RECOMMENDER SYSTEM VOOR FILMBEOORDELINGEN

Simcha van Helvoort (2747278)
Jeroen Kortus (2798921)

Toegepaste Wiskunde
Fontys Hogescholen

22 februari 2019

Samenvatting

In de wereld van online consumenten proberen bedrijven aanbevelingen te doen aan de hand van gebruikerservaringen. Aan de hand van een beoordeling over deze ervaringen kan vergeleken worden welke consumenten het meest op elkaar lijken. Wanneer een consument een product positief heeft beoordeling, kan dit product ook aan andere consumenten aanbevolen worden die gelijke aankopen een gelijke beoordelingen hebben gegeven. Grote bedrijven in de big data wereld doen deze aanbevelingen aan de hand van een Recommender System. Een paar methodes van deze machine learning toepassing worden in dit artikel behandeld. Naast de theoretische werking wordt ook gekeken naar een praktische werking op de MovieLens database waar gebruikers aanbevelingen krijgen op basis van hun filmbeoordelingen.

I Introductie

Consumenten baseren het kopen van producten op basis van ervaringen en meningen van andere mensen. Bij online aankopen of beslissen welke film er die avond wordt gekeken vinden mensen het prettig om af te gaan op aanbevelingen van vrienden of familie [1]. Rond deze tijd worden veel aanbevelingen op internet gedaan door grote bedrijven zoals Netflix, Amazon en Google om mensen tegemoet te komen met deze behoefte. Dit zijn weliswaar geen aanbevelingen van vrienden, maar aanbevelingen van klanten die dezelfde service gebruiken en bijvoorbeeld dezelfde films leuk vinden; producten hebben gekocht of nieuws hebben gelezen. Niet alleen doen bedrijven dit om hun klanten tegemoet te komen, maar hoe meer/betere aanbevelingen een gebruiker krijgt, hoe meer de gebruiker die service zal blijven gebruiken.

Dit document is geschreven voor de cursus machine learning. Een methode voor Recommender Systems wordt uitgelegd met theoretisch en toepasbare toelichting. In paragraaf ii worden Content Based Recommender Systems (CB), Collaborative Filtering (CF) behandeld. CB, CF en combinaties daarvan worden veel gebruikt, maar CF is toch wel het populairst [2]. Met CB is getest, aan de hand van de MovieLens database[1] hoe goed recommender systems werken. Achteraf wordt ook besproken wat aan te raden voor vervolg onderzoek.

II Methode

II.a Probleem formulering

Het idee van recommender system is dat een beoordeling van een gebruiker wordt voorspeld. De beoordelingen voor alle producten per gebruiker worden in matrix $\boldsymbol{Y}$ gezet (alle matrices worden dik gedrukt geschreven met een hoofdletter. Alle vectoren ook dik gedrukt, maar dan een kleine letter). De producten die hier behandeld worden zijn films en zullen verder ook zo genoemd worden. Y_{ij} is de beoordeling van de film i en gebruiker

[1] http://files.grouplens.org/datasets/movielens/ml-latest-small.zip

j. $\boldsymbol{Y}$ is een $n_m \times n_u$ matrix waarbij n_m en n_u respectievelijk staan voor het aantal films en aantal gebruikers. Dit betekent wel dat er waarden ontbreken in die matrix, omdat niet elke gebruiker elke film heeft beoordeeld. $\hat{\boldsymbol{Y}}$ is een matrix van gelijke grote waarin de voorspelde beoordelingen staan. Dit wordt gedaan met lineaire regressie. $Y_{ij} \in \{1, 2, 3, 4, 5\}$ terwijl $\hat{Y}_{ij} \in [1, 5]$. Dit komt omdat de voorspelling niet perse discreet hoeft te zijn aangezien op basis hiervan een aanbeveling wordt gemaakt. $r(i, j)$ is 1 als gebruiker j film i heeft gezien en 0 als dat niet zo is. Bij zowel CB en CF wordt er gebruik gemaakt van film features ($\boldsymbol{x}$) en gebruikersprofielen ($\boldsymbol{\theta}$). De lineaire regressie voorspelling wordt uitgevoerd met verg. (1)[Ng].

$$\hat{Y}_{ij} = (\boldsymbol{\theta}^{(j)})^T \boldsymbol{x}^{(i)} \tag{1}$$

Hierin is $(\boldsymbol{\theta}^{(j)})^T$ de getransponeerde gebruikers vector voor gebruiker j en $\boldsymbol{x}^{(i)}$ de feature vector voor film i. In tegendeel tot normale lineaire regressie en de lineaire regressie voorgesteld heeft de lineaire regressie voor CB en CF een aparte parameters en vectoren voor zowel de gebruikers als de films.

II.B Content Based Recommender System

De essentie van CB is dat de films geanalyseerd worden en daar features op bepaald worden. Deze features zijn voordat het model wordt vastgesteld dus al bepaald. Dit is waarschijnlijk de reden dat CF populairder is dan CB (waarom dat zo is wordt later behandeld). Op basis van de eigenschappen van de film wordt er gekeken wat de gebruiker interessant vindt. Het aanbevelingsproces bestaat dan eigenlijk alleen maar uit films vinden die het meest lijken op de films die de gebruiker leuk vindt[3]. Als de gebruikersprofiel sterk overeenkomt met de daadwerkelijke interesse van de gebruiker zal dit een enorm voordeel zijn voor de effectiviteit van het model.

Om CB te laten werken is het nodig om je features per film te weten ($\boldsymbol{x}^{(i)}$) en daarmee bepaal je de gebruiker parameters per gebruiker (($\boldsymbol{\theta}^{(j)}$). Dit kan gedaan worden door de kost-functie voorgesteld in verg. (2) te optimaliseren. Deze vergelijking moet geminimaliseerd worden per gebruiker.

$$\min_{\boldsymbol{\theta}^{(j)}} \quad J(\boldsymbol{\theta}^{(j)}) + R(\boldsymbol{\theta}^{(j)}) \tag{2}$$

$$J(\boldsymbol{\theta}^{(j)}) = \frac{1}{2} \sum_{i:r(i,j)=1}^{n_u} \left((\boldsymbol{\theta}^{(j)})^T \boldsymbol{x}^{(i)} - Y_{i,j}\right)^2$$

$$R(\boldsymbol{\theta}^{(j)}) = \frac{\lambda}{2} \sum_{k=1}^{n} (\theta_k^{(j)})^2$$

Hierin is $J(\boldsymbol{\theta}^{(j)})$ de Sum of Squared Error van de voorspellingen en λ de regularization parameter die controle heeft over de fitting parameter $R(\boldsymbol{\theta}^{(j)})$. Dit voorkomt overfitting doordat nu afgewogen wordt tussen twee dingen:

- SSE verlagen
- Individuele waarde van $\boldsymbol{\theta}$ verhogen

Als de SSE verlaagt kan worden, maar daardoor $\boldsymbol{\theta}$ te groot wordt, dan zal deze keuze niet gemaakt worden door het algoritme.

II.C Collaborative filtering

Een nadeel van CB is het feit dat de methode afhankelijk is van een analyse over je product of film. De kwaliteit van het model is gelimiteerd tot het aantal en soort features die beschikbaar zijn. Een voordeel van CF is dat je niet meer afhankelijk bent van de analyses en de data die beschikbaar zijn vanuit een andere bron. Het is namelijk mogelijk om aan de hand van enquêtes gebruikersprofielen aan te maken. Met deze gebruikersprofielen (die je zelf op kan stellen en elke stukje informatie van de gebruiker op kan vragen) en de beoordelingen van de gebruiker kan vastgesteld worden wat de features per film is. De vergelijkingen die hierbij horen lijken veel op verg. (2), maar dan wordt $\boldsymbol{x}^{(i)}$ de variabele die gewijzigd kan worden. Dit leidt tot verg. (3) waarin wederom de squared error en de fitting parameter geminimaliseerd wordt per film. Net als bij CB moet deze kostfunctie ook per film geminimaliseerd worden.

$$\min_{\boldsymbol{\theta}^{(j)}} \quad J(\boldsymbol{x}^{(i)}) + R(\boldsymbol{x}^{(i)}) \tag{3}$$

$$J(\boldsymbol{x}^{(i)}) = \frac{1}{2} \sum_{i:r(i,j)=1}^{n_u} \left((\boldsymbol{\theta}^{(j)})^T \boldsymbol{x}^{(i)} - Y_{i,j}\right)^2$$

$$R(\boldsymbol{x}^{(i)}) = \frac{\lambda}{2} \sum_{k=1}^{n} (x_k^{(i)})^2$$

II.D Combinatie van CB en CF

Na allebei de methodes zijn achteraf beide parameters beschikbaar. Het is daardoor mogelijk om de parameters nauwkeurig bij te stellen. Bij CB zouden de features per film getuned kunnen worden en bij CF kunnen de features per gebruiker achteraf getuned worden. Dit zou weer herhaald kunnen worden totdat

III Implimentatie

III.A MovieLens dataset

Bij de implimentatie van het recommender system is gebruik gemaakt van de MovieLens dataset. Deze dataset is samengesteld door een onderzoeks groep aan de University of Minnesota. Het bevat 100.000 beoordelingen over 9125 films gegeven door 671 gebruikers. De beoordelingen lopen van 0.5 tot 5 met een interval van 0.5. Voor iedere film zijn de genres gegeven. In de dataset zijn 20 verschillende genres.

III.B Content based recommender system

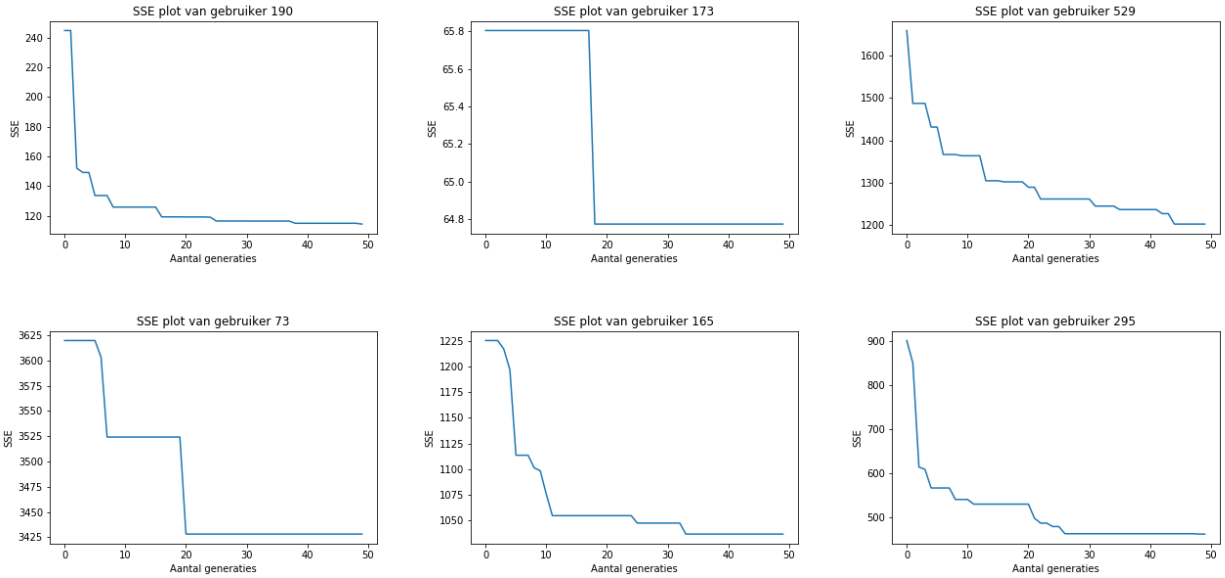
Voor het maken van een recommender system voor de MovieLens dataset is gebruik gemaakt van CB recommender system. Hierbij wordt er aan de hand van features van de film ($\boldsymbol{x}^{(i)}$) de $\boldsymbol{\theta}^{(j)}$ voor de gebruikers vastgesteld. De features van de films bestaat in dit geval uit de genres. Hierbij wordt 1 punt verdeeld over het aantal genres dat de film bevat. Een film met drie genres zal dus een $\boldsymbol{x}^{(i)}$ hebben die bestaat uit een nullen met op de plaatsen van de genres 0.33. $\boldsymbol{\theta}^{(j)}$ zal dus betekenen hoe belangrijk een genre is voor een bepaalde gebruiker. De waardes van $\boldsymbol{\theta}^{(j)}$ liggen tussen 0 en 5.

III.C Optimalisatie met genetic algorithm

Het optimaliseren van $\boldsymbol{\theta}^{(j)}$ wordt gedaan met een Genetic Algorithm (GA). Het GA gebruikt tijdens het testen van CB een populatie van 5 toegelaten oplossingen. De populatie bestaat uit twee elites, twee crossovers en een immigrant. De elites zijn de twee oplossingen met de beste waarde voor de doelfunctie. De doelfunctie minimaliseerd de SSE voor een grbuiker. Een crossover is een combinatie van de twee elites, waarbij de elites gesplit zijn op een random punt in de vector en samengevoegd zijn. Hierbij bestaat de eerste crossover uit het eerste deel van elite 1 en het tweede deel van elite 2. Voor de tweede crossover is dit precies andersom. De immigrant is een random gekozen oplossing met waardes tussen 0 en 5. Het genetic algorithm loopt per gebruiker voor 50 generaties. Na iedere generatie worden de twee beste oplossingen meegenomen als elites. De crossovers worden vastgesteld en er wordt een immigrant toegevoegd aan de populatie. Aan het einde van de laatste generatie wordt de beste oplossing uit de huidige populatie de gebruikersparameter $\boldsymbol{\theta}^{(j)}$. Dit proces wordt voor iedere gebruiker uitgevoerd.

IV Resultaten

Het uiteindelijke resultaat zijn alle geoptimaliseerde $\boldsymbol{\theta}^{(j)}$voor alle gebruikers. Met deze thetas kunnen voorspellingen worden gedaan voor films die gebruiker j nog niet gezien heeft. Hoe goed het optimalisatie proces is gegaan kan gevisualiseerd worden door de Sum of Squared Errors (SSE) in een grafiek te zetten tegenover het aantal generaties. Een aantal voorbeelden staan uitgewerkt in fig. 1. Let op: de y-as heeft wel per gebruiker wel een andere schaal. Sommige gebruikers hebben meer films beoordeeld. Dit zorgt voor een grotere test- en trainingset. Als gevolg hiervan een hogere SSE. Een nadeel hiervan is dat niet bekend is hoe goed het model uiteindelijk geoptimaliseerd is. De doelfunctie heeft een ander optimum voor elke gebruiker.



Figuur 1: Sum of Squared Errors geplot voor 6 gebruikers. Dit laat de vooruitgang van het optimalisatie model zien.

Metriek	Waarde
minimum	0.2
1^{e} percentiel	0.8
mediaan	1.0
3^{de} percentiel	1.1
maximum	3.0

Tabel 1: 5 number summary van de errors.

Een manier om het model te evalueren is door naar de afwijkingen te kijken. In tabel 1 staat een 5 number summary voor de errors. Wat interessant is om op te merken is dat het minimum 0.2 is. Dit betekent eigenlijk dat elke voorspelling nooit helemaal correct is geweest, maar op zijn minst 0.2 punten afwijkt van de daadwerkelijke score. Het feit dat het maximum op 3 punten ligt, en de mediaan op 1, verteld dat het model niet naar behoren werkt.

Een andere methode om te testen is door gebruik te maken van een confussion matrix. Deze staan weegegeven in bijlage A. Hierin is te zien dat (na het afronden van de voorspellingen) er soms wel iets goeds wordt voorspeld. Bovendien zitten de fouten die gemaakt zijn over het algemeen dichtbij de daadwerkelijke beoordeling. Daarmee kan gezegd worden dat dit model het beter doet dan willekeurig voorspellen of altijd dezelfde waarde geven.

V Conclusie

Consumenten kopen producten op basis van aanbevelingen of adviezen van anderen. Door beoor-

delingen van consumenten met elkaar te vergelijken kan een internet service de consument adviseren. Tegenwoordig zitten hier algoritmes achter die dat snel doen voor veel gebruikers.

Content Based Recommender System (CB) is een van die algoritmes, deze is getest tijdens dit onderzoek. CB vergelijkt de beoordelingen van gebruikers met de features van de producten (in dit onderzoek zijn dat films van de MovieLens database). Daarmee stelt CB een gebruikersprofiel op die beschrijft hoe belangrijk een gebruiker een bepaalde feature vindt.

Het optimaliseren van gebruikersprofielen is gedaan met een Genetic Algorithm. Deze optimalisatie verbetert het model drastisch in vergelijking met willekeurige gokken, maar het is nog niet optimaal. Zo is de mediaan van de fouten 1.0 punten afwijking van de daadwerkelijke beoordeling. De maximale fout is 3.0 punten. Dit model kan dus nog flink verbetert worden.

VI Discussie & aanbeveling

Voor de optimalisatie van $\boldsymbol{\theta}^{(j)}$wordt gebruik gemaakt van een genetic algorithm. Voor dit optimalisatie probleem lijkt het genetic algorithm niet efficiënt te zijn. Een betere methode zou gradient decent zijn. Deze hangt niet af van de willekeurig gekozen immigrant, maar gaat op basis van de gradient richting het optimale punt.

De gebruikte dataset voor de implementatie van dit recommender system heeft verschillende versies. De gebruikte versie is de kleinste variant met maar 100.000 datapunten. De dataset is ook beschikbaar in een één miljoen en een 20 miljoen versie. Deze zouden mogelijk betere uitkomsten kunnen geven, omdat deze meer datapunten hebben.

Er is gebruik gemaakt van een CB recommender system. Een betere methode zou mogelijk zijn om een combinatie tussen CB en CF recommender systems te gebruiken. Hiermee worden de features van de film aangepast op basis van de voorkeuren van de gebruikers. In dit onderzoek zijn de waardes voor de genres gelijk gehouden. Wanneer een film drie genres bevat, tellen deze even zwaar mee. Door de combinatie tussen CB en CF toe te passen veranderd dit dus, wat mogelijk een realistischer beeld geeft. Daarnaast is het mogelijk dat er een genre wordt toegevoegd die eerder niet aan die film was toegewezen.

Referenties

[1] Kim, B.-D. and Kim, S.-O. (2001). A new recommender system to combine content-based and collaborative filtering systems. *Journal of Database Marketing & Customer Strategy Management*, 8(3):244–252.

[2] Li, G., Zhang, Z., Wang, L., Chen, Q., and Pan, J. (2017). One-class collaborative filtering based on rating prediction and ranking prediction. *Knowledge-Based Systems*, 124:46–54.

[3] Lops, P., De Gemmis, M., and Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer.

[Ng] Ng, A. Recommender systems, CS229 lecture notes.

A Confusion matrix

In deze bijlage worden de confusion matrix van 4 willekeurig gekozen gebruikers weergegeven.

19	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
0.5	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	2	0	1	0	0
1.5	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	3	1	0	0
2.5	0	0	0	0	0	0	0	0	0	0
3	0	0	0	2	2	22	8	4	0	0
3.5	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	3	8	7	4	0	0
4.5	0	0	0	0	0	0	0	0	0	0
5	0	0	0	1	3	9	3	1	0	0

Tabel 2: Confusion matrix van gebruiker 19

15	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
0.5	0	0	5	2	14	0	0	0	0	0
1	0	1	8	8	27	8	2	0	0	0
1.5	0	3	4	5	11	4	0	0	0	0
2	0	0	7	6	20	3	1	0	0	0
2.5	0	1	8	1	14	6	1	0	0	0
3	0	2	13	6	32	9	1	0	0	0
3.5	0	1	4	4	12	6	1	0	0	0
4	0	1	4	6	24	9	1	0	0	0
4.5	0	0	2	1	8	2	0	0	0	0
5	0	1	3	5	10	1	0	0	0	0

Tabel 3: Confusion matrix van gebruiker 15

656	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
0.5	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
1.5	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
2.5	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	1	0	0	0
3.5	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	1	0	2	1	2
4.5	0	0	0	0	0	0	0	0	0	0
5	0	0	0	1	1	0	3	5	2	6

Tabel 4: Confusion matrix voor gebruiker 656

339	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
0.5	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
1.5	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1	0	0	0
2.5	0	0	0	0	0	0	1	0	0	0
3	0	0	0	0	0	0	1	1	0	0
3.5	0	0	0	0	0	0	2	0	1	0
4	0	0	0	0	0	3	2	2	0	0
4.5	0	0	0	0	0	0	3	0	0	0
5	0	0	0	0	0	0	4	0	0	0

Tabel 5: Confusion matrix gebruiker 339