Abstrakt

Denne kandidatavhandlingen presenterer anbefalingssystemer, og kollaborative filtreringsmetoder. Kartlegger problematikken i anbefalingssystemer, og hvordan vi kan gjøre anbefalinger basert på kundens tidligere data, likesinnede brukere, og produktets egenskaper. Formålet med denne avhandlingen er å gi ett overblikk over hvordan de systemene fungerer.

Nøkkelord: Anbefalingssystem, kollaborativ filtrering,

Innhold

Innledning

Bakgrunn

I dagens samfunn opplever vi på en daglig basis input av store mengder informasjon. Vi bruker digitale medier for å tilegne oss internasjonale og personlige nyheter. Vi har mulighet til å sjekke opp og tilegne oss informasjon igjennom internett. Handel igjennom nettbutikker øker. Som en følge av å ha all denne informasjonen kan vi oppleve at det blir vanskelig å bearbeide denne informasjonen. Vi kaller dette informasjons-overbelastning. Vi kan definere det som at vi har mye informasjon tilgjengelig at det ikke lenger er mulig å bruke den effektivt. Dette kan lede til flere problem for brukeren. Frustrasjon over å ikke finne relevant informasjon blant irrelevant informasjon. Redsel for å ha oversett relevant informasjon. Manglende evne til å bearbeide mengden informasjon man har tilegnet seg. Forskere har kommet til konklusjonen at hjernen ikke klarer å behandle mye informasjon samtidig. Informasjonsmengden via internett er uten filter og alt kommer på en gang. Dette gjør vi får mye irrelevant input av informasjon. Vi må bla igjennom irrelevant informasjon for å finne relevant informasjon. Vi kan se på det som at informasjon er gjemt i annen informasjon. Anbefalingssystemer velger informasjon til kunden basert på hva den tror kunden vil like. Systemet forsøker å se hva som er mest relevant for ditt søk basert på kontekst, søke-mønster, og hva lignende brukere gjorde. Tanken er at vi vil hjelpe kunden finne riktig informasjon eller riktig vare. Anbefalingssystemer gjør at kunden kan finne hva de vil ha raskere og effektivt. Det kan også hjelpe kunden finne varer, filmer, eller informasjon som han/hun ville oversett. Med personlige anbefalinger kan vi tilby kunden flere relevante valgmuligheter.

Problemstilling

I denne kandidatavhandlingen er den følgende problemstillingen i fokus

Either: *Hvordan anbefaler Netflix videoer til brukerene?*

Or: *Hva er ett anbefalingssystem, og hvordan anbefaler vi element til brukeren?*

Ordliste

Recommender system – Anbefalingssystem. Direkte oversetning fra engelsk til norsk.

Collaborative Filtering – Kollaborativ Filtrering[KF]. Eventuelt Samarbeidsfiltrering. Jeg har valgt å bruke kollaborativ filtrering over samarbeidsfiltrering då det forrige minner mer om originalen.

Anbefalingssystemer

Anbefalingssystem er definert som mjukvare verktøy, og teknikker som automatisk forutser anbefalinger av ett produkt til en kunde. Basert på tidligere oppførsel, forhold/likheter til andre kunder, produktets likheter, kontekst. [1]

La C være ett sett av alle brukere og la S være ett sett av alle mulige tilrådelige produkt. La s og c være delmengder av respektive S og C. $s \in S$ og $c \in C$

La u være en verktøysfunksjon som måler nytten av produkt s til bruker c.

u : C X S → $\mathbb{R}$, der R er det totale ordnede settet.

For hver bruker $c \in C$ vil vi velge $s \in S$ for å maksimalisere u.

Kan utrykkes med følgende funksjon:

$\forall c \in C,\ s'_c = \operatorname{argmax}(u(c,s))_{s \in S}$ [3]

For å få problematikken i fokus er to gode eksempel:

1. Kunde-eposter anbefaler kunden produkter basert på tidligere besøkte produkter, og kjøpemønster.
2. Videostrømmingssider anbefaler videoer basert på kundens interesser, likte filmer, og kunder med lignende seer-mønster.

Målet er å anbefale ett produkt kunden vil dra nytte av.

Inholdsbaserte anbefalinger

Inholdsbaserte anbefalinger baseres på lignende særtrekk i andre elementer. Vi kan sette ulike trekk for å definere elementene. Trekkene varierer, men kan være funksjonelt hva som helst som definerer filmen: ett par eksempel: sjanger, skuespiller, filmunivers, familiefilm, animert, osv. Målet med denne metoden er å bygge en interesse profil for brukeren. Basert på hvilke

elementer kunden har gitt høy vurdering på, så kan vi se på de lignende trekkene i elementene. Om vi kan lage en relevant kundeprofil så har vi en indikator på kundens interessefelt. Det kan brukes til å gjøre filtrerte søk og vise de elementene som brukeren mest sannsynlig har interesse av.

Denne metoden har fordelen at den er fremst basert på den aktive brukerens vurdering. Det er lett å kommunisere til kunden hvorfor de har fått denne anbefalingen da den er knyttet til ett tidligere element kunden så. Det vanlige problemet innen AS med «nytt element» motvirkes igjennom elementet blir anbefalt basert på «trekk» i stedet for vurderinger fra andre brukere. Dette gjør at nye elementer kan bli fortere satt i omløp.

Likhetsfunksjoner Metoder

Euklidsk distanse

Metoden benyttes til å finne likhet eller distanse mellom punkter. Euklidsk distanse er en veldig vanlig metode å bruke for slike beregninger. I matematikken er euklidsk distanse en rett linje mellom to punkt i ett n-dimensjonalt rom, det vil si den korteste distansen mellom punktene. Formelen som håndterer dimensjoner opp til n defineres under.

$$d(x,y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$ [1]

Der n er nummeret på dimensjoner også kjent som nummer på attributter(element). Xk og yk er den k' attributtenes av data objektene av respektive x og y. Om vi reduserer antall dimensjoner til to ser vi at formelen blir Pytagoras læresetning.

$$d(P,Q) = \sqrt{(Q_1 - P_1)^2 + (Q_2 - P_2)^2}$$

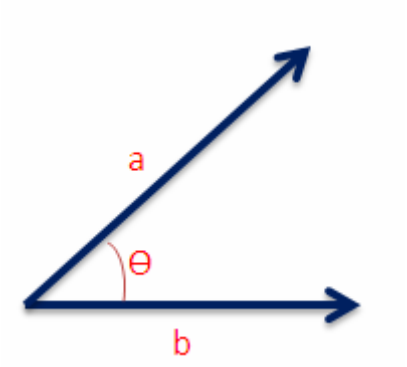
Pearson Korrelasjon

Vi kan også se på en forholdet mellom to objekt ofte kalt en korrelasjon. Ett eksempel på en slik metode er Pearson korrelasjon. Vi kan med en gitt kovarians av datapunkter x og y ∑, og standard avvik σ, nytte av Pearson korrelasjons formel.

$$Pearson(x,y) = \frac{\Sigma(x,y)}{\sigma_x \times \sigma_y}$$

Cosinus likhet

La →a og →b være to vektorer i intervallet 0-180 grader. Et skalarprodukt er en regneoperasjon mellom to vektorer. Meningen er at vi vil dra ut ett reelt tal ut fra to vektorer. Produktet av de to vektorene kan bli skrivet som: $\vec{a} \cdot \vec{b} = |\vec{a}| \cdot |\vec{b}| \cdot cos(\theta)$. $|\vec{a}|$ og $|\vec{b}|$ er absoluttverdiene som vektorenes lengder. Cos defineres som forholdet mellom hosliggende katet og hypotenusen.



Brukerens data kan bli summert og bli presentert som en vektor med høye dimensjoner. Vi kan da beregne vinkelen mellom to brukere ved hjelp av modifisert skalarprodukt-formel. Dataen blir behandlet som en vinkel mellom 0-180 grader. Nummeret på dimensjoner er irrelevant siden dataen blir presentert som en vinkel isteden. Om produktet av formelen er nærmere null betyr det at brukerne ligner hverandre. Produktet av formelen avgir vinkelen mellom de vektorene. Vinkelens grad minker det mer kundene ligner hverandre [2].

$$cos(x,y) = \frac{(x \bullet y)}{||x||||y||}$$

[1]

Kollaborative Filtreringsmetoder (KF)

KF systemer anbefaler produkter til en kunde basert på lignende kjøpemønster i andre kunder eller produkter [2].

Matrise

I KF-systemer har vi to entitetsklasser brukere og produkt. Brukere viser preferanse til hvilke produkt de liker igjennom eksplisitt og implisitte handlinger. Eksplisitte handlinger innebærer en rangering av produktet. Implisitte handlinger innebærer «klikk», lesninger, søk, om elementet er lagt til i handlekurven. Produkter som brukeren har vært i kontakt gir en referanse til en potensiell interesse [2]. Innsamlet data kan effektivt modelleres i en matrise for å representere brukerne. Ett kjapt besøk før bytte til annen side kan også ses på som potensielt dårlig implisitt interesse.

Det finnes flere positive sider ved eksplisitte handlinger. Systemet får en konkret og relevant vurdering ut av brukeren i grad av hvor godt de likte ett element. Systemet lagrer en vurdering, og vurderingen kan også slettes. En rekke implisitte handlinger viser interesse, men uten en vurdering kan vi ikke bedømme graden av interessen. Det er da funksjonelt mer ideelt for anbefalingssystemet å vite en konkret vurdering. Men det er lettere å utføre implisitte handlinger for kunden.

Eksempel 1

I figuren viser vi brukere [A, B, C, D] og deres eksplisitte meninger om ulike filmer. Filmene er rangert fra 1-5, der 5 er høyest rank. Blanke bokser betyr at brukeren ikke har rangert filmen.

	Trainspotting	Memento	Fight Club	Breaking Bad	Archer
A	?		5	4	
B	5		4		

C	4			3	3
D	5			?	4

Figur 1.

Figuren (1) er ett lite utklipp av en fiktiv videostrømmingsdatabase som ett eksempel på matrise-modellering. I realiteten finnes det mye større mengder filmer og brukere. Dette tilsier at mangelen på stemmer vil dramatisk øke i en virkelig database.

Målet med anbefalingssystem er å forutsi de mest relevante tomme boksene. For å motvirke informasjonsoverbelastning presenterer systemet en liste med element som brukeren vil dra nytte av.

Når vi bruker kollaborativ filtrering går vi utfra at tidligere oppførsel vil være lignende fremtidig oppførsel. Dette er ikke alltid sannheten da brukere kan opptre i forskjellige kontekster.

Metoder for klassifisering

Ett anbefalingssystem bruker bruker-data for å finne lignende brukere. For å kunne gjøre en prediksjon behøver vi å finne brukere med lignende interesser. Vi kan bruke ulike algoritmer og metoder for å finne en gruppering av lignende brukere. Slike relasjoner blir kalt bruker «nabolag». Nærliggende brukere tar referanser fra hverandre da vi antar likesinnede kunder ofte tenker likt. Algoritmen kalkulerer likheten/vekten mellom to brukere. Vi danner ett nabolag av lignende brukere å gjør ett gjennomsnitt av hvilke element de likte. Denne informasjonen blir sendt tilbake til en bruker i form av top N forslag for vedkommende. Ved å holde anbefalingene til en bare de top N elementene unngår vi informasjonsoverbelastning. [1,4]

Redusere dimensjoner denoising (Sparsity problem and Curse of Density) // On hold

Når vi omhandler datasett med mange attributter ender vi opp med veldig høye dimensjoner. I tillegg finns det mangler i attributtene da ikke alle er rangert av brukeren. I en «clustering»

setting er det viktig med tetthet og kort distanse mellom punkter. Begge av de punktene blir mindre optimale da vi arbeider med høye dimensjoner. Punktene blir mer unike når de er plasserte i en høy dimensjonal graf. Lengden til nærmeste punkt vokser de mer unike punktene er. Derfor er det ett naturlig valg å redusere dimensjoner.

Prinsipiell komponent analyse er en slik metode. PKA metoden finner mønster i høy dimensjonale datasett.

En vanlig teknikk i datamining er å behandle datasettet før vi analyserer det. Vi setter ett mål for hvilken informasjon vi vil ha deretter tar vi bort irrelevante attributter. Såkalt «denoising» er ett vanlig term i pre-prosessering innen datamining. Tanken er at denne informasjonen er bakgrunns-støy når vi forsøker å finne mer spesifikk informasjon [1,2].

Nabolagsmetode Nærmeste Nabo

Brukerbasert KF

Brukerfokuserte nabolagsanbefalinger metoder danner en vurdering/nummer for ett element basert på lignende brukeres vurdering av det samme elementet [1]. Vi drar nytte av nabolaget til brukeren for å gjøre denne vurderingen.

Elementbasert KF

Elementbaserte KF metoder ser etter lignende vurderinger på lignende elementer. Metoden ser på elementer som den aktive kunden har vurdert og kalkulerer likhet mellom to element basert på vurderinger av andre brukere.

Eksemel 2: Bruker A vurderer om han kommer til å like filmen «Trainspotting». Han gjør sin analyse basert på filmer han allerede har sett. Han legger merke til at brukere som har gitt bra vurdering til «Trainspotting, har også gitt bra vurdering til «Fight Club» og «Breaking Bad».

Bruker A liker Fight Club og Breaking Bad, så han konkluderer med han burde like «Trainspotting».

Upersonlig vs. Personlig KF

Ett av problemene med kollaborativ filtrering er at det behøver eksisterende kundedata. Det såkalte «cold start» problemet refererer til «ny bruker» / «nytt element» problem. For å kunne gjøre prediksjoner behøver vi data vi kan sammenligne med andre brukere/element. Ett nytt element behøver rangeringer for å kunne bli anbefalt til andre. Og en ny kunde behøver rangere noen element for å kunne bli anbefalt nye produkter. Dette er tradisjonelt personlig KF da vi anbefaler topp N elementer for en spesifikk kunde basert på personlige preferanser. Brukerens anbefalinger er også basert på lignende kunder, og de naboene er forskjellige for alle andre kunder.

En måte å komme rundt problemet med «ny kunde» er å gjøre såkalt upersonlig KF. Vis vi tar gjennomsnittet av alle brukere får vi ett resultat som utgjør flertallet av kunder. Vi kan anbefale element som gjennomsnittet av brukere liker. Upersonlig KF gir en ny kunde automatiske goder av anbefalingssystemet. Det er også bedre å kunne anbefale noen film i stedet for ingen filmer. [3]

New item problem ←

Klustering

Kildeliste:

[1] Recommender Systems Handbook – Franceso Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor

[2] Mining Massive Datasets

[3] Recommender Systems(Machine Learning Summer School 2014 @ CMU) av Xavier Amatriain. Presentation slide 15/248. Downloaded (23.02.2017) http://technocalifornia.blogspot.fi/2014/08/introduction-to-recommender-systems-4.html

[4] A Survey of Collaborative Filtering – Xiayuan Su and Taghi M. Khoshgoftaar. 2009. Downloaded(22.02.2017) http://dl.acm.org/citation.cfm?id=1722966

Other topics:

Data Sparsity – Cold start

Ratings matrix: Missing data, Explicit voting, Implicit voting

Diskutere memory based vs model based?