Normalisoitu pakkausetäisyys: sovelluksia ja variaatioita

Timo Sand

Kandidaatintutkielma
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Helsinki, 3. joulukuuta 2013

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Matemaattis-luonnontieteellinen	Tietojenkäsittelytieteen laitos

Tekijä — Författare — Author
Timo Sand

Työn nimi — Arbetets titel — Title
Normalisoitu pakkausetäisyys: sovelluksia ja variaatioita

Oppiaine — Läroämne — Subject
Tietojenkäsittelytiede

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Kandidaatintutkielma	3. joulukuuta 2013	14

Tiivistelmä — Referat — Abstract

Tähän tulee tutkielman tiivistelmä

ACM Computing Classification System (CCS):
Information systems - Similarity measures
Theory of computation - Unsupervised learning and clustering
Data mining - Clustering

Avainsanat — Nyckelord — Keywords
Samankaltaisuus

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Sisältö

1 Johdanto

Kaikki data on luotu samanveroiseksi, mutta jotkut datat ovat samankaltaisempia kuin toiset. Esitämme tavan, jolla ilmaista tämä samankaltaisuus, käyttäen Cilibrasin ja Vitanyin esittelemää samankaltaisuuden metriikkaa *(engl. similarity metric)*, joka perustuu tiedoston pakkaamiseen. [RV05] Metriikka on parametriton, eli se ei käytä datan ominaisuuksia tai taustatietoja, ja sitä voi soveltaa eri aloihin ilman muunnoksia. Metriikka on universaali siten, että se approksimoi parametrin, joka ilmaisee parin hallitsevan piirteen samankaltaisuutta pareittain vertailuissa. Se on vakaa siinä mielessä, että sen tulokset ovat riippumattomia käytetystä pakkaajasta [RV05]. Pakkaajalla tarkoitetaan pakkausohjelmaa kuten *gzip, ppmz, bzip2*.

Pakkaukseen perustuva samankaltaisuus *(engl. Compression-Based Similarity)* on universaali metriikka, jonka kehittivät Cilibrasi ja Vitanyi [RV05]. Yksinkertaistettuna tämä tarkoitta, että kaksi objektia ovat lähellä toisiaan, jos voimme "pakata" yhden objektin huomattavasti tiiviimmin toisen objektin datalla. Abstraktina ideana toimii se, että voimme kuvailla ytimekkäämmin yhden objektin toisen avulla, mikäli objektit ovat samankaltaisia. Tämän esittelemme luvussa 2 ja samalla käymme läpi, mihin teoriaan algoritmi perustuu sekä miten se toimii. Edellä mainitun vakauden esittämiseen voimme käyttää useaa tosielämän pakkausalgoritmia: tilastollista (PPMZ), Lempel-Ziv -algoritmiin pohjautuvaa hakemistollista (gzip), lohkoperusteista (bzip2) tai erityistarkoitusta varten suunniteltua (Gencompress).

Kaikissa kohdissa operoimme $O(logn)$ tarkkuudella, joka on hinta siitä että siirrytään Kolmogorov-kompleksisuudesta laskettavaan approksimaatioon.

Tarkoituksemme on koota yhteen samankaltaisuuden metriikkaan kaikki tehokkaat etäisyydet *(engl. effective distances)*: tehokkaat versiot Hammingin etäisyydestä, Euklidisesta etäisyydestä, Lempel-Ziv etäisyydestä ja niin edelleen. Tämän metriikan tulee olla niin yleinen, että se toimii yhtäläisesti ja samanaikaisesti kaikilla aloilla: musiikilla, tekstillä, kirjallisuudella, ohjelmilla, genomeilla, luonnollisen kielen määrityksillä. Sen pitäisi pystyä samanaikaisesti havaitsemaan kaikki samankaltaisuudet objektien välillä, joita muut

etäisyydet havaitsevat erikseen.

Kun määrittelemme ryhmän sallittavia etäisyyksiä *(engl. admissible distances)* haluamme sulkea pois epärealistiset, kuten $f(x, y) = \frac{1}{2}$ jokaiselle parille $x \neq y$. Saavutamme tämän rajoittamalla objektien lukumäärän annetussa etäisyydessä objektiin. Teemme tämän huomioimalla vain todellisia etäisyyksiä seuraavasti: Oletamme kiinnitetyn ohjelmointikielen, joka toimii tutkielman ajan viitekielenä. Tämä ohjelmointikieli voi olla yleinen kieli kuten LISP, Ruby tai se voi myös olla kiinnitetty universaali Turingin kone. [RV05, CV07] Valinnalla ei kuitenkaan ole merkitystä, sillä teoria on invariantti ohjelmointikielin muutoksille, kunhan pysytään tehdyssä valinnassa.

Jotta voimme soveltaa tätä ideaalia ja tarkkaa matemaattista teoriaa käytäntöön, pitää meidän korvata ei laskettavissa oleva Kolmogorov-kompleksisuus approksimaatiolla käyttäen standardia pakkaajaa.

Luvussa 3 esittelemme algoritmin käyttökohteita monelta eri alueelta. Aloitamme siitä, miten yleisesti *Normalisoidun pakkausetäisyyden* (*NCD*) avulla pystymme klusteroimaan tuloksia eri kategorioihin; miten musiikkikappaleet klusteroituvat saman artistin alle, miten kuvantunnistuksessa saamme ryhmitettyä samankaltaiset kuvat ja miten sienten genomeista saamme tarkan lajiryhmityksen.

Luvun lopussa syvennymme musiikin, kuvantunnistuksen ja dokumenttien kategorisoinnin tuloksiin.

Luvussa 4 esitellään NCD:n ominaisuuksia ja ongelmia. Ensiksi esitellään NCD:n kohinansietokykyä, eli katsotaan mitä tapahtuu kun lisätään vähitellen kohinaa toiseen tiedostoista, jota pakataan, ja mittaamalla samankaltaisuutta tämän jälkeen [CAO07]. Saamme nähdä miten paljon kohina vaikuttaa NCD:n laskemiin etäisyyksiin ja huonontaako se klusteroinnin tuloksia.

Mikään algoritmi ei ole täydellinen, ja NCD-algoritmillakin on ongelmansa. Algoritmissa itsessään ei ole selvää heikkoutta, mutta sen käytössä on otettava pakkaajan valinta huomioon, koska monet suosituista pakkausalgoritmeista ovat optimoituja tietyn kokoisille tiedostoille. Niissä on niin kutsuttu ikkunakoko *(engl. window size)*, joka määrittelee mikä tiedostokoko on sopiva [CAO05].

Jos tiedostokoko on pienempi kuin ikkunakoko, niin pakkaus on tehokasta, mutta kun mennään siitä yli, pakkauksesta tulee huomattavasti tehottomampaa. Esittelemme tuloksia eri pakkausalgoritmien vertailuista ja mikä näistä algoritmeista on parhaimmaksi havaittu NCD:n kanssa käytettäväksi.

NCD ei ole ainut metriikka, jolla voidaan mitata samankaltaisuutta. Internetiä hyödyntäen on tehty metriikka, joka käyttää hakukoneita samankaltaisuuden tutkimiseen; tämä on nimetty Google-samankaltaisuusetäisyydeksi *(engl. Google Similarity Distance)*. Tämä toimii myös muilla hakukoneilla kuten Bing. Luvussa 5 esittellemme tämän sekä muita samankaltaisuuden metriikoita.

2 Normalisoitu iakkausetäisyys

2.1 Kolmogorov-kompleksisuus

Merkkijonon x *Kolmogorov-kompleksisuus* on pituus bitteinä, lyhimmästä tietokoneohjelmasta joka ilman syötettä palauttaa merkkijonon x; tämä merkitään $K(x) = K(x|\lambda)$, λ esittää tyhjää syötettä. Lyhimmän tietokoneohjelman pituus, joka palauttaa x syötteellä y, on Kolmogorov-kompleksisuus x:stä syötteellä y; tämä merkitään $K(x|y)$. Yksi tapa hahmottaa Kolmogorov-kompleksisuutta $K(x)$ on ajatella se pituutena bitteinä x:n tiiviimmin pakatussa muodossa, josta voidaan purkaa x pakkauksenpurkuohjelmalla.

2.2 Normalisoitu informaatioetäisyys

Artikkelissa [RV05] on esitelty *informaatioetäisyys* $E(x, y)$, joka on määritelty lyhimpänä binääriohjelman pituutena, joka syötteellä x laskee y:n ja syötteellä y laskee x:n. Tämä lasketaan seuraavasti:

$$E(x, y) = max\{K(x|y), K(y|x)\}. \tag{1}$$

Normalisoitu versio informaatioetäisyydestä $E(x, y)$, jota kutsutaan *normalisoiduksi informaatioetäisyydeksi*, on määritelty seuraavasti

$$NID(x, y) = \frac{max\{K(x|y), K(y|x)\}}{max\{K(x), K(y)\}}. \tag{2}$$

Tämä on paras mahdollinen *samankaltaisuuden metriikka*, ja sen on osoitettu [RV05] täyttävän vaatimukset etäisyyden metriikaksi. *NID* ei kuitenkaan ole laskettava tai edes semi-laskettava, koska Turingin määritelmän mukaan Kolmogorov-kompleksisuus ei ole laskettava [RV05]. Nimittäjän approksimointi annetulla pakkaajalla C on $max\{C(x), C(y)\}$. Osoittajan paras approksimaatio on $max\{C(xy), C(yx)\} - \min\{C(x), C(y)\}$ [RV05]. Kun *NID* approksimoidaan oikealla pakkaajalla, saadaan tulos jota kutsutaan *normalisoiduksi pakkausetäisyydeksi*. Tämä esitellään formaalisti myöhemmin.

2.3 Normaali pakkaaja

Seuraavaksi esitämme aksioomia, jotka määrittelevät laajan joukon pakkaajia ja samalla varmistavat *normalisoidussa pakkausetäisyydessä* halutut ominaisuudet. Näihin pakkaajiin kuuluvat monet tosielämän pakkaajat.

Pakkaaja C on *normaali* jos se täyttää seuraavat aksioomat:

1. *Idempotenssi*: $C(xx) = C(x)$ ja $C(\lambda) = 0$, jossa λ on tyhjä merkkijono,
2. *Monotonisuus*: $C(xy) \geq C(x)$,
3. *Symmetrisyys*: $C(xy) = C(yx)$ ja
4. *Distributiivisuus*: $C(xy) + C(z) \leq C(xz) + C(yz)$.

2.4 Normalisoitu Pakkausetäisyys

Jotta voimme soveltaa tätä ideaalia ja tarkkaa matemaattista teoriaa käytäntöön, pitää meidän korvata ei laskettavissa oleva Kolmogorov-kompleksisuus

approksimaatiolla käyttäen standardia pakkaajaa.

Normalisoitua versiota *hyväksyttävästä etäisyydestä* $E_c(x, y)$, joka on pakkaajaan C pohjautuva approksimaatio normalisoidusta informaatioetäisyydestä, kutsutaan nimellä *Normalisoitu Pakkausetäisyys (NCD)* [RV05]. Tämä lasketaan seuraavasti

$$NCD(x, y) = \frac{C(xy) - min\{C(x), C(y)\}}{max\{C(x), C(y)\}}. \tag{3}$$

NCD on funktioiden joukko, joka ottaa argumenteiksi kaksi objektia (esim. tiedostoja tai Googlen hakusanoja) ja tiivistää nämä, erillisinä ja yhdistettyinä. Tämä funktioiden joukko on parametrisoitu käytetyn pakkaajan C mukaan.

Käytännössä *NCD*:n tulos on välillä $0 \leq r \leq 1 + \epsilon$, joka vastaa kahden tiedoston eroa toisistaan; mitä pienempi luku, sitä enemmän tiedostot ovat samankaltaisia. Tosielämässä pakkausalgoritmit eivät ole yhtä tehokkaita kuin teoreettiset mallit, joten virhemarginaali ϵ on lisätty ylärajaan. Suurimmalle osalle näistä algoritmeista on epätodennäköistä että $\epsilon > 0.1$.

Luonnollinen tulkinta *NCD*:stä, jos oletetaan $C(y) \geq C(x)$, on

$$NCD(x, y) = \frac{C(xy) - C(x)}{C(y)}. \tag{4}$$

Eli etäisyys x:n ja y:n välillä on suhde y:n parannuksesta, kun y pakataan käyttäen x:ää, ja y:n pakkauksesta yksinään; suhde ilmaistaan etäisyytenä bittien lukumääränä kummankin pakatun version välillä.

Kun pakkaaja on normaali, niin *NCD* on normalisoitu hyväksyttävä etäisyys, joka täyttää metriikan yhtälöt, eli se on samankaltaisuuden metriikka.

Taulukossa 1 esitellään muutama arvo joilla NCD lasketaan ja mikä tulos näistä saadaan. Taulukon arvot saatu paperista [CAO05]

Arvot	Laskutoimitus	NCD:n arvo
$C(xx) = 26, C(x) = 17$	$\frac{26-17}{17}$	0.529
$C(xx) = 33, C(x) = 22$	$\frac{33-22}{22}$	0.5
$C(xx) = 30, C(x) = 17$	$\frac{30-17}{17}$	0.765
$C(xx) = 20, C(x) = 14$	$\frac{20-14}{14}$	0.428
$C(xx) = 16, C(x) = 14$	$\frac{16-14}{14}$	0.143
$C(xx) = 28, C(x) = 14$	$\frac{28-14}{14}$	1

Taulukko 1: Normalisoituja pakkausetäisyyksiä eri syötteille.

3 Käyttökohteet

3.1 Klusterointi

- Miten klusteroidaan
- Etäisyysmatriisi

3.1.1 Nelikkomenetelmä

- puun rakentaminen etäisyysmatriisista
- Puun operaatiot
- normalisoitu puun hyötyarvo $S(T)$
- Satunnaisuuden käyttö tarkistusiteraatiossa parhaan puun approksimoimiseksi

3.1.2 Tuloksia

- Musiikin genrepuu
- Musiikin pieni vertaus
- Musiikin keskikokoinen vertaus
- Musiikin suuri vertaus

3.2 Kuvantunnistus

4 Algoritmin ongelmat ja ominaisuudet

4.1 Kohinansietokyky

Kun NCD:tä käytetään kahteen eri tiedostoon toista näistä voi pitää kohinallisena versiona ensimmäisestä. Kohinan lisääminen progressiivisesti tiedostoon voi tuottaa tietoa mittarista *(engl. measure)* itsestään. Tämän vastaavuuden perusteella voimme tehdä teoreettisen päätelmän odotetusta kohinan lisäämisen vaikutuksesta algoritmiin, mikä selittää miksi NCD voi saada suurempia arvoja kuin 1 joissain tapauksissa. [CAO07]

4.2 Pakkaajan valinta

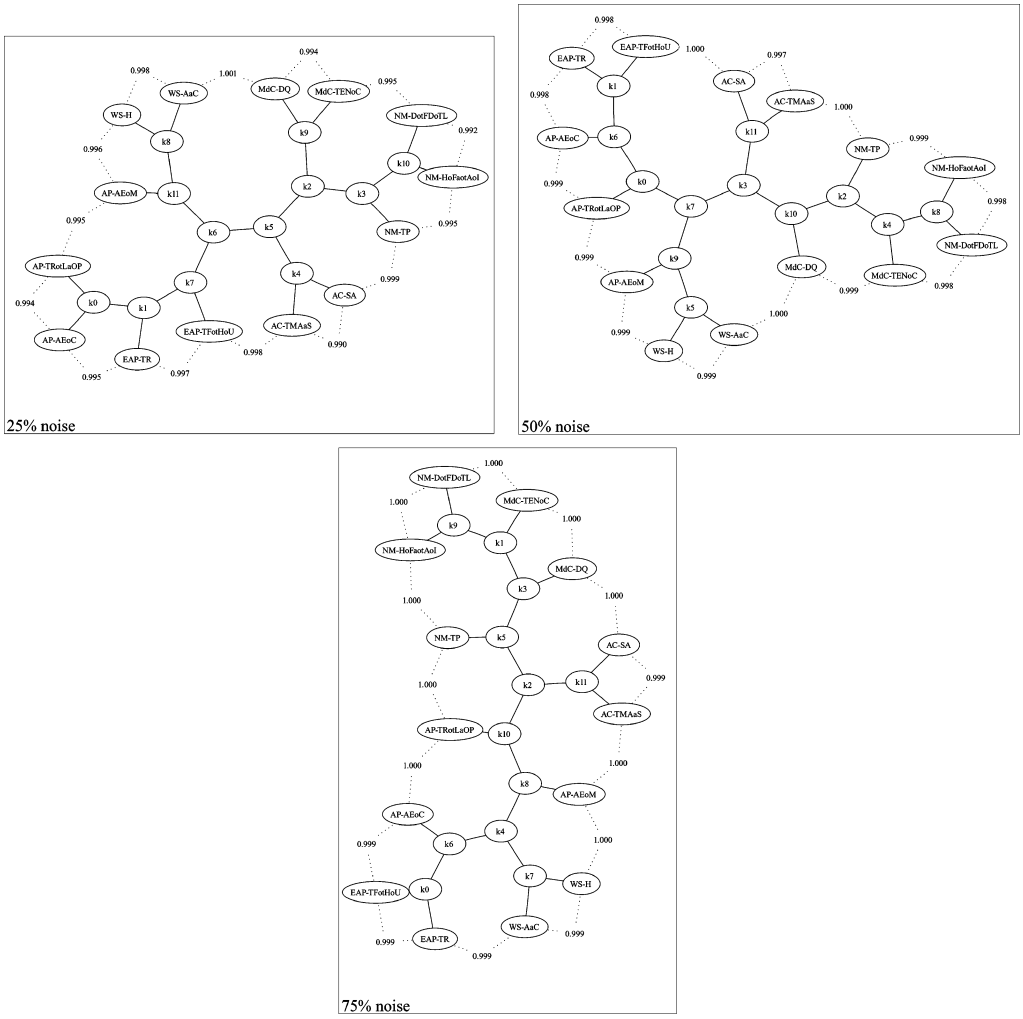
NCD vaikuttaa tuottavan vaikuttavia tuloksia klusteroinnissa, mutta tulokset ovat hyvin vahvasti riippuvaisia käytetyn pakkaajan ominaisuuksista. Suosittuja pakkaajia *bzip2*, *gzip* ja *PPMZ* on vertailtu NCD:n kanssa [CAO05], selvittääkseen mitä heikkouksia, jos mitään, näillä on.

Vertailussa käytettiin Cilibrasin toteuttamaa CompLearn -työkalua [Cil], josta löytyy bzip2 ja gzip pakkaajat. Aineistona käytettiin tunnettua Calgary Corpus -kokoelmaa [WBC], joka on 18 tiedoston kokoelma jolla mitataan pakkausalgoritmien suorituskykyä. Kokoelmassa on 9 eri tyyppistä tiedostoa, jotta voidaan saada laaja näkemys pakkausalgoritmin toiminnasta. Mukana muun muassa on kuva, kirjoja, artikkeleita, lähdekoodia ja tietokoneohjelmia.

Kaikkia vertailun objekteja käsitellään merkkijonoinen, jotka koostuvat tavuista. Jotta voitiin todeta pakkaajan idempotenssin (2.3) pätevyys, kaikki objektit vertailtiin itsensä kanssa.

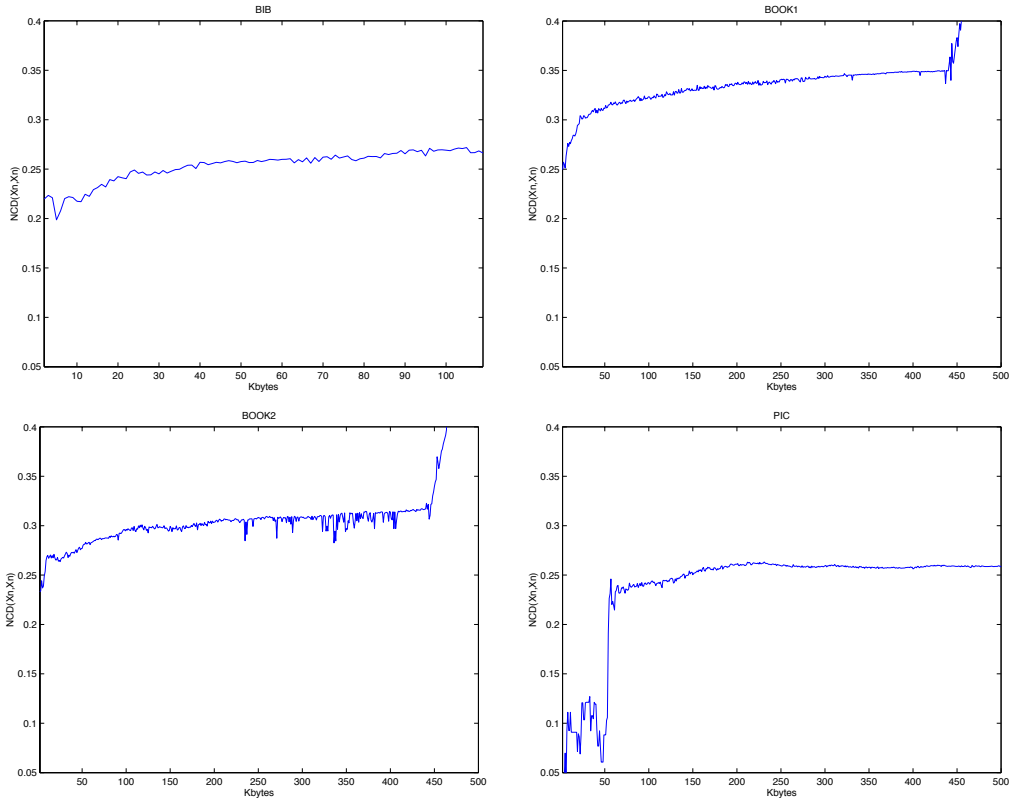
Näistä vertailtiin ensiksi bzip2 pakkaajaa, jolle pitää määrittää lohkon koko

- bzip2 ja lohkon koko
- gzip, liukuva ikkuna ja eteenpäinkatselikkuna



Kuva 1: Normalisoitujen pakkausetäisyyksien pohjalta klusteroituja kirjoja eri kirjoittajilta ja vaihteleva määrä kohinaa lisättynä. Merkinnän ensimmäiset pari merkkiä ovt kirjailijan initials: AC = Agatha Christie, AP = Alexander Pope, EAP = Edgar Allan Poe, WS = William Shakespeare ja NM = Niccolo Machiavelli. [CAO07]

- ppmz, hidasta mutta tehokasta, koska ei mitään rajoitteita materiaalin koolle
- Lopputulos, jos pyritään klusteroimaan NCD:llä pitäisi aina käyttää PPMZ:taa tai vastaavaa pakkaajaa joka ei rajoita tiedostonkokoa



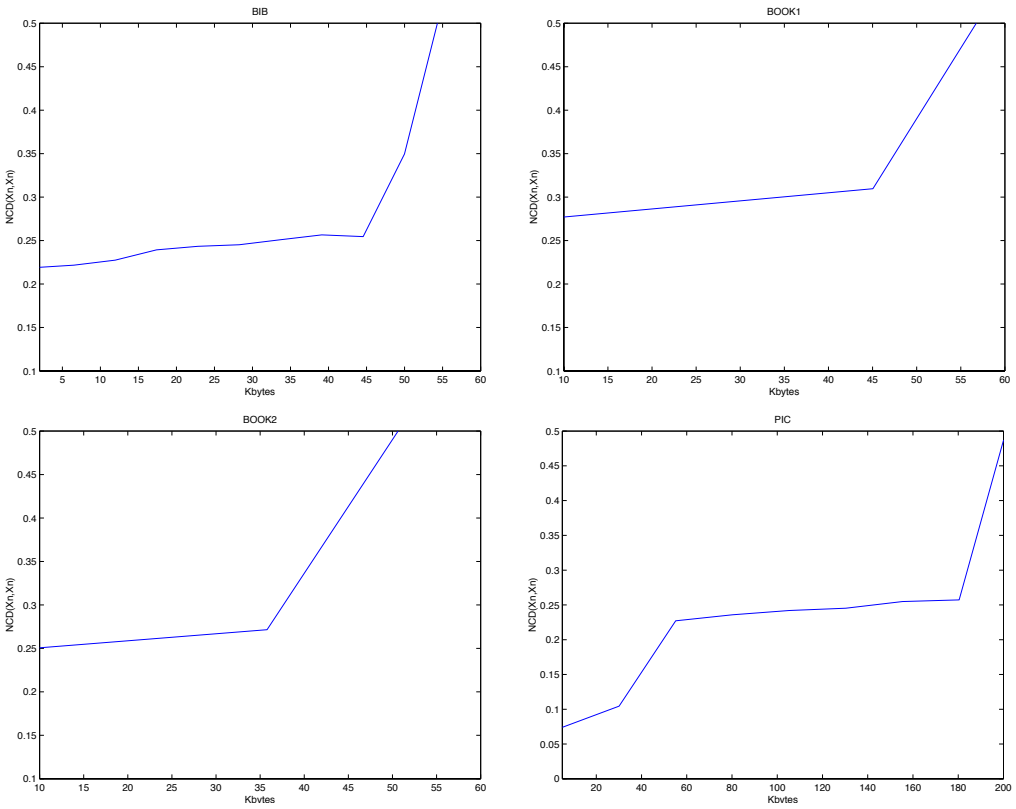
Kuva 2: Normalisoidut pakkausetäisyydet ensimmäisestä n tavusta, neljälle tiedostolle Calgary Corpus -kokoelmasta, käyttäen *bzip2* pakkaajaa asetuksella '–best' [CAO05]

5 Muita samankaltaisuuden metriikoita

Tässä kappaleessa esitellään muita samankaltaisuuden metriikoita, kuten Google-samankaltaisuusetäisyys *(engl. Google Similarity Distance)*

5.1 Google-samankaltaisuusetäisyys

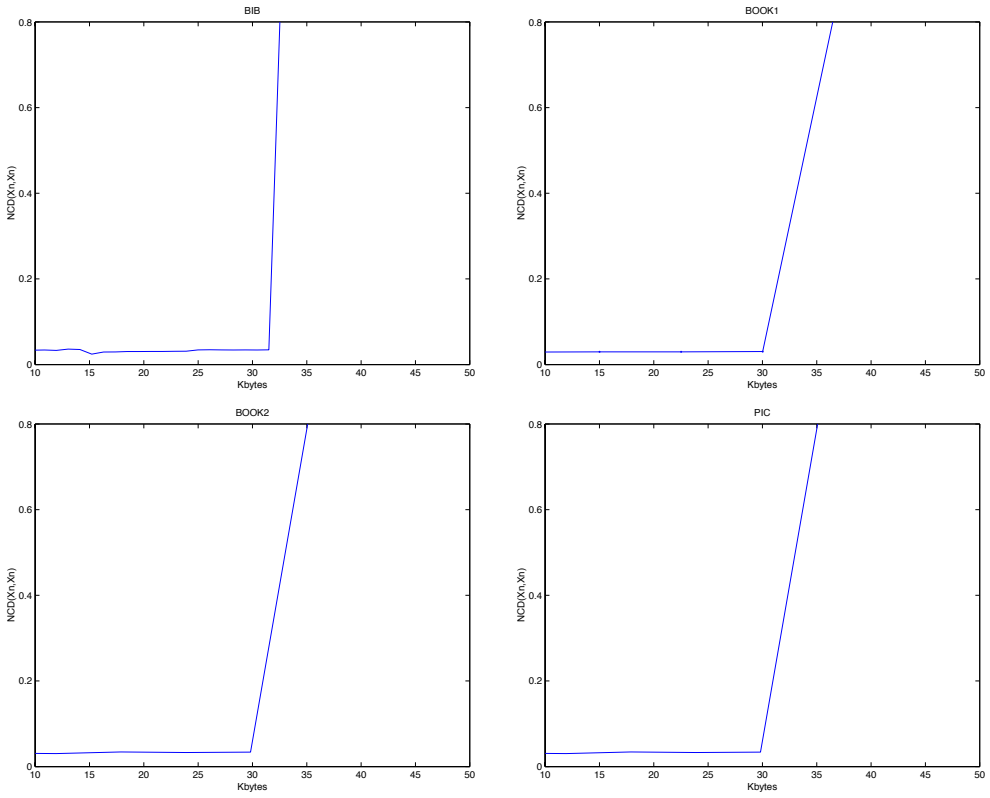
Internetin kasvu on houkutellut miljoonia käyttäjiä luomaan miljardeja internetsivuja, jotka ovat sisällöltään ja piirteiltään hyvin vaihtelevia. Suunnaton tiedon määrä miltei mistä tahansa aiheesta tekee siitä mahdollisen, että ääripäät kumoutuvat ja täten surin osa tai keskiverto on meaningful heikko-



Kuva 3: Normalisoidut pakkausetäisyydet ensimmäisestä n tavusta, neljälle tiedostolle Calgary Corpus -kokoelmasta, käyttäen *bzip2* pakkaajaa asetuksella '–fast' [CAO05]

laatuisena approksimaationa. Täten kehitettiin yleinen metodi hyödyntämään tätä matalalaatuista tietoa, jota saa ilmaiseksi Internetistä. Tämä tietovarasto on kaikille käytettävissä käyttäen mitä tahansa hakukonetta, joka pystyy palauttamaan aggregoidun sivulukumääräarvion, kuten Google.

Google-samankaltaisuusetäisyys (*GSD*) on hyvin vahvasti verrattavissa Normalisoituun pakkausetäisyyteen 2.4, sillä kummatkin algoritmit perustuvat samoihin tekniikoihin, kuten Kolmogorov-kompleksisuuteen (2.1) ja Normalisoituun informaatioetäisyyteen (2.2). Eroavaisuuksia esiintyy siinä, että mitä nämä kaksi algoritmia käyttävät samankaltaisuuden arvioimiseksi. Siinä missä NCD pakkaa ja vertaa sisältöä toisiinsa, niin GSD vertaa asioitten nimiä esiintymistiheyteen. [CV07]

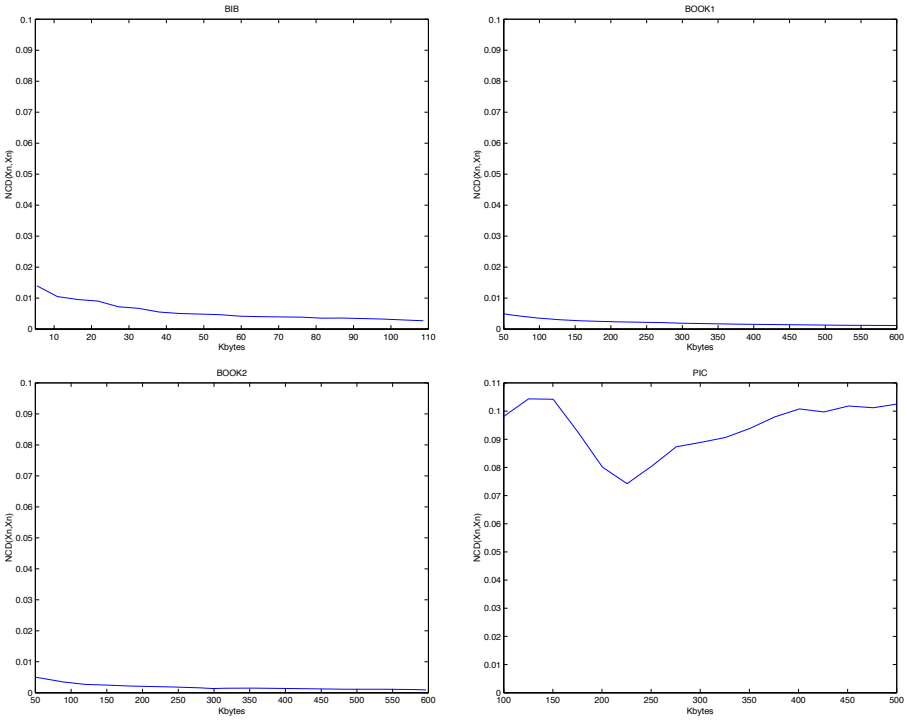


Kuva 4: Normalisoidut pakkausetäisyydet ensimmäisestä n tavusta, neljälle tiedostolle Calgary Corpus -kokoelmasta, käyttäen *gzip* pakkaajaa asetuksella '–best' [CAO05]

5.1.1 Käytännön esimerkki

GSD muodostetaan siten että haetaan ensiksi yhdellä hakutermillä, sitten toisella ja sen jälkeen kummallakin yhdessä. Hakujen lukumäärät vertaillaan keskenään ja normalisoidaan ja tästä saadaan samankaltaisuusetäisyys.

Paperissa [CV07] käytettiin hakusanoja "horse" ja "rider" esimerkkinä ja vuonna 2007 tuloksena oli $NGD(horse, rider) \approx 0.443$. Toistimme kyseisen haun 13.11.2013 ja tuloksena oli $NGD(horse, rider) \approx 0.233$. Arvojen suurta eroa on vaikea selvittää ilman laajempaa tutkimusta, mutta vaikuttavia tekoja on internetin kasvu, sekä epäselvyys mikä on tarkka lukumäärä Googlen indeksoituja sivuja.



Kuva 5: Normalisoidut pakkausetäisyydet ensimmäisestä n tavusta, neljälle tiedostolle Calgary Corpus -kokoelmasta, käyttäen *PPMZ* pakkaajaa [CAO05]

5.1.2 Okeitus hakukoneiden käytölle samankaltiasuudessa

Googlen indeksöityjen internetsivujen määrä on kasvanut suunnattomaksi ja mikä tahansa yleinen hakutermi esintyy miljoonilla internetsivulla. Internetin sisällöntuottajiakin on suunnaton määrä ja voidaan olettaa näiden olevan erittäin kuvaava otanta isosta osasta ihmiskuntaa. Tämän perusteella voidaan argumentoida että, Google-haun frekvenssi on kuvaava approksimaatio todellisista relatiivisistä frekvensseistä yhteiskunnassa. Haun tulos on frekvenssi haun tuottamista osumista ja Googlen indeksoitujen sivujen lukumäärästä. [CV07]

5.1.3 Google-jakauma

Olkoon yksittäisten Google-hakujen joukko S. Jatkossa käytämme yksittäisten ja pareittaisten hakujen joukkoja $\{\{x, y\} : x, y \in S\}$. Olkoon joukko

Googlen indeksöimiä internetsivuja Ω. Joukon Ω koko merkitään $M = |\Omega|$. Oletamme että internetsivut ovat tasajakautuneita, eli todennäköisyys että se palautuu Google-hausta on $\frac{1}{\Omega}$. Ω:n osajoukkoa kutsutaan *tapahtumaksi* ja määrittelemme jokaisen hakuehdon olevan tapahtuma x, $x \subseteq \Omega$

5.1.4 Google-koodi

On olemassa $|S|$ yksittäistä hakuja ja $\binom{|S|}{2}$ pareittain hakua. Määrittelemme $N = \sum_{\{x,y\} \subseteq S} |x \cap y|$. $N \geq M$. Internetsivuilla ei keskivertona ole enempää kuin vakio α hakutermiä, täten $N \leq \alpha M$. Määrittelemme

$$g(x) = g(x, x),\ g(x, y) = L(x \cap y)M/N = |x \cap y|/N. \tag{5}$$

6 "Loppukaneetti"

- Nähty että toimii klusterointiin
- riippumaton tiedosta
- kohinansieto
- pakkaajan valinta

Muita klusteroinnin menetelmiä

- Hieraskinen klusterointi
- k-means klusterointi
- Muita: `http://en.wikipedia.org/wiki/Data_clustering#Clustering_algorithms`

Lähteet

[CAO05] Cebrian, M.l, Alfonseca, M. ja Ortega, A.: *Common pitfalls using the normalized compression distance: What to watch out for in a compressor*. Communications in Information & Systems, 5(4):367–384, 2005.

[CAO07] Cebrian, M., Alfonseca, M. ja Ortega, A.: *The Normalized Compression Distance Is Resistant to Noise*. Information Theory, IEEE Transactions on, 53(5):1895–1900, 2007.

[Cil] Cilibrasi, R.: *CompLearn Toolkit*. URL: `http://complearn.org/`. Luettu 26. marraskuuta 2013.

[CV07] Cilibrasi, R. ja Vitanyi, P.: *The google similarity distance*. Knowledge and Data Engineering, IEEE Transactions on, 19(3):370–383, 2007.

[RV05] R., Cilibrasi ja Vitanyi, P.: *Clustering by Compression*. IEEE Transactions on Information Theory, 51(4):1523–1545, Huhtikuu 2005.

[WBC] Witten, I., Bell, T. ja Cleary, J.: *Calgary Corpus*. URL: `http://corpus.canterbury.ac.nz/descriptions/#calgary`. Luettu 26. marraskuuta 2013.