VŠB TECHNICKÁ UNIVERZITA OSTRAVA
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Analýza struktur v komplexních sítích zaměřená na posouzení kvality vybraných metod

Analysis of Structures in Complex Networks Focused on Assessing the Quality of Selected Methods

Ondřej Chovanec

Diplomová práce

Vedoucí práce: doc. Mgr. Miloš Kudělka, Ph.D.

Ostrava, 2021

Abstrakt

Při analýze dat se často využívá komplexních sítí. Analýzou struktury sítě můžeme získat mnoho důležitých informací a používá se k tomu mnoho metod a algoritmů s různými účely. Cílem této práce je provést experimenty s vybranými algoritmy a porovnat jejich výsledky. Jednou z klasických úloh je hledání komunit v síti. V první části této práce jsme použili dvě metody detekce překrývajících se komunit. Naším cílem bylo porovnat a ohodnotit kvalitu nalezených komunit na základě tří měr. Pro analýzu vektorových dat lze také využít sítě. Nejdříve je ale nutné zkonstruovat síť z vektorových dat pomocí vybrané metody, ale není vždy jasné, kterou by bylo nejvhodnější použít. V druhé části práce je popsán provedený experiment, při kterém jsme využili čtyř metod konstrukce sítí z vektorových dat a následně popsali vlastnosti a odlišnosti získaných sítí. Využijeme také v této práci blíže popsanou tzv. podobnostní účelovou funkci pro ohodnocení kvality konstruovaných sítí.

Klíčová slova

sítě, překrývající se komunity, detekce komunit, konstrukce sítě z vektorových dat, hodnocení metod

Abstract

Complex networks are often used in data analysis. By analyzing the structure of the network, we can obtain a lot of important information and many methods and algorithms are used for different purposes. The aim of this work is to perform experiments with selected algorithms and compare their results. One of the classic tasks is finding communities on the network. In the first part of this work, we used two methods of detecting overlapping communities. Our goal was to compare and evaluate the quality of the found communities on the basis of three measures. Networks can also be used to analyze vector data. First, however, it is necessary to construct a network from vector data using the selected method, but it is not always clear which would be the most appropriate to use. The second part of the work describes the performed experiment, in which we used four methods of construction of networks from vector data and then described the properties and differences of the obtained networks. We will also use the so-called similarity purpose function described in more detail in this work to evaluate the quality of constructed networks.

Keywords

networks, overlapping communities, community detection, network construction from vector data, evaluation of methods

Poděkování

Rád bych poděkoval panu doc. Mgr. Miloši Kudělkovi, Ph.D. za jeho rady, dobré a věcné připomínky a vstřícnost odpovědět na jakýkoli můj dotaz během vypracovávání této práce.

Obsah

Seznam použitých zkratek a symbolů

DEMON	–	Democratic Estimate of the Modular Organization of a Network
DBLP	–	Digital Bibliography & Library Project
KNN	–	k-Nearest Neighbors

Seznam obrázků

Seznam tabulek

Kapitola 1

Úvod

Sítě hrají důležitou roli v mnoha oblastech informatiky a nás bude zajímat jejich uplatnění při analýze dat. Cílem této práce je provést experimenty s algoritmy týkající se detekce komunit v sítích a metodami konstrukce sítí z vektorových dat. Komunity se hledají v sítích, abychom našli skupiny uzlů v síti s podobnými vlastnostmi a mohli si vytvořit představu o její struktuře. Jenomže pro hledání komunit není vždy jednoduché vybrat vhodnou metodu a následně ohodnotit kvalitu nalezených komunit. V první části této práce jsme využili identifikované (tzv. ground-truth) komunity a algoritmy Ego-zones a DEMON pro nalezení překrývajících se komunit tří sítí. Následně jsme využili různých měr pro ohodnocení a porovnání kvality těchto komunit. Druhá část práce se věnuje metodám konstrukce sítě z vektorových dat. Požadavkem standardně je, aby vzniklá struktura sítě co nejvíce odpovídala vektorovým datům. Na to, abychom mohli ohodnotit kvalitu sítě vzhledem k datům, využijeme v práci popsanou podobnostní účelovou funkci. Dalším cílem bylo zjistit, jaké charakteristiky mají sítě získané různými metodami konstrukce a porovnat je mezi sebou.

V kapitole 2, uvádějící první část, si nejdřív řekneme, jaká je definice komunity, a v kapitole 3 jsou popsány použité algoritmy detekce komunit. Kapitola 4 obsahuje definici použitých měr a kapitola 5 obsahuje charakteristiky použitých sítí. Popis experimentu s algoritmy detekce komunit a jeho výsledky obsahuje kapitola 6.

Druhá část začíná kapitolou 7, ve které je vysvětleno z jakých kroků se skládá konstrukce sítě z vektorových dat. Jsou v ní také popsány námi použité metody pro konstrukci. Pro ohodnocení kvality konstrukce budeme využívat účelovou funkci, kterou popisujeme v kapitole 8. Nakonec kapitola 9 obsahuje popis a porovnání vlastností získaných sítí.

Poznámka 1 V následujícím textu se v několika případech používají originální anglické názvy, jelikož neexistují zažité české ekvivalenty.

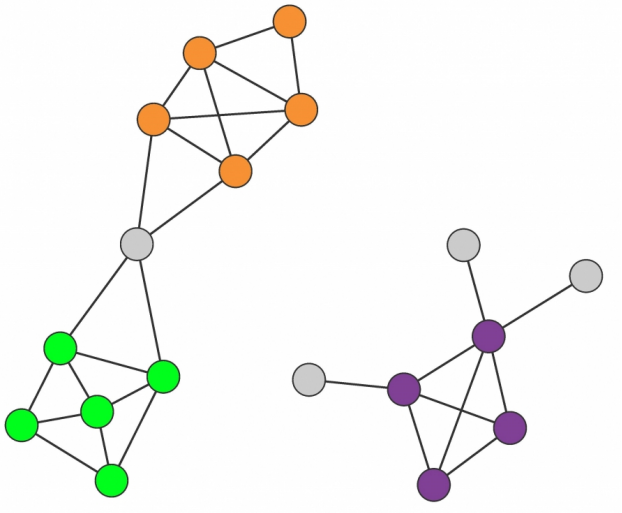
Kapitola 2

Komunita

Pro komunity neexistuje jediná obecně přijímaná definice, ale většinou se za komunity považují souvislé lokálně hustě propojené podgrafy v síti. [1] Ukázka komunit je na obrázku 2.1, kde různé komunity jsou obarveny různými barvami.

Souvislost Každá komunita musí být součástí souvislého podgrafu. Pokud je síť tvořena více komponentami, tak není možné, aby uzly z různých komponent byly součástí stejné komunity. Stejně tak platí, že komunita se nemůže skládat ze dvou podgrafů stejné komponenty, pokud mezi nimi neexistuje hrana.

Lokální hustota Uzly v komunitě se pravděpodobněji připojí k ostatním uzlům stejné komunity než k uzlům v jiných komunitách. Platí tedy, že existuje výrazně více hran mezi uzly stejné komunity než mezi uzly komunity a zbytkem sítě.



Obrázek 2.1: Komunity [1]

Příklady komunit na reálných sítích můžou být například skupiny lidí, které sdílejí stejné zájmy či profesní obory. Lidé, kteří se zabývají chemií budou mezi sebou komunikovat více než s profesionálními sportovci. Stejně tak webové stránky budou pravděpodobněji odkazovat na jiné stránky obsahující podobný obsah a vytvoří komunity.

Komunity lze dále rozlišovat podle toho, jestli se navzájem překrývají. Pokud každý uzel patří maximálně do jedné komunity, tak se jedná o **nepřekrývající** se komunity. Naopak pokud může uzel patřit do více komunit, tak hovoříme o **překrývajících** se komunitách. Speciálním případem jsou **vnořené** komunity, jež jsou komunity, které jsou celé součástí větší komunity.

Například mějme komunitu lidí, jež mají rádi fotbal a komunitu lidí, jež mají rádi hokej. V reálném světě samozřejmě existují lidé, jež mají rádi oba tyto sporty a patří do obou těchto komunit, které se tedy navzájem překrývají. Obě tyto komunity jsou vnořené komunitě lidí, jež mají rádi sport.

Kapitola 3

Algoritmy detekce komunit

Komunity v sítích nalezneme pomocí algoritmů detekce komunit. Tyto algoritmy lze dělit podle toho, jestli se nalezené komunity překrývají nebo ne. Mezi nepřekrývající patří například dobře známá **Louvain metoda** [2]. My se ale budeme zabývat **ground-truth** překrývajícími se komunitami a algoritmy detekce překrývajících se komunit konkrétně **Ego-zones** a **DEMON**.

3.1 Ground-truth

Ground-truth [3] není algoritmus detekce komunit, ale jedná se o komunity, které byly získané na základě znalostí, jež o síti máme, či po její detailní manuální inspekci. Jedná se tedy o opravdové komunity, které se v sítí vyskytují. Tyto *ground-truth* komunity lze pak použít pro určení kvality algoritmu detekce komunit podle toho, jak moc se shodují s těmi, které algoritmus nalezl. Příklad *ground-truth* komunit může být, že v sociální síti identifikujeme komunity lidí na základě jejich zájmů. Lidé se stejným zájmem by byli členy stejné komunity.

3.2 Ego-zones

Ego-zones [4] je algoritmus, jež spojuje uzly, které jsou navzájem závislé do tzv. **ego-zón**. *Ego-zóny* vychází z **ego sítí**. *Ego síť* je podsítí, jež je tvořena tzv. **egem**, což je libovolný uzel původní sítě, a jeho sousedy tvořící tzv. *sousedství*. Někdy můžeme zvětšit *ego síť* tak, aby obsahovala všechny uzly do definované vzdálenosti od *ega*.

3.2.1 Dependency

Dependency (česky závislost) hraje klíčovou roli při hledání *ego-zón*. *Dependency* vyjadřuje, jak moc jsou uzly na sobě závislé a jak moc si jsou podobné. *Ego-zones* využívá pouze strukturu sítě a závislost mezi dvojicí uzlů, je proto definována pomocí počtu společných sousedů, kteří zvyšují

závislost, a počtu vrcholů, jež nemají společné, kteří naopak závislost snižují. Vzorec pro výpočet závislosti pro váženou síť je definován následovně:

$$D(x,y) = \frac{w(x,y) + \sum_{v_i \in CN(x,v_i)} w(x,y) \cdot r(x,v_i,y)}{\sum_{v_j \in N(x)} w(x,v_j)}$$

$$r(x,v_i,y) = \frac{w(v_i,y)}{w(x,v_i) + w(v_i,y)}$$

kde $CN(x,y)$ je množina všech společných sousedů x a y, $N(x)$ je množina všech sousedů x, $w(x,y)$ je váha hrany mezi x, y a $r(x,v_i,y)$ je koeficient závislosti uzlu x na uzlu y prostřednictvím společného souseda v_i.

Závislost není symetrická, tedy $D(x,y)$ se nemusí rovnat $D(y,x)$. Pokud bychom pracovali s neváženými sítěmi, tak budou váhy všech hran rovny 1 a $r(x,v_i,y)$ se bude vždy rovnat 0,5. Hodnota v čitateli proto bude stejná pro obě závislosti, ale hodnoty jmenovatelů se mohou lišit. Pokud platí $D(x,y) \geq 0,5$, pak říkáme, že x je závislé na y. Matice závislostí A je matice, jejíž element a_{xy} je roven $D(x,y)$. Na základě závislostí můžeme přidělit uzel do jedné z několika kategorií.

Prominentní Není závislý na žádném uzlu a existuje aspoň jeden uzel, který je na něm závislý.

Slabě prominentní Je závislý aspoň na jednom uzlu a existuje aspoň jeden uzel, který je na něm závislý.

Neprominentní Není na něm závislý žádný uzel.

3.2.2 Ego-zóna

Ego-zóna je definována následovně:

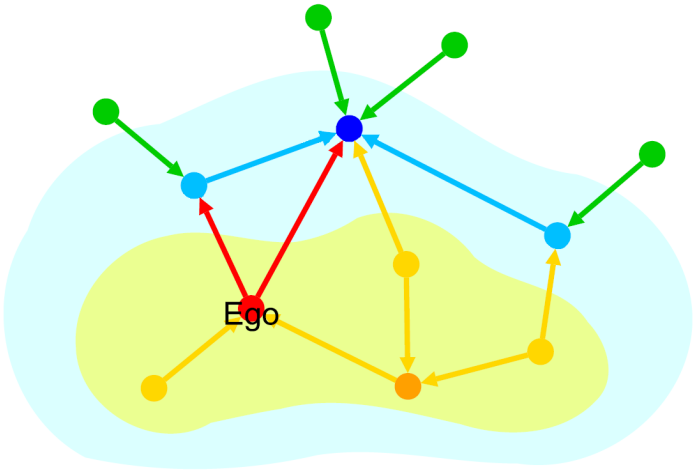
1. Výchozím členem *ego-zóny* je jakýkoli síťový uzel zvaný *ego*.

2. Členem *ego-zóny* je jakýkoli uzel, který je závislý na *egu* nebo jiném uzlu ego-zóny. Množina všech takových uzlů včetně ega se nazývá **vnitřní zóna** (inner zone).

3. Členem *ego-zóny* je každý uzel mimo vnitřní zónu, na kterém je alespoň jeden uzel vnitřní zóny závislý. Množina všech těchto uzlů se nazývá **vnější zóna** (outer zone).

Na obrázku 3.1 je zobrazena *ego-zóna*. Žlutá část na obrázku zobrazuje vnitřní a světle modrá vnější zónu *ego-zóny*. Zelené vrcholy nepatří do *ego-zóny*.

Na *ego-zóny* (zóny závislosti) můžeme nahlížet jako na komunity, mezi kterými dochází k velkým překryvům. Toto je důležitá vlastnost, jelikož komunity, jež existují v reálných sítích, často mezi sebou mají velké překryvy a problémem současných metod detekce komunit bývá, že buď najdou komunity s jenom malými překryvy, anebo nalezené komunity nemají překryvy vůbec.



Obrázek 3.1: Ego-zóna [4]

3.2.3 Složitost

Podrobný popis složitosti je možno nalézt v [4]. My v této práci budeme stručnější. *Ego-zones* se skládá z výpočtu závislostí (sekce 3.2.1) a nalezení *ego-zón* (sekce 3.2.2). Hustá síť je obecně nejhorší případ a časová složitost hledání společných sousedů dvou uzlů je $O(n^2)$. To znamená, že výpočet matice závislostí má v nejhorším případě složitost $O(m \cdot n^2)$, kde n je počet uzlů sítě a m je počet hran sítě. Pro řídké sítě složitost hledání společných sousedů souvisí s průměrným stupněm sítě d a časová složitost je v tomto případě $O(m \cdot d^2)$.

Za předpokladu, že se detekuje každá zóna zvlášť, je časová složitost detekce zón pro všechny uzly sítě v nejhorším případě $O(n \cdot m)$ pro husté sítě. Nicméně, musíme také uvážit případy, kdy může být mnoho uzlů v síti izolováno. V případě, že zóna obsahuje pouze uzel ega, je časová složitost detekce zóny $O(1)$. Můžeme tedy očekávat, že časová složitost bude nižší pro sítě v reálném světě.

Ukazuje se, že v sítích z reálného světa je časová složitost výpočtu závislostí $O(m \cdot \log n)$. Podobně se ukazuje, že časová složitost detekce *ego-zón* v reálných sítích je $O(m)$. Celkově lze potom předpokládat, že má algoritmus v reálných sítích složitost $O(m \cdot \log(n + m))$. Můžeme tedy předpokládat, že složitost výpočtu je v reálném světě podstatně nižší než ve výše uvedených nejhorších případech.

3.3 DEMON

DEMON [5, 6] je zkratka pro *Democratic Estimate of the Modular Organization of a Network*. Stejně jako *Ego-zones* využívá *ego sítí*, v nichž hledá lokální komunity. Algoritmus funguje následovně:

1. Pro každý uzel sítě se získá tzv. **ego minus ego** síť, což je ego síť bez ego uzlu.

2. Na získanou *ego minus ego* síť se použije **label propagation algoritmus** [7] pro nalezení množiny komunit C v této podsíti.

3. Každá nalezená komunita c v C se potom spojí s komunitou i z množiny dosud nalezených komunit I, pokud platí, že maximálně $\epsilon\%$ uzlů komunity c není součástí komunity i. Kde ϵ je vstupním parametrem algoritmu.

Časová složitost algoritmu pro sítě odpovídající modelu malého světa je pak $O(nK^{3-\alpha})$. Kde n je počet uzlů sítě, K je maximální stupeň v síti a α je exponent mocninného rozdělení. Tento algoritmus je tedy možno použít i na sítě o stovkách miliónů uzlů.

Kapitola 4

Míry hodnocení kvality komunit

Nyní potřebujeme ohodnotit komunity, abychom mohli porovnávat algoritmy detekce komunit. Použijeme na to tři míry: modularitu (sekce 4.1), vodivost (sekce 4.2) a CRanku (sekce 4.3). Každá míra je založená na jiných vlastnostech komunit a sítě. CRank je nejkomplexnější mírou z nich a je pro nás tedy nedůležitější.

4.1 Modularita

Modularita [8] je velmi často používaná míra, která posuzuje kvalitu rozdělení sítě na části. Tato míra se používá k optimalizaci v mnoha komunitních algoritmech, mezi které patří dobře známá a už zmiňovaná *Louvain metoda*. Čím vyšší modularita tím lepší rozdělení sítě. Modularita nabývá hodnot z $\langle -\frac{1}{2}, 1\rangle$ a standardní modularita pro sítě se spočítá následovně:

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

kde m je počet hran, i a j jsou uzly sítě, k_i, k_j jsou stupně těchto uzlů a c_i, c_j jsou komunity, do kterých tyto uzly náleží. A je matice sousednosti a δ je funkce *Korneckerovo delta*, která je 1, pokud uzly patří do stejné komunity, jinak je rovna 0.

Modularitu lze použít pro porovnání kvality shlukování mezi různými algoritmy, ale problém je ji použít pro porovnání komunit získaných z různých sítí, které se výrazně liší velikostí. Modularita totiž s počtem komunit a tedy i velikostí sítě roste. Modularita má i další nevýhody. Někdy má tendenci slučovat menší *ground-truth* (sekce 3.1) komunity do větších, anebo naopak velké *ground-truth* komunity rozdělovat na menší. Jedná se o tzv. limit rozlišení (anglicky resolution limit).

Pokud bychom však chtěli ohodnotit jenom kvalitu jedné komunity c_i, abychom ji mohli porovnat vůči ostatním komunitám, můžeme vypočíst modularitu pro případ, kdy máme jenom 2 komunity, komunitu c_i a druhou komunitu $c_j = V \setminus c_i$, kde V je množina všech uzlů sítě.

4.2 Vodivost

Vodivost [9] sítě (anglicky conductance) je míra, která říká, jak dobře je síť propojená. Mějme řez sítí $\left(S, \bar{S}\right)$. Vzorec pro výpočet vodivosti pro tento řez je:

$$\varphi(S) = \frac{\sum_{i \in S, j \in \bar{S}} A_{ij}}{\min(|E_S|, |E_{\bar{S}}|)}$$

kde A je matice sousednosti, E_S je množina hran řezu S a $E_{\bar{S}} = V - E_S$. Množina S reprezentuje komunitu, pro kterou měříme kvalitu. Čím nižší hodnota vodivosti tím lepší komunita, protože to značí, že je lépe oddělená od zbytku sítě. Celková vodivost sítě G je pak minimem ze všech možných řezů:

$$\phi(G) = \min_{S \subseteq V} \varphi(S)$$

kde V je množina uzlů sítě. Čím vyšší vodivost tím lépe je síť propojená. Tento vzorec však nebudeme potřebovat.

4.3 CRank

CRank [10] je algoritmus, který byl vytvořen speciálně pro hodnocení kvality komunit a pro nás bude nejzajímavější mírou. Vstupem pro algoritmus je seznam hran sítě a seznam komunit. Výstupem je pak ohodnocení komunit podle toho, jak dobře jsou vytvořeny, a je to ukazatelem, které komunity jsou vhodné k dalšímu rozboru, jelikož pro velké sítě často není čas dopodrobna zkoumat všechny.

CRank upřednostňuje komunity na základě robustnosti a významu několika strukturálních vlastností každé komunity. Robustnost vlastnosti je definována jako změna v hodnotě vlastnosti mezi původní sítí a její náhodně pozměněnou verzí získanou přepojováním hran s tím, že je zachována distribuce stupňů uzlů. Parametr α vyjadřuje intenzitu změny sítě. Pro hodnotu 0 je síť nezměněná, kdežto 1 je maximálně pozměněná verze, jež odpovídá náhodné síti.

Vzhledem ke strukturní vlastnosti f je definována metrika r_f pro stanovení priority, aby byla kvantifikována změna hodnoty f mezi původní a pozměněnou sítí.

$$r_f(C, \alpha) = \frac{f(C)}{1 + d_f(C, \alpha)}$$

kde $f(C)$ je hodnota vlastnosti f komunity C v původní síti, α určuje intenzitu změny sítě, $d_f(C, \alpha) = |f(C) - f(C|\alpha)|$ je změna hodnoty vlastnosti pro komunitu C mezi originální sítí a její α-pozměněnou verzí a $f(C|\alpha)$ je hodnota vlastnosti f v α-pozměněné síti. Pro určení pořadí komunit využívá CRank 4 metrik, pro které se počítají priority za použití $\alpha = 0{,}15$:

Pravděpodobnost Metrika pravděpodobnosti komunity (community likelihood) celkově kvantifikuje spojitost dané komunity. Měří pravděpodobnost sítě mít komunitní strukturu. Kvantifikuje, jak dobře lze pozorované hrany vysvětlit komunitou. Intuicí je, že vysoce kvalitní komunita přispěje vysokou pravděpodobností k vysvětlení pozorované hrany. Výpočet metriky pravděpodobnosti komunity je definován následovně:

$$f_l(C|\alpha) = \prod_{u \in C} p_C(u) \prod_{v \in C} s_C(u, v|\alpha)$$

kde $s_C(u, v|\alpha)$ je definováno jako:

$$s_C(u, v|\alpha) = \begin{cases} p_C(v)p_C(u, v|\alpha) & \text{if } (u, v) \in E \\ p_C(v)(1 - p_C(u, v|\alpha)) & \text{if } (u, v) \notin E \end{cases}$$

$p_C(u, v|\alpha)$ je příspěvek komunity C k vytvoření hrany (u, v) při aplikování změn na síť o intenzitě α.

Hustota Hustota komunity (community density) jednoduše měří celkovou sílu hran v komunitě. Hustota implicitně bere v úvahu potenciálně hierarchické a překrývající se komunitní struktury. Když je komunita vnořená do jiných komunit, tyto obklopující komunity přispívají ke zvýšené hustotě vnitřních hran komunity. Formálně je definována hustota komunity jako pravděpodobnost hran mezi členy komunity.

$$f_d(C|\alpha) = \prod_{(u,v) \in E, u \in C, v \in C} p(u, v|\alpha)$$

kde $p(u, v|\alpha)$ je pravděpodobnost hrany (u, v) při intenzity změny sítě α.

Hranice Hranice komunity (community boundary) doplňuje vnitřní spojitost měřenou pomocí hustoty komunity. Hranice komunity bere v úvahu sílu hran vedoucích ven z komunity. Strukturním rysem vysoce kvalitní komunity je její dobré oddělení od okolních částí sítě. Jinými slovy, vysoce kvalitní komunita by měla mít ostrou hranici.

$$f_b(C|\alpha) = \prod_{u \in C, v \in V \setminus C} (1 - p(u, v|\alpha))$$

Věrnost Věrnost komunity (community allegience) je definována jako preference uzlů připojovat se k uzlům patřící do stejné komunity. Věrnost měří zlomek uzlů v komunitě, pro které je celkový počet hran směřujících dovnitř komunity vyšší než počet hran, které směřují ven z komunity.

$$f_a(C|\alpha) = \frac{1}{|C|} \delta \left(\sum_{v \in N_u \cap C} \delta(p(u, v|\alpha) \geq \sum_{v \in N_u \setminus C} p(u, v|\alpha) \right)$$

kde N_u je množina sousedů uzlu u a δ je funkce, $\delta(x) = 1$, když x je pravda, v opačném případě $\delta(x) = 0$.

Problémem je, že metriky mohou být neobjektivní, mít vysokou odchylku a chovat se odlišně v různých sítích. Po zjištění hodnot těchto metrik je nutné, je spojit do jednoho souhrnného hodnocení. CRank používá iterativní metodu agregace pozic, která kombinuje pořadí určených pomocí jednotlivých metrik do jednotného pořadí komunit. Bere v úvahu skutečnost, že důležitost jednotlivých metrik se v různých sítích a metodách detekce komunit liší. CRank vytvoří pořadí komunit pro každou metriku, které má přiřazenou sadu vah důležitosti. Metoda poté vypočítá agregované pořadí komunit.

Kapitola 5

Sítě

Při experimentu jsme použili následující neorientované nevážené sítě:

Amazon Jedná se o síť z webu *Amazon* [11], kde uzly jsou produkty a hrany značí, že jsou tyto produkty často nakupovány dohromady. Každá kategorie produktů společnosti *Amazon* definuje *ground-truth* komunitu.

DBLP Bibliografie informatiky *DBLP* [12] poskytuje ucelený seznam výzkumných prací z oblasti informatiky. Jedná se o síť spoluautorů, kde jsou spojeni dva autoři, pokud společně zveřejnili alespoň jednu práci. Autoři, kteří publikovali ve stejném časopise nebo na stejné konferenci, tvoří *ground-truth* komunitu.

Youtube *Youtube* [13] je web pro sdílení videí, jehož součástí je i sociální síť. V sociální síti *Youtube* si uživatelé navzájem vytvářejí přátelství a uživatelé mohou vytvářet skupiny, ke kterým se mohou ostatní uživatelé připojit. Tyto skupiny považujeme za *ground-truth* komunity.

Statistika	**Amazon**	**DBLP**	**Youtube**
Uzly	334 863	317 080	1 134 890
Hrany	925 872	1 049 866	2 987 624
Průměrný shlukovací koeficient	0,3967	0,6324	0,0808
Trojúhelníky	667 129	2 224 385	3 056 386
Poměr uzavřených trojúhelníků	0,07925	0,1283	0,002081
Průměr	44	21	20
90% percentil průměru	15	8	6,5

Tabulka 5.1: Statistiky použitých sítí

Statistické údaje sítí můžeme vidět v tabulce 5.1. Vidíme, že Amazon a DBLP si jsou podobné počtem uzlů a hran, ale DBLP obsahuje mnohem více trojúhelníků, má vyšší průměrný shlukovací koeficient a kvůli tomu má i nižší průměr. Z toho lze usoudit, že DBLP více odpovídá modelu malého

světa než Amazon, což dává smysl, jelikož se jedná o sociální síť. Youtube je přibližně třikrát větší než Amazon a DBLP, co do počtu uzlů i hran. Má ale nižší průměrný shlukovací koeficient a průměr. To značí, že v Youtube existuje buď pár obrovských center nebo velké množství menších center. Všechny tyto sítě a data o nich byly získány z [14].

Kapitola 6

Experiment s hodnocením komunit

Nyní konečně popíšeme testování kvality nalezených komunit. Hodnotili jsme *ground-truth* komunity (sekce 3.1) a komunity získané pomocí algoritmů *Ego-zones* (sekce 3.2) a *DEMON* (sekce 3.3).

Pro výpočet měr byla použita C++ aplikace vytvořená autory článku [10], který je dostupný ke stažení na [15]. Pro získání komunit pomocí *Ego-zones* byla použita C# aplikace napsaná Milošem Kudělkou, která je dostupná ke stažení na [16]. Pro získání komunit pomocí *DEMON* byla použita python knihovna *demon* napsaná autory článků [5, 6].

Minimální velikost hledaných komunit pro algoritmy detekce komunit byla nastavena tak, aby odpovídala minimální velikosti *ground-truth* komunit. Pro Amazon je tato minimální velikost komunit 3, pro DBLP 6 a pro Youtube 2.

6.1 Obecné charakteristiky nalezených komunit

Nejdřív se podíváme na počty nalezených komunit. Pro různé kombinace sít/metoda můžeme vidět hodnoty v tabulce 6.1. Z této tabulky si můžeme udělat několik úsudků.

Je vidět, že počet *ground-truth* komunit se liší na základě toho, jak byly definovány. Jelikož Amazon má hodně kategorií produktů, tak má asi 5,5 krát více *ground-truth* komunit než DBLP, kde jsou *ground-truth* komunity určeny časopisy a konferencemi. Zajímavé je, jak nízký počet *ground-truth* komunit je definováno v síti Youtube, která má 3 krát více uzlů než zbytek sítí, ale počet *ground-truth* komunit srovnatelný s DBLP. Možným vysvětlením je, že primárním smyslem Youtube je sdílení videí, a ne sociální interakce mezi lidmi, tedy tvoření skupin lidí není moc využívaná funkce této sítě.

	Amazon	**DBLP**	**Youtube**
Ground-truth	75 149	13 477	16 386
Ego-zones	74 697	46 957	199 180
Demon	20 510	15 109	10 796

Tabulka 6.1: Počet nalezených komunit

Když se podíváme na komunity získané pomocí *Ego-zones* můžeme si všimnout, jak jejich počty mohou být ovlivněny průměrným shlukovacím koeficientem. Dávalo by to i smysl, jelikož *Ego-zones* využívá společných sousedů pro hledání komunit, které, bude-li jich v síti více, zvýší shlukovací koeficient a závislost mezi uzly. Kvůli vysoké závislosti se následně uzly spojí do větších komunit, kterých pak tedy bude méně. I když Amazon a DBLP jsou si velikostí podobné, tak *Ego-zones* nalezl o 30 000 komunit méně pro DBLP, což odpovídá našemu předpokladu, protože DBLP má vyšší průměrný shlukovací koeficient. Tento údaj je však výrazně ovlivněn i tím, že jsme hledali komunity s různými minimálními velikostmi. Pro Amazon to bylo 3, kdežto pro DBLP 6. Takže je těžké říct, jak moc velký efekt shlukovací koeficient hraje.

Pokud bychom porovnávali *Ego-zones* oproti *ground-truth* komunitám, tak vidíme, že pro Amazon se počet komunit téměř shoduje, ale pro DBLP a Youtube jich našel podstatně více. Je zřejmé, že buď rozdělil *ground-truth* komunity na menší komunity, nebo našel mnoho komunit neodpovídajících těm *ground-truth*.

Když se podíváme na počet komunit získaných pomocí *DEMON*, tak vidíme, že pro DBLP a Youtube nalezl podobný počet komunit jako je *ground-truth* komunit, avšak pro Amazon nalezl méně než jejich třetinu. Obecně nalezl pokaždé méně komunit něž *Ego-zones*, a to několikanásobně. Zajímavé je, že pro Youtube, jež je největší sítí nalezl nejmenší počet komunit, kdežto Ego-zones pro větší síť nalezlo více komunit, což je očekávaný výsledek.

Nyní se podíváme na průměrné velikosti komunit (tabulka 6.2), počet komunitních asociací (tabulka 6.3) a počet uzlů bez komunitních asociací (tabulka 6.4), které mezi sebou souvisí. Komunitní asociace je vztah mezi uzlem a komunitou, znamenající, že uzel patří do komunity. Uzlem bez asociací myslíme uzel, jež není součástí žádné komunity, tedy nemá žádné komunitní asociace. Jelikož hledáme překrývající se komunity, tak vrchol může být asociován s více komunitami.

	Amazon	**DBLP**	**Youtube**
Ground-truth	30,227	53,411	7,885
Ego-zones	7,716	13,684	6,802
Demon	25,771	28,891	173,997

Tabulka 6.2: Průměrná velikost komunit

Pro *ground-truth* komunity, když se podíváme na Youtube, tak vidíme, že naprostá většina uzlů (95,36%) nepatří do žádné komunity. Toto odpovídá naší dřívější úvaze, že většina lidí na Youtube nevyužívá možnosti tvořit skupiny. Z hodnoty průměrné velikosti komunit je jasné, že když už nějaké komunity existují, tak jsou dosti malé v porovnání s Amazon a DBLP. Oba tyto poznatky jenom více potvrzuje počet asociací, který je pro Youtube mnohonásobně menší než pro Amazon a DBLP. Co se Amazon a DBLP týče, tak průměrná velikost je větší u DBLP, ale to není nic divného, jelikož u Amazonu jsme hledali i menší komunity. Jelikož má Amazon podstatně více *ground-truth* komunit, tak má taky více asociací než DBLP. Uzlů bez komunitních asociací není zdaleka tolik jako pro Youtube. Pro Amazon je 5,28% a pro DBLP 17,69% uzlů bez komunitních asociací.

	Amazon	DBLP	Youtube
Ground-truth	2 271 543	719 820	129 202
Ego-zones	576 348	642 565	1 354 786
Demon	528 563	436 510	1 878 469

Tabulka 6.3: Počet komunitních asociací

Ego-zones v průměru nalezlo nejmenší komunity pro všechny sítě. Zároveň jich však našlo nejvíc, což značí, že má tendenci větší komunity rozdělovat do několika menších. V Amazonu je 23,88%, v DBLP 16,08% a v Youtube 14,51% uzlů bez komunitních asociací. V porovnání s *ground-truth* jich tedy je pro Amazon několikrát více, pro DBLP stejně a pro Youtube mnohonásobně méně.

	Amazon	DBLP	Youtube
Ground-truth	17 669	56 082	1 082 215
Ego-zones	79 958	51 000	164 719
Demon	69 775	138 188	873 160

Tabulka 6.4: Počet uzlů bez komunitních asociací

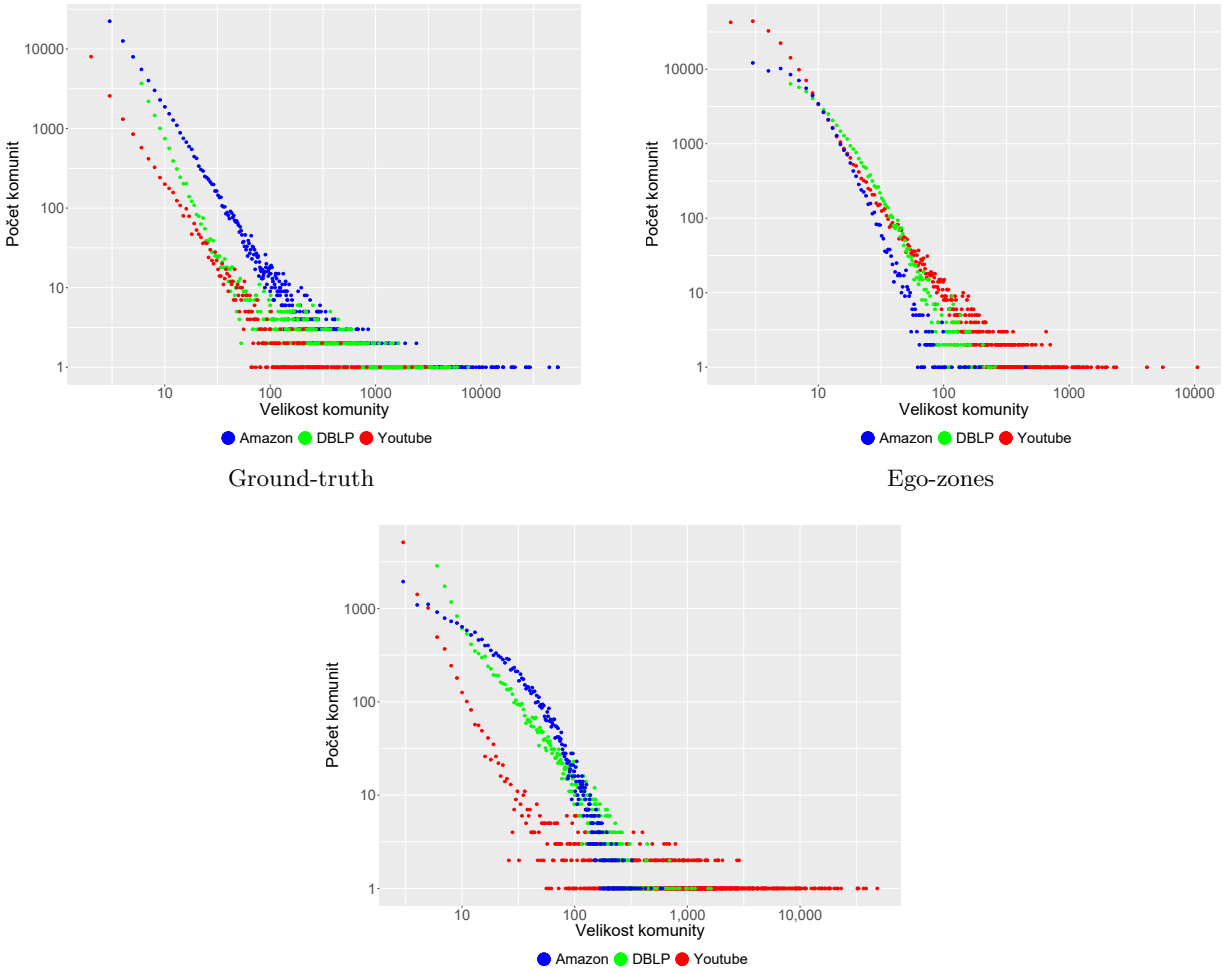
Pro *DEMON* jsme dosáhli zajímavých výsledků. U sítě Youtube dosáhl nekompromisně největší průměrné velikosti komunit, přibližně 22 krát více než *ground-truth*. Zároveň však stále zůstalo 76,94% uzlů bez komunitních asociací, což je podstatně blíže ke *ground-truth*, než jsme dostali pro *Ego-zones*. Toto je vidět i na počtu asociací, kterých je 14,5 krát větší než pro *ground-truth*. Z těchto údajů můžeme soudit, že v Youtube má *DEMON* tendenci spojovat menší komunity do větších a zároveň se tyto komunity hodně překrývají. U ostatních sítí se mu nepovedlo dosáhnout stejné průměrné velikosti jako *ground-truth*, ale jsou několikrát větší než ty získané pomocí *Ego-zones*.

6.2 Distribuce velikosti komunit

Podíváme se na distribuci velikosti komunit. Pro jednotlivé metody jsou zobrazeny na obrázku 6.1 a pro jednotlivé sítě na obrázku 6.2.

Můžeme si všimnout, že *ground-truth* má mocninné rozdělení komunit pro všechny sítě. Amazon má pro všechny velikosti nejvíce komunit a má hodně komunit, které jsou mnohem větší než největší komunity ostatních sítí. Tyto velké komunity v Amazonu lze vysvětlit tím, že pod některé asi více obecné kategorie spadá obrovské množství produktů v porovnání s těmi specifickými. Pro Youtube platí přesný opak, má pro většinu velikostí nejmenší počet komunit a jeho největší komunity jsou menší v porovnání s největšími komunitami ostatních sítí. A to i přestože velikostí je asi třikrát větší, ale jak jsme zjistili z tabulky 6.4, tak je to způsobeno tím, že naprostá většina uzlů je bez komunitních asociací. Toto znamená, že DBLP počtem i velikostí komunit leží mezi Amazon a Youtube.

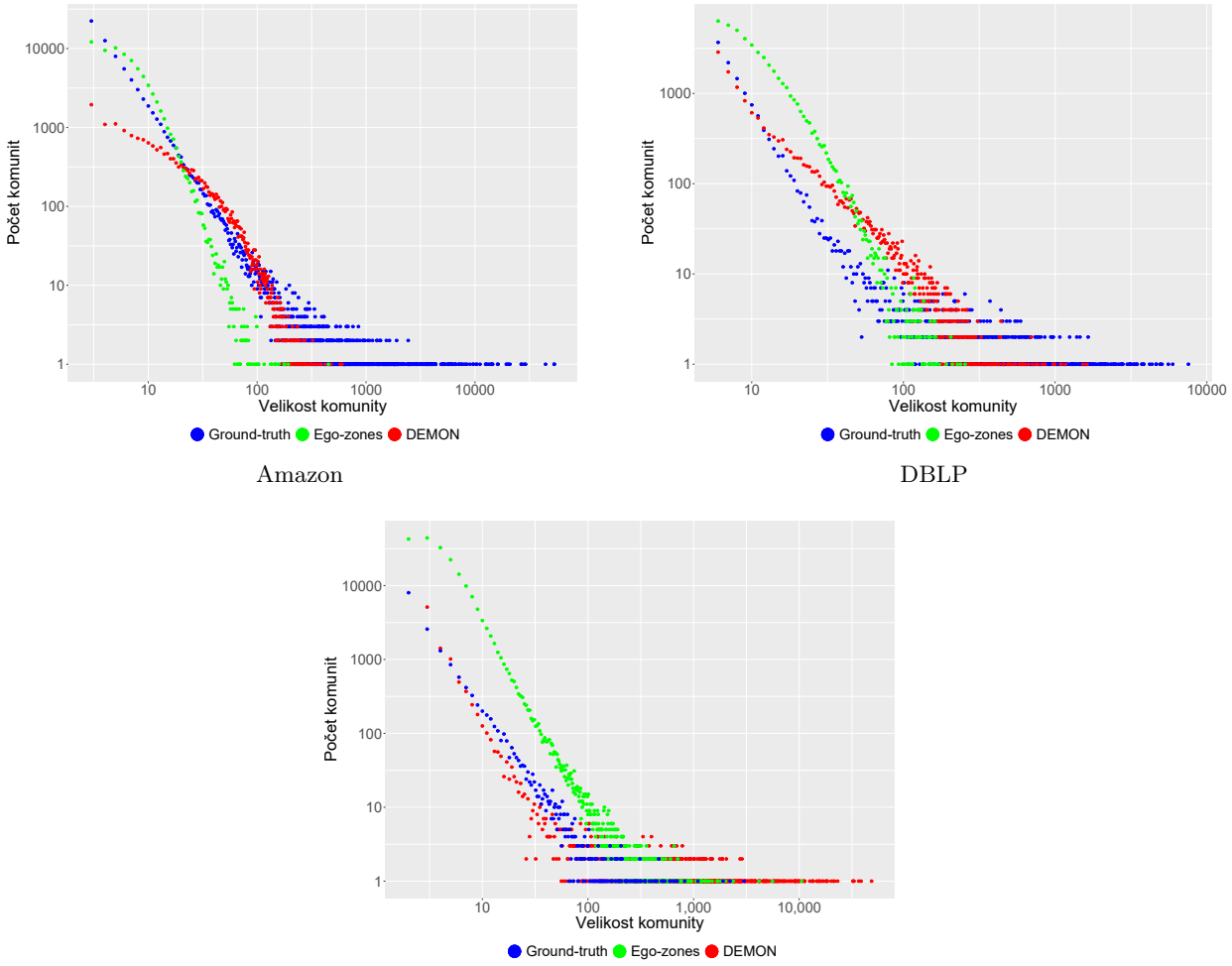
Pro *Ego-zones* vidíme zajímavější průběh. Opět je to podobné mocninnému rozdělení, jenom pro nižší hodnoty se trochu liší. Oproti *ground-truth* komunitám vidíme, že nejvíce komunit má většinou



Obrázek 6.1: Distribuce velikosti komunit pro různé metody v logaritmickém měřítku

Youtube, i když mezi velikostmi 10 a 100 má více komunit DBLP. Youtube má taky mnohonásobně větší největší komunity než ostatní sítě. Naopak Amazon je tentokrát síť s nejmenším počtem komunit pro dané velikosti, ale dosahuje maximální velikosti, která je větší než pro DLBP, avšak jedná se o výjimečný případ.

Z *DEMON* jsme dostali vůbec nejzajímavější výsledky. Pro DBLP a Youtube vidíme mocninné rozdělení, ale Amazon se zásadně liší. I když pro Amazon se počet komunit s jejich velikostí snižuje, tak k tomu nedochází tak prudce, jak by se očekávalo u klasického mocninného rozdělení. Díky tomuto je Amazon mezi velikostmi 10 a 100 síť s největším počtem komunit. Youtube je stejně jako pro *ground-truth* síť s nejmenším počtem komunit, ale dosahuje největších velikostí komunit. Pomocí *DEMON* se nám ale pro Youtube nepovedlo najít žádné komunity velikosti 2. DBLP má pro velikosti menší než 10 více komunit než Amazon a komunity větší než 100, jsou na tom podobně, ale DBLP dosahuje větších maximálních velikostí.



Obrázek 6.2: Distribuce velikosti komunit pro různé sítě v logaritmickém měřítku

Když porovnáme metody detekce komunit mezi sebou (obrázek 6.2), tak pro Amazon vidíme, že pro menší velikosti má *Ego-zones* podobný spíše větší počet než *ground-truth*, ale od velikosti 19 začíná nacházet menší počet a není zdaleka schopné najít komunity maximálních velikostí *ground-truth* komunit. *DEMON* nacházel mnohem méně komunit menších velikostí než zbylé dvě metody. Pro větší velikosti než 22 nachází *DEMON* podobný počet jako *ground-truth*, ale také není zdaleka schopen najít velikosti rovny těm největším *ground-truth* komunitám. Největší velikost komunity pro *ground-truth* je 53551 (otázkou je, zda skupiny této velikosti lze v kontextu detekčních metod ještě považovat za komunitu), pro *Ego-zones* jenom 451 a pro *DEMON* jenom 600.

Pro DBLP *Ego-zones* nalezlo o několik tisíc více komunit pro menší velikosti než ostatní metody. *DEMON* tentokrát pro malé velikosti našel jenom o trochu menší počet komunit, než je *ground-truth* komunit, ale od velikosti 12 i *DEMON* začíná nacházet větší počet než *ground-truth* a od velikosti 54 i více než *Ego-zones*. Kolem velikosti 100 *Ego-zones* a *ground-truth* mají počty komunit už

jenom v jednotkách, kdežto *DEMON* stále nachází více než 10 komunit pro tyto velikosti. Největší velikosti, které metody nalezly, jsou pro *ground-truth* 7556, *Ego-zones* 253 a *DEMON* 1617, takže je vidět, že *Ego-zones* nedokázalo najít větší komunity, a ani *DEMON* nenašel zdaleka komunity rovny velikostem *ground-truth* komunit.

Největší síť Youtube je stejná jako ostatní sítě v tom, že *Ego-zones* nalezlo mnohem více menších komunit než ostatní metody. *DEMON* nachází pro menší velikosti podobné počty komunit jako *ground-truth*. Kdežto *ground-truth* a *DEMON* nalézají jenom jednotky komunit už od hodnoty 77, tak *Ego-zones* se pod počet komunit 3 dostane poprvé na velikosti 119, ale i poté nachází větší počet komunit než další metody. Největší velikost komunity pro *ground-truth* je tentokrát 3001, pro *Ego-zones* 10538 a pro *DEMON* až 48460. *DEMON* v tomto případě nelezl jednoznačně největší komunity.

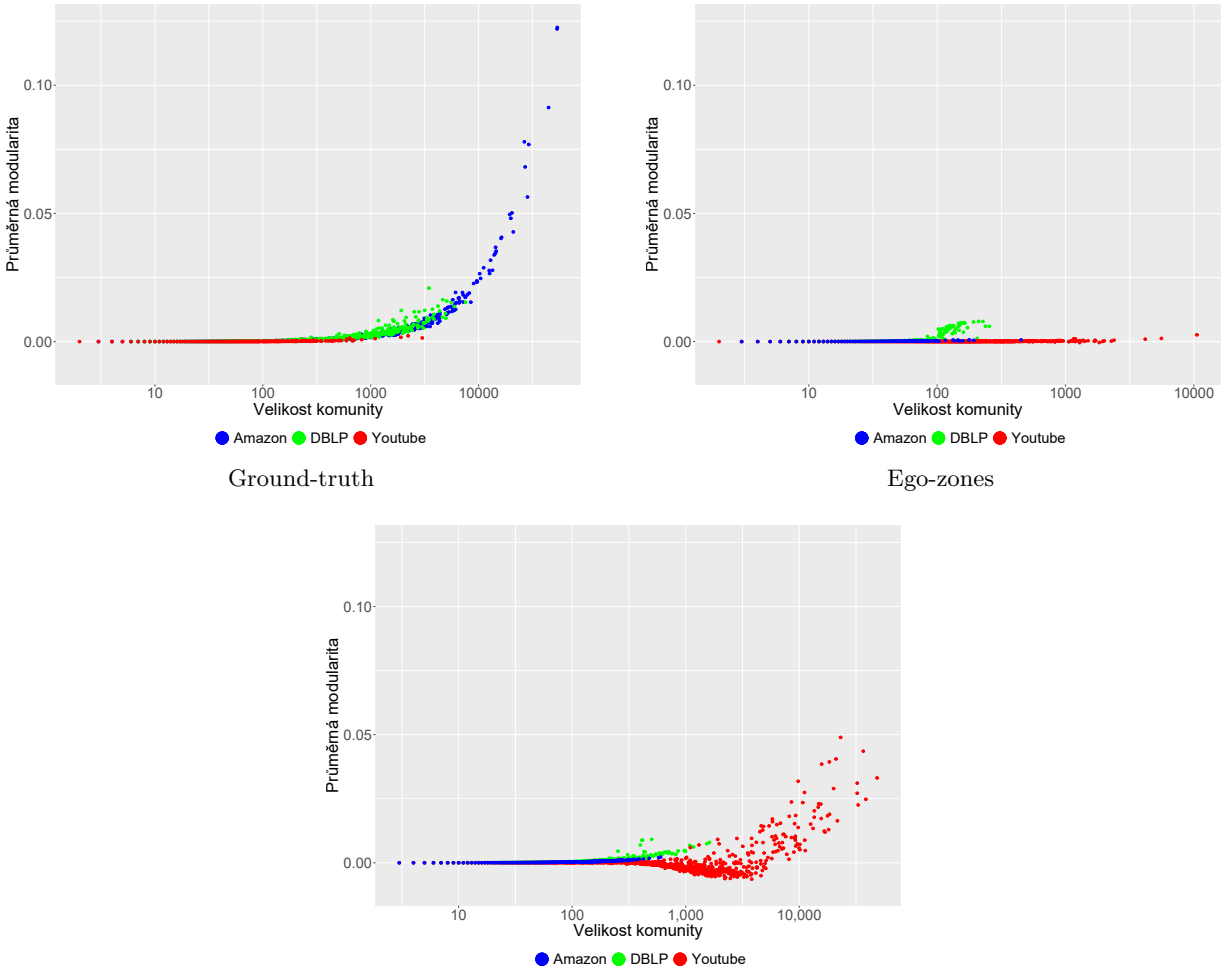
Pokud to shrneme, tak je trochu obtížné porovnávat podle velikostí *ground-truth* komunity s těmi nalezenými pomocí metod detekce komunit, protože k jejich určení používáme informace, které nelze vyčíst ze struktury sítě. Nejvíce je toto patrné na síti Youtube, kde je pro *ground-truth* obrovské množství uzlů bez komunitních asociací, jelikož málo lidí tvoří skupiny na této sociální síti. Takže i když lidé navzájem tvoří komunity tím, že se navzájem znají a jsou přátelé, tak nikdy nevytvoří skupinu, tedy algoritmy pro detekci komunit naleznou větší množství komunit. Opačný problém je možné vidět u sítě Amazon a trochu i u DBLP. U Amazonu zřejmě existují obrovské obecné kategorie, kde výrobky mezi sebou ani nemusí mít moc společného a lidé je tedy nekupují moc dohromady. Tyto komunity pak algoritmy detekce komunit nenajdou. U DBLP by něco podobného mohly být populární vědecké časopisy, do kterých přispívá velké množství lidí, ale kde většina z nich v životě nevydala článek dohromady. Pokud bychom porovnávali *Ego-zones* a *DEMON*, tak *Ego-zones* je spíše vhodnější, pokud bychom chtěli najít větší počet menších komunit a pokud chceme menší počet komunit větších velikostí, tak použijeme *DEMON*.

6.3 Míry

Pro ohodnocení kvality jednotlivých komunit byly použity míry popsané v kapitole 4. Porovnáváme naměřené míry pro jednu metodu detekce komunit pro různé sítě a potom naopak pro jednu síť porovnáme výsledky různých metod.

6.3.1 Modularita

Začneme modularitou, jelikož je to klasická míra pro měření kvality komunit. Modularitu pro komunitu počítáme tak, jak bylo řečeno v sekci 4.1. Počítáme modularitu pro dvě komunity, kde jedna je komunita, pro níž chceme spočítat modularitu a druhá celý zbytek sítě. Na obrázku 6.3 můžeme vidět, jakou průměrnou modularitu dostáváme pro různé metody.



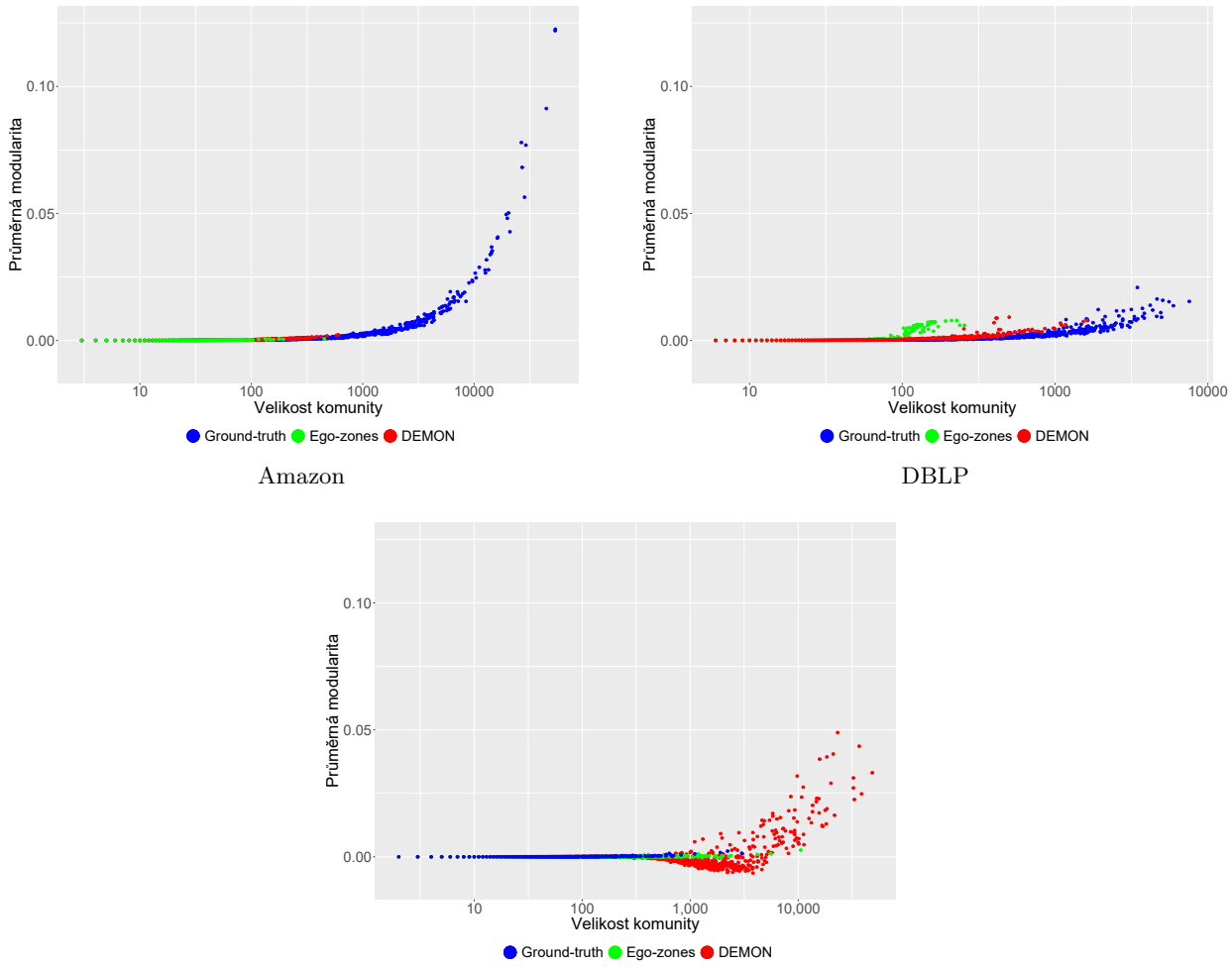
Obrázek 6.3: Průměrná modularita pro různé metody

Vidíme, že pro *ground-truth* má graf modularity pro různé sítě stejný průběh. Pro menší komunity je velice nízká s hodnotou blízké 0 a s velikostí komunit se její hodnota zvyšuje až na Youtube, jehož průměrná hodnota modularity se nikdy nevzdálí od 0.

Pro *Ego-zones* vidíme, že modularita se nikdy nevzdálí od hodnoty 0 a maximální hodnotu dostáváme pro DBLP, jež je rovna přibližně 0,0079. Toto je však pochopitelné, jelikož jak jsme zjistili v sekci 6.2, tak *Ego-zones* neumí moc dobře najít vetší komunity, které by dosahovaly vyšších hodnot modularity. Pokud se podíváme znovu na *ground-truth*, tak pro stejné velikosti se od *Ego-zones* moc neliší, akorát existuje spousta větších *ground-truth* komunit pro sítě Amazon a DBLP. Můžeme vidět, že pro DBLP *Ego-zones* najde komunity s lepší modularitou.

DEMON má pro Amazon a DBLP velice podobný průběh jako ostatní metody. Pro Youtube však můžeme vidět zajímavou věc, mezi velikostmi 300-4000 hodnoty modularity začínají klesat a dostávají se dokonce většinou do záporných hodnot, poté však začne modularita znovu prudce

růst. Maximální hodnota průměrné modularity byla 0,049, kterou získal *DEMON* pro komunitu v síti Youtube.



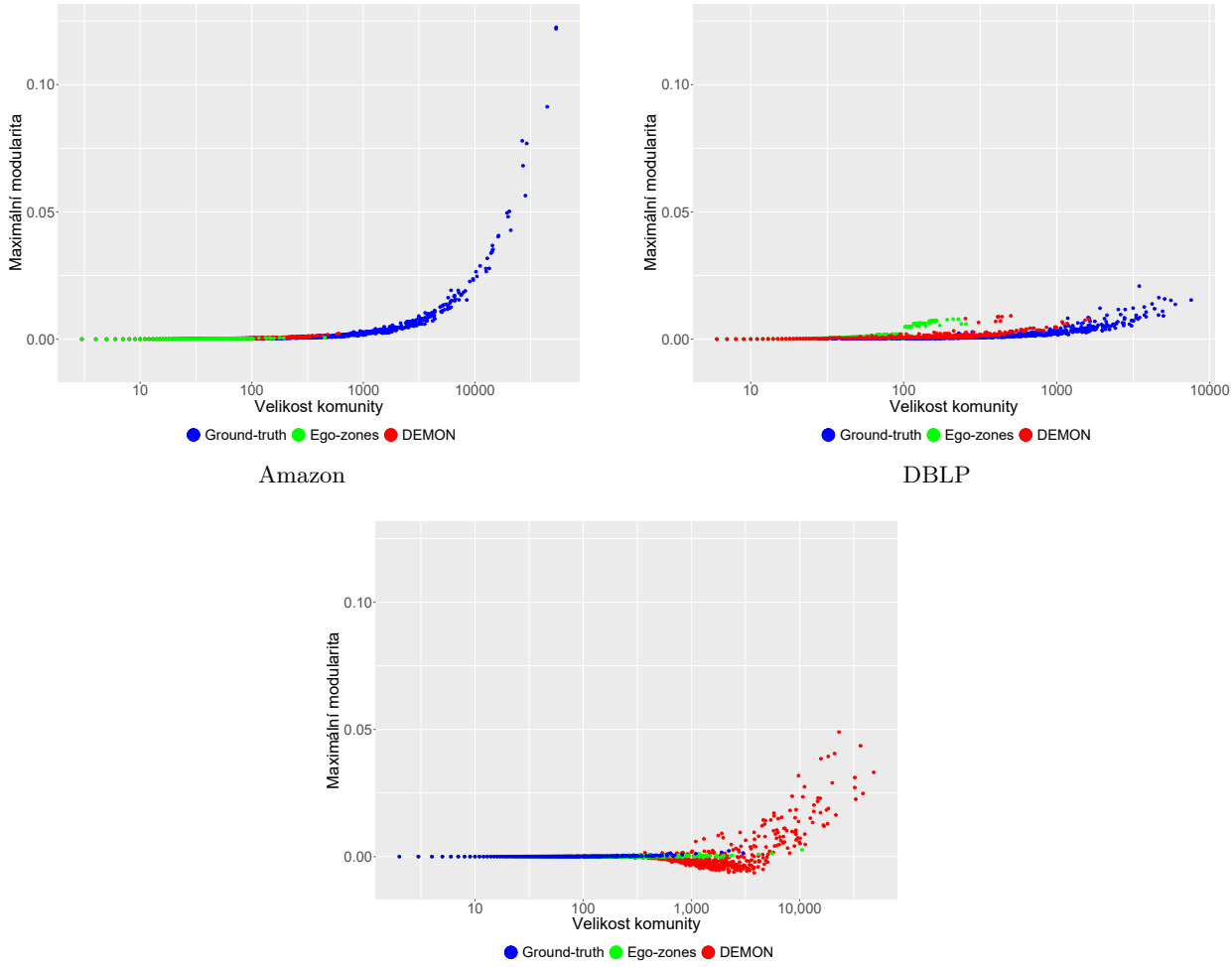
Obrázek 6.4: Průměrná modularita pro různé sítě

Porovnání jednotlivých metod pro jednotlivé sítě je na obrázku 6.4. Pro Amazon nevidíme nic zajímavého, modularita roste s velikostí komunit a záleží jenom na tom, jak velké komunity metoda byla schopna najít.

DBLP je o trochu zajímavější, opět modularita roste s velikostí, avšak vidíme, že rychlost růstu modularity se liší. *Ego-zones* sice nenašlo větší komunity, ale komunity kolem velikosti 100 byly mnohem lepší v porovnání s ostatními metodami. Pro DBLP měly *ground-truth* komunity nejhorší průměrnou modularitu, jelikož pro stejnou velikost byl lepší *DEMON* a *Ego-zones* bylo většinou ještě lepší než *DEMON*.

Pro Youtube vidíme to samé, co z obrázků předtím a to, že pro *DEMON* nějakou dobu průměrná modularita i klesá, jinak s velikostí modularita roste pro všechny metody podobně.

Průměrné hodnoty však nemusí vypovídat o všem. Jelikož pokud je komunit větší množství, jež se hodně liší hodnotami míry, tak nevidíme, že metoda najde i hodně dobré komunity. Ty špatné ale při analýze nemusíme brát v potaz a dále pracujeme jenom s těmi lepšími. Proto na obrázku 6.5 jsou zobrazeny pouze maximální hodnoty pro dané velikosti komunit. Vidíme však, že grafy se moc neliší od těch s průměrnou modularitou, takže většina komunit nabývá hodnot velice blízkých průměru.



Amazon

DBLP

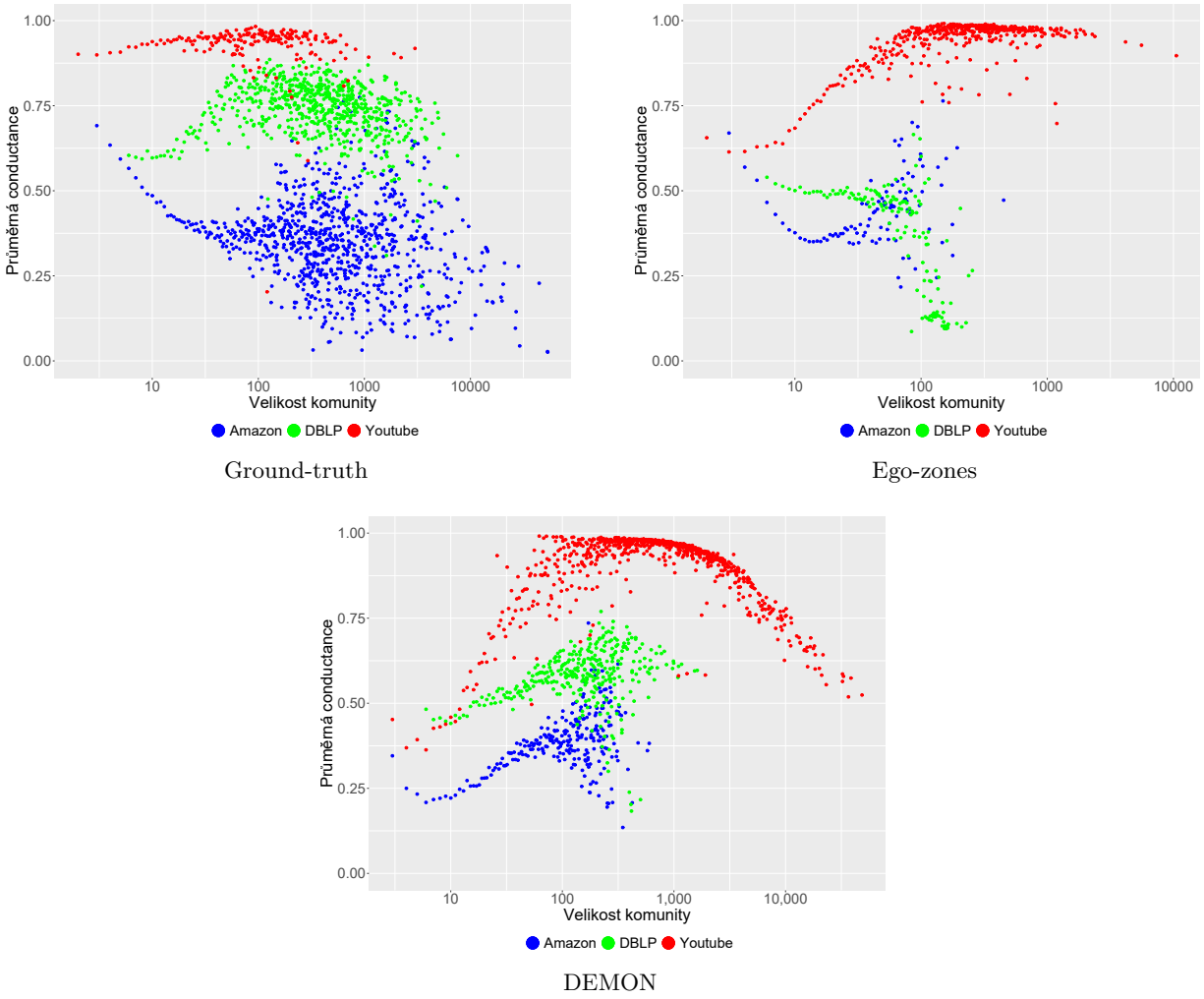
Youtube

Obrázek 6.5: Maximální modularita pro různé sítě

Pokud bychom to měli shrnout, vidíme, že s velikostí komunit většinou roste i modularita. Modularita nebude nabývat vysokých hodnot, pokud komunita tvoří pouze malou část sítě, což je pochopitelné. Hodnoty modularity pro jednu kombinaci velikosti, sítě a metody dosahují velice podobných hodnot, tedy jsou všechny velice blízké průměru. Metody detekce komunit *Ego-zones* a *DEMON* dosahují často lepší modularity než *ground-truth* komunity, což lze očekávat, jelikož modularita stejně jako *Ego-zones* a *DEMON* pracuje se strukturou sítě na rozdíl od *ground-truth* komunit.

6.3.2 Vodivost

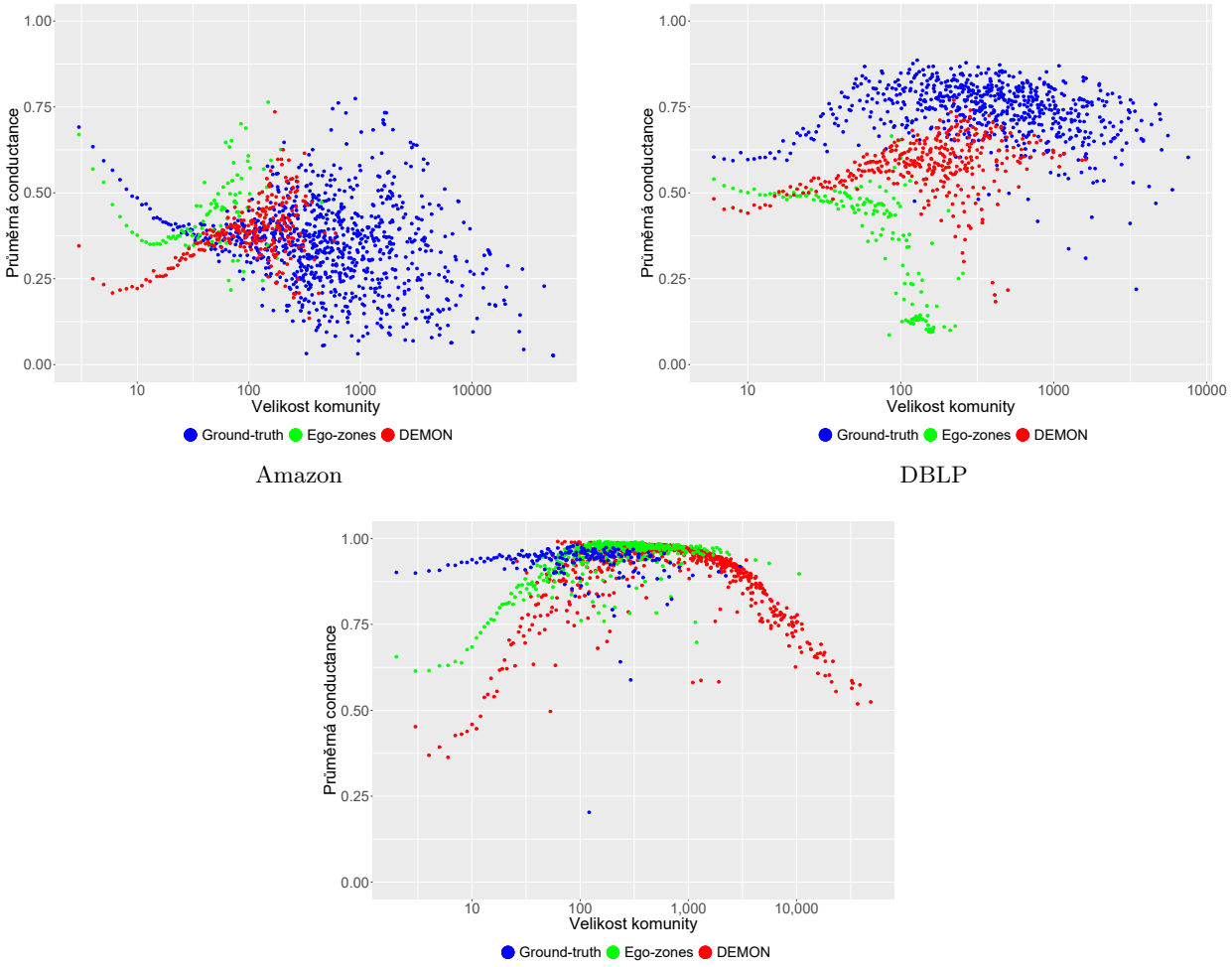
Nyní se podíváme na vodivost (sekce 4.2). Zde na rozdíl od modularity platí čím menší vodivost tím lepší komunita. Nejdříve porovnáme mezi sebou průměrné vodivosti jednotlivých sítí pro různé metody, jež vidíme na obrázku 6.6.



Obrázek 6.6: Průměrná vodivost pro různé metody

Když se podíváme na *ground-truth*, tak můžeme vidět, že Amazon měl nejlepší, DBLP druhou nejlepší a Youtube nejhorší vodivost pro naprostou většinu velikostí, avšak existuje několik výjimek. Nemůžeme však říct, že by průměrné hodnoty vodivosti byly závislé na velikosti komunit.

U *Ego-zones* a *DEMON* vidíme stejnou věc. Amazon je nejlepší a Youtube nejhorší až na pár výjimek. Jedině u *Ego-zones* dochází k tomu, že kolem velikosti 100 jsou hodnoty vodivosti pro Amazon a DBLP podobné. Zajímavější je porovnávat rozdíly mezi metodami detekce komunit, pro což jsou grafy zobrazeny na obrázku 6.7.



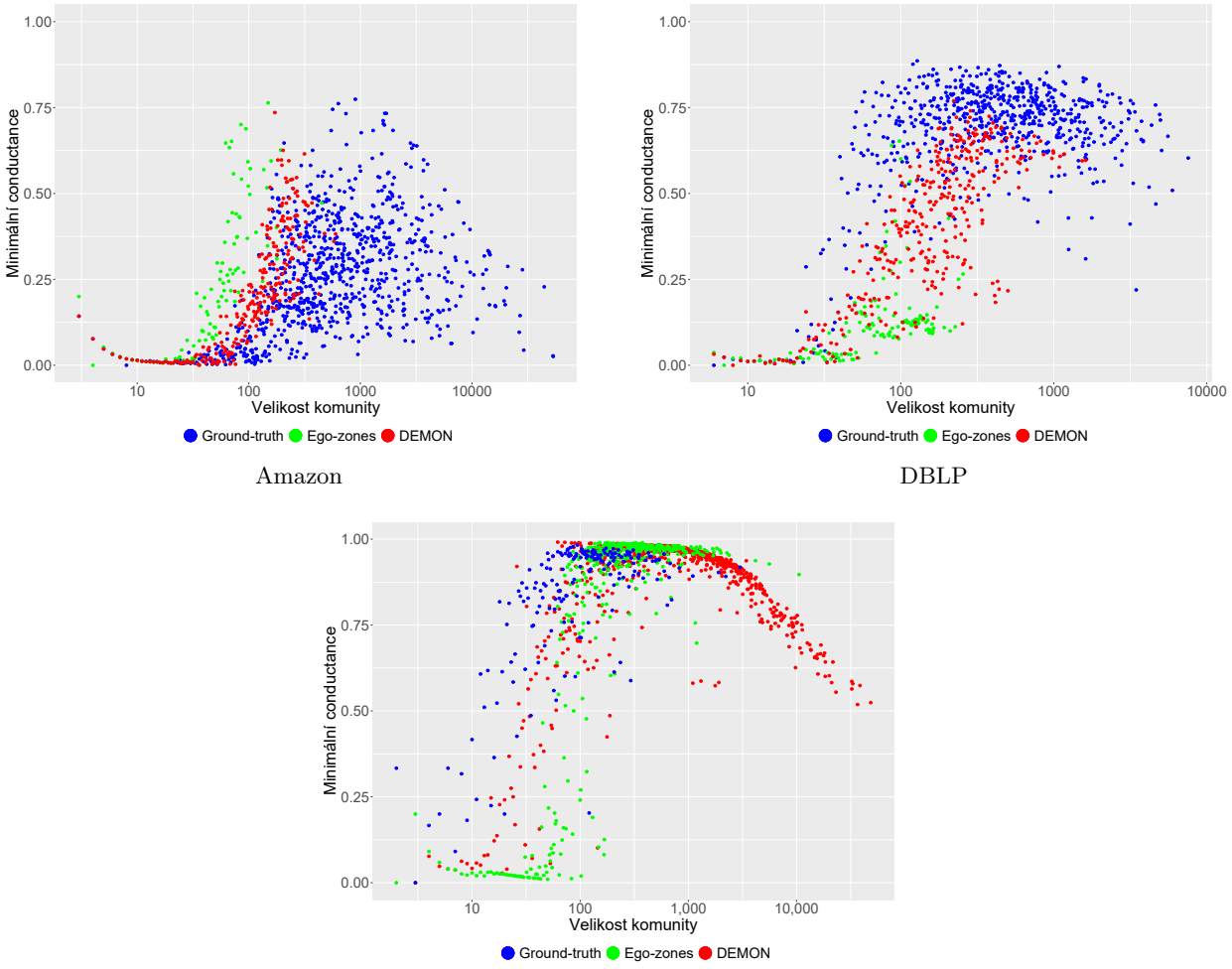
Obrázek 6.7: Průměrná vodivost pro různé sítě

Pro Amazon vidíme, že pro malé velikosti je nejlepší *DEMON* a *ground-truth* komunity jsou naopak nejhorší. Pro všechny metody se potom vodivost začne blížit k hodnotě 0,38 a všechny ji dosáhnou kolem velikosti 40. Pro větší velikosti komunit hodnoty vodivosti všech metod kolísají kolem této hodnoty a s rostoucí velikostí se zvětšují i rozdíly od této hodnoty.

Pro DBLP vidíme odlišný vývoj. Sice opět platí, že pro menší velikosti je *ground-truth* nejhorší a *DEMON* nejlepší, ale tentokrát se vodivost neblíží k žádné hodnotě. *Ground-truth* je nejhorší pro skoro všechny velikosti. Vodivost *ground-truth* a *DEMON* se s rostoucí velikostí zhoršuje, kdežto *Ego-zones* se s velikostí zlepšuje. Díky tomu má *Ego-zones* od velikosti 26 lepší vodivost než *DEMON*.

U Youtube vidíme podobné znaky jako u Amazonu. Opět pro menší velikosti je nejlepší *DEMON* a nejhorší *ground-truth*. Stejně jako u Amazonu se hodnota vodivosti postupně blíží k jedné hodnotě, která je přibližně 0,95, a všechny metody ji dosáhnout kolem velikosti 100. Na rozdíl od Amazonu se ale vodivost komunit pro velikosti větší než 1000 začíná zase vzdalovat od této hodnoty. Nejvíce

je to zřetelné pro *DEMON*. Toto znamená, že podle vodivosti jsou dobré malé a větší komunity, ale ty středních velikostí už moc ne.



Obrázek 6.8: Minimální vodivost pro různé sítě

Stejně jako u modularity se podíváme na nejlepší hodnoty vodivosti, což jsou nyní minima, a grafy jsou na obrázku 6.8. Když porovnáme průměrné s minimálními, tak nyní vidíme výrazné rozdíly. Obecně platí, že pro malé velikosti máme komunity, které se výrazně liší od průměru, což značí, že existují taky komunity, které jsou mnohem horší než průměr. Pro větší velikosti, kdy už dostáváme počty komunit v jednotkách, tak se grafy podobají průměru.

Pro Amazon se v porovnání s průměrnými hodnotami teď situace zásadně změní. Pro menší velikosti není vidět výrazný rozdíl a pro větší velikosti vychází nejlépe *ground-truth*, nejhůře *Ego-zones* a *DEMON* je něco mezi nimi.

DBLP se moc neliší od Amazonu, co se pořadí metod týče. Na začátku jsou si stejně, jako u Amazonu, všechny metody rovny, ale pro větší komunity dostáváme stejné pořadí jako pro prů-

měrné hodnoty vodivosti. Tedy *Ego-zones* je nejlepší a *ground-truth* nejhorší, i když rozdíly mezi metodami už nejsou tak veliké.

Pro Youtube pak platí, že komunity o velikostech větších než 100 se minima neliší moc od průměru. Oproti průměrným hodnotám však vidíme rozdíly pro menší velikosti. *Ego-zones* je nejlepší téměř ve všech případech, a to s velkým rozdílem. Toto je pochopitelné, jelikož nachází velké množství menších komunit, takže je pravděpodobné, že některá z nich bude dosahovat lepší hodnoty, než ostatní metody. I přesto je však zajímavé, že *ground-truth* se nepřiblížilo k jeho hodnotám skoro v žádném případě a *DEMON* v méně než polovině případů.

Pokud bychom to měli shrnout, tak výsledky se výrazně liší mezi jednotlivými sítěmi. Většinou je průměrná vodivost pro danou velikost nejlepší pro *DEMON* a nejhorší pro *ground-truth* v případě, že se hodnoty výrazně liší. Výjimkou je však síť DBLP, kde pro hodně velikostí vyšlo nejlépe *Ego-zones*, a to s výrazným rozdílem. Co se nejlepších hodnot týče, tak opět se hodně liší mezi sítěmi a pro menší velikosti se hodně liší od průměru. Pro DBLP a Youtube našlo nejlepší komunity *Ego-zones* a nejhorší byly *ground-truth* komunity a u Amazonu to bylo přesně naopak.

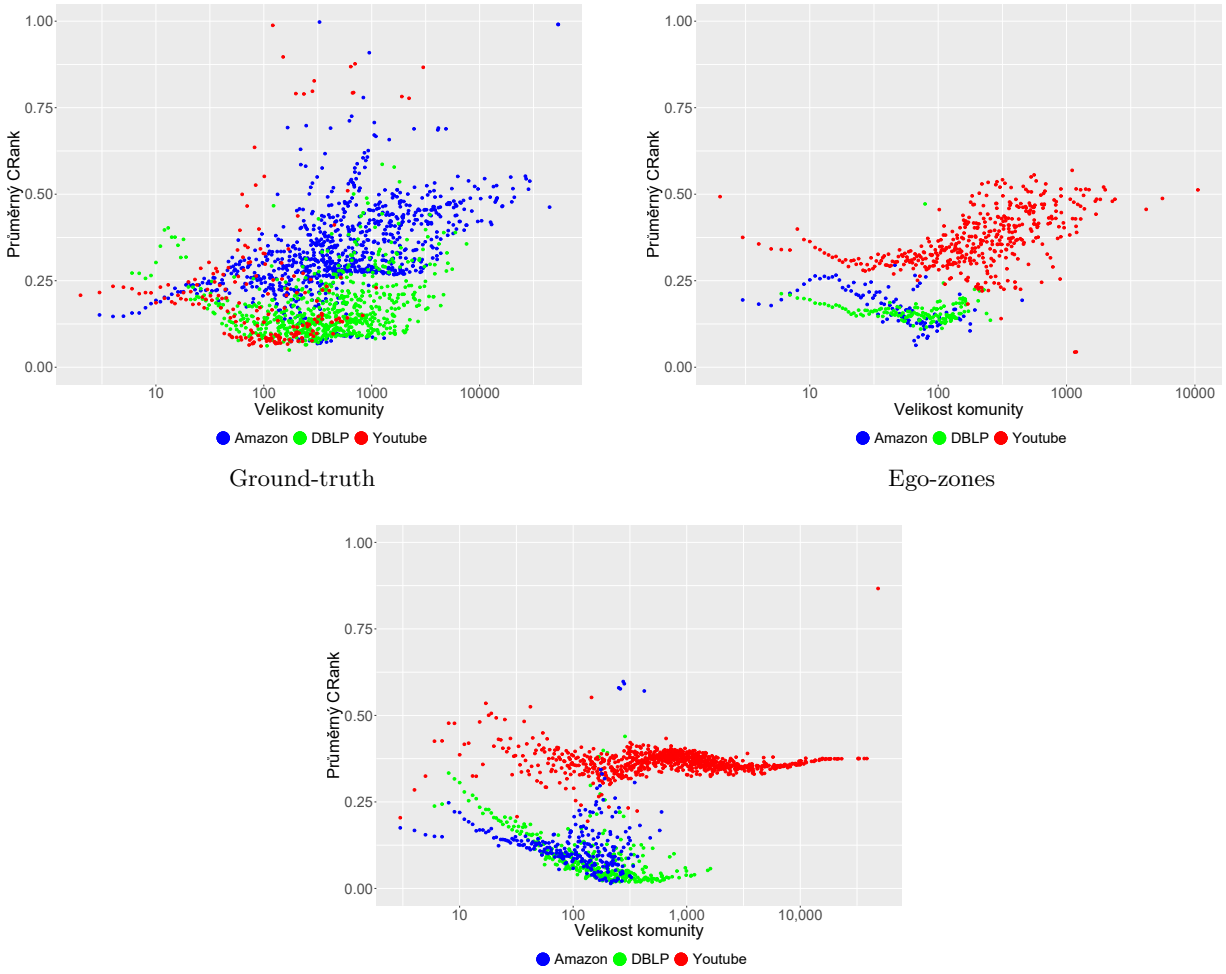
6.3.3 CRank

Nakonec se podíváme na CRank (sekce 4.3), což je nejkomplexnější míra, takže by měla mít nejvíce vypovídající hodnotu.

Nejdřív se zběžně podíváme, jak se liší výsledky mezi jednotlivými sítěmi (obrázek 6.9). Pro *ground-truth* komunity to vypadá, že pro různé sítě dostáváme různé výsledky, ale komunity pro Amazon většinou dosahují nejlepšího hodnocení a pro Youtube nejhoršího. Můžeme taky vidět množství velikostí komunit, které dosahují buď velice dobrých či velice špatných výsledků. Toto je většinou způsobeno tím, že jedná o velikosti, pro které byla nalezena jenom jedna komunita. To nám taky prozrazuje, že hodnoty CRanku hodně kolísají a často bývají hodně vzdálené od průměru. Když se podíváme na ostatní velikosti, tak vidíme, že pro *ground-truth* dostáváme často hodnoty pod 0,5, jelikož většina průměrů je pod touto hodnotou.

Pro *Ego-zones* a *DEMON* dostáváme podobné výsledky lišící se od *ground-truth*. Youtube dosahuje nejlepších hodnot CRanku. Amazon a DBLP průměrné hodnoty CRanku si jsou pro stejné velikosti komunit podobné. V porovnání s *ground-truth* vidíme jen velice málo velikostí, které by se zásadně lišily od zbytku a dosahovaly hodně nízkých nebo hodně vysokých hodnot, přestože taky často existuje jenom jedna komunita dané velikosti.

Pokud se podíváme na rozdíly mezi průměry jednotlivých metod (obrázek 6.10) získáme zajímavější informace. Když je porovnáme pro síť Amazon, tak vidíme, že *ground-truth* komunity jsou jednoznačně nejlepší pro větší velikosti a pro menší jsou si metody vcelku podobné. S rostoucí velikostí kvalita komunit pro *ground-truth* roste a vidíme ony hodně nízké nebo hodně vysoké hodnoty, kterých jsme si už všimli na obrázku 6.9. *Ground-truth* v některých případech dosahuje i nejvyšší možné hodnoty 1. Metody *Ego-zones* a *DEMON* našly spíše horší komunity než *ground-truth* a s ve-

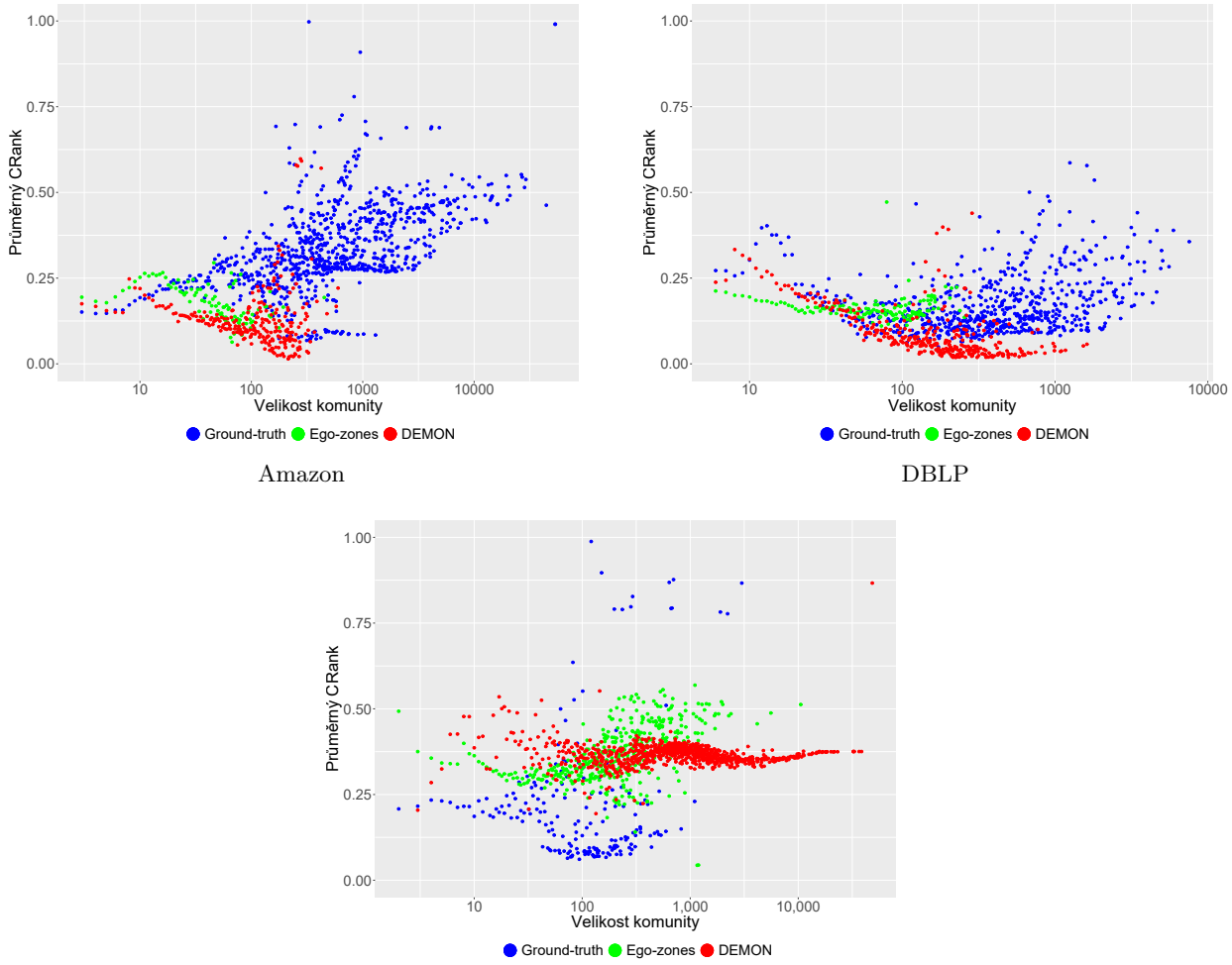


Obrázek 6.9: Průměrný CRank pro různé metody

likostí se kvalita jejich komunit zhoršuje. Pro *DEMON* vidíme pár dobrých, ale i hodně špatných hodnot CRanku pro větší velikosti.

Pro DBLP vidíme, že menší komunity jsou nejhorší pro *Ego-zones* a s rostoucí velikostí se pro něj hodnota CRanku téměř nemění. Oproti tomu hodnoty CRanku pro *DEMON* s velikostí klesají. Proto pro komunity větší než 30 dosahuje *Ego-zones* lepších výsledků než *DEMON*, který je od velikosti 30 většinou nejhorší. *DEMON* dosahuje docela špatných hodnot, které jsou velice blízké 0. Opět vidíme několik vyšších hodnot, které se liší od zbytku pro všechny metody. Především se vyskytují pro *ground-truth*, několik pro *DEMON* a jedna pro *Ego-zones*. Nejlepší průměrné hodnoty se blíží jenom hodnotě 0,6, což není ani zdaleka rovno 1, již jsme dostali pro Amazon.

Youtube je podobné DBLP tím, že pro menší velikosti je *DEMON* lepší a Ego-zones je lepší pro ty větší. Rozdíl je ten, že *DEMON* se nyní udržuje na stejné hodnotě CRanku pro různé velikosti, kdežto *Ego-zones* komunity se s velikostí zlepšují. *Ground-truth* komunity jsou pro Youtube nejhorší

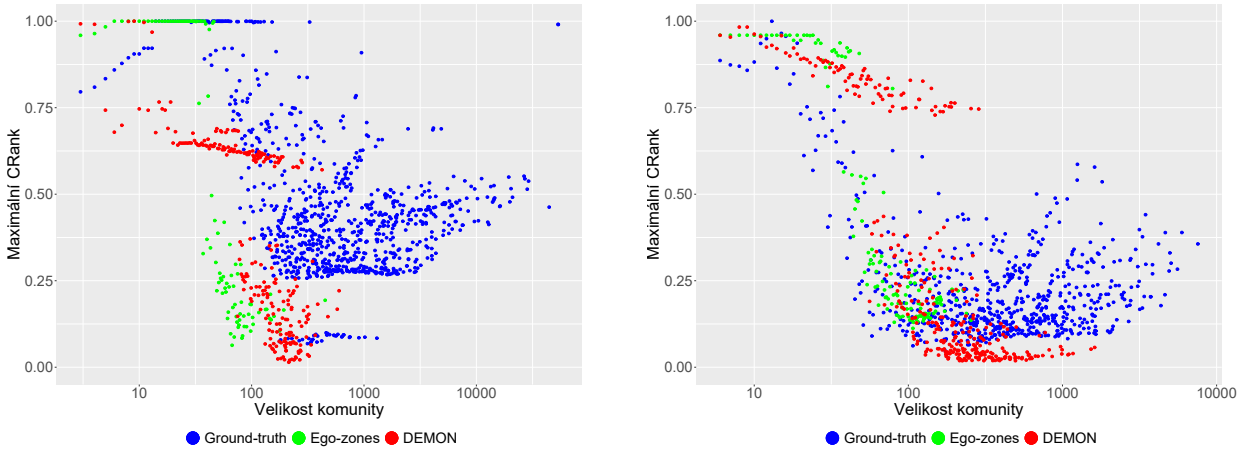


Obrázek 6.10: Průměrný CRank pro různé sítě

komunity, i když opět vidíme pár vysokých hodnot pro větší velikosti, které kvalitou předčí ostatní metody.

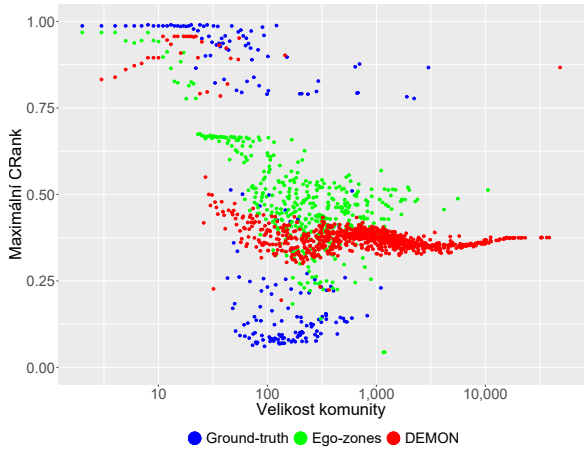
Opět se podíváme na maximální hodnoty CRanku (obrázek 6.11). U Amazonu vidíme, že metody *ground-truth* a *Ego-zones* pro velikosti mezi 10 a 100 dosahují maxima hodnoty 1. Pro *DEMON* dostáváme hodnoty 1 minimálně. Zajímavá je, že pro *DEMON* se maxima mezi velikostmi 20-74 drží kolem hodnoty 0,65 a poté dojde k velkému propadu kvality komunit. Podobnou věc vidíme i pro *Ego-zones*, akorát ty se drží kolem hodnoty 1 pro velikosti 6-36 a poté taky dochází k prudkému propadu kvality. Pro větší velikosti se pro všechny metody hodnoty podobají průměru zřejmě kvůli tomu, že to jsou už velikosti pouze s jednou nalezenou komunitou.

Pro DBLP vidíme opět, že menší komunity dosahují větších maxim, tentokrát však nedosahují hodnoty 1. Ale existuje jedna *ground-truth* komunita, které se povedlo dosáhnout na hodnotu 0,99. Menší *Ego-zones* komunity dosahují velice dobrých maximálních hodnot. *DEMON* je na tom lépe



Amazon

DBLP



Youtube

Obrázek 6.11: Maximální CRank pro různé sítě

než u Amazonu, ale stále zaostává za *Ego-zones*, i když mezi jejich hodnotami už není moc velký rozdíl. *Ground-truth* maxima jsou většinou nejhorší. Stejně jako u Amazonu vidíme prudké propady kvalit komunit u všech metod. Pro *Ego-zones* k tomuto propadu došlo u menších velikostí než pro *DEMON*. Kvůli toho byla často maxima pro *DEMON* komunity nejlepší mezi velikostmi 43-79. Pro všechny metody potom, co dojde k propadu, se maximální hodnoty CRanku pro větší velikosti komunit velice podobají průměrným hodnotám.

Pro Youtube vidíme podobné věci. Maximální hodnoty pro malé velikosti se opět blíží hodnotě 1, ale opět ji není nikdy dosaženo úplně. Pro tyto menší velikosti jsou nejlepší komunity pro *ground-truth*, což je trochu překvapující vzhledem k tomu, že průměry byly pro tuto metodu absolutně nejhorší. *DEMON* a *Ego-zones* se následně střídají na druhém místě. Následně opět dochází k prudkým propadům a potom se hodnoty pro *DEMON* a *ground-truth* hodně podobají průměru. Výjimkou je *Ego-zones*, kde dojde ke dvěma menším propadům hodnot místo jednoho.

Pro shrnutí se hodnoty CRanku opět hodně liší mezi různými sítěmi. Pro Amazon byly nejlepší *ground-truth* komunity, pro DBLP byla pro různé velikosti dobrá každá z metod a pro Youtube byly pro různé velikosti nejlepší metody *DEMON* a *Ego-zones*. I přesto pro všechny sítě maximální průměrné hodnoty dosáhly *ground-truth* komunity. Z maxim jsme vyčetli, že maximálních hodnot většinou dosahují menší komunity a že od určité velikosti dochází k značným propadům maximálních hodnot. I když maximální velikosti jsou pro menší komunity nejlepší, tak v průměru jsou lepší spíše vyšší hodnoty. Toto značí, že většina menších komunit nabývá spíše horších CRank hodnocení, avšak zároveň se občas vyskytnou i nějaké velice dobré.

Kapitola 7

Konstrukce sítí z vektorových dat

Při analýze vektorových dat často bývá výhodné zkonstruovat z nich sítě. Obvykle se konstruují vážené neorientované sítě. Každý vektor je potom reprezentován jedním uzlem a hrany značí jejich vzájemnou podobnost. Výhodou sítí je například možnost vizualizace dat, což pro vektorová data s větší dimenzí není možné. Na síť potom můžeme použít klasické postupy analýzy sítě a z výsledků si udělat úsudek o struktuře dat.

7.1 Algoritmy konstrukce sítí z vektorových dat

Konstrukce sítě z vektorových dat se skládá ze dvou kroků v následujícím pořadí: (1) výpočet podobností mezi všemi dvojicemi vektorů, přičemž výsledkem je matice podobnosti a (2) prořídnutí (sparsifikace), jehož cílem je výběr těch hran, které jsou pro konstruovanou síť důležité. Nejčastěji používanými podobnostmi během prvního kroku jsou:

Gaussian kernel podobnost

$$GaussianKernel(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$$

Gaussian kernel podobnost je podobnost založená na eukleidovské vzdálenosti mezi vektory x a y. Parametr σ se standardně nastavuje na hodnotu 1.

Kosinová podobnost

$$Cosine(x, y) = \frac{x \cdot y}{\|x\|\|y\|}$$

Kosinová podobnost je kosinus úhlu mezi vektory x a y. Je tedy založená na proporcích vektorů.

Jaccardova podobnost

$$Jaccard(x, y) = \frac{\sum_i min(x_i, y_i)}{\sum_i max(x_i, y_i)}$$

Pro vektory $x = (x_1, x_2, \ldots, x_n)$ a $y = (y_1, y_2, \ldots, y_n)$ platí $x_i, y_i \geq 0$. Jaccardova podobnost je také známá jako Růžičkova podobnost.

Všechny tyto podobnosti nabývají hodnot z intervalu $\langle 0, 1 \rangle$. Typ podobnosti vybíráme podle toho, z jakého pohledu chceme data analyzovat. Druhým krokem je prořídnutí, kde se na základě podobností vybere, které vrcholy budou navzájem propojeny. Klasickými postupy jsou:

ε-radius Parametrem je velikost podobnosti ε. Pokud je podobnost mezi dvojicí vektorů větší nebo rovna ε, pak je v síti mezi nimi hrana.

k-Nearest Neighbors (KNN) Parametr k určuje, ke kolika nejpodobnějším vrcholům je každý vrchol připojen.

Combination Nejdřív se použije ε-radius a pokud má vrchol potom méně sousedů než k, je použito KNN pro jejich nalezení.

Pro konstrukci sítí pomocí těchto klasických metod jsme vytvořili aplikaci, která je popsána v příloze B.

7.1.1 LRNet [17]

Algoritmus LRNet pro konstrukci vážené sítě s využitím místní reprezentativnosti se skládá ze čtyř kroků:

1. Vytvořte matici podobnosti **S** datového souboru **D**.
2. Vypočítejte reprezentativnost všech objektů $\boldsymbol{O_i}$.
3. Vytvořte množinu **V** uzlů sítě **G** tak, že uzel $\boldsymbol{v_i}$ sítě **G** představuje objekt $\boldsymbol{O_i}$ datové sady **D**.
4. Vytvořte množinu hran **E** sítě **G** tak, aby **E** obsahoval hranu $\boldsymbol{e_{ij}}$ mezi uzly $\boldsymbol{v_i}$ a $\boldsymbol{v_j}$ $(i \neq j)$, pokud $\boldsymbol{O_j}$ je reprezentativní soused $\boldsymbol{O_i}$. Uzly přidávají různé počty hran.

Vzorce pro výpočet reprezentativnosti lze nalézt v [17]. Pro konstrukci pomocí metody LRNet jsme použili aplikaci vytvořenou Milošem Kudělkou, která je dostupná ke stažení na [18].

Kapitola 8

Účelové funkce kvality konstrukce

Problémem je, že bývá těžké posoudit, zda struktura získané sítě dobře odpovídá vektorovým datům. K tomu budeme využívat účelovou funkci.

8.1 Daitchova účelová funkce

Předtím než vysvětlíme způsob, jakým budeme my hodnotit kvalitu konstrukce sítě, je nutno popsat tzv. **Daitchovu účelovou funkci** [19], jelikož námi použitá účelová funkce z ní vychází. Hodnota Daitchovy účelové funkce je určena tímto vzorcem:

$$\sum_{k=1}^{d}\sum_{i=1}^{n}(x_i^k - est_i^k)^2$$

kde $\boldsymbol{n}$ je počet vektorů a $\boldsymbol{d}$ dimenze datasetu. $\boldsymbol{x_i^k}$ je pak hodnota v datasetu na $\boldsymbol{i}$-tém řádků a $\boldsymbol{k}$-tém sloupci. $\boldsymbol{est_i^k}$ je rovno:

$$est_i^k = \frac{1}{d_i}\sum_j w_{i,j}x_j^k$$

kde $\boldsymbol{d_i}$ je vážený stupeň vrcholu $\boldsymbol{i}$ a $\boldsymbol{w_{i,j}}$ je váha hrany mezi vrcholy $\boldsymbol{i}$ a $\boldsymbol{j}$.

Daitch využívá tuto účelovou funkci v jeho metodě konstrukce sítě z vektorových dat, při které se snaží minimalizovat její hodnotu. Při výpočtu hodnoty této funkce se pro každý uzel počítají vážené průměry atributů jeho sousedů, kde použitými váhami jsou váhy hran. $\boldsymbol{est_i^k}$ je vážený průměr atributu k sousedů uzlu i. Hodnota funkce je pak suma druhých mocnin rozdílu mezi hodnotou atributu k uzlu i a $\boldsymbol{est_i^k}$. Jinak řečeno se jedná o sumu druhých mocnin vzdáleností mezi vektory uzlů a vektory, jejichž hodnoty jsou vážené průměry atributů jejich sousedů v síti. Jelikož chceme, aby rozdíly mezi uzlem a jeho sousedy byly co nejmenší, tak chceme taky, aby tato vzdálenost byla co nejmenší. Platí tedy, že čím nižší je hodnota Daitchovy účelové funkce, tím lépe síť odpovídá vektorovým datům.

Daitchova účelová funkce má však jednu nevýhodu. Jelikož je založena na vzdálenosti, tak není schopna dobře porovnávat sítě, které byly získány podobnostmi, které vzdálenost nepoužívají např. kosinová podobnost. Aby tedy bylo možné ohodnotit sítě získané použitím libovolné podobnosti, tak Miloš Kudělka navrhl novou tzv. **podobnostní účelovou funkci**.

8.2 Podobnostní účelová funkce

Vstupem pro podobnostní účelovou funkci je vážená neorientovaná síť. Hodnotu funkce je vypočítána následovně:

1. Nechť je vrchol V_A sítě reprezentován vektorem $x_A = (x_{A1}, \ldots, x_{AD})$, kde D je dimenze vektorového datasetu. Předpokládejme, že vrchol V_A má k sousedů $V_1, \ldots, V_k$, přičemž má s jednotlivými sousedy váhy hran $w_1, \ldots, w_k$. Nechť jednotliví sousedé vrcholu V_A jsou reprezentováni vektory $x_1, \ldots, x_k$. Pak průměrný soused V_{Avg} vrcholu V_A je reprezentován vektorem $x_{Avg} = (x_{Avg1}, \ldots, x_{AvgD})$, jehož složky x_{Avgi} jsou vážené průměry hodnot odpovídajících složek sousedů vrcholu V_A:

$$x_{Avgi} = (w_i \cdot x_{1i} + \ldots + w_k \cdot x_{ki})/(w_i + \ldots + w_k), i = 1, \ldots, D$$

2. Nechť $s(x_A, x_{Avg})$ je funkce vektorové podobnosti (např. Gaussian kernel, kosinová podobnost) mezi vektorem x_A reprezentujícím vrchol V_A a vektorem x_{Avg} reprezentujícím jeho průměrného souseda V_{Avg}. Pak lokální kvalitu vrcholu V_A definujeme:

$$q(V_A) = s(x_A, x_{Avg})^2$$

3. Nechť má síť N vrcholů $V_1, \ldots, V_N$. Globální kvalitu konstrukce sítě z vektorových dat pak definujeme jako geometrický průměr lokálních kvalit všech vrcholů sítě:

$$G(V_1, \ldots V_N) = geom(q(V_1), \ldots, q(V_N))$$

První krok je tedy stejný s Daitchovou účelovou funkcí v tom, že počítáme průměrného souseda pro každý uzel. Ovšem v druhém kroku není použitá vzdálenost, ale libovolná vektorová podobnost, jež by měla odpovídat podobnosti použité k získání sítě. Tímto se řeší problém Daitchovy účelové funkce. Další výhodou podobnostní účelové funkce oproti Daitchově funkci je, že z její hodnoty ihned vidíme geometrický průměr podobnosti uzlů sítě a jejich sousedů. Podobnostní účelová funkce nabývá hodnot z intervalu $\langle 0, 1 \rangle$ a čím vyšší je hodnota funkce, tím lépe odpovídá síť vektorovým datům.

Problémem, který je nutno dořešit, jsou izolované uzly. Říkáme, že uzel je izolovaný, pokud nemá žádné sousedy. Jelikož izolovaný uzel nemá sousedy, není možné pro něj vypočítat průměrného souseda. Izolované uzly v sítí po konstrukci většinou nechceme, takže se řeší následovně:

1. Pro izolovaný uzel O definovat lokální kvalitu jako $q(O) = 0$.
2. Izolované uzly nezapočítávat do výpočtu globální kvality G.
3. Výslednou globální kvalitu G redukovat (vynásobit) koeficientem: $r = 1 - N_o/N$, kde N_o je počet izolovaných uzlů a N počet všech uzlů.

Pro výpočet hodnoty podobnostní účelové funkce jsme vytvořili aplikaci, která je popsána v příloze B.

Kapitola 9

Experiment s metodami konstrukce sítí z vektorových dat

Provedli jsme experiment, jehož cílem bylo zjistit, vlastnosti a kvalitu sítí při použití různých metod konstrukce sítě z vektorových dat. Použili jsme datasety **Iris**, **Ecoli**, **Seeds** a **Nuclear cortex** [20] (charakteristiky v tabulce 9.1), jež jsou dostupné na [21] a metody zmíněné v sekci 7.1. Nuclear cortex obsahuje chybějící hodnoty, které jsme doplnili průměrem z dostupných hodnot pro daný atribut. Pro každou kombinaci datasetu a metody jsme vytvořili 4 sítě, 2 za použití Gaussian Kernel, a 2 za použití kosinové podobnosti. Před použitím Gaussian kernel podobnosti byly datasety min-max normalizovány. Pro každou kombinaci datasetu, metody a podobnosti tedy máme 2 sítě, pro které jsme volili parametry metody konstrukce, tak aby průměrný stupeň pro první síť byl co nejblíže hodnotě 3 a pro druhou co nejblíže hodnotě 7. Pro Combination jsme nastavovali parametr k na hodnotu 1 pro průměrný stupeň 3 a na hodnotu 3 (4 pro Seeds/Cosine) pro průměrný stupeň 7.

Dataset	**Počet vektorů**	**Dimenze**	**Počet tříd**
Iris	150	4	3
Ecoli	336	7	8
Seeds	210	7	3
Nuclear cortex	1080	77	8

Tabulka 9.1: Charakteristiky použitých datasetů

9.1 Vlastnosti získaných sítí

Nyní porovnáme vlastnosti získaných sítí se stejnou použitou podobností a stejným průměrným stupněm. Vlastnosti, které nás zajímaly (v závorkách jsou jejich názvy v grafech), jsou kvalita sítě (Quality, sekce 8.2), maximální stupeň (MaxDeg), průměrný shlukovací koeficient (AvgClustCoef),

počet komponent (Components), modularita (Modularity, sekce 4.1), průměrný CRank (AvgCRank, sekce 4.3), počet komunit (Communities), minimální, průměrná a maximální velikost komunity (MinCommSize, AvgCommSize, MaxCommSize). Tabulky s hodnotami vlastností zkonstruovaných sítí jsou v příloze A.

Vytvořili jsme paprskové grafy, kde jedna osa reprezentuje jednu vlastnost. Rozsahy os jsou v popisku osy. Maximem osy je maximální nalezená hodnota atributu a minimem osy je minimální hodnota atributu, od níž byla odečtena čtvrtina rozdílu mezi maximem a minimem. Smyslem posunutí minima osy bylo, aby byly trochu lépe vidět poměry velikostí hodnot.

ε-radius jako jediná metoda tvoří izolované uzly. Ostatní metody pro naše nastavení vždy najdou aspoň jednoho souseda každému uzlu. Je nutné ještě objasnit, že do počtu komponent nepočítáme izolované uzly, tedy komponenty jsou aspoň velikosti 2. Komunity byly získány pomocí Louvain metody a pro ně byla vypočítána modularita a průměrný CRank. Stejně jako pro komponenty při hledání komunit nepočítáme s izolovanými uzly. Pro nalezení komunit a výpočet modularity jsme využili knihovny LouvainSharp [23] vytvořenou Markusem Mobiem.

9.1.1 Gaussian kernel podobnost a průměrný stupeň 3

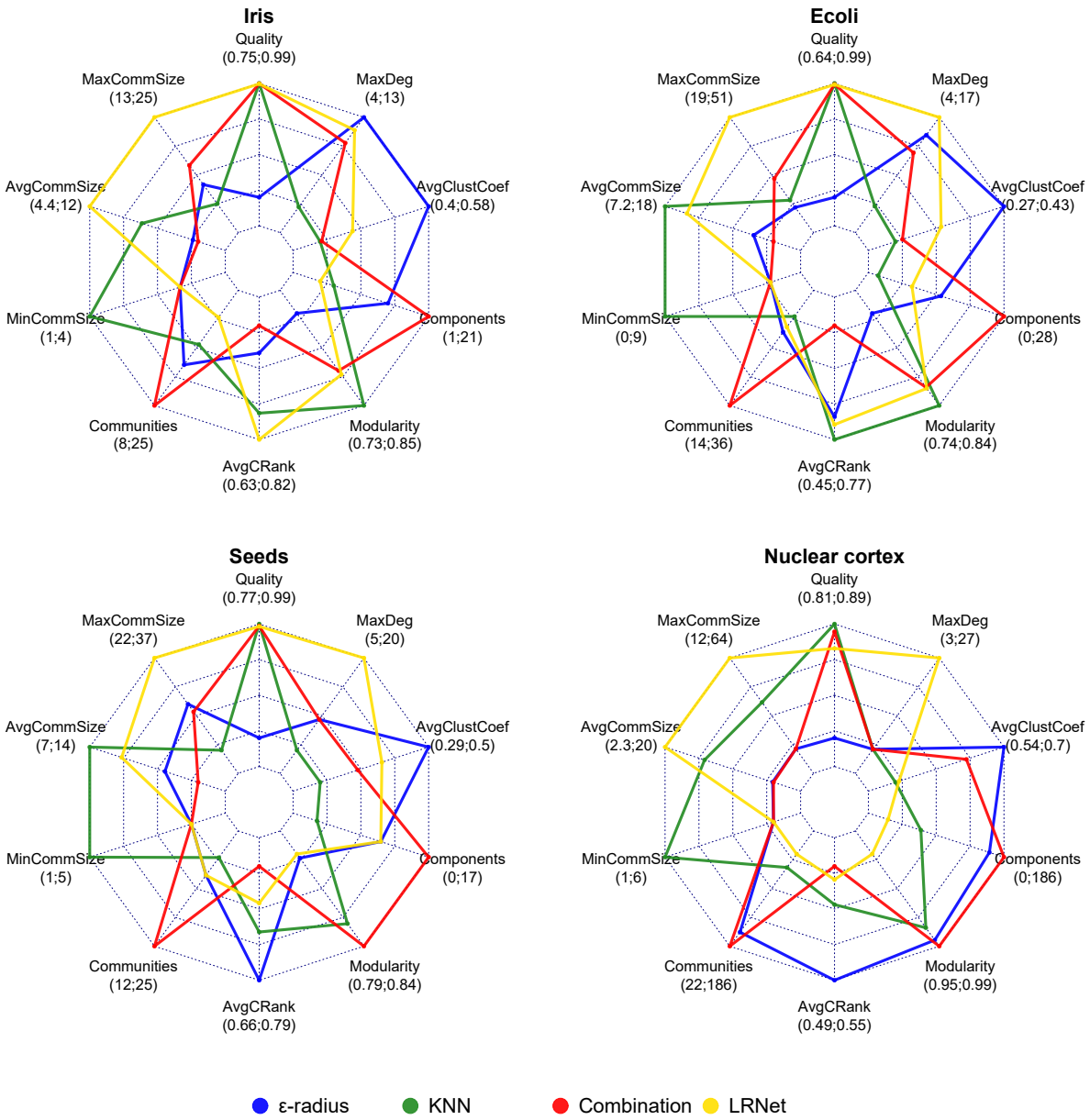
Vlastností získaných sítí pro Gaussian kernel podobnost a průměrný stupeň 3 jsou na obrázku 9.1.

Kvalita

Když se podíváme na kvalitu sítí, tak vidíme, že všechny metody až, na ε-radius, dosahují podobné kvality. ε-radius dosahuje výrazně nižší kvality než ostatní metody, což souvisí s izolovanými uzly, které ε-radius jako jediná metoda tvoří. Jelikož hodnota podobnostní účelové funkce je snižována podle počtu izolovaných uzlů, tak dostáváme nižší hodnoty kvality. Ostatní metody dosahují téměř shodných hodnot, LRNet jenom dosáhl trochu nižší kvality pro Nuclear cortex. Vidíme také, že pro menší datasety s menší dimenzí, dostáváme velice vysoké hodnoty kvality. Kvalita pro metody, jež netvoří izolované uzly se blíží 1, což je nejvyšší možná hodnota. Ovšem pro větší sítě, kde je atributů více, jež se více liší hodnotami, jako je Nuclear cortex, dostáváme hodnoty o něco nižší.

Maximální stupeň

Co se maximálního stupně týče, tak vidíme, že LRNet většinou má největší maximální stupeň. Toto značí, že LRNet má více rozmanitou distribuci stupňů a schopnost tvořit centra v síti. Dalším poznatkem je, že Combination a ε-radius mají podobný maximální stupeň. K tomu dochází pravděpodobně, jelikož Combination také využívá metodu ε-radius. Pro KNN je vzhledem k omezenému počtu přidávaných hran více problematické vytvářet uzly s tak velkými stupni, jako mají sítě ostatních metod, a proto má nejmenší maximální stupně. Nejzajímavější výsledky vidíme pro dataset Nuclear cortex. LRNet vytvořil síť o maximálním stupni 27, kdežto všechny ostatní metody měly maximální stupeň jenom hodnoty 8. Zajímavé je, že KNN a ε-radius dosáhlo stejné hodnoty tak



Obrázek 9.1: Vlastnosti sítí získaných pro Gaussian kernel podobnost a průměrný stupeň 3

i to, že LRNet má o tolik vyšší maximální stupeň. Možná to taky způsobilo to, že dosáhl o něco nižší kvality než KNN a Combination, jelikož spojil hranou někdy i méně podobné uzly.

Průměrný shlukovací koeficient

Průměrný shlukovací koeficient je vždy nejvyšší pro ε-radius a nejnižší pro KNN. LRNet a Combination jsou někde mezi nimi, většinou však mají nižší hodnoty blíže ke KNN. Ovšem vyšší průměrný shlukovací koeficient pro ε-radius může být svázán s izolovanými uzly, jelikož ti do něho nejsou započítáváni. Ostatní metody netvoří izolované uzly a vektory, jež jsou více odlišné od ostatních, pak mají málo sousedů a je tedy i méně pravděpodobné, že by byly součástí trojúhelníků. Tyto vektory značně snižují průměrný shlukovací koeficient. Po ε-radius má nejvyšší průměrný shlukovací koeficient většinou LRNet, což značí, že častěji tvoří trojúhelníky než zbylé metody. KNN naopak tvoří nejméně trojúhelníků.

Počet komponent

Nejvíce komponent dostáváme pro Combination. ε-radius jich má o něco méně a KNN s LRNetem mají většinou podobný nižší počet. Z toho můžeme usoudit, že Combination a ε-radius tvoří menší komponenty, kde jsou si vektory navzájem hodně podobné. KNN a LRNet pak některé tyto komponenty spojují dohromady.

Modularita

Pro modularitu vidíme zajímavé výsledky. KNN a Combination dosahují vysokých hodnot modularity pro všechny sítě. Pro Iris a Ecoli má nejvyšší modularitu KNN a pro Seeds a Nuclear Cortex má nejvyšší modularitu Combination. Toto značí, že ve struktuře sítí je jednoduché najít komunity, tedy mezi jednotlivými hustě propojenými komunitami je jen velice málo hran.

ε-radius má většinou nejnižší modularitu kromě výjimky pro Nuclear cortex. Toto je trochu překvapivé, protože by se dalo čekat, že ε-radius vytvoří skupiny uzlů, které jsou mezi sebou hustě propojeny. Avšak zřejmě dojde k tomu, že je složité najít jasnou hranici mezi jednotlivými skupinami uzlů. Mějme například dvě skupiny uzlů, které si nejsou navzájem podobné a třetí skupinu, které je podobna oběma z nich. První a druhá skupina poté budou spojeny do jedné větší skupiny kvůli třetí skupině, avšak zároveň mezi sebou nebudou mít žádné hrany. Tedy vnitřní hustota hran není veliká, avšak není možné je od sebe jasně oddělit.

LRNet má pro Iris a Ecoli vysokou modularitu, ale pro Seeds a Nuclear cortex nižší. Tyto rozdíly v modularitě lze vysvětlit tím, že pro různé datasety jsou vhodné sítě s různými počty hran (průměrnými stupni), aby metoda pro ně vytvořila sítě s dobrými komunitními strukturami.

Průměrný CRank

Nejnižší hodnoty průměrného CRanku dosahuje ve všech případech Combination. LRNet dosahuje vysokých hodnot pro Iris a Ecoli, ale nižších hodnot pro Seeds a Nuclear cortex. ε-radius má většinou vysoký průměrný CRank, avšak pro Iris dosahuje horších výsledků. KNN má spíše vyšší hodnoty.

Komunity

Počet komunit je úzce svázán s počtem komponent, jelikož komunit nemůže být méně, než je komponent a zároveň je pravděpodobné, že malá komponenta bude zároveň komunitou. Nejvíce komunit se vždy povedlo najít pro Combination, což odpovídá jeho největšímu počtu komponent. LRNet a KNN mají spíše menší počet komunit, což opět koresponduje s počtem komponent. Poměr počtu komunit ku počtu komponent je pro ε-radius v porovnání s ostatními metodami nižší. Bylo tedy problematické komponenty dále rozdělit na komunity, což se odrazilo na nižší modularitě. Ovšem výjimku vidíme pro Nuclear cortex, kde ε-radius má větší poměr komunit ku komponentám a taky má vysokou modularitu.

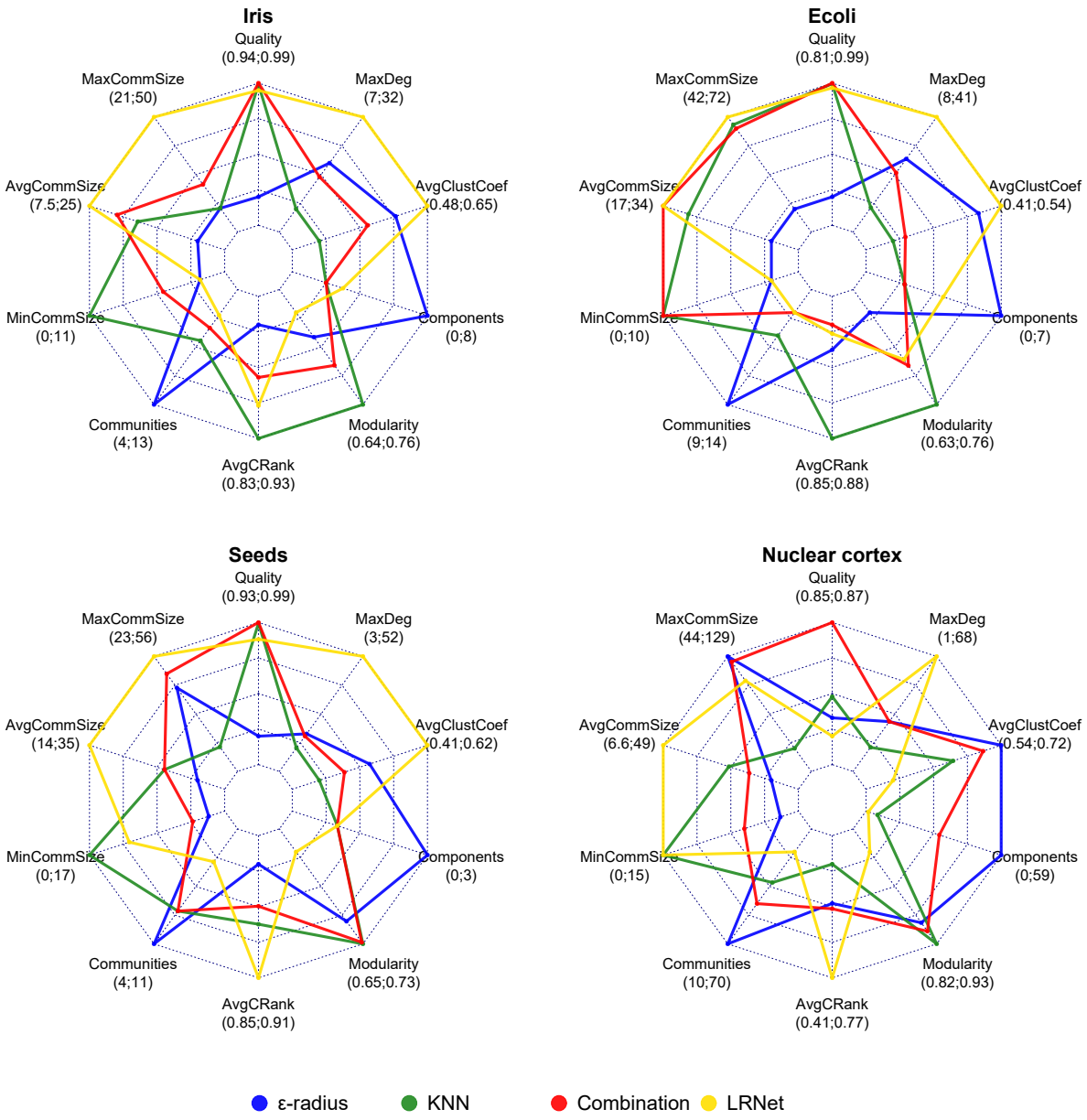
Když se podíváme na velikost komunit, tak můžeme vidět, že KNN je nejméně rozmanité. Má největší minimální a nejmenší maximální velikost komunit. Ovšem jeho průměrná velikost je vyšší než pro ε-radius a Combination, což je očekávané vzhledem k tomu, že jich má menší počet. Naopak Combination má nejmenší průměrnou velikost, jelikož má nejvíce komunit a ty jsou tedy většinou menší velikosti. Zároveň Combination má nejmenší minimální velikost, ale kromě sítě pro Nuclear cortex má vyšší maximum. Toto značí, že je schopno vytvořit i větší komunity. Pro ε-radius vidíme opět závislost průměrné velikosti na počtu komunit. Jelikož někdy má menší a někdy větší počet komunit, stejně kolísá i jeho průměrná velikost. Minimum a maximum je velice podobné Combination. Tyto hodnoty jsou však také nižší vzhledem k tomu, že ε-radius tvoří izolované uzly, kteří nejsou součástí komunit, tedy komunity se tvoří na menším počtu uzlů. LRNet tvoří spíše větší komunity, což vidíme z průměrné a především z maximální velikosti. Ovšem vidíme, že je schopen tvořit i menší komunity, jelikož minimum se shoduje s ε-radius a Combination.

9.1.2 Gaussian kernel podobnost a průměrný stupeň 7

Vlastností získaných sítí pro Gaussian kernel podobnost a průměrný stupeň 7 jsou na obrázku 9.2. Počet hran v těchto sítích je více než dvakrát větší, a kvůli toho se výrazně změní některé charakteristiky sítí.

Kvalita

Kvalita vypadá velice podobně jako pro sítě s průměrným stupněm 3. Rozdíl je v hodnotě kvality pro ε-radius, jelikož nyní dostáváme menší počet izolovaných uzlů vzhledem k vyššímu průměrnému stupni. Stále však tvoří málo kvalitní sítě v porovnání s ostatními metodami.



Obrázek 9.2: Vlastnosti sítí získaných pro Gaussian kernel podobnost a průměrný stupeň 7

Ke změně však dochází pro dataset Nuclear cortex, kde pro všechny metody dostáváme velmi malé rozdíly v kvalitě. Pro Combination došlo k nejmenšímu zhoršení, a je tedy nyní nejlepší metodou, co se kvality týče. KNN se zhoršilo o trochu více a je druhé nejlepší. LRNet s tímto zhoršením dohromady se zlepšením ε-radius je nyní jeho síť kvalitativně nejhorší. Je nutno podotknout, že rozdíly v kvalitě mezi metodami nejsou větší než 0,2, takže jsou si kvalitativně hodně podobné.

Maximální stupeň a průměrný shlukovací koeficient

Maximální stupně sítě s vyšším průměrným stupněm taky vzrostly, avšak metody je opět možné stejně seřadit. LRNet má nejvyšší maximální stupeň, KNN nejnižší a Combination a ε-radius jsou někde mezi nimi a jsou si navzájem velice podobné. Znovu vidíme schopnost LRNetu tvořit centra v síti.

Hodnoty průměrného shlukovacího koeficientu se také s vyšším průměrným stupněm zvýšily. Ale mezitím co předtím mělo nejvyšší shlukovací koeficient vždy ε-radius, tak nyní pro Iris, Ecoli a Seeds má nejvyšší průměrný shlukovací koeficient LRNet. Toto je asi ovlivněno snížením počtu izolovaných uzlů pro ε-radius. Zbytek je však velice podobný sítím o stupni 3.

Počet komponent

Počet komunit se s vyšším průměrným stupněm zmenšil, což je očekáváno. Nejvíce komponent nyní dostáváme pro ε-radius. Pro stupeň 3 měla nejvíce komponent Combination, parametr k měl nyní vyšší hodnotu, tak se také spíše blíží počtem komponent ke KNN. KNN má opět nejmenší počet komponent kromě sítě pro Nuclear cortex, kde nejméně má LRNet. LRNet pro Nuclear cortex vytvořil jenom jednu souvislou komponentu. Pro ostatní sítě však vytvořil stejně nebo více komponent než Combination a KNN.

Modularita

Modularita je nižší v porovnání s průměrným stupněm 3, ovšem metody se chovají podobně. KNN má nyní vždy nejvyšší modularitu a je následované Combination, které je o trochu horší. ε-radius už nezaostává o tolik za ostatními metodami, ovšem pro Ecoli je stále jasně nejhorší. LRNet má nyní o poznání nižší modularitu v porovnání s ostatními metodami. Pro Iris má nyní nejhorší modularitu a jenom pro Ecoli dosahuje modularity podobné jako ostatní metody.

Průměrný CRank

V porovnání s průměrným stupněm 3 vidíme, že hodnoty CRanku jsou ve většině případu vyšší pro stupeň 7. Máme mnohem méně komunit, což naznačuje, že menší komunity jsou CRankem hodnoceny hůře. Pro ε-radius dostáváme jiné výsledky než pro stupeň 3, průměrný CRank je pro něj většinou nízký. LRNet také dosáhl odlišných výsledků. Pro Seeds a Nuclear cortex má nyní

nejvyšší CRank, pro Iris ho má stále vysoký, ale pro Ecoli nyní získal nízký CRank. KNN má nejvyšší CRank pro Iris a Ecoli. Pro Seeds stále dosahuje dobrých výsledků, ale které jsou horší než výsledky pro LRNet. Pro Nuclear Cortex byly komunity pro KNN ohodnoceny nejhůře. Combination bylo pro stupeň 3 nejhorší. Nyní stále nedosahuje dobrých hodnot CRanku, ale až na výsledek pro Ecoli došlo ke zlepšení.

Komunity

Jelikož počet komunit je ovlivněn počtem komponent, tak vidíme, že nyní má nejvíce komunit ε-radius, stejně tak jako má nejvíce komponent. LRnet, přestože měl více nebo stejně komponent jako Combination a KNN, tak komunit má vždy nejméně. Toto značí, že jednotlivé komponenty není jednoduché dále rozdělit na komunity v porovnání s ostatními metodami, a toto asi způsobuje i nízkou modularitu. Combination a KNN jsou počtem komunit mezi ε-radius a LRNetem.

Pro KNN znovu vidíme vysokou minimální a malou maximální velikost komunit, tedy komunity se mezi sebou liší velikostí málo v porovnání s ostatními metodami. ε-radius má nejnižší minima, ovšem maxima se liší pro různé datasety. Pro Iris a Ecoli má ε-radius nejnižší maxima, ale pro Seeds a Nuclear cortex dosahuje vysokých maxim. LRNet se chová obráceně. Má vždy vysoká maxima, minima pro Iris a Ecoli jsou nízká, ale pro Seeds a Nuclear cortex jsou vysoká. Combination není moc konzistentní, co se minim a maxim týče. Minima jsou někde mezi ε-radius a KNN a maxima dosahují různě vysokých hodnot v porovnání s ostatními metodami. Můžeme vidět, že pro většinu metod je těžké vypozorovat pravidla pro minima a maxima velikostí komunit a spíše než metodou, jsou ovlivněny datasetem.

9.1.3 Kosinová podobnost a průměrný stupeň 3

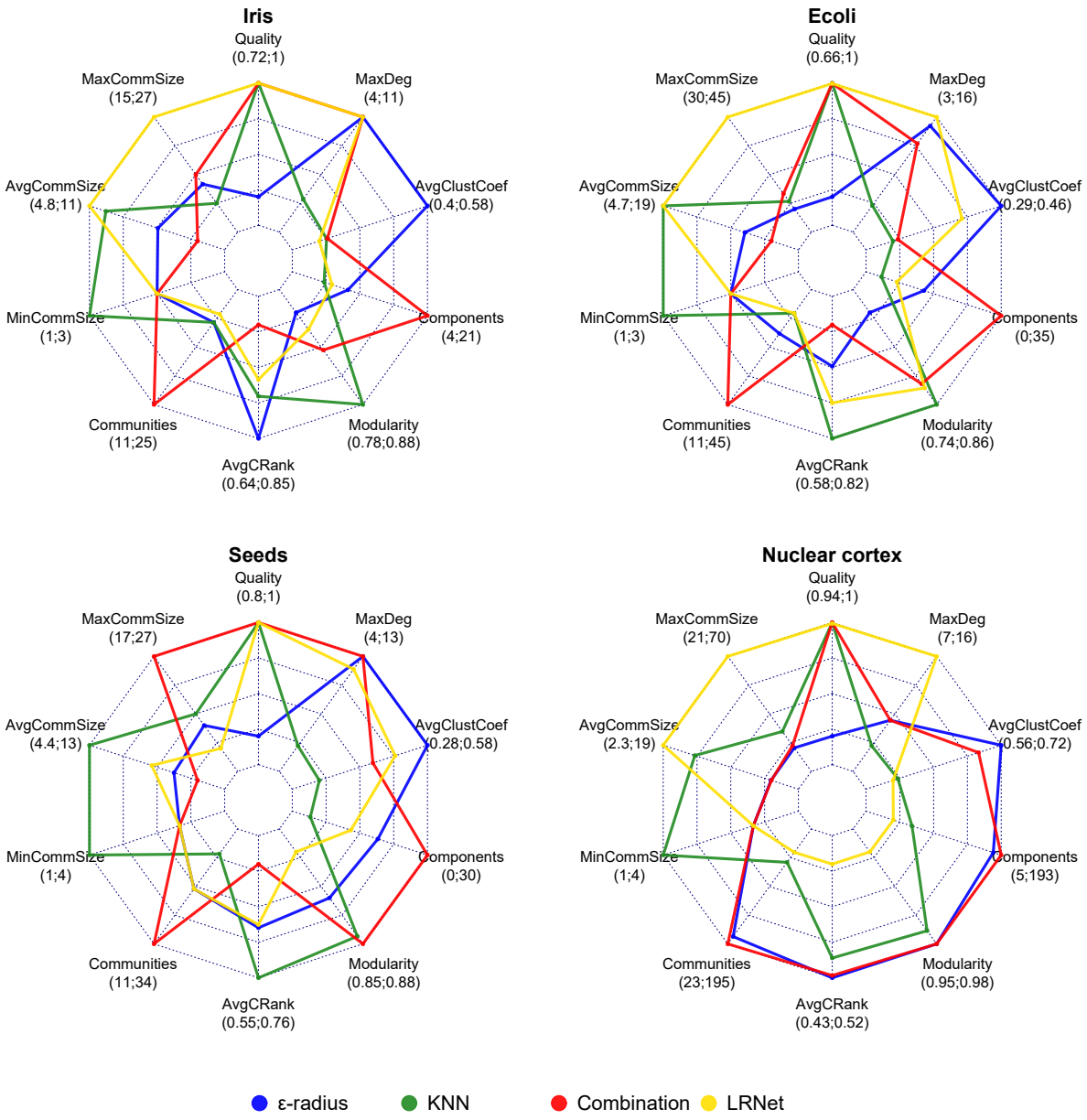
Vlastnosti získaných sítí pro kosinovou podobnost a průměrný stupeň 3 vidíme na obrázku 9.3. Porovnáme také tyto sítě se sítěmi o průměrném stupni 3, získaných pomocí Gaussian kernel podobnosti (sekce 9.1.1).

Kvalita

Všechny metody kromě ε-radius dosáhly kvality blízké 1 pro všechny datasety. Stejně jako pro Gaussian kernel, vidíme, že KNN, Combination a LRNet dosahují velice dobrých téměř identických výsledků. ε-radius oproti nim zaostává, což je opět způsobeno izolovanými uzly.

Maximální stupeň

Výsledky pro maximální stupeň jsou téměř shodné s Gaussian kernel podobností. Maximální stupeň pro LRNet je ve třech případech nejvyšší, Combination má podobný maximální stupeň jako ε-radius a KNN má nejnižší maximální stupeň. Jediný výrazný rozdíl vidíme pro dataset Seeds,



Obrázek 9.3: Vlastnosti sítí získaných pro kosinovou podobnost a průměrný stupeň 3

kde Combination a ε-radius mají vyšší maximální stupeň než LRNet. Toto je způsobeno tím, že maximální stupeň pro LRNet je podstatně nižší, než byl pro Gaussian kernel. I pro Nuclear cortex má LRNet nižší maximální stupeň než pro Gaussian kernel, avšak stále je vyšší než pro ostatní metody.

Průměrný shlukovací koeficient a počet komponent

Pro průměrný shlukovací koeficient nevidíme nic nového. Výsledky pro jednotlivé metody jsou téměř shodné jako předchozí výsledky pro Gaussian kernel. Jediný viditelný rozdíl je, že pro Iris LRNet nedosáhl vyššího koeficientu než Combination a KNN jako pro Gaussian kernel podobnost.

Počet komponent je také téměř shodný s výsledky pro Gaussian kernel. Jeden výrazný rozdíl je, že ε-radius a Combination vytvořily téměř dvojnásobně více komponent pro Seeds.

Modularita

Modularita pro Ecoli a Nuclear cortex je stejná jako modularita pro Gaussian kernel, jak hodnotami i pořadím metod. Iris a Seeds jsou také podobné výsledkům pro Gaussian kernel, ale můžeme najít pár rozdílů. Zaprvé modularita sítí pro tyto datasety většinou vzrostla (především pro Seeds). Ovšem pár metod se velikostí změny modularity liší od ostatních. LRNet pro Iris má stejnou hodnotu, jako měl pro Gaussian kernel, kdežto ostatní metody mají o trochu lepší. Pro Seeds se modularita pro ε-radius zlepšila výrazně více než pro ostatní metody.

Průměrný CRank

Hodnoty průměrného CRanku při použití kosinové podobnosti jsou téměř shodné s hodnotami pro sítě získané při použití Gaussian kernel podobnosti. Tedy na kvalitu komunit nemá použitá podobnost moc velký vliv. KNN má vysoké hodnoty CRanku, kdežto Combination je má většinou nízké. LRNet je průměrným CRankem mezi hodnotami pro KNN a Combination. ε-radius je docela nekonzistentní a někdy má vyšší a někdy nižší hodnoty pro různé datasety.

Dataset Nuclear cortex se liší od ostatních. LRNet zde má nejnižší průměrný CRank, kdežto Combination ho má s ε-radius nejvyšší. KNN má stále vysokou hodnotu CRanku.

Komunity

Chování metod, co se počtu komunit týče, je podobné jako pro Gaussian kernel. Combination má opět nejvyšší počet ve všech případech. Ostatní metody mají většinou o poznání méně a moc se mezi sebou neliší.

Z minimální a maximální velikosti komunit můžeme znovu vyčíst, že komunity pro KNN se velikostně neliší od průměrné velikosti tolik jako ostatní metody. ε-radius tvoří komunity malých velikostí, což můžeme vidět z hodnot pro minimální, průměrné a maximální velikosti komunit.

Combination také tvoří spíše menší komunity, avšak stále je občas schopno získat větší komunitu, což můžeme vidět na Seeds datasetu, kde pro Combination máme jednoznačně největší komunitu. LRNet je opakem ε-radius, jelikož tvoří sítě, ve kterých nalézáme spíše velké komunity, což vidíme z vysokých hodnot průměrné a maximální velikosti komunit. Můžeme ale vidět, že pro Seeds LRNet vytvořil menší komunity.

9.1.4 Kosinová podobnost a průměrný stupeň 7

Vlastností získaných sítí pro kosinovou podobnost a průměrný stupeň 7 vidíme na obrázku 9.4. Opět porovnáme také tyto sítě se sítěmi o průměrném stupni 7, získaných pomocí Gaussian kernel podobnosti (sekce 9.1.2).

Kvalita a maximální stupeň

Tak jako ve většině předchozích případů vidíme, že kvalita pro KNN, Combination a LRNet je téměř shodná a velice blízká 1. Toto platí i pro Nuclear cortex, který pro Gaussian kernel podobnost měl kvality LRNetu a KNN spíše nižší a blízké kvalitě ε-radius.

LRNet má tak jako ve většině předchozích případů nejvyšší maximální stupeň. Stejně tak KNN ho má opět vždy nejnižší. Combination a ε-radius mají znovu velice podobné maximální stupně.

Průměrný shlukovací koeficient a počet komponent

Průměrný shlukovací koeficient je velice podobný výsledkům pro Gaussian. Vidíme, že LRNet má vysoký průměrný shlukovací koeficient pro všechny datasety kromě Nuclear cortex, kde ho má naopak nejnižší. KNN má nízký průměrný shlukovací koeficient většinou dokonce nejnižší. Combination je přibližně uprostřed mezi maximem a minimem ostatních metod a ε-radius má většinou vysoký průměrný shlukovací koeficient.

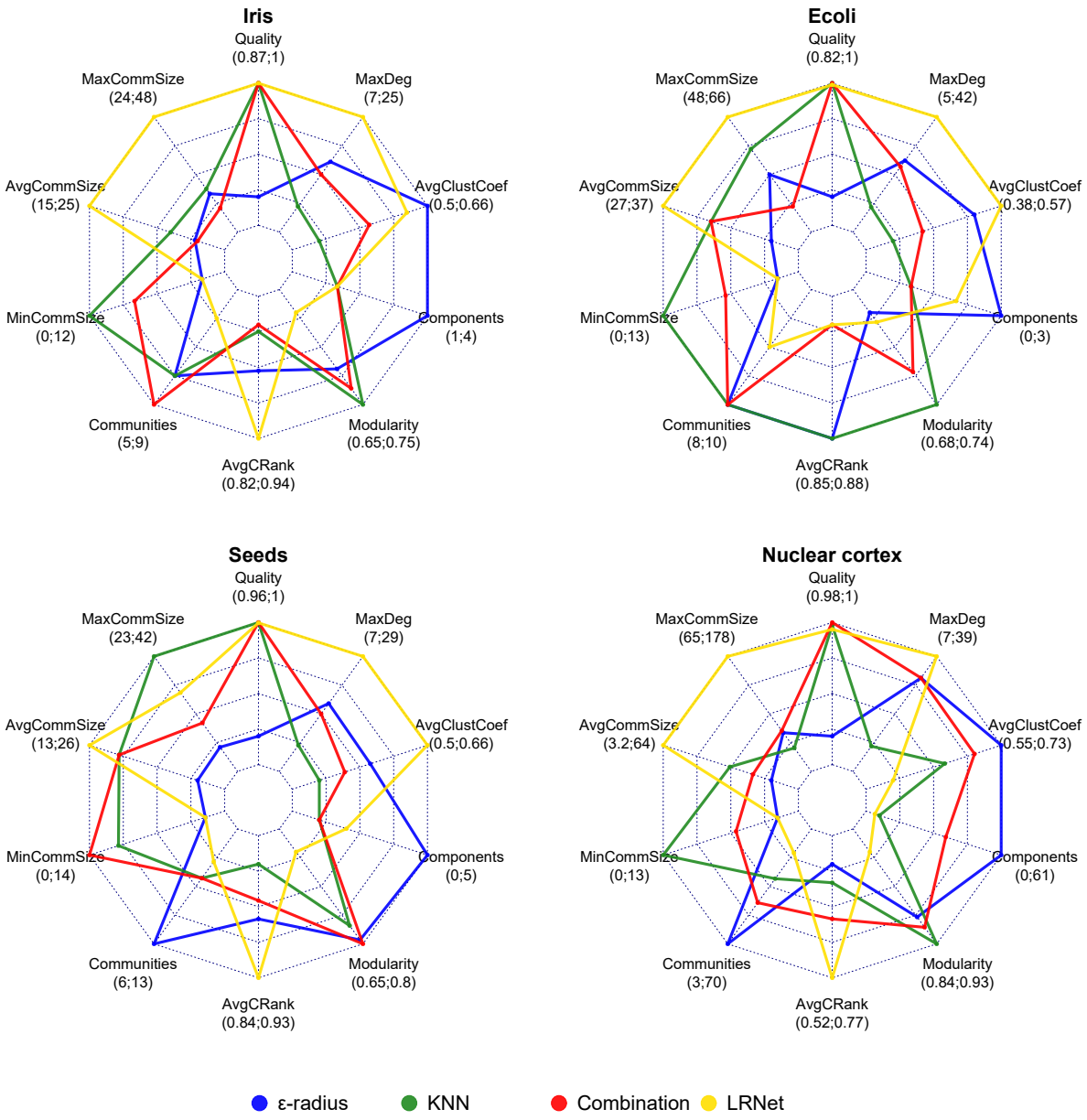
Pro kosinovou podobnost dostáváme méně nebo stejně komponent jako pro Gaussian kernel podobnost. Pořadí a rozdíly mezi metodami jsou také shodné.

Modularita

Modularita se moc neliší od modularity pro Gaussian kernel podobnost. LRNet má nízkou a většinou nejnižší modularitu v porovnání s ostatními metodami. KNN ji má naopak většinou nejvyšší. Combination má vyšší modularitu, která je velice blízká modularitě pro KNN. ε-radius nemá modularitu o moc nižší než KNN a Combination, ale pro Ecoli má nejnižší modularitu.

Průměrný CRank

Opět vidíme, že hodnoty průměrného CRanku pro kosinovou podobnost a pro Gaussian kernel podobnost jsou téměř shodné. LRNet má stejně jako pro Gaussian kernel opět vyšší hodnoty pro



Obrázek 9.4: Vlastnosti sítí získaných pro kosinovou podobnost a průměrný stupeň 7

Iris, Seeds a Nuclear cortex a nízké hodnoty pro Ecoli. KNN naopak dosahuje vysokých hodnot pro Ecoli a nízkých pro ostatní datasety. Combination dosahuje podobných hodnot jako pro Gaussian kernel podobnost, tedy CRank je spíše nižších hodnot. ε-radius není moc stabilní a dosahuje různě dobrých hodnot pro různé datasety.

Komunity

Pro počty komunit je složité vyčíst nějaké pravidla, jelikož pořadí metod ze zásadně liší pro různé datasety. Platí však, že pro LRNet se vždy povedlo najít nejméně komunit. ε-radius má často vyšší počet komunit, což se opět shoduje s jeho vyšším počtem komponent. Za poznámku stojí to, že pro dataset Ecoli sítě všech metod obsahovaly podobný počet komunit (9 LRNet, 10 ostatní).

Velikosti komunit jsou podobné s Gaussian kernel podobností. Pro ε-radius standardně dostáváme komunity malých velikostí. LRNet má klasicky největší komunity a Combination má komunity středních velikostí v porovnání s ostatními metodami. Pro KNN však dostáváme odlišné výsledky. KNN má stále vysokou minimální velikost, ale už nemá tak nízkou maximální velikost. Pro Seeds KNN předčilo maximální velikostí komunit i LRNet.

9.1.5 Shrnutí

Mohli jsme si všimnout, že pro různé nastavení měly metody často podobné charakteristiky. Například pro průměrný stupeň 3 jsme dostávali velice podobné výsledky, ať už byly sítě získány pomocí Gaussian kernel nebo kosinové podobnosti. Toto není zas tak překvapivé, jelikož hodně charakteristik sítě je úzce spjato s počtem hran, který pro stejný průměrný stupeň bude téměř identický, ať už byla použita libovolná podobnost. Navíc se očekává, že sítě získané stejnou metodou konstrukce sítí z vektorových dat budou mít podobné charakteristiky a naším hlavním cílem bylo tyto charakteristiky nalézt.

Kvalita sítí pro KNN, Combination a LRNet byla téměř vždy shodná a podstatně vyšší než kvalita ε-radius. Toto je způsobeno tím, že ε-radius vytváří velký počet izolovaných uzlů.

Maximální stupeň měl skoro vždy nejvyšší LRNet, z toho můžeme usoudit, že LRNet tvoří nejvíce rozmanitou distribuci stupňů. Zároveň to ukazuje na schopnost tvořit v síti centra, která bývají při následující analýze dat důležitá a lze je využít například při shlukování [22]. KNN má naopak nejnižší maximální stupeň. Combination a ε-radius měly velice podobné maximální stupně, jež hodnotou ležely mezi KNN a LRNetem.

Průměrný shlukovací koeficient mělo nejvyšší často ε-radius. Ovšem toto může být ovlivněno izolovanými vrcholy, jelikož ty nejsou započítávány. LRNet byl další metodou s vysokým shlukovacím koeficientem, pro Nuclear cortex měl však většinou průměrný shlukovací koeficient nejnižší. KNN mělo většinou nízký shlukovací koeficient a Combination dosahovalo středně vysokých hodnot.

KNN a LRNet tvoří menší počet komponent. ε-radius tvoří velký počet komponent a pro průměrný stupeň 7 jich má nejvíce. Combination má nejvíce komponent pro průměrný stupeň 3 a pro stupeň 7 má méně komponent než KNN.

Modularita byla hodně proměnlivá. KNN mělo vždy vysokou modularitu. Combination mělo spíše vyšší modularitu stejnou jako nebo o trochu nižší než KNN. ε-radius bylo nekonzistentní. Pro Nuclear cortex mělo vyšší hodnoty a pro Ecoli nižší. Ale pro Seeds a Iris dosahovala modularita pro různá nastavení sítí různě vysokých hodnot. LRNet měl zpravidla nízkou modularitu, ale v několika případech dosáhl modularity stejně vysoké jako Combination. Zde je nutné ale poznamenat, že vysoká modularita nemusí vždy znamenat lepší síť. Vysoká modularita může také ukazovat na to, že síť zachycuje jenom některé charakteristiky dat, a proto je v ní jednodušší oddělit komunity od sebe.

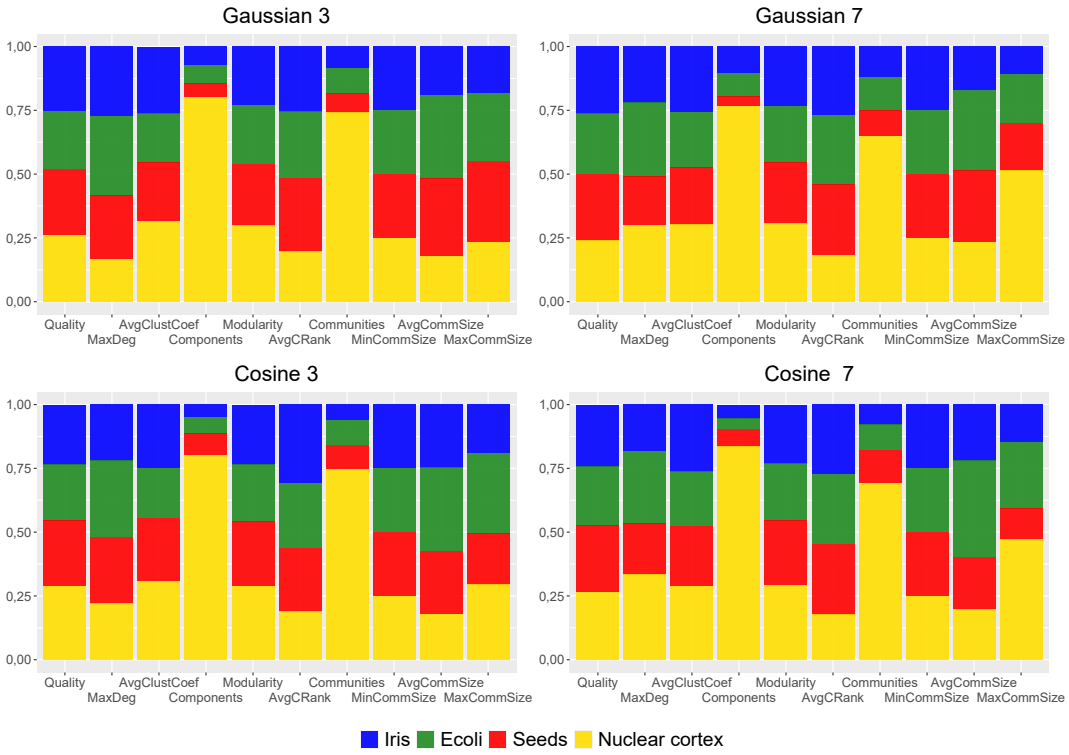
Průměrný CRank byl hodně proměnlivý stejně jako modularita. LRNet měl vysoké hodnoty pro Iris, Seeds a Nuclear cortex pro průměrný stupeň 7 obou podobností. Combination dosahovalo skoro vždy nízkých hodnot. KNN dosahuje většinou hodně nízkých anebo hodně vysokých hodnot. Pro Ecoli má KNN vždy vysoké hodnoty, ale pro ostatní datasety nedostáváme konzistentně dobré nebo špatné hodnoty. ε-radius se zdá být naprosto náhodné, nelze najít žádné pravidlo pro jeho hodnoty CRanku.

Počet komunit byl částečně svázán s počtem komponent. Proto ε-radius a Combination měly vyšší počet komunit, pokud měly jejich sítě více komponent. LRNet měl málo komunit v porovnání s ostatními metodami a KNN také mělo spíše méně komunit.

Velikosti komunit byly hodně konzistentní mezi různými sítěmi. LRNet většinou tvořilo jednoznačně největší komunity, a zároveň také získá i několik menších komunit. Tato vlastnost odpovídá přirozenému chování reálných sítí, ve kterých se komunity výrazně liší velikostí. ε-radius mělo spíše menší komunity, což je pravděpodobně způsobeno výskytem izolovaných uzlů v jeho sítích. KNN je metoda, jež má vysoká minima a nízká maxima velikostí komunit. Tvoří však větší komunity než ε-radius a Combination. Combination má také spíše menší komunity, ale existuje i množství případů, kdy průměrná velikost komunit byla srovnatelná s KNN. Toto dává smysl vzhledem k tomu, že Combination využívá ε-radius i KNN.

9.2 Stabilita chování metod

Další věcí, co nás zajímá, je, jestli metody konzistentně tvoří sítě s podobnými vlastnostmi. Jelikož námi použité datasety se moc neliší počtem vektorů, tak by měly jejich sítě získané stejnou metodou nabývat podobných vlastností. Vytvořili jsme 4 sloupcové grafy pro každou metodu, jeden graf pro každé nastavení sítě (podobnost, průměrný stupeň). Každý sloupec reprezentuje jednu vlastnost a je obarven 4 barvami, kde každý dataset má přiřazen 1 barvu. Každý sloupec zobrazuje relativní hodnoty vlastnosti pro sítě různých datasetů ku součtu těchto hodnot. Tedy ideálně aby byla vlastnost konzistentní, tak by části sloupce, které jsou obarveny různými barvami, měly být stejně velké.

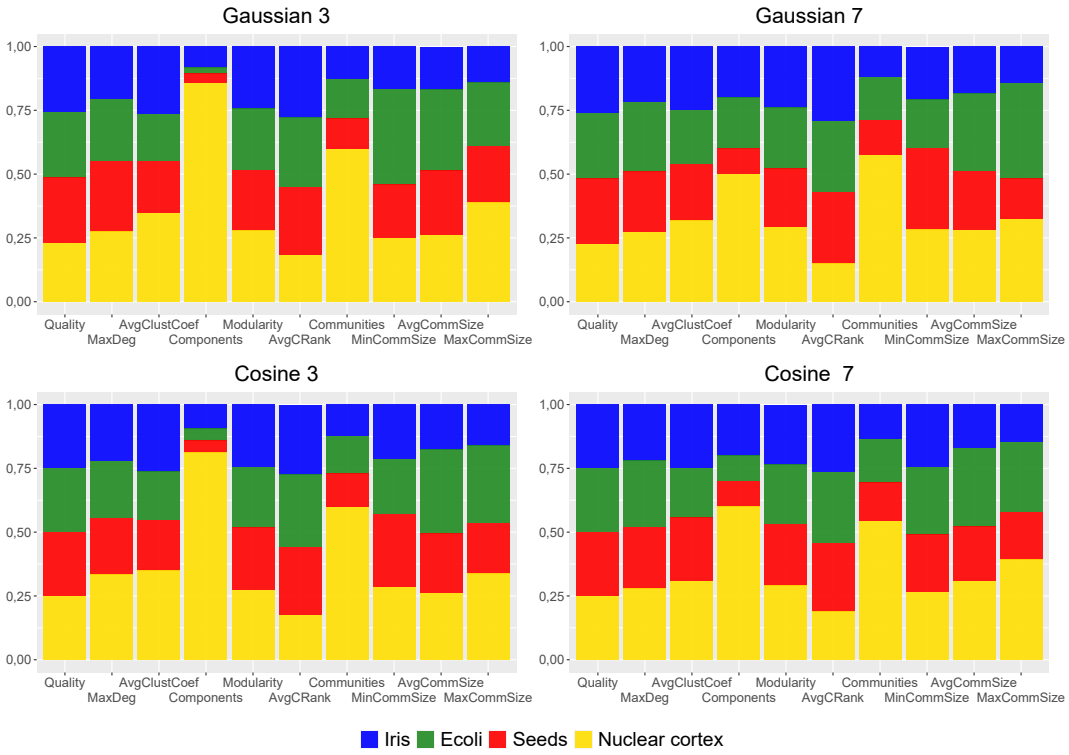


Obrázek 9.5: Vlastnosti sítí získaných pro ε-radius

Nuclear cortex je větší než ostatní datasety, takže vlastnosti, které jsou ovlivněny počtem uzlů, jako jsou počty komponent a komunit, se mohou pro tento dataset lišit. Vytvořili jsme také tabulku 9.2, ve které jsou směrodatné odchylky pro relativní hodnoty (stejné jako ve sloupcových grafech) podle metod. Čím nižší hodnota směrodatné odchylky, tím je metoda pro vlastnost stabilnější. V posledním sloupci tabulky je průměr těchto směrodatných odchylek pro každou metodu, což vyjadřuje její celkovou stabilitu. Jak dále uvidíme nejstabilnější jsou metody LRNet a KNN.

9.2.1 ε-radius

Na obrázku 9.5 vidíme, že ε-radius tvoří sítě, pro které je kvalita velice stabilní. Maximální stupeň je méně konzistentní. Vidíme, že sítě pro Ecoli mají o poznání vyšší stupeň než sítě ostatních datasetů pro Gaussian kernel podobnost. Také různé podobnosti způsobují změny pro maximální stupeň, jelikož pro kosinovou podobnost má Nuclear cortex vyšší maximální stupně než pro Gaussian kernel. Průměrný shlukovací koeficient je trochu stabilnější než maximální stupeň a vidíme, že Ecoli a Seeds mají nižší hodnotu než Iris a Nuclear cortex. Počet komponent a komunit je pro ε-radius hodně závislý na velikosti datasetu. Vidíme, že sítě pro Nuclear cortex mají jednoznačně nejvíce komponent a komunit. I ostatní datasety, co se neliší velikostí tolik, mají výrazně rozdílné hodnoty. Modularita je hodně stabilní, ale Nuclear cortex dosahuje o trochu vyšších hodnot. Průměrný CRank je také hodně stabilní, ale tentokrát Nuclear cortex dosahuje nižších hodnot. Minimální velikost komunit



Obrázek 9.6: Vlastnosti sítí získaných pro KNN

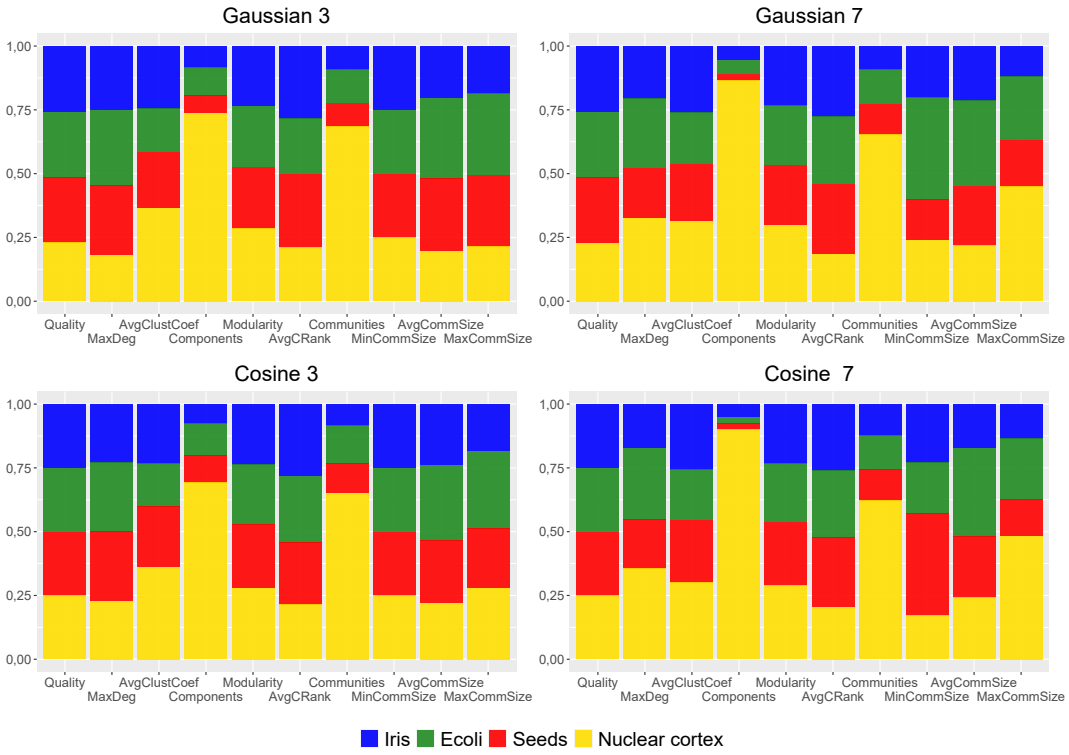
je velmi stabilní, všechny sítě ji měly rovnou dvěma. Průměrná a maximální velikost komunit jsou málo stabilní.

9.2.2 KNN

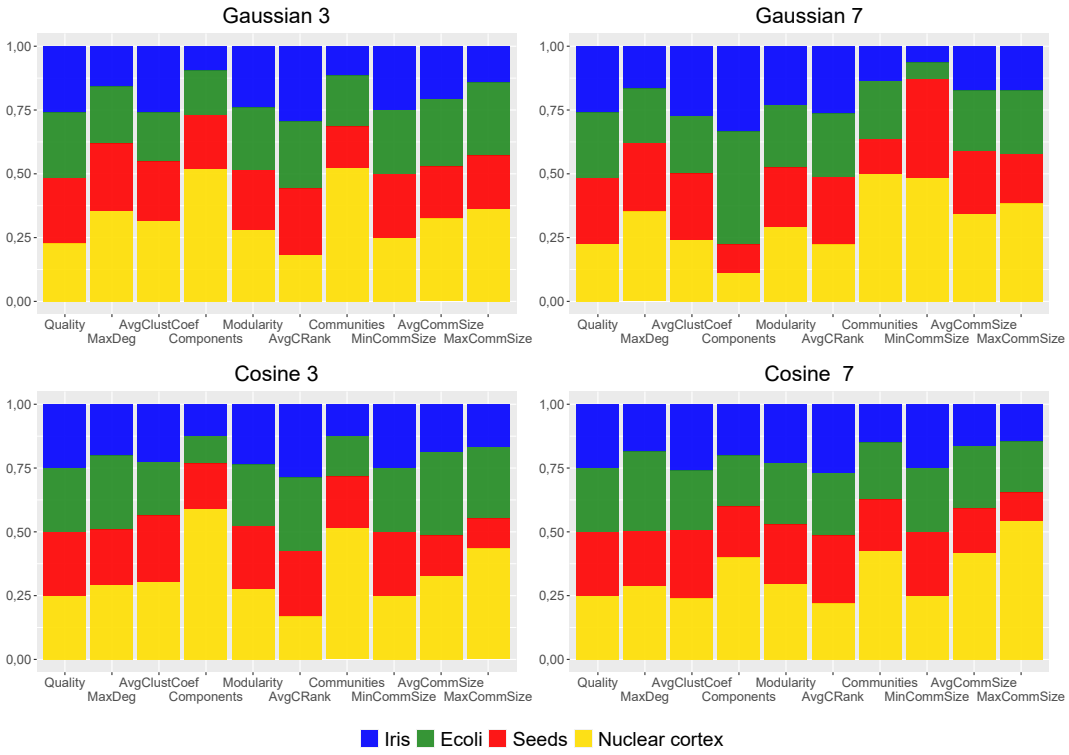
Na obrázku 9.6 si můžeme všimnou velice stabilního chování KNN s ohledem na kvalitu sítě a modularitu. Maximální stupeň je stabilnější než pro ε-radius, ale stále je možno si všimnou drobných rozdílů. Průměrný shlukovací koeficient je docela stabilní, avšak méně než pro ε-radius. Počet komponent a komunit je opět hodně závislý na velikosti datasetu. I přestože je počet komunit málo konzistentní, tak pro KNN je výrazně konzistentnější než pro ε-radius. Výsledky pro průměrný CRank jsou téměř shodné s ε-radius, tedy jsou hodně stabilní. Velikosti komunit nejsou moc stabilní. Maximální velikost je stabilnější než pro ε-radius, ale průměrná a minimální velikost jsou o poznání méně stabilní.

9.2.3 Combination

Kvalita a modularita jsou velmi stabilní, což je zobrazeno na obrázku 9.7. Maximální stupeň je docela stabilní, ale vidíme drobné rozdíly. Je podobně stabilní jako maximální stupeň pro ε-radius. Průměrný shlukovací koeficient je celkem stabilní podobně jako pro ostatní metody. Počet kompo-



Obrázek 9.7: Vlastnosti sítí získaných pro Combination



Obrázek 9.8: Vlastnosti sítí získaných pro LRNet

nent a komunit je opět velice nestabilní. Pro kosinovou podobnost jsou stabilnější než pro Gaussian kernel podobnost, ale rozdíly mezi datasety jsou stále veliké. Průměrný CRank je hodně stabilní, tak jako pro ostatní metody. Velikosti komunit jsou celkem stabilní pro Gaussian kernel podobnost, ale pro kosinovou podobnost naopak nejsou stabilní skoro vůbec.

9.2.4 LRNet

I pro LRNet na obrázku 9.8 vidíme, že kvalita a modularita jsou, jako pro všechny metody, velmi stabilní. Maximální stupeň a průměrný shlukovací koeficient jsou jako pro ostatní metody méně stabilní, ale rozdíly mezi datasety nejsou veliké. Počet komponent a komunit, jsou o poznání konzistentnější než u ostatních metod. Hlavním důvodem je, že počet vektorů v datasetu nemá na tyto vlastnosti pro LRNet tak velký vliv. Dokonce pro Gaussian kernel podobnost a průměrný stupeň 3 mají sítě pro Seeds a Nuclear cortex jenom jednu komponentu. Rozdíly mezi datasety jsou ale pořád výrazné. Průměrný CRank je hodně stabilní. Velikosti komunit jsou pro LRNet hodně nestabilní, ať už se jedná o minimální, průměrnou nebo maximální velikost.

9.2.5 Shrnutí

Mohli jsme vidět, že hodně vlastností je stejně stabilních pro všechny metody. Kvalita, modularita a průměrný CRank byly hodně stabilní, kdežto počet komponent a komunit naopak nejsou stabilní skoro vůbec. Maximální stupeň a průměrný shlukovací koeficient byly méně stabilní vlastnosti a metody se v nich nejvíce lišily. Průměrná a maximální velikost komunit jsou nekonzistentní. Minimální velikost komunit je stabilní jenom pro ε-radius a KNN.

Když se podíváme na relativní hodnoty směrodatných odchylek v tabulce 9.2, tak vidíme, že hodnoty kvality a modularity se moc neliší. Pro maximální stupeň vidíme, že nejkonzistentnější je KNN, což vzhledem k pevně určenému přidávání počtu hran dává smysl. LRNet má směrodatnou odchylku největší, což má nejspíš na svědomí dříve zjištěná skutečnost, že tvoří centra. Jelikož datasety mají různý počet vektorů a strukturu dat, tak lze očekávat, že centra budou mít různé stupně. Průměrný shlukovací koeficient má nejlepší LRNet, i když je jenom nepatrně lepší než ε-radius. Pro počet komponent a komunit vidíme, že LRNet je výrazně lepší než ostatní metody, čehož jsme si všimli už při analýze sloupcových grafů. Hodnoty průměrného CRanku má nejkonzistentnější Combination, ale ostatní metody nezaostávají o moc. Pro minimální velikost komunit je nejstabilnější ε-radius, hodnoty dokonce byly vždy stejné. Pro průměrnou velikost komunit je nejstabilnější Combination a pro maximální velikost komunit je nejstabilnější KNN. LRNet je nejméně stabilní pro velikosti komunit. Nejlepší průměr směrodatných odchylek má LRNet a můžeme tedy říct, že LRNet je nejstabilnější metodou. Viděli jsme, že byl výrazně stabilnější pro počet komponent a komunit, než ostatní metody. Sice byl málo stabilní pro velikosti komunit, ale rozdíl oproti ostatním metodám nebyl veliký.

Metoda	Quality	MaxDeg	AvgCC	Comp.	Modul.	AvgCR.	Comm.	Community size			Průměr
								Min	Avg	Max	
ε-radius	0,018	0,051	0,039	0,329	0,029	0,040	0,274	**0,000**	0,065	0,114	0,096
KNN	**0,009**	**0,033**	0,056	0,280	**0,021**	0,046	0,197	0,053	0,055	**0,093**	0,084
Combination	**0,009**	0,052	0,059	0,332	0,023	**0,033**	0,241	0,066	**0,051**	0,103	0,097
LRNet	0,010	0,062	**0,032**	**0,158**	0,022	0,035	**0,149**	0,097	0,077	0,124	**0,077**

Tabulka 9.2: Relativní směrodatné odchylky vlastností sítí

Kapitola 10

Závěr

V rámci první části této práce jsme provedli experiment s algoritmy detekce komunit Ego-zones a DEMON. Nalezené komunity a ground-truth komunity jsme porovnali z pohledu distribuce jejich velikostí a několika měr pro ohodnocení jejich kvality. Ukázalo se, že každý algoritmus má své výhody a nevýhody. Nejvíce záleží na tom, jaké komunity chceme najít. Pokud bychom chtěli menší komunity o vyšším počtu, je vhodnější použít Ego-zones, a pokud chceme nižší počet komunit velké velikosti, tak použijeme DEMON. Kvalita komunit nalezených různými metodami také hodně záleží na vlastnostech sítě (např. průměrném stupni), ve které komunity detekujeme. Obecně platí pravidlo, že je lepší použít více algoritmů a zjistit, který více vyhovuje našemu účelu.

V druhé části jsme mezi sebou porovnali metody konstrukce sítí z vektorových dat ε-radius, KNN, Combination a LRNet. Vypočítali jsme kvality zkonstruovaných sítí těmito metodami pomocí podobnostní účelové funkce. Následně jsme zjistili, jaké charakteristiky sítě získané určitou metodou mají. Ukazuje se, že stejně jako algoritmy detekce komunit každá metoda tvoří sítě s odlišnými vlastnostmi a je nutno si z nich vybrat podle toho, jaké charakteristiky sítě si přejeme. Například chceme-li v síti uzly s vyššími stupni tzv. centra použijeme LRNet, chceme-li vysoký počet komunit použijeme Combination atd. Porovnali jsme metody také vzhledem ke stabilitě vlastností konstruovaných sítí a zjistili jsme, že LRNet s KNN jsou stabilnější než ε-radius a Combination.

Na oba tyto experimenty je možné dále navázat. V první části jsme použili menší počet algoritmů, ale existuje jich více. Ovšem narazili jsme na problém, že hodně těchto algoritmů vrací vícevrstvé hierarchie komunit. Problémem potom je, jakou vrstvu je vhodné porovnávat s výsledky algoritmů netvořící tuto hierarchii. Za tím účelem by bylo nutné navrhnout trochu odlišný experiment. Také je tu možnost využít dalších měr pro porovnání kvality komunit. V druhém experimentu je možné konstruovat další sítě při použití jiných nastavení, nicméně pro naše experimenty jsou použitá data dostatečná. Kromě Nuclear cortex byly naše datasety o menším počtu vektorů, jelikož pro takové datasety je snadnější interpretovat výsledky. Navíc se jedná o známé referenční datasety, na kterých se často metody konstrukce sítí testují a porovnávají. Do budoucna by nicméně mohlo být zajímavé otestovat metody konstrukce sítí z vektorových dat také na velkých datasetech.

V průběhu práce vzniklo velké množství souborů se sítěmi, komunitami atd. Také jsme vytvořili aplikaci pro analýzu vlastností sítí a aplikaci pro konstrukci sítí z vektorových dat. Všechny tyto soubory a aplikace jsou v popsány příloze B.

Literatura

1. BARABÁSI, Albert-László; PÓSFAI, Márton. *Network science* [online]. Cambridge: Cambridge University Press, 2016 [cit. 2020-12-04]. ISBN 978-110-7076-266. Dostupné z: `http://networksciencebook.com/`.
2. BLONDEL, Vincent D; GUILLAUME, Jean-Loup; LAMBIOTTE, Renaud; LEFEBVRE, Etienne. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* [online]. 2008-10-01, roč. 2008, č. 10 [cit. 2020-12-07]. ISSN 1742-5468. Dostupné z DOI: `10.1088/1742-5468/2008/10/P10008`.
3. YANG, Jaewon; LESKOVEC, Jure. Defining and Evaluating Network Communities Based on Ground-Truth. *2012 IEEE 12th International Conference on Data Mining* [online]. 2012, roč. 2012, s. 745–754 [cit. 2020-12-07]. ISBN 978-1-4673-4649-8. Dostupné z DOI: `10.1109/ICDM.2012.138`.
4. KUDELKA, Milos; OCHODKOVA, Eliska; ZEHNALOVA, Sarka; PLESNIK, Jakub. Ego-zones: non-symmetric dependencies reveal network groups with large and dense overlaps. *Applied Network Science* [online]. 2019, roč. 4, č. 1 [cit. 2020-12-14]. ISSN 2364-8228. Dostupné z DOI: `10.1007/s41109-019-0192-6`.
5. COSCIA, Michele; ROSSETTI, Giulio; GIANNOTTI, Fosca; PEDRESCHI, Dino. DEMON. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12* [online]. 2012, s. 615– [cit. 2020-12-08]. ISBN 9781450314626. Dostupné z DOI: `10.1145/2339530.2339630`.
6. COSCIA, Michele; ROSSETTI, Giulio; GIANNOTTI, Fosca; PEDRESCHI, Dino. Uncovering Hierarchical and Overlapping Communities with a Local-First Approach. *ACM Transactions on Knowledge Discovery from Data* [online]. 2014-10-28, roč. 9, č. 1, s. 1–27 [cit. 2020-12-08]. ISSN 1556-4681. Dostupné z DOI: `10.1145/2629511`.
7. RAGHAVAN, Usha Nandini; ALBERT, Réka; KUMARA, Soundar. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E* [online]. 2007, roč. 76, č. 3 [cit. 2020-12-08]. ISSN 1539-3755. Dostupné z DOI: `10.1103/PhysRevE.76.036106`.

8. NEWMAN, M. E. J. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* [online]. 2006-06-06, roč. 103, č. 23, s. 8577–8582 [cit. 2020-12-10]. ISSN 0027-8424. Dostupné z DOI: 10.1073/pnas.0601602103.

9. SCHAEFFER, Satu Elisa. Graph clustering. *Computer Science Review* [online]. 2007, roč. 1, č. 1, s. 27–64 [cit. 2020-12-10]. ISSN 15740137. Dostupné z DOI: 10.1016/j.cosrev.2007.05.001.

10. ZITNIK, Marinka; SOSIČ, Rok; LESKOVEC, Jure. Prioritizing network communities. *Nature Communications* [online]. 2018, roč. 9, č. 1 [cit. 2020-12-10]. ISSN 2041-1723. Dostupné z DOI: 10.1038/s41467-018-04948-5.

11. *Amazon* [online] [cit. 2020-12-15]. Dostupné z: https://www.amazon.com/.

12. *Dblp: Computer science bibliography* [online]. Leibnitz: Schloss Dagstuhl [cit. 2020-12-15]. Dostupné z: https://dblp.org/.

13. *Youtube* [online] [cit. 2020-12-15]. Dostupné z: https://www.youtube.com/.

14. LESKOVEC, Jure. *Stanford Large Network Dataset Collection* [online]. Stanford: Stanford university [cit. 2020-12-15]. Dostupné z: http://snap.stanford.edu/data/index.html.

15. *Prioritizing network communities* [online]. Stanford: Stanford university [cit. 2020-12-15]. Dostupné z: http://snap.stanford.edu/crank/.

16. KUDELKA, Milos. *Ego-zones: non-symmetric dependencies reveal network groups with large and dense overlaps* [online]. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava [cit. 2020-12-15]. Dostupné z: https://homel.vsb.cz/~kud007/ego_zones_files/.

17. OCHODKOVA, Eliska; ZEHNALOVA, Sarka; KUDELKA, Milos. Graph Construction Based on Local Representativeness. *Computing and Combinatorics*. 2017, s. 654–665. ISBN 978-3-319-62388-7. Dostupné z DOI: 10.1007/978-3-319-62389-4_54.

18. *Graph Construction Based on Local Representativeness*. Dostupné také z: https://homel.vsb.cz/~kud007/lrnet_files/.

19. DAITCH, Samuel I.; KELNER, Jonathan A.; SPIELMAN, Daniel A. Fitting a graph to vector data. *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*. 2009, s. 1–2. ISBN 9781605585161. Dostupné z DOI: 10.1145/1553374.1553400.

20. HIGUERA, Clara; GARDINER, Katheleen J.; CIOS, Krzysztof J.; HERAULT, Yann. Self-Organizing Feature Maps Identify Proteins Critical to Learning in a Mouse Model of Down Syndrome. *PLOS ONE*. 2015-6-25, roč. 10, č. 6. ISSN 1932-6203. Dostupné z DOI: 10.1371/journal.pone.0129126.

21. *UCI: Machine Learning Repository* [online]. Irvine: University of California, School of Information a Computer Sciences, 2017 [cit. 2021-04-10]. Dostupné z: https://archive.ics.uci.edu/ml/.

22. TOMASEV, Nenad; RADOVANOVIC, Milos; MLADENIC, Dunja; IVANOVIC, Mirjana. The Role of Hubness in Clustering High-Dimensional Data. *IEEE Transactions on Knowledge and Data Engineering* [online]. 2014, roč. 26, č. 3, s. 739–751 [cit. 2021-04-22]. ISSN 1041-4347. Dostupné z DOI: 10.1109/TKDE.2013.25.

23. *LouvainSharp - Fast Louvain Method of Community Detection in C#* [online]. Markus Mobius [cit. 2021-04-29]. Dostupné z: https://www.markusmobius.org/software/louvainsharp-fast-louvain-method-community-detection-c.

Příloha A

Tabulky hodnot vlastností sítí pro experiment s metodami konstrukce sítí z vektorových dat

V této příloze jsou tabulky s vlastnostmi sítí z experimentu s metodami konstrukce sítí z vektorových dat. Obsahují i hodnoty průměrného stupně (AvgDeg). Každá tabulka je rozdělena na 4 části. První a druhá část obsahuje hodnoty pro sítě získané za použití Gaussian kernel podobnosti. První část jsou sítě s průměrným stupně 3 a druhá část s průměrným stupněm 7. Třetí část obsahuje sítě o průměrném stupni 3 a čtvrtá část o průměrném stupni 4, jež byly získány použitím kosinové podobnosti.

Metoda	AvgDeg	Quality	MaxDeg	AvgCC	Comp.	Modul.	AvgCR.	Comm.	Community size		
									Min	Avg	Max
ε-radius	3,080	0,7971	13	0,580	15	0,756	0,705	19	2	6,316	18
KNN	2,787	0,9944	6	0,436	7	0,854	0,782	16	4	9,375	16
Combination	2,973	0,9938	11	0,438	21	0,817	0,670	25	2	6,000	20
LRNet	3,227	0,9935	12	0,479	5	0,821	0,816	12	2	12,500	25
ε-radius	7,013	0,9490	22	0,610	8	0,691	0,847	13	2	11,000	27
KNN	6,667	0,9938	12	0,514	2	0,765	0,931	8	11	18,750	27
Combination	7,027	0,9943	19	0,575	2	0,722	0,886	7	5	21,429	33
LRNet	7,013	0,9914	32	0,650	3	0,664	0,907	6	2	25,000	50
ε-radius	3,040	0,7732	11	0,585	11	0,803	0,853	15	2	7,733	20
KNN	2,760	0,9997	6	0,447	8	0,879	0,790	15	3	10,000	18
Combination	3,000	0,9997	11	0,448	21	0,834	0,683	25	2	6,000	21
LRNet	3,067	0,9997	11	0,437	9	0,817	0,765	14	2	10,714	27
ε-radius	7,027	0,8998	18	0,664	4	0,719	0,880	8	2	16,875	32
KNN	6,760	0,9997	11	0,532	2	0,750	0,847	8	12	18,750	33
Combination	6,973	0,9997	16	0,593	2	0,736	0,841	9	8	16,667	29
LRNet	7,040	0,9996	25	0,639	2	0,670	0,938	6	2	25,000	48

Tabulka A.1: Hodnoty vlastností sítí pro Iris dataset

Metoda	AvgDeg	Quality	MaxDeg	AvgCC	Comp.	Modul.	AvgCR.	Comm.	Community size		
									Min	Avg	Max
ε-radius	2,982	0,7075	15	0,428	15	0,759	0,716	22	2	10,864	26
KNN	2,905	0,9870	7	0,302	2	0,845	0,766	19	9	17,684	28
Combination	3,012	0,9845	13	0,310	28	0,828	0,516	36	2	9,333	34
LRNet	3,179	0,9839	17	0,355	9	0,829	0,733	21	2	16,000	51
ε-radius	7,018	0,8484	29	0,516	7	0,658	0,860	14	2	20,500	48
KNN	6,804	0,9862	15	0,433	2	0,762	0,876	11	10	30,545	70
Combination	7,006	0,9870	25	0,445	2	0,718	0,855	10	10	33,600	69
LRNet	7,113	0,9812	41	0,538	4	0,711	0,857	10	2	33,600	72
ε-radius	3,012	0,7278	15	0,456	15	0,767	0,700	24	2	10,208	33
KNN	2,845	0,9962	6	0,321	4	0,856	0,823	18	3	18,667	34
Combination	3,018	0,9954	13	0,327	35	0,836	0,630	45	2	7,466	35
LRNet	3,083	0,9952	16	0,407	8	0,840	0,762	18	2	18,667	45
ε-radius	6,994	0,8581	28	0,534	3	0,696	0,883	10	2	28,900	57
KNN	6,756	0,9964	13	0,416	1	0,744	0,883	10	13	33,600	61
Combination	6,988	0,9963	26	0,459	1	0,727	0,855	10	7	33,600	52
LRNet	7,006	0,9944	42	0,573	2	0,701	0,855	9	2	37,333	66

Tabulka A.2: Hodnoty vlastností sítí pro Ecoli dataset

Metoda	AvgDeg	Quality	MaxDeg	AvgCC	Comp.	Modul.	AvgCR.	Comm.	Community size		
									Min	Avg	Max
ε-radius	3,029	0,8115	12	0,503	11	0,801	0,790	17	2	10,118	31
KNN	2,848	0,9887	8	0,336	3	0,833	0,745	15	5	14,000	25
Combination	3,000	0,9858	12	0,394	17	0,844	0,683	25	2	8,400	30
LRNet	3,067	0,9849	20	0,431	11	0,799	0,718	17	2	12,353	37
ε-radius	7,010	0,9422	19	0,532	3	0,712	0,865	11	2	18,182	47
KNN	6,629	0,9886	13	0,452	1	0,726	0,887	9	17	23,333	30
Combination	7,000	0,9889	18	0,492	1	0,725	0,880	9	4	23,333	51
LRNet	6,905	0,9820	52	0,624	1	0,669	0,907	6	12	35,000	56
ε-radius	3,105	0,8428	13	0,576	19	0,865	0,687	23	2	7,696	21
KNN	2,667	0,9999	6	0,338	4	0,875	0,763	16	4	13,125	22
Combination	3,095	0,9998	13	0,456	30	0,877	0,591	34	2	6,176	27
LRNet	2,905	0,9998	12	0,505	13	0,853	0,682	23	2	9,130	19
ε-radius	6,990	0,9665	20	0,593	5	0,790	0,891	13	2	15,615	27
KNN	7,600	0,9998	12	0,535	1	0,772	0,855	9	11	23,333	42
Combination	6,933	0,9999	18	0,564	1	0,795	0,879	9	14	23,333	31
LRNet	7,010	0,9997	29	0,658	2	0,679	0,931	8	2	26,250	36

Tabulka A.3: Hodnoty vlastností sítí pro Seeds dataset

Metoda	AvgDeg	Quality	MaxDeg	AvgCC	Comp.	Modul.	AvgCR.	Comm.	Community size		
									Min	Avg	Max
ε-radius	3,000	0,8239	8	0,699	166	0,986	0,545	166	2	5,916	23
KNN	3,759	0,8871	8	0,575	72	0,982	0,516	74	6	14,595	44
Combination	3,006	0,8829	8	0,656	186	0,988	0,501	186	2	5,806	23
LRNet	3,270	0,8736	27	0,579	27	0,958	0,506	55	2	19,636	64
ε-radius	7,004	0,8609	30	0,717	59	0,912	0,580	70	2	15,071	129
KNN	7,444	0,8638	15	0,653	5	0,932	0,478	38	15	28,421	61
Combination	7,007	0,8742	30	0,693	32	0,920	0,594	49	6	22,041	125
LRNet	6,985	0,8583	68	0,573	1	0,845	0,772	22	15	49,091	111
ε-radius	2,994	0,9500	11	0,724	182	0,985	0,520	184	2	5,582	31
KNN	3,757	0,9986	9	0,596	69	0,981	0,508	73	4	14,795	38
Combination	3,000	0,9987	11	0,696	193	0,985	0,518	195	2	5,538	32
LRNet	3,252	0,9982	16	0,590	43	0,957	0,450	58	2	18,621	70
ε-radius	7,020	0,9857	33	0,731	61	0,907	0,571	70	2	15,229	103
KNN	7,381	0,9981	14	0,657	6	0,928	0,603	32	13	33,750	88
Combination	6,998	0,9984	33	0,696	36	0,915	0,665	46	6	23,478	105
LRNet	6,933	0,9976	39	0,589	4	0,856	0,767	17	2	63,529	178

Tabulka A.4: Hodnoty vlastností sítí pro Nuclear cortex dataset

Příloha B

Příloha v IS EDISON

Příloha obsahuje následující složky a soubory:

2021_CHO0163_DP.pdf

Kopie této práce ve formátu PDF/A.

Algoritmy detekce komunit

Tato složka obsahuje data použitá během experimentu s algoritmy detekce komunit (kapitola 6). Obsahuje tyto soubory a složky:

Složka Sítě Obsahuje soubory s použitými sítěmi. Soubory jsou pojmenované po síti, kterou obsahují. Jeden řádek v těchto souborech reprezentuje jednu hranu sítě, zapsanou dvojicí čísel oddělených tabulátorem reprezentující id uzlů hrany. Tyto soubory byly vstupy pro CRank.

Složka Komunity Obsahuje soubory s komunitami. Jména souborů obsahují název sítě, jíž nalezené komunity náleží, a název metody jež byla k nalezení komunit použita. Každá komunita je v souborech na samostatném řádku, kde na začátku je id komunity, které je následované id čísly uzlů, které jsou odděleny mezerou. Tyto soubory byly v kombinaci s příslušným souborem se sítí vstupem pro CRank.

Složka CRank Obsahuje soubory s výstupy CRanku. Jsou to soubory obsahující ohodnocení jednotlivých komunit mírami. Jména souborů obsahují název sítě, jejíž komunity byly hodnoceny, a název metody jež byla k nalezení komunit použita.

Složka Statistiky Obsahuje námi vytvořené soubory, jež obsahují statistiky komunit, které jsme použily k vytvoření grafů v první části práce. Jména souborů obsahují název sítě, které patří komunity, kterých se statistiky týkají, a metody, jež byla k nalezení komunit použita.

Script.R R skript, který načte statistiky ze složky Statistiky a vytvoří grafy použité v části práce zabývající se algoritmy detekce komunit.

Aplikace *Graph generator*

Obsahuje námi vytvořenou C# WinForms aplikaci, která je určena ke konstrukci sítí z vektorových dat. Aplikace využívá C# dynamickou knihovnu LouvainSharp [23] napsanou Markusem Mobiem, která implementuje Louvain metodu. Složka obsahuje soubory a složky:

Graph generator.exe Spustitelný soubor aplikace. Pro spuštění aplikace je vyžadován operační systém Windows či nainstalovaný Mono framework.

Graph generator.exe.config Konfigurační soubor aplikace.

LouvainCommunityPL.dll Dynamická knihovna s třídami určenými pro nalezení komunit pomocí Louvain metody a výpočet modularity. Toto je soubor s knihovnou LouvainSharp.

LouvainCommunityPL.dll.config Konfigurační soubor knihovny.

Návod k použití aplikace.txt Stručný návod k použití aplikace.

Složka Visual Studio projekt Obsahuje zdrojové kódy a soubory, z nichž byla sestavena aplikace. Obsahuje soubor **Graph generator.sln**, jež otevře projekt ve Visual Studiu. Je vyžadováno Visual Studio 2019 či novější. Ve Visual Studiu lze zdrojové kódy zkompilovat a aplikaci sestavit.

Metody konstrukce sítí z vektorových dat

Tato složka obsahuje data použitá během experimentu s metodami konstrukce sítí (grafů) z vektorových dat (kapitola 9). Obsahuje soubory a složky:

Složka Datasety Obsahuje soubory s použitými datasety. Soubory jsou pojmenované po datasetu, který obsahují. Ke každému datasetu je tam i jeho normalizovaná verze.

Složka Sítě Obsahuje 4 podsložky, kde každá je pojmenována po jednom datasetu. Každá tato podsložka obsahuje soubory se sítěmi vytvořené pro daný dataset. Soubory obsahují jméno použité podobnosti, jméno použité metody, průměrný stupeň získané sítě a použitou hodnotu parametru metody. Každá síť je zde uložena dvakrát. Jednou je v csv souboru, obsahující standardní seznam hran sítě, kde jeden řádek je jedna hrana ve formátu uzel1;uzel2;váha. Podruhé je uložena v gdf soubor, který lze otevřít pomocí nástrojů používaných k vizualizaci sítí, jako je například program Gephi. Kromě souborů se sítěmi obsahuje každá podsložka ještě složku Communities, kde jsou soubory obsahující komunity pro každou síť (jsou pojmenované

stejně jako sítě akorát s com_ na začátku). Komunity jsou uloženy ve stejném formátu, jako při experimentu s algoritmy detekce komunit.

Vlastnosti sítí.xlsx Excel soubor, který obsahuje data o vlastnostech získaných sítí.

Load datasets.R R skript pro načtení dat uložených v souboru Vlastnosti sítí.xslx nutných pro vytvoření grafů.

Graphs.R R skript pro vytvoření grafů použitých v části práce zabývající se metodami konstrukce sítí z vektorových dat. Před jeho použitím je nutné nahrát data pomocí skriptu Load datasets.R.

Aplikace *Network quality*

Obsahuje námi vytvořenou C# WinForms aplikaci, která je určena k výpočtu hodnoty pravděpodobností účelové funkce (sekce 8.2). Složka obsahuje soubory a složky:

Network quality form.exe Spustitelný soubor aplikace. Pro spuštění aplikace je vyžadován operační systém Windows či nainstalovaný Mono framework.

Network quality.dll Dynamická knihovna s třídami reprezentující síť potřebných pro spuštění aplikace.

Network quality form.exe.config Konfigurační soubor aplikace.

Návod k použití aplikace.txt Stručný návod k použití aplikace.

Složka Visual Studio projekt Obsahuje zdrojové kódy a soubory, z nichž byla sestavena aplikace. Obsahuje soubor **Network quality.sln**, jež otevře projekt ve Visual Studiu. Je vyžadováno Visual Studio 2019 či novější. Ve Visual Studiu lze zdrojové kódy zkompilovat a aplikaci sestavit.