Normalisoitu pakkausetäisyys: sovelluksia ja variaatioita

Timo Sand

Kandidaatintutkielma
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Helsinki, 26. marraskuuta 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	26. marraskuuta 2013	9

Tiivistelmä — Referat — Abstract

Tähän tulee tutkielman tiivistelmä

Avainsanat — Nyckelord — Keywords

Samankaltaisuus

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Sisältö

1 Johdanto

Kaikki data on luotu samanveroiseksi, mutta jotkut ovat samankaltaisempia kuin toiset. Esitämme tavan jolla esittää tämä samankaltaisuus, käyttäen Cilibrasin ja Vitanyin esittelemää samankaltaisuuden metriikkaa *(engl. similarity metric)*, joka perustuu tiedoston pakkaamiseen. [CV05] 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 kaikissa pareittain vertailuissa ilmaisee samankaltaisuutta hallitsevassa piirteessä. Se on vakaa siinä mielessä, että sen tulokset ovat riippumattomia käytetystä pakkaajasta [CV05]. Pakkaajalla tarkoitetaan pakkausohjelmaa kuten *gzip, ppmz, bzip2.*

Pakkaukseen perustuva samankaltaisuus *(engl. Compression-Based Similarity)* on universaali metriikka, jonka kehittivät Cilibrasi ja Vitanyi [CV05]. 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 palan toisen avulla, mikäli palat 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 pakkausalgoritmiä: tilastollista (PPMZ), Lempel-Ziv -algoritmiin pohjautuvaa hakemistoa (gzip), lohkoperusteista (bzip2) tai erityistä (Gencompress).

Tarkoituksemme on koota yksittäiseen samankaltaisuuden metriikkaan kaikki todelliset etäisyydet; tehokkaat versiot Hammingin etäisyydestä, Euklidisestä etäisyydestä, Lempel-Ziv etäisyydestä ja niin edelleen. Tämän metriikan pitäisi olla niin yleinen, että se toimii, yhtäläisesti ja samanaikaisesti, kaikille aloille: musiikki, teksti, kirjallisuus, ohjelmat, genomit, luonnollisen kielen määritykset. Sen pitäisi pystyä samankaltaisesti havaitsemaan kaikki samankaltaisuudet, joita muut etäisyydet havaitsevat erikseen, palojen välillä.

Kun määrittelemme ryhmän sallittavia etäisyyksia *(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 etäisyyksia seuraavasti: Oletamme kiinnitetyn ohjelmointikielen, joka toimii tutkielman ajan viitekielenä. Tämä ohjelmointikieli voi olla yleinen kieli kuten LISP, Ruby tai se voi myös olla kiinitetty universaali Turingin kone. [CV05, CV07] Valinnalla ei kuitenkaan ole merkitystä, sillä teoria on invariantti ohjelmointikielin muutoksille, kunhan pysytään tehdyssä valinnassa.

Luvussa 3 esittelemme algoritmin käyttökohteita monelta eri alueelta. Aloitamme siitä, miten yleiseti NCD:n 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.

Syvennymme musiikin, kuvantunnistuken ja dokumenttien kategorisoinnin tuloksiin luvun lopussa.

Luvussa 4 esitellään NCD:n kestävyyttä ja ongelmia. Ensiksi esitellään NCD:n kohinansietokykä, 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 niin NCD-algoritmilläkin on ongelmansa. Algoritmissä itsessään ei ole selvää heikkoutta, mutta sen käytössä on otettava pakkaajan valinta huomioon, koska monet suosituista pakkausalgoritmeistä 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, kun mennään siitä yli, niin pakkauksesta tulee huomattavasti tehottomampaa. Esittelemme tuloksia eri pakkausalgoritmien vertailuista ja mikä näistä algoritmeistä on parhaimmaksi havaittu NCD:n kanssa käytet-

täväksi.

NCD ei ole ainut metriikka, jolla voidaan mitata samankaltaisuutta. Internettiä 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 esitellemme tämän sekä muita samankaltaisuuden metriikoita.

2 Normalisoitu Pakkausetäisyys: Mistä se koostuu, miten se toimii?

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 Kolmogorv-kompleksisuutta, $K(x)$ on ajatella se pituutena bitteinä x:n parhaimmin pakatussa muodossa, josta voidaan purkaa x pakkauksenpurkuohjelmalla.

2.2 Normalisoitu informaatioetäisyys

Artikkelissa [CV05] 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

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

Tätä kutsutaan *samankaltaisuuden metriikaksi*, koska tämän on osoitettu [CV05] täyttävän vaatimukset etäisyyden metriikaksi. *NID* ei kuitenkaan ole laskettava tai edes semi-laskettava, koska Turingin määritelmän mukaan Kolmogorov-kompleksisuusei ole laskettava [CV05]. Nimittäjän approksimoin-

ti annetulla pakkaajalla C on $max\{C(x), C(y)\}$. Osoittajan paras approksimaatio on $max\{C(xy), C(yx)\} - \min\{C(x), C(y)\}$ [CV05]. Kun *NID* approksimoidaan oikealla pakkaajalla, saadaa 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, $O(logn)$ termiin saakka:

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

2.4 Normalisoitu Pakkausetäisyys

Normalisoitua versiota *hyväksyttävästä etäisyydestä* $E_c(x, y)$, joka on pakkaajaan C pohjautuva approksimaatio normalisoidusta informaatioetäisyydestä, kutsutaan nimellä *Normalisoitu Pakkausetäisyys (NCD)* [CV05]. Tämä lasketaan seuraavasti

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

NCD on funktioden joukko, joka ottaa argumenteiksi kaksi objektia (esim. tiedostoja tai Googlen hakusanoja) ja tiivistää nämä, erillisinä ja yhdistet-

tyinä. Tämä funktioden 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 algoritmeistä 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.

3 Käyttökohteet

3.1 Klusterointi

3.1.1 Tuloksia

3.2 Kuvantunnnistus

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* vertailtiin NCD:n kanssa [CAO05], selvittääksen mitä heikkouksia, jos mitään, näillä on.

Vertailussa käytettiin Cilibrasin toteuttamaa CompLearn -työkalua [Cil]

Näistä vertailtiin ensiksi bzip2 pakkaajaa, jolle pitää määrittää lohkon koko

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

5 Muita samankaltaisuuden metriikoita

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

5.1 Google samankaltaisuusetäisyys

Google samankaltaisuusetäisyys (*GSD*) on hyvin vahvasti verrattavissa NCD:iin, kummatki pohjautuvat samoihin tekniikoihin, kuten Kolmogorv-kompleksisuus 2.1 ja Normalisoitu informaatioetäisyys 2.2. Eroavaisuus esiintyykin siinä, että mitä nämä kaksi käyttävät samankaltaisuuden arvioimiseksi. Siinä missä NCD pakkaa ja vertaa sisältöä toisiinsa, niin GSD vertaa asioitten nimiä esiintymistiheyteen.

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.

Lähteet

[CAO05] Cebrian, Manuel, Alfonseca, Manuel ja Ortega, Alfonso: *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, ISSN 0018-9448.

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

[CV05] Cilibrasi, Rudi ja Vitanyi, Paul M. B.: *Clustering by Compression*. IEEE Transactions on Information Theory, 51(4):1523–1545, Huhtikuu 2005.

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