UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Mitja Kuščer

Generiranje naključnih omrežij z metodo strojnega učenja

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: izr. prof. dr. Marko Bajec

Ljubljana, 2009

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

Namesto te strani **vstavite** original izdane teme diplomskega dela s podpisom mentorja in dekana ter žigom fakultete, ki ga diplomant dvigne v študentskem referatu, preden odda izdelek v vezavo!

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Mitja Kuščer,

z vpisno številko 63030308,

sem avtor diplomskega dela z naslovom:

Generiranje naključnih omrežij z metodo strojnega učenja

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom izr. prof. dr. Marka Bajca
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 25.11.2009 Podpis avtorja:

Zahvala

Zahvalil bi se mentorju, izr. prof. dr. Marku Bajcu, za pomoč in usmerjanje med izdelavo tega diplomskega dela. Prav tako bi se posebej zahvalil Lovru Šublju za ves trud, ideje in nasvete, s katerimi je močno pripomogel k uspešnemu zaključku tega dela. Ne nazadnje pa bi se za moralno podporo in spodbudo med celotnim študijem zahvalil svojim staršem in Katji.

Kazalo

Seznam uporabljenih kratic in simbolov

Oznaka	Opis
N	omrežje
V	množica vozlišč
E	množica povezav
n	velikost omrežja, podana s številom vozlišč
e	število povezav
v_i	i-to vozlišče
w	pot
$k(v)$	stopnja vozlišča v
k_i	stopnja i-tega vozlišča
$p(k)$	verjetnost, da ima vozlišče stopnjo k
k_{max}	največja stopnja vozlišč
α	eksponent potenčne funkcije
R^2	napaka pri aproksimaciji potenčne funkcije
d_{ij}	najkrajša razdalja med vozlišči i in j
l	premer omrežja
h	dolžina poti w
$N_h(v)$	velikost množice (okolice) vozlišč v oddaljenosti h od vozlišča v
N_h	velikost celotne okolice v oddaljenosti h
$M(v,h)$	množica vozlišč v oddaljenosti h od vozlišča v
b	najnižji bit v dvojiškem zapisu, ki ni postavljen
H	eksponent hop
$V_{i,j}$	vrednost pri hierarhičnem razvrščanju
C_i	lokalni koeficient razvrščanja za vozlišče i
C	koeficient razvrščanja
n_i	število povezav med sosednjimi vozlišči vozlišča v_i
p	verjetnost

$N_{n,p}$	naključni model omrežij
$\bar{z}$	povprečna stopnja vozlišč
$I(v)$	vhodna stopnja vozlišča
s	število najbližjih vozlišč

Povzetek

Pri raziskovanju odnosov med podatkovnimi entitetami je najnaravnejši način predstavitve le-teh z omrežji. Pri gradnji omrežij iz podatkov pa pogosto naletimo na omejitev, da nimamo na razpolago dovolj ustreznih podatkov, s katerimi bi lahko zgradili popolno omrežje. V takšnih primerih lahko zgradimo zgolj manjša ali nepopolna omrežja, katerih uporabnost je v nadaljnjih analizah relativno omejena. Takrat se pogosto zatekamo h gradnji novih, naključnih omrežij.

V nalogi predstavimo nov pristop generiranja naključnih omrežij, ki ga poimenujemo M-generator. Naloga M-generatorja je samodejno analizirati razpoložljivo omrežje in na podlagi izbranih lastnosti generirati naključno omrežje, ki te lastnosti v čim večji meri posnema. Uporabimo metodo strojnega učenja, ki na podlagi analize nad osnovnim omrežjem izbere najprimernejši model naključnega omrežja, s katerim nato generira naključno omrežje poljubne velikosti. Analiza in izbira naključnega modela poteka povsem samodejno, tako da posredovanje domenskega eksperta pri izbiri lastnosti omrežij in izbiri naključnega modela ni potrebno.

Delovanje modela preizkusimo nad realnim naborom omrežij, pri čemer lastnosti naključno generiranih omrežij lepo sledijo realnim. Vendar zaradi nekoliko manjšega vzorca in pomanjkanja označenih podatkov ne moremo sklepati o uspešnosti pristopa v splošnem. Kljub temu smo z dobljenimi rezultati zelo zadovoljni, saj smo z modelom uspeli samodejno generirati zelo dobra naključna omrežja.

Ključne besede:

omrežja, modeli omrežij, M-generator, strojno učenje

Abstract

When researching relationships between data entities, the most natural way of presenting them is by using networks. When constructing networks from data, the lack of relevant data often prevents us from building a complete network. In such cases, we are only able to build small or incomplete networks, which are of very limited use in the further analysis. We then often solve this problem by constructing new, random networks.

This paper presents a new approach to generating random networks, which is called M-generator. The task of M-generator is to automatically analyze the available network, and on the basis of selected properties generate a random network that follows these properties. To select optimal network model to generate random network, we use a machine learning method, based on the analysis of the original network. Analysis and selection of a random network is fully automated, so that the presence of a domain expert in selecting network properties and selecting a random network model is not required.

Model operation was tested on real world data, where random network properties seemed to follow the real world ones. However, due to slightly smaller sample size and the lack of labelled data, we can not estimate the efficiency in general. Despite that, we are satisfied with the results, as we managed to automatically generate really good random networks.

Key words:

networks, network models, M-generator, machine learning

1 Uvod

Živimo v svetu informacijske tehnologije, kjer je velika količina informacij ali podatkov dostopna v elektronski obliki. Podatki so navadno shranjeni v velikih relacijskih podatkovnih bazah, predstavimo pa jih lahko na več načinov. V primeru raziskovanja odnosov med podatkovnimi entitetami je najnaravnejši način predstavitve podatkov z omrežji. V zadnjem času so še zlasti velikega zanimanja deležna socialna omrežja, kjer vozlišča predstavljajo opazovane osebke, odnose med njimi pa predstavljajo povezave.

Pri gradnji omrežij pa pogosto ugotovimo, da nimamo na voljo dovolj podatkov, s katerimi bi lahko zgradili popolno omrežje. Zato lahko v takšnih primerih zgradimo zgolj manjša in nepopolna omrežja. Uporabnost le-teh je v nadaljnjih analizah ali testiranjih relativno omejena, zato se pogosto zatekamo h gradnji novih, naključnih omrežij. Tako želimo zgraditi naključna omrežja, ki bi posnemala realna omrežja, njihova velikost pa ne bi bila omejena z razpoložljivim naborom podatkov.

V preteklosti se je pojavilo veliko različnih pristopov h generiranju omrežij. Težava oziroma pomanjkljivost večine pristopov pa je v tem, da so navadno zelo specifični in ozko usmerjeni ter zato uspejo generirati zgolj omrežja določene vrste. Pri generiranju naključnega omrežja potrebujemo pristop, ki bo primeren za naš primer, zato se pogosto vprašamo, katero metodo naj uporabimo oziroma ali sploh obstaja primerna metoda za naš specifičen primer.

V nadaljevanju naloge najprej v razdelku 1.1 predstavimo sorodno delo, ki mu v razdelku 1.2 sledi še natančen opis ciljev naloge. Nato v razdelku 2 na kratko predstavimo teorijo omrežij, ki jo razširimo z opisom omrežij iz realnega sveta (in njihovimi lastnostmi). V razdelku 3 opišemo modele naključnih omrežij, ki jih uporabimo v nadaljevanju naloge. Strojno učenje iz naključnih omrežij in podrobnejši opis našega pristopa predstavimo v razdelku 4. Na koncu naloge v razdelku 5 podamo še kritično oceno in predloge za nadaljnje delo.

1.1 Sorodno delo

Sami modeli naključnih omrežij so bili v literaturi podrobno proučevani. Avtorji so predlagali različne modele [8, 9, 15, 30, 40], ki imajo vsak svoje prednosti in pomanjkljivosti. Tako se veliko pozornosti posveča analizi modelov, vendar pa se izbira modela v končni fazi vedno prepusti domenskemu ekspertu.

Avtomatičnemu odločanju o izbiri najprimernejšega modela iz množice modelov ni bilo posvečene večje pozornosti. Bezáková et al. [5] predlaga izbiro najustreznejšega modela naključnih omrežij s pomočjo največje verjetnosti (*maximum likelihood*). Pristop temelji na dejstvu, da modeli naključnih omrežij predstavljajo verjetnostno porazdelitev čez omrežja in tako za model naključnega omrežja izbere tistega, ki v svoji porazdelitvi realnemu omrežju pripiše največjo verjetnost. Slaba stran pristopa je v tem, da potrebujemo algoritem za vsak model naključnih omrežij, ki izračuna največjo verjetnost, za kar predlaga uporabo Monte Carlo Markovskih verig (*Monte Carlo Markov Chains*). Pri našem pristopu dodatni algoritmi niso potrebni in je dovolj, da z modeli naključnih omrežij generiramo zadostno število naključnih omrežij, kar omogoča preprostejše dodajanje modelov.

Podobna našemu pristopu so eksponentna naključna omrežja (*exponential random graphs*) oziroma v bolj splošni obliki modeli *p** [42], ki prav tako omogočajo gradnjo omrežja, ki naj bi sledila lastnostim osnovnega omrežja. Osnovno omrežje se analizira in zanj izbere množica lastnosti (na primer število povezav ali število vozlišč določene stopnje), ki se zdijo pomembne. Omrežje se nato konstruira z verjetnostjo enako uteženi kombinaciji izbranih lastnosti. Vendar pa se omejenost pristopa pokaže v tem, da ne omogoča razširitve z novimi modeli, ampak se v celoti zanaša na lastno generiranje omrežij.

Podoben, a že v začetni fazi različen pristop predlaga Leskovec [26]. Predlagani model, Kroneckerjev graf, naključno omrežje generira s postopnim povečevanjem osnovnega modela. Temelji na Kroneckerjevem produktu (*Kronecker product*) nad matrikami, ki jih izvaja nad matriko sosednosti (*adjacency matrix*). Pristop je vsekakor zelo zanimiv in si zasluži večje pozornosti, vendar za nas ne pride v poštev, saj bi radi naključno omrežje zgradili povsem samostojno, brez določitve začetnega omrežja.

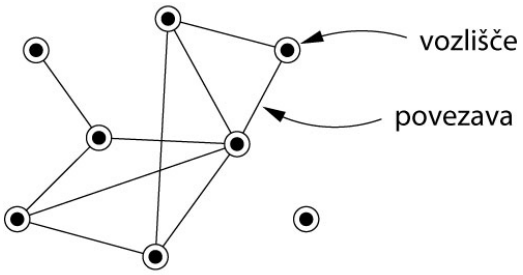
1.2 Cilj

Primarni cilj naloge je razvoj novega pristopa za generiranje naključnih omrežij. S pristopom želimo generirati naključna omrežja, ki v svojih lastnostih čimbolj posnemajo realna omrežja. Slednje želimo doseči brez posredovanja domenskega eksperta in tako povsem avtomatizirati postopek generiranja naključnega omrežja. Predlagan pristop tudi kritično ocenimo in preizkusimo na množici različnih realnih primerov.

2 Teorija omrežij

V nadaljevanju podamo teoretično podlago omrežij. Nato si v naslednjem razdelku ogledamo razdelitev realnih omrežij v smiselne skupine ter zatem značilne lastnosti omrežij, ki se pojavijo v realnih omrežjih.

Omrežje[1] N definiramo kot množico vozlišč in povezav med njimi. Označimo $N = (V, E)$, kjer je $V = \{v_1, v_2, \ldots, v_n\}$ množica vozlišč v omrežju in $E \subseteq V \times V$ množica povezav med vozlišči. Velikost omrežja je velikost množice V, ki jo označimo z n.



Slika 2.1: Primer neusmerjenega omrežja z osmimi vozlišči in desetimi povezavami. [30]

Za neusmerjeno omrežje velja, da vsebuje le neusmerjene povezave, pri katerih je vsaka povezava predstavljena z množico dveh vozlišč.

$$E \subseteq \{\{v_i, v_j\} | v_i, v_j \in V\} \tag{2.1}$$

Za usmerjeno omrežje pa velja, da so vse povezave usmerjene. Usmerjene povezave so tako predstavljene z urejenim parom vozlišč

[1] V matematični teoriji omrežja imenujemo grafi.

$$E \subseteq \{(v_i, v_j) \mid v_i, v_j \in V\}, \quad \textbf{(2.2)}$$

kjer usmerjena povezava (v_i,v_j) poteka iz vozlišča v_i v vozlišče v_j.

Povezavam (v_i, v_i) pravimo zanka, dvema povezavama z isto začetno in končno točko pa vzporedni povezavi. Če omrežje nima zank in vzporednih povezav ga imenujemo enostavno omrežje.

Omrežje lahko v matematičnem zapisu predstavimo z matriko sosednosti (*adjacency matrix*). Matrika sosednosti vozlišč omrežja N z n vozlišči je $n \times n$ kvadratna matrika $A(N)$, kjer element matrike a_{ij} predstavlja število povezav iz vozlišča v_i v v_j. Enostavna omrežja lahko prikažemo z binarno matriko, kjer vsak element matrike a_{ij} zavzame zgolj vrednost *0* ali *1*, elementi na diagonali (a_{ii}) pa so vedno enaki *0*. V primeru neusmerjenega omrežja velja, da je matrika sosednosti simetrična.

Pot w po omrežju N je zaporedje vozlišč $v_{i_1}, v_{i_2}, ..., v_{i_t}$ kjer je $(v_{v_i}, v_{v_{i+1}}) \in E$ za vse $1 \le i < t$. Vozlišče v_i je sosed vozlišča v_j (in nasprotno), če med njima obstaja pot dolžine ena. Omrežje je povezano, če za poljubni dve točki omrežja obstaja pot med njima. Takšnim omrežjem tudi pravimo, da so sestavljena iz ene same komponente. Omrežje je v splošnem lahko sestavljeno iz največ n komponent.

Omrežje je lahko uteženo ali neuteženo. Pri uteženih omrežjih ima vsaka povezava določeno neko ne-negativno utež. Povezave lahko poleg uteži vsebujejo tudi druge numerične ali nenumerične lastnosti. Enako velja za vozlišča (na primer spol, starost ipd. v socialnih omrežjih). Povezave v omrežjih lahko med seboj povežejo tudi več kot dve vozlišči. Takšnim povezavam pravimo hiperpovezave, omrežjem pa hiperomrežja.

V naši raziskavi se omejimo na neusmerjena, enostavna in neutežena omrežja s poljubnim številom komponent.

2.1 Omrežja v realnem svetu

Teorijo omrežij smo že spoznali v prejšnjem razdelku. Da pa bi lahko uspešno generirali naključna omrežja, moramo najprej poznati njihove lastnosti. Različna omrežja v realnem svetu se med seboj razlikujejo, imajo pa tudi presenetljivo veliko skupnih značilnosti.

V zadnjem času so matematiki proučevali predvsem naslednja omrežja [30]:

- **Socialna omrežja**
  Omrežja, ki so v zadnjem času najbolj aktualna, so vsekakor socialna omrežja. Socialno omrežje je množica ljudi, ki predstavljajo vozlišča, povezave med njimi pa

predstavljajo neko interakcijo med njimi. Mednje sodijo omrežja prijateljstev oziroma poznanstev med ljudmi, omrežja poslovnih sodelovanj med podjetji in omrežje telefonskih klicev med uporabniki mobilne telefonije.

- **Informacijska omrežja**
  Najbolj splošno znano informacijsko omrežje je omrežje spletnih strani in povezav med njimi (*World Wide Web* – a ga ne smemo zamenjevati z internetom kot fizičnimi povezavami med računalniki, ki jih bomo spoznali v nadaljevanju). V stroki je klasični primer omrežje citiranj med znanstvenimi članki. Vozlišča predstavljajo članki, usmerjene povezave iz vozlišča A v B pa prikazujejo, da je članek A citiral članek B.

- **Tehnološka omrežja**
  Pri tehnoloških omrežjih gre za omrežja, ki jih je ustvaril človek za distribucijo dobrin, kot je na primer energija ali voda. Značilen primer je električno omrežje, kjer proučujemo električno omrežje države ali večjega mesta. Veliko pozornosti se posveča tudi internetu. Tukaj govorimo o fizičnem omrežju povezav med različnimi računalniki, ki so skupaj povezani v eno veliko omrežje. Vendar pa je celotno sliko fizičnih povezav med vsemi računalniki praktično nemogoče zajeti, saj pripadajo posamezni deli omrežja različnim organizacijam.

- **Biološka omrežja**
  Omrežja so bila proučevana tudi za uporabo v biologiji. Primeri bioloških mrež so omrežja metaboličnih poti, gensko urejevalno omrežje in nevronska omrežja. Zanimiv je tudi primer biološkega omrežja prehranjevanja, kjer vsako vozlišče predstavlja določeno vrsto živali, usmerjena povezava med vozliščema A in B pa pomeni, da se žival A prehranjuje z živaljo B.

2.2 Lastnosti omrežij

Kljub temu, da so vsa opisana omrežja različna in da nastanejo neodvisno v povsem različnih okoliščinah, ugotavljamo da imajo veliko skupnih značilnosti. V nadaljevanju si ogledamo te značilnosti ter jih skušamo primerjati z naključno generiranimi omrežji. Skušamo naključno generirati takšna omrežja, ki imajo podobne lastnosti kot prava omrežja.

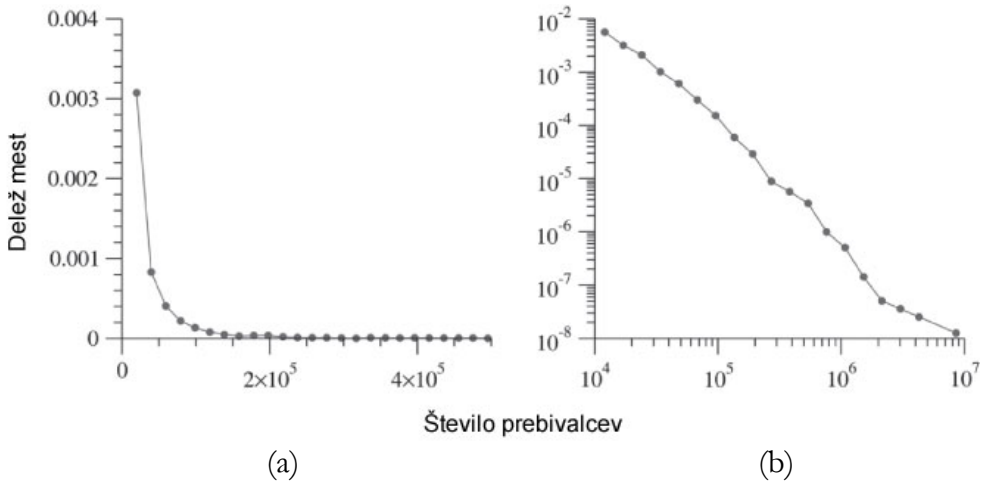
Najpomembnejše se zdijo naslednje lastnosti [8]: potenčni zakoni, majhni premeri in skupnosti. Te si ogledamo najprej, poleg njih pa še največjo stopnjo vozlišča, povprečno stopnjo vozlišča ter eksponent hop (*hop exponent*). Ti so, kot ugotovimo v raziskavi, prav tako pomembni pri primerjanju omrežij ter generiranju naključnih omrežij.

2.2.1 Potenčni zakon

Da bi razumeli potenčni zakon, si najprej pogledamo, kaj je to porazdelitev stopenj in kaj sploh je stopnja vozlišča.

Stopnja vozlišča je preprosto povedano število povezav, ki se povezujejo z vozliščem. Ni nujno, da je stopnja vozlišča enaka številu njegovih sosedov, saj lahko vozlišče vsebuje zanke ali vzporedne povezave. V usmerjenih omrežjih poznamo vhodno in izhodno stopnjo vozlišč. Vhodna stopnja vozlišča nam pove, koliko povezav je usmerjenih v vozlišče, izhodna pa, koliko jih gre iz vozlišča.

S $p(k)$ določimo delež vozlišč omrežja N, ki imajo stopnjo k. V nadaljevanju je pomembno tudi dejstvo, da je $p(k)$ verjetnost, da ima naključno izbrano vozlišče stopnjo k. S histogramom stopenj vozlišč lahko nato narišemo porazdelitev stopenj omrežja.



Slika 2.2: Histogram števila prebivalcev v mestih ZDA (a). Histogram z istimi podatki, izrisan z logaritemsko skalo (b). [29]

Z analizo realnih omrežij ugotavljamo, da imajo večinoma porazdelitev stopenj močno razpotegnjeno v desno (*right skewed*) in so daleč od binomske porazdelitve stopenj. Izkaže se, da za veliko omrežij velja, da njihova porazdelitev stopenj sledi potenčnemu zakonu [10].

Potenčna funkcija je funkcija oblike $y(x) = Ax^{-\alpha}$, kjer sta A in α pozitivni konstanti. α imenujemo tudi eksponent potenčne funkcije. Naključna spremenljivka pa je porazdeljena po potenčni porazdelitvi, če velja da njena porazdelitev sledi funkciji

$$p(x) = Ax^{-\alpha}, \alpha > 1, x \geq x_{\min} \qquad \textbf{(2.3)}$$

Lepa lastnost potenčne porazdelitve je ta, da se jo da eksperimentalno zelo lepo opaziti. Na primer, če ima omrežje porazdelitev stopenj, ki sledi potenčnemu zakonu, in to porazdelitev narišemo na graf z logaritemsko skalo (»log-log« graf), se mora porazdelitev stopenj prilegati premici (slika 2.2b). Koeficient A pri premici oblike $y(x) = Ax + B$ na logaritemski skali ustreza eksponentu potenčne funkcije α. Na ta način se da z uporabo linearne regresije eksponent potenčne funkcije zelo preprosto in predvsem učinkovito izračunati.

Omrežja s porazdelitvijo stopenj, ki sledi potenčnemu zakonu, pogosto imenujemo *scale-free* omrežja [3]. Izraz *scale-free* se uporablja, ker se porazdelitev stopenj ne spreminja s spreminjanjem velikosti omrežja, ampak ne glede na velikost omrežja ostaja enaka. Eden prvih primerov *scale-free* omrežja je omrežje citiranj med znanstvenimi članki, ki ga je leta 1965 objavil Price [34] in zanj navedel eksponent α med *2.5* in *3*. Pozneje je objavil natančnejše rezultate, kjer je izračunal eksponent $\alpha = 3.04$ [33].

Pogosto nas pri računanju porazdelitve stopenj in analiziranju omrežja zanima tudi največja stopnja vozlišča v omrežju k_{max}. Za večino omrežij je k_{max} odvisna od velikosti omrežja. Obstajajo različni pristopi [30], ki k_{max} ocenijo iz velikosti omrežja in znane porazdelitve stopenj (na primer potenčna). Vendar pri naši raziskavi takšne ocene niti ne potrebujemo, saj se ukvarjamo z omrežji takšnih velikosti, da lahko k_{max} vedno natančno poiščemo. Podobno velja tudi za povprečno stopnjo vozlišč, ki jo preprosto izračunamo iz histograma porazdelitve stopenj vozlišč.

Poglejmo še nekaj težav, na katere naletimo pri računanju eksponenta potenčne funkcije. Veliko realnih omrežij ne sledi potenčni funkciji čez celotno porazdelitev, ampak se ta lahko pojavi samo v repu porazdelitve, kar lahko povzroči nenatančnost ocenjevalne funkcije. Zato se priporoča, da se računanje eksponenta ne izvaja čez celotno porazdelitev, ampak samo na delu, ki dejansko sledi potenčni funkciji. Ugotoviti, kje se ta del nahaja, je spet problem zase.

Prav tako mnogi avtorji trdijo, da je preprosto računanje naklona premice z linearno regresijo na porazdelitvi z logaritemsko skalo premalo natančno, saj obstajajo s tem postopkom vsaj trije problemi [8]:

- uporaba tega pristopa lahko vodi do pristranskih rezultatov [20],
- včasih se potenčna funkcija pojavi samo v repu porazdelitve, česar ta metoda ne upošteva,
- desna stran porazdelitve utegne vsebovati veliko šuma [29].

Preprosta izboljšava je, če širino zajema podatkov eksponentno povečujemo. To naredimo tako, da znotraj vsakega območja preštejemo število točk in jih delimo s širino območja. S tem zmanjšamo šum v repu porazdelitve, vendar zmanjšamo tudi natančnost. Eksponent potenčne funkcije lahko izračunamo tudi na krivulji skupne porazdelitve (*cumulative distribution function - CDF*), ki je določena s funkcijo

$$F(x) = P(X \geq x) = \sum_{z=x}^{\infty} p(z) = \sum_{z=x}^{\infty} Az^{-y} \qquad \textbf{(2.4)}$$

Eksponent α je v tem primeru enak $\alpha = (y - 1)$ [8]. S tem se sicer izognemo izgubi natančnosti, vendar pa zaporedne točke med seboj niso neodvisne, kar lahko povzroči probleme.

Poleg računanja naklona z linearno regresijo v logaritemskem prostoru, obstaja še vrsta drugih pristopov [10, 20, 29], ki zagotavljajo boljše rezultate, vendar v praksi niso velikokrat uporabljani. Večinoma že osnovni pristop zagotavlja zadovoljive rezultate, kar je poleg preprostosti verjetno ključni razlog, da se ne posega po drugih metodah. Tako je tudi v našem primeru, ko se zadovoljimo že z rezultati osnovnega pristopa.

2.2.2 Majhen premer

Leta 1969 sta Travers in Milgram [41] izvedla znani poskus, kjer sta udeležencem zadala nalogo poslati verižno pismo naključno izbranim posameznikom. Kljub temu, da velika večina pisem ni nikoli prispela do cilja, jih je vseeno prispelo dovolj, da sta prišla do osupljivih zaključkov. Izkazalo se je, da je bila povprečna dolžina poti prispelih pisem samo šest korakov, kar je presenetljivo malo glede na velikost populacije, iz katere so bili udeleženci izbrani. To je bil prvi dokaz tako imenovanega učinka majhnega sveta, iz katerega se je šele pozneje razvil izraz »Šest stopenj separacije« (*Six degrees of separation*). S premerom skušamo zajeti točno to lastnost omrežja.

Vzemimo dve vozlišči v_i in v_j. Najkrajša razdalja d_{ij} med njima je dolžina najkrajše poti w med njima, ki smo jo že predhodno definirali. V splošnem je premer definiran kot povprečna razdalja med vsemi vozlišči v omrežju

$$l = \frac{1}{\left(\frac{n(n+1)}{2}\right)} \sum_{i>j} d_{ij} \qquad \textbf{(2.5)}$$

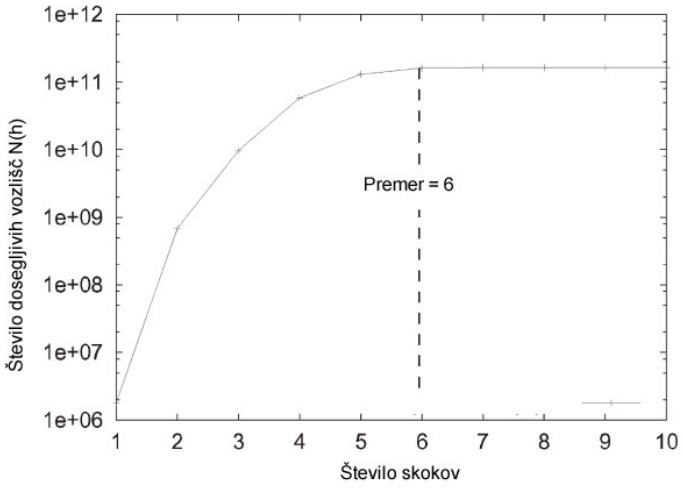
Problem s to enačbo se pojavi, ko naletimo na omrežja, sestavljena iz več komponent. Razdalja med dvema vozliščema v različnih komponentah je po tej enačbi neskončno in posledično je tudi premer vedno enak neskončno, kar nam ne poda praktično nobene informacije o omrežju.

Zato premer rajši računamo na drugačen način. Z $N_h(v)$ označimo velikost množice vozlišč, dosegljive iz vozlišča v z dolžino poti največ h. Začnemo v vozlišču v in poiščemo število

vozlišč v njegovi okolici $N_h(v)$. To ponovimo za vsa vozlišča v omrežju in seštejemo rezultate, da dobimo velikost celotne okolice

$$N_h = \sum_u N_h(u) \qquad \textbf{(2.6)}$$

Nato narišemo graf h proti N_h, ki ga imenujemo *hop-plot*. S pomočjo grafa *hop-plot* lahko izračunamo različne lastnosti omrežja (slika 2.3).



Slika 2.3: Primer grafa *hop-plot*. Vidimo, da se število dosegljivih vozlišč splošči pri 6 skokih, kar kaže na to, da je efektivni premer v tem primeru 6. [8]

Za nas je trenutno najbolj pomemben premer. S pomočjo grafa *hop-plot* lahko preprosto izračunamo tako imenovani efektivni premer [40]. Efektivni premer je minimalno število skokov, pri katerem nek del vseh vozlišč lahko doseže drug drugega. Avtorji večinoma navajajo, da je uporaben del vseh vozlišč pri *90 %* [8, 32, 40].

Omenimo še, da tudi funkcija $N(h) \approx h^H$ v veliko realnih omrežjih sledi potenčnemu zakonu [32]. Eksponent H imenujemo tudi eksponent hop (*hop exponent*) in se ga lahko izračuna na enak način, kot eksponent potenčne funkcije, ki smo ga spoznali pri porazdelitvi stopenj. Eksponent hop nam lahko poda tudi neformalno dimenzionalnost omrežja. Cikel ima eksponent hop 1, mreža povezav pa 2, kar nam da neko mero o dimenzionalnosti omrežja. Kljub temu, da ima eksponent hop kar nekaj lepih lastnosti, je po navadi izračunan iz dokaj omejenega števila točk – premeri so po navadi manjši od 10 – zato je njegova uporaba relativno omejena [8].

Pri analiziranju testne množice realnih omrežij kmalu naletimo na težavo, saj je implementacija opisanega algoritma prepočasna. Iterativno pregledovanje omrežja in

računanje funkcije $N(h)$ na zgoraj opisan način se izkaže za prepotratno. Funkcijo $N(h)$ zato rajši aproksimiramo z metodo ANF [32].

Zamisel algoritma ANF je naslednja. Namesto da velikost okolice $N_h(v)$ iščemo iz vozlišča v v h korakih, okolico h vozlišča v rajši izračunamo tako, da najprej poiščemo tista vozlišča, ki jih lahko njegovi sosedi dosežejo v $h-1$ korakih. Definiramo množico $M(v,h)$, ki vsebuje vsa vozlišča v oddaljenosti h korakov od vozlišča v. Začnemo z $M(v,0) = \{v\}$, saj je očitno edino vozlišče, ki je dosegljivo z 0 koraki, vozlišče samo. V vsakem koraku okolico h povečujemo za 1. Če sta vozlišči v_i in v_j sosednji, velja naslednje

$$M(v_i, h) = M(v_i, h-1) \cup M(v_j, h-1) \qquad \textbf{(2.7)}$$

S pomočjo te enačbe nato v korakih gradimo množico $M(v,h)$ za vsa vozlišča omrežja. Očitno je, da je velikost množice $M(v,h)$ enaka že prej definirani funkciji $N_h(v)$.

Za hranjenje množice $M(v,h)$ v pomnilniku vsakemu vozlišču priredimo svoj bit, *1* pomeni, da je vozlišče v množici, *0* pa, da ga ni. Unijo med množicama v tem primeru predstavlja preprosta *ALI* (*OR*) operacija med množicama. Vendar je pomnilniška zahtevnost tega pristopa zelo velika ($O(n^2)$), saj naenkrat v pomnilniku hranimo množice vozlišč za vse oddaljenosti od vseh vozlišč.

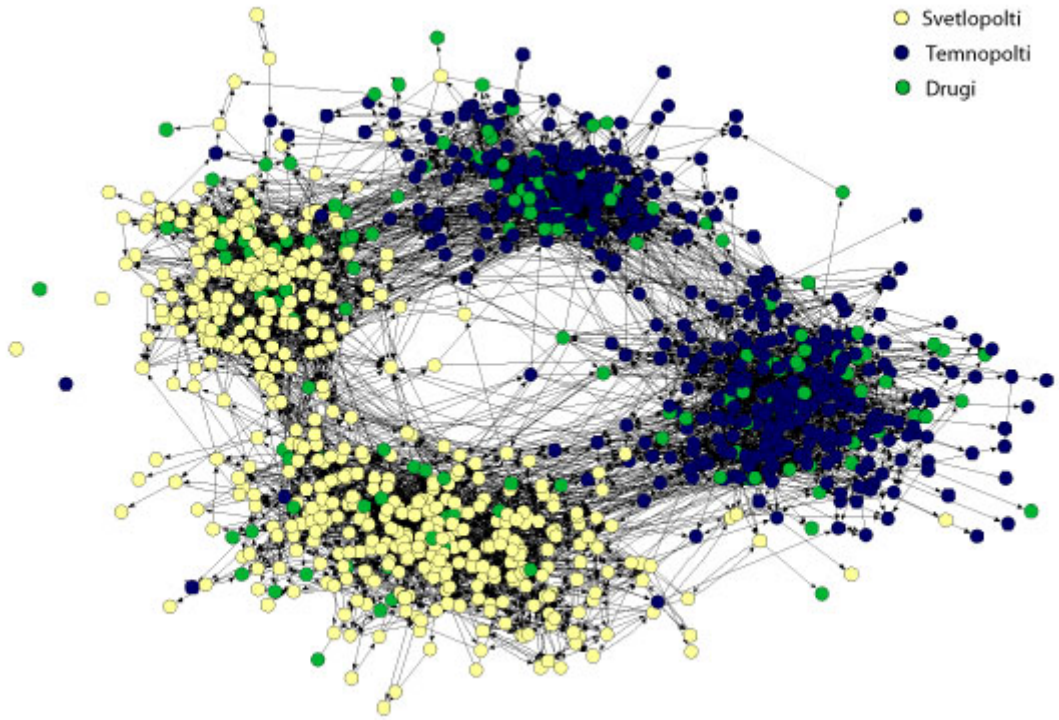
Pri hranjenju teh množic v pomnilniku zato posežemo po aproksimaciji. Pomagamo si z algoritmom za verjetnostno preštevanje (*probabilistic counting algorithm*) [16]. Namesto hranjenja n bitov za vsako množico vozlišč uporabimo množico velikosti $\log(n)$. Polovica vozlišč dobi bit 0, četrtina bit 1, osmina bit 2 itd. (vozlišču je prirejen bit i z verjetnostjo $1/2^{i+1}$). Prisotnost vozlišča še vedno označimo s postavitvijo njegovega bita na 1, unijo pa z operacijo *ALI* (*OR*). Začetno množico $M(v,0)$ napolnimo naključno z eksponentno porazdelitvijo.

Velikost množice vozlišč se oceni na naslednji način. Če predpostavimo, da je bitu 1 prirejeno 25 % vozlišč ter je bit 1 postavljen na 0 (nismo naleteli na nobeno od teh vozlišč), lahko predvidevamo, da nismo videli več kot štirih vozlišč. Velikost množice se lahko tako oceni z 2^b, kjer je z b označen najnižji bit, ki še ni postavljen. Vendar kot vidimo, samo enkratna aproksimacija ni dovolj robustna, zato naredimo k vzporednih aproksimacij in za velikost množice vzamemo njihovo povprečje [32].

Algoritem ANF se s tem ne konča, ampak ponuja nadaljnjo optimizacijo za potrebe večjih omrežij, ki jih ne moremo v celoti shraniti v pomnilnik. Omogoča deljenje množice $M(v,h)$ na več podmnožic, ki jih nato selektivno prenašamo v pomnilnik. Za omrežja v tej raziskavi to ni potrebno, saj je že opisani algoritem dovolj zmogljiv. Računanje premera z algoritmom ANF se tako izkaže za zelo učinkovito, saj z lahkoto izračunamo velikost okolice N_h. Tudi glede natančnosti ne zaostaja preveč za eksaktnim izračunom, zato je v nadaljnjih analizah uporabljen ta pristop.

2.2.3 Skupnosti

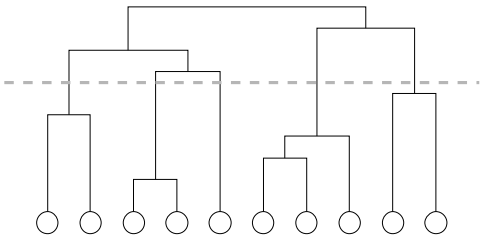
Za skupnosti v splošnem velja, da je množica vozlišč znotraj skupnosti med seboj povezana s krajšimi razdaljami, kot z vozlišči zunaj skupnosti. Pojav skupnosti je značilen za mnoga realna omrežja, še zlasti pa je izrazit v socialnih omrežjih. Avtorji so odkrili pojav skupnosti znotraj rase in starosti v omrežju prijateljstev na ameriški šoli (slika 2.4) [28], skupnosti na podlagi skupnih interesov v e-poštnih sporočilih [35] ter skupnosti spletnih strani na svetovnem spletu [17].



Slika 2.4: Omrežje prijateljstev na šoli v ZDA, kjer se lepo vidijo skupnosti znotraj omrežja. Vidimo, da se po večini med seboj družijo otroci iste rase. Vodoravna razdelitev se pojavi med otroci osnovne in srednje šole. [28, 30]

Skupnosti lahko odkrivamo in raziskujemo na različne načine. Tradicionalen pristop se imenuje hierarhično razvrščanje (*hierarchical clustering*). Pri tem pristopu vsakemu paru vozlišč v_i, v_j v omrežju priredimo neko vrednost $V_{i,j}$. To vrednost priredimo vsem parom vozlišč, ne samo povezanim, in se razlikuje od uteži povezav med vozlišči. Ta vrednost je lahko razdalja med vozlišči ali število različnih poti med vozlišči. Postopek hierarhičnega razvrščanja začnemo z množico vozlišč brez povezav, ki ji postopoma dodajamo nove povezave, najprej med tistimi vozlišči v_i, v_j z najvišjo vrednostjo $V_{i,j}$. V vsakem koraku vsaka povezana komponenta v omrežju predstavlja skupnost. Postopek lahko končamo po poljubnem številu korakov, ko dobimo želeno število komponent oziroma skupnosti.

Celoten postopek se lahko prikaže z drevesom oziroma dendrogramom, kjer skupnosti na posameznem koraku predstavlja vodoravni prerez skozi drevo.



Slika 2.5: Primer dendrograma z desetimi vozlišči. Vodoravna prekinjena črta prikazuje prerez skozi drevo, ki razdeli vozlišča v skupnosti. V tem primeru dobimo pet skupnosti. [30]

Vendar pri analiziranju omrežja ta pristop ni najbolj primeren, saj potrebujemo neko numerično oceno, ki poda oceno o skupnostih v omrežju. Zato si v nadaljevanju ogledamo še tranzitivnost oziroma koeficient razvrščanja (*clustering coefficient*), ki v literaturi še zlasti izstopa kot mera za gostoto skupnosti.

Koeficient razvrščanja izmeri število tranzitivnih povezav v omrežju. V veliko realnih omrežjih se namreč izkaže, da če sta med seboj povezani vozlišči v_i in v_j ter v_j in v_k, obstaja velika verjetnost, da sta med seboj povezani tudi vozlišči v_i in v_k. V socialnih omrežjih se to kaže tako, da je prijatelj mojega prijatelja z veliko verjetnostjo tudi moj prijatelj.

Lokalni koeficient razvrščanja C_i [45] lahko na posameznem vozlišču v_i izračunamo tako, da preštejemo število sosedov vozlišča k_i ter število povezav med sosednjimi vozlišči n_i

$$C_i = \begin{cases} \frac{n_i}{k_i}; k_i > 1 \\ 0; \text{sicer} \end{cases} \quad \textbf{(2.8)}$$

Večja vrednost koeficienta pomeni večjo stopnjo tranzitivnosti, kar kaže na omrežje z bolj gostimi skupnostmi. Koeficient razvrščanja C na celotnem omrežju je enak povprečju lokalnih koeficientov

$$C = \sum_{i=1}^{N} \frac{C_i}{N} \quad \textbf{(2.9)}$$

Še boljši je naslednji pristop, ki se pogosto uporablja v sociološki literaturi in analizah [30]. Vemo, da tranzitivnost nastopi, če in samo če v omrežju obstajajo trikotniki povezav. To izkoristimo in koeficient razvrščanja C izračunamo kot

$$C = \frac{\text{3x število trikotnikov}}{\text{število povezanih trojk}}, \qquad \textbf{(2.10)}$$

kjer s povezano trojko imenujemo tri vozlišča, kjer je centralno vozlišče povezano z ostalima dvema vozliščema. S tem preštejemo delež povezanih trojk, ki imajo dopolnjeno tudi tretjo povezavo in so tako povezane v trikotnik. Faktor 3 je uporabljen zato, ker vsak trikotnik ustreza trem povezanim trojkam in s tem zagotovimo, da C leži v območju $0 \le C \le 1$.

Kot vidimo, tako definirani koeficient razvrščanja C ustreza naši predhodni definiciji tranzitivnosti in nam poda verjetnost, da sta dve vozlišči, ki sta sosednji istemu vozlišču, tudi sami povezani med seboj.

2.2.4 Ostale lastnosti

Poleg potenčnega zakona, premera in skupnosti so bile v literaturi proučevane tudi številne druge lastnosti omrežij, ki se pogosto pojavljajo v velikih realnih omrežjih. Prva od njih je elastičnost (*resilience*), ki proučuje odpornost omrežja na odpovedi oziroma odstranitev vozlišč. Iz že omenjene raziskave Traversa in Milgrama poleg majhnega premera sledi še en zanimiv pojav, in sicer navigacija v omrežju. Veliko je bila proučevana tudi vmesna centralnost vozlišč (*betweenness centrality*). Omenimo naj še velikost največje komponente, ki se v nekaterih primerih prav tako izkaže za pomembno lastnost. V komunikacijskih omrežjih, kot je na primer internetno omrežje, velikost največje komponente pove, kolikšen je del omrežja, znotraj katerega je mogoča komunikacija [30].

Elastičnost omrežja pove, kako odporno je omrežje na odstranjevanje vozlišč in/ali povezav med njimi. Omrežja se za svoje delovanje zanašajo na poti, ki potekajo po povezavah med vozlišči. Če se kateri od teh ključnih elementov odstrani, se poti podaljšujejo, vozlišča pa utegnejo postati nedostopna in komunikacija ni več mogoča. Realna omrežja imajo različno odpornost na odpovedi (elastičnost), večina jih je odpornih proti naključnim odpovedim, vendar se odpornost močno zmanjša, če odstranjujemo izbrana vozlišča (na primer vozlišča z največjimi stopnjami).

Travers in Milgram sta v svojem poskusu z verižnimi pismi ugotovila, da v socialnih omrežjih obstajajo kratke poti med posamezniki, ki se zdijo daleč narazen. Kar sta očitno spregledala, in je šele pozneje ugotovil Kleinberg [23, 24], je to, da ne samo, da obstajajo kratke poti med posamezniki, ampak tudi to, da se običajni ljudje odrežejo dobro pri iskanju le-teh. To je morda še bolj presenetljivo kot prvotno odkritje. Ljudje v poskusu niso imeli nobene informacije o omrežju, ki jih povezuje z njihovim naslovnikom, vendar so vseeno uspeli najti pot do njih. To nakazuje na posebno zgradbo omrežja, ki za naključna omrežja ne velja. Če bi bili sposobni zgraditi informacijska omrežja z isto stopnjo navigacije, kot velja za socialna

omrežja, bi lahko zgradili veliko učinkovitejše podatkovne baze in povezave med računalniki v internetnem omrežju [1, 2, 44].

V literaturi se pogosto pojavlja tudi vmesna centralnost vozlišč (*betweenness centrality*) [18]. Vmesna centralnost vozlišča pove, koliko najkrajših poti poteka čez vozlišče. S tem mislimo na najkrajše poti, ki potekajo med vsemi vozlišči v omrežju, enako kot pri računanju premera. Vmesna centralnost tako poda neko »pomembnost« vozlišča, podobno kot njegova stopnja. Pomembna je tudi pri opazovanju elastičnosti omrežja, saj pove koliko poti se bo podaljšalo, če izbrano vozlišče odstranimo. Ko analiziramo elastičnost omrežja, pri odstranjevanju vozlišč pogosto odstranjujemo vozlišča glede na njihovo vmesno centralnost. Prav vozlišča z največjo vmesno centralnostjo povzročijo največjo škodo omrežju, če se jih odstrani.

3 Modeli naključnih omrežij

Pri izvajanju testiranj in raziskovanj smo pogosto omejeni z velikostjo razpoložljive množice podatkov, zaradi česar lahko po navadi sestavimo zgolj omrežja omejenih velikosti, ali pa realnega omrežja za naš problem sploh ne moremo sestaviti. Zato si pogosto pomagamo z generiranjem naključnih omrežij, s čimer lahko v splošnem sestavimo omrežja poljubnih velikosti. Vendar morajo lastnosti naključno generiranih omrežij, ki smo jih spoznali v prejšnjem razdelku, ustrezati lastnostim realnih omrežij, sicer bi bilo omrežje za nas neuporabno. Modeli naključnih omrežij nam prav tako podajo sliko, kako se realna omrežja razvijajo in kateri procesi v omrežju privedejo do določenih vzorcev.

Modele naključnih omrežij lahko v grobem razdelimo v pet skupin [8]:

- **Naključni modeli omrežij**[2] (*Random graph models*)
  Omrežja se generira z naključnim procesom. Ti modeli so zanimivi predvsem zaradi lepih matematičnih lastnosti. Kljub temu, da ne modelirajo najbolje realnega sveta, se jih je v preteklosti podrobno preučevalo.

- **Modeli po načelu prednostne povezanosti** (*Preferential attachment models*)
  Ti modeli temeljijo na načelu »bogati bogatijo« (*the rich get richer*) in s tem pripeljejo do potenčnih zakonov v omrežju. V to skupino spada danes nekaj najzanimivejših modelov.

- **Geografski modeli** (*Geographical models*)
  Pri teh modelih se pri generiranju omrežja upošteva tudi geografska lokacija vozlišč. To je še zlasti pomembno pri generiranju komunikacijskih omrežij. Vpliv geografske lokacije se opazi tudi pri socialnih omrežjih.

[2] Poimenovanje naključni modeli omrežij ni najbolj posrečeno, saj se zlahka zamenja z bolj splošnim izrazom modelov naključnih omrežij. Kljub temu se v literaturi najbolj pogosto uporablja prav ta izraz.

- **Optimizacijski modeli** (*Optimization-based models*)
  Optimizacijski modeli skušajo optimizirati neko mero modela z uporabo čim manjšega števila sredstev, kar naj bi prav tako privedlo do potenčnih zakonov.

- **Modeli, prilagojeni posebnostim interneta** (*Internet-specific models*)
  Ti modeli so prilagojeni posebnim lastnostim internetnega omrežja.

V raziskavi želimo uporabiti čim bolj splošne modele, ki so uporabni v čim večjem številu primerov, zato si v nadaljevanju ogledamo glavne predstavnike prvih treh skupin. Te modele skušamo nato razširiti z uporabo novega modela R-MAT, ki je opisan v razdelku 3.4.

3.1 Naključni modeli omrežij

Najpreprostejši predstavnik naključnih modelov omrežij – in verjetno tudi vseh modelov naključnih omrežij nasploh – je Poissonov model omrežij. Model sta odkrila Solomonoff in Rapoport [38] in neodvisno Erdös in Rényi [15].

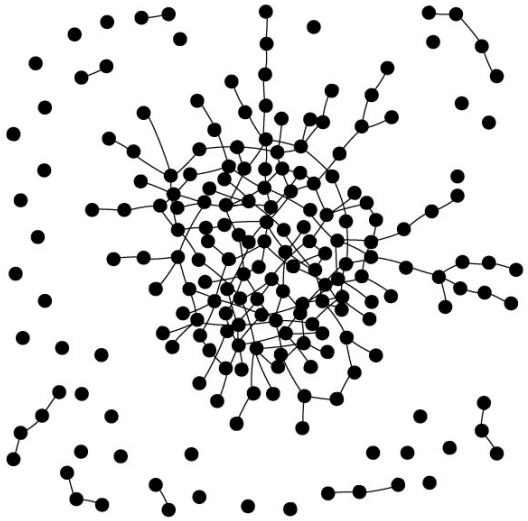
Definicija modela je preprosta. Vzamemo n vozlišč in povežemo vsak par vozlišč neodvisno med seboj z verjetnostjo p, ter dobimo model omrežja $N_{n,p}$. Porazdelitev stopenj vozlišč je binomska oziroma Poissonova v limiti, ko $n \to \infty$:

$$p(k) \approx \frac{\bar{z}^k e^{-\bar{z}}}{k!}, \tag{3.1}$$

kjer je $\bar{z}$ povprečna stopnja vozlišč omrežja, $\bar{z} = (n-1)p$. Od tu tudi ime Poissonov model omrežij.

Najpomembnejši lastnosti Poissonovega modela sta velikost največje komponente in obstoj točke prehoda faze (*phase transition*). Pri majhnih vrednostih p dobimo omrežje z nizko gostoto povezav, kjer imamo veliko majhnih komponent. Velikosti komponent so porazdeljene eksponentno. S povečevanjem parametra p se izoblikuje ena velika komponenta, ki vsebuje večino vseh vozlišč omrežja (reda velikosti $O(n)$). Velikosti preostalih, manjših komponent so spet porazdeljene eksponentno. Točka prehoda faze, kjer se iz veliko majhnih komponent izoblikuje ena velika komponenta, se zgodi pri $p = 1/n$ [8].

Premer omrežja Poissonovega modela je približno enak $\log(n)/\log(\bar{z})$ [30]. To pomeni, da ima Poissonov model majhen premer, kot smo ga definirali v prejšnjem razdelku, in sledi tako imenovanemu učinku majhnega sveta. Slabše se odreže pri ostalih lastnostih. Za porazdelitev stopenj smo že ugotovili, da je Poissonova in tako ne upošteva potenčnega zakona, ki je značilno za realna omrežja.



Slika 3.1: Primer omrežja generiranega s Poissonovim modelom. [39]

Slabo se odreže tudi pri izoblikovanju skupnosti. Verjetnost, da bosta dve vozlišči povezani med seboj, je vedno enaka p, ne glede na to, ali imata skupnega soseda ali ne. Posledično je koeficient razvrščanja C enak p,

$$p = \frac{\tilde{\chi}}{n}, \tag{3.2}$$

kjer je $\tilde{\chi}$ povprečna stopnja vozlišč. Iz tega očitno sledi, da je koeficient razvrščanja C v limiti $n \to \infty$ enak 0 [45].

Kljub tem pomanjkljivostim, ki so razlog, da se danes Poissonov model za generiranje realnih omrežij ne uporablja prav pogosto, je bil za razvoj drugih modelov naključnih omrežij zelo pomemben. Obstoj točke prehoda faze in ene velike komponente ter majhnega premera so lastnosti, ki so prisotne v vseh poznejših modelih naključnih omrežij. Vse to se je prvič odkrilo prav na omrežjih generiranih s Poissonovim modelom.

3.2 Modeli po načelu prednostne povezanosti

Sredi petdesetih let prejšnjega stoletja je Simon [37] ugotovil, da potenčni zakoni nastanejo z uporabo izreka »bogati bogatijo« (*the rich get richer*). Price je ob raziskovanju omrežja citatov med znanstvenimi članki [34] ugotovil, da porazdelitev tako vhodnih kot tudi izhodnih stopenj vozlišč sledi potenčnemu zakonu. Na podlagi teh idej je predstavil še danes veljavno razlago za nastanek potenčnih zakonov v porazdelitvi stopenj vozlišč [33]. Ogledamo si jo v nadaljevanju.

Omrežje gradimo postopoma, z dodajanjem vozlišč v vsakem koraku, pri čemer naj bodo povezave usmerjene. V vsakem koraku dodamo vozlišče s fiksno izhodno stopnjo, kar ponazarja število citatov, ki jih članek citira. Izhodna stopnja se med vozlišči lahko razlikuje, dokler ostaja povprečna stopnja enaka z. Vsaka izhodna povezava vozlišča se povezuje na neko obstoječe vozlišče z verjetnostjo sorazmerno vhodni stopnji obstoječega vozlišča. Tukaj naletimo na težavo, saj imajo na začetku vsa vozlišča vhodno stopnjo enako nič. Price se je temu izognil tako, da je vhodni stopnji dodal neko konstanto k_0:

$$p(\text{povezava do obstoječega vozlišča}) = \frac{I(v) + k_0}{\sum_i (I(i) + k_0)}, \tag{3.3}$$

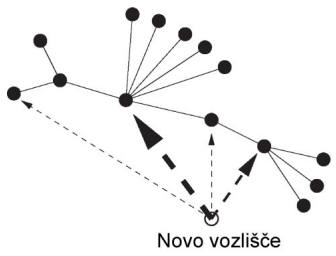
kjer smo z $I(v)$ označili trenutno vhodno stopnjo vozlišča v.

Podoben model, ki je postavil podlago za veliko poznejšega dela, sta predlagala Barabási in Albert (v nadaljevanju model BA) [3, 4]. Njun model odpravi kar nekaj pomanjkljivosti Priceovega modela. Model BA ne ločuje vhodne in izhodne stopnje vozlišča ter zato gradi samo omrežja z neusmerjenimi povezavami, kar je sicer manjša pomanjkljivost, če potrebujemo usmerjeno omrežje. Vendar se tako izogne začetnemu problemu Priceovega modela z vhodnimi stopnjami.

Začnemo z manjšim začetnim naborom (nepovezanih) vozlišč m_0. V vsakem koraku dodamo novo vozlišče, ki ga povežemo z m obstoječimi vozlišči v omrežju (m je v tem primeru fiksen). Obstoječa vozlišča se izbirajo sorazmerno glede na njihovo stopnjo:

$$p(\text{povezava do obstoječega vozlišča}) = \frac{k(v)}{\sum_i k(i)}, \tag{3.4}$$

kjer smo s $k(v)$ označili stopnjo vozlišča v. Višja kot je stopnja vozlišča, večja je verjetnost, da se bo novo vozlišče povezalo z njim (slika 3.2), kar je v skladu z izrekom »bogati bogatijo«.

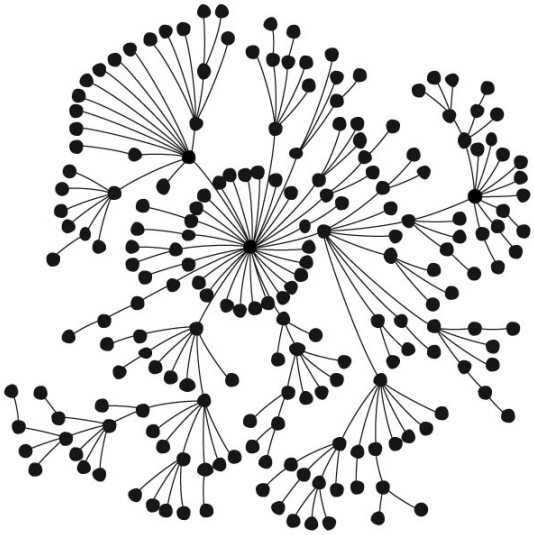


Slika 3.2: Vozlišča z višjo stopnjo imajo večjo verjetnost, da dobijo novo povezavo. [8]

Porazdelitev stopenj v modelu BA sledi potenčnemu zakonu, in sicer je Dorogovtsev et al. [14] pokazal, da

$$p(k) \approx k^{-3} \quad \textbf{(3.5)}$$

za velike vrednosti *k*. Kar pomeni, da ima BA model porazdelitev stopenj, ki v repu porazdelitve (velike vrednosti *k*) sledi potenčnemu zakonu z eksponentom *3*, neodvisno od vrednosti *m*. Vrednost eksponenta se zdi ustrezna, saj je blizu vrednosti *3.04*, ki jo je, kot smo že omenili, za omrežje citiranj izračunal Price [33].



Slika 3.3: Primer omrežja generiranega z modelom BA. [39]

Tudi premer omrežja generiranega z BA modelom se izkaže za majhnega. In sicer naj bi se premer povečeval počasneje kot $\log(n)$ za $m=1$ ter počasneje kot $\log(n)/\log\log(n)$ za $m \geq 2$ [6]. Tudi ta model sledi tako imenovanemu učinku majhnega sveta, saj je njegov premer v splošnem veliko manjši kot število vozlišč.

Zanimiva je tudi korelacija vozlišč, saj je opaziti [25], da imajo starejša vozlišča višjo povprečno stopnjo. Druga korelacija je v stopnji sosednjih vozlišč, kar pomeni, da imajo vozlišča s podobnimi stopnjami večjo verjetnost, da bodo povezana med seboj. Vendar ta verjetnost asimptotično pada proti *0*, ko gre $n \to \infty$.

Vendar ima model BA tudi nekaj pomanjkljivosti. Model glede na testiranja ne generira nobene skupnosti med vozlišči. Eksponent potenčne funkcije v porazdelitvi stopenj je fiksen, večina realnih omrežij pa ima ta eksponent različen. Prav tako je velika pomanjkljivost, da imajo vsa generirana omrežja samo eno veliko povezano komponento, medtem ko so realna omrežja po navadi sestavljena iz ene velike in več manjših komponent. Premer omrežja se s povečevanjem števila vozlišč *n* povečuje, kljub temu da se pri veliko realnih primerih ta vrednost z večanjem omrežja zmanjšuje.

Tem pomanjkljivostim navkljub je model BA danes izredno priljubljen, saj na izredno preprost način generira zelo dobra naključna omrežja. Pravzaprav večina novejših modelov naključnih omrežij temelji na zaključkih modela BA.

3.3 Geografski modeli

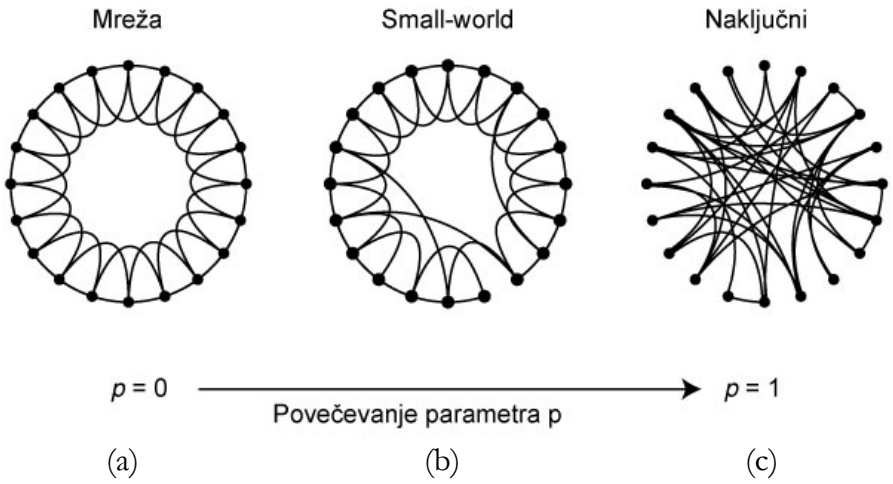
Noben od modelov, ki smo jih spoznali do sedaj, ni pri generiranju omrežja upošteval geografske lokacije vozlišč. Verjame se, da položaj vozlišč v prostoru vpliva na strukturo omrežja. V socialnih omrežjih ima osebek za prijatelje po navadi osebe, ki jih pogosto srečuje, kar pomeni, da živijo blizu skupaj, na primer v istem kraju ali mestu. To značilnost omrežij skušamo modelirati z geografskimi modeli.

Najbolj značilen predstavnik te skupine je model majhnega sveta (*The small-world model*) [43, 45], ki temelji na dveh lastnostih realnih omrežij. Prvo lastnost, da imajo realna omrežja po navadi majhne premere, smo že omenili. Drugo pa je leta 1973 odkril Granovetter [21], in sicer, da imajo omrežja kljub majhnemu premeru velik koeficient razvrščanja. Razlog za to vidi v tem, da v omrežju obstajajo zaprte skupine (klike), ki imajo šibke povezave z drugimi skupinami. Povezave v skupinah so razlog za velik koeficient razvrščanja, povezave med različnimi skupinami pa poskrbijo za majhen premer. Kot primer je navedel rezultate analize, kako posamezniki iščejo zaposlitev. V nasprotju s pričakovanji, da bo večina iskalcev zaposlitve le-to našla po dolgi poti, se je izkazalo da jo je večina našla po zelo kratki poti, dolžini samo ena ali dva skoka. Intervjuvanci so tudi dejali, da jim za zaposlitev ni povedal prijatelj, ampak znanec, kar kaže na šibko povezavo med osebkoma in se sklada s teoretično definicijo.

Model majhnega sveta deluje po principu, da začnemo gradnjo omrežja z mrežo, v kateri nato premikamo ali dodajamo povezave, da ustvarimo tako imenovane bližnjice med skupinami. Najpogosteje se uporablja mreža na krožnici, kjer je vsako od n vozlišč na krožnici povezano s s najbližjimi vozlišči, da dobimo

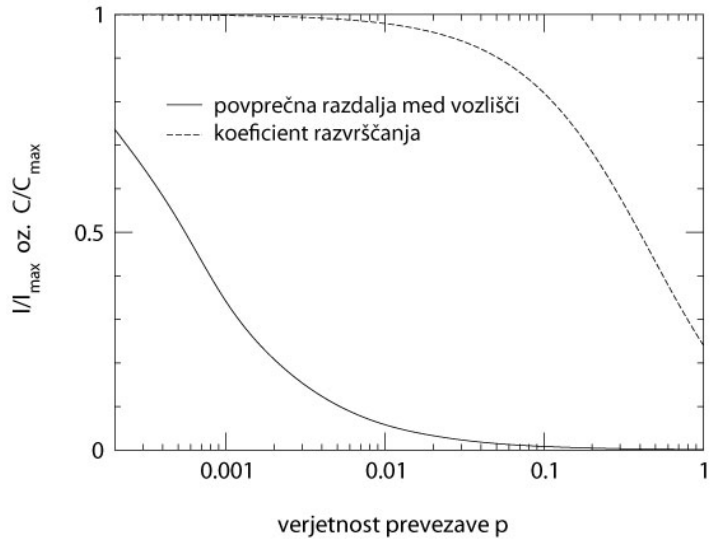
$$e_s = \frac{ns}{2} \tag{3.6}$$

povezav, kot to prikazuje slika 3.4a. Da se ustvari tako imenovani učinek majhnega sveta, se nato vsaki povezavi iz omrežja z neko verjetnostjo p zamenja eno od vozlišč ter s tem ustvari bližnjico skozi omrežje (slika 3.4b).



Slika 3.4: Slika prikazuje, kako se spreminja oblika omrežja s povečevanjem parametra p. [45]

S parametrom p lahko nadziramo obliko omrežja. Pri p=0 (slika 3.4a) imamo navadno mrežo z visokim koeficientom razvrščanja $C \approx (3k-3)/(4k-2)$, vendar pa ostane premer omrežja zelo visok, $l \approx n/2s$ [45]. Pri p=1 se vsaka povezava preveže na drugo vozlišče in dobimo omrežje podobno naključnemu modelu (slika 3.4c). Koeficient razvrščanja je v tem primeru zelo nizek $C \approx s/n$, prav tako premer $l \approx \log(n)/\log(s)$ [45]. Vendar pa med tema ekstremoma obstaja območje za p, kjer ima omrežje majhen premer ob visokem koeficientu razvrščanja [8, 30, 45].



Slika 3.5: Prikaz območja p, kjer ima omrežje majhen premer in visok koeficient razvrščanja. [30]

Model majhnega sveta je tako zelo uspešen pri združevanju dveh pomembnih lastnosti realnih omrežij, in sicer majhnega premera in velikega koeficienta razvrščanja. Kljub temu pa

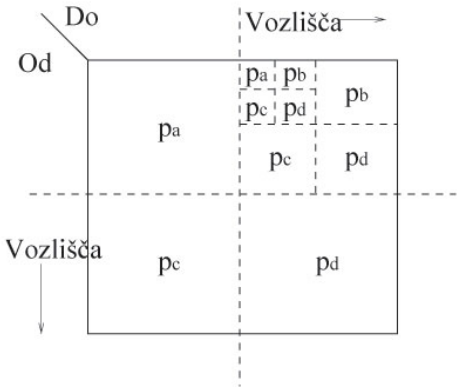
mu ne uspe ustvariti porazdelitve stopenj, kot jo poznamo iz realnih omrežij. Vsa vozlišča imajo na začetku stopnjo k, ki se spremeni samo v primeru prevezave povezav. Porazdelitev je podobna porazdelitvi stopenj naključnega modela in ima vrh v k, ki nato pada eksponentno.

Želeli bi si razširitev modela, ki bi poleg majhnega premera in velikega koeficienta razvrščanja upoštevala tudi potenčno porazdelitev stopenj. S takšnim modelom bi nato lahko generirali naključna omrežja zelo podobna realnim. Vendar zaenkrat postopka, s katerim bi nam to v celoti uspelo, še ne poznamo.

3.4 Model R-MAT

Vidimo, da se do sedaj opisani modeli osredotočajo na eno od lastnosti omrežij, ki jo modelirajo dobro, medtem ko obenem zapostavijo ostale lastnosti omrežij. Model BA generira omrežja s potenčno porazdelitvijo stopenj, ne uspe pa mu ustvariti nobenih skupnosti vozlišč. Po drugi strani model majhnega sveta ustvari omrežje z zelo velikim koeficientom razvrščanja (kar kaže na skupnosti), vendar so stopnje vozlišč porazdeljene eksponentno, kar ni v skladu z realnimi omrežji.

Želeli bi si model naključnih omrežij, ki bi poleg potenčne porazdelitve stopenj generiral tudi skupnosti znotraj omrežja. Obetajoč se zdi model R-MAT (*Recursive MATrix*) [9], ki skuša po trditvah avtorjev zajeti obe lastnosti in v omrežju ustvariti tako potenčno porazdelitev stopenj kot tudi skupnosti.



Slika 3.6: Model R-MAT: Rekurzivno delimo matriko sosednosti na štiri bloke [9].

Model R-MAT generira usmerjena omrežja z 2^n vozlišči in e povezavami. Začnemo s prazno matriko sosednosti, ki jo razdelimo na štiri bloke enakih velikosti. Z nekimi verjetnostmi p_a-p_b-p_c-p_d, kjer velja $p_a+p_b+p_c+p_d=1$, izberemo enega od blokov, kot kaže slika 3.6. Ta blok nato rekurzivno delimo naprej, dokler nam znotraj bloka ne ostane samo en element, ki predstavlja

eno povezavo. Vozlišči, ki sovpadata s tem elementom, povežemo s povezavo in to označimo v matriki sosednosti. To nato ponovimo *e-krat*, da dobimo v omrežju *e* povezav.

Za neusmerjeno omrežje velja, da mora biti matrika sosednosti simetrična. To dosežemo tako, da generiramo usmerjeno omrežje, kjer je $p_b=p_c$, da dobimo na vsaki strani diagonale približno polovico povezav. Nato zgornjo polovico matrike nad diagonalo zavržemo in čeznjo skopiramo spodnjo polovico matrike. Tako dobimo simetrično matriko in neusmerjeno omrežje.

Avtorji modela navajajo, da porazdelitev stopenj sledi potenčnemu zakonu, prav tako pa je mogoče s spremembo parametrov p_a-p_b-p_c-p_d doseči drugačno porazdelitev za omrežja, katerih stopnje niso porazdeljene po potenčnem zakonu. Poleg tega naj bi ta pristop generiral tudi skupnosti. Bloka z verjetnostmi p_a in p_d namreč predstavljata dve ločeni skupnosti znotraj omrežja, bloka p_b in p_c pa predstavljata povezave med temi skupnostmi. Rekurzivna gradnja omrežja hkrati pomeni, da bomo generirali tudi skupnosti znotraj skupnosti.

4 Strojno učenje iz naključnih omrežij

V prejšnjih dveh razdelkih smo podrobneje predstavili teoretično ozadje omrežij in modelov naključnih omrežij. Le-to uporabimo v nadaljevanju, kjer si pogledamo, kako si pri generiranju naključnih omrežij pomagamo z metodami strojnega učenja.

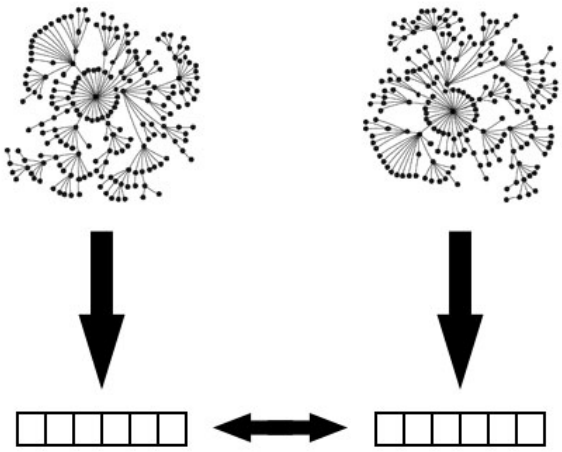
Generiranje naključnih omrežij, ki bi do podrobnosti posnemala realna omrežja, je težak zalogaj. V literaturi obstaja mnogo različnih modelov naključnih omrežij, ki pa so povečini ozko usmerjeni in dobro generirajo samo specifična omrežja. Splošnih modelov, ki bi bila uspešna pri vrsti različnih omrežij, pravzaprav ni.

Prav zaradi obstoja ogromne množice različnih modelov, je v praksi težko izbrati enega, ki bo v danem primeru najbolj ustrezen. V razdelku 4.2 zato predstavimo pristop, ki ga poimenujemo M-generator (Meta generator), s pomočjo katerega olajšamo izbiro najprimernejšega modela in pa samo generiranje naključnih omrežij. Pred tem si v razdelku 4.1 ogledamo še omejitve in težave na katere naletimo pri strojnem učenju nad omrežji.

4.1 Omejitve pri strojnem učenju nad omrežji

Znanje o omrežjih je danes relativno omejeno, s tem pa je omejeno tudi njihovo proučevanje. Strukturo in lastnosti realnih omrežij sicer poznamo, ne vemo pa, kako takšna omrežja nastanejo. Pri tem nas zanima predvsem, kateri dejavniki vplivajo oziroma so pomembni pri nastanku omrežij. Prav tako je tudi primerjanje samih omrežij med seboj še nerešen problem. Natančneje, ne poznamo nobene mere, s katero bi lahko enolično označili omrežja in jih tako primerjali med seboj.

Ker so velika omrežja kot celota neobvladljiva, jih navadno preslikamo v n-dimenzionalni vektor (slika 4.1). Z vektorjem omrežja skušamo zajeti vse oziroma čim več pomembnih lastnosti omrežja. Ključnega pomena je tako izbira ustreznih (numeričnih) parametrov, ki opisujejo omrežje. V predlaganem pristopu za parametre izberemo lastnosti omrežij, ki smo jih opisali v razdelku 2.2.



Slika 4.1: Preslikava omrežja v vektor.

Pri primerjanju dveh omrežij se tako po navadi zatekamo zgolj k primerjanju njihovih lastnosti (izbranih parametrov). Sicer pa poznamo še druge metode primerjanja omrežij, kot so *edit distance* in največje skupno podomrežje (*maximum common subgraph*) [7, 46]. Vendar pa so takšni pristopi za naš problem časovno prepotratni, saj so v večini namenjeni omrežjem manjših velikosti.

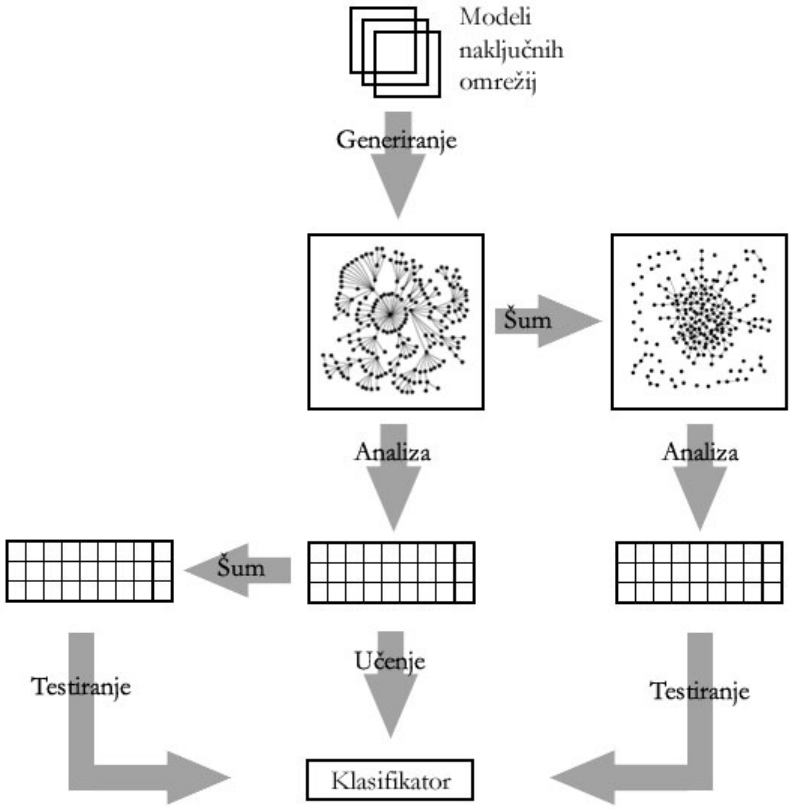
Zanimiva se zdijo tudi tako imenovana jedra omrežij (*graph kernels*), ki so deležna vse večjega zanimanja. Z jedri navadno primerjamo strukturne lastnosti (ali strukturo) dveh omrežij. Obetajoča so predvsem jedra, ki temeljijo na naključnih sprehodih čez omrežje in najkrajših poteh skozi omrežje [18, 22]. Vendar se tudi tu izkaže primerjanje takšnih poti prekompleksno za velika omrežja. Za primerjanje teh se predlagajo učinkovitejši pristopi, kot je preštevanje manjših skupnih podomrežij dveh omrežij [7, 36]. S pomočjo jeder bi lahko primerjali realna in naključno generirana omrežja ter tako označili začetno učno množico omrežij.

Vendar pa se pri predlaganem pristopu za težavno izkaže tudi pomanjkanje začetnega nabora podatkov. Ker želimo z M-generatorjem omogočiti generiranje čim bolj raznolikih omrežij, bi potrebovali tudi raznoliko začetno množico omrežij iz katere bi zgradili potreben klasifikator. Slednje pa je v splošnem praktično nemogoče doseči. Obstaja namreč preveč različnih vrst omrežij, ki jih je težko odkriti. Prav tako se vedno znova in znova odkrivajo nova omrežja, ki

jih s takšnim pristopom ne bi pokrili. Problematično pa bi bilo seveda tudi zbiranje podatkov za tako veliko število omrežij.

4.2 Model M-generator

Zaradi omenjenih težav se generiranja omrežij lotimo na nekoliko drugačen način. Namesto, da bi klasifikator učili na podatkih realnih omrežij, ga učimo na lastnostih naključno generiranih omrežij. Predpostavimo, da s tem zajamemo lastnosti omrežij, ki so značilne za posamezne modele naključnih omrežij in tako pravilno klasificiramo omrežja, ki jih želimo generirati. V našem pristopu prav tako predpostavimo, da je najprimernejši model za generiranje naključnih omrežij tisti, ki generira omrežja najbolj podobna realnemu omrežju v danem primeru.



Slika 4.2: Prikaz gradnje in testiranja klasifikatorja modela M-generator.

Pri generiranju omrežij z M-generatorjem v prvi fazi izberemo modele naključnih omrežij. Izbira modelov močno vpliva na končna omrežja, ki jih generiramo z M-generatorjem, vendar pa le-ta na tem mestu ponuja (ključno) prednost pred ostalimi pristopi. Pri izbiri modelov naključnih omrežij namreč ni potrebno biti selektiven. Zagotoviti moramo zgolj, da v množico razpoložljivih modelov vključimo čim več potencialno primernih.

Z izbranimi modeli nato generiramo različna naključna omrežja. Generiramo čim večjo množico različnih omrežij in s tem zagotovimo reprezentativnost vzorca. Pri tem modelu naključnih omrežij podamo zgolj omejene informacije o lastnostih omrežja, ki ga želimo generirati. Natančneje, velikost omrežja (število vozlišč) in gostoto omrežja (povprečno stopnjo vozlišča). Dodatnih parametrov modelom ne nastavljamo, saj bi sicer naleteli na težavo pri konstruiranju klasifikatorja – omrežij generiranih z istim modelom, a različnimi parametri, pri klasifikaciji ne bi razlikovali. Ker pa nastavitev parametrov v nekaterih primerih vseeno potrebujemo, predlagamo, da se pri teh vsako vrednost parametra obravnava kot samostojen model.

Kot smo omenili že v prejšnjem razdelku, so velika omrežja za nas neobvladljiva v celoti, omrežja zato predstavimo zgolj z njihovimi lastnostmi. S tem gotovo izgubimo določeno količino informacije o omrežju, stremimo pa k temu, da bi bila ta izguba čim manjša. V modelu M-generator uporabimo lastnosti predstavljene v razdelku 2.2.

Omrežja generirana z izbranimi modeli sedaj analiziramo in poiščemo izbrane lastnosti (preslikamo v vektor). Tako dobimo lepo strukturirane podatke (atributni zapis), nad katerimi lahko uporabimo eno od standardnih metod strojnega učenja. Za razred vsakega omrežja izberemo model, s katerim je bil generiran in na ta način dobimo označeno učno množico omrežij.

Omenimo, da analiza omrežja ne sme biti časovno potratna, saj želimo z M-generatorjem analizirati veliko število naključnih omrežij. V nasprotnem primeru bi nam počasna analiza onemogočila generiranje (analizo) velikega vzorca omrežij in s tem zmanjšala njegovo reprezentativnost.

V naslednji fazi gradnje M-generatorja izberemo ustrezen klasifikator. Izkaže se, da zaradi specifičnosti omrežij posameznega generatorja večina klasifikatorjev naključna omrežja lepo razloči med seboj. Pri testiranju in izbiri klasifikatorjev zato podatkom dodamo šum in tako bolj temeljito pretestiramo njihovo delovanje.

Poleg učne množice omrežij tako generiramo še novo, testno množico omrežij, ki jim pred analizo dodamo šum. Tu se tako pojavi vprašanje na kakšen način naj dodamo šum. Če v splošnem ne znamo generirati realnih omrežij, bo podobno verjetno tudi pri dodajanju šuma. Izkaže se, da je dodajanje šuma v omrežje podobno kompleksen problem kot samo generiranje omrežja. Zato se odločimo, da šum dodamo na naslednji način.

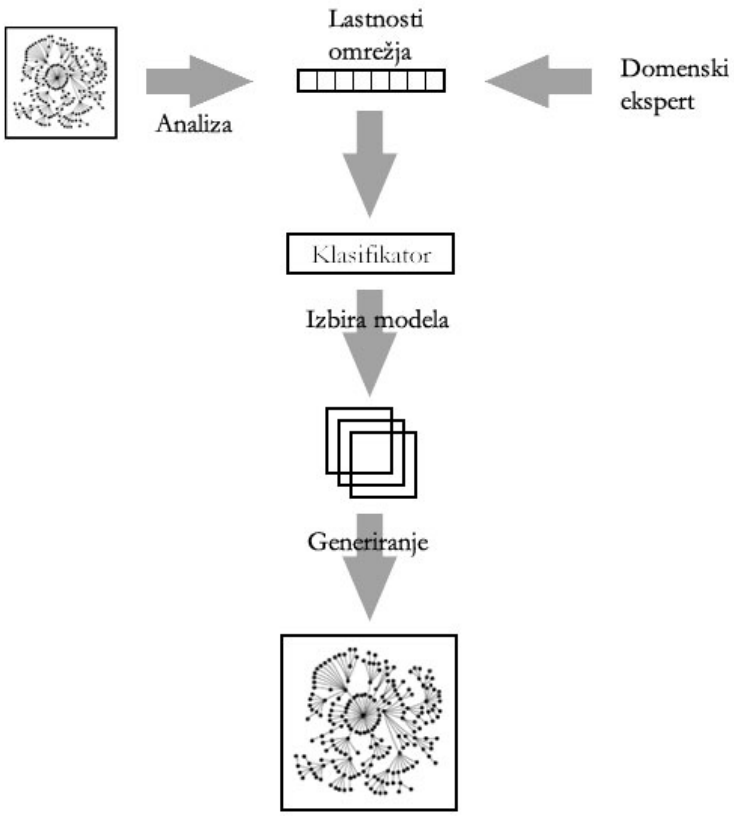
Najprej v omrežju naključno izberemo nek delež povezav. Tem povezavam nato odstranimo obe končni vozlišči (da povezava *visi* v zraku) in jih prevežemo na neka druga, naključno izbrana vozlišča. S tem sicer dodamo šum v omrežje, vendar le-ta ne posnema realnih lastnosti omrežij. Dodani šum se namreč obnaša podobno kot naključni model omrežij. Kar pomeni, da z dodajanjem šuma zmanjšujemo skupnosti v omrežju, porazdelitev stopenj oddaljujemo od potenčne porazdelitve ipd. Ko prevežemo vse povezave v omrežju (100-

odstotni šum), dobimo iz kateregakoli začetnega omrežja končno omrežje, ki je podobno omrežjem generiranim z naključnim modelom omrežja.

S tako »zašumljenimi« omrežji sicer ni nič narobe, vendar pa vseeno klasifikatorje vzporedno testiramo še na podatkih, ki jim dodamo šum na drugačen način. Najprej analiziramo osnovna omrežja (brez šuma), šum pa dodamo šele vektorjem, ki predstavljajo lastnosti omrežja (Gaussova porazdelitev).

Klasifikatorje sedaj testiramo in primerjamo nad obema testnima množicama. V našem pristopu smo za M-generator primerjali več različnih klasifikatorjev na različnih množicah atributov. Rezultate podajamo v razdelku 5.2. Na podlagi testiranj na koncu izberemo najbolj ustrezen klasifikator, ki ga uporabimo v modelu.

S tem smo zaključili gradnjo modela M-generator. Slika 4.3 prikazuje avtomatsko klasifikacijo novih omrežij in generiranje naključnih omrežij.



Slika 4.3: Prikaz generiranja omrežja z modelom M-generator.

Kadar imamo pri generiranju omrežja z M-generatorjem na voljo realno omrežje, le-to najprej analiziramo. Analiza poteka na enak način kot pri analizi naključnih omrežij. Na koncu analize dobimo vektor z lastnostmi, ki opisujejo realno omrežje. Na podlagi teh lastnosti nato

klasifikator izbere najbolj primeren model naključnih omrežij, M-generator pa ga uporabi za samodejno generiranje novega naključnega omrežja.

V primeru, da neko realno omrežje ni na voljo, lahko vektor z lastnostmi nastavimo tudi ročno. Seveda pa tu potrebujemo pomoč domenskega eksperta, ki ustrezno nastavi parametre omrežja.

5 Eksperimentalni rezultati in diskusija

V pričujočem razdelku predstavimo eksperimentalne rezultate, dobljene z modelom M-generator. Pri tem pa najprej uporabimo zgolj osnovne tri modele naključnih omrežij, saj se zdi, da le-ti najbolje predstavijo glavne skupine naključnih omrežij (razdelek 3). Tako v razdelku 5.1 sprva analiziramo naključna omrežja, ki jih generiramo z osnovnimi tremi modeli naključnih omrežij. V naslednjem razdelku nato predstavimo analizo klasifikatorjev, ki jih uporabimo pri izbiri modela naključnega omrežja. V razdelku 5.3 pa predstavimo še rezultate testiranja M-generatorja.

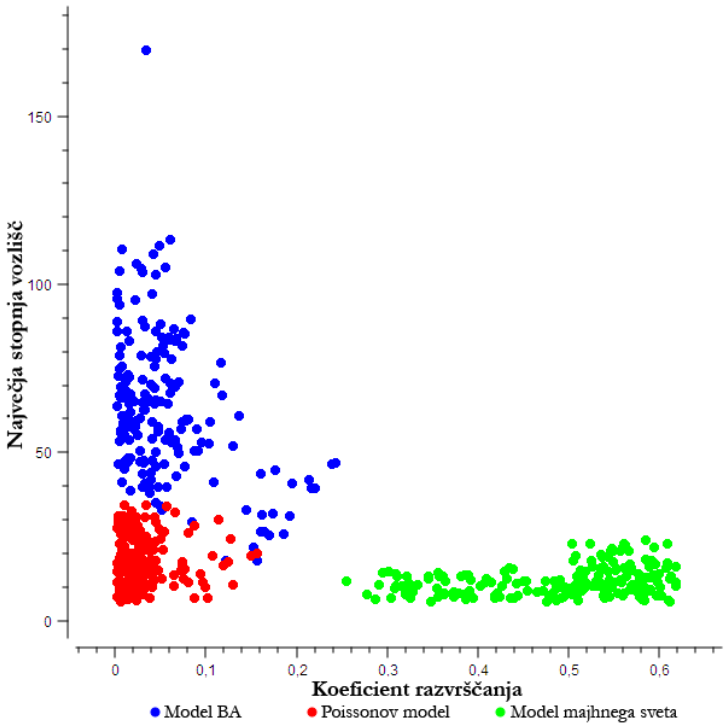
Na koncu v razdelku 5.4 analiziramo tudi model R-MAT. Z njim skušamo razširiti in izboljšati delovanje M-generatorja ter analizirati delovanje klasifikatorja M-generatorja.

5.1 Analiza modelov naključnih omrežij

Za začetek si izberemo tri osnovne modele naključnih omrežij - Poissonov model, model BA in model majhnega sveta (razdelek 3). Z vsakim modelom generiramo več naključnih omrežij velikosti od 100 do 1200 vozlišč z gostotami od 2 do 8 povezav na vozlišče. Različnih parametrov modelom ne nastavljamo, ampak uporabimo tiste, ki se eksperimentalno izkažejo za optimalne. S temi naključnimi omrežji pokrijemo prostor realnih omrežij, ki jih imamo na razpolago za testiranje (opišemo jih v razdelku 5.3).

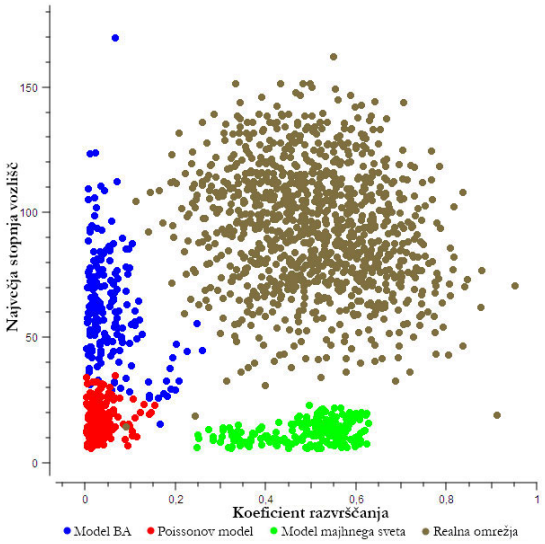
Dobljena naključna omrežja najprej analiziramo. Izkaže se, da omrežja sledijo teoretičnim zakonitostim modelov naključnih omrežij (razdelek 3). Tako imajo vsa omrežja generirana s Poissonovim modelom majhen premer, a tudi majhen koeficient razvrščanja, porazdelitev stopenj pa ne kaže potenčne porazdelitve. Omrežja, generirana z modelom BA, očitno sledijo potenčni porazdelitvi, imajo izrazito veliko največjo stopnjo vozlišča, vendar pa imajo še

vedno majhen koeficient razvrščanja. Največji koeficient razvrščanja kažejo omrežja generirana z modelom majhnega sveta, ki se že samo po tej lastnosti lepo loči od ostalih dveh modelov. Slika 5.1 prikazuje projekcijo omrežij glede na omenjene lastnosti. Koeficient razvrščanja kaže na skupnosti v omrežju, največja stopnja vozlišč pa na potenčno porazdelitev stopenj.



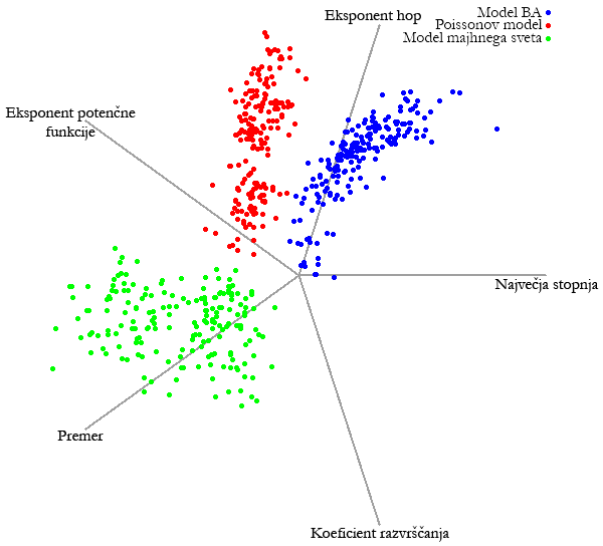
Slika 5.1: Porazdelitev omrežij generiranih z modeli naključnih omrežij na grafu.

Če sedaj naključno generirana omrežja primerjamo še z realnimi omrežji, vidimo (slika 5.2), da veliko realnih omrežij stoji na območju velikih največjih stopenj in visokih koeficientov razvrščanja. Kot smo že predhodno spoznali, je to prav območje, ki ga želimo pokriti z modeli naključnih omrežij, vendar to v celoti ne uspe še nobenemu od modelov naključnih omrežij.



Slika 5.2: Poleg naključnih omrežij sedaj na grafu prikažemo še območje realnih omrežij[3] (rjave barve).

Odkrili pa smo še več. Izkaže se, da lahko z uporabo več lastnosti omrežja le-ta lepo razdelimo v disjunktne množice. Iz slike 5.3 vidimo, da se z uporabo največje stopnje vozlišč, koeficienta razvrščanja, premera, eksponenta potenčne funkcije in eksponenta hop vsi trije modeli lepo ločijo med seboj. To spoznanje izkoristimo tudi pri gradnji klasifikatorja. Na sliki (desno spodaj) opazimo praznino, ki kaže na manko modelov naključnih omrežij z visokim koeficientom razvrščanja in veliko največjo stopnjo vozlišč.



Slika 5.3: Prikaz porazdelitve naključnih omrežij na grafu s petimi osmi. Slika je bila generirana z orodjem VizRank iz paketa Orange [13].

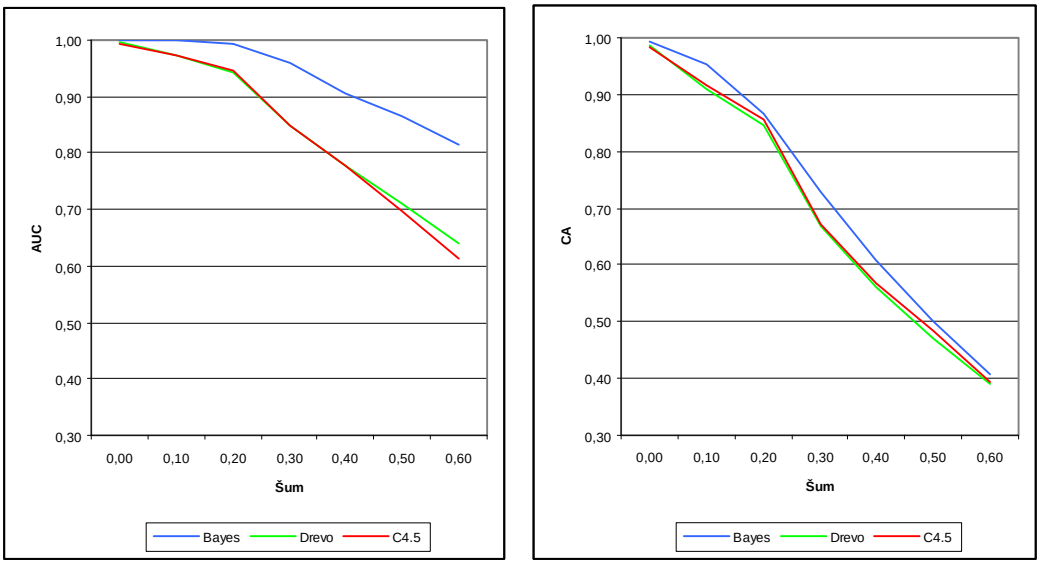
[3] Zaradi pomanjkanja realnih omrežij smo za boljšo predstavo na sliko dodali dodatne točke, ki predstavljajo le-ta, vendar pa ne temeljijo na analizi realnih omrežij.

5.2 Analiza klasifikatorjev

Zaradi uporabe običajnih klasifikatorjev strojnega učenja, ki zahtevajo podatke v atributnem zapisu, podatke pretvorimo v ustrezno obliko. Uporabimo naslednje atribute: število vozlišč, število povezav, eksponent potenčne funkcije (α), napaka potenčne funkcije (R^2), največja stopnja vozlišč, povprečna stopnja vozlišč, premer, eksponent hop in koeficient razvrščanja. Množico atributov med testiranjem skrčimo še na množico atributov, ki jo vidimo na sliki 5.3, ter rezultate primerjamo s polno množico atributov. Za razred vsakega omrežja določimo model, ki je to omrežje generiral.

Pri analizi uporabimo tri klasifikacijske algoritme strojnega učenja - naivni Bayesov algoritem, »navadna« odločitvena drevesa [13] in odločitvena drevesa C4.5. Vsak algoritem poženemo na polni množici atributov ter na skrčeni množici. Uspešnost algoritmov ocenjujemo s klasifikacijsko točnostjo (*classification accuracy* – *CA*) in območjem pod krivuljo ROC (*area under curve* – *AUC*).

Na sliki 5.4a je prikazana vrednost AUC v odvisnosti od stopnje šuma, ki ga dodamo v omrežje pred analizo. Šum dodajamo v korakih po 10 %, kjer prevežemo določen delež povezav, kot smo to opisali v razdelku 3.4. Opazimo, da je najbolj robusten naivni Bayesov klasifikator (modra barva), medtem ko uspešnost »navadnega« drevesa (zelena barva) in drevesa C4.5 (rdeča barva) pri dodajanju šuma pada izrazito hitreje. Podobno je pri opazovanju CA (slika 5.4b), vendar pa so tukaj razlike med posameznimi klasifikatorji manj opazne. Opazimo tudi, da krivulja vrednosti CA pada prav premo sorazmerno s povečevanjem šuma. Očitno pri naključnem prevezovanju določenega deleža povezav nato klasifikatorji napačno klasificirajo približno enak delež vseh omrežij.

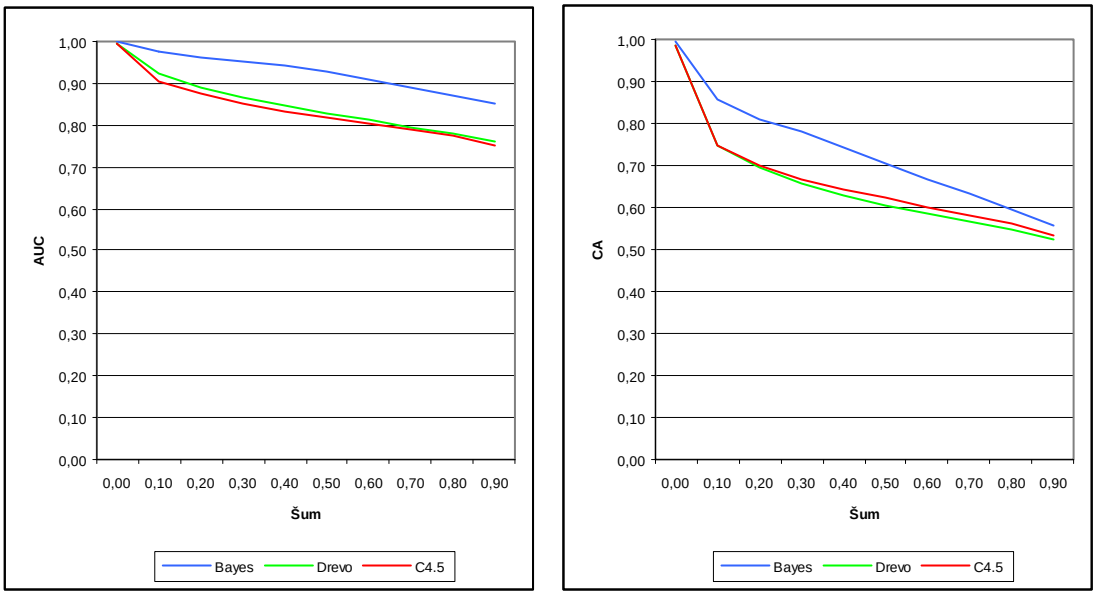


(a) (b)

Slika 5.4: Vrednosti AUC (a) in CA (b) pri različnih stopnjah šuma.

Oglejmo si še razlike med klasifikatorji, ko šum dodamo vektorjem lastnosti (slika 5.5). Na tem mestu opozorimo, da se dodajanje šuma med omenjenima pristopoma bistveno razlikuje in ju ne gre enačiti. V prejšnjem primeru delež šuma označuje delež prevezanih vozlišč, medtem ko v tem primeru delež šuma označuje delež velikosti standardne deviacije atributa. Šum tako v tem primeru dodamo vsakemu atributu, neodvisno med seboj (Gaussov šum s povprečno vrednostjo *0* in izbrano standardno deviacijo).

Tudi v tem primeru se najbolje odreže naivni Bayesov klasifikator, medtem ko ostala dva klasifikatorja zaostajata. Nakloni krivulj (vpliv šuma) se zaradi omenjenih razlik močno razlikujejo od prejšnjega primera (pazi na različni skali na slikah 5.4 in 5.5).



(a) (b)

Slika 5.5: Vrednosti AUC (a) in CA (b) pri šumu dodanem vektorjem lastnosti.

Klasifikatorje smo testirali tako na polni množici atributov (rezultati prikazani na slikah 5.4 in 5.5), kot tudi na skrčeni množici petih atributov iz slike 5.3. Naivni Bayesov klasifikator se je vedno odrezal bolje na polni množici atributov, medtem ko pri ostalih dveh klasifikatorjih razlika ni bila vedno tako očitna. Zaradi preglednosti podamo zgolj rezultate pri šumu velikosti 30 % (tabela 5.1).

Tabela 5.1: Primerjava polne in skrčene množice atributov pri 30-odstotnem šumu (CA / AUC).

Množica atributov	Naivni Bayes	Odločitveno drevo	Algoritem C4.5
Polna množica atr.	0.730 / 0.960	0.667 / 0.850	0.673 / 0.849
Skrčena množica atr.	0.676 / 0.948	0.612 / 0.706	0.630 / 0.820

V prejšnjem razdelku smo videli, da se naključna omrežja na skrčeni množici atributov lepo razdelijo v povsem disjunktne skupine (slika 5.3), vendar pa to ne velja več, ko omrežjem

dodamo šum. Verjetno se zato klasifikatorji povečini odrežejo bolje, ko imajo na voljo vse atribute omrežja.

Lahko zaključimo, da z naivnim Bayesovim klasifikatorjem dosežemo najboljše rezultate, dobro pa se odreže tudi pri klasifikaciji omrežij, ki smo jim dodali visok delež šuma. Tako za klasifikator M-generatorja izberemo naivni Bayesov klasifikator, ki ga konstruiramo z uporabo polne množice atributov. Zaradi preprostosti uporabimo kar klasifikator s privzetimi nastavitvami [13]. V nadaljevanju si pogledamo, kako se M-generator obnese pri generiranju naključnih omrežij, ki naj bi v svojih lastnostih čim bolj posnemala realna omrežja.

5.3 Testiranje modela M-generator

V nadaljevanju si najprej ogledamo nekaj realnih omrežij, nato pa skušamo z M-generatorjem generirati naključna omrežja, ki bi ta realna omrežja čim bolje posnemala. Pri analizi uporabimo naslednja omrežja:

- **Omrežje 500-tih največjih komercialnih letališč v ZDA** [11]
  Omrežje vsebuje 500 vozlišč, ki predstavljajo letališča v ZDA. Povezava med dvema vozliščema pomeni, da je bil v letu 2002 med tema letališčema opravljen vsaj en polet. Za omrežje pričakujemo potenčno porazdelitev stopenj in majhen premer. Pričakuje se tudi skupnosti znotraj omrežja in vpliv geografske lokacije posameznih vozlišč.

- **Omrežje soavtorstva člankov** [31]
  Vozlišča predstavljajo avtorje člankov, povezave pa predstavljajo sodelovanje dveh avtorjev pri skupnem članku. Omrežje vsebuje 16,264 avtorjev in 47,594 povezav.

- **Omrežje ženskega kluba** [12]
  Omrežje prikazuje 18 žensk, ki so se udeleževale srečanj znotraj nekega kluba. Dve ženski sta med seboj povezani, če sta se udeležili vsaj enega skupnega srečanja.

- **Električno omrežje zahodnega dela ZDA** [45]
  Omrežje predstavlja topologijo električnega omrežja. Vsebuje 6,594 povezav med 4,941 vozlišči.

- **Socialno omrežje delfinov** [27]
  Zelo zanimivo omrežje predstavlja pogoste interakcije med 62 opazovanimi delfini.

- **Nevronsko omrežje** [45]
  Gre za nevronsko omrežje črva *Caenorhabditis elegans*, kjer povezava povezuje dva nevrona, če med njima obstaja sinapsa (oziroma povezava).

Naivni Bayesov klasifikator nam za navedena omrežja izbere modele naključnih omrežij, ki jih vidimo v tabeli 5.2. Opazimo, da klasifikator v vseh primerih izbere model BA z veliko večjo verjetnostjo, kot pri ostalih dveh modelih. To kaže na to, da so naključna omrežja, generirana z modelom BA (oziroma njihove opazovane lastnosti), bolj podobna ustreznim realnim omrežjem, kot omrežja, generirana z ostalima modeloma. Sicer izbrani modeli ustrezajo naši teoretični osnovi in pričakovanjem, vendar je za večje zaupanje v rezultate potrebna še nadaljnja analiza.

Tabela 5.2: Modeli, ki jih za modeliranje realnih omrežij izbere algoritem naivni Bayesov klasifikator.

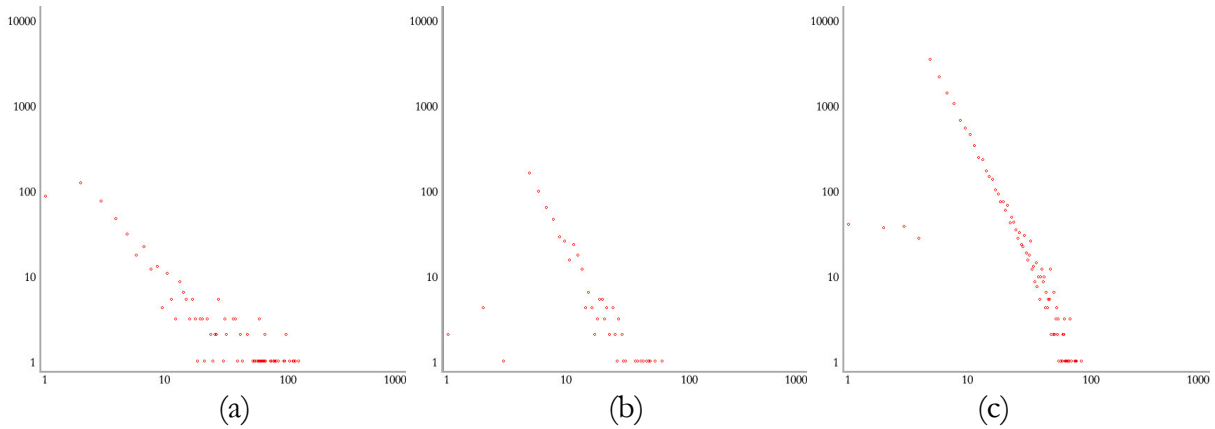
Analizirano omrežje	**Izbran model**	**Verjetnosti**
Omrežje 500-tih največjih komercialnih letališč v ZDA	Model BA	0.943
Omrežje sodelovanja med znanstveniki	Model BA	0.951
Omrežje ženskega kluba	Poissonov model	0.741
Električno omrežje zahodnega dela ZDA	Model majhnega sveta	0.518
Socialno omrežje delfinov	Model majhnega sveta	0.629
Nevronsko omrežje	Model BA	0.995

Podrobnejše si oglejmo po eno omrežje za vsak izbran generator. Klasifikator za omrežje letališč izbere model BA. S slednjim generiramo dve naključni omrežji, enega enake velikosti kot originalno omrežje (500 vozlišč), drugega pa večjega (10,000 vozlišč). Vidimo, da je večina lastnosti med omrežji primerljivih. Model se slabše odreže pri generiranju skupnosti v omrežju, saj je koeficient razvrščanja v obeh primerih veliko manjši kot pri originalnem omrežju. Tudi največja stopnja vozlišč je pri obeh naključno generiranih omrežjih precej nižja kot pri originalnem. Na sliki porazdelitev stopenj (slika 5.6) se to kaže tako, da graf naključnih omrežij ni toliko razpotegnjen v desno, kot pri realnem primeru.

Tabela 5.3: Primerjava lastnosti omrežja letališč in naključno generiranih omrežij.

	Vozlišča	**Povezave**	α	**Napaka** R^2	**Največja stopnja**	**Povpr. stopnja**	**Premer**	**Eksp. hop**	**Koeficient razvrščanja**
Omrežje letališč:	500	2,980	-0.76	0.6581	145	11	4	2.610	0.6175
Model BA:	500	2,375	-0.70	0.3014	68	9	3	3.307	0.04987
Model BA:	10,000	47,500	-1.71	0.6771	112	9	5	4.455	0.002675

Iz slik porazdelitev stopenj (slika 5.6) vidimo tudi, da originalno omrežje sledi potenčnemu zakonu (premica na »log-log« grafu). Podobna ugotovitev sledi iz slik naključno generiranih omrežij, kjer se pri manjšem omrežju porazdelitev še ni povsem ustalila, medtem ko je pri večjem potenčna funkcija že zelo lepo vidna.



Slika 5.6: Porazdelitev stopenj omrežja letališč (a), naključno omrežje modela BA s 500 vozlišči (b) in naključno omrežje z 10,000 vozlišči (c). Pri prikazu smo uporabili »log-log« graf.

Poissonov model klasifikator izbere zgolj v enem primeru, in sicer za omrežje ženskega kluba. Rezultate analize omrežja ženskega kluba in naključno generiranih omrežij s Poissonovim modelom podamo v tabeli 5.4. Kot vidimo je originalno omrežje zelo gosto, vsebuje zgolj 18 vozlišč in kar 139 povezav. Zato v tem primeru za večje naključno omrežje izberemo omrežje s 300 vozlišči, saj že tu število povezav preseže 40,000. Vidimo, da so lastnosti vseh omrežij zelo podobne. Zaradi izredno gostega omrežja je koeficient razvrščanja pri vseh zelo visok, prav tako pa tudi eksponent hop (graf »hop-plot« je praktično navpična premica). Večje omrežje generirano s Poissonovim modelom pa ima izrazito večjo največjo in povprečno stopnjo, kar je za tako gosto omrežje normalno. Slik porazdelitev stopenj tukaj ne podamo, saj nobeno od omrežij ne sledi potenčnemu zakonu.

Tabela 5.4: Primerjava lastnosti omrežja ženskega kluba in naključno generiranih omrežij.

	Vozlišča	**Povezave**	α	**Napaka** R^2	**Največja stopnja**	**Povpr. stopnja**	**Premer**	**Eksp. hop**	**Koeficient razvrščanja**
Omr. žen. kluba:	18	139	0.34	0.2532	17	15	1	10^300	0.9367
Poissonov m.:	18	141	0.34	0.2514	17	15	1	10^300	0.9208
Poissonov m.:	300	40,735	0.18	0.1796	282	271	1	10^300	0.9083

Za konec si oglejmo še omrežje delfinov, za katerega klasifikator izbere model majhnega sveta. V tabeli 5.5 opazimo, da imajo vsa omrežja visok koeficient razvrščanja, kar je tudi značilno za model majhnega sveta. Porazdelitev stopenj pri nobenem od naključnih omrežij ne sledi potenčnemu zakonu (koeficient α ni negativen), pri originalnem omrežju pa je število vozlišč premajhno, da bi lahko z gotovostjo potrdili obstoj le-tega. Sicer pa so lastnosti naključno generiranih omrežij primerljive z originalnim, tako da smo tudi v tem primeru z izbiro modela zadovoljni.

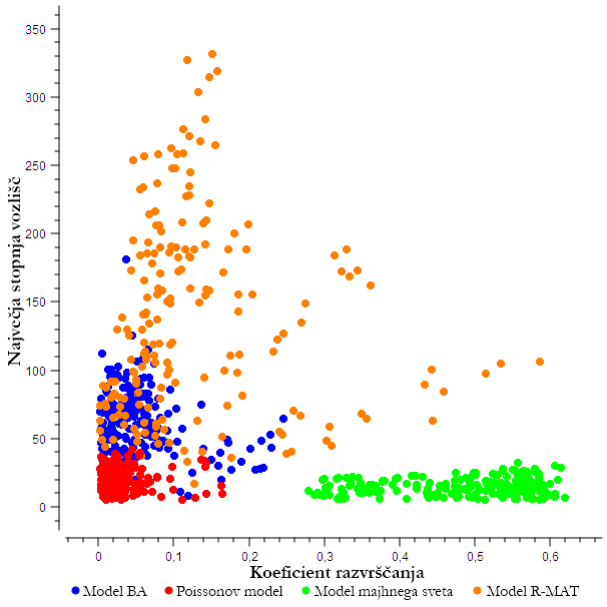
Tabela 5.5: Primerjava lastnosti omrežja delfinov in naključno generiranih omrežij.

	Vozlišča	Povezave	α	Napaka R^2	Največja stopnja	Povpr. stopnja	Premer	Eksp. hop	Koeficient razvrščanja
Omr. delfinov:	62	159	-0.19	0.0436	12	5	5	1.281	0.2590
Model maj. sv.:	62	124	0.75	0.1935	7	4	5	1.543	0.2122
Model maj. sv.:	10,000	20,000	1.53	0.1710	8	4	11	3.309	0.2670

5.4 Analiza modela R-MAT

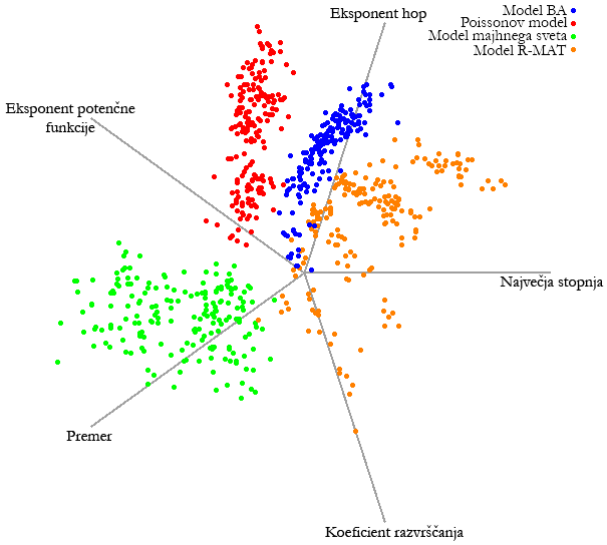
Za zaključek analize nabor modelov naključnih omrežij modela M-generator razširimo še z modelom R-MAT. Ogledamo si lastnosti omrežij generiranih z modelom R-MAT in rezultate pridobljene z razširjenim modelom M-generator.

Avtorji modela R-MAT navajajo, kot smo zapisali v razdelku 3.4, oblikovanje skupnosti znotraj omrežja in potenčno porazdelitev stopenj. Na slikah 5.7 in 5.8 vidimo z oranžno barvo označena omrežja modela R-MAT pri enakih projekcijah, kot smo si poprej ogledali osnovne modele. Če sliko 5.7 primerjamo s sliko 5.2, opazimo da se omrežja generirana z modelom R-MAT precej bolj približajo območju realnih omrežij, kot omrežja osnovnih modelov.



Slika 5.7: Primerjava porazdelitve osnovnih modelov in modela R-MAT na grafu.

Prav tako na sliki 5.8 opazimo, da imajo omrežja modela R-MAT potenčno porazdelitev stopenj, podobno kot model BA, vendar pa imajo višji koeficient razvrščanja, kar kaže na bolj goste skupnosti znotraj omrežja. Z omrežji modela R-MAT smo tako praznino v spodnjem desnem kotu grafa na sliki 5.8 uspešno zmanjšali, vendar pa je le-ta še vedno prisotna.



Slika 5.8: Prikaz porazdelitve osnovnih naključnih omrežij in modela R-MAT na grafu s petimi osmi. Slika je bila generirana z orodjem VizRank iz paketa Orange [13].

Iz tabele 5.6 (v primerjavi s tabelo 5.2) vidimo, da je klasifikator zamenjal izbiro modela BA z modelom R-MAT. Razlika je v primeru omrežja sodelovanja med znanstveniki, kjer sedaj namesto modela BA, klasifikator izbere model majhnega sveta. Tu je možno, da gre za napačno klasifikacijo, saj je tudi verjetnost izbire modela nižja od ostalih.

Klasifikator tako modela BA ni izbral v nobenem primeru. Glede na sliko 5.8 je to nekako očitno, saj model R-MAT generira boljše skupnosti v omrežju ob ohranjanju potenčne porazdelitve stopenj. Vendar pa je klasifikator model R-MAT vedno izbral z manjšo verjetnostjo, kot v prejšnjem primeru model BA, kar kaže na to, da sta si modela podobna in zato klasifikator modela R-MAT ne izbere z večjo verjetnostjo.

Tabela 5.6: Izbrani modeli za modeliranje realnih omrežij.

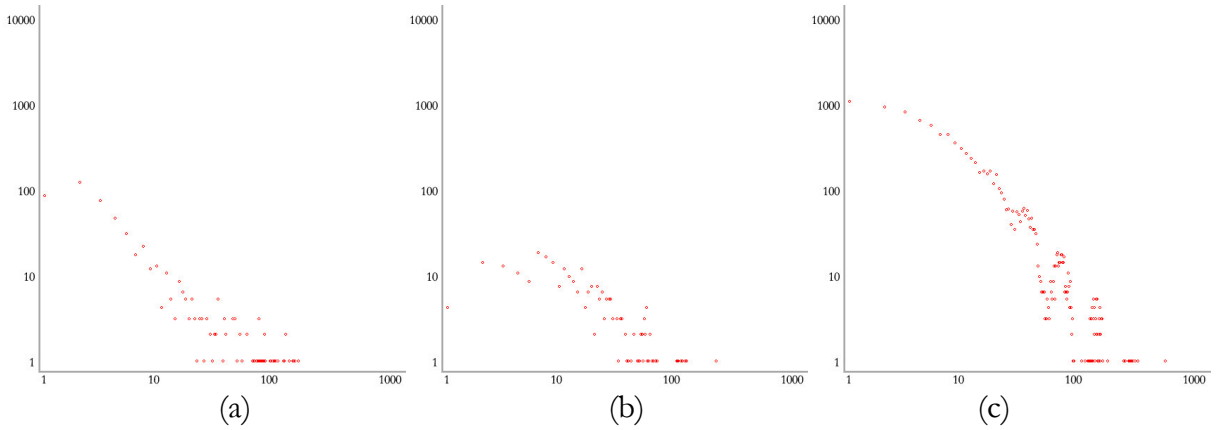
Analizirano omrežje	**Izbran model**	**Verjetnost**
Omrežje 500-tih največjih komercialnih letališč v ZDA	Model R-MAT	0.785
Omrežje sodelovanja med znanstveniki	Model majhnega sveta	0.606
Omrežje ženskega kluba	Poissonov model	0.995
Električno omrežje zahodnega dela ZDA	Model majhnega sveta	0.829
Socialno omrežje delfinov	Model majhnega sveta	0.722
Nevronsko omrežje	Model R-MAT	0.930

Podobno kot prej, naredimo še analizo za omrežje ameriških letališč. Kot vidimo iz tabele 5.7, se lastnosti generiranega omrežja zelo lepo ujemajo z realnim primerom (slika 5.9). Kljub temu opazimo, da skupnosti v omrežju še vedno niso tako goste, kot bi si želeli. Koeficient razvrščanja je sicer večji kot pri modelu BA, vendar je še vedno daleč od realnega.

Tabela 5.7: Primerjava realnega omrežja in omrežij generiranih z modelom R-MAT.

	Vozlišča	Povezave	α	Napaka R^2	Največja stopnja	Povpr. stopnja	Premer	Eksp. hop	Koeficient razvrščanja
Omrežje letališč:	500	2,980	-0.761	0.6581	145	11	4	2.610	0.6175
Model R-MAT:	500	2,874	-0.656	0.7414	206	11	3	2.257	0.1759
Model R-MAT:	10,000	49,862	-1.175	0.7664	539	9	4	4.422	0.01397

Iz slik porazdelitve stopenj (slika 5.9) opazimo, da porazdelitev stopenj omrežij modela R-MAT v repu porazdelitve sledi potenčnemu zakonu. V primerjavi z omrežji modela BA (slika 5.6) opazimo, da se porazdelitev stopenj omrežij modela BA lepše prilega premici. To je nekako pričakovano, saj se model BA pri gradnji omrežja osredotoči zgolj na potenčno porazdelitev stopenj, model R-MAT pa poleg tega skrbi tudi za ostale lastnosti omrežja.



Slika 5.9: Porazdelitev stopenj za originalno omrežje letališč (a), porazdelitev stopenj za naključno omrežje generirano z modelom R-MAT velikosti 500 vozlišč (b) in 10,000 vozlišč (c). Uporabljen je »log-log« graf.

Model R-MAT se izkaže za dobro izboljšavo obstoječih modelov. Sicer generirana omrežja še vedno ne posnemajo do potankosti realnih omrežij, vendar je predstavljeni model vsekakor korak v pravo smer.

Povzamemo lahko, da klasifikator nov model izbere povsod, kjer se pokažejo izboljšave v primerjavi s starim modelom BA. Z modelom R-MAT tako izboljšamo sam M-generator, poleg tega pa potrdimo tudi primernost klasifikatorja.

6 Zaključek

V nalogi predstavimo nov pristop h generiranju naključnih omrežij, ki ga poimenujemo M-generator (Meta generator). Omogoča generiranje naključnih omrežij, ki posnemajo lastnosti realnega omrežja brez da bi bilo potrebno te lastnosti poznati vnaprej. Tako lahko v nadaljnjih raziskavah uporabimo naključno omrežje z zaupanjem, da se obnaša podobno, kot bi se omrežje realnega nabora podatkov.

M-generator za generiranje naključnih omrežij uporablja že poznane modele naključnih omrežij. Njegova prednost je ta, da omrežja generirana s posameznim modelom naključnih omrežij samodejno analizira ter konstruira tudi ustrezen klasifikator. Ob generiranju novih naključnih omrežij s pomočjo klasifikatorja izbere model, ki naj bi v danem primeru najbolj ustrezal. Izbira je povsem avtomatska in pri generiranju omrežij tako ni potrebno posredovanje domenskega eksperta.

Delovanje M-generatorja smo preizkusili nad realnim naborom omrežij. Lastnosti naključno generiranih omrežij smo primerjali z realnimi in bili z rezultati zadovoljni, saj so lastnosti naključno generiranih omrežij lepo sledile realnim. Vendar zaradi pomanjkanja označenih podatkov ne moremo sklepati o uspešnosti pristopa v splošnem. Kljub temu je M-generator sposoben samodejno generirati zelo dobra naključna omrežja, tako da lahko zaključimo, da so cilji naloge v veliki meri doseženi.

Model pa bi bilo moč z različnimi klasifikatorji in atributi (lastnosti omrežij) še izboljšati. Med lastnostmi omrežij se zdita obetajoči elastičnost omrežja in vmesna centralnost vozlišč. Pri obeh pa naletimo na težavo pri preslikavi v atributni zapis primeren za konstruiranje klasifikatorja. V obeh primerih lahko narišemo graf porazdelitve, vprašanje pa je če so si grafi omrežij med seboj podobni, ter kako jih med seboj učinkovito primerjati.

Možno izboljšavo bi predstavljalo tudi konstruiranje in testiranje klasifikatorja nad realnimi omrežji. Pri tem bi potrebovali veliko in reprezentativno množico omrežij, katero bi morali dodatno označiti s pomočjo domenskega eksperta (ali eno od metod primerjanja omrežij).

Izvedba te izboljšave je dejansko možna za manjši, specifični nabor omrežij. V splošnem pa bi bila vpeljava takšne izboljšave težje izvedljiva.

Seznam slik

Seznam tabel

Viri in literatura

[1] L. A. Adamic, R. M. Lukose, A. R. Puniyani, B. A. Huberman, "Search in power-law networks", *Physical Review*, št. 64, 046135, 2001.

[2] L. A. Adamic, R. M. Lukose, B. A. Huberman, "Local Search in Unstructured Networks", *Handbook of Graphs and Networks: S. Bornholdt, H. G. Schuster*, 2003.

[3] A. L. Barabási, R. Albert, "Emergence of Scaling in Random Networks", *Science*, št. 286, str. 509-512, 1999.

[4] A. L. Barabási, R. Albert, H. Jeong, "Mean-field theory for scale-free random networks", *Phys. A*, št. 272, str. 173-187, 1999.

[5] I. Bezáková, A. Kalai, R. Santhanam, "Graph Model Selection using Maximum Likelihood", v zborniku *The 23rd International Conference on Machine Learning*, Pittsburgh, 2006.

[6] B. Bollobás, O. Riordan, "The Diameter of a Scale-Free Random Graph", *Combinatorica*, 2002.

[7] K. M. Borgwardt, "Graph Kernels", Doktorska Disertacija, Universität München, 2007.

[8] D. Chakrabarti, C. Faloutsos, "Graph Mining: Laws, Generators, and Algorithms", *ACM Computing Surveys*, št. 38, zv. 2, 2006.

[9] D. Chakrabarti, Y. Zhany, C. Faloutsosz, "R-MAT: A Recursive Model for Graph Mining", v zborniku *SIAM Data Mining Conference*, Philadelphia, PA, ZDA, 2004.

[10] A. Clauset, C. R. Shalizi, M. E. J. Newman, "Power-law Distributions in Empirical Data", 2009, dostopno na: http://arxiv.org/abs/0706.1062.

[11] V. Colizza, R. Pastor-Satorras, A. Vespignani, "Reaction-Diffusion Processes and Metapopulation Models in Heterogeneous Networks", *Nature Physics*, št. 3, str. 276-282, 2007.

[12] A. Davis, B. B. Gardner, M. R. Gardner, "Deep South", *University of Chicago Press*, Chicago, IL, ZDA, 1941.

[13] J. Demšar, B. Zupan, "Orange: From Experimental Machine Learning to Interactive Data Mining", Fakulteta za Računalništvo in Informatiko, Univerza v Ljubljani, 2004.

[14] S. N. Dorogovtsev, J. F. F. Mendes, A. N. Samukhin, "Structure of Growing Networks with Preferential Linking", *Phys. Rev. Lett.*, št. 85, str. 4633-4636, 2000.

[15] P. Erdös, A. Rényi, "On Random Graphs", *Publ. Math. Debrecen*, št. 6, str. 290-297, 1959.

[16] P. Flajolet, G. N. Martin, "Probabilistic Counting Algorithms for Data Base Applications", *Journal of Computer and System Sciences*, št. 31, zv. 2, str. 182-209, 1985.

[17] G. W. Flake, S. Lawrence, C. L. Andgiles, "Efficient identification of Web communities", v zborniku *The Sixth International Conference on Knowledge Discovery and Data Mining*, Boston, MA, ZDA, aug. 2000, str. 150-160.

[18] T. Gärtner, P. Flach, S. Wrobel, "On Graph Kernels: Hardness Results and Efficient Alternatives", *LNAI 2777*, str. 129-143, 2003.

[19] K.-I. Goh, E. Oh, H. Jeong, B. Kahng, D. Kim, "Classification of scale-free networks", *PNAS*, št. 99, zv. 20, str. 12583-12588, 2002.

[20] M. L. Goldstein, S. A. Morris, G. G. Yen, "Problems with Fitting to the Power-Law Distribution", *The European Physical Journal B*, št. 41, zv. 2, str. 255-258, 2004.

[21] M. Granovetter, "The Strength of Weak Ties: A Network Theory Revisited", *Sociological Theory*, št. 1, str. 201-233, 1983.

[22] H. Kashima, A. Inokuchi, "Kernels for Graph Classification", *ICDM Workshop on Active Mining*, 2002.

[23] J. M. Kleinberg, "Navigation in a small world", *Nature*, št. 406, str. 845, 2000.

[24] J. M. Kleinberg, "The Small-World Phenomenon: An Algorithmic Perspective", v zborniku *The 32nd Annual ACM Symposium on Theory of Computing, Association of Computing Machinery*, New York, 2000, str. 163-170.

[25] P. L. Krapivsky, S. Redner, "Organization of Growing Random Networks", *Physical Review E*, št. 63, zv. 6, 2001.

[26] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, Z. Ghahramani, "Kronecker graphs: An Approach to Modeling Networks", 2009, dostopno na: http://arxiv.org/abs/0812.4905v2.

[27] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, S. M. Dawson, *Behavioral Ecology and Sociobiology*, št. 54, str. 396-405, 2003.

[28] J. Moody, "Race, School Integration, and Friendship Segregation in America", *Amer. J. Sociol.*, št. 107 zv. 3, str. 679-716, 2001.

[29] M. E. J. Newman, "Power laws, Pareto Distributions and Zipf's law", *Contemporary Physics*, št. 46, zv. 5, str. 323-351, 2005.

[30] M. E. J. Newman, "The Structure and Function of Complex Networks", *SIAM Review*, št. 45, zv. 2, str. 167-256, 2003.

[31] M. E. J. Newman, "The Structure of Scientific Collaboration Networks", v zborniku *The National Academy of Sciences of the United States of America*, št. 98, str. 404-409, 2001.

[32] C. R. Palmer, P. B. Gibbons, C. Faloutsos, "ANF: A Fast and Scalable Tool for Data Mining in Massive Graphs", v zborniku *The Eighth ACM SIGKDD International Conference Discovery and Data Mining*, Edmonton, Alberta, Kanada, jul. 2002.

[33] D. J. de S. Price, "A General Theory of Bibliometric and Other Cumulative Advantage Processes", *Journal of the American Society for Information Science*, št. 27, str. 292-306, 1976.

[34] D. J. de S. Price, "Networks of Scientific Papers", *Science*, št. 149, str. 510-515, 1965.

[35] M. F. Schwartz, D. C. M. Wood, "Discovering Shared Interests Using Graph Analysis", *Comm. ACM*, št. 36, zv. 8, str. 78-89, 1993.

[36] N. Shervashidze, S.V.N. Vishwanathan, T. H. Petri, K. Mehlhorn, K. M. Borgwardt, "Efficient graphlet kernels for large graph comparison", v zborniku *The 12th International Conference on Artificial Intelligence and Statistics*, Clearwater Beach, Florida, ZDA, 2009.

[37] H. Simon, "On a Class of Skew Distribution Functions", *Biometrika*, št. 43, str. 425-440, 1955.

[38] R. Solomonoff, A. Rapoport, "Connectivity of Random Nets", *Bulletin of Mathematical Biophysics*, št. 13, 1951.

[39] S. H. Strogatz, "Exploring Complex Networks", *Nature*, št. 410, str. 268-276, 2001.

[40] S. L. Tauro, C. Palmer, G. Siganos, M. Faloutsos, "A Simple Conceptual Model for the Internet Topology", *Global Internet*, 2001.

[41] J. Travers, S. Milgram, "An Experimental Study of the Small World Problem", *Sociometry*, št. 32, vol. 4, str. 425-443, 1969.

[42] S. Wasserman, P. Pattison, "Logit Models and Logistic Regressions for Social Networks", *Psychometrika*, št. 61, zv. 3, str. 401-425, 1996.

[43] D. J. Watts, "Networks, Dynamics, and the Small-World Phenomenon", *AJS*, št. 105, zv. 2, str. 493-527, 1999.

[44] D. J. Watts, P. S. Dodds, M. E. J. Newman, "Identity and search in social networks", *Science*, št. 296, str. 1302-1305, 2002.

[45] D. J. Watts, S. H. Strogatz, "Collective Dynamics of 'Small-World' Networks", *Nature*, št. 393, str. 440-442, 1998.

[46] L. Zager, "Graph Similarity and Matching", Magistrsko delo, Massachusetts Institute of Technology, 2005.