VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Kursinis darbas

Dokomentų klasterizacija

(Document clustering)

Atliko: 3 kurso 1 grupės studentas
Dominykas Ablingis (parašas)

Darbo vadovas:
lekt. Rimantas Kybartas (parašas)

Vilnius
2016

Turinys

Sąvokų apibrėžimai

Sutartinių ženklų, simbolių, vienetų ir terminų sutrumpinimų sąrašas (jeigu ženklų, simbolių, vienetų ir terminų bendras skaičius didesnis nei 10 ir kiekvienas iš jų tekste kartojasi daugiau nei 3 kartus).

Įvadas

`Įvade apibūdinamas darbo tikslas, temos aktualumas ir siekiami rezultatai. Darbo įvadas neturi būti dėstymo santrauka. Įvado apimtis 1-2 puslapiai. taip ir anaip o gal ir kitaip`

Šiais laikais kai kiekvienas žmogus turintis priėjimą prie interneto gali dalintis informacija, daugybė knygu yra skaitmenizuojamos kiekvieną dieną ir mokslo institucijos dalinasi savo moksline informacija, pasiekemos informacijos kiekis didėja su kiekviena diena ir pagrindinė problema nebėra informacijos trūkumas, o atradimas ko reikia. Tam spresti buvo ir yra kuriama ivairūs mechanizmai: paiškos varikliai... Šiame darbe autorius nagrinės viena iš šios problemos sprendimo metodų, klasterizacijos algoritmus ir jų panaudojimą textinems dokumentams. Dokomentų klasterizacijos panaudojimai:

- Automatinis dokumentų organizavimas. Dažnai alternatyvus šios problemos sprendimas yra dokumentų žymėjimas (angl. taging)
- Temų **išgavimas**
- Greitas informacijos paiška ir filtravimas, ypač

Šiame darbe taip pat bus paminėtos kitos su dokumentu klasterizacija susijusios problemos ir ši informacija bus išdėstyta eilės tvarka kaip būtu sprendžiamos užduotys.

1. Duomenų išgavimas iš skirtingų medijų
2. Duomenų apdorojimas
3. Algoritmai
4. Algoritmų testavimas
5. Rezultatų vizuoalizacija

1. Pagrindinė tiriamoji dalis

Pagrindinėje tiriamojoje dalyje aptariama ir pagrindžiama tyrimo metodika; pagal atitinkamas darbo dalis, nuosekliai, panaudojant lyginamosios analizės, klasifikacijos, sisteminimo metodus bei apibendrinimus, dėstoma sukaupta ir išanalizuota medžiaga.

2. Duomenų išgavimas ir apdorojimas

Duomenų apdorojimas Prieš naudojant bet kokį mašininio mokymosi metodą dažniausiai reika pradėti nuo duomenų apdorojimo. Šiuo atveju duomenys yra dokumentai kurie gali būti pateikti įvairiais formatais: moksliniai darbai Latex formatu, internetiniai puslapiai html fotmatu, … Dažniausiai šie formatai yra transliuojami i patogesnią apdorojimui formą. Taip pat dažnai susiduriant su žmogišku tekstu reikia ji paildomai apdoroti. Dažnai išemami stop words (nekaitomos kalbos dalys)(nebent atilekama frazių analizė) ir ženklai(!?...), žodžiai pakeičiami į bendrinię formą, kad supaprastini apdorojamą teksta neprarandant gilesnės teksto minties.

2.1. Duomenų išgavimas

Informacijos išgavimas iš interneto ir kitų šaltinių XML atvėju `<tag>Content</tag>` tag galime interpretuoti kaip kintamąjį .Galima interpretuoti kaip kintamoji Content kaip kintamojo reikšmę, arba tag kaip teksto anotacija. Bet tai negalioja html atveju tagai Skirti nurodyti svetainės išdėstymą. Pavyzdžiui <h1> dažniausiai reiškia pavadinimą ar antrašnę ir XML atvėju turbūt būtu <title>. Taip pat Interneto svetainės turi kitos tekstinės informacijos kaip navigacija, komentarai, kontaktai ir panašiai. Šia problema sprendžiam keliais žingsniais: Pirma galime pasinaudoti tuo kad HTML yra DOM … todėl tai yra medžio struktūra. Supaprastinti darba galima apdorojant tik specifines šakas, bet tarp skirtingu svetainių šis medis atrodys skirtingai, taigi to pakaks tik dalinai Kurkas paprastesnis būdas pasinaudoti statisniu metodu(Finn's method). Atskireme tagus į skirtus tekstui(<bold>, <italic>...) ir neskirtus(<head>, <body>...). Tada stebime tagų(neskirtu tekstui) pasiskirstyma puslapyje lyginat su tekstu, ten kur ju daug galime numanyti kad tai nera pagrindinis puslapio turinys ir vice versa. Tai pavaizdavus grafike pamatytume išsilyginima

2.2. Duomenų apdorojimas

2.2.1. Tokenization - teksto išskaldymas į reikšmingas dalis

Žodžius, frazes ir t.t. vadinamas token. Vėliau šie žodžiai bus naudojami kaip ideksai žodyne. Tekstų tokenizacija yra vis dar aktyviai tiriama sritis, ypač kalboms kuriose nėra aiškių žodžių ribų (`https://en.wikipedia.org/wiki/Scriptio_continua`). Taip pat problematiški žodžiai su ženkalais viduje: I.B.M.; pre-diabetes. Tokiais atvėjais netinka nei atskirti nei sujungti nes abejais atvėjais prarandama prasmė ir sukuremi netikslūs token'ai. Yra keli sprendimai: vienas padaryti abu (dalinti ir jungti) tokiu butu atsiranda daugiau triukšmo duomenyse, bet tai neturetu buti problema jeigu tvarkingi svoriai . Morfologinė variacija. Problema su žodžiais kur viena šaknis gali turėti kelias skirtingas prasmes. Arabų kalba ypač sudėtinga šituo atžvilgiu nes turi palygint mažai šaknų, bet labai daug variacijų. Pokyčiai gali būti prieš, po ir pačioje šaknyje.

2.2.2. Stemming - žodžio šaknies išgavimas.

(https://en.wikipedia.org/wiki/Stemming). Dažnai programos (stemmers) gražina ne žodžius o šaknis. Egzistuoja keli implementacijos būdai: Paremti taisyklėmis, taisykles + išimčių žodynas. Šie veikia neblogai bet reikalauja labai gero kalbos pažinimo ir yra labai komplikuoti. Kitas būdas raidžių n-gramos, panašiuose žodžiuose panašios n-gramos. Standartiškai Europinėms kalboms n = 4. Stemming'as gali ivykti skirtinguose apdorojimo etapuose priklausomai nuo užduoties. Stop-words išėmimas. Tokie žodžiai kaip of, the, to turi palyginti mažai reikšmės, nenaudingi bandant diskriminuoti, pagrinde yra teksto "klyjai". You vertinga išmesti nes sunaudoja daug resursų, pvz. "of" + "the" = sudaro 10% kalbos. Todėl dažniausiai naudojami Stop-word žodynai. Reikia atkreipti dėmėsį, kad skirtingose srityse turi skirtingus stop-word(pvz internete žodis "click"). Taip pat kai kurios frazės gali būti sudarytos iš stop-word ir turėti prasmę ("to be or not to be").

2.2.3. Sinonimiškumas ir Polisemija

Indexų pavyzdžiai: Wordnet, MeSH. Taip pat yra daug statistinių metodų. Kartais prašoma vartotojų Prašoma nurodyti kurią iš sinonimo reikšmių jie turėjo omeny arba vertinti paieškos rezultatus nurodant kurie atitinka o kurie ne, tokiu butu paieškos problema paversti į supervised learning problemą (relevance feedback).

3. Algoritmai

Klasterizacijos algoritmai yra skirstomi į šias rūšis: Klasterizacijos algoritmai gali padėti atsakyti į klausimus:

- Ar ir kiek sub-populiaciju turi mano duomenys
- Kokio dydzio tos populiacijos
- Ar sub-populiacijos turi bendrų savybių
- Ar sub-populiacijos yra vientisos ar jas galima papildomai išskaldyti

Klasteringo algoritmai skirstomi į šiuos tipus tipai

- monothetic – visi populiacijos nariai turi kažkokia bendrą savybę(vyrai nuo 20 iki 25 metų amžiaus)
  polythetic – nariai yra panšūs, bet neturi konkrečios bendros savybės (kai atstumas tarp narių nurodo priklausomybę grupei)
- Hard(kieti) klasteriai – kai grupės neturi bendrų narių. Kartais gali atsirasti atėjai kai elementas “matematiškai” gali priklausyti ne vienai grupei, bet toks atvėjis yra retas.
  Soft(minkštas)klasteriai – kai grupės gali turėti bendrų narių. Tokiu atveju galime nagrinėti kaip konkretus elementas priklauso skirtingoms grupėms ir kaip gerai jas atitinka.
- flat (plokščia) – kai elementai suskirstomi į grupes, kurios viena kitai yra lygios.
  Hierarchical(herarkiškas)(taxonomy) – Kai grupė gali būti sudaryta iš kelių “konkretesnių” grupių. Pvz. retryveris -> šuo -> žinduolis -> gyvūnas

3.1. K-means

Šis algoritmas suskirsto duomenis į k klasterių.

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}\in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

Praktikoje dažniausiai naudojamas Duodama n vektroių(su d dimensijų) ir numeris k. Sustatome k centroidų c į atsitiktines vietas.

1. Kiekvienam taskui(vektoriui) randame artimiausia c
2. Kiekviena centroida pakeiti taip kad jis geriau atitiktu jam priskirtą klusterį(sumažinti atstuma iki visų taškų arba kitaip tariant rasti taškų centrą). Tada eini i žingsni 1. tol kol visi klusteriai išlieka vienodi.

Big O (iteracijos*K*n*d) Produce hard, flat, polythetic cluster.

3.2. Hierarchical clustering

Šis metodas neskirsto dokumentu į konkrečias grupes(nors vėliau parodysim kad gali), bet vietoj to suskirsto dokuemntus į herarhicja. Tam atlikti reikiai būdo kaip matuoti dokumentų panašumą.

3.3. Latent Dirichlet allocation

Latent Dirichlet allocation(toliau LDA, nesumaišyti su Linear discriminant analysis, kas yra kitas mašininio mokymosi metodas). Yra kurkas sudetingesnis dokumentu analizavimo metodas. Kartais priskiremas atskirai algoritmu klasei Topic modeling, kuri bando ne grupuoti duomenis o suteigti jiems temas. Palyginant su praeitais metodais dokumentai turėjo priklausyti vienai iš klasterių ar jų herarkijai, bet dokumentas gali turėti kelias temas. Todėl LDA gražina ne sugrupuotus duomenis, o pasiskirstymą kiek kokių temų turi kiekvienas dokumentas

4. Algoritmų testavimas

Viena iš fundamentalių neprižiūrimo mokymosi problemų yra modelių testavimas. Skirtingai nei ”prižurimame mokymasi” kur svarbu atdidėti dali duomenų su kuriais nėra mokomasi o tik testuojama sugeneruoti modeliai, ”neprižiurimame mokymasi” mes to nelgalim atlikti..... Bet egzsituoja keltatas metodu kaip galima patikrinti sudarytus klasterius.

- Pirma galima sugeneruotus klasterius leisti tikrinti **žmonėms** . Tam yra keli būdai. Galima tiesiog duoti sugeneruotus klasterius ir bandyti nuspresti ar jie tinkami. Taipat galima parainkti du atsitiktinius dokumentus ir spėti ar jie turėtu būti viename ar atskiruose klasteriuose, ir tada palyginti su kompiuterio sugeneruotu rezultatu.

- Taip yra metodų kaip atlikti testavimą automatiškai. Vienas jų paimti 2(ar daugiau) dokumentus iš skirtingų klasterių ir apkeisti juos vietomis, tada patikrinti klasteriu //stipruma. Tai atlike dokybe kartų galime spręsti kaip sėkmingai sekėsi klasterizacija, jeigu apmainant jų kokybė nukrito tai reikškia, kad dokumentai sėkmingai suklaserizuoti, bet jeigu nesikeite tai reiške, kad klasteriai mažai vienas nuo kito skiresi ir klasreizacija nesekminga.

Testuojame ilga teksta Testuojame ilga tekstaame ilga teksta Testuojame ilga ame ilga teksta Testuojame ilga ame ilga teksta Testuojame ilga ame ilga teksta Testuojame ilga enteris ame ilga teksta Testuojame ilga ame ilga teksta Testuojame ilga ame ilga teksta Testuojame ilga ame ilga teksta Testuojame ilga ame ilga teksta Testuojame ilga ame ilga teksta Testuojame ilga ame ilga teksta Testuojame ilga ame ilga teksta Testuojame ilga

Dvigubas enterisame ilga teksta Testuojame ilga ame ilga teksta Testuojame ilga ame ilga teksta Testuojame ilga ame ilga teksta Testuojame ilga ame ilga teksta Testuojame ilga ame ilga teksta Testuojame ilga ame ilga teksta Testuojame ilga ame ilga teksta Testuojame ilga

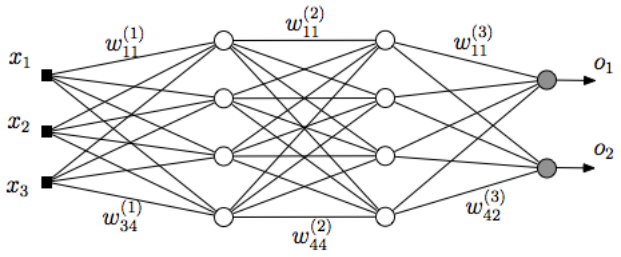
5. Rezultatų vizuoalizacija

Išvados

Išvadose ir pasiūlymuose, nekartojant atskirų dalių apibendrinimų, suformuluojamos svarbiausios darbo išvados, rekomendacijos bei pasiūlymai.

Priedas Nr. 1

Niauroninio tinklo struktūra



1 pav. Paveikslėlio pavyzdys

Priedas Nr. 2
Eksperimentinio palyginimo rezultatai

1 lentelė. Lentelės pavyzdys

Algoritmas	$\bar{x}$	σ^2
Algoritmas A	1.6335	0.5584
Algoritmas B	1.7395	0.5647