привабливою. При такому відображенні поєднуються дві розвинені області досліджень – нелінійні методи аналізу часових рядів та методи теорії складних мереж. З'являється можливість застосувати багаті, добре розвинені методи аналізу складних мереж для аналізу складних структур, наприклад, фрактальних, часових рядів. Викладемо міркування наведені у зазначеному вище міркуванні.

Не зважаючи на будь-який основний процес (і, отже, на будь-яке фізичне, хімічне, економічне або якесь інше значення його просто числових значень), ми можемо розглядати часові ряди просто як упорядкований набір значень та грати в наївну математичну гру, перетворюючи цей набір на інший математичний об'єкт за допомогою абстрактного відображення та дивитися, що відбувається: які властивості вихідного набору зберігаються, які трансформуються і що ми можемо сказати про математичне уявлення, просто подивившись на інше... Ця вправа представляє самостійний інтерес з математичної точки зору. Крім того, виявляється, що тимчасові ряди або сигнали є універсальним методом вилучення інформації з динамічних систем у будь-якій галузі науки. Тому попередня математична гра набуває несподіваного практичного інтересу тому, що відкриває можливість аналізу часових рядів (тобто результатів динамічного процесу) з альтернативної точки зору. Звичайно, інформація, що зберігається у вихідному часовому ряді, має якимось чином зберігатися у відображенні. Мотивація завершується, коли нове уявлення відноситься до відносно зрілої математичної області, в якій інформація, закодована в такому поданні, може бути ефективно виділена та оброблена. Коротко це перша мотивація для відображення тимчасових рядів у мережі.

Ця мотивація посилюється двома взаємозалежними чинниками: по-перше, як і раніше, аналіз часових рядів є зрілої областю, він має деякі обмеження, коли йдеться про вивчення складних сигналів. За межами лінійного режиму існує широкий спектр явищ (які обмежуються фізикою), які зазвичай входять до області складних систем. В основі цього нечіткого визначення лежить загальна характеристика: значний вплив нелінійності у тому математичному представленні. Ця властивість може виявлятися в тимчасовій еволюції (принаймні одній з) змінних, що описують систему, і вимагає використання

специфічних інструментів для нелінійного аналізу. Динамічні явища, такі як хаос, стохастичні процеси з далекодією, мультифрактальність тощо є прикладами складних явищ, де аналіз часових рядів виходить за його межі. Аналіз нелінійних часових рядів розвивається за допомогою таких технік, як нелінійні кореляційні функції, алгоритми вкладення, мультифрактальний метод спектру, теореми проекції та інші інструменти, які стають складнішими разом із збільшенням складності процесу/ряду, що вивчається. Нові підходи, нові парадигми для роботи зі складністю не лише вітаються, а й необхідні.

Ідея відображення часових рядів у графи здається привабливою тому, що вона поєднує дві плідні галузі сучасної науки: аналіз нелінійних сигналів та теорію складних мереж. Ця ідея привернула увагу кількох дослідницьких груп, які зробили свій внесок у цю тему, використовуючи різні стратегії відображення. Коротко опишемо деякі з них.

J. Zhang і M. Small[33] розробили метод, який відображав кожен цикл псевдоперіодичного часового ряду у вузол графа. Зв'язок між вузлами встановлювався на основі порогового значення відстані у відновленому фазовому просторі або на основі лінійного коефіцієнта кореляції між циклами у присутності шуму. Шумні періодичні часові ряди відображалися у випадкові графи, а хаотичні часові ряди – у масштабно-вільні мережі з невеликими світами через наявність нестійких періодичних орбіт. Цей метод був застосований для характеристики динаміки серця.

X. Xu, J. Zhang і M. Small[34] зосередилися на відносних частотах появи мотивів усередині конкретного графа, щоб класифікувати їх у певне сімейство мереж, відповідне

[33] J. Zhang, M. Small, Phys. Rev. Lett. 96, 238701 (2006)
[34] X. Xu, J. Zhang, M. Small, Proc. Natl. Acad. Sci. U.S.A. 105, 19601 (2008)

специфічній динаміці відображеного часового ряду. У даному випадку метод відображення полягав у вкладенні часового ряду у відповідний фазовий простір, де кожна точка відповідала вузлу мережі. Був встановлений поріг не тільки для мінімальної відстані між двома сусідами (тимчасовий поділ повинен бути більшим за середній період даних), але й для максимальної кількості сусідів, яка може мати вузол. Різні сімейства були виявлені для хаотичної, випадкової та галасливої періодичної динаміки, а також були знайдені унікальні характеристики для конкретних динамічних систем усередині сімейств.

R.V. Donner і його співавтори[35] представили метод, що ґрунтується на властивостях рекурентності у фазовому просторі динамічної системи. Більш точно, матриця рекурентності, отримана шляхом встановлення порогу для мінімальної відстані між двома точками у фазовому просторі, інтерпретувалася як матриця суміжності неорієнтованого, незваженого графа. Властивості таких графів було досліджено на кількох парадигматичних системах (відображення Хенона, система Росслера, система Лоренца, відображення Бернуллі).

Якщо в роботі J. Zhang, M. Small було запропоновано використати близькість координат у перерізі Пуанкаре часового ряду, то в роботах [36,37] було запропоновано алгоритм побудови Natural Visibility Graph (NVG) (граф натуральної видимості) та Hrizontal Visibility Graph (HVG) (граф горизонтальної видимості).

---

[35] R.V.Donner, Y.Zou, J.F.Donges, N.Marwan, J.Kurths, Phys.Rev.E 81, 015101 (2010)

[36] *L.Lacasa, B.Luque, F.Ballesteros, J.Luque, J.C. Nuno.* From time series to complex networks: The visibility graph, Proceedings of the National Academy of Sciences, 105 (13) 4972-4975, 2008

[37] *B. Luque, L. Lacasa, F. Ballesteros, J. Luque.* Horizontal visibility graphs: Exact results for random time series, Physical Review E 80 (4), 046103, 2009

Розглянемо, яким чином часовий ряд може бути перетворений на граф.

Нехай часовий ряд заданий у вигляді $\{x(t_i), i=1..N\}$, де $x(t_i)$ – значення досліджуваної величини в момент часу t_i. NVG будується в такий спосіб. На горизонтальній осі відзначаються точки t_i, до яких у перпендикулярному напрямі будуються відрізки заввишки $x(t_i)$. Під вершинами NVG розуміються зовнішні кінці збудованих відрізків. Зв'язок між вершинами графа вважається існуючим, якщо пряма, що з'єднує кінці відрізків, не перетинає жодного з відрізків, що знаходяться між ними.

Дещо пізніше був запропонований «близький за духом» до NVG новий Horizontal Visibility Graph (HVG). У цьому HVG алгоритмі, існування зв'язку між вершинами визначається тим, чи можна з'єднати відрізки горизонтальною лінією, не перетинаючи жодного з відрізків, що знаходяться між ними. Таким чином, як NVG так і HVG є зв'язковими графами і є єдиним кластером. Структура цього кластера містить у прихованому вигляді закономірності вихідного часового ряду.

Застосування NVG і HVG алгоритмів дозволило описати і дослідити тимчасові ряди складної структури, пов'язані з різними явищами: пульсаціями турбулентних течій, stock market indices, human heartbeat dynamics – стохастичними та хаотичними рядами.

У всіх згаданих вище алгоритмах перекладу time series у complex networks кожному часовому ряду $x(t_i)$ відповідає свій граф (див. рис. 33).

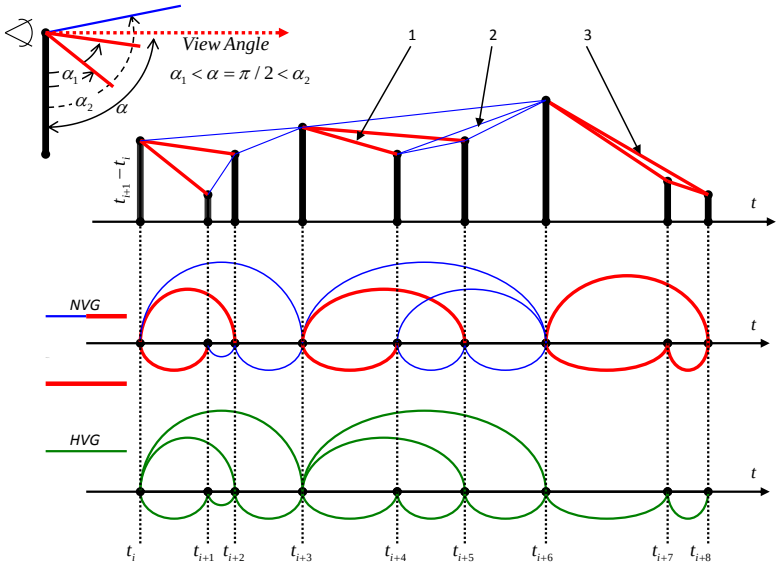


Рисунок 33 – Алгоритми побудови графів видимості

На рисунку 33 ліворуч угорі – схема вибору зв'язку для DVG(α). Зв'язок із кутом α_1 (червона лінія) належить PVG(α), зв'язок із α_2 (синя лінія) – ні. На горизонтальній осі часу відкладено послідовний набір моментів часу $\{t_i, i = 1..N\}$, величина vertical bars для моменту часу t_i дорівнює $t_{i+1} - t_i$. Як приклад, тут кут зору обраний рівним $\pi/2$– пунктирна стрілка на рисунку вгорі зліва. Лінії, з'єднуючі вершини vertical bars є зв'язками NVG. З них тонкі лінії (сині) – 2 мають кут більше ніж кут зору, і тому належать тільки NVG, а товсті (червоні online) – 1 менше ніж кут зору і тому належать як NVG так і DVG($\pi/2$), 3 – приклад зв'язку, що належить NVG та DVG, але не належить HVG. b. товсті лінії – зв'язки DVG($\pi/2$), товсті та тонкі – NVG. HVG – зелені лінії.

Опишемо ще один алгоритм[38] - алгоритм побудови Parametrical Visibility Graph (PVG), коли кожному часовому ряду $x(t_i)$ відповідає набір графів. DVG algorithm зводиться до наступного: для кожного із зв'язків NVG стандартного графа взаємної видимості обчислюється кут нахилу зв'язку стосовно вертикальної осі. Зв'язками PVG графа будуть лише ті, нахил яких менший за заданий кут α. Кут зору α – параметр, що безперервно змінюється, а кожному значенню цього параметра відповідає свій граф, що є підграфом стандартного графа взаємної видимості (NVG). Таким чином, PVG алгоритм можна вважати узагальненням NVG алгоритму.

Можливість довільно змінювати кут зору α додала назву алгоритму визначення динамічний. Далі будемо використовувати також скорочене позначення PVG(α). PVG алгоритм дозволяє будувати набори PVG(α) і потім досліджувати залежність параметрів побудованого графа від кута зору α.

Почнемо з конструювання складної мережі з часових міток (даних) , які представляють послідовний набір моментів часу $\{t_i,\ i=1..N\}$, в які відбуваються деякі події, наприклад, удари серця. Зіставимо цьому набору часовий ряд $\{x(t_i)=t_{i+1}-t_i,\ i=1..N-1\}$. При цьому ми вважаємо, що подія $x(t_i)$ відбувається у час t_i.

Розглянемо спочатку фрагмент стандартного графа взаємної видимості VG.

Визначимо формальний критерій видимості, тобто умова, при виконанні якої зв'язок NVG графа належатиме

[38] Bezsudnov I.V., Snarskii A.A. From the time series to the complex networks: The parametric natural visibility graph, Physica A: Statistical Mechanics and its Applications 414, 53-60

PVG(α). Для цього розглянемо два довільні моменти часу t_i та $t_k, i<k$ і всі моменти часу між ними t_j, $i<j<k$. Для PVG(α) критерій видимості, тобто існування зв'язку між вузлами i та k в NVG:

$$x_k < x_i + \left(x_j - x_i\right)\frac{t_k - t_i}{t_j - t_i}, \quad i<j<k$$

повинен бути доповнений умовою, що обмежує кут зору α

$$\alpha > \alpha_{ik} = arctg\,\frac{x_k - x_i}{t_k - t_i}\,.$$

Зазначимо, що з кута зору $\alpha = \pi$ PVG(π) перетворюється на NVG. Однак PVG($\pi/2$) не переходить у HVG, що на наш погляд підкреслює відмінності графів, побудованих за цими двома алгоритмами. Наприклад на рисунку 1 показано 3 зв'язки, що належать HVG. Зв'язок 1 належить як VG, PVG($\pi/2$) і HVG графу, зв'язок 2 – належить VG і HVG графу, тоді як він відсутній в PVG($\pi/2$), зв'язок 3 належить як VG, DVG($\pi/2$), але не належить HVG.

Зауважимо також, що PVG алгоритм має принципову відмінність від NVG та HVG алгоритмів. Графи, до яких приводять NVG та HVG алгоритми не залежать від того, в яку сторону спрямована тимчасова вісь. Тобто алгоритми NVG та HVG для часового ряду $x_1, x_2, ...x_N$ та зворотного до нього $x_N, x_{N-1}, ...x_1$ призводять до того ж самого графу.

У випадку PVG алгоритму, завжди знайдеться такий кут зору α, коли графи побудовані за часовим рядом $t_1, t_2, ...t_N$ і $t_N, t_{N-1}, ...t_1$ відрізнятимуться. Це видно на прикладі (див. рис. 33) для останніх трьох вертикальних смуг. При

обраному на цьому рисунку куті зору ($\alpha = \pi / 2$) при прямій побудові, вузли мережі, відповідні трьом останнім вертикальним смугами - з'єднані, у разі зворотного порядку – ні.

Закінчуючи обговорення PVG алгоритму та його зв'язку з NVG та HVG алгоритмами, необхідно зауважити, що можливі й інші алгоритми побудови графів, що враховують геометричні або інші особливості вихідного часового ряду. Наприклад, коли за схожим правилом з'єднуються не вершини вертикальних смуг, а будь-які точки.

Розглянемо кілька властивостей NVG та HVGT.

Середній ступінь горизонтального графа видимості, пов'язаного з нескінченним періодичним поруч із періодом T (без повторюваних значень усередині періоду), дорівнює:

$$\langle k(T) \rangle = 4\left(1 - \frac{1}{2T}\right) .$$

Цікавим наслідком попереднього результату є те, що кожен тимчасовий ряд, витягнутий з динамічної системи, має пов'язаний з ним горизонтальний граф видимості із середнім ступенем

$$2 \le \langle k \rangle \le 4 .$$

Залежність відносного середнього ступеня вузлів графа від кута зору для часових рядів, отриманих зі випадкового однорідного розподілу, рядів, що відповідають розподілу Пуассона та модулю значень ряду Вейєрштраса з фрактальною розмірністю:

$$P(k) = \frac{2}{3}\left(\frac{1}{3}\right)^{k-2}, \quad k = 2,3,4,...$$

Зауважимо, що середній ступінь $\langle k \rangle$ горизонтального графа видимості, пов'язаного з некорельованим випадковим процесом, визначається таким чином:

$$\langle k \rangle = \sum kP(k) = \sum_{k=2}^{\infty} \frac{k}{3} \left(\frac{2}{3} \right)^{k-2} = 4 ,$$

що добре відповідає передбаченню з попереднього рівняння у граничному випадку $T \to \infty$, тобто для аперіодичної послідовності.

Розглянемо тепер деякі характеристики PVG. Позначимо, як $\bar{k}_\alpha$ середній ступінь вузлів графа, де індекс α свідчить про те, що обчислення проводилося для PVG(α). У випадку $\alpha = \pi$, тобто коли PVG перетворюється на NVG, характеристика $\bar{k}_{\alpha=\pi}$ – це «звичайний» середній ступінь вузлів графа $\langle k \rangle = \sum_{i=1}^{N} k_i / N$, де N – повне число вузлів.

Відносний середній ступінь вузлів графа PVG(α) визначатиметься як:

$$K(\alpha) = \bar{k}_\alpha / \bar{k}_\pi .$$

При цьому для окремого випадку $\alpha = \pi$, коли DVG переходить в NVG, ця характеристика завжди $K(\alpha = \pi) = 1$. Зауважимо, що також завжди має місце $K(\alpha < \pi / 4) = 0$.

Відносною середньою довжиною зв'язку $\Lambda(\alpha)$ довгого зв'язку графа видимості назвемо величину часового інтервалу, що поділяє дві події у часі видимі одна одній, тобто якщо між вузлами мережі t_i та t_{i+k} існує зв'язок, то довжиною зв'язку будемо називати величину $l_{i+k,i} = t_{i+k} - t_i$. Позначимо середню довжину зв'язку графа видимості як $\bar{l}_\alpha$, де індекс α означає те, що усереднення проводилося з усіх зв'язків PVG(α), побудованого при куті зору α. Таким чином, відносна середня довжина зв'язку дорівнює

$$\Lambda(\alpha) = \bar{l}_\alpha / \bar{l}_\pi .$$

Як і щодо відносного середнього ступеня вузлів графа $K(\alpha)$ має місце $\Lambda(\alpha = \pi) = 1$ і $\Lambda(\alpha < \pi/4) = 0$.

Число кластерів (не пов'язаних один з одним) у PVG(α) – $Q(\alpha)$. Цей параметр є специфічною для PVG(α) характеристикою. До кута $\alpha = \pi/4$ зв'язків у графі немає – $Q(\alpha < \pi/4) = 0$. При зміні куту зору α від $\pi/4$ до π, кількість зв'язків графу зростає, причому спочатку, при кутах, поблизу $\pi/4$, граф буде складатися з великої кількості невеликих кластерів, які при подальшому збільшенні кута зору об'єднуватимуться, а їх кількість буде зменшуватися. Зрештою, коли кут зору досягне свого максимального значення рівного π, і PVG перетвориться на NVG, у графі залишиться лише один кластер, що включає всі вузли – $Q(\pi) = 1$.

Розрахуємо залежності описаних вище параметрів $K(\alpha)$, $\Lambda(\alpha), Q(\alpha)$ – від кута зору в діапазоні $-\pi/4 < \alpha < \pi/2$. Розрахунок проводився для трьох різних випадкових розподілів: випадкового однорідного (a), ряду, що відповідає розподілу Пуассона (b) і модулю значень ряду Вейєрштрасса (c), що має фрактальну розмірність.

$$x_k = x_{k-1} + r , \qquad \text{(a)}$$

$$x_k = x_{k-1} - 1/\lambda \ln(r) , \qquad \text{(b)}$$

$$x_k = \left| \sqrt{2}\sigma \frac{\sqrt{1 - b^{2D-4}}}{\sqrt{1 - b^{(2D-4)(N+1)}}} \sum_{n=0}^{N} \left[b^{(D-2)n} \sin\left(2\pi(sb^n k + r)\right) \right] \right| , \qquad \text{(c)}$$

де r, – випадкова величина, розподілена рівномірно на відрізку $[0,1]$. Було вибрано такі чисельні значення параметрів: для розподілу Пуассона – $\lambda = 1$, для ряду Вейєрштраса – $\sigma = 3.3$, $b = 2.5$, $N = 10$, D – фрактальна розмірність ряду вибиралася в діапазоні $[1.0 \ldots 1.99]$. Для побудови PVG(α) використовувався часовий ряд завдовжки 10^5.

На рисунку 34 представлена залежність відносного середнього ступеня вузлів графа – $K(\alpha)$, фрактальна розмірність ряду Вейєрштраса $D = 1.3$.

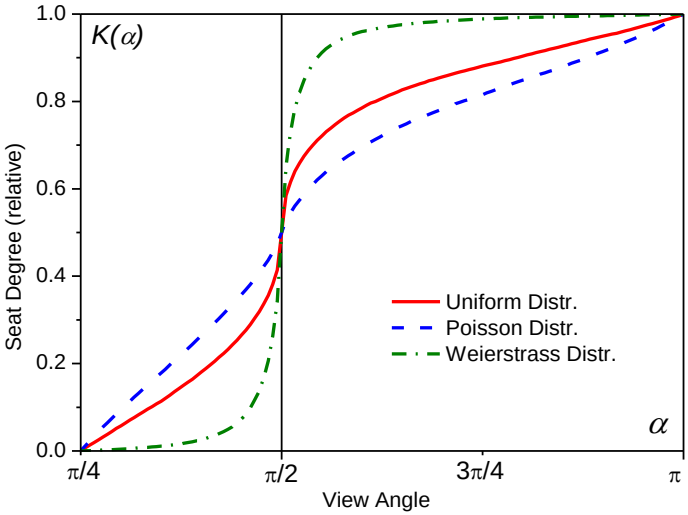


Рисунок 34 – Залежність відносного середнього ступеня вузлів графа $K(\alpha)$ від кута зору для тимчасових рядів, отриманих з випадкового однорідного розподілу, рядів, що відповідають розподілу Пуассона та модулю значень ряду Вейєрштраса з фрактальною розмірністю $D = 1.3$

Залежність $K(\alpha)$ досліджених рядів має гладку s-подібну форму. При $\alpha = \pi/2$, тобто при горизонтальному промені зору, всі три розподіли дають те саме значення $K(\alpha = \pi/2)$. Однак похідна в цій точці $\partial K(\alpha)/\partial\alpha\big|_{\alpha=\pi/2}$ для цих розподілів, як видно з рисунку 34 різна для різних розподілів. Зауважимо також, що хоча за визначенням, $K(\alpha = \pi) = 1$ для будь-яких розподілів, наявність «динаміки» залежності властивостей PVG(α) від кута зору і при $\alpha = \pi$ дозволяє розрізнити розподіли, наприклад, за їх похідною в цій точці – $\partial K(\alpha)/\partial\alpha\big|_{\alpha=\pi}$.

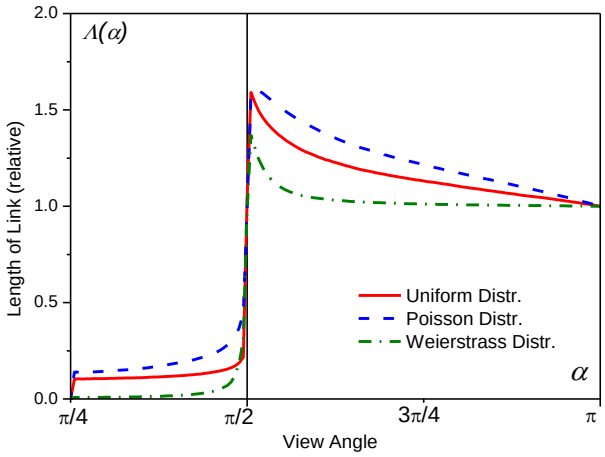


Рисунок 35 – Залежність довжини зв'язку графа динамічної видимості $\Lambda(\alpha)$ від кута видимості для тимчасових рядів отриманих з випадкового однорідного розподілу, рядів відповідного розподілу Пуассона і модуля значень ряду Вейєрштраса з фрактальною розмірністю $D = 1.3$

На рисунку 35 наведено залежність довжини зв'язку графа динамічної видимості $\Lambda(\alpha)$ від кута видимості для тих самих розподілів. Як видно з рисунку, для кожного розподілу характерний свій максимум поблизу кута $\pi/2$.

Цей пік у кутовій залежності пов'язаний з тим, що при збільшенні кута зору відразу після кута $\pi/2$ стає «видно далеко», тобто з'являється велика кількість довгих зв'язків, наприклад $t_i - t_{i+3}$ (див. рис. 32). При подальшому підвищенні кута зору починає з'являтися все більше коротких зв'язків, наприклад, $t_{i+2} - t_{i+3}$ або $t_{i+4} - t_{i+5}$ (див. рис. 35), які при обчисленні середнього за довжиною зв'язку знижують значення $\Lambda(\alpha)$.

Для розподілу Вейєрштраса можна вирішити і зворотну задачу, знаючи залежність $\Lambda(\alpha)$, $\alpha = 0..\pi$ визначити параметр розподілу D. На рисунку 36 показано, як $\Lambda(\alpha)$, $\alpha = 0..\pi$, так і залежність $\max\limits_{\alpha=0..\pi} \Lambda(\alpha)$ від D – фрактальної розмірності функції Вейєрштраса.

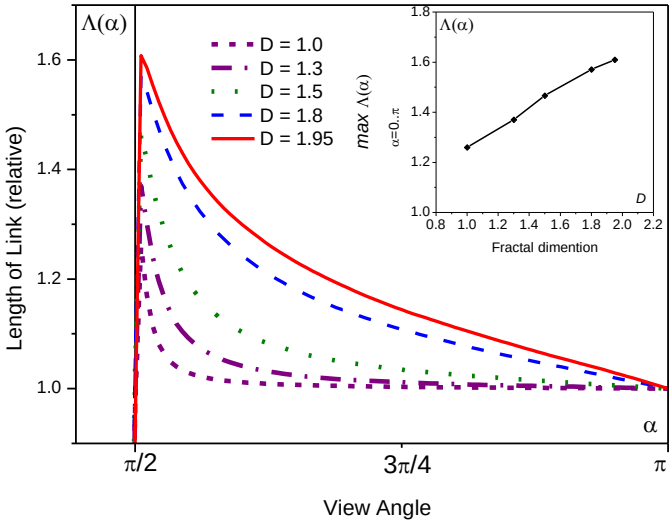


Рисунок 36 – Залежність відносної довжини зв'язку графа $\Lambda(\alpha)$ для розподілу Вейєрштраса з різними параметрами D

(вказані на рисунку), На врізанні – залежність $\max_{\alpha=0..\pi} \Lambda(\alpha)$ від фрактальної розмірності D

Як видно з урізання (дитв. рис. 36) має місце монотонна (практично лінійна) залежність максимального значення $\max_{\alpha=0..\pi} \Lambda(\alpha)$ від параметра D (фрактальної розмірності), тобто можна говорити про однозначний зв'язок між максимумом $\Lambda(\alpha)$ та фрактальною розмірністю розподілу Вейєрштраса. Таким чином, володіючи інформацією про тип розподілу вихідних інтервалів, виявляється можливим визначити і самі параметри (частково або повністю) вихідного розподілу.

На рисунку 37 показано число незв'язних кластерів у DVG як функція кута зору α – $Q(\alpha)$.

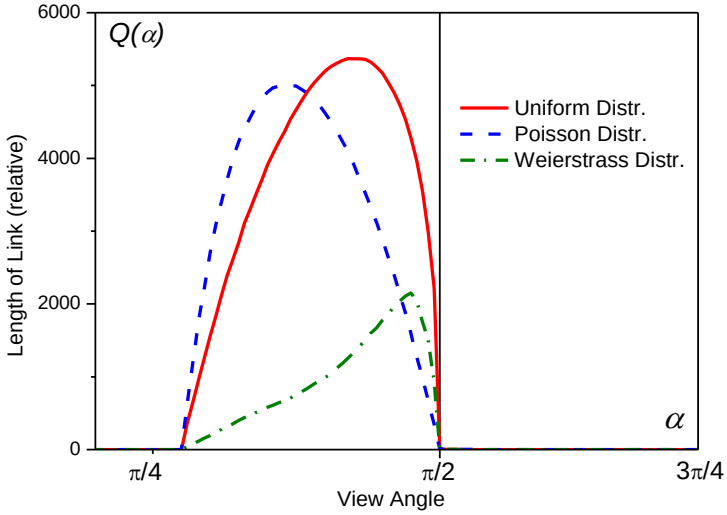


Рисунок 37 – Число незв'язаних кластерів $Q(\alpha)$ у DVG як функція кута зору α для часових рядів отриманих з випадкового однорідного розподілу, рядів, що відповідає

розподілу Пуассона і модулю значень ряду Вейєрштраса з фрактальною розмірністю $D = 1.3$

Форма залежності $Q(\alpha)$ свідчить про те, як змінюється число кластерів зі збільшенням кута зору α. Починаючи з деякого кута зору $\alpha > \pi/4$ з'являються кластери, кількість їх з одного боку зростає за рахунок появи зв'язків між ближніми вузлами, з іншого боку падає, за рахунок появи зв'язків між кластерами та їхнього об'єднання. Спочатку перемагає перший механізм – механізм зростання, потім (максимум на рис. 5) починає перемагати другий механізм – об'єднання. І, зрештою, при куті зору рівному приблизно $\pi/2$ всі кластери об'єднуються в один.

Залежності, наведені на рисунках 34, 35, 36 та 37 показують, що характеристики досліджених часових рядів суттєво відрізняються одна від одної.

Можна також розрахувати і залежність інших параметрів PVG(α). Серед цих параметрів різні середні, наприклад, середній кут зв'язку в PVG(α), а також параметри, що розраховуються для складних мереж, такі як коннективіті і асортативити графів, що утворюються.

Розглянемо часові ряди RR інтервалів ЕКГ здорових та хворих людей. Дані ЕКГ було взято з бази даних PhysioNet. Було вибрано три типи серцевого ритму. Перший з них – здорові пацієнти, всього 72 записи. Дані серцевого ритму людей з Congestive Heart Failure, всього 44 записи та дані серцевого ритму для людей з Atrial Fibrillation – 25 записів. Довжина запису не постійна, кожен запис містить $6 \div 11 \times 10^4$ RR інтервалів.

Для кожного запису були побудовані залежність параметрів PVG(α) від кута видимості, у тому числі, $\Lambda(\alpha)$, $K(\alpha)$ і $Q(\alpha)$. Потім було виконано зосередження знайдених залежностей за типами серцевого ритму. На рисунку 38 наведено залежності відносного середнього

ступеня вузла $Q(\alpha)$ від кута зору α та параметричну залежність $Q(\alpha)$ від $K(\alpha)$.

Кожен тип серцевого ритму продукує власну форму залежності (див. рис. 38), що відрізняє тип ритму від іншого. Наприклад, у наведених на рисунку 38 залежностях $Q(\alpha)$ виявляється можливим вказати такі діапазони кутів зору α або значення середнього ступеня кута $K(\alpha)$, коли добре помітні три розглянуті типи серцевого ритму – нормальний, з Congestive Heart Failure і Atrial Fibrillation.

Усі показані вище залежності параметрів DVG(α) як модельні (див. рис. 35-38), так і експериментальні (див. рис. 35) дозволяють розрізняти, ідентифікувати і описувати різні часові ряди. Алгоритм побудови графа динамічної видимості дає можливість визначення нових, динамічних характеристик графа видимості. Вони дозволяють більш детально та глибоко характеризувати випадкові розподіли часових інтервалів.

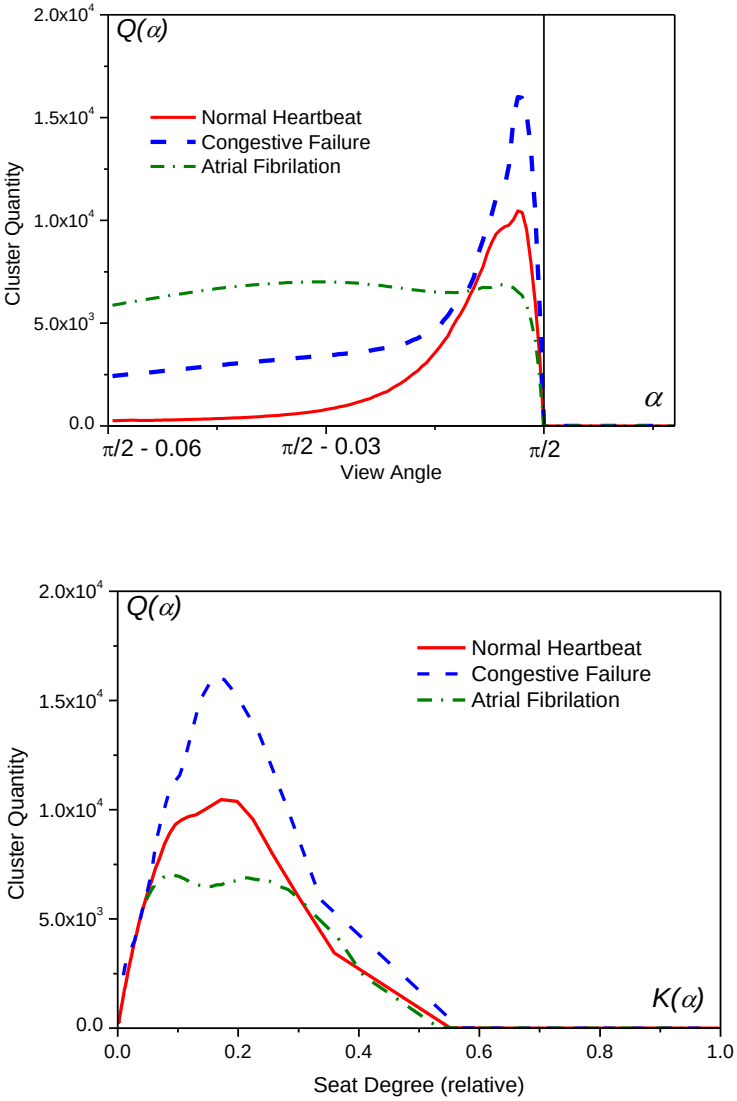


Рисунок 38 – Число незв'язаних кластерів $Q(\alpha)$ як функція кута зору α, і як функція відносного середнього ступеня вузла $K(\alpha)$ для різних типів RR-інтервалів

6.2 Графи горизонтальної видимості як засіб витягу визначальних слів тексту

На даний час актуальним є завдання визначення того, які з важливих структурних елементів тексту виявляються інформаційно-значущими, тобто такими, що визначають інформаційну структуру тексту. Використання таких елементів в якості опорних слів дозволяє формувати онтології, тезауруси, пошукові образи, зокрема, при обробці законодавчих актів та іншої нормативно-правової інформації. Такі елементи можуть, зокрема, використовуватися також для ідентифікації таких компонент тексту, як коллокація та надфразова єдність.

Опірні слова для пошуку в тексті та автоматичного екстрагування значущих фрагментів вибираються з урахуванням такої властивості слів, як «розпізнавальна» або дискримінантна сила. При аналізі текстів з правової тематики, зокрема, при вирішенні завдання формування електронної енциклопедії на основі аналізу всього масиву законодавчих актів України, оцінка дискримінантної сили окремих слів має найважливіше значення[39].

У цій роботі запропоновано методику виявлення опірних слів за допомогою, так званих, мереж мови (Language Network). Разом з послідовним аналізом текстів, побудова мереж, вузлами яких є їх елементи – слова або словосполучення, фрагменти природної мови, дозволяє виявляти структурні елементи тексту, без яких він втрачає свою зв'язність. Відомо декілька підходів до побудови мереж з текстів, так званих, мереж мови, і різні способи інтерпретації вузлів і зв'язків, що приводить, відповідно, до різних видів представлення таких мереж. Вузли можуть бути сполучені між собою, якщо відповідні їм слова

[39] Ланде Д.В. Методи оцінки рівня дискримінантної сили слів у текстах з правової тематики // Правова інформатика, 2012. – № 3 (35). – С. 5-9.

стоять поряд у тексті, належать до одного речення або абзацу[40], сполучені синтаксично або семантично [8, 9].

У рамках концепції складних мереж (Complex Networks) запропоновано декілька методів побудови мереж на основі часових рядів, серед яких можна назвати декілька методів побудови графів видимості, зокрема, так званий граф горизонтальної видимості (Horizontal Visibility Graph – HVG). Ці підходи також дозволяють будувати мережеві структури на підставі текстів, в яких окремим словам або словосполученням деяким спеціальним чином поставлені у відповідність числові вагові значення. Як функція, що ставить у відповідність слову число, можна розглядати, наприклад, порядковий номер унікального слова у тексті, довжину слова, загальноприйняту оцінку $TF \cdot IDF$ (у канонічному виді, рівну добутку частоти слова у фрагменті тексту – term frequency – на двійковий логарифм від величини, зворотної кількості фрагментів тексту, в яких це слово зустрілось, – inverse document frequency) або її варіанти, а також інші вагові оцінки.

Для підрахунку вагової оцінки $TF \cdot IDF$ з повного тексту, що складається з N слів, текст розбивається на фрагменти, які містять задану кількість слів M (наприклад, $M = 500$). Після цього для кожного слова i, що входить до тексту, підраховується кількість фрагментів $df(i)$, в яких міститься це слово, а також загальна кількість входження даного слова i у текст – $n(i)$. Після цього за формулою

$$tfidf(i) = \frac{n(i)}{N} \log\left(\frac{N}{M \times df(i)}\right)$$

розраховується середнє значення TFIDF вагової оцінки для кожного слова.

---

[40] Caldeira S. M. G., Petit Lobao T. C., Andrade R. F. S., Neme A., Miranda J. G. V. The network of concepts in written texts // Preprint physics/0508066 (2005).

При побудові мереж слів у цьому разі застосовується дисперсійна оцінка ваги слів, яка обчислюється наступним чином чином: деяке слово, наприклад A, позначається як A_k^n, де індекс $k = 1,2,...,K$ – номер появи даного слова у тексті, а n – позиція даного слова у тексті[41]. Наприклад, A_3^{50} означає, що на 50-й позиції тексту знаходиться слово A, яке зустрілось третій раз. Інтервалом між послідовними появами слова при таких позначеннях буде величина $\Delta A_k = A_{k+1}^m - A_k^n = m - n$, де на m-й та n-й позиції в тесті знаходиться слово A, яке зустрілось $k+1$-й і k-й рази.

Дисперсійна оцінка розраховується як

$$\sigma_A = \frac{\sqrt{\langle \Delta A^2 \rangle - \langle \Delta A \rangle^2}}{\langle \Delta A \rangle},$$

де: $\langle \Delta A \rangle$ – середнє значення послідовності $\Delta A_1, \Delta A_2, ..., \Delta A_K$, $\langle \Delta A^2 \rangle$ – послідовність $\Delta A_1^2, \Delta A_2^2, ..., \Delta A_K^2$, K – кількість появ слова A у тексті.

Ряди з цифрових значень, відповідних словам, перетворяться в графи горизонтальної видимості, в яких вузлам відповідають не лише цифрові значення, але й самі слова, що виражають певне змістовне значення. Мережа мови з використанням алгоритму горизонтальної видимості будується в три етапи. На першому на горизонтальній осі відзначається ряд вузлів, кожен з яких відповідає словам в порядку появи в тексті, а по вертикальній осі відкладаються вагові чисельні оцінки (візуально – набір вертикальних ліній, див. рис. 39).

---

[41] Ланде Д.В. Елементи комп'ютерної лінгвістики в правовій інформатиці. - К.: НДІІП НАПрН України, 2014. - 168 с. ISBN 978-966-2344-33-2

На другому етапі будується традиційний граф горизонтальної видимості. Для цього між вузлами існує зв'язок, якщо вони знаходяться в "прямій видимості", тобто якщо їх можна з'єднати горизонтальною лінією, що не перетинає ніяку іншу вертикальну лінію. Цей (геометричний) критерій можна записати таким чином: два вузли (слова), наприклад, B_3^n і C_7^m $(m = n+5)$ поєднуються зв'язком, якщо (див. рис. 39) $\sigma_n, \sigma_m > \sigma_p$ для усіх $n < p < m$.

Алгоритм побудови можна представити зручним для обчислення способом. Так наприклад, на рис. 39 для вузла-слова A_1^{n+2} суміжними в мережі вважаються слова B_3^n та C_1^{n+5} і встановлюються ребра-зв'язки, такі що B_3^n – найближче зліва від A_1^{n+2} слово, з ваговою оцінкою $\sigma_n = \sigma_B$, що перевищує вагову оцінку слова A $\sigma_{n+2} = \sigma_A$, а $C_7\,(m = n+5)$ – найближче справа від A_1^{n+2} слово, для якого $\sigma_{105} > \sigma_{102}$.

На третьому, завершальному етапі, отримана на попередньому етапі мережа компактифікується. Усі вузли з цим словом, наприклад словом A, об'єднуються в один вузол. Усі зв'язки таких вузлів також об'єднуються. Важливо відмітити, що між будь-якими двома вузлами при цьому залишається не більш за один зв'язок – кратні зв'язки вилучаються. Зокрема це означає, що міра (число зв'язків) вузла не перевищує суми степенів $\sum_k A_k^n$. У результаті виходить нова мережа слів – компактифікований граф горизонтальної видимості (КГГВ), див. рис. 39, 40.

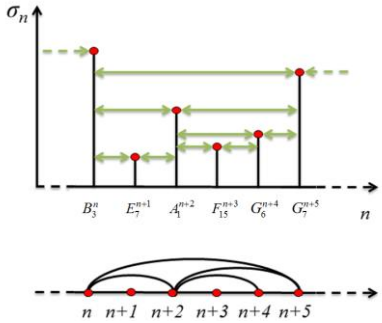


Рисунок 39 – Приклад побудови графу горизонтальної видимості

У якості текстів при побудові мереж слів розглядалася добірка законодавчих актів України, що відносяться до формування та розвитку інформаційного простору держави (Закони України «Про доступ до публічної інформації», «Про Основні засади розвитку інформаційного суспільства в Україні на 2007-2015 роки», «Про телекомунікації», «Про захист персональних даних», «Про основи національної безпеки України»).

Для усіх побудованих КГГВ-мереж слів було визначено розподіл степенів вузлів, який виявився близьким до статечного ($p(k) = Ck^{\alpha}$), тобто ці мережі є безмасштабними. Були проведені розрахунки параметрів мереж для усіх розглянутих літературних творів. В результаті виявилось, що для усіх з них коефіцієнт α змінювався в діапазоні від –1 до –0,95.

Якщо позначити через Ψ множину із N різних слів (розглядається випадок N = 100), що відповідають найбільш вагомим вузлам наведеної простої мережі мови, а Λ – множину слів, що відповідають найбільш вагомим вузлам КГГВ, то множина $\Omega = \Lambda \setminus \Psi$ відповідає інформативним словам, що мають, крім того, важливе значення і для зв'язності тексту. У Додатку приведені зіставлення 100 найбільш вагомих вузлів для трьох даних

типів мереж слів за текстами Законів України «Про телекомунікації» і «Про захист персональних даних».

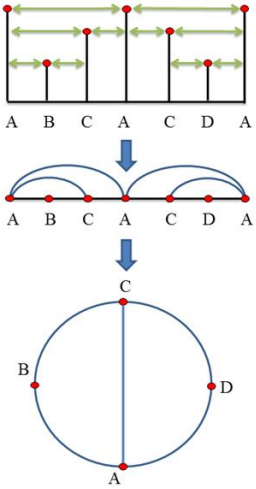


Рисунок 40 – Етапи побудови компактифікационного графа горизонтальної видимості

До складу вузлів з найбільшими ступенями в КГГВ-мережі, разом з особистими займенниками і іншими службовими словами (частки, прийменники, союзи і так далі), потрапили слова, що визначають інформаційну структуру тексту[42].

Для порівняння була додатково досліджена поведінка простих мереж мови, коли не першому етапі побудови мережі зв'язуються сусідні слова, що входять в текст (*L*-простор, *R* = 1), а на другому - відбувається компактифікація мережі. Очевидно, що вага вузлів в цій мережі відповідає частоті появи слів, а їх розподіл – закону Ципфа [*Zipf, 1949*]. При цьому найбільші ступені мають вузли, що відповідають словам з найбільшою частотою, – союзам, прийменникам, займенникам і тому подібне, що

[42] Giora R. Segmentation and Segment Cohesion: On the Thematic Organization of the Text // Text. An Interdisciplinary Journal for the Study of Discourse Amsterdam. – 3. – № 2. – P. 155-181 (1983)

мають велике значення для зв'язності тексту, але є малоцікавими з точки зору дослідження інформаційної структури.

У КГГВ-мережі по тексту Закону України «Про телекомунікації» з урахуванням значень TFIDF до складу множини Ω потрапили такі слова, як «Державне», «Регулювання», «Ринку», «Інтернет», «Провайдер», «Трафік». У КГГВ-мережі для цього ж тексту за ваговими значеннями слів, що відповіднають дисперсійним оцінкам, додатково до складу множини Ω потрапили такі слова, як «Суб'єкт», «Ресурс», «Переоформлення», «Рішення», «Споживачів» та інші.

При аналізі тексту Закону України «Про захист персональних даних» до множини Ω (для КГГВ-мережі з урахуванням вагових значень слів за алгоритмом TFIDF) потрапили такі слова, як «Інформація», «Відстрочення», «Орган», «Баз», «Виключено».

У КГГВ-мережі для тексту цього законодавчого акту за ваговими значеннями слів, відповідними дисперсійним оцінкам, до складу множини Ω потрапили додатково такі слова, як «Використання», «Прав», «Уповноважений», «Особа».

У результаті проведених досліджень мереж було отримано:

1. Запропонований алгоритм побудови компактифікованого графа горизонтальної видимості (КГГВ).

2. На основі послідовності дисперсійних оцінок слів тексту і КГГВ, побудовані мережі слів різних текстів.

3. Для літературних текстів серед вузлів відповідних КГГВ з найбільшими мірами присутні слова, не лише структури тексту, що забезпечують зв'язність, але й що визначають його інформаційну структуру, відбивають семантику літературних творів.

4. Алгоритм визначення ваги слів, що базується на дисперсійній оцінці виявився ефективнішим для визначення інформаційно-значущих слів, що грають важливе значення для структурної зв'язності в літературних текстах, ніж алгоритм TFIDF.

Питання для самоконтролю

1. Які алгоритми використовуються для перетворення часового ряду в граф?
2. Які характеристики динамічних систем можуть бути вивчені за допомогою графів?
3. Як PVG-алгоритм відрізняється від NVG та HVG алгоритмів?
4. Які важливі параметри алгоритму PVG можна змінювати? Як це впливає на результат?
5. Які критерії визначення зв'язків між вершинами у PVG графі?
6. Які особливості графів визначаються кутом зору в PVG-алгоритмі?
7. Які можуть бути практичні застосування цих алгоритмів? Наведіть конкретні приклади.
8. Які переваги та обмеження використання графів для аналізу часових рядів?
9. Як змінюється середня довжина зв'язку у графі видимості PVG при зміні кута зору? Які тенденції спостерігаються при збільшенні кута зору?
10. Яка характеристика графа видимості PVG залежить від кількості кластерів? Як змінюється кількість кластерів при збільшенні кута зору?
11. Яка основна мета використання опорних слів у тексті?
12. Які властивості опорних слів роблять їх інформаційно-значущими? Яким чином оцінюється дискримінантна сила слів?
13. Як побудова мереж мови (Language Network) допомагає виявляти структурні елементи тексту? Що може бути вузлами цих мереж, і як вони можуть бути сполучені?
14. Які підходи до побудови графів видимості базуються на текстових даних? Яку функцію можна використовувати для призначення числових вагових значень словам у тексті?

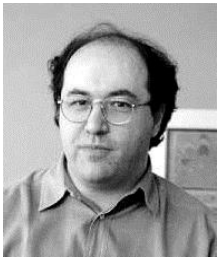
15. Як використовується підрахунок вагової оцінки для слів у тексті? Які параметри використовуються при розбитті тексту на фрагменти та підрахунку кількості входжень слів?
16. Як обчислюється дисперсійна оцінка ваги слів у тексті, і які параметри використовуються при цьому?
17. Які ряди числових значень, відповідних словам, можуть бути перетворені в графи горизонтальної видимості? Які основні етапи побудови таких графів?
18. Яким чином визначається "пряма видимість" між вузлами у графі горизонтальної видимості?
19. Які характеристики КГГВ-мереж слів було визначено після побудови? Яким чином алгоритм визначення ваги слів на основі дисперсійної оцінки відрізняється від алгоритму TFIDF, і чому він виявився ефективнішим для визначення інформаційно-значущих слів в літературних текстах?

7. Клітинні автомати

Дж. Фон. Нейман
(1903-1957)

Концепція клітинних автоматів була вперше запропонована більше півстоліття тому Дж. Фон Нейманом (J. Von Neumann)[43] і розвинута С. Вольфрамом (S. Wolfram) у фундаментальній монографії[44].

Клітинні автомати є корисними дискретними моделями на дослідження динамічних систем. Дискретність моделі, а точніше можливість представити модель у дискретній формі, може вважатися важливою перевагою, оскільки відкриває широкі можливості використання комп'ютерних технологій.



С. Вольфрам

Тривалий час клітинні автомати сприймалися як кумедна гра, яка не має практичної цінності. Але, у зв'язку з розвитком комп'ютерних технологій, вони починають швидко входити до арсеналу інструментальних засобів, які використовуються на практиці в різних галузях науки та техніки[45].

Клітинний автомат являє собою дискретну динамічну систему, сукупність однакових клітин, однаково сполучених між собою.

Усі клітини утворюють мережу (решітку) клітинних автоматів. Стан кожної клітини визначаються станом

---

[43] *Von Neumann, J. and A. W. Burks* (1966). Theory of self-reproducing automata. Urbana, University of Illinois Press.

[44] *Wolfram S. (*2002). A New Kind of Science. – Champaign, IL: Wolfram Media Inc.

[45] *Toffoli, Tommaso & Margolus, Norman* (1987), Cellular Automata Machines: A New Environment for Modeling, MIT Press.

клітин, що входять до її локального околу і називаються найближчими сусідами. Околом клітини з номером j – $O(j)$ називається множина його найближчих сусідів. Стан j-ої клітини на момент часу $t+1$ визначається деяким правилом F, яке можна описати, наприклад, мовою булевої алгебри:

$$y_j(t+1) = F\left(y_j, O(j), t\right). \qquad (3.9.1)$$

У багатьох випадках вважається, що сама клітина належить до своїх найближчих сусідів, тобто. $y_j \in O(j)$, у такому разі формула спрощується: $y_j(t+1) = F\left(O(j), t\right)$. Клітинні автомати задовольняють наступним правилам:

- зміна значень всіх клітин відбувається одночасно (одиниця виміру - такт);
- мережа клітинних автоматів однорідна, тобто. правила зміни станів всім клітин однакові;
- на клітину можуть вплинути лише клітини з її локального околу;
- множина станів клітини кінцева.

Окіл фон Неймана: Окіл Мура:

Рисунок 41 – Приклади околів фон Неймана та Мура

Клітинні автомати можуть мати будь-яку розмірність, проте найчастіше розглядають одновимірні та двовимірні системи клітинних автоматів.

У випадку двомірної решітки, елементами якої є квадрати, кожну клітину зручно задавати двома індексами – $y_{i,j}$. Найближчими сусідами, що входять до околу елемента $y_{i,j}$, є клітини, розташовані вгору-вниз і вліво-вправо від нього (так званий окіл фон Неймана: $O^N(y_{i,j}) = \left(y_{i-1,j}, y_{i,j-1}, y_{i,j}, y_{i,j+1}, y_{i+1,j}\right)$, можна додати і діагональні елементи – окіл Мура (G. Moore):

$$O^M(y_{i,j}) = \left(y_{i-1,j-1}, y_{i-1,j}, y_{i-1,j+1}, y_{i,j-1}, y_{i,j}, y_{i,j+1}, y_{i+1,j-1}, y_{i+1,j}, y_{i+1,j+1}\right).$$

У моделі Мура кожна клітка має вісім сусідів. Для усунення крайових ефектів, можна грати топологічно - «згортання у тор», тобто, перший рядок вважається продовженням останнього, а останній – попереднім першим – так звані граничні умови.

Це дозволяє визначати загальне співвідношення значення клітини на кроці $t+1$ у порівнянні з кроком t:

$$y_{i,j}(t+1) = (y_{i-1,j-1}(t), y_{i-1,j}(t), y_{i-1,j+1}(t), y_{i,j-1}(t),$$
$$y_{i,j}(t), y_{i,j+1}(t), y_{i+1,j-1}(t), y_{i+1,j}(t), y_{i+1,j+1}(t)).$$

Розглянемо один із прикладів використання клітинних автоматів – модель поширення інновацій[46] та її узагальнення – модель поширення новин. Модель дифузії (розповсюдження) інновацій функціонує за такими правилами: кожен індивід, який здатний прийняти інновацію, відповідає одній квадратній клітині на двовимірній площині.

При цьому: 1) кожна клітина може перебувати у двох станах: 1 – нововведення прийняте; 0 – нововведення не прийняте; 2) автомат, сприйнявши інновацію один раз, запам'ятовує її назавжди (стан 1, який може бути зміненим); 3)

[46] *Bhargava S.C., Kumar A., Mukherjee A.* A stochastic cellular automata model of innovation diffusion. Technological forecasting and social change, 1993. – Vol. 44. – № 1. – pp. 87-97

автомат схвалює рішення щодо прийняття новини, орієнтуючись на думку восьми найближчих сусідів, тобто якщо в околі цієї клітини (використовується окіл Мура) є m прихильників новинки, p – ймовірність її прийняття, (якщо $pm > R$ (R – фіксоване значення – поріг), то клітина набуває інновації (значення 1).

Клітинне моделювання дозволяє будувати значно реалістичніші моделі ринку інновацій, ніж традиційні підходи.

У моделі дифузії інформації передбачалося, що клітина може бути в одному із трьох станів: 1 – «свіжа новина» (клітина забарвлюється у чорний колір); 2 – новина, яка застаріла, але збережена у вигляді відомостей (сіра клітина); 3 – клітина не має інформації, переданої повідомленням новин (клітина біла, інформація не дійшла або вже забута). У моделі прийнято такі правила розповсюдження повідомлень:

- спочатку все поле складається з білих клітин за винятком однієї, чорної, яка першою «прийняла» новину (рис. 42 а);
- біла клітина може перефарбовуватися тільки в чорний колір або залишатися білою (вона може отримувати новину або залишатися "невідомою");
- біла клітина перефарбовується, якщо виконується умова, аналогічна до моделі дифузії інновацій: $pm > 1$ (ця умова модифікується для $m \le 2$: $1{,}5 \cdot pm > 1$);
- якщо клітина чорна, а навколо неї виключно чорні та сірі, то вона перефарбовується у сірі кольори (новина застаріває, але зберігається як відомості);
- якщо клітина сіра, а навколо неї виключно сірі та чорні, то вона перефарбовується у білий колір (відбувається старіння новини за її загальновідомості).

Описана система клітинних автоматів якісно відбиває процес поширення повідомлень серед окремих інформаційних джерел. З'ясувалося, що стан системи клітинних автоматів повністю стабілізується за обмежену кількість ходів, тобто процес еволюції виявився схожим. Приклад роботи моделі наведено на рис. 42.

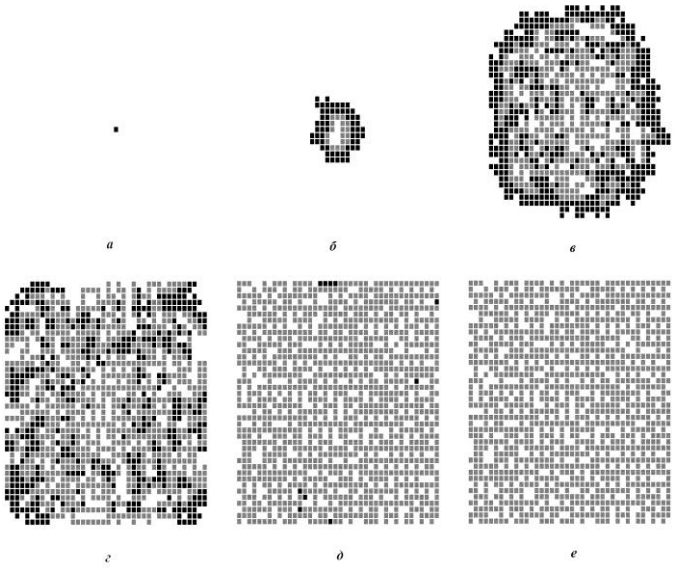


Рисунок 42 – Процес еволюції системи клітинних автоматів «дифузії новин»: а – вихідний стан; б-д – проміжні стани; е – кінцевий стан

Типові залежності кількості клітин, які у різних станах залежно від кроку ітерації наведено на рис. 43. При цьому, очевидно, що сумарна кількість клітин, які знаходяться у всіх трьох станах на кожному кроці ітерації, є постійною і дорівнює кількості клітин, а при стабілізації системи клітинних автоматів співвідношення сірих, білих і чорних клітин приблизно становить: 3:1:0.

Детальний аналіз отриманих залежностей дозволив провести аналогію даної моделі «дифузії інформації» з деякими аналітичними міркуваннями. Результати

моделювання дають підстави припустити, що еволюція сірих клітин описується деякою безперервною функцією:

$$x_g = f\left(t, \tau_g, \gamma_g\right),$$

де t – час (крок еволюції), τ_g – зсув часу, що забезпечує отримання необхідного фрагмента аналітичної функції, γ_g – параметр крутості цієї функції.

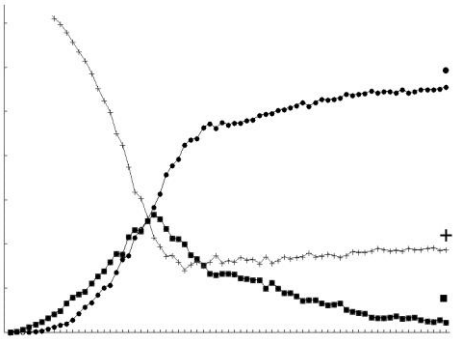


Рисунок 43 – Кількість клітин кожного із кольорів залежно від кроку еволюції: білі клітини – (+); сірі клітини – (•); чорні клітини – (■)

Відповідно, динаміка білих клітин x_w (кількість клітин у момент t) кількість клітин на момент x_g з аналогічними параметрами:

$$x_w = 1 - f\left(t, \tau_w, \gamma_w\right).$$

Оскільки, як було зазначено вище, завжди виконується умова балансу, тобто загальна кількість клітин у будь-який момент часу є завжди постійною, то умову нормування можна записати так:

$$x_g + x_w + x_b = 1,$$

де x_b – кількість чорних клітин у момент часу t.

Таким чином, отримуємо:

$$x_b = 1 - x_g - x_w = f\left(t, \tau_w, \gamma_w\right) - f\left(t, \tau_g, \gamma_g\right).$$

Вигляд залежності на рис. 44 дозволяє припустити, що в якості функції $f\left(t, \tau, \gamma\right)$ може бути вибраний наступний вираз (логістична функція):

$$f\left(t, \tau, \gamma\right) = \frac{C}{1 + e^{\gamma(t-\tau)}},$$

де C – деяка нормуюча константа.

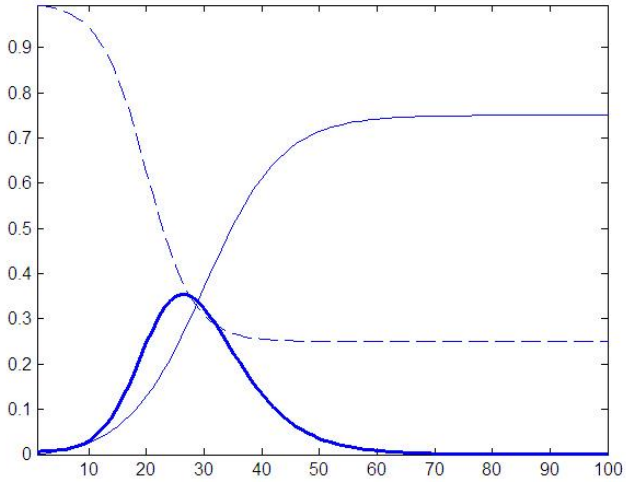


Рисунок 44 – Безперервні залежності, отримані внаслідок аналітичного моделювання, залежно від кроку еволюції: суцільна лінія – сірі (x_g); пунктирна лінія – білі (x_w);суцільна жирна лінія – чорні (x_b)

На рис. 44 наведено графіки залежності x_g, x_w, x_b від кроку еволюції системи клітинних автоматів, отримані внаслідок аналітичного моделювання.

Слід зазначити, що залежність дифузії новин, яка отримана в результаті моделювання, добре узгоджується з «життєвою» поведінкою тематичних інформаційних

потоків в інтернет-джерелах (веб-сайтах), а на локальних проміжках часу – з традиційними моделями.

Питання для самоконтролю

1. Які можливості використання комп'ютерних технологій відкриває дискретність моделей, зокрема клітинних автоматів?
2. Яким чином клітинний автомат можна описати як дискретну динамічну систему? Які клітини утворюють мережу (решітку) клітинних автоматів?
3. Яким чином визначається стан кожної клітини в клітинному автоматі? Що називається найближчими сусідами клітини?
4. Які основні правила властиві клітинним автоматам? Опишіть їх характеристики.
5. Яка різниця між околом фон Неймана і околом Мура в клітинних автоматах? Які сусіди входять до околу фон Неймана? Які сусіди входять до околу Мура?
6. Які можливі розмірності клітинних автоматів, і які найбільш поширені типи розмірності?
7. Яким чином уникнути крайових ефектів при роботі з клітинними автоматами у двовимірному просторі? Що таке граничні умови у цьому випадку?
8. Які основні правила функціонування моделі дифузії інновацій? Які два стани може мати кожна клітина в моделі дифузії інновацій?
9. Які три стани може мати клітина в моделі поширення інформації? Які умови перефарбовування клітини в сірий колір встановлені для моделі поширення інформації?
10. Які спостереження підтверджують відповідність результатів моделювання в моделі поширення інформації реальній поведінці тематичних інформаційних потоків у веб-сайтах?

8. Перколяція

Добре вивченими та важливими у практичному відношенні є перколяційні мережі[47].

Розглянемо одну з найпростіших постановок перколяційної задачі. Нехай існує квадратна сітка (нескінченна), кожен зв'язок якої має опір. Такий зв'язок для зручності називатимемо чорним. Випадковим чином чорні (провідні) зв'язки розриваються. Можна говорити, що при цьому чорні зв'язки замінюються на білі з опором $r_2 = \infty$. Завдання – необхідно знайти таку концентрацію чорних зв'язків p_c при і вище якої існує зв'язкова частина чорних зв'язків, якою можна дістатися з однієї нескінченності в іншу, без перестрибування через білий зв'язок. Така, зв'язкова частина, що тягнеться на нескінченність називається нескінченним кластером. Звичайно, реальна сітка завжди кінцева, тому, в даному випадку, передбачається, що її розмір набагато більший за, так звану, кореляційну довжину. У цьому випадку різні реалізації структури чорних зв'язків, що отримуються при випадковому вирізанні, мають одні й ті самі властивості. Імовірність спеціальних вироджених розподілів чорних зв'язків зневажливо мала. Тут має місце аналогія з випадковим розподілом молекул газу у певному об’ємі. Імовірність того, що в одній половині об’єму збереться весь газ, настільки малоймовірна, що ніколи не береться до уваги. У той же час, якщо молекул всього дві, то ця ймовірність дорівнює $1/4$.

Окрім геометричної постановки задачі перколяції – виникнення нескінченного чорного кластеру, можна запропонувати і фізичну постановку задачі, наприклад,

[47] Dietrich Stauffer, Ammon Aharony. Introduction To Percolation Theory. CRC Press, 2018. – 192 p.

про перебіг струму по чорних зв'язках. Чорні зв'язки проводять струм, білі - ні. Необхідно визначити опір (провідність) сітки загалом.

При $p > p_c$ провідність усієї сітки в цілому – G не дорівнює нулю (струм знаходить свій шлях від одного контакту на «нескінченності» до іншого по чорному нескінченному кластеру). При $p < p_c$ і $G = 0$, відповідно, опір усієї сітки: $R = 1/G = \infty$.

Звичайно, теоретично не обов'язково розглядати саме двовимірну квадратну сітку. Можливі будь-які розмірності та типи сіток (однорідних у середньому). Крім того, можна говорити не тільки про визначення зв'язків, а й вузлів, коли всі зв'язки провідні, а провідні (чорні) вузли випадково з даною ймовірністю вирізані.

Як виявилося, задача протікання, що виникла при формулюванні прикладної інженерної задачі про протікання газу або рідини через пористий фільтр, є одним з найпростіших прикладів теорії фазових переходів другого роду і критичних явищ. Так, багато характеристик, що описують геометричні та фізичні властивості поблизу порогу протікання поводяться універсальним чином, описуються критичними індексами, чисельне значення яких не залежить від виду сітки.

Розглянемо деякі геометричні характеристики перколяційної сітки. Таких геометричних показників багато – середня кількість кластерів, розподіл кластерів за розмірами, середній розмір кластера, потужність нескінченного кластера, властивості різних частин нескінченного кластера скелета, скелетона, мертвих кінців,…) тощо. Тут ми розглянемо лише деякі характеристики. Перша з них – це $n_s\ p$ – розподіл кінцевих кластерів за величиною, тобто кількість кластерів з s вузлів (зв'язків) що припадають на один вузол (зв'язок) решітки. Друга характеристика вдало підходить до ролі параметра порядку $P\ p$ – потужність нескінченного

кластера, ймовірність того, що довільний вузол (зв'язок) належить нескінченному кластеру. Потужність нескінченного кластера – $P\ p$ виражається через $n_s\ p$. Для цього достатньо врахувати, що можливість потрапити на чорний вузол – p є сума ймовірності потрапити на нескінченний кластер $P\ p$ або на будь-який скінченний: $\sum_s sn_s(p)$:

$$P(p)+\sum_s sn_s(p)=p,$$

звідки:

$$P(p)=p-\sum_s sn_s(p).$$

Поблизу порога протікання p_c потужність нескінченного кластера поводиться аналогічно параметру порядку теорії фазових переходів другого роду:

$$P(p)\sim(p-p_c)^b,(p-p_c)/\ p_c.$$

Роль температури T і критичної температури T_c у фазових переходах тепер відіграють концентрація p добре провідних зв'язків/вузлів (чорна фаза) та поріг протікання – p_c.

Близькість до порогу протікання будемо позначати:

$$\tau=\frac{p-p_c}{p_c}.$$

Розглянуту аналогію між теорією фазових переходів і теорією перебігу можна поглибити, увівши в теорію перебігу аналог безрозмірного магнітного поля h. У геометричних характеристиках перколяційних систем, що розглядаються тут, це робиться досить майстерним і штучним чином. Вводиться, так званий, демон Кастеляйна-Фортуїна – чорний вузол поза ґратами, пов'язаний з кожним чорним вузлом з ймовірністю

$1-\exp\ -h$. Аналог вільної енергії в теорії фазових переходів може бути записаний як:

$$G(t,h)=\sum_s n_s e^{-hs},$$

де $e^{-hs}=\ e^{-h\ \ s}$ – частка кінцевих кластерів із вузлів, у яких жоден із вузлів не пов'язаний з демоном Кастеляйна-Фортуїна.

Параметр порядку в теорії фазових переходів можна знайти з G як похідна по полю h, при $h=0$

$$-\left.\frac{d(G)}{dh}\right|_{h=0}=-\eta,$$

і знаходимо:

$$\left.\frac{dG}{dh}\right|_{h=0}=-\sum_s s n_s e^{-hs}\Big|_{h=0}=-\sum_s s n_s,$$

що дає головну (сингулярну) частину $P\ \ p$:

$$P(p)=p-\sum_s s n_s=p-\left.\frac{dG}{dh}\right|_{h=0}.$$

При нульовому полі $h=0$ Параметр порядку $P\ \ p$ нижче порогу протікання $p<p_c$ дорівнює нулю, що цілком аналогічно ситуації в теорії фазових переходів - при $T>T_c$ феромагнітний стан (намагніченість $m\neq 0$) переходить в парамагнітний ($m=0$). При $h\neq 0$ і $T>T_c$ є ненульовий параметр порядку «зобов'язаний» зовнішньому магнітному полю – h і тому пропорційний йому.

Легко побачити, що запровадження демона Кастеляйна-Фортуїна також залишає параметр порядку теорії

протікання – $P\ p$ не рівним нулю, нижче порога протікання, тобто і при $p < p_c$ існує нескінченний кластер, величина якого пропорційна h. Насправді, при $p < p_c$ формально немає нескінченного чорного кластера, але при $h \neq 0\,(h << 1)$ кожен чорний вузол пов'язаний з іншим через демона Кастеляйна-Фортуїна з ймовірністю

$$1 - e^{-h} \approx 1 - 1 + h = h,$$

пропорційною полю.

Що означає існування нескінченного кластера ($P(p) \neq 0$) пропорційного h:

$$P\ \ p < p_c\ \ \sim h.$$

Звернемося тепер до фізичних характеристик теоретичного перебігу, що дозволяє набагато наочнішим чином пояснити основні закономірності фазових переходів. Тепер вважатимемо, що чорні зв'язки в перколяційній мережі мають опір r_1, а розірвані (білі) – $r_2 = \infty$. При розмірах сітки $L >> \xi$ набагато більше, так званого, кореляційного радіусу ξ, вплив конкретного випадкового розподілу чорних і білих зв'язків (випадкова реалізація структури) стає несуттєвою і добре визначеною величиною - повним опором R. Для того, щоб абстрагуватися від конкретного розміру сітки (L) зручно перейти від опору всього зразка (решітки) до питомої ефективної провідності – σ_e:

$$R = \frac{1}{\sigma_e} \frac{L}{L^{d-2}},$$

де $d = 2,3...$ розмірність сітки.

За визначенням на розмірах порядку та більшого кореляційного радіусу всі властивості сітки, в цілому (в даному випадку питома ефективна провідність,) однакові та відповідно мають бути однакові й основні, головні елементи їхньої структури.

Як параметр порядку, що описує фазовий перехід, зручно ввести величину пропорційну ефективній провідності – σ_e. Як і для параметра порядку P p ефективна провідність σ_e при $p > p_c$ не дорівнює нулю, а нижче за поріг при $r_2 = \infty$ дорівнює σ_e $p < p_c$ $= 0$. Таку поведінку легко пояснити: вище порогу протікання струм може протікати від одного контакту до іншого (які можуть бути формально рознесені і на нескінченну відстань), проходячи по чорних провідних зв'язках.

Це означає, що існує нескінченний чорний кластер. При $p < p_c$ існують лише кінцеві кластери, з розміром менше кореляційної довжини, вони ізольовані один від одного тому, що $r_2 = \infty$ і тому струм через сітку пройти не може. Таким чином, при $r_2 = \infty$ ефективна провідність дорівнює нулю: σ_e $p < p_c$ $= 0$. Якщо r_2 велике, але скінченне, тобто $h = r_1 / r_2 << 1$, але не нуль, то струм зможе протікати від одного кінцевого кластера до іншого. При цьому, звичайно ж, провідність сітки буде пропорційна h σ_e $p < p_c$ $\sim h$ і при $h \to 0$ ефективній провідності $s_e(p < p_c) \to 0.$

Таким чином, немає необхідності вводити демон Кастеляйна-Фортуїна. Його роль відіграють білі опори з великим, але кінцевим опором. Роль зовнішнього поля тепер відіграє $h = r_1 / r_2 << 1.$

Критична поведінка поблизу порога протікання має не тільки щільність нескінченного кластера $P(p)$, але й

множина інших важливих характеристик перколяційної мережі, наприклад, кореляційна довжина, яка розходиться при наближенні до p_c:

$$\xi \sim \left(p - p_c\right)^{-\nu} ,$$

де ν – критичний індекс кореляційної довжини.

Критичним чином поводиться і питома ефективна провідність p_e. Поблизу порога протікання вище $\left(p > p_c\right)$ та нижче $\left(p < p_c\right)$ має місце:

$$\rho_e = \begin{cases} \rho_1 \tau^{-t}, & p > p_c, \\ \rho_1 h \tau^q \equiv \rho_2 \tau^q, & p < p_c. \end{cases}$$

Аналогія між фазовим переходом другого роду і перколяційним переходом проявляється тут у тому, що якщо критична температура – T_c і пороги протікання – p_c для кожного матеріалу, або, відповідно, решітка має своє чисельне значення, то критичні індекси є універсальними, та залежать тільки від розмірності задачі, але не залежать від типу решітки.

Розглянемо питання застосування ренорм-групового методу для обчислення критичних індексів.

Поблизу порогу протікання структура зв'язкових частин перколяційної мережі (нескінченний кластер, при $p > p_c$ і «граткові звірі» при $p < p_c$) мають фрактальну структуру, тобто. є статистично самоподібними, Таким чином, переходячи від одного масштабу до іншого і вимагаючи масштабної інваріантності, можна отримати наближене значення критичних індексів. Нижче ми розглянемо кілька прикладів використання методу ренорм-групи для обчислення порогів перебігу та критичного індексу кореляційної довжини ν.

Приклад 1. Поріг протікання трикутних грат, визначення вузлів

Для зручності будемо міркувати в термінах протікання струму – провідний вузол (чорний) проводить струм, непровідний вузол (білий) - не проводить зі всіма зв'язками. На рис. 44 – трикутні грати з позначенням провідних (чорних) та непровідних (білих) вузлів, з розміром трикутного осередку рівним b; б – ренормовані решітки, трикутні осередки (позначені сірим) представляють тепер нові вузли, зв'язки між якими позначені жирними чорними лініями. Нові вузли утворюють нову (ренормовану) трикутну решітку з розміром комірки $b' = \sqrt{3} \cdot b$.

Правила перетворення чорних вузлів наступні – сірий трикутник решітки переходить у чорний вузол ренормованої решітки, якщо у нього 2 або 3 вершини (вузли) чорні, інакше він перетворюється на білий, непровідний вузол.

Імовірність чорного вузла в трикутній решітці дорівнює p, тому ймовірність зустріти в новій ренормовані решітці провідний, чорний вузол дорівнює $p^3 + 3p^2(1-p)$, де перший доданок «обов'язаний» своїм походженням сірому трикутнику з трьома чорними вузлами, а другий - з двома. У наслідок того, що розташування чорних і білих вузлів у другому випадку можливе трьома різними способами, другий доданок має співмножник 3. Таким чином, ймовірність зустріти чорний вузол у ренормованій решітці p' дорівнює:

$$p' = p^3 + 3 \cdot p^2 (1-p).$$

Мережа, що знаходиться на порозі перебігу при ренорм-груповому перетворенні залишається на порозі:

$$p'(p_c) = p_c,$$

тобто p_c є нерухомою точкою перетворення. Тоді з рівняння, що зв'язує між собою p' і p отримуємо:

$$p_c = p_c^3 + 3 \cdot p_c^2 (1 - p_c).$$

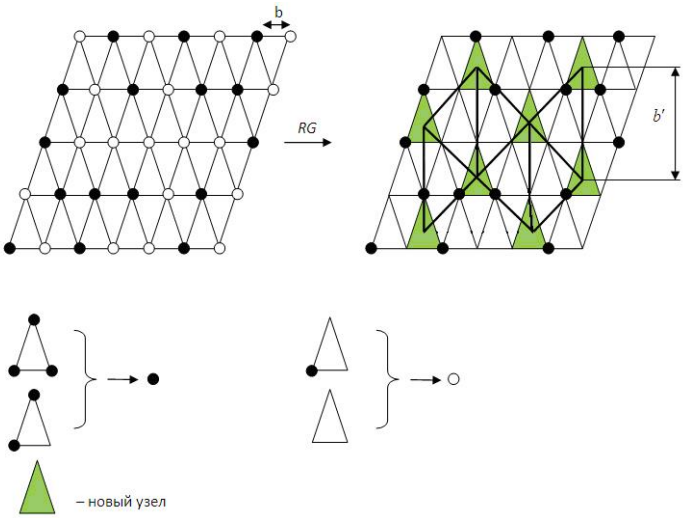


Рисунок 45 – Схематичне зображення кластерів чорної (добре проводить фази)

Це рівняння має три розв'язки: $p_c = 1$, $p_c = 0$, $p_c = 1/2$. Перші два з них тривіальні – повністю «біла» або «чорна» решітка залишається такою. Третє рішення

$$p_c = \frac{1}{2},$$

і є потрібний поріг протікання трикутної решітки для завдання вузлів.

У цьому прикладі, на відміну від кількох інших, все складається настільки вдало, що отриманий ренорм-груповим методом вираз для порогу протікання збігається з точним значенням.

Покажемо тепер, як можна виразити критичний індекс кореляційної довжини – ν.

Нехай у деякій решітці $\xi = \xi_0 \left| p - p_c \right|^{-\nu}$, тоді в ренормованій $\xi' = \xi_0 \left| p' - p_c \right|^{-\nu}$ і $\xi' = \xi / a$, де $a = b' / b$. Таким чином:

$$a \left| p' - p_c \right|^{-\nu} = \left| p - p_c \right|^{-\nu},$$

звідки для критичного індексу знаходимо:

$$\frac{1}{\nu} = \frac{\ln \dfrac{p' - p_c}{p - p_c}}{\ln a} \equiv \frac{\ln \lambda}{\ln a}, \qquad \lambda = \frac{p' - p_c}{p - p_c}.$$

Спрямовуючи концентрацію до порогової p_c для λ можна записати:

$$\lambda = \lim_{p \to p_c} \frac{p' - p_c}{p - p_c} = \left. \frac{dp'}{dp} \right|_{p = p' = p_c},$$

і для критичного індексу виходить простий вираз:

$$\frac{1}{\nu} = \frac{\ln \left(dp' / dp \big|_{p_c} \right)}{\ln a}.$$

Раніше, для трикутних решіток було отримано $p' = p^3 + 3 \cdot p^2 (1 - p)$ і $p_c = 1/2$, звідки:

$$\nu = \frac{\ln a}{\ln b} = \frac{1}{2} \frac{\ln 3}{\ln (3/2)} \approx 1.355.$$

Точне значення (яке можна отримати для цієї решітки) $\nu = 4/3 \approx 1.33$, і т.ч. застосування методу ренорм-групи дає дуже гарне наближення.

Приклад 2. Квадратні грати, визначення зв'язків

Одна з труднощів застосування ренорм-групового методу є визначення осередку, що ренормалізується, і вимог протікання до нього. У трикутній решітці цієї труднощі не було, осередок вибирався трикутним і протікання визначалося наявністю двох і більше чорних вузлів. У випадку квадратних ґрат потрібно більш акуратне міркування. Під перколюючим кластером ми розуміємо як кластер, що з'єднує «верх і низ», і кластер, що з'єднує «ліво і право». Тому в якості типу протікання для осередку виберемо, наприклад, як протікання «зліва направо». Тоді як осередок квадратної решітки можна вибрати конфігурацію, зображену на рис. 46.

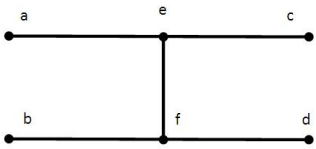


Рисунок 46 – Комірка квадратної решітки

Комірка квадратної решітки, для дослідження протікання «зліва-направо»:. a та b – ліві, c і d – праві контакти. Кожен із зв'язків – ae, ec,... є провідним з ймовірністю p .

При ренормалізації комірка перетворюється на один зв'язок:

Відповідно, $a = b'/b = 2$.

Нижче зображено, із зазначенням ймовірності протікання, всі конфігурації комірки, що проводять «зліва направо», пунктиром вказані розірвані (викинуті) зв'язки (див. рис. 47).

Таким чином, можливість отримати ренормовану конфігурацію, що забезпечує протікання, дорівнює:

$$p' = p^5 + 5p^4(1-p) + 8p^3(1-p)^2 + 2p^2(1-p)^3 .$$

Підставляючи у праву та ліву частини $p' = p = p_c$, рішення отриманого рівняння, отримаємо:

$$p_c = \frac{1}{2} .$$

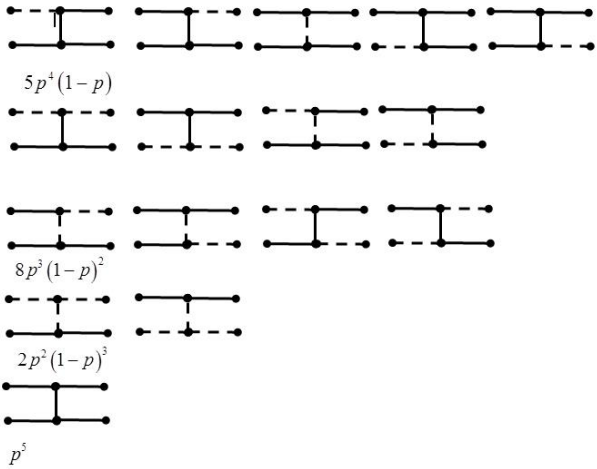


Рисунок 47 – Провідні конфігурації

Відразу ж можна вирахувати і критичний індекс кореляційної довжини:

$$\nu = \frac{\ln b}{\ln \lambda}, \quad \lambda = \left.\frac{dp'}{dp}\right|_{p=p_c} ,$$

беручи похідну від $p' = p'(p)$, по p і підставляючи $p = p_c = 1/2$ знаходимо:

$$\left.\frac{dp'}{dp}\right|_{p=p_c} = \left(10p^4 - 20p^3 + 6p^2 + 4p\right)\Big|_{p=p_c=1/2} = \frac{13}{8} .$$

Таким чином:

$$\nu = \frac{\ln 2}{\ln\left(13/8\right)} \approx 1.43$$

при точному значенні $\nu = 4/3$.

Далеко не завжди метод ренормгрупування дає такі добрі результати. Для уточнення необхідно брати великі розміри ренормуючої комірки, наприклад, замість комірки з $b = 2$ брати комірку з $b = 5$.

На перший погляд, таке уточнення не становить значного ускладнення. Однак, насправді, це не просто значне, а принципово значне ускладнення, тому що замість 32-х для випадку $b = 2$ комбінацій з яких 20 таких, що забезпечується протікання, у разі $b = 5$ число комбінацій дорівнює $2^{25} \approx 3 \cdot 10^7$, з яких для початку необхідно вибрати такі, що забезпечується протікання, а потім ще й знайти ймовірність їх появи.

Питання для самоконтролю

1. Яка постановка задачі перколяції є важливою в практичному аспекті?
2. Які властивості має дана квадратна сітка у контексті перколяційної задачі?
3. Що таке перколяція і як вона пов'язана з теорією ймовірностей та геометрією?
4. Які фізичні чи прикладні явища можуть бути описані за допомогою моделей перколяції? Наведіть приклади.
5. Яка роль порогової ймовірності в теорії перколяції, і як вона впливає на виникнення перколяційних кластерів?
6. Які методи чисельного моделювання використовуються для вивчення перколяції, та які характеристики системи вони дозволяють оцінити?
7. Що таке гранична ймовірність у контексті перколяції?
8. Як пов'язані гранична ймовірність та поява перколяційних кластерів?

9. Що таке нескінченний кластер у контексті задачі перколяції?
10. Які можливі розмірності та типи сіток в теорії перколяції?
11. Яка аналогія існує між задачею перколяції та випадковим розподілом молекул газу?
12. Які характеристики геометрії перколяційної сітки можна вивчати?
13. Як визначається характеристика в контексті перколяції?
14. Яким чином можна виразити потужність нескінченного кластера через параметр порядку та інші ймовірності?
15. Які ролі відіграють температура та критична температура у теорії перколяції?
16. Як можна використовувати аналогію з теорією фазових переходів у теорії перколяції?
17. Як визначається параметр порядку в теорії перколяції?
18. Які характеристики перколяційної мережі можна вивчати з фізичного погляду, і які з них будуть важливими для вивчення близькості до порогу протікання?
19. Які зв'язки вводяться для опису опорів чорних та білих зв'язків в перколяційній мережі?
20. Як визначається ефективна провідність у перколяційній мережі, і яка роль цієї величини?
21. Які важливі характеристики перколяційної мережі демонструють критичну поведінку навколо порогу протікання?
22. Які спільні риси проявляються між фазовим переходом другого роду і перколяційним переходом?
23. Як використовується ренорм-груповий метод для обчислення критичних індексів в перколяційних системах?
24. Яку фрактальну структуру мають зв'язкові частини перколяційної мережі близько порогу протікання?
25. В який спосіб здійснюється перехід від одного масштабу до іншого при застосуванні методу ренорм-групи для обчислення критичних індексів?
26. Які правила перетворення чорних вузлів використовуються в прикладі з трикутними гратами?
27. Як визначається імовірність зустріти чорний вузол у ренормованій решітці після перетворень?
28. Які величини зберігаються при ренорм-груповому перетворенні для мережі на порозі перебігу?

9. Інструменти аналізу і візуалізації мереж

На цей час існує досить багато програм, які забезпечують візуалізацію невеликих графів[48]. Серед таких програм – uDraw (Graph), розроблена в Бременському університеті (див. рис. 48).

9.1 uDraw (Graph)

uDraw(Graph) – це програма, що забезпечує візуалізацію графів (http://www.informatik.uni-bremen.de/uDrawGraph/).

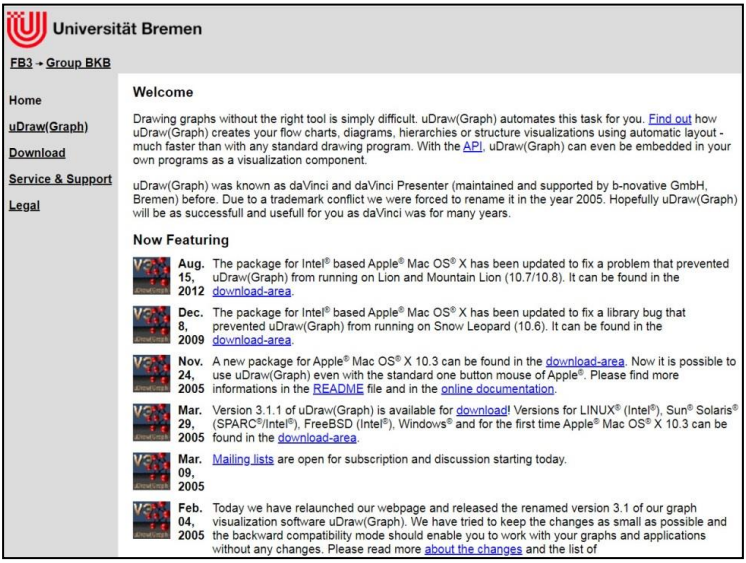


Рисунок 48 – вебсторінка uDraw(Graph)

[48] Ландэ Д.В., Субач І.Ю. В візуалізаціята аналіз мережевих структур : навчальний посібник. - Київ : КПІ ім. Ігоря Сікорського, Вид-во "Політехніка", 2021. - 80 с. ISBN 978-966-2577-14-3

На рис. 49 наведено приклад графу, створеного за допомогою програми uDraw (Graph).

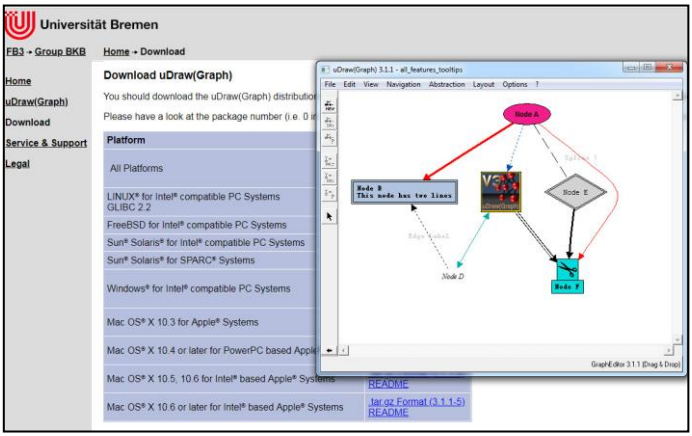


Рисунок 49 – Приклад графа, створеного за допомогою uDraw (Graph)

На рис. 50 приведена панель навігації, за допомогою якої здійсюється «переміщення» по вузлах графу. При побудові графу є можливість вибору фігур, якими позначаються вузли, змінювати кольори фону, формати вузлів і ребер.

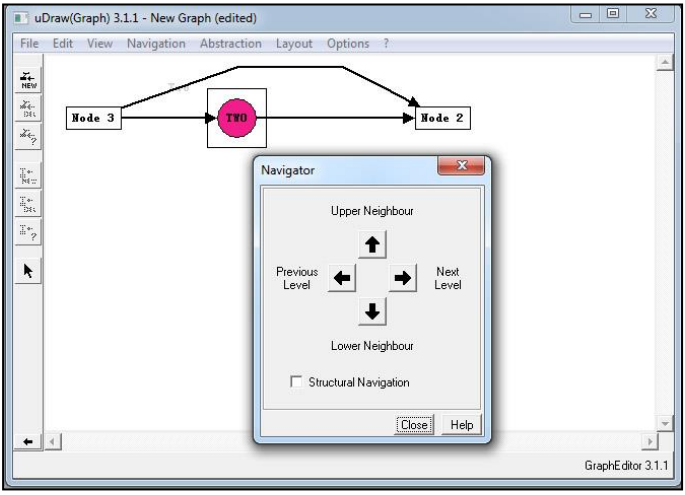


Рисунок 50 – Панель навігації uDraw (Graph)

Сформований граф можна зберігати у вигляді зображень у форматах GIF, TIF, JPEG, PNG (режим «Експорт», рис. 51).

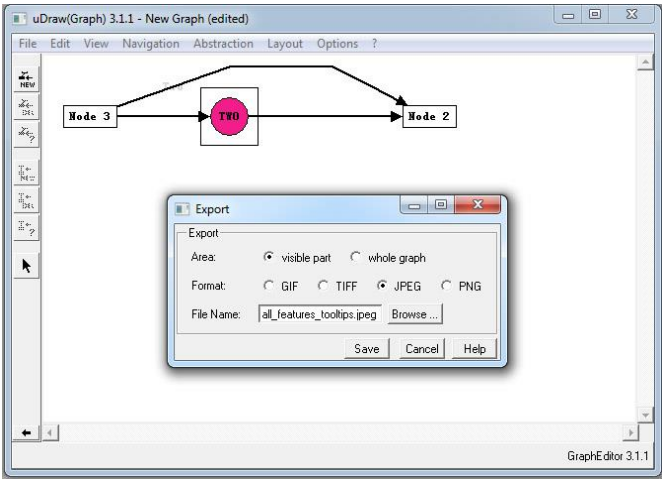


Рисунок 51 – Панель експорту графа, створеного за допомогою uDraw (Graph)

9.2 Social Network Visualizer

Одним з кращих програмних застосунків для візуалізації та аналізу мережевих структур на цей час є Social Networks Visualizer (SocNetV). Це крос-платформний програмний застосунок для аналізу та візуалізації соціальних мереж, розроблений мовою C++. SocNetV є вільним програмним забезпеченням, ліцензованим відповідно до GNU General Public License 3 (GPL3).

Вихідний код програми SocNetV, пакети та файли для Windows, Linux і MacOS доступні на веб-сайті http://socnetv.org.

У SocNetV є можливості ручного введення мережі або завантаження існуючого образу мережі, представленого у форматах GraphML, UCINET, Pajek і т.д., обчислення стандартних показників зв'язності мереж, таких як щільність, діаметр, геодезичні та відстані, зв'язність, ексцентриситет, коефіцієнт кластеризації, взаємність і т. д., значень центральності, застосування різних алгоритмів компонування, заснованих на центральності чи посередництві (Betweeness) вузлів чи динамічних моделей.

Базові можливості SocNetV:

- імпорт із мережевих форматів (*GraphML*, *Adjacency*, *Pajek*, *UCINET*, тощо);
- експорт у формати GraphML, Pajek, Adjacency;
- завантаження і редагування мереж;
- швидкий розрахунок індексів зв'язності, щільності, ступенів вузлів, ексцентриситету, коефіцієнту кластеризації тощо;
- розрахунок додаткових метрик для аналізу саме соціальних мереж, таких як індекси центральності і значущості;
- розрахунок *PageRank*;

- матричні обчислення: визначення матриць суміжності, Лапласа, цитування та ін.;
- аналіз структурної еквівалентності з використанням ієрархічної кластеризації, подібності акторів та відмінностей профілів зв'язків, коефіцієнтів Пірсона;
- наявність швидких алгоритмів виявлення спільнот, такі як знаходження тріад, клік тощо;
- розрахунок різних індексів центральності (центральність власного вектору і близькості, центральність проміжності, центральність інформації, центральність влади, близькість і престиж сторінки).
- можливість завантаження та редагування мультиреляційної мережі. Можна завантажити соціальну мережу, що складається з кількох відносин, або створити мережу самостійно і додати до неї кілька відносин;
- наявність datasets для аналізу соціальних мереж;
- наявність різних моделей компонування, заснованих або на індексах помітності (тобто кругових, рівневих і вузлових розмірах за показником центральності), або на силовому розміщенні (тобто Камада-Каваї, Фрухтерман-Рейнгольд і т. д.) для значної візуалізації соціальних мереж;
- створення випадкових мереж з використанням різних моделей генерації випадкових мереж (Barabási-Albert Scale-Free, Erdős-Rényi, Watts-Strogatz Small-World, d-regular, кільцеві грати тощо);
- наявність матричних підпрограм для розрахунку: графіка суміжності, матриці Лапласа, матриці ступенів і т. д.

- наявність вбудованого веб-сканера для автоматичного створення «соціальних мереж» із посилань, знайдених у заданій вихідній URL-адресі.
- наявність повної документації, доступної як в режимі онлайн, так і всередині додатку, в якій докладно пояснюється кожна функція та алгоритм SocNetV.

Програма Social Networks Visualizer може бути корисним інструментом для аналізу та візуалізації мереж понять, може допомогти візуалізувати та аналізувати зв'язки між окремими поняттями, що дозволяє спрощувати сприйняття інформації та виділяти ключові елементи мережі.

За допомогою програми SocNetV можна здійснювати:

- візуалізацію графа: SocNetV може відображати мережі понять у вигляді графу, де поняття представлені вузлами, а зв'язок між ними - ребрами. Це дозволяє легко побачити структуру мережі та взаємозв'язки між поняттями;
- аналіз зв'язків: програма дозволяє аналізувати рівень взаємозв'язків між поняттями, таких як частота зв'язків чи сила взаємодії між ними. Це допоможе визначити ключові поняття та їх вплив на всю мережу;
- фільтрування та розфарбування вузлів: SocNetV дозволяє фільтрувати вузли за різними параметрами, наприклад, частотою згадування, щоб сконцентруватися на найбільш значущих поняттях. Також можна розфарбовувати вузли за категоріями, що допоможе виділити різні групи понять;
- експорт даних: програма дозволяє експортувати візуалізовані мережі до різних форматів, таких як зображення або файли даних, що спростить подальше використання та обробку інформації.

9.3. Формат GraphML

Чимало програм візуалізації і аналізу графів використовують мову представлення даних GraphML як універсальну, що задовольняє специфікаціям XML.

Зокрема, в програмах SocNetV і Gephi наявні можливості завантаження вже існуючого, і, відповідно, вивантаження представлення мережевих структур у форматі GraphML.

Файл у форматі GraphML (GraphML-документа) призначений для визначення графу. Розглянемо граф показаний на рис. 52. Цей граф містить 11 вузлів і 12 ребер.

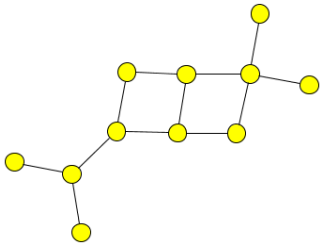


Рисунок 52 – Приклад графу

За допомогою *GraphML* цей простий граф описується таким чином:

```
   <?xml version="1.0" encoding="UTF-8"?>
   <graphml xmlns="http://graphml.graphdrawing.org/xmlns"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xsi:schemaLocation="http://graphml.graphdrawing.org/xmln
s

http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
     <graph id="G" edgedefault="undirected">
       <node id="n0"/>
       <node id="n1"/>
       <node id="n2"/>
       <node id="n3"/>
```

```
      <node id="n4"/>
      <node id="n5"/>
      <node id="n6"/>
      <node id="n7"/>
      <node id="n8"/>
      <node id="n9"/>
      <node id="n10"/>
      <edge source="n0" target="n2"/>
      <edge source="n1" target="n2"/>
      <edge source="n2" target="n3"/>
      <edge source="n3" target="n5"/>
      <edge source="n3" target="n4"/>
      <edge source="n4" target="n6"/>
      <edge source="n6" target="n5"/>
      <edge source="n5" target="n7"/>
      <edge source="n6" target="n8"/>
      <edge source="n8" target="n7"/>
      <edge source="n8" target="n9"/>
      <edge source="n8" target="n10"/>
    </graph>
  </graphml>
```

GraphML-документ складається із заголовку, елемента *graphml* і ряду часткових елементів: *graph*, *node*, *edge*. Розглянемо перераховані елементи детальніше, і покажемо, як саме вони визначають граф.

Заголовок із посиланням на *XML*-схему має вигляд:

```
  <?xml version="1.0" encoding="UTF-8"?>
  <graphml xmlns="http://graphml.graphdrawing.org/xmlns"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xsi:schemaLocation="http://graphml.graphdrawing.org/xmln
s

http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
    ...
  </graphml>
```

Перший рядок документу визначає те, що документ відноситься до стандарту *XML* 1.0 і наведений у кодуванні *UTF*-8.

Другий рядок містить кореневий елемент *GraphML*-документу: *graphml*. Елемент *graphml*, також як і всі інші

елементи мови *GraphML*, належить іменному простору http://graphml.graphdrawing.org/xmlns. Наступні два атрибути визначають стандартну *XML*-схему даного документу, розташовану на сервері graphdrawing.org. Атрибут *xmlns*: *xsi* = "http://www.w3.org/2001/XMLSchema-instance" – визначає *xsi*, як префікс іменного простору *XML*-схеми. Атрибут *xsi*: schemaLocation = "http://graphml.graphdrawing.org/xmlns визначає місцезнаходження *XML*-схеми для елементів простору імен *GraphML*.

Граф описується за допомогою тега (елементу) *graph*. Елементи, які розташовані всередині тега *graph*, забезпечують визначення вузлів і ребер. Вузол визначається за допомогою елемента *node*, а ребро за допомогою елемента *edge*.

Визначення графу:

```
<graph id="G" edgedefault="directed">
   <node id="n0"/>
   <node id="n1"/>
   ...
   <node id="n10"/>
   <edge source="n0" target="n2"/>
   <edge source="n1" target="n2"/>
   ...
   <edge source="n8" target="n10"/>
 </graph>
```

У *GraphML* порядок проходження елементів *node* і *edge* в описі не встановлено.

Граф у *GraphML* може містити спрямовані і неспрямовані ребра. Спрямованість ребер, що визначається за замовчуванням, задається *XML*-атрибутом *edgedefault* тега *graph*. Цей *XML*-атрибут може приймати одне з двох значень: *directed* і *undirected*.

Вузол у графі визначається за допомогою елемента *node*. Кожен вузол має унікальний (в межах даного документа) ідентифікатор, який задається за допомогою *XML*-атрибута

id. Графу може бути присвоєно ідентифікатор, якщо на даний граф потрібно організувати посилання.

Ребро у графі задається за допомогою елемента edge. Кожне ребро має дві кінцеві точки, що задаються за допомогою *XML*-атрибутів *source* і *target*. Значення атрибутів *source* і *target* повинні містити ідентифікатори вузлів, визначених у тому ж документі що і ребро. *XML*-атрибут *directed* визначає спрямованість ребра, задану в явному вигляді. Значення *true* задає спрямоване ребро, а *false* – неспрямоване. Якщо спрямованість в явному вигляді не задана, то застосовується спрямованість задана за замовчуванням при оголошенні графу.

Додатково, за допомогою *XML*-атрибуту *id*, може бути заданий ідентифікатор ребра, якщо необхідно організувати посилання на дане ребро.

Для оптимізації синтаксичного розбору документа використовуються спеціальні метадані, які можуть бути додані до деяких *GraphML*-елементів.

GraphML підтримує вкладені графи, тобто графи в яких вузли ієрархічно впорядковані. Ієрархія виражається через структуру *GraphML*-документа. Вузол в *GraphML*-документі може мати елемент *graph*, який містить вузли ієрархічно вкладені у даний вузол. Нижче наводиться приклад вкладеного графу і відповідний йому *GraphML*-документ (див. рис. 53).

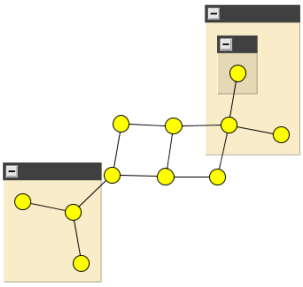


Рисунок 53 – Приклад вкладеного графу

Представленому графу відповідає *GraphML*-документ з вкладеними графами:

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml    xmlns="http://graphml.graphdrawing.org/xmlns"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://graphml.graphdrawing.org/xmln
s
http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <graph id="G" edgedefault="undirected">
    <node id="n0"/>
    <node id="n1"/>
    <node id="n2"/>
    <node id="n3"/>
    <node id="n4"/>
    <node id="n5">
        <graph id="n5:" edgedefault="undirected">
          <node id="n5::n0"/>
          <node id="n5::n1"/>
          <node id="n5::n2"/>
          <edge          id="e0"          source="n5::n0"
target="n5::n2"/>
          <edge          id="e1"          source="n5::n1"
target="n5::n2"/>
        </graph>
    </node>
    <node id="n6">
        <graph id="n6:" edgedefault="undirected">
          <node id="n6::n0">
              <graph                          id="n6::n0:"
edgedefault="undirected">
                <node id="n6::n0::n0"/>
               </graph>
          </node>
          <node id="n6::n1"/>
          <node id="n6::n2"/>
          <edge          id="e10"          source="n6::n1"
target="n6::n0::n0"/>
          <edge          id="e11"          source="n6::n1"
target="n6::n2"/>
        </graph>
    </node>
    <edge id="e2" source="n5::n2" target="n0"/>
    <edge id="e3" source="n0" target="n2"/>
    <edge id="e4" source="n0" target="n1"/>
    <edge id="e5" source="n1" target="n3"/>
    <edge id="e6" source="n3" target="n2"/>
    <edge id="e7" source="n2" target="n4"/>
    <edge id="e8" source="n3" target="n6::n1"/>
```

```
    <edge id="e9" source="n6::n1" target="n4"/>
  </graph>
</graphml>
```

GraphML підтримує поняття гіперребер. Гіперребро – це смислове об'єднання ребер яке не тільки пов'язує дві кінцеві точки, а й описує залежність між довільним числом кінцевих точок (наприклад, опис найкоротшого шляху – примітка перекладача). Гіперребра визначаються за допомогою елемента *hyperedge*. Кожній кінцевій точці яка входить у дане гіперребро відповідає елемент *endpoint*. Елемент *endpoint* повинен мати *XML*-атрибут *node*, який містить ідентифікатор вузла в документі.

Вузли можуть містити різні точки підключення ребер і гіперребер – так званні порти.

Порти вузла оголошуються за допомогою елементів port, які є дочірніми по відношенню до відповідного елементу node. Порти можуть бути вкладеними. Кожен елемент *port* має *XML*-атрибут *name*, який ідентифікує цей порт. Елемент *edge* має необов'язкові *XML*-атрибути *sourceport* і *targetport* які задають для ребра вихідний і вхідний порти вузла, відповідно. Аналогічно елемент *endpoint* має необов'язковий *XML*-атрибут *port*.

9.4 Gephi

9.4.1 Загальна інформація

Gephi (https://gephi.org/) – це найпопулярніша програма для персональних комп'ютерів щодо візуалізації та аналізу мереж і графів («мережевих графів»). Gephi забезпечує швидке компонування, ефективне фільтрування та інтерактивне дослідження даних, а також один з кращих варіантів для візуалізації великомасштабних мереж. Gephi – це мультиплатформне програмне забезпечення, яке поширюється з відкритим кодом згідно з ліцензіями CDDL 1.0 та GNU General Public License v3. За адресою https://gephi.org/ доступні версії

для Mac OS X, Windows та Linux вихідних кодів. Для роботи програми потрібна мова програмування Java 1.7+.

Розробники Gephi описують цю програму як "як Photoshop, але для даних".

Gephi дозволяє завантажувати дані мереж у форматах GEXF, GDF, GML, GraphML, Pajek (NET), GraphViz (DOT), CSV, UCINET (DL), Tulip (TPL), Netdraw (VNA) та таблиць Excel. Крім того, Gephi дозволяє експортувати дані мереж у форматах JSON, CSV, Pajek (NET), GUESS (GDF), Gephi (GEFX), GML та GraphML. Завдяки цьому Gephi може взаємодіяти з іншими системами аналізу і візуалізації графів.

Програма Gephi надає багато різних методів для укладання графів (розташування вузлів та зв'язків на площині) і дозволяє користувачеві налаштовувати кольори, розміри та мітки у графах. Вона є інтерактивним програмним забезпеченням, що надає інструменти для виявлення спільнот у мережах, а також дозволяє розраховувати найкоротші шляхи або відносну відстань від одного сайту до іншого.

Gephi підтримує плагіни, які дозволяють розширювати її функціональність та додавати нові алгоритми, макети та інструменти вимірювань. Завдяки багатопотоковій схемі обробки даних, Gephi дозволяє виконати кілька видів аналізу одночасно, що підвищує ефективність роботи з великими та складними графами.

Інтерфейс користувача системи Gephi включає три основні розділи (вікна):

- Data Laboratory: тут зберігаються всі вихідні дані про мережі та додаткові розрахункові значення.
- Overview: тут відбувається більша частина операцій користувачів, включаючи ручне редагування мереж, тестування макетів та встановлення фільтрів.

- "Попередній перегляд": тут уточнюється форма виведення графа, зазвичай, за допомогою набору інструментів графіка, граф доопрацьовується, у тому числі з естетичної точки зору. У цьому вікні реалізовано виклик експорту графа у форматах PDF, PNG і SVG.

Ці три основні розділи охоплюють множину вкладок, що дозволяють користувачеві реалізовувати окремі функції. Нижче розглядається кожне з основних вікон – розділів.

9.4.2 Data Laboratory

Data Laboratory в Gephi містить:

- усі вихідні дані про мережі, що були імпортовані до програми;
- додаткові розрахункові значення та метадані, пов'язані з мережевими даними;
- можливість перегляду, редагування та управління вузлами та зв'язками в мережі;
- інструменти для фільтрації та обробки даних, такі як видалення вузлів чи зв'язків за певними критеріями;
- можливість перегляду та редагування атрибутів вузлів та зв'язків, таких як мітки, кольори, розміри тощо;
- різні функції для роботи з даними, такі як сортування, пошук та угруповання вузлів та зв'язків.

Хоча Data Laboratory може мати вигляд електронної таблиці, її функціональність не слід плутати з Excel або Google Spreadsheet. Деякі операції з обробки даних можуть бути виконані саме тут, проте найкраще підготувати основні дані про мережу перед їх імпортом у Gephi. Для створення різних масивів більших обсягів найкраще використовувати інструменти електронних таблиць.

Аналогічно, значення полів, засновані на певній схемі сортування, краще створювати поза Gephi.

Однак це не означає, що дані, що зберігаються в лабораторії, є цілком статичними. Наприклад, усі статистичні обчислення та кластеризація автоматично додаватимуть нові значення для кожного вузла під час запуску процесу. Також є можливість додавати стовпці до таблиці, копіювати дані з одного стовпця до іншого, видаляти стовпці тощо.

Слід зазначити, що внесення масових змін на рівні вузлів або ребер може бути дуже трудомістким, особливо якщо досліджуваний набір даних про мережу складається з тисяч значень.

9.4.3 Overview

Всі дані про мережу спочатку проглядаються в розділі Overview, де Gephi надає початкове представлення досліджуваної мережі. Початковий вид мережі може бути простим, але потім із цим поданням проводиться спеціальна обробка. Всі функції, пов'язані з укладанням мережі, фільтрацією, сегментацією, забарвленням та будь-якими іншими налаштуваннями макета, видно насамперед саме в цьому вікні.

Вікно графіки прилягає до кількох панелей інструментів, кожна з яких містить багато функцій. Функціональність кожного з цих варіантів, зазвичай, інтуїтивно зрозуміла.

Розділ Overview у системі Gephi призначений для надання загального огляду та візуалізації графа, щоб користувач міг отримати уявлення про структуру мережі та взаємозв'язки між вузлами та зв'язками. Цей розділ пропонує різні можливості для більш глибокого розуміння даних про мережу та для прийняття рішень щодо подальших дій.

Основні можливості розділу Overview:

- візуалізація графа: розділ Overview надає можливість візуалізувати граф за допомогою різних методів відображення, таких як форсовані розкладки (Force-Directed Layout), радіальні розкладки (Radial Layout) та ін. Це дозволяє побачити структуру мережі, визначити основні групи вузлів та зв'язків, а також виявити особливості та патерни в даних;

- масштабування та навігація: Користувач може збільшувати або зменшувати масштаб графа та переміщатися по ньому, щоб детальніше розглянути окремі вузли чи зв'язки або отримати загальну картину мережі;

- інтерактивність: у розділі Overview користувач може взаємодіяти з графом, наприклад, вибирати вузли чи зв'язки, застосовувати виділені фільтри, переміщувати вузли для кращої видимості тощо;
- відображення атрибутів: користувач може налаштовувати відображення вузлів та зв'язків на основі їх атрибутів, таких як колір, розмір, форма тощо. Це допомагає виділити важливі властивості та різні групи даних;
- огляд статистичних показників: розділ Overview також може надати коротке зведення статистичних показників про граф, таких як кількість вузлів та зв'язків, середні значення атрибутів та ін.;
- панель управління: тут користувач може налаштовувати різні параметри візуалізації, а також вибирати різні алгоритми розкладки та макети для найкращого представлення графу.

Розділ Overview надає загальний огляд даних про мережу, допомагає візуалізувати граф для кращого розуміння його структури та дає можливість приймати поінформовані рішення щодо подальшого аналізу та візуалізації мережевих даних.

9.4.4 Preview

Розділ Preview (Попередній перегляд) у системі Gephi призначений для уточнення форми виведення графа за допомогою набору інструментів. У цьому розділі користувач може переглянути та відредагувати граф.

Вікно попереднього перегляду в Gephi дозволяє налаштувати атрибути, які були створені у вихідному вікні графіки. При цьому можна налаштувати мітки вузлів, вибрати шрифт, розмір, колір, контури і т.д.

Зовнішній вигляд вузла визначається параметрами ширини кордону, кольору кордону і прозорості. При цьому можна перемикатися у вікно Overview, щоб виконати низку налаштувань у Gephi, а потім повернутися у вікно попереднього перегляду та оновити відображення графу.

Для налаштування зовнішнього вигляду ребер передбачені такі опції, як налаштування товщини, кольору, прозорості, можливість кривих ребер та встановлення міток. Для орієнтованих ребер можна настроїти стрілки ребер.

Основні можливості розділу Preview у Gephi:

- візуалізація налаштування: користувач може налаштувати різні параметри візуалізації, такі як розмір та колір вузлів та зв'язків, ширина ліній, форма тощо. Це дозволяє створювати більш привабливі та інформативні візуалізації;
- розміщення елементів: користувач може переміщати вузли та зв'язки графа вручну, щоб наголосити на певних структурах або показати важливі взаємозв'язки;
- фільтрування та масштабування: користувач може застосовувати фільтри до графа для приховування неактуальних елементів та зосереджуватися на найважливіших вузлах та зв'язках. Також можливе

збільшення чи зменшення масштабу кращого сприйняття деталей графу;
- експорт: у розділі Preview користувач може зберегти графік у різних форматах, таких як PNG, PDF, SVG та інших, щоб поділитися візуалізацією з іншими або використовувати в документації та презентаціях;
- розширені настройки: користувач може налаштовувати додаткові параметри візуалізації, такі як відображення тегів, включення легенди, додавання фонового зображення тощо;
- налаштування візуалізації: користувач може проводити різні експерименти з візуалізацією, щоб знайти найбільш зручний та зрозумілий спосіб подання даних про мережу.

Розділ Preview дозволяє користувачеві візуально перевірити та оптимізувати подання графу, щоб зробити його більш інформативним, привабливим та легким для розуміння. Це важливий крок, який дозволяє отримати якісні та професійні візуалізації графів у системі Gephi.

9.4.5 Створення нового графу в Gephi

Існує три основні режими створення нового графа в Gephi:

- через інтерфейс Graph як Overview;
- через Data Laboratory;
- через експорт даних графа із зовнішнього файлу (найпростіше з файлу у форматі CSV з роздільниками крапка з комою).

Створення нового графу в режимі Overview

Після запуску програми та закриття спливаючого під час завантаження екрану, одразу активізується інтерфейс Overview, в рамках якого можна створити новий граф. Для цього достатньо активізувати новий проект та використовувати інструменти, позначені у правій частині вікна.

Для ручного нанесення вузлів за допомогою власних інструментів Gephi слід скористатися кнопкою "Олівець для малювання вузлів".

Вибравши місце на екрані Олівцем для малювання вузлів можна нанести нові вузли і за допомогою інструмента меню Розмір (значок – розмір графа) змінювати їх розмір.

За допомогою інструмента "Олівець для малювання ребер" можна розставити ребра графу. Після встановлення ребер можна перейти в режим фарбування (значок «фарба»).

При цьому існує можливість або зафарбовувати всіх сусідів обраного вузла, або скористатися індивідуальним забарвленням (праве верхнє меню робочої області). В результаті чого отримуємо остаточно сформований граф.

Створення нового графіу в режимі Data Laboratory

Для створення графа також зручно перейти в режим Data Laboratory, де в табличній формі відображається вся інформація про поточний стан графу. Причому інформація відображається у вигляді, придатному для зміни (редагування). Наприклад, можна додавати нові ребра (вузли), видаляти чи змінювати існуючі. Data Laboratory зручна також і для нанесення текстових міток вузлів.

У результаті на вкладці Graph в режимі Owerview відображаються мітки вузлів та ребер, шрифти, розмір та яскравість яких можна змінювати за допомогою інструментів, представлених у нижній частині екрана.

9.4.6 Експорт даних із зовнішнього файлу

Дані графа можна завантажувати в Gephi із текстового формату, в якому елементи-мітки вузлів розділені знаком "точка з комою". У цьому випадку до вузла, що відповідає першій мітці, у рядку "приєднуються" всі інші вузли, мітки

яких наведені у цьому рядку. Наприклад, нехай зовнішній файл містить такі записи:

```
Node1;Node2;Node3;Node4;Node5

Node5;Node3
```

У цьому випадку після завантаження в систему Gephi та їх обробки (підготовки до візуалізації вже описаним способом) отримуємо відображення (див. рис. 54):

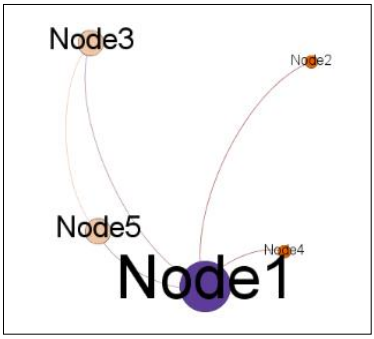


Рисунок 54 – Відображення графа після завантаження та обробки

Слід зазначити, що основним варіантом експорту даних графа із зовнішнього файлу є завантаження початкових мережевих даних у форматі CSV, у якому елементи розділені знаком «крапка з комою». У цьому випадку в CSV-файлі фактично повинна бути розширена мітками матриця інцидентності мережі. Нижче наведено приклад для мережі з п'яти вузлів:

```
;Node1;Node2;Node3;Node4;Node5
Node1;0;1;0;1;0
Node2;1;0;0;1;0
Node3;0;1;0;0;1
Node4;1;1;1;0;0
Node5;0;1;0;1;0
```

Після завантаження в систему Gephi та обробки вже описаним способом, отримуємо відображення (див. рис. 55):

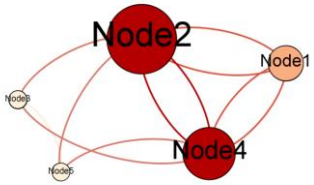


Рисунок 55 – Відображення графа, завантаженого з файлу у форматі CSV

Файл для завантаження можна підготувати в Excel і зберегти його у форматі CSV. Зверніть увагу, що перед завантаженням у Gephi необхідно у файлі CSV замінити всі коми (",") на точки із комами (";"). Це особливість Gephi.

Компонування та ранжування

При аналізі великих і щільних мереж швидке компонування (розташування вузлів графів) є вузьким місцем, оскільки більшість складних алгоритмів компонування вимагають значних ресурсів процесора, пам'яті та часу виконання. У той же час Gephi поставляється з ефективними алгоритмами компонування, такими як Yifan-Hu, Force-directed. Зокрема алгоритм Yifan-Hu є ідеальним варіантом для застосування після інших, більш швидких і грубих алгоритмів. У той час як більшість запропонованих в Gephi методів можуть бути виконані в розумний час, поєднання, наприклад OpenOrd і Yifan-Hu, надає найкращі візуальні уявлення. Звичайно, правильна параметризація будь-якого алгоритму компонування може вплинути як на час роботи, так і результат візуалізації.

Gephi має важливе значення для завдань аналізу та візуалізації семантичних мереж, включаючи ті, що

отримані авторами з використанням великих лінгвістичних моделей. Ось кілька ключових аспектів, які роблять Gephi цінним інструментом для роботи з семантичними мережами:

- гнучкий імпорт даних: Gephi підтримує імпорт даних із різних джерел, включаючи формати CSV, Excel, бази даних та інші. Це дозволяє користувачам легко завантажувати свої семантичні мережі, створені за допомогою великих лінгвістичних моделей, для подальшого аналізу та візуалізації.
- потужні алгоритми компонування: Gephi пропонує ефективні алгоритми компонування графів, такі як Yifan-Hu та Force-directed, які дозволяють автоматично укладати вузли семантичної мережі таким чином, щоб виявляти структури, патерни та угруповання;
- візуалізація та інтерактивність: Gephi забезпечує потужні інструменти візуалізації, які дозволяють користувачеві налаштовувати зовнішній вигляд вузлів та зв'язків, застосовувати фільтри, масштабувати та переміщатися семантичною мережею. Інтерактивність дає можливість взаємодіяти з графом та досліджувати його в деталях;
- виявлення спільнот та аналіз мережевих характеристик: Gephi надає функціонал для виявлення спільнот та аналізу різних мережевих характеристик, таких як центральність, ступінь важливості та інші, що дозволяє авторам семантичних мереж досліджувати їх структуру та властивості;
- можливість розширення: Gephi підтримує плагіни, що дозволяє розширити його функціональність та додати нові алгоритми та інструменти для роботи з семантичними мережами.

Все це робить Gephi потужним та зручним інструментом для аналізу та візуалізації семантичних мереж, отриманих

з використанням великих лінгвістичних моделей. Він дозволяє дослідникам проводити більш глибокий аналіз та розуміння складних взаємозв'язків між поняттями у таких мережах.

9.5 GraphViz

Сучасний рівень аналізу та візуалізації мережевих структур надають системи, створені великими колективами розробників, наприклад система Graphviz (Graph Visualization Software, http://graphviz.org). Ця система розроблена фахівцями лабораторії AT&T, поширюється з відкритими вихідними файлами за ліцензією EPL (Eclipse Public License), працює на багатьох операційних системах, у тому числі Linux, Mac OS, Unix-подібних ОС, Microsoft Windows.

Graphviz – це набір утиліт, бібліотек та програм з графічним інтерфейсом, представлених у вигляді опису мовою DOT, а також додаткових текстових та графічних програм, віджетів та бібліотек, що використовуються при розробці програмного забезпечення для візуалізації структурованих даних.

Graphviz включать такі інструменти:

- dot - Інструмент створення багаторівневого графа з можливістю виведення зображення результуючого графа у різних форматах (PNG, PDF, PostScript, SVG та інших);
- neato - інструмент для створення графа на основі "пружинної" моделі ("spring model", "energy minimised");
- twopi – інструмент для створення графа на основі "радіальної" моделі;
- circo – інструмент для створення графа на основі "кругової" моделі;

- fdp – інструмент створення неорієнтованого графа на основі моделі fdp;
- dotty – графічний інтерфейс для створення графів;
- lefty - програмований графічний віджет.

До пакета утиліт входить програмний модуль "dot" - автоматичний візуалізатор орієнтованих графів, який приймає на вхід текстовий файл мовою DOT з поданням графа у вигляді суміжних списків, а на виході формує граф у вигляді графічного, векторного або текстового файлу.

Програма dot - автоматичний візуалізатор орієнтованих графів, приймає на вхід текстовий файл з поданням графа у вигляді суміжних списків, а на виході формує граф у вигляді графічного, векторного або текстового файлу.

Програма DOT підтримує наступні формати вихідного файлу:

- PNG,
- GIF,
- JPEG,
- SVG(xml),
- DOT (txt),
- imap (html),
- VRML,
- PostScript и другие.

Для побудови графа в системі Graphviz достатньо в режимі Edit задати його опис спеціальною мовою опису DOT, а потім у режимі Graph вибрати вкладку Layout, щоб візуалізувати граф (рис. 55).

У режимі Graph → Setting можна змінювати параметри графа вручну без необхідності безпосереднього опису

мовою DOT, наприклад, змінювати форму вузлів, колір фону, колір вузлів тощо.

Вхідний файл для програми dot є звичайним текстовим файлом мовою DOT. Структура файлу DOT дуже проста. Програма "dot" сама розпізнає всі зв'язки графа та впорядковує його таким чином, щоб була мінімальна кількість перетинів.

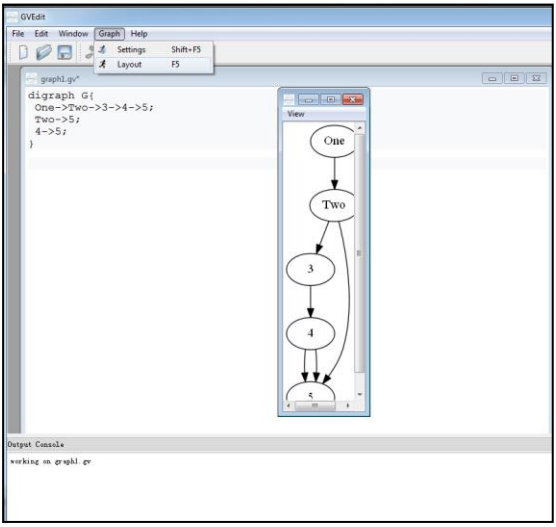


Рисунок 56 – Опис графа мовою DOT, візуалізація та вікно діагностики

Граф мовою DOT описується у вигляді списку субграфів, кожен з яких має вигляд:

```
graph %имя_графа% {

}
```

У цьому випадку у фігурних дужках {} є коментарі та інструкції, що описують окремий субграф. Інструкції описують вузли та ребра цільового графа та поділяються крапкою з комою.

Мова DOT підтримує коментарі у стилі мов C і C++ - // та /**/, а також символ #, як перший символ однорядкового коментаря.

Допускається подання неорієнтованих чи орієнтованих графів.

Неорієнтований граф мовою DOT описується списком вузлів та ребер, представлених назвою вузлів та подвійним тире (--) між зв'язаними вузлами, наприклад:

```
graph graphname {
    a;
    b;
    c;
    d;
    a -- b;
    b -- c;
    b -- d;
}
```

Припустимий скорочений опис:

```
graph graphname {
     a -- b -- c;
     b -- d;
}
```

Орієнтований граф мовою DOT описується списком вузлів та ребер, представлених назвою вузлів та стрілкою (→) між зв'язаними вузлами, наприклад:

```
digraph graphname {
     a -> b -> c;
     b -> d;
}
```

Припустимий і надмірний опис:

```
digraph graphname {
    a;
    b;
    c;
    d;
    a -> b;
    b -> c;
    b -> d;
}
```

При описі графів мовою DOT можна використовувати атрибути, що визначають колір, форму та стиль вузлів та ребер. Атрибути описуються парами ключ = значення, укладених у квадратні дужки ([ключ = значення]).

Для кожного елемента графа може бути визначено декілька атрибутів, розділених пробілом.

```
graph graphname {
     // label - видима назва вершини
     a [label="Foo"];
     // shape - визначення форми вершини
     b [shape=box];
     // color - визначення кольору ребра
     a -- b -- c [color=blue];
     // style - визначення стилю ребра
     b -- d [style=dotted];
}
```

Інтерпретатори мови DOT при візуалізації отримують елементи автоматично. Програма «dot» сама розпізнає всі зв'язки графа та впорядковує його таким чином, щоб була найменша кількість перетинів. Для корекції візуального представлення використовуються графічні редактори, серед яких представлений у даному пункті Graphviz.

Graphviz (Graph Visualization Software) має кілька переваг у порівнянні з іншими безкоштовними засобами аналізу та візуалізації графів під час роботи з мережами понять:

- простота використання: Graphviz надає простий та інтуїтивно зрозумілий спосіб візуалізації мереж понять. Після отримання даних за допомогою Chat GPT їх можна легко перетворити у формат DOT, який розуміє Graphviz, а потім візуалізувати графи без складних додаткових кроків;
- автоматичне укладання графів: Graphviz надає сильні алгоритми для автоматичного укладання графів, що робить процес візуалізації мереж понять більш ефективним та зручним. Це особливо важливо під час роботи з великими та складними графами;

- різноманітність форматів виводу: Graphviz підтримує множину форматів виведення візуалізацій, включаючи PNG, PDF, SVG та інші. Це дозволяє легко зберігати графи в потрібному форматі та використовувати їх у різних контекстах, таких як презентації, звіти або інтерактивні веб-програми;
- широка підтримка та активна спільнота: Graphviz має довгу історію розробки та активну спільноту користувачів та розробників. Це забезпечує доступ до оновлень, виправлень та нових можливостей, а також можливість отримання підтримки та допомоги за потреби;
- інтеграція з різними мовами програмування: Graphviz надає API інтерфейси для роботи з різними мовами програмування, що дозволяє інтегрувати його в існуючі проекти та використовувати для автоматизації процесу аналізу та візуалізації мереж понять.

Таким чином, GraphViz є потужним інструментом для візуалізації мереж понять, отриманих, наприклад, за допомогою Chat GPT або інших мовних моделей, який надає API інтерфейси для роботи з різними мовами програмування, зокрема Perl, PHP, Java, Python. Завдяки цій особливості його можна інтегрувати в множину додатків, що робить його універсальним засобом для роботи з графами.

Внутрішні алгоритми укладання графів у GraphViz дозволяють автоматично визначити розташування вузлів та зв'язків у графі таким чином, щоб мінімізувати перетини та забезпечити оптимальну читаність. Це робить його дуже зручним інструментом для створення красивих та інформативних графічних уявлень даних.

GraphViz підтримує різні формати виводу, включаючи SVG (Scalable Vector Graphics). Формат SVG дозволяє створювати інтерактивні графи з можливістю виведення в

Інтернет. Такі інтерактивні графи можуть бути використані для візуалізації складних структур даних, а також інтерактивних веб-додатках або звітах. Простота використання GraphViz, можливість автоматичного укладання графів, підтримка різних форматів та інтеграція з мовами програмування роблять його зручним та ефективним вибором для аналізу та візуалізації таких мереж.

У програмі, описаній в наступному пункті, GraphViz використовується для створення графів та збереження їх у форматі SVG, щоб надати інтерактивні мережі та візуалізації у веб-застосунку або інтернет-ресурсі. Це дозволяє користувачам взаємодіяти з графами, проводити пошук у зовнішніх пошукових системах, отримувати додаткову інформацію під час взаємодії з елементами графа.

9.6 CSV2Graph

При використанні наведених вище інструментів аналізу та візуалізації мережевих структур перед аналітиками виникають дві проблеми, а саме:

1. Необхідність встановлення програмних продуктів, що не завжди можливо, особливо якщо виникає потреба у роботі з мобільних пристроїв, нових операційних систем або в умовах обмежень на встановлення стороннього програмного забезпечення.
2. Необхідність вникати в особливості функціонування цих систем, розбиратися з десятками параметрів, режимами укладання графів, кластеризації тощо.

Gephi-Lite

Якщо з першою проблемою можуть допомогти розібратися онлайн системи аналізу за візуалізації графів, серед яких, до кращих, на думку авторів, можна віднести Lite версію системи Gephi - Gephi-Lite (https://gephi.org/gephi-lite) і веб-версію системи Graphviz

- WebGaphViz (http://www.webgraphviz.com), то для вирішення другого завдання в рамках оперативної побудови та відображення моделей предметних областей виникла необхідність розробки власного сервісу. Зазвичай, для опису графів використовуються спеціальні формати, серед яких можна виділити, такі як GML, GraphML, Pajek (NET), GraphViz (DOT), то для опису графових структур аналітиком предметних областей потрібно більш простий формат, що охоплює назви сутностей (вузлів), об'єднаних попарно. Кожна пара відображує ребро графа та напрямок (від першого вузла пари до іншої). Як спрощений формат найкраще підходить формат CSV, який, на жаль, не підтримується згаданими системами.

Для вирішення поставленої задачі на основі бібліотеки (API) системи GrahViz була розроблена програма, яка стала основою сервісу CSV2Graph, у даний час доступного в мережі Інтернет за адресою https://bigsearch.space/uli.html. Це сервіс забезпечує первинний аналіз та відображення графів, інформація про які відповідає формату CSV, кожен запис якого є назвою пари сутностей, розділених точкою з комою.

Дані вводяться у спеціальне текстове вікно, після чого вибирається тип графа (направлений/ненаправлений) та шляхом активізації клавіші Draw виконується відображення графа. На рис. 57 Наведено заповнену форму введення даних для сервісу CSV2Graph.

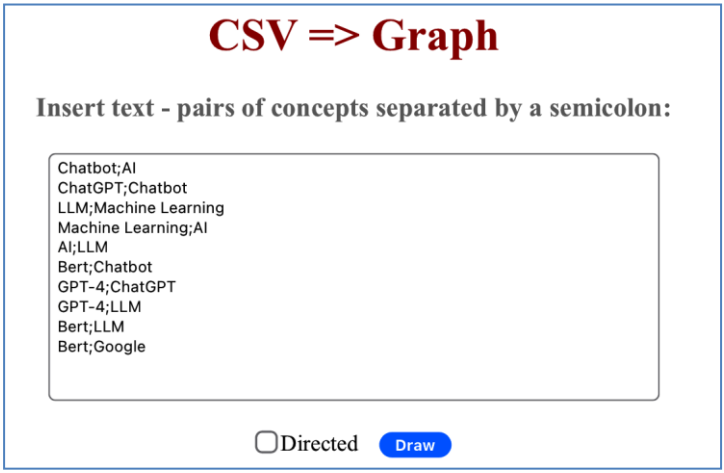


Рисунок 57 – Дані для подальшого аналізу та візуалізації

У результаті виконання програми формується відображення спрямованих та ненаправлених графів, ранжуються за ступенем та розфарбовуються вузли, визначається товщина та напрямок ребер. Укладання графа на площині виконується вбудованими у GraphViz методами. При цьому формується зображення графа у форматі SVG, завдяки чому реалізована можливість формування гіперпосилань, що ведуть пошукові форми системи Google News від вузлів і ребер графа. Scalable Vector Graphics (SVG) є форматом для визначення двох-dimensional graphics за допомогою XML. Це підтримує interactivity and animation. SVG specification, open standard розробили в World Wide Web Consortium, химерні зображення, щоб розрізати без втрати цінності. Ці зображення є завантаженими в XML-текстові файли, з можливістю їх пошуку (searchable), індексуємі (indexable), скриптовані (scriptable), і стислі (compressible). Починаючи з 2011 року, всі основні десктопні браузери почали підтримувати SVG.

Дані у файлі SVG є текстом, а не зображенням, тому в нього можна вбудовувати інтерактивні можливості,

зокрема гіперпосилання на веб-ресурси, що використовується у CSV2Graph. SVG-документи легко інтегруються з HTML- та XHTML-документами. Крім того, SVG – відкритий стандарт. На відміну від деяких інших форматів, SVG не є чиєюсь власністю.

На рис. 58 представлена мережа, що згенерована за допомогою CSV2Graph шляхом обробки даних, представлених на рис. 57.

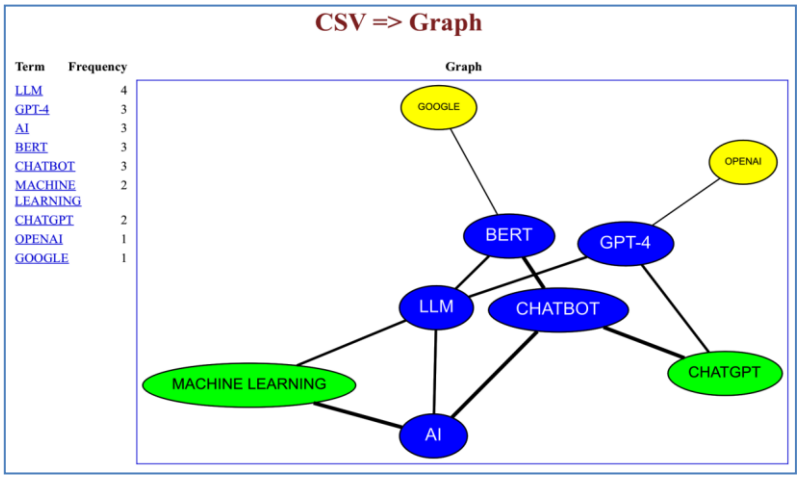


Рисунок 58 – Граф, згенерований на підставі даних у форматі CSV

Представлений сервіс успішно використовується при проведенні аналітичних досліджень для відображення великих моделей різних предметних областей, представлених багатьма мовами.

9.7 Основи роботи з Neo4j

Для роботи із мережевими задачами в рамках концепції великих даних» (Big Data) створюються спеціальні

програмні системи[49]. У тому числі, Neo4j — графова система керування базами даних із відкритим вихідним кодом, реалізована на Java. Neo4j вважається найпоширенішою графовою системою керування базами даних (СКБД) на серверних платформах, у той час як на клієнтських платформах лідером залишається Gephi.

Стартову сторінку веб-серверу розробника ситеми Neo4j (https://neo4j.com) наведено на рис. 59.

Дані в Neo4j зберігаються у форматі, пристосованому для представлення графової інформації. Цей підхід у порівнянні з моделюванням графової бази даних засобами реляційних систем керування базами даних дозволяє застосовувати спеціальну оптимізацію у разі мережевих даних.

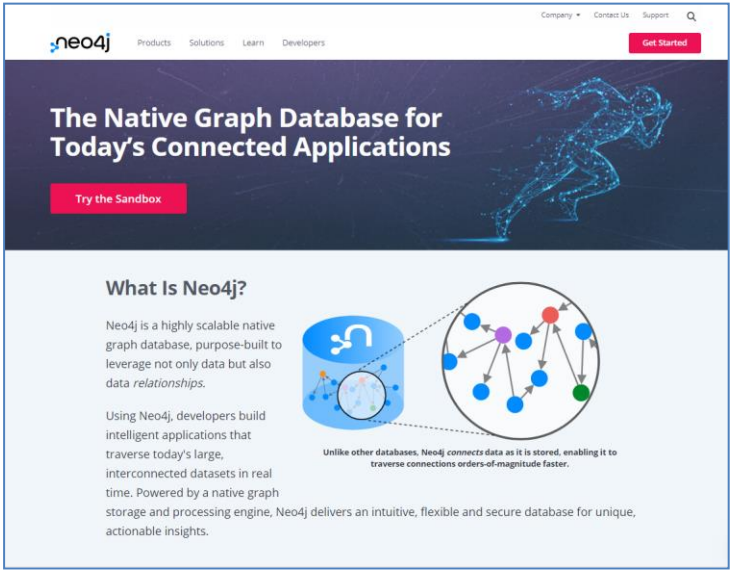


Рисунок 59 – Стартова веб-сторінка Neo4j

[49] Оброблення надвеликих масивів даних (Big Data) : навчальний посібник. / Д.В. Ланде, І.Ю. Субач, А.Я. Гладун. / - Київ 2021. - 168 с. ISBN 978-966-2344-83-7

Для обробки великого графа засобами Neo4j не потрібне його розміщення цілком в оперативну пам'ять сервера, таким чином, такі графи можуть завантажуватись і оброблятись частинами.

Інтерфейс прикладного програмування для СКБД Neo4j реалізований для багатьох мов програмування, включаючи Java, Python, Clojure, Ruby, PHP. У свою чергу, API реалізований у стилі REST.

У СКБД Neo4j використовується спеціальна мова Cypher, яка є не тільки мовою запитів, але й мовою маніпулювання даними, бо вона надає функції CRUD для графового сховища.

Для встановлення Neo4j необхідно перейти до центру завантаження за адресою: https://neo4j.com/download-center/.

Для встановлення Neo4j мають виконуватись такі початкові вимоги:

- для роботи СКБД Neo4j необхідно мінімум 2Gb оперативної пам’яті, а для стабільної роботи рекомендується 16Gb.
- в якості дискового масиву рекомендується застосовувати SSD-диски.
- Neo4j реалізована на Java, тому необхідне встановлення JVM8.

Для роботи з базою даних застосовується мережевий протокол bolt – «легкий» клієнт-серверний протокол.

Після запуску системи Neo4J за допомогою звичайного браузера можна перейти за посиланням http://localhost:7474/browser/, після чого з'являється інтерфейс Neo4j Browser (див. рис. 60).

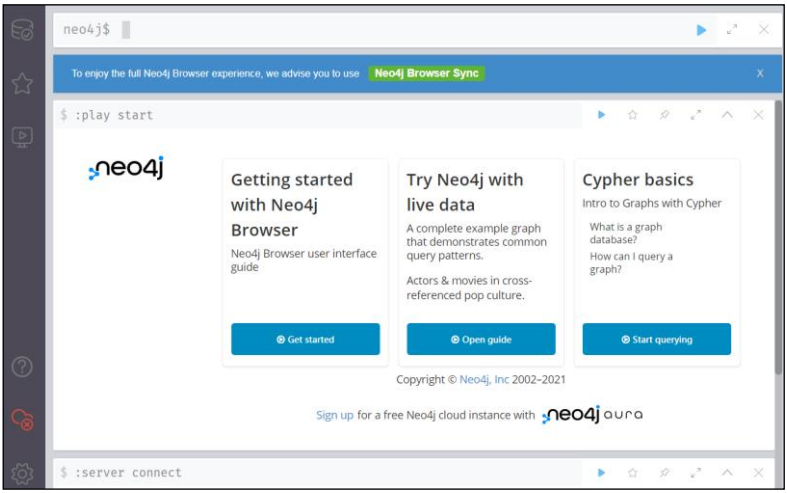


Рисунок 60 – Neo4j Browser

У верхній частині вікна Neo4j Browser розташовується рядок редактора для керування роботою у цьом середовищі.

Для створення і подальшої обробки графів можна застосовувати мову маніпулювання даними Cypher, яка надає функції CRUD для графового сховища.

На початку створення невеликого графу треба перейти в редактор і наберати першу команду мовою Cypher:

```
CREATE (u1:Person {name: "Василь", from: "Житомир"})
```

Після виконання команди Browser надає результат (див. рис. 61).

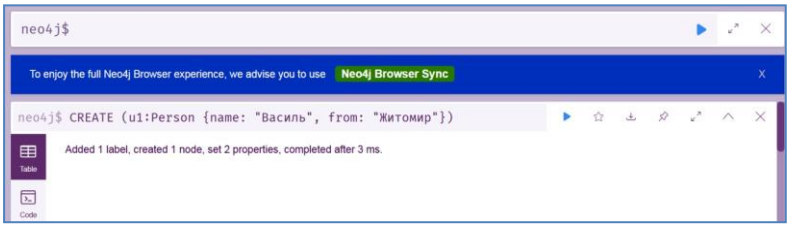


Рисунок 61 – Приклад створення вузла

Додаємо ще один вузол:

```
CREATE (u2:Person {name: "Дмитро", from: "Київ"})
```

Після цього можна запитати всі вузли типу Person і вивести значення властивості name (див. рис. 62):

```
MATCH (ee:Person) RETURN ee.name
```

Існує можливість упорядкувати отримані дані за якимось полем:

```
MATCH (ee:Person) RETURN ee.name ORDER BY ee.name
```

Зв'язки між вузлами в Cypher можна додавати за допомогою команди CREATE (див. рис. 63, 64).

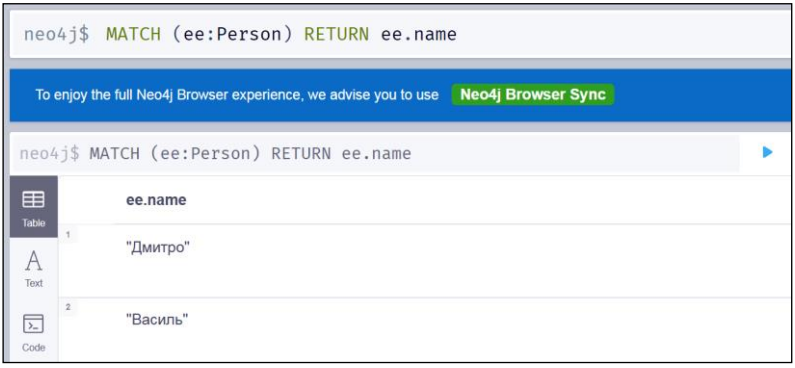


Рисунок 62 – Відображення списку вузлів

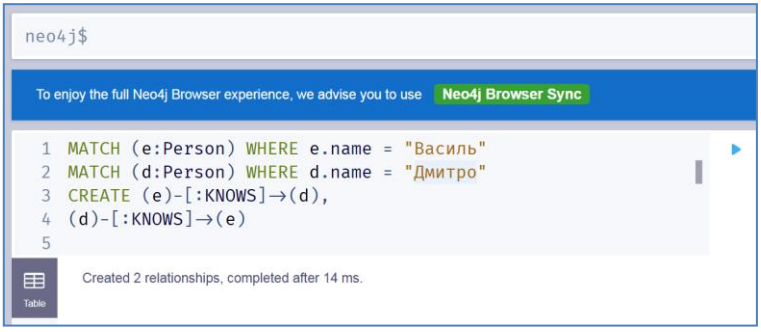


Рисунок 63 – Команди для графічного представлення вузлів

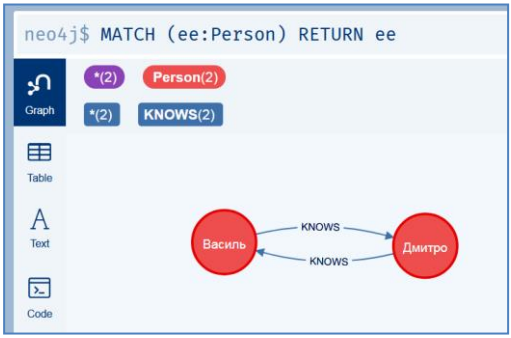


Рисунок 64 – Відображення графу

За допомогою мови Cypher можна також виконувати різні операції над графами, наприклад, запитувати суміжні вершини, видаляти ребра і вершини тощо.

Також можна налаштовувати Neo4j Browser на різні стилі відображення вузлів і зв'язків в залежності від наданих їм міток.

Імпорт даних із формату .csv

Розглянемо приклад, сукупність рядків, в яких кожен рядок виражає відношення суміжності між двома вузлами, наприклад, 1->2, 2->3, 1->3, тощо.

```
1,2
2,3
1,3
1,4
2,5
3,4
3,5
4,5
```

Цей файл можна розміщувати на локальній машині. Ім’я файлу, наприклад, «csv». Тоді команди завантаження файлу і формування графу будуть мати вигляд:

```
LOAD CSV FROM 'file:///csv' AS line
MERGE (a:node {name:line[0]})
MERGE (b:node {name:line[1]})
```

```
MERGE (a)-[:connects]->(b);
```

Після чого мережу можна відобразити (див. рис. 65) командою:

```
MATCH (ee:name) RETURN ee
```

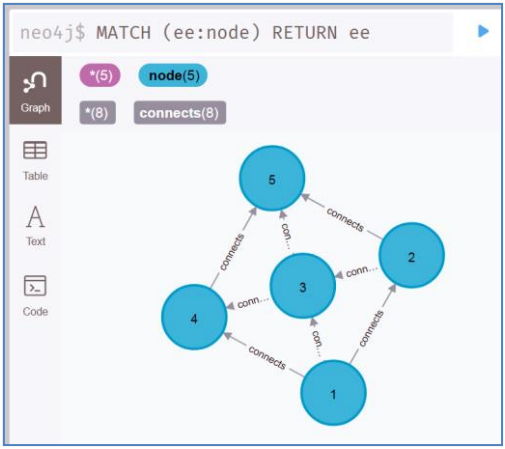


Рисунок 65 – Прикад відображення графа

Питання для самоконтролю

1. Яка основна мета програми uDraw(Graph) та які основні можливості вона надає користувачам?
2. Яке призначення програми Social Networks Visualizer (SocNetV)? Які є основні можливості цього програмного забезпечення?
3. Яку роль відіграють індекси зв'язності мереж у програмі SocNetV? Наведіть приклади деяких з таких індексів.
4. Які формати графів можна імпортувати та експортувати в програмі SocNetV? Як це може бути корисним для дослідження мереж?
5. Які алгоритми виявлення спільнот присутні у програмі SocNetV? Чому вони можуть бути корисними для аналізу соціальних мереж?
6. Які методи експорту даних надає програма SocNetV? Як це може бути корисним для подальшого використання або обробки візуалізованої інформації?

7. Що таке GraphML і яку роль він відіграє у програмах візуалізації графів? Які інші програми також використовують цей формат для зберігання і представлення мережевих структур?
8. Яким чином можна визначити граф у форматі GraphML? Які елементи використовуються для опису графа, вузлів та ребер?
9. Які атрибути заголовка XML-документу в форматі GraphML визначають стандартну XML-схему для документу? Яка їх роль у структурі документа?
10. Яким чином визначається спрямованість ребра у GraphML? Які значення може приймати XML-атрибут edgedefault елемента graph, і яка роль цього атрибута?
11. Які атрибути використовуються для оголошення вузла в GraphML? Яким чином ідентифікатор вузла задається за допомогою XML-атрибута id?
12. Які атрибути використовуються для оголошення ребра у GraphML? Як визначаються початковий та кінцевий вузол для кожного ребра?
13. Яким чином можна задати спрямованість ребра в GraphML? Які значення XML-атрибуту directed може приймати, і як вони впливають на спрямованість ребра в графі?
14. Які ліцензії використовуються для поширення Gephi, і які платформи підтримуються для встановлення програми? Яка версія Java необхідна для коректної роботи програми Gephi?
15. Які формати даних мереж підтримує Gephi для завантаження та експорту?
16. Які можливості надають інструменти укладання графів в програмі Gephi?
17. Які основні розділи (вікна) інтерфейсу користувача включає Gephi? На якій основній функціональності кожен з цих розділів спрямований?
18. Які основні можливості візуалізації мережі пропонує розділ Overview? Вкажіть декілька методів відображення.
19. Що таке розділ Preview в Gephi, і які основні можливості надаються користувачам для налаштування вигляду графа?

20. Яка ліцензія використовується для розповсюдження системи Graphviz, і на яких операційних системах вона підтримується?
21. Які інструменти входять до складу Graphviz, і які основні функції надають ці інструменти для візуалізації графів?
22. Яким чином інструмент "dot" допомагає візуалізувати графи на основі вхідних файлів у форматі DOT? В які формати можливо зберегти візуалізований граф?
23. Як описується граф мовою DOT? Які структури використовуються для визначення вузлів та ребер у графі, і як вони організовані у текстовому файлі?
24. Які дві основні проблеми виникають при використанні інструментів аналізу та візуалізації мережевих структур перед аналітиками?
25. Що таке Gephi-Lite і WebGraphViz, і як вони допомагають аналізувати та візуалізувати графи онлайн?
26. Який формат даних найбільш підходить для опису графових структур аналітиками предметних областей, і чому саме цей формат вибрано?
27. Які основні кроки аналітик повинен виконати, щоб використовувати сервіс CSV2Graph для аналізу та візуалізації графів?
28. Які можливості надає сервіс CSV2Graph для аналізу та візуалізації графів? Як відбувається ранжування вузлів, визначення товщини та напрямку ребер?
29. Яким чином використовується Scalable Vector Graphics (SVG) у сервісі CSV2Graph, і які переваги цього формату для візуалізації графів?

10. Приклади формування мереж

10.1 Мережа понять по сервісу Wikipedia

Мережа статей в Wikipedia взаємодіє між собою завдяки гіперпосиланням. Система Wikipedia є доступною в Інтернеті та не потребує підписки, крім того, її можна завантажити повністю. Проводячи аналіз певної частини цієї мережі, пов'язаної з конкретною темою, можна побудувати мережу, яка охоплює значну частину даної предметної області. Як приклад такої теми можна вказати ім'я дослідника, і на основі цього можна створити мережу, що об'єднує різні пов'язані з ним концепції.

Для вхідного доступу до системи був використаний термін, на який існує відповідна стаття у Wikipedia. Створена стаття піддавалась редагуванню експертами-авторами. Ясно, що мережа понять у рамках Wikipedia може стати значною за розмірами, якщо її не обмежувати конкретною тематичною областю. Вільний перехід за гіперпосиланнями може призвести до явища "зсуву тематики" (Topic Drift). Для подолання цього ефекту використовується проста тематична фільтрація: аналізу піддаються лише ті статті з Wikipedia, які містять основний термін, визначений експертом-аналітиком. Визначення кластерів у таких мережах може бути використано як основа для виявлення окремих наукових напрямків.

Розглядався алгоритм побудови мереж понять за даними сервісу Wikipedia, який припускає подолання ефекту Topic Drift:

1. Обирається перший термін-поняття, з якого починається зондування Wikipedia.
2. Відкривається сторінка веб-сервісу (стаття Wikipedia), яка відповідає обраному терміну-поняттю. До створюваної мережі додаються всі терміни-поняття, які відповідають гіперпосиланнями на обраній сторінці. Формуються ребра-зв'язки до цих

вузлів з початкового вузла.

3. Статті, що відповідають гіперпосиланням на попередній сторінці, визначаються як базові, якщо на них міститься гіперпосилання на статтю, відповідну першому терміну-поняттю, з якого починалося зондування.

4. Зі списку вузлів формованої мережі визначається той, за яким ще не здійснювалося переходу та на сторінку якого планується перейти для подальшого аналізу. Цей вузол повинен відповідати вимозі, наведеній у попередньому пункті, і входити до складу тих вузлів, до сторінок яких вже був здійснений перехід.

5. Якщо такий базовий вузол обраний, то здійснюється перехід до пункту 2.

6. Якщо такого вузла не існує, то вважається, що мережа, яка відповідає моделі предметної області, побудована.

Відповідно до наведеної процедури, процес отримання інформації з Wikipedia, який розпочинається з конкретного вузла-поняття, припиняється, коли неможливо здійснити новий перехід до іншого вузла (оскільки базові вузли для переходів вже вичерпані), і, отже, "циклічність" неможлива.

Для побудови мережі понять був використаний спеціальний інтернет-робот, результат роботи якого – набір даних у форматі CSV. Потім у програмі Gephi відкривається цей CSV-файл, який візуалізується з укладкою за алгоритмом Yifan Hu, розмір вузлів визначається по їх ступеню, розфарбування – по класу модулярності (див. рис. 6б).

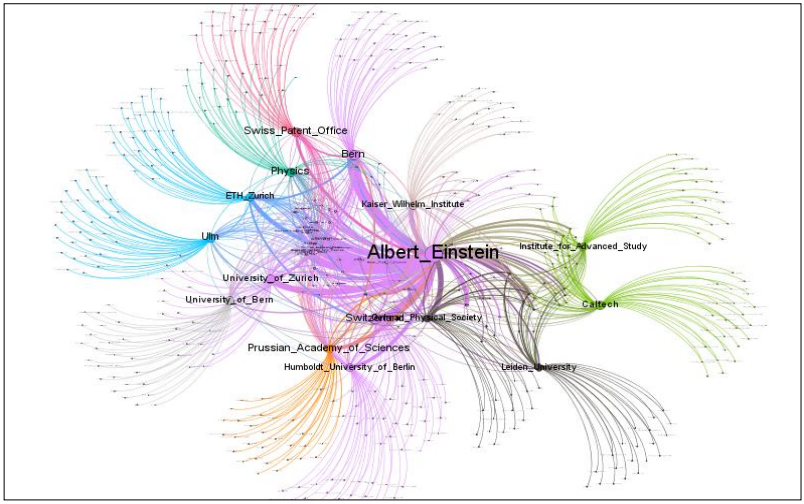


Рисунок 66 – Мережа понять, що відповідає сутності «Albert Einstein»

10.2 Мережа співавторства вчених

Для вирішення широкого спектра завдань, пов'язаних з управлінням науковою діяльністю, можна використовувати мережу співавторства між вченими. Ці завдання включають створення наукових колективів, експертних груп, оцінку рейтингів окремих дослідників тощо. Для створення такої мережі був використаний спеціальний інтернет-робот, який аналізує мережеву наукометричну базу даних Google Scholar, працюючий за таким алгоритмом:

1. Обирається перший автор (вузол), зі сторінки якого в Google Schoolar починається сканування мережі.
2. За допомогою експертної процедури формується обмежений перелік основних тегів, які відображають ключові концепції, пов'язані з автором, наприклад, теги computer, cyber security, information security.
3. Відкривається сторінка веб-сервісу Google Schoolar, який відповідає обраному автору. До створюваної

мережі додаються всі співавтори, що містяться на сторінці обраного учасника. Формуються ребра-зв'язки до цих вузлів (співавторів) з початкового вузла (автора).

4. З сформованого списку вузлів мережі випадковим чином вибирається той, на сторінку якого планується перейти для подальшого аналізу. Цей вузол також повинен задовільняти тематиці обраної предметної області – його теги входять до складу дескрипторів, визначених на кроці 2, і не входять до складу тих вузлів, до сторінок яких вже був здійснений перехід.
5. Якщо такий вузол-автор обраний, то здійснюється перехід до кроку 3.
6. Якщо такого вузла не існує, то вважається, що мережа співавторів, побудована.

За цим алгоритмом процес вивчення мережі, що розпочинається з певного вузла, припиняється при виявленні "циклічності" - тобто коли згідно з алгоритмом повинен настати перехід до вузла, який вже був відвіданий, а також у випадку, якщо сусідні вузли відхиляються від основної тематики (що визначається на підставі лексичного складу тегів).

Результат роботи програми сканування Google Schoolar – набір даних у вже поданому форматі CSV. Потім в програмі Gephi відкривається цей CSV-файл, який обробляється із застосуванням укладки за алгоритмом Yifan Hu, розмір вузлів обирається заPageRank, а розфарбування – по класу модулярності (див. рис. 67).

Ще одна мережа співавторства (в науці про мережі, 1589 співавторів, +2742 зв'язки), зібрана М. Нейманом у травні 2006 року (https://gephi.org/datasets/netscience.gml.zip) після укладки і кластеризації граф приймає вид, представлений на рис. 68.

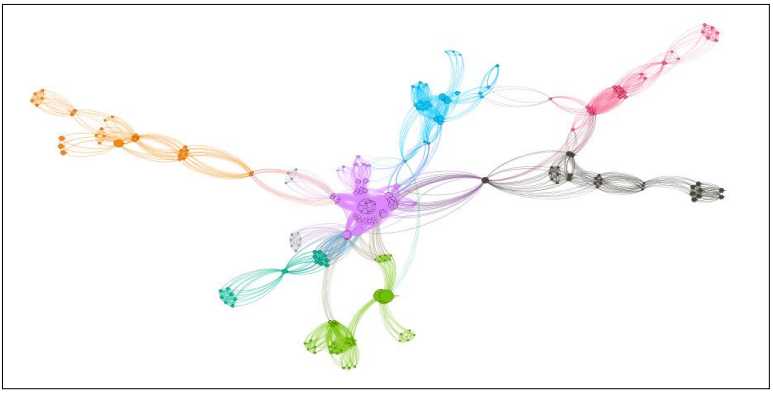


Рисунок 68 – Мережа співавторства вчених, розділена на кластери

10.3 Мережі на основі ChatGPT

Останнім часом великі лінгвістичні моделі, такі як ChatGPT набувають все більшого поширення в багатьох областях. Найпоширеніші застосування – це машинний переклад, реферування текстів, узагальнення різного рівня, наприклад, формулювання питань до навчальних матеріалів.

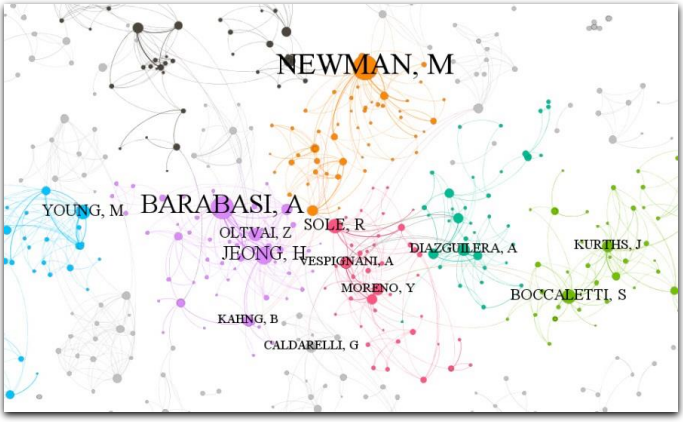


Рисунок 69 – Мережа співавторства в науці про мережі (укладка Yifan Hu)

Зокрема, ChatGPT від OpenAI – це Генеративний Попередньо навчений Трансформер (GPT), який використовує обробку природної мови для виконання промтів користувачів, використовуючи широкі можливості області штучного інтелекту[50].

Величезні можливості в екстрагуванні основних понять, іменних сутностей дозволяють використовувати ChatGPT у фактографічних системах, зокрема, в медицині, економіці[51]. Інтелектуальні чати інтегруються із зовнішніми системами, такими як геоінформаційні, системи аналізу та візуалізації графів, мереж[52]. Зокрема, у роботі[53] показано, як формувати мережі зв'язків персонажів літературних творів, мережі предметних областей зі зв'язками типу «загальне-приватне».

Ця робота присвячена опису методики формування причинно-наслідкових (каузальних) мереж шляхом багаторазового звернення до системи ChatGPT, а також візуалізації та аналізу цих мереж за допомогою системи Gephi (gephi.org) – найпопулярнішої програми візуалізації графових структур із вільною ліцензією[54]. Для завантаження даних у середу Gephi цілком підходить

---

[50] St. Wolfram. "What Is ChatGPT Doing ... and Why Does it Work?". – Wolfram Media, Inc. March 9, 2023. 112 p.

[51] Brady D. Lund, Ting Wang, Nishith Reddy Mannuru, Bing Nie, Somipam Shimray, Ziang Wang. ChatGPT and a new academic reality: Artificial Intelligence-written research papers and the ethics of the large language models in scholarly publishing. JASIST, 2023. Vol. 74, Iss. 5. pp. 570-581.

[52] Tamilla Triantoro. Graph Viz: Exploring, Analyzing, and Visualizing Graphs and Networks with Gephi and ChatGPT (March 30, 2023). ODSC Community.

[53] Dmytro Lande, Leonard Strashnoy. GPT Semantic Networking: A Dream of the Semantic Web – The Time is Now. – Kyiv: Engineering, 2023. – 168 p. Available at https://bigsearch.space/datasets/Lande_Str_Book.pdf

[54] Ken Cherven. Mastering Gephi Network Visualization. Packt Publishing, 2015. 378 p.

формат CSV, тому всі запити до ChatGPT будуть супроводжуватись вимогою до цього формату.

Сформовані причинно-наслідкові мережі забезпечать можливість переходу до сценарного аналізу. Основна проблема, що виникає під час проведення сценарного аналізу на основі каузальних мереж полягає саме у створенні таких систем, що у традиційних випадках потребує великих ресурсних витрат та залучення експертів.

10.3.1 Формування мережі на базі простого ієрархічного звернення до ChatGPT

Нехай нас, наприклад, цікавить проблематика витоку даних та його причини. Попросимо у ChatGPT видати відомі їй причини цього явища. Тобто центральним вузлом майбутньої мережі має стати поняття "Data Leakage". Успішне відпрацювання такого запиту визначить другий рівень ієрархії – поняття пов'язані з витоком даних та його причини. Після цього для кожного такого поняття також вимагають множину причин, що вплинули на нього. Такий процес може тривати нескінченно, але в роботі зупинимося на трьох рівнях. Незважаючи на ієрархічне формування такої каузальної мережі, отримана мережа загалом не буде строго ієрархічною структурою.

Запропонувавши ChatGPT відпрацювати деякий запит, отримаємо множину причин первинного поняття. Система ChatGPT може допомогти отримати зміст CSV-файлу (поля, відповідні іменам понять, розділені точкою з комою). Для цього можна застосувати, наприклад, такий запит до системи ChatGPT:

> →List the causes of **data leakageage** in cyber security. The reason is to use no more than three words. The results should be presented in the format "cause;**data leakage**". Each such entry - from a new line

Система видає відповідь приблизно такого вигляду:

human error; data leakage
weak passwords; data leakage
insider threats; data leakage
misconfigured systems; data leakage
phishing attacks; data leakage
unpatched software; data leakage
malware infection; data leakage
social engineering; data leakage
third-party access; data leakage
stolen devices; data leakage

Запити наступного рівня будуть ставитись до наведених у відповіді концептів і мати вигляд, що повністю відповідає первинному запиту, наприклад:

→List the causes of **human error** in cyber security. The reason is to use no more than three words. The results should be presented in the format "cause; **human error** ". Each such entry - from a new line

Об'єднані в одному CSV-файлі відповіді ChatGPT завантажуються для аналізу та візуалізації програми Gephi.

Завантаживши отримані дані до системи Gephi, вибираємо розмір вузлів, пропорційний ступеню (кількості суміжних зв'язків) і розділивши мережу на кластери за критерієм модулярності отримуємо наочний граф (див. рис. 69).

Найбільш впливові вузли цієї мережі (найбільший Out-Degree), це: human error (5), social engineering (4), weak passwords(3), phishing attacks(2).

Очевидно, сформована мережа слабопов'язана, неповна, представлені в ній концепти можуть не точно відображати причини та наслідки. Вважатимемо, що це мережа, отримана в результаті опитування лише одного штучного експерта.

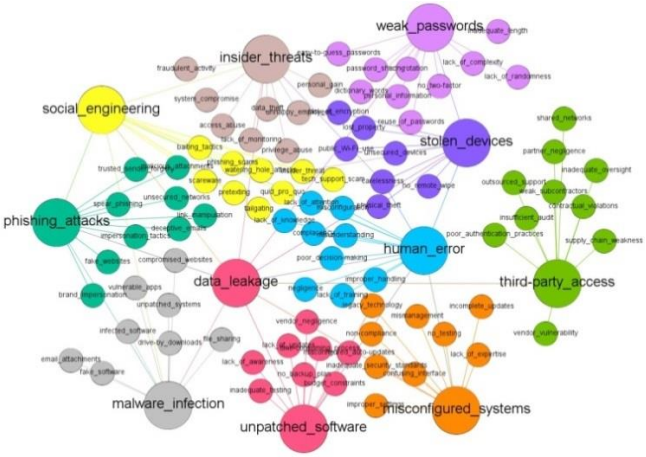


Рисунок 69 – Спрямована первинна каузальна мережа, отримана шляхом найпростішого ієрархічного звернення до ChatGPT

10.3.2 Формування мережі на основі ієрархічного звернення до ChatGPT

Система ChatGPT у різні моменти під час обробки тексту може видавати різні варіанти відповідей, причому правильні, і з погляду людської логіки цілком «обґрунтовані». Кожну таку відповідь можна сприймати як відповідь деякого віртуального експерта. Можна припустити, що узагальнюючи відповіді множини (рою) подібних експертів можна отримати більш повну та точну відповідь. Реалізуючи рій віртуальних експертів ми по кілька разів задаємо одні й ті самі запити, що розглядаються в минулому випадку, які стосуються як першого, так і другого рівня ієрархії. Після отримання відповідей від системи, об'єднуємо їх у загальний CSV-файл і передаємо для аналізу та візуалізації програмі Gephi. Завантаживши отримані дані до системи Gephi, отримуємо граф, поданий на рис. 70. На практиці мережа може поповнюватися доти, доки не стане достатньо повною за оцінкою експерта-людини.

Найбільш впливові вузли цієї мережі (найбільший Out-Degree), це: Human error (7), social engineering (4), weak passwords(3), phishing attacks(2), unpatched systems(2), insider threats(2).

Як бачимо, кількість важливих концептів збільшилася порівняно з попереднім випадком.

10.3.3 Узагальнення поняття віртуальних експертів

Сформований у попередньому прикладі граф, маючи відносно велику повноту концептів, водночас може містити неточну інформацію, помилково видану ChatGPT при обробці окремих запитів. З припущення, що ймовірність появи тих самих помилок - невелика, можна винести з розгляду при побудові мережі концепти, які зустрічаються рідше заданого порогу.

У наведеному нижче випадку (див. рис. 70) не розглядалися концепти, які зустрічалися рідше двох разів.

Найбільш впливові вузли цієї мережі (найбільший Out-Degree), це:

Human error (5), social engineering (3), phishing attacks(2), unpatched systems(2).

На підставі експертних оцінок можна зробити висновок, що первинна каузальна мережа, отримана шляхом найпростішого ієрархічного звернення до ChatGPT, охоплює найбільшу кількість концептів, які відносно слабко пов'язані (мережа близька до ієрархічної), але завдяки повноті може бути непоганою «сировиною для подальшої аналітичної обробки».

Статистично оброблена друга мережа, каузальна мережа, отримана шляхом ієрархічного звернення роя віртуальних експертів до ChatGPT, є більш точною, ніж первинна мережа і, нарешті, третя мережа, отримана шляхом узагальнення ієрархічного звернення роя віртуальних експертів до ChatGPT, що має найбільший

середній свідчить про найбільшу взаємодію окремих концептів, що впливають на ціль у цій причинно-наслідковій мережі. Мабуть, така мережа є найбільш прийнятною для подальшого застосування сценарного аналізу.

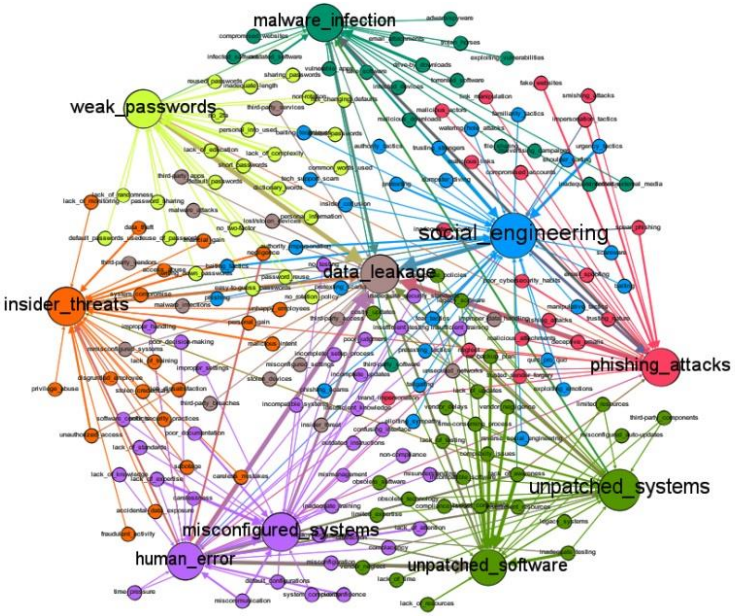


Рисунок 70 – Спрямована повна каузальна мережа, отримана шляхом ієрархічного звернення рою віртуальних експертів до ChatGPT

Незважаючи на суттєвий виграш у ресурсах (як часових, так і людських), важливо зазначити, що як сам процес побудови каузальних мереж, так і інтерпретація результатів, вимагають від датасайнсиста досвіду в предметній галузі, що вивчається, і як і раніше необхідно спостереження з боку людини для забезпечення достовірності та точності результатів.

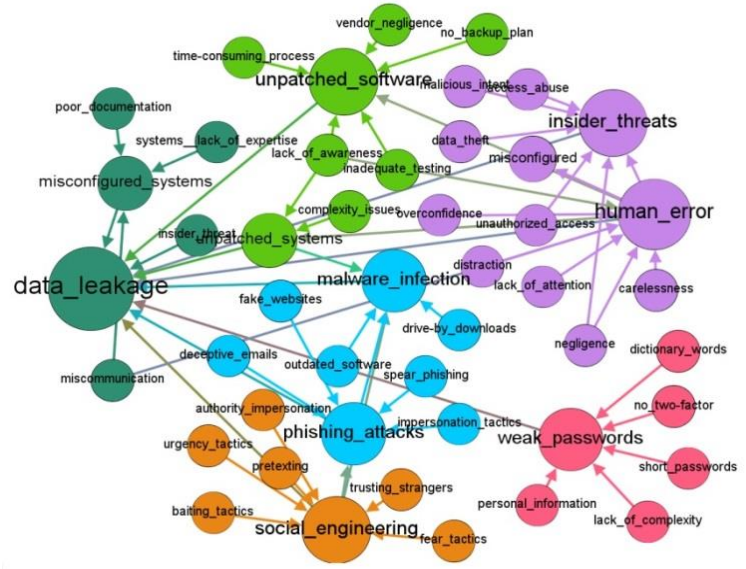


Рисунок 70 – Направлена каузальна мережа, отримана шляхом узагальнення ієрархічного звернення рою віртуальних експертів до ChatGPT

Питання для самоконтролю

1. Яким чином побудована мережева енциклопедія Wikipedia, і які її основні особливості?
2. Як можна побудувати мережу, що охоплює більшу частину предметної області, використовуючи гіперпосилання в Wikipedia?
3. Як ефект "зсуву тематик" (Topic Drift) впливає на побудову мережі понять з Wikipedia, і як можна його подолати?
4. Який алгоритм використовується для побудови моделей предметних областей за даними Wikipedia, і які кроки включає цей алгоритм?
5. Які вимоги встановлюються до базових термінів-понять для ефективного застосування алгоритму побудови моделей з Wikipedia?
6. Які етапи включає процес збору інформації з Wikipedia за наведеним алгоритмом, і коли цей процес припиняється?
7. Яким чином інтернет-робот використовується для збору даних з Wikipedia для подальшої побудови мережі понять?

8. Які особливості використання програми Gephi для обробки отриманого CSV-файлу та побудови мережі понять, і які параметри встановлюються для відображення мережі?
9. Для яких конкретних завдань можна використовувати мережу співавторства учених, побудовану на основі даних Google Scholar?
10. Яким чином працює інтернет-робот для побудови мережі співавторства з бази даних Google Scholar, і які кроки включає алгоритм роботи цього інструмента?
11. Які кроки включає алгоритм побудови мережі співавторства на основі даних Google Scholar, та які критерії вибору співавторів в цьому процесі?
12. Яким чином забезпечується припинення процесу побудови мережі співавторства при зацикленні та врахуванні основної тематики?
13. Які дані отримуються в результаті роботи програми сканування Google Scholar, і у якому форматі вони представлені?
14. Які параметри використовуються для обробки отриманого CSV-файлу в програмі Gephi при побудові мережі співавторства?
15. Які особливості використання алгоритму Yifan Hu для укладання мережі співавторства в програмі Gephi, і яким чином розмір вузлів та їх розфарбування визначаються?
16. Чому важлива кластеризація мережі співавторства, і як це впливає на подальший аналіз даних про співавторів?
17. Які головні застосування великих лінгвістичних моделей, таких як ChatGPT, в різних областях індустрії?
18. Що таке ChatGPT, і як він використовує обробку природної мови для задоволення потреб користувачів?
19. Які можливості надає використання великих лінгвістичних моделей для екстрагування основних понять і іменних сутностей в різних областях, таких як медицина та економіка?
20. Як інтелектуальні чати інтегруються з зовнішніми системами, і як це може сприяти покращенню аналізу даних?
21. Яким чином програма Gephi використовується для візуалізації і аналізу мереж, побудованих на основі великих лінгвістичних моделей?

22. Яка роль формату CSV у роботі з даними для програми Gephi, і чому саме цей формат використовується?
23. Які особливості створення причинно-наслідкових мереж за допомогою великих лінгвістичних моделей, і чому це може бути важливим для аналізу в різних галузях?
24. Які можливі виклики, пов'язані з проведенням сценарного аналізу на основі каузальних мереж і як це вирішується за допомогою великих лінгвістичних моделей?
25. Яким чином використання рою віртуальних експертів може поліпшити точність і повноту відповідей, наданих системою ChatGPT?
26. Які основні кроки процесу побудови каузальних мереж за допомогою рою віртуальних експертів, які ви описали?
27. Які фактори визначають впливовість вузлів у побудованих каузальних мережах, і чому саме ці вузли є найбільш впливовими?
28. Як змінюється кількість важливих концептів під час роботи з роєм віртуальних експертів порівняно з первинною мережею?
29. Яким чином обробка більш великої кількості запитів від рою впливає на точність та повноту каузальних мереж?

Список рекомендованої літератури

1. **Albert R., Jeong H., Barabasi** A. Error and attack tolerance of complex networks. Nature, 2000. – Vol. 406. – pp. 378-382.
2. Barabási A.L., Albert R. Emergence of scaling in random networks. Science, 1999. – Vol. 286 (5439): 509–512.
3. Bezsudnov I.V., Snarskii A.A. From the time series to the complex networks: The parametric natural visibility graph, Physica A: Statistical Mechanics and its Applications 414, 53-60.
4. Brady D. Lund, Ting Wang, Nishith Reddy Mannuru, Bing Nie, Somipam Shimray, Ziang Wang. ChatGPT and a new academic reality: Artificial Intelligence-written research papers and the ethics of the large language models in scholarly publishing. JASIST, 2023. Vol. 74, Iss. 5. pp. 570-581.
5. Brin S., Page L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. WWW7, 1998.
6. Caldeira S. M. G., Petit Lobao T. C., Andrade R. F. S., Neme A., Miranda J. G. V. The network of concepts in written texts // Preprint physics/0508066 (2005)
7. Ken Cherven. Mastering Gephi Network Visualization. Packt Publishing, 2015. 378 p.
8. Dorogovtsev S.N., Mendes J.F.F. Evolution of Networks: From Biological Networks to the Internet and WWW. – Oxford, USA: Oxford University Press, 2003. – 280 p.
9. Erdős P., Rényi A. On the evolution of random graphs. Publ. Math. Inst. Hungar. Acad. Sci., 1960. – № 5. – pp. 17-61.
10. Ferrer-i-Cancho, R., Sole R.V. The small world of human language // Proc. R. Soc. Lond. B 268, 2261 (2001)
11. Lacasa L., Luque B., Ballesteros F., Luque J., Nuno J.C. From time series to complex networks: The visibility graph, Proceedings of the National Academy of Sciences, 105 (13) 4972-4975, 2008
12. Dmytro Lande, Leonard Strashnoy. GPT Semantic Networking: A Dream of the Semantic Web – The Time is Now. – Kyiv: Engineering, 2023. – 168 p. ISBN 978-966-2344-94-3
13. Douglas A. Luke. A User's Guide to Network Analysis in R. Springer; 1st ed. 2015 edition (December 21, 2015). 250 p. ISBN: 3319238825
14. Luque B., Lacasa L., Ballesteros F., Luque J. Horizontal visibility graphs: Exact results for random time series, Physical Review E 80 (4), 046103, 2009
15. Eric Ma and Mridul Seth. Network Analysis Made Simple An introduction to network analysis and applied graph theory using Python and NetworkX. Leanpub, 2021. – 191 p.

16. Kleinberg J.M. Authoritative sources in a hyperlink environment. In Processing of ACM-SIAM Symposium on Discrete Algorithms, 1998, 46(5):604-632.
17. Masinde, N., Graffi, K. Peer-to-Peer-Based Social Networks: A Comprehensive Survey. SN COMPUT. SCI. 1, 2020. p. 299.
18. Mark Newman, Albert-Laszlo Barabasi, Duncan J. Watts. The Structure and Dynamics of Networks: (Princeton Studies in Complexity). – Princeton, USA: Princeton University Press, 2006. –624 p.
19. Von Neumann, J. and A. W. Burks (1966). Theory of self-reproducing automata. Urbana, University of Illinois Press.
20. Newman M.E.J., Moore C. and Watts D.J. Mean-field solution of the small-world network model. Phys. Rev. Lett. 84, 2000. 3201–3204.
21. Newman M.E.J. and Watts D.J.. Scaling and percolation in the small-world network model. Phys. Rev. E 60, 1999. 7332–7342.
22. Ortuño M., Carpena P., Bernaola P., Muñoz E., Somoza A.M. Keyword detection in natural languages and DNA // Europhys. Lett, – 57(5). – P. 759-764 (2002)
23. Tamilla Triantoro. Graph Viz: Exploring, Analyzing, and Visualizing Graphs and Networks with Gephi and ChatGPT (March 30, 2023). ODSC Community.
24. Watts D.J., Strogatz S.H. Collective dynamics of "small-world" networks. Nature, 1998. – Vol. 393. – pp. 440-442.
25. Wolfram S. (2002). A New Kind of Science. – Champaign, IL: Wolfram Media Inc.
26. St. Wolfram. "What Is ChatGPT Doing ... and Why Does it Work?". – Wolfram Media, Inc. March 9, 2023. 112 p.
27. Zeinalipour-Yazti D., Kalogeraki V., Gunopulos D. Information Retrieval in Peer-to-Peer Networks. IEEE CiSE Magazine, Special Issue on Web Engineering, 2004. – pp. 1-13.
28. Головач Ю., Пальчиков В. Лис Микита і мережі мови. Журнал фізичних досліджень. – Т. 11, №. 1, 2007. – С. 22-33.
29. Ланде Д.В. Методи оцінки рівня дискримінантної сили слів у текстах з правової тематики // Правова інформатика, 2012. – № 3 (35). – С. 5-9.
30. Ланде Д.В., Субач І.Ю. Візуалізація та аналіз мережевих структур : навчальний посібник. – Київ : КПІ ім. Ігоря Сікорського, Вид-во "Політехніка", 2021. – 80 с. ISBN 978-966-2577-14-3
31. Ланде Д.В., Субач І.Ю., Гладун А.Я. Оброблення надвеликих масивів даних (Big Data) : навчальний посібник. – Київ: Інжиніринг, 2021. – 168 с. ISBN 978-966-2344-83-7

Предметний показчик

Навчальне видання

Снарський Андрій Олександрович
Ланде Дмитро Володимирович
Субач Ігор Юрійович

ОСНОВИ ТЕОРІЇ СКЛАДНИХ МЕРЕЖ

Навчальний посібник

В авторській редакції
Надруковано з оригінал-макета замовника

Інститут спеціального зв'язку та захисту інформації
Національного технічного університету України
«Київський політехнчний інститут імені Ігоря Сікорського»
вул Верхньоключова, 4, м. Київ, Україна
тел. 204-91-51

ISBN 978-966-2344-95-0

Підписано до друку 01 11 .2023. Формат 60x84/16. Папір офс.
Гарнітура Times. Спосіб друку – ризографія. Обсяг 9,3 авт. арк.

ТОВ "Інжиніринг"

ISBN 978-966-2344-95-0