# Graaf voor suggesties

## Om muziekvoorkeuren voor verschillende gebruikers te kunnen berekenen, wordt een graaf gebruikt. In deze graaf worden de relaties tussen verschillende muziekgenres weergeven, eerst met gegevens die een gemiddelde muzieksmaak vertegenwoordigen, maar na meer gebruik van het systeem past deze graaf zich steeds meer aan aan de gebruiker.

## Initialisatie van de graaf

De graaf wordt op twee manieren geïnitialiseerd: via de rating data uit de Yahoo dataset en via de voorkeuren die de gebruiker zelf doorgeeft.

### Via Yahoo

In de Yahoo dataset zit een klein gedeelte van de waarderingen die Yahoo gebruikers hebben gegeven aan muzieknummers. Elke gebruiker uit de dataset heeft minstens 10 songs een waardering toegekend. De waardering bestaat uit een getal tussen de 1 en 5. Een 1 geeft hierbij een lage waardering aan, een 5 juist een hele hoge.

De songs zelf hebben willekeurige nummers gekregen, waardoor deze niet zijn te herleiden naar een daadwerkelijke artiest en nummer of MBID. Wel is er een koppeltabel toegevoegd waarin de songs zijn gekoppeld aan een bepaald genre. Deze informatie kan worden gebruikt om de genres uit de graaf te initialiseren.

Hiervoor is eerst een gedeelte van de data (30 miljoen waarderingen) uit de tekstbestanden van Yahoo via een Java programma ingelezen. Deze data is vervolgens in een database gezet. In de database worden de volgende gegevens opgeslagen:

* Genre id
* Genre naam
* Het aantal gebruikers dat per genre heeft gestemd
* De waarderingen van alle gebruikers voor een bepaald genre

De gemiddelde waardering wordt hiermee per genre bepaalt door het algoritme dat IMBD gebruikt:

(v ÷ (v+m)) × R + (m ÷ (v+m)) × C

Met:

R = gemiddelde van het genre, ofwel alle waarderingen gedeeld door het aantal gebruikers  
 v = het aantal gebruikers dat heeft gestemd  
 m = minimum aantal stemmen die nodig zijn (wij hebben gekozen voor 1500)  
 C = de gemiddelde waardering van alle genres bij elkaar

Door het bepalen van het gemiddelde op deze manier, worden genres waar maar een beperkt aantal mensen op stemmen (vaak de wat alternatievere) weggelaten. Ook krijgen de genres die juist door heel veel mensen worden beoordeeld, relatief gezien een hogere beoordeling.

De gemiddelde waardering die hiermee wordt berekend, is uiteraard een score tussen de 1 en 5. Deze score wordt met de factor 30 gewogen en vervolgens afgerond. Van al deze gemiddelde waarderingen wordt het gemiddelde genomen. Dit wordt vervolgens van de score afgetrokken.

De wegingen die daaruit komen (ongeveer variërend tussen de +15 en -30) worden opgeteld bij de nodes in de graaf, welke een standaard waarde van 100 hebben.

### Via de gebruikerswaarderingen

Na de registratie bij het systeem, komt de gebruiker als eerste een pagina tegen waarop hij gevraagd wordt om zelf aan enkele vooraf gedefinieerde genres een score toe te kennen. Deze scores vormen naast de geïmporteerde data uit de Yahoo-dataset een zeer algemene indruk van wat een gebruiker graag zou luisteren.

De scores die een gebruiker kan geven zijn op een schaal van 1 tot 5: een 1 staat voor niet leuk, een 3 voor neutraal en een 5 voor zeer leuk. Bij het opbouwen van de graaf worden deze omgezet in een percentage tussen de 80 en 120 (80 voor een score van 1, 120 voor een score van 5). De waarde van de node waarop deze van toepassing is wordt aangepast naar dit percentage ten opzichte van de originele waarde.

|  |  |
| --- | --- |
| Figuur 1: standaardgraaf | Figuur 2: graaf met gebruikersvoorkeuren |

In dit voorbeeld is te zien hoe dat in de praktijk toegepast wordt: in figuur 1 is de situatie voor het toepassen van de scores weergeven, in figuur 2 is er een score van 5 gegeven aan Genre 4, een score van 1 aan Genre 1 en aan de overige genres een score van 3.

## Verbeteren van de graaf

Om de gebruiker meer invloed te geven op de suggesties die door de graaf gegeven worden, wordt de graaf aangepast aan user input.

### Positieve waarderingen

Bij een positieve waardering voor een suggestie, worden er 3 dingen aangepast in de graaf. Als eerste worden de scores van alle nodes die corresponderen aan de suggestie verhoogd.

Vervolgens wordt er naar de edges gekeken die aan de nodes gekoppeld zijn. Alle edges die aan een veranderde node vastzitten worden verkort.

Als laatste wordt er gekeken of er directe verbindingen tussen alle nodes die bij deze suggestie horen zijn. Als een verbinding nog niet bestaat, wordt deze aangemaakt en krijgt een score van 90% van de standaardlengte 50.

|  |  |
| --- | --- |
| Figuur 3: voorbeeld van positieve waardering |  |

In figuur 3 staat een tekening van de graaf, nadat er een stem is uitgebracht op een suggestie die gekoppeld was aan Genre2 en aan Genre3.

Zoals er te zien is, zijn de scores op Genre2 en Genre3 verhoogd met 10%, en is er een nieuwe edge ontstaan tussen deze 2 nodes met lengte 45.

### Negatieve waarderingen

Bij een negatieve waardering voor een suggestie, worden er net als bij een positieve waardering 2 dingen aangepast.

De score op een node wordt in dit geval verlaagd, en de edges tussen de nodes die bij deze suggestie horen worden langer. Waar er echter bij een positieve waardering ook edges toegevoegd werden, gebeurd dat hier niet, maar ook worden er geen edges verwijderd.

Hiervoor is gekozen omdat het niet erg zinnig is om een negatieve relatie uit te drukken in een graaf waar gezocht wordt op positieve relaties, en ook omdat het weghalen van edges een veel grotere impact heeft op de structuur van de graaf dan het toevoegen ervan.

|  |  |
| --- | --- |
| Figuur : voorbeeld van een negatieve waardering |  |

In figuur 4 is te zien wat er gebeurd ten opzichte van de situatie in figuur 1 wanneer er een negatieve waardering wordt gegeven aan een suggestie die gekoppeld is aan de nodes Genre2 en Genre3. De nodes krijgen een lagere score en enkele edges worden langer.

### Het verwijderen van suggesties

Suggesties kunnen ook worden verwijderd uit het systeem. Als een suggestie verwijderd wordt, worden er 2 acties ondernomen. Zo wordt er als eerste voor voor gezorgd dat deze suggestie niet meer terug komt in de zoekresultaten.

Als tweede worden er ook in dit geval acties ondernomen op de graaf. Een verwijderde suggestie wordt in de database opgeslagen als een negatieve waardering (een -2 om precies te zijn), en dat vertaald zich in de graaf naar exact dezelfde actie als dat er een negatieve waardering gegeven zou zijn.

## Het geven van de suggesties

Nu er duidelijk is hoe de graaf opgebouwd wordt en hoe gegevens daarin verwerkt worden, is het laatste punt wat de graaf kan doen het geven van genresuggesties. In de volgende stukjes worden de 3 algoritmes besproken die dit mogelijk maken. Verder wordt er ook nog een algoritme besproken dat losstaat van het geven van suggesties. De algoritmes worden besproken in de volgorde waarop ze aan het systeem zijn toegevoegd: het oudste algoritme komt eerst, gevolgd door latere.

### Het eerste algoritme: het bouwen van een MST

Het algortime om een Minimum spanning tree door de boom te maken, heeft het meeste weg van Prim’s algorithm. Er wordt een startpunt gekozen, namelijk de node met de hoogste score, en van daaruit wordt er gekeken naar welke node er het ‘beste’ toegevoegd kan worden aan de boom. Hiervoor wordt er gekeken naar de scores van alle nodes die met minstens 1 edge aan de boom verbonden is, waar de lengte van de edge als ‘straf’ wordt afgetrokken. Om te kunnen vergelijken met andere algoritmen, is er besloten om deze straf te compenseren door de gemiddelde lengte van alle edges weer op te tellen bij de score.

Na x iteraties van dit algoritme, is het resultaat een lijst van x+1 nodes die voldoen aan de voorwaarden die gesteld zijn.

### Het tweede algoritme: sorteren op score

Het eerste algoritme is de meest simpele: van alle nodes worden de scores opgevraagd, en de nodes worden hierop aflopend gesorteerd. De eerste x nodes uit de resulterende lijst worden teruggeven (waarbij x een gewenst aantal resultaten is).

### Het derde algoritme: de combinatie van beiden

Beide eerder besproken algoritmes hebben hun eigen nadelen: het tweede algoritme verandert het resultaat snel na kleine wijzigingen en geeft een gespreid resultaat, terwijl het eerste algoritme heel erg op elkaar lijkende suggesties geeft en wat minder snel veranderd na veranderingen aan de graaf.

Het derde algoritme is een poging om deze nadelen een beetje uit te middelen en als voordeel in te zetten: beide algoritmes wordt gevraagd om een suggestie voor een nieuwe node, en er wordt vervolgens gekeken naar welke methode de beste suggestie opleverde.

### Het vierde algoritme: de laagste waarderingen

Het vierde algoritme is een rare eend in de bijt. Dit algoritme werkt net als het tweede algoritme, maar dan andersom. De teruggeven van de minst gewaardeerde nodes wordt gebruikt voor de statistiekenpagina en is niet van invloed op de suggesties die gemaakt worden.

### Het geven van een suggestie: een voorbeeld

In de praktijk wordt het geven van de best scorende genres afgehandeld door algoritme 3. Dit komt neer op algoritme 1 waar aanpassingen aan gemaakt zijn om algoritme 2 ook toe te kunnen passen.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Figuur : voorbeeldgraaf | Tabel : uitkomst algoritme 3   |  |  |  | | --- | --- | --- | | Rang | Naam | Algoritme | | 1 | Genre4 | 3 / 2 | | 2 | Genre 7 | 1 | | 3 | Genre 2 | 2 | | 4 | Genre 3 | 2 | | 5 | Genre 8 | 2 | | 6 | Genre 6 | 1 | | 7 | Genre 5 | 2 | | 8 | Genre 9 | 2 | | 9 | Genre 1 | 2 | |

In dit voorbeeld is te zien in welke volgorde de nodes in deze graaf gesorteerd worden. De eerste stap wordt vastgelegd door het derde algoritme (wat erop neerkomt dat die de oplossing van het tweede algoritme forceert). Vervolgens laat het derde algoritme de andere 2 algoritmes een suggestie voor de volgende node doen, wat als 2e node Genre oplevert die aangedragen is door het eerste algoritme.

In dit voorbeeld is er doorgegaan met het blijven uitvoeren van algoritme 3 totdat alle nodes gesorteerd zijn, in de praktijk zal hier een limiet op worden toegepast: van een lijst van honderden nodes zouden anders toch slechts enkelen gebruikt worden.