Faculteit
Ingenieurswetenschappen

Departement
Elektrotechniek – ESAT

KATHOLIEKE
UNIVERSITEIT
LEUVEN

Muziek modellering en profilering voor hitbeoordeling

Eindwerk voorgedragen tot het behalen van het diploma van Burgerlijk werktuigkundig-elektrotechnisch ingenieur, richting elektrotechniek, optie multimedia en signaalverwerking

Toon Raes

Promotor:
Prof. dr. Marc Moonen
Prof. dr. Johan Suykens
Dagelijkse begeleiding:
ir. Samuel Corvelyn
dr. Kristiaan Pelckmans
ir. Toon van Waterschoot

2007 – 2008

Voorwoord

In de eerste plaats gaat mijn dank uit naar mijn begeleiders dr. Kristiaan Pelckmans, ir. Toon van Waterschoot en ir. Samuel Corvelyn, voor hun deskundige hulp bij het maken van deze thesis. Tevens verdienen natuurlijk ook mijn promotors prof. Marc Moonen en Johan Suykens een woord van dank voor hun hulp en het mogelijk maken van dit eindwerk.

Verder is een thesis een werk van vele mensen en ik wil hen allen bedanken voor de geboden hulp. Iedereen vermelden zou een aparte appendix vereisen maar jullie hulp is ten zeerste geapprecieerd. Enkele personen kan ik natuurlijk niet nalaten om speciaal te vermelden en dat zijn mijn ouders en mijn familie voor al hun steun. Bedankt. En hoe zou deze thesis compleet zijn zonder mijn vriedin Saskia, dankjewel schat.

Toon Raes

Abstract

Het opzet van deze thesis is het onderzoeken van de modelleerbaarheid van hitmuziek. Om dit onderzoek uit te voeren werd uitgegaan van een structuur met drie componenten; een database, een featureextractie en een classificatiemodel.

Allereerst werd er een database gecreëerd door het digitaliseren van cd's afkomstig uit de bibliotheek. Op deze manier werd een primaire database bekomen van 9026 liedjes, afkomstig van 450 cd's. Deze database werd vervolgens gerefereerd naar de UK singles chart Top 100, waardoor de classificatiedatabase bekomen werd van 1861 liedjes afkomstig van 1010 artiesten. De database werd tevens zo opgezet dat deze een zo random mogelijke sampling biedt van de hitlijst.

Voor elk liedje in deze database werden vervolgens twee sets van features berekend, een set van audiofeatures en een van internetfeatures. De features werden initieel geselecteerd uit de literatuur en eventueel aangepast op basis van de classificatievereisten. Tevens werden enkele nieuwe features gedefiniëerd.

Deze features werden vervolgens gebruikt als input voor een classificatiemodel. Als model werd gekozen voor support vector machines en een variant hierop, least-squares support vector machines. Deze werden eerst met standaard kernelfuncties gebruikt, waarna gekeken werd naar andere kernels om de performantie te verhogen.

Als beste gemiddelde resultaat werd een ROCa waarde van 0.64 bekomen, wat overeenkwam met ongeveer 60% juiste classificatie (bij een random performantie van 50%). Op basis van dit resultaat kan besloten worden dat hitmuziek te modelleren valt. Als nevenresultaten werd ook nog de relatieve performantie van elk feature beschouwd.

Inhoudsopgave

Lijst van symbolen

$\mathcal{D}$	Dataset
$\mathbf{x}$	Inputvector voor het classificatiemodel
y	Label
$\mathbb{R}^d$	Inputruimte met dimensie d
$\mathbb{R}^{d'}$	Getransformeerde ruimte met dimensie d'
$K(.)$	Kerneloperatie
Ω	Kernelmatrix
γ	Regularisatieconstante
ξ	Slackvariabele
α	Lagrangevermenigvuldiger
$\mathbf{w}$	Gewichtsvector
b	biascorrectie term
Φ	Inputtransformatie
f	Frequentie
$s(n)$	Het discrete tijdssignaal
$S\,(f)$	Spectrum
T	Venstergrootte
$T_{overlap}$	Vensteroverlap
$p(f)$	Kansverdeling van het spectrum
μ	Centroïde
σ^2	Spreiding
γ_1	Skewness
γ_2	Kurtosis
$S_{mel}(i)$	Melgetransformeerde spectrum
DCT	Discrete cosinus transformatie
$\mathbf{C_j}$	Set van clustercentra
V	Intraclustervariantie
k	Aantal clusters
h	Bandbreedte
NGD	Genormaliseerde google afstand
t_i	tag behorende tot liedje met index i
w_{t_i}	gewichtsfactor behorende bij tag t
T_i	Set van unieke tags
IDF	Inverse Document Frequency
B_{T_i}	Gewichtsfactor bij tag T
D_{KL}	Kullback-Leibler divergentie
D_{KS}	Kolmogorov-Smirnov teststatistiek

Lijst van figuren

Lijst van tabellen

Hoofdstuk 1

Inleiding

Music information retrieval (MIR) is een multidisciplinaire tak waarin disciplines zoals musicologie, signaalverwerking, psychoacoustiek, computer wetenschap, statistiek samenkomen om de manier waarop wij interageren met muziek te bestuderen en eventueel te veranderen. Centraal in deze studie staat het concept van "music similarity", de mate waarin wij muziek van elkaar onderscheiden en vereenzelvigen met elkaar.

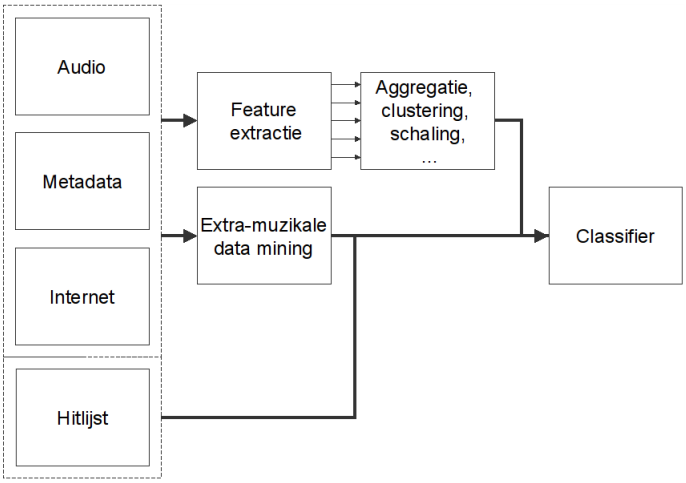
Reeds zeer vroeg werd onderkent dat music similarity geen standaard euclidische maat is en verscheidene pogingen zijn sindsdien dan ook ondernomen om een correcte maatstaf te vinden. Waar deze pogingen vroeger echter beperkt waren tot kleinschalige studies, werd er het laatste decennium door vooruitgang in artificiële intelligentie en computertechnologie MIR een nieuwe tool aangereikt om grootschalig experimenten uit te voeren en voor het eerst echt te proberen om music similarity te quantifieren.

Een plethora van manieren om dit te onderzoeken bestaan, de ene al succesvoller dan de andere. Een van de belangrijkere deeldomeinen, waarin tevens goede resultaten worden behaald, is automatische genreclassificatie. Echter ondanks deze goede performantie kampt genreclassificatie als maatstaf voor music similarity met enkele problemen, waarvan het voornaamste is dat genres, trouw aan de menselijke natuur, een bij uitstek vaag gedefinieerd concept zijn.

Daarom wordt in deze thesis een andere methode voor het meten van muzikale gelijkheid voorgesteld, namelijk het analyseren van hitlijsten. Hitlijsten vormen een uitmiddeling van de muzikale smaak van velen en vormen zo bij uitstek een maatstaf voor veelgesmaakte muziek. Daarenboven bieden hitlijsten een makkelijke manier om het succes van liedjes te meten.

Om deze stelling te testen werd in deze thesis dan ook een experiment uitgevoerd om hitmuziek te modelleren.

De gebruikte aanpak hiervoor weerspiegelt de standaard aanpak in MIR onderzoeken en bestaat uit het opzetten van een database, het opbouwen van een representatie van deze database door middel van featureextractie en tot slot de features te gebruiken als input voor een classificatiemodel. Het gebruikte model wordt weergegeven in onderstaande figuur en vormt tevens de leidraad voor deze thesistekst.



Aangezien de opbouw van het experiment en de gewenste eigenschappen van de features bepaald worden door het gekozen classificatiemodel, wordt dit eerst overlopen in hoofdstuk een. Hierin wordt voor de gekozen classificatiemodellen, support vector machines en een variant, least squares support vector machines, de theoretische onderbouw gegeven en vervolgens het proces van modeloptimalisatie door de keuze van geschikte kernels en hyperparameters toegelicht.

Het tweede hoofdstuk bespreekt de gebruikte databases waarvan achtereenvolgens de vereisten en de implementaties van worden overlopen. Tevens wordt de rol van de hitlijstdatabase belicht en de interactie tussen de verschillende databases. Een laatste deel geeft enkele statistieken over de databases.

Het derde hoofdstuk behandelt de gebruikte features. In de eerste sectie worden de audiofeatures behandeld. Hiervan wordt allereerst de selectie van de gebruikte features uit de literatuur besproken waarna elk feature apart wordt gedefinieerd. Tevens wordt de nood aan verdere verwerking van de features besproken en worden de verschillende methodes die hiervoor gebruikt werden opgesomd. Het laatste deel van deze sectie geeft een overzicht van de audiofeatures zoals deze aan de classifier worden aangeboden. Tevens wordt hierin de variantie in featurewaarden tussen liedjes onderling geanalyseerd.

Het vierde hoofdstuk behandelt een tweede klasse van features, namelijk deze gebaseerd op de analyse van extra-muzikale data. Een analyse van deze features wordt gemaakt waarna de features en hun specifieke eigenschappen gedefinieerd worden. De implementatie van deze features wordt in het daaropvolgende deel behandeld en het hoofdstuk wordt afgesloten met een overzicht van de internetfeatures zoals deze gepresenteerd worden aan de classifier.

Het vijfde hoofdstuk tenslotte behandeld de uitgevoerde experimenten. Hierin wordt allereerst de opbouw van de experimenten gedefinieerd samen met hun implementatie. Vervolgens worden de verschillende experimenten overlopen en hun resultaten gepresenteerd en besproken.

Tot slot is er het besluit met een evaluatie van de resultaten en een kritische beschouwing van het geleverde werk.

Hoofdstuk 2

Support Vector Machines

Voor de constructie van het classificatiemodel wordt gebruik gemaakt van support vector machines en de variant Least-Squares support vector machines. De theoretische opbouw van beide wordt in de volgende sectie kort uitgelegd en is gebaseerd op \Suykens.

2.0.1 Lineaire Support Vector Machines

2.0.1.1 Scheidbare data

Veronderstel een set lineair scheidbare data $\mathcal{D} = \{(\mathbf{x}_k, y_k)\}_{k=1}^{N}$, $\mathbf{x}_k \in \mathbb{R}^d$, $y_i \in \{-1, +1\}$ en cardinaliteit N, verdeeld in twee klassen dewelke strikt scheidbaar zijn door de set van hypervlakken $\mathcal{H} = \{H_i : \mathbf{w}_i^T.\mathbf{x}_k + \mathrm{b}_i\}$ waarvoor geldt dat als een punt x op het hypervlak H_i ligt, $\mathbf{w}_i^T.\mathbf{x}_k + \mathrm{b}_i = 0$.

Voor elk hypervlak wordt zijn corresponderende marge gedefinieerd als $d_+ + d_-$ met d_+, d_- de kortste afstand van het hypervlak tot het dichtstbijzijnde positieve, respectievelijk negatieve, punt. Indien we vervolgens een schaling van de set $\mathcal{D}$ definiëren zodat $min_k \left|\mathbf{w}_i^T.\mathrm{x}_k + \mathrm{b}_i\right| = 1$ en:

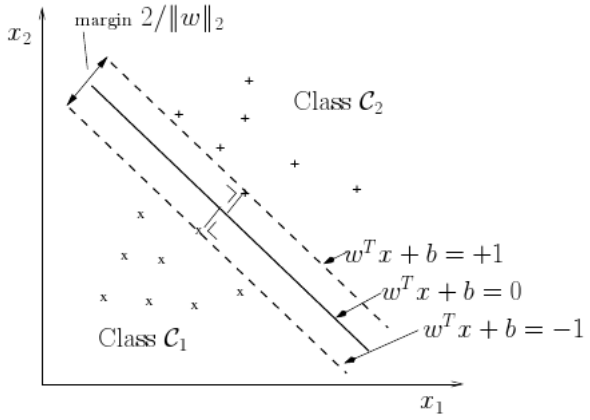
$$\begin{cases} \mathbf{w}_i^T.\mathbf{x}_k + \mathrm{b}_i \geqslant +1, & voor\ y_k = +1 \\ \mathbf{w}_i^T.\mathbf{x}_k + \mathrm{b}_i \leqslant -1, & voor\ y_k = -1 \end{cases} \tag{2.1}$$

equivalent aan:

$$y_k \left[\mathbf{w}_i^T.\mathrm{x}_k + \mathbf{b}_\mathrm{i}\right] \geqslant 1,\ \forall k \tag{2.2}$$

bekomen we de canonieke vorm van de hypervlakken zodanig dat $d_+ = d_- = \frac{1}{||\mathbf{w}||}$. De marge van een hypervlak is bijgevolg gelijk aan $\frac{2}{||\mathbf{w}||}$.

Het support vector machine algoritme stelt nu dat het hypervlak dat het beste de data scheidt het hypervlak H_i is dat de marge maximaliseert. Dit hypervlak kan makkelijk gevonden worden door $||\mathbf{w}||$ te minimaliseren, gegeven de beperkingen (2.2). Men bekomt m.a.w. het volgende



Figuur 2.1: Illustratie van het hyperplane en zijn marge in een tweedimensionale dataset met twee lineair scheidbare klassen. Overgenomen uit [1]

primale optimalisatieprobleem in $\mathbf{w}$:

$$\min_{\mathbf{w},\mathrm{b}} J(\mathbf{w}) = \frac{\mathbf{w^T w}}{2} \quad onderworpen\, aan: \; y_k \left[\mathbf{w}^T.\mathrm{x}_k + \mathrm{b}\right] \geqslant 1, \; \forall k \tag{2.3}$$

met als bijhorende classificatieregel:

$$y(\mathbf{x}) = sign\left(\mathbf{w}^T.\mathbf{x} + b\right) \tag{2.4}$$

Als men de Lagrangiaan opstelt en uitwerkt bekomt men het duale probleem in de Lagrange-vermenigvuldigers α_k:

$$\max_{\alpha} J(\alpha) = \frac{1}{2}\sum_{k,l=1}^{N} y_k y_l \mathbf{x}_k^T \mathbf{x}_l \alpha_k \alpha_l + \sum_{k=1}^{N} \alpha_k \quad onderworpen\, aan: \begin{cases} \sum_{k=1}^{N} \alpha_k y_k = 0 \\ \alpha_k \geq 0, \; \forall k \end{cases} \tag{2.5}$$

met als resulterende classifier:

$$y(\mathbf{x}) = sign\left(\sum_{k=1}^{N} \alpha_k y_k \mathbf{x}_k^T \mathbf{x} + b\right) \tag{2.6}$$

2.0.1.2 Niet scheidbare data

Indien de klassen wegens overlappende distributies niet exact scheidbaar is, moet voorgaande formulering aangepast worden om misclassificaties te tolereren. Deze extensie werd in 1995 door Vapnik & Cortes [2]uitgevoerd en bestaat uit het invoeren van positieve slackvariabelen ξ_k. Beschouwen we dezelfde dataset $\mathcal{D}$, maar deze keer verdeeld in twee niet lineair scheidbare klassen, dan wordt (2.2) aangepast in:

$$y_k \left[\mathbf{w}_i^T.\mathbf{x}_k + \mathrm{b_i}\right] \geqslant 1 - \xi_k, \; \forall k \tag{2.7}$$

wat aanleiding geeft tot volgende primale optimalisatieprobleem in $\mathbf{w}$ en ξ_k:

$$\min_{\mathbf{w},\mathrm{b},\xi} J(\mathbf{w},\xi) = \frac{\mathbf{w^T w}}{2} + c\sum_{k=1}^{N}\xi_k \qquad (2.8)$$
$$onderworpen\ aan: \quad y_k\left[\mathbf{w}^T.\mathbf{x}_k + \mathrm{b}\right] \geqslant 1,\ \forall k$$
$$\xi_k \geqslant 0,\ \forall k$$

met c een positieve reële constante en bijbehorende classifier (2.4).

Het duale optimalisatieprobleem in α wordt vervolgens:

$$\max_{\alpha} \frac{1}{2}\sum_{k,l=1}^{N} y_k y_l \mathbf{x}_k^T \mathbf{x}_l \alpha_k \alpha_l + \sum_{k=1}^{N}\alpha_k \qquad (2.9)$$
$$onderworpen\ aan: \quad \textstyle\sum_{k=1}^{N}\alpha_k y_k = 0,\ \forall k$$
$$0 \leqslant \alpha_k \leqslant c,\ \forall k$$

met c een positieve reële constante en bijbehorende classifier.

$$y(\mathbf{x}) = sign\left(\sum_{k=1}^{N}\alpha_k y_k \mathbf{x}_k^T \mathbf{x} + b\right) \qquad (2.10)$$

2.1 Niet lineaire Support Vector Machines

Aangezien vele problemen echter een niet lineaire oplossing hebben is er de noodzaak om bovenstaande methodes uit te breiden naar het niet lineaire geval. De generalisatie naar niet lineaire scheidingsfuncties voor SVM voorgesteld door Vapnik is om de lineaire classifier te gebruiken in een geschiktere ruimte. Meer concreet beschouwen we een (niet-lineaire) transformatie Φ die de inputruimte $\mathbb{R}^d$ mapt op een andere, geschiktere ruimte $\mathbb{R}^{d'}$ met d' mogelijk oneindig.

$$\Phi : \mathbb{R} \rightarrow \mathbb{R}^{d'} \qquad (2.11)$$

Werken in de primale ruimte vereist dat deze transformatie Φ expliciet geparameteriseerd wordt (wegens (2.8)) ,maar in de duale ruimte kan dit vermeden worden door gebruik te maken van de kernel transformatie. Immers indien er een kernel functie K bestaat die het inproduct van de getransformeerde inputvariabelen $\mathbf{\Phi}(\mathbf{x}_k), \mathbf{\Phi}(\mathbf{x}_l)$ uit (2.5) en (2.6) rechtstreeks uitrekent in R^d kan de parameterisatie van $\mathbb{R}^{d'}$ vermeden worden. Meer bepaald vereisen we een kernel functie K waarvoor geldt:

$$K(\mathbf{x}_k, \mathbf{x}_l) = \mathbf{\Phi}(\mathbf{x}_k)^T.\mathbf{\Phi}(\mathbf{x}_l) \qquad (2.12)$$

Gebruik makende van (2.12) kan dan het duale optimalisatieprobleem herschreven worden naar het niet-lineaire geval:

$$\max_{\alpha} \frac{1}{2} \sum_{k,l=1}^{N} y_k y_l K(\mathbf{x}_k, \mathbf{x}_l)\alpha_k\alpha_l + \sum_{k=1}^{N} \alpha_k \tag{2.13}$$
$$onderworpen\ aan: \quad \sum_{k=1}^{N} \alpha_k y_k = 0,\ \forall k$$
$$0 \leqslant \alpha_k \leqslant c,\ \forall k$$

Met bijbehorende classifier:

$$f(\mathbf{x}) = sign\left(\sum_{k=1}^{N} \alpha_k y_k K(\mathbf{x}_k, \mathbf{x}) + b\right) \tag{2.14}$$

2.2 Least-squares Support Vector Machines

Least-squares support vector machines is een modificatie van de standaard support vector machine voorgesteld door Suykens et al. waarbij enkele beperkingen van SVM worden geprobeerd te overkomen. De modificatie bestaat erin om een L_2norm te gebruiken voor de slackvariabelen i.p.v. de L_1 norm gebruikt in de standaard svm. Aanpassing van (2.8) geeft dan de primale vorm van LS-SVM's:

$$min_{\mathbf{w},\mathrm{b},\xi}\, J(\mathbf{w},\xi) = \frac{\mathbf{w^T w}}{2} + \frac{\gamma}{2}\sum_{k=1}^{N} \xi_k^2 \tag{2.15}$$
$$onderworpen\ aan: \quad y_k\left[\mathbf{w}^T.\Phi(\mathbf{x}_k) + \mathrm{b}\right] = 1 - \xi_k,\ \forall k$$

(2.16)

Met γ een te optimaliseren regularisatieconstante.

De Lagrangiaan $\mathcal{L}$ van dit optimalisatieprobleem is de volgende:

$$\mathcal{L}(\mathbf{w}, b, \xi; \alpha) = J(\mathbf{w}, \xi_\mathbf{k}) - \sum \alpha_k \left(y_k\left(\mathbf{w}^T\Phi(\mathbf{x}_k) + b\right) - 1 + \xi_k\right)$$

Na opstellen van de Karush-Kuhn-Tucker condities voor optimaliteit,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \rightarrow \mathbf{w} = \sum_{k=1}^{N} \alpha_k y_k \Phi(\mathbf{x}_k) \tag{2.17}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \rightarrow \sum_{k=1}^{N} \alpha_k y_k = 0 \tag{2.18}$$

$$\frac{\partial \mathcal{L}}{\partial \xi_k} = 0 \rightarrow \alpha_k = \gamma \xi_k \ , \forall k \tag{2.19}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 \rightarrow y_k \left[\mathbf{w}^T . \Phi(\mathbf{x}_k) + \mathrm{b} \right] - 1 + \xi_k = 0 \ , \forall k \tag{2.20}$$

wordt via eliminatie van $\mathbf{w}$ en de residues ξ de duale vorm van de LS-SVM bekomen

$$\max_{\alpha} \mathcal{J}_\gamma(\alpha) = \frac{1}{2} \alpha^T \left(\Omega + \frac{1}{\gamma} I_N \right) \alpha - Y^T \alpha \tag{2.21}$$

met $Y = [y_1; ...; y_N]$,$\alpha = [\alpha_1; ...; \alpha_N]^T$ en de kernelmatrix $\Omega = [\Phi(\mathbf{x}_1); ...; \Phi(\mathbf{x}_N)]^T . [\Phi(\mathbf{x}_1); ...; \Phi(\mathbf{x}_N)]$. De oplossing van dit optimalisatieprobleem wordt gevonden door evaluatie van het stelsel:

$$Y = \frac{\left(\Omega + \frac{1}{\gamma} I_N \right)}{\alpha}$$

De bijbehorende classifier is identiek aan (2.14).

$$f(\mathbf{x}) = sign \left(\sum_{k=1}^{N} \alpha_k y_k K(\mathbf{x}_k, \mathbf{x}) + b \right)$$

LS-SVM hebben als voornaamste voordeel dat het convexe optimalisatieprobleem uit (2.13) getransformeerd wordt in een lineair stelsel van vergelijkingen met als belangrijkste kost dat de spaarsheid van de oplossing opgeofferd wordt.

2.3 Kernels

Door gebruik te maken van kernelfuncties kan de expliciete parameterisatie van de transformatie vermeden worden. Een kernelfunctie is geldig voor een transformatie Φ en corresponderende ruimte H, indien deze voldoet aan de Mercer conditie.

De meest gebruikte kernel zijn de lineaire, polynoom en de RBF kernel:

$$K(x, y) = x^T y$$
$$K(x, y) = (x^T y + \tau)^d$$
$$K(x, y) = e^{\frac{||x-y||_2^2}{\sigma^2}}$$

De keuze van de meest geschikte kernel voor een bepaald probleem is een model selectie probleem en kan aangepakt worden door het minimaliseren van de (cross)validatiefout zoals beschreven in volgend hoofdstuk.

2.4 Optimalisatie van het model en zijn hyperparameters

Gegeven verschillende modellen $\mathcal{M}_i$ met hyperparameters h (de regularisatieconstante γ voor LS-SVM en c voor de reguliere SVM en eventuele kernelparameters) is (cross)validatie een manier om het meest performante model te selecteren. Door de performantie van het model op een aparte validatieset te testen kan het beste model geselecteerd worden. Deze validatieset kan enerzijds een aparte set zijn of een ongebruikt deel van de trainingsset zoals in crossvalidatie. In L-voudige crossvalidatie wordt de trainingsset L maal verdeeld in een (sub)trainingsset en een (sub)testset, waarna de modelperformantie op deze L sets getest wordt. Dit heeft als voordeel dat geen aparte validatieset gedefinieerd moet worden.

Hoofdstuk 3

Databases

Het begin van elk classificatieexperiment is de verzameling van de juiste data in een database. Voor dit experiment houdt dit twee subdatabases in, namelijk een audio database en een metadata-database, met elk hun eigen vereisten.

De audio database is de verzameling van de muzikale data en heeft enkele belangrijke vereisten met betrekking tot legaliteit, flexibiliteit, grootte en inhoud. Zo is het wenselijk dat de database zo veel mogelijk hitlijstgenoteerde liedjes heeft, toelaat om vele verschillende features te berekenen en legaal bekomen en bruikbaar is. Het (legaal) bekomen van deze database is echter een probleem dat zich geregeld stelt binnen MIR onderzoek, en enkele oplossingen zijn hiervoor voorgesteld.

Een eerste methode is het vermijden van gecopyrighte muziek door bijvoorbeeld gebruik te maken van creative-commons muziek, klassieke muziek zonder copyright of zelfgecomponeerde muziek om de legaliteit van hun databases te verzekeren. Deze methode is echter wegens het gebrek aan commerciële muziek niet interessant voor deze toepassing.

Een andere gangbare aanpak met betrekking tot muziekdatabases is het delen van de geëxtraheerde muziekfeatures i.p.v. de muziek zelf. Algemeen wordt hiervan aangenomen dat dit legaal is zolang de features de muziek niet kunnen reconstrueren. Dit biedt echter geen flexibiliteit tot de selectie van de features en hun parameters en is daarom minder geschikt.

Een laatste oplossing is het gedistribueerd opslaan en het uitrekenen van features on demand. Deze aanpak is veelbelovend, maar een bruikbare opstelling laat nog op zich wachten.

Een gedetailleerd overzicht van de verschillende databases wordt gegeven in [3].

Gezien het gebrek aan een geschikte audiodatabases werd daarom besloten om deze zelf te construeren door cd's te ontlenen in de Leuvense stadsbibliotheek Tweebronnen en deze te digitaliseren. Deze aanpak is flexibel met betrekking tot de inhoud van de database en is tevens ook volledig legaal aangezien het ontlenen en dupliceren van werken uit de bibliotheek voor onderzoeksdoeleinden legaal is [4, 5]. Het belangrijkste nadeel is dat de opbouw van de database echter zeer tijdrovend is.

Voor de opbouw van de audiodatabase zijn drie punten belangrijk:

- Sampling: in functie van de validiteit van de latere experimenten is de selectie van de

albums om op te nemen in de database van groot belang, een bias in artiesten of genres beïnvloet immers de latere resultaten. Daarom wordt gepoogd een zo random mogelijke selectie te bekomen.
- Digitalisatie: om de kwaliteit van de audio en hun daaruitvolgende feautures te verzekeren is het belangrijk om hier veel aandacht aan te besteden, en dit des te meer omdat de bibliotheekcd's niet in optimale staat zijn. Tevens moet in dit proces de juiste metadata aan elk liedje worden gekoppeld en is het belangrijk dat alles voldoende vlot verloopt om de audiodatabase zo groot mogelijk te maken.
- Opslag: aangezien er bij muziekdatabases al snel over ettelijke tientallen gigabytes aan data gesproken wordt, is de manier van opslag en aanspreking zeer belangrijk voor de latere bruikbaarheid.

De metadatabase daarentegen groepeert de niet-muzikale data van elk liedje. Dit houdt o.a. de titel en artiestinformatie in, maar tevens informatie over de kwaliteit, het genre en dergelijke. Informatie over de posities in de hitlijst vallen hier eveneens onder. Drie punten zijn in deze database van groot belang, namelijk:

- Kwaliteit: muzikale metadata is notoir voor zijn ambiguïteit en bijbehorende fouten, gaande van foute schrijfwijzes en foute veldinformatie tot ronduit foute informatie. Daarom moeten voldoende controles worden ingebouwd om de kwaliteit te verzekeren.
- Interne consistentie: wegens de dikwijls gebrekkige kwaliteit van metadata, is de consistentie van data afkomstig van meerdere bronnen niet gegarandeerd en moet deze verbeterd worden.
- Opslag: wegens de vele informatie in de metadatabase moet deze flexibel voorhanden zijn.

3.1 Implementatie van de AudioDB

3.1.1 Sampling

Idealiter zou deze selectie random uit de gebruikte hitlijst moeten zijn, maar enkele praktische gegevens beletten dit. Zo is het niet praktisch haalbaar om op liedjesniveau te selecteren, is het aanbod beperkt tot de ,weliswaar ruime, collectie van de bibliotheek en is daarenboven het aantal hits per cd eerder laag. Daarom werd besloten om een meer pragmatische aanpak te hanteren en de selectie van de cd's in twee fases uit te voeren. Allereerst werden zoveel mogelijk relevante verzamelcd's gekozen, aangezien deze meestal een groot percentage songs hebben die in de hitlijsten voorkomen.

Na deze stap werd vervolgens een random selectie gemaakt uit de verzameling albums van diverse artiesten (uit de pop/rock afdeling sinds het aantal hitlijst-genoteerde artiesten in de andere categorien vrijwel nihil is). Deze selectie werd quasi random uitgevoerd, met weliswaar een voorkeur voor best-of cd's, wederom omwille van de hogere dichtheid van songs die in de charts hebben gestaan.

In de USPOP2002[6] database wordt er wegens de hoge dichtheid van bekende nummers gekozen voor live-cd's, maar deze aanpak werd niet gevolgd wegens de dikwijls lage representativiteit van live-uitvoeringen ten opzichte van hun hitlijst-genoteerde uitvoeringen.

3.1.2 Digitalisatie

3.1.2.1 De digitalisatieopstelling

Voor de implementatie van het digitalisatieproces zijn twee factoren zeer belangrijk.

Ten eerste moet zoals gezegd de kwaliteit van het digitalisatieproces zo hoog mogelijk zijn en daarenboven ook meetbaar. De kwaliteitseis is zeer belangrijk wegens de onvoorspelbare invloed op de features die een slechte digitalisatie kan hebben (fouten kunnen zich namelijk op verschillende manieren manifesteren bv. vertragingen, extra ruis, een clicks, stiltes, ...). De meetbaarheid van de kwaliteit van het digitalisatieproces is belangrijk omdat gezien hun afkomst de cd's meestal in meer of mindere mate beschadigd zijn, zodat fouten in de digitalisatie onvermijdelijk zijn. Een meetbare kwaliteit laat dan toe om deze foutieve tracks er makkelijk uit te filteren.

Ten tweede moest het proces ook voldoende vlot zijn om toe te laten een groot aantal cd's te digitaliseren, zowel in termen van snelheid als in automatisatie van het proces. Van zeer groot belang in deze automatisatie is de link van het digitalisatieprogramma met een metadataservice om automatisch de gedigitaliseerde songs van metadata te voorzien. (Deze data is spijtig genoeg meestal niet op de cd zelf opgeslagen, dit is historisch gegroeid omdat dit in de begindagen van de cd niet nodig/nuttig was, en de extensie van de cd die dit wel incorporeerde de CD_EXTRA nooit is doorgebroken).

In gedachte deze twee vereisten werd een selectie gemaakt uit verschillende mogelijke digitalisatieprogramma's en CD-ROM stations.

Voor CD-ROM station werd geopteerd voor een oudere (LG HL-DT-STRW/DVD(GCC-4521B)) wegens de ondersteuning en correctheid van C2 pointers, Accuratestream en de afwezigheid van cache, allen belangrijke eigenschappen voor zowel de kwaliteit als de snelheid van het digitalisatieproces. [7, 8, 9]

Voor de selectie van het digitalisatieprogramma werd gekozen uit de volgende mogelijke kandidaten: Windows media player 10, Itunes 7.3.2, DbPowerAmp R12.4[1], EAC V0.99 prebeta 3 en CD extractor

Uiteindelijk werd EAC weerhouden wegens de uitgebreide featureset, de hoge kwaliteit van de rips, de mogelijkheid copy protection te omzeilen en de kostprijs (gratis).

De uiteindelijke configuratie van de digitalisatieopstelling was de volgende:

> Exact Audio Copy V0.99 prebeta 3 from 28. July 2007
>
> Used drive : HL-DT-STRW/DVD GCC-4521B Adapter: 0 ID: 1
>
> Read mode : Secure Utilize accurate stream : Yes
>
> Defeat audio cache : No

[1] Op het moment van schrijven is er een nieuwe versie R13 (released 2/6/2008) die enkele belangrijke nieuwe features aanbiedt, met name de tool PerfectMeta die de gegevens van 4 metadata services combineert.

Make use of C2 pointers : Yes

Read offset correction : 6

Overread into Lead-In and Lead-Out : No

Fill up missing offset samples with silence : Yes

Delete leading and trailing silent blocks : No

Null samples used in CRC calculations : No

Used interface : Native Win32 interface for Win NT & 2000

Gap handling : Not detected, thus appended to previous track

Used output format : Internal WAV Routines

Sample format : 44.100 Hz; 16 Bit; Stereo

Deze setup laat toe om een normale cd te rippen met een goede kwaliteit in gemiddeld 10 minuten. EAC geeft in zijn logs tevens nog enkele aanduidingen van de kwaliteit van het digitalisatieproces mee. Een typisch logextract is het volgende:

TOC of the extracted CD

Track — Start — Length — Start sector — End sector

---

1 — 0:00.00 — 0:08.52 — 0 — 651

2 — 0:08.52 — 2:13.60 — 652 — 10686

3 — 2:22.37 — 1:09.58 — 10687 — 15919

4 — 3:32.20 — 2:36.62 — 15920 — 27681

5 — 6:09.07 — 2:22.58 — 27682 — 38389

6 — 8:31.65 — 2:22.32 — 38390 — 49071

7 — . . .

8 — . . .

Track 1

Filename E:\MUSICDB\The Kinks__BBC Sessions 1964-1977 (disc 1)__01__The Kinks__Interview__2001__Rock__Data__030BAA15.wav

Peak level 53.4 %

Track quality 100.0 %

Test CRC B79FFB0A

Copy CRC B79FFB0A

Accurately ripped (confidence 4) [8BA69A37]

Copy OK

Track 2 .....

De TOC (table of contents) geeft de verschillende start/stop locaties van alle liedjes op de cd weer. Een fout hierin (bv. niet alle liedjes worden gevonden) resulteert in alle waarschijnlijkheid in een foute digitalisatie. De verschillende kwaliteitsindicatoren per lied zijn:

- Track quality: dit geeft weer hoe vaak EAC een slechte sector op de cd is tegen gekomen en geeft een ondergrens op de kwaliteit. Het is slechts een ondergrens aangezien EAC enkele correctiefeatures voorziet voor slechte sectoren (zo zal een slechte sector tot 82 maal uitgelezen worden, en dit eventueel aan verschillende snelheden, totdat de er voldoende keren dezelfde waarde is uitgelezen). Indien de correctiefeatures van EAC tekortschieten is er echter nog een tweede correctiemechanisme in de cd-standaard zelf namelijk de cross-interleaved Reed-Solomon code. Indien ook deze tekortschiet zal er interpolatie gebruikt worden zodat er enigszins gracefull degradation is. Enkel bij het tekortschieten van al het voorgaande zal een harde fout voorkomen.
- Test & Copy CRC: In de gekozen modus zal EAC het liedje twee keer digitaliseren en telkens een CRC berekenen. Indien deze identiek zijn geeft dit een extra indicatie van het correct zijn van de digitalisatie.
- Accuraterip: Dit is een verderzetting van het idee van CRC checks, bij accuraterip wordt immers ook een CRC berekend, maar deze wordt vergeleken met de CRC waarden van andere personen (en digitalisatiesetups) die zijn opgeslagen op een centrale internetserver. De confidencelevel geeft dan weer hoeveel personen dezelfde CRC waarde bekomen en dit kan een sterke indicatie zijn van de kwaliteit. Accuraterip is echter niet voorhanden voor elke cd en wordt ook beïnvloed door verschillende pressings van een cd zodat sommige liedjes verkeerdelijk incorrect worden gelabeld.
- Copy status: Dit geeft weer of EAC het gehele liedje heeft kunnen digitaliseren binnen het gegeven tijdsbestek (om snelheidsredenen is de maximale tijd die aan de digitalisatie van een liedje besteed kan worden beperkt tot twee maal de lengte van het liedje. Dit om het overmatig veel tijd besteden aan “slechte” liedjes tegen te gaan.)

Een randopmerking die nog bij deze kwaliteitsparameters moet vermeld worden is dat deze enkel geldig zijn bij standaard (red book) cd’s, niet conforme cd’s (o.a. met copy-protection

systemen) geven dikwijls incorrecte resultaten.[7, 9, 8]

3.1.3 Opslag

Voor de opslag van audiofiles is de belangrijkste parameter de gekozen compressie wegens zijn invloed op de bestandsgrootte, kwaliteit en de laadtijden (het benodigde transport over het netwerk samen met de decodering).

Bij compressie stelt zich de keuze tussen lossy en lossless compressie. Lossy verlaagt de bestandsgrootte aanzienlijk en de toepassing van een human auditory model zoals bij bv. mp3 kan een gunstige invloed hebben op de perceptuele betekenis van de features. Anderzijds biedt lossless meer flexibiliteit en zekerheid naar de features toe (de invloed van lossy codering op features en hun berekening is niet empirisch gekend hoewel [10] stelt dat ten minste MFCC's robust zijn aangaande MP3 codering en classsificatie) en daarom is uiteindelijk ook voor lossless compressie gekozen. Verschillende codecs zijn hiervoor voorhanden. Uit de mogelijke kandidaten werden WavPack en FLAC weerhouden voor hun mix van goede compressie (gemiddeld tot ongeveer 60% van de originele bestandsgrootte), goede decodeersnelheid, maturiteit en opensource implementatie[11].

Bij gebruik op de HPC cluster bleken echter de laadtijden van ongecompresseerde en gecompresseerde audio vergelijkbaar te zijn, en aangezien er voldoende schijfruimte voorhanden was op het HPC bestandssyteem, werd besloten deze stap uiteindelijk over te slaan.

3.2 Implementatie van de metadatabase

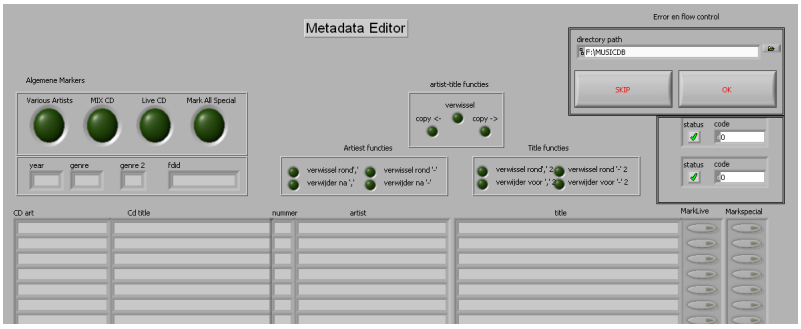
3.2.1 Bronnen van metadata

Er wordt onderscheid gemaakt tussen enerzijds de hitlijstmetadata en de andere metadata (in volgende voor de duidelijkheid de muziekmetadata genoemd).

3.2.1.1 Muzikale metadata

Er zijn verschillende services op het internet die muziekmetadata aanbieden met allen hun eigen voor- en nadelen op het vlak van aanbod, correctheid en kostprijs. De meest bekende providers zijn: All Music Guide (betalend, gebruikt in o.a. Windows Media Player), Gracenote (betalend, Itunes, ...), FreeDB[2] (gratis, user submitted content) en MusicBrainz (gratis, user submitted content). Deze bieden allen minstens een CD gebaseerde muziekmetadata service aan waarbij de CD wordt geïdentificeerd aan de hand van de TOC. Recentelijk zijn enkelen ook liedjesgebaseerde metadataservices beginnen aanbieden met behulp van acoustic fingerprinting zoals o.a. AMG Lasso, Gracenote MusicID, MusicBrainz via MusicIP's MusicDNS,... maar deze service is minder betrouwbaar en uitgebreid dan de CD gebaseerde methodes en wordt daarom niet gebruikt.

[2]Gracenote en FreeDB delen een gemeenschappelijke voorouder namelijk CDDB



Figuur 3.1: Screenshot van de ontworpen metadata-editor

Naast deze grote metadataservices zijn er ook nog een veelvoud van andere bronnen mogelijk zoals bv. Amazon, Discogs, ... maar voor het ophalen van muziekmetadata werden deze hier niet beschouwd.

Na deze factoren in overweging te hebben genomen is gekozen voor FreeDB wegens zijn open/gratis karakter, groot aantal CD's in de database en de integratie met EAC. In andere MIR projecten wordt echter dikwijls gekozen voor AMG metadata als grondwaarheid, doch dit was hier niet opportuun wegens het uitgesproken amerikaans karakter van de service waardoor er te veel CD's ontbraken en/of fout gelabeld waren, waardoor op een steekproef de kwaliteit van AMG niet beter was dan FreeDB.

3.2.1.2 Hitlijsten

Als hitlijst werd gekozen voor de Engelse hitlijst, "The UK singles chart". De keuze hiervoor was een pragmatische aangezien dit de enige hitlijst was die gratis en volledig online beschikbaar was [12] en dit voor zijn hele geschiedenis, gaande van een top 12 in 1952 tot een top 100 heden ten dagen, goed voor 11778 artiesten met 33672 genoteerde liedjes.

De hitlijst is echter wel enkel beschikbaar in html en werd daarom eerst integraal gedownload en nadien geparset tot wekelijks hitlijsten.

3.2.2 Metadata opstelling

De twee hoofdbronnen van muzikale metadata zijn enerzijds de FreeDB data dewelke vanuit EAC wordt doorgegeven in de bestandsnaam van de gedigitaliseerde song (deze bestandsnaam is opgebouwd uit alle FreeDB velden gescheiden door "__") en anderzijds de logfile van het digitalisatieproces. Deze data wordt dan bewerkt met de hiervoor gecreëerde Metadata editor (LabView/Matlab) en nadien opgeslagen.

Het programma neemt zoals vermeld als input de bestandsnamen van de liedjes van een cd tezamen met de logfile. Uit de bestandsnamen van de audiofiles worden de 9 verschillende FreeDB velden (8 metadata velden plus de FreeDB ID code) geëxtraheerd namelijk : Artiest, Titel, CD artiest, CD titel, tracknummer, Jaar, ID3 genre, Genre en de FreeDB ID. Deze worden manueel gecontroleerd en eventueel gecorrigeerd (hiertoe zijn enkele functies toegevoegd

om veelvoorkomende fouten (semi-)automatisch te verbeteren zoals bv. het omwisselen van artiest en titelvelden, alles in een veld,...). Nadien wordt de cd/songs nog van extra markers voorzien; op cd niveau zijn er markers voor live-cd, mix-cd en verzamel-cd en op songniveau markers voor live-versie en speciale-versie (bv. een remix). Nadat uit de logfile de verschillende parameters van het ripproces zijn bekomen (Header, Peak level, Track quality, Test CRC, Copy CRC, AccurateRip data, Copy status en Suspicious, de eventuele posities van slechte frames) wordt de data naar de opslag verplaatst.

3.2.3 Opslag

Voor de opslag van de metadata stelt zich vooral het probleem van flexibiliteit en laadtijden. Om deze database op te bouwen zijn verschillende oplossingen voorhanden maar na het bekijken van deze werd gekozen om voor elke geluidsbestand apart een matlab-metadatabestand aan te maken voor overzicht, flexibiliteit en gebruiksgemak. Het grote nadeel hiervan is echter dat de databasefunctionaliteit volledig zelf ontworpen moeten worden en wegens de vele files die telkens ingeladen moeten worden om de gehele database te doorzoeken, queries vrij traag zijn.

3.2.4 Verzekering interne consistentie van de metadatabase

Hoewel zowel bij de hitlijsten als bij de muziekmetadata de kwaliteit van de data zo veel mogelijk bewaakt werd, is het linken van beide toch niet voor de hand liggend. Dit is te wijten aan spelfouten, meerdere mogelijke spellingen van artiesten, meerdere aliasen van artiesten, verschillen in notatie (bv. “feat.”, “ft.” of “featuring”), incomplete titels,...

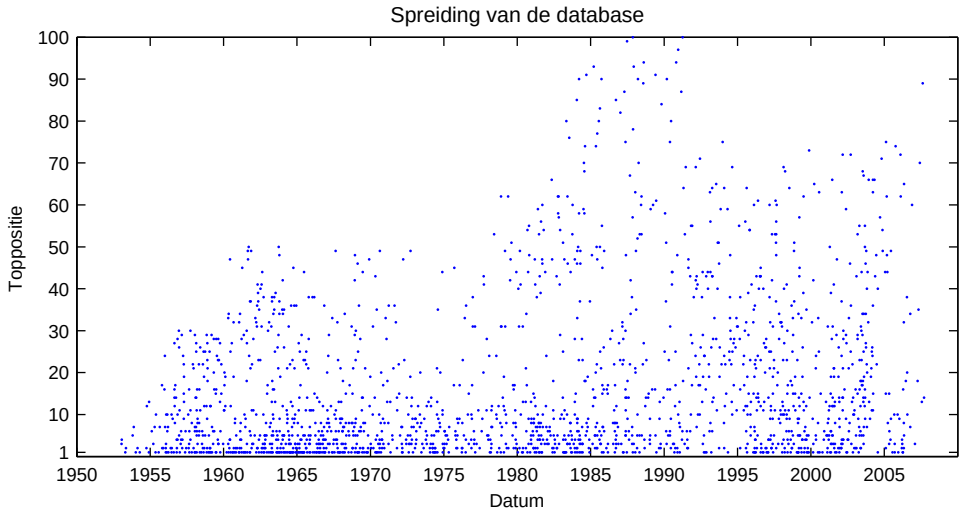
Om deze problemen te verhelpen werden er twee verschillende strategieën geïmplementeerd. Zo wordt voor het linken niet de volledige artiestennamen/songtitels gebruikt, maar wordt er gebruik gemaakt van een handle om de robustheid van de velden tegen voorgenoemde fouten te verhogen. De handle wordt met behulp van volgende regels opgebouwd (cfr. [3] \EXTRA):

- “the” werd verwijderd
- “and” werd vervangen door “&”
- afkortingen worden consistent geschreven (bv. featuring, feat., ... wordt ft.)
- leestekens en speciale tekens worden verwijderd (behalve “&”)
- alles tussen haakjes wordt verwijderd
- spaties worden verwijderd
- alles in lowercase

Aggressievere regels zijn mogelijk maar deze brengen de uniekheid van de handle in gevaar en werden daarom niet geïmplementeerd.

Indien voorgaande aanpak faalt wordt een andere strategie gevolgd gebaseerd op de editor-afstand.

De editor-afstand meet de afstand tussen twee strings en indien deze lager is dan een threshold worden de strings manueel vergeleken en eventueel intern consistent gemaakt.



Figuur 3.2: Visualisatie van de spreiding van de liedjes uit de database hun toppositie en de datum waarop deze werd bereikt

3.3 Overzicht van de database

De totale database telt 9023 liedjes[3], afkomstig van 450 cd's. Van deze 9023 konden er 1861 gerefereerd worden naar de gebruikte hitlijst. De spreiding van deze liedjes hun toppositie in de hitlijst en de datum waarop deze bereikt werd is weergegeven in figuur 3.3. De totale database (audio + liedjes) is 315Gb groot.

[3] Eventuele dubbels meegerekend, maar liedjes met een onvoldoende digitalisatiekwaliteit niet meegeteld

Hoofdstuk 4

Audiofeatures

Muzikale featureextractie is het proces dat poogt het ruwe audiosignaal om te vormen tot een dataset die de redundantie minimaliseert, de muzikale informatie synthetiseert en toelaat om, perceptueel relevante, vergelijkingen te maken tussen muziekstukken [13]. Als bijkomende vereiste aan het proces kan nog gesteld worden dat deze dataset aangepast moet zijn aan de gewenste classificatie en de gebruikte classificatiealgoritmen. Gezien deze eisen en de belangrijke invloed van de features op de performantie, is het dan ook niet verwonderlijk dat er een plethora van mogelijke features is gedefinieerd. [14, 15, 16] geven hiervan een overzicht en de MPEG-7 standaard probeert om een framework te bieden.

Ongeacht de exacte algoritmen kan men deze features echter grofweg indelen in twee categoriëen, namelijk die met een laag of een hoog niveau van muzikale abstractie. De laag niveau features bieden een eerder ruw beeld van de muziek, wat slechts moelijk vertaald kan worden in muzikale concepten. Voorbeelden zijn bv. statistische eigenschappen van het tijdssignaal of spectrum. Hoog niveau features daarentegen incorporeren meer muziektheorie om de muziek te beschrijven in musicologisch relevante eigenschappen zoals bijvoorbeeld tempo, toonaard en bezetting.

Voor de classificatie van hits lijken hoog niveau features zeer aanlokkelijk, maar door de grote corpus van gevarieerde muziek nodig in deze classificatietaak hebben ze enkele belangrijke praktische nadelen. Zo vereisen zij meestal veel rekentijd, generaliseren ze slecht naar verschillende soorten muziek (bv. beatextractie heeft geheel andere eisen bij jazz dan bij dance), zijn ze zeer foutgevoelig en beperkt tot het huidige musicologische framework (wat niet noodzakelijk correleert met de menselijke perceptie). Daarom werd gekozen voor de meer bottom-up aanpak van de laag-niveau features, waardoor de taak van het zoeken van muzikaal belangrijke concepten verschuift van de features naar de classificatie.

Onderstaand deel geeft een overzicht van de features die uit de literatuur geselecteerd werden voor de classificatie en is grotendeels gebaseerd op [14, 15, 16]. De selectie is gebaseerd op de resultaten uit o.a. [17, 18, 6, 19, 20, 21, 22, 23, 24]

4.1 Temporele features

Voor de berekening van de temporele features wordt het tijdssignaal $s(n)$ verdeeld in frames van lengte T met een interframe overlapping van $T_{overlap}$, waarna de features voor elk frame berekend worden. Typische waarden voor T en overlap $T_{overlap}$ zijn 30ms resp. 20ms.

4.1.1 Zerocrossing rate

De zerocrossing rate, het aantal keer dat het signaal de nul as kruist, is een indicator van zowel de periodiciteit van het signaal (periodische signalen neigen meer naar lagere zero crossing rate waarden dan ruizige signalen) als een ruwe indicator van de ogenblikkelijke frequentie in het frame.

$$\begin{aligned} zero\,crossing\,rate &= \frac{1}{T}\sum_{n=0}^{T-1} R\left(sign\left[s(n)s(n-1)\right]\right) \\ & \quad Met \begin{cases} R(-1) & = 1 \\ R(1) = R(0) & = 0 \end{cases} \end{aligned}$$

4.1.2 RMS

De RMS van een frame weerspiegelt de energie en is op deze manier losjes gecorreleerd met de geluidssterkte (perceptuele geluidssterkte is o.a. ook nog frequentie en tijdsafhankelijk).

$$RMS = \sqrt{\frac{1}{T}\sum_{n=0}^{T-1}\left[s(n)\right]^2} \tag{4.1}$$

4.2 Spectrale features

Gegeven het frameontbonden signaal uit sectie 4.1 wordt hiervan de STFT genomen met een hammingwindow, zodat voor elk frame een spectrum $S(f)$ wordt bekomen. De features worden vervolgens voor elk frame berekend.

4.2.1 Spectrale momenten

Vier momenten werden geselecteerd om beknopte informatie te verschaffen over de algemene verdeling van het spectrum. Beschouwen we het spectrum $S(f)$ als een distributie met $p(f)$, de kans om de frequentie f waar te nemen, als volgt gedefinieerd:

$$p(f) = \frac{|S(f)|}{\int |S(f)|df} \tag{4.2}$$

Dan worden de gebruikte momenten als volgt gedefinieerd

- centroïde μ: de verwachte waarde van de distributie

$$\mu = \int fp(f) \tag{4.3}$$

- spreiding σ^2: het tweede centrale moment

$$\sigma^2 = \int (f-\mu)^2 p(f)df \tag{4.4}$$

- skewness γ_1: het derde gestandardiseerde moment

$$m_3 = \int (f-\mu)^3 p(f)df \tag{4.5}$$

$$\gamma_1 = \frac{m_3}{\sigma^3} \tag{4.6}$$

- kurtosis γ_2: het vierde gestandariseerde moment

$$m_4 = \int (f-\mu)^4 p(f)df \tag{4.7}$$

$$\gamma_2 = \frac{m_4}{\sigma^4} \tag{4.8}$$

4.2.2 Perceptuele spectrale features

Deze features geven een meer perceptueel gemotiveerde beschrijving van het spectrum.

4.2.2.1 Spectrale roll-off

De spectrale roll-off frequentie is die frequentie $f_{rolloff,c}$ waarvoor een bepaalde fractie c van de totale energie vertegenwoordigd is in het deel $[0, f_{rolloff,c}]$. Als waarde voor c is zowel 0.95 [22], als 0.85 [25] voorgesteld. Beide waarden worden gebruikt als feature.

$$f_{rolloff,c} \leftrightarrow \int_0^{f_{rolloff,c}} |S(f)|^2 df = c \int |S(f)|^2 df \tag{4.9}$$

4.2.2.2 Spectrale helderheid

Een duale methode aan de spectrale roll-off is het vaststellen van een cut-off frequentie en de hoeveelheid energie boven deze frequentie te meten. Als cut-off frequentie is zowel 1500 Hz als 3000Hz [26] voorgesteld.

$$Helderheid = \frac{\int_0^{f_c} |S(f)|^2 df}{\int_0^{f_{nyq}} |S(f)|^2 df} \tag{4.10}$$

4.2.2.3 Totale volume

Als benadering van het totale volume N wordt de sommatie van de amplitude in de mel-banden na auditieve verwerking $S_{mel,auditief}(i)$ gebruikt (zie sectie 4.3.1.2 voor de berekening van de mel-auditieve banden)

$$N = \sum S_{mel,auditief}(i) \tag{4.11}$$

4.2.2.4 Perceptuele scherpte

De scherpte is het perceptuele equivalent van de spectrale centroïde, maar dan berekent over de mel-auditieve banden.

$$Scherpte = \frac{\sum i S_{mel,auditief}(i)}{N} \tag{4.12}$$

4.2.2.5 Perceptuele spreiding

De perceptuele spreiding meet de afstand tussen de mel-auditieve band met de grootste amplitude en het totale volume N.

$$Perceptuele\, spreiding = \left(\frac{N - max_i S_{mel,auditief}(i)}{N} \right)^2 \tag{4.13}$$

4.2.3 Varia

4.2.3.1 Spectrale vlakheid

De spectrale vlakheid wordt gedefinieerd als de verhouding tussen het geometrisch en het rekenkundig gemiddelde en biedt een maatstaf voor de vlakheid/gepiektheid van het spectrum. De spectrale vlakheid geeft daardoor ook een indicatie van de ruisheid/sinusoïdaliteit van het spectrum.

$$flatness = \frac{\sqrt[N]{\prod |S(f)|}}{\mu} \tag{4.14}$$

verwant met de vlakheid wordt ook een tonaliteitindicator gedefinieerd als

$$tonaliteit = min\left(\frac{10\log(flatness)}{-60}, 1\right) \tag{4.15}$$

deze indicator is dicht bij 1 voor tonale signalen en dicht bij 0 voor ruizige signalen.

4.2.3.2 Spectrale entropie

De spectrale entropie geeft de entropie weer van het spectrum.

$$Spectral\,entropy = \sum p(f)\log p(f) \tag{4.16}$$

4.3 Cepstrale features

4.3.1 Mel Frequency Cepstral Coefficients (MFCC)

4.3.1.1 MFCC opbouw

De mel frequency cepstral coefficients (MFCC) is een feature die een compacte, perceptueel gemotiveerde benadering geeft van de spectrale omhullende. Meer bepaald wordt het spectrum onderworpen aan een frequentiewarping geinspireerd op het gehoorstelsel, een logcompressie om de perceptie van volume te imiteren en een DCT om een compacte voorstelling te bekomen.

De frequentiewarping is een non-lineaire transformatie gebaseerd op de mel schaal, dit is een empirisch bepaalde schaal die de frequentie relateert aan de gepercipieerde afstand in toonhoogte t.o.v. een referentietoon van 1000Hz. Als benadering van deze schaal wordt meestal gebruik gemaakt van de formule van Fant [27]4.1

$$mel = 2595\log_{10}\left(1 + \frac{f}{700}\right) \tag{4.17}$$

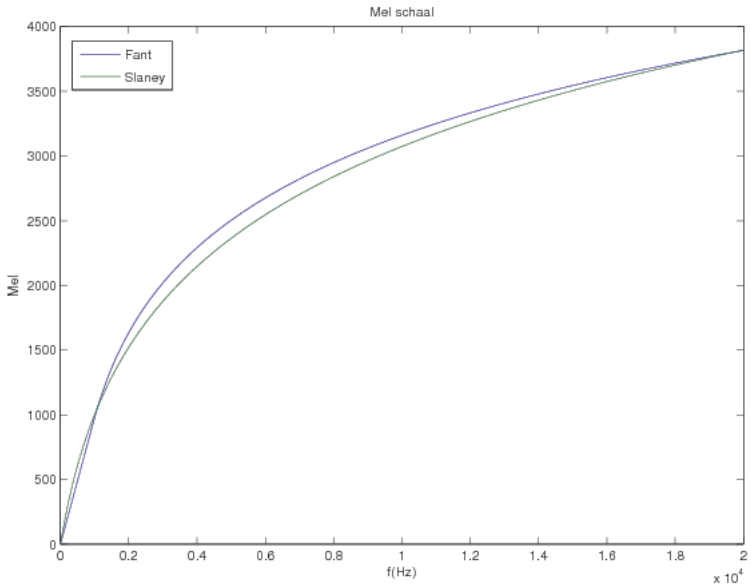
Op basis van deze schaal wordt een filterbank geconstrueerd met driehoekige filters H_i dewelke volgens de melschaal uniform gespatieerd worden. Het mel getransformeerde spectrum S_{mel}wordt bekomen na filtering en integratie van het originele magnitudespectrum S voor elke filterband.

$$S_{mel}(i) = \int H_i(f)|S(f)|df \tag{4.18}$$

De mfcc coefficienten worden dan gegeven door:

$$MFCC = DCT(log|S_{mel}(i)|) \tag{4.19}$$

Sinds de introductie van de MFCC's, zoals hierboven beschreven, in 1980 door Davis en Mermelstein werden echter al verschillende aanpassingen voorgesteld om het originele schema te verbeteren. Een vergelijking van de performantie van deze op de sprekeridentificatie taak kan gevonden worden in [27].



Figuur 4.1: Conversie van Hz naar de melschaal volgens twee benaderingen van de Melschaal; (donker) volgens de formule van Fant en (licht) volgens de formule van Slaney

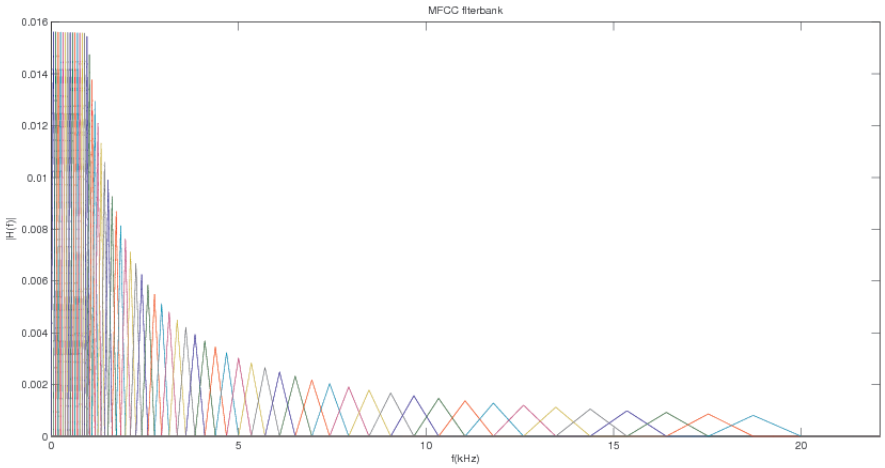
4.3.1.2 MFCC Implementatie

Het gebruikte schema voor MFCC's te berekenen is echter licht afwijkend aan het voorgaande en is gebaseerd op hetgene gebruikt wordt voor de constructie van de USPOP2002 database \USPOP2002. Deze is echter verbeterd met de conclusies uit [27] door gebruik te maken van een MFCC filterbank volgens Slaney [1] . Meer bepaald werden de volgende aanpassingen doorgevoerd t.o.v. de originele MFCC:

1. Een filterbank gebaseerd op het werk van Slaney.
   Slaney's filterbank gebruikt een licht andere parameterisatie van de mel schaal om de filters te verdelen en dit gecombineerd met een herschaling van de filters zodat elke filterband ongeveer dezelfde energie heeft (de bredere filters bij hogere frequenties aggregeren anders te veel energie). De spatiering van de filters is als volgt:
   - 15 filters lineair geplaatst tussen 0 en 1000 Hz[2]
   - 45 filters logaritmisch geplaatst tussen 1000 Hz[2] en 20000 Hz.
     Elke filter H_i wordt vervolgens herschaald naar gelijke oppervlakte zodat $\sum_{k=1}^{N} H_i = 1,\ Voor\, i = 1...60.$
2. Een perceptueel correctere weging van de geluidssterkte door de Hynek formule[28] te gebruiken.

[1] In [27]Ganchev04 presteert de HFFC filterbank nog beter, maar deze vereist het tunen van een extra parameter wat voor dit experiment niet gewenst is.

[2] Wegens uitbreiding van de oorspronkelijke filterbank van Slaney avn 40 naar 60 filters om het frequentiebereik van 0 tot 20kHz te accomoderen, is het exacte breakpoint echter iets verschoven naar 979 Hz



Figuur 4.2: De ontworpen MFCC filterbank

4.3.2 Delta MFCC's

Om de tijdsevolutie van de MFCC's te volgen worden delta- en delta-delta-MFCC's gebruikt.

$$\delta MFCC = \frac{\partial MFCC}{\partial t} \tag{4.20}$$

$$\delta\delta MFCC = \frac{\partial \partial MFCC}{\partial t^2} \tag{4.21}$$

4.4 Featureverwerking

De features hierboven vormen een eerste synthese van het muzieksignaal, maar zijn nog steeds te hoog-dimensionaal om rechtstreeks te gebruiken in het classificatieproces. Daarom worden enkele technieken gebruikt om deze ruwe features verder te synthetiseren naar een vorm bruikbaar voor de classificatiealgoritmen. Meer bepaald wordt er gebruik gemaakt van enkele technieken om de dimensie te verkleinen van de data en om de tijdsevolutie van de data te modelleren aangezien de gebruikte classifiers hiervoor ongevoelig zijn. In de literatuur worden hiervoor wederom verschillende technieken gebruikt, maar gezien de schaal van het experiment en de computationele intensiteit werden volgende technieken weerhouden.

4.4.1 Kernel dichtheid schatter

Kernel density estimation is een manier om de probability density function $f(x)$ van een random variabele te schatten. Gegeven een serie observaties x_i, een kernelfunctie $K(z)$ en bandbreedte h, wordt de pdf $f(x)$ benaderd als:

$$\hat{f}(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right) \tag{4.22}$$

Indien als kernelfunctie $K(z) = I(|z| < 1)$wordt genomen bekomen we het histogram, maar andere kernelfuncties zoals een gaussiaan kunnen geselecteerd worden om de pdf te smoothen[29].

4.4.2 K-means

K-means is een iteratief clusterings algoritme dat een dataset partioneert in k mutueel exclusieve clusters. Gegeven een dataset $\mathcal{D} = \{\mathbf{x_i}\}^N$met dimensionaliteit d en cardinaliteit N en een set van k initiele clustercentra $\{\mathbf{C}_{j,l=0}\}_{j=1}^{j=k}$ is het algoritme als volgt:

1. Associeer elk punt $\mathbf{x_i}$ met zijn dichtsbijzijnde clustercentrum $\mathbf{C}_{j,l}$
2. Bereken voor elke partitie het nieuwe clustercentrum $\mathbf{C}_{j,l+1}$ als de centroïde van de set van alle geassocieerde punten $A_{j,l}$

$$\mathbf{C}_{j,l+1} = \frac{\sum_{i \in A_{j,l}} \mathbf{x_i}}{|A_{j,l}|} \tag{4.23}$$

3. Herhaal stap 1 en 2 totdat de totale intraclusterafstand V niet meer daalt

$$V_l = \sum_{j=1}^{k} \sum_{i \in A_j} (\mathbf{x_i} - \mathbf{C_{j,l}}) \tag{4.24}$$

Het resultaat van k-means is sterk afhankelijk van k en de initiële clustercentra en levert niet gegarandeerd het globale optimum, zodat meestal meerdere iteraties voor verschillende waarden van k en initiele clustercentra gebruikt worden.

Het basis k-means algoritme is sinds zijn ontstaan in 1956 al op verschillende manieren geoptimaliseerd, de in deze thesis gebruikte variant is diegene beschreven in [30] dewelke gebruik maakt van de driehoeksongelijkheid om de berekeningen te versnellen.

4.4.3 Markov keten

Ter synthese van tijdsdata kan deze gemodelleerd worden als een k^e-orde markovketen[3]. Gegeven een discrete tijdsserie $x(n)$ van N verschillende discrete staten S_i , $met\ i = 1 \ldots N$ wordt een k^e-orde Markov-keten gedefinieerd aan de hand van zijn transitieprobabiliteiten van een geordende set van k-1 voorloperstaten $A_j = \{S_{k-1}, \ldots, S_0\}$naar de staat S_i. Deze transitieprobabiliteit p_{i,A_j} wordt gegeven door:

$$p_{i,A_j} = p\left(x(n+1) = S_i | \{x(n), \ldots, x(n-k)\} = A_j\right) = \frac{C(S_i, A_j)}{\sum_i \sum_j C(A_j)} \tag{4.25}$$

met $C(S_i, A_j), C(A_j)$ de frequentie van voorkomen van de sequentie (S_i, A_j) resp. (A_j) in x(n).

[3]De approximatie van muziekdata met markovketens is natuurlijk erg grof, maar wegens o.a. het grote aantal herhalingen van bepaalde patronen in muziek wel nuttig geacht.

Omdat een compleet k^e-orde model met alle mogelijke combinaties (S_i, A_j) een matrix met N^k transitieprobabiliteiten vereist, en aangezien voor hogere orde modellen vele transitieprobabiliteiten (bijna) nooit voorkomen worden, wordt dit model nadien nog geprunened om onnodige staten te verwijderen en zo de dimensionaliteit te reduceren.

4.5 Overzicht van de gebruikte audiofeaturevectors

4.5.1 Temporele en spectrale features

De temporele en spectrale features worden, na een eventuele schaling, herleid tot een histogram. De schaling, grenzen en bandbreedte van het histogram worden zo gekozen dat deze een optimum vormen tussen maximale resolutie en minimaal aantal bins. Hierdoor blijft de lengte van de featurevector beperkt en kan verdere smoothing vermeden worden.

Als initiële waarde voor de bandbreedte wordt gekozen voor de schatting van Freedman-Diaconis, $h = 2\frac{\text{IQR}(x)}{n^{1/3}}$, waarna het genormaliseerde gemiddelde histogram en enkele individuele genormaliseerde histogrammen werden gebruikt om de parameter eventueel aan te passen.

De bekomen featurevectors worden hieronder overlopen, samen met hun gemiddelde genormaliseerde histogrammen (berekend over 60 random liedjes) en enkele individuele genormaliseerde histogrammen om de typische variantie van het histogram over de corpus van liedjes weer te geven. Een grote onderlinge variantie staat immers voor belangrijke onderlinge verschillen tussen de liedjes onderling voor deze feature, wat voor classificatieperformantie een indicatie kan geven over de belangrijkheid van de feature.

4.5.1.1 Zerocrossing rate

Empirisch bereik: $[0, 22050]\, Hz$

Schaling : $\log(zcr + 1)$

Bereik histogram: $[0, 10]$

Bandbreedte h: 0.05

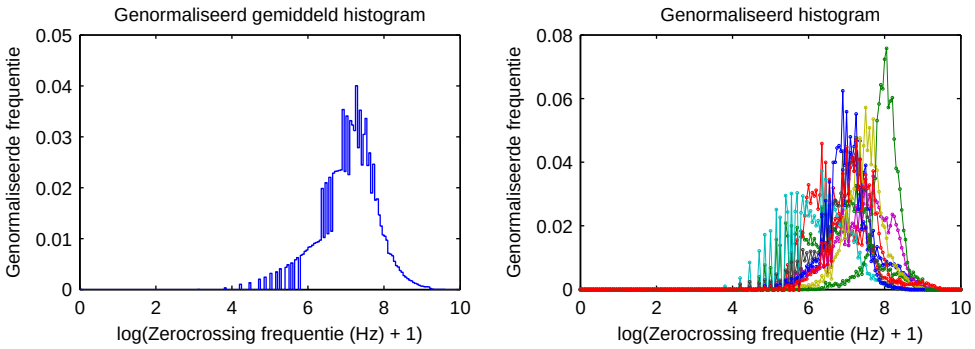
4.5.1.2 RMS

Empirisch bereik: $[0, 1]\, A_{RMS}$

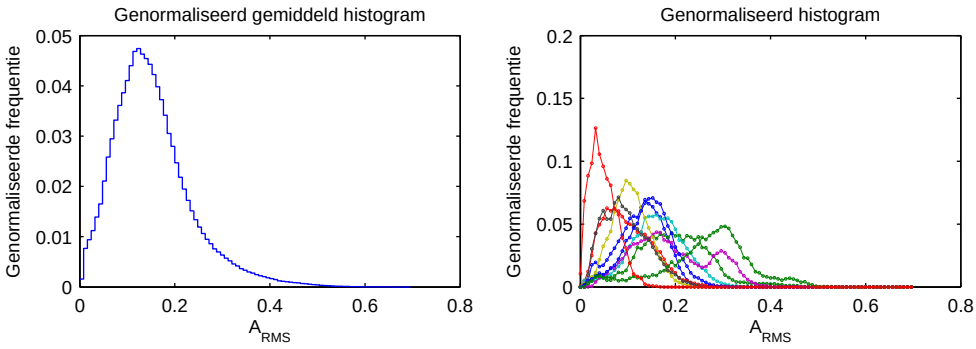
Schaling : /

Bereik histogram: $[0, 0.8]$

Bandbreedte h: 0.008



Figuur 4.3: Bereikanalyse van de zerocrossing rate, (links) genormaliseerd gemiddeld histogram en (rechts) het genormaliseerde histogram van enkele liedjes. De x-as geeft de geschaalde zerocrossingrate weer, de y-as de genormaliseerde frequentie. Er is duidelijk variantie van de specifieke liedhistogrammen ten opzichte van het gemiddelde histogram.



Figuur 4.4: Bereikanalyse van de Root Mean Square amplitude, (links) genormaliseerd gemiddeld histogram en (rechts) het genormaliseerde histogram van enkele liedjes. De x-as geeft de RMS waarde weer, de y-as de genormaliseerde frequentie. Er is tevens duidelijk variantie van de specifieke liedhistogrammen ten opzichte van het gemiddelde histogram.

4.5.1.3 Centroïde μ

Empirisch bereik: $[0, 22050]\ Hz$

Schaling : /

Bereik histogram: $[0, 20000]\ Hz$

Bandbreedte h: $50\,Hz$

4.5.1.4 Spreiding σ^2

Empirisch bereik: $\left[0, 11025^2\right]$

Schaling : $\log\left(\sigma^2 + 1\right)$

Bereik histogram: $[3, 8]$

Bandbreedte h: 0.02

4.5.1.5 Skewness γ_1

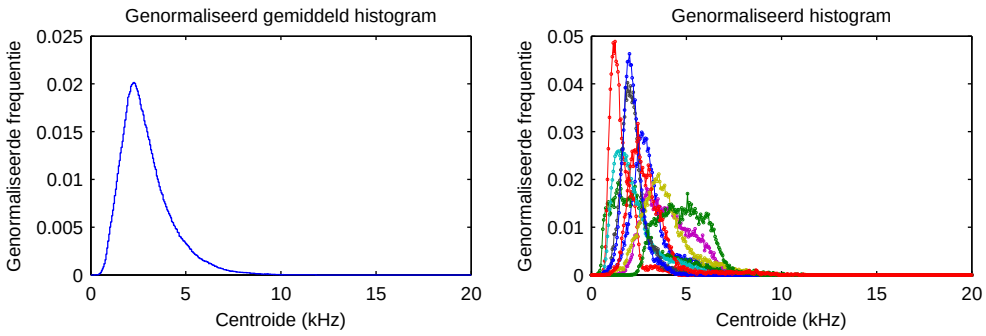
Empirisch bereik: /

Schaling[4] : $\log(\alpha)\ met\ \alpha = \begin{cases} \\ \alpha + 50000 & als \\ \alpha + 50000 \geq 1 \\ 1 & als \\ \alpha + 50000 < 1 \end{cases}$

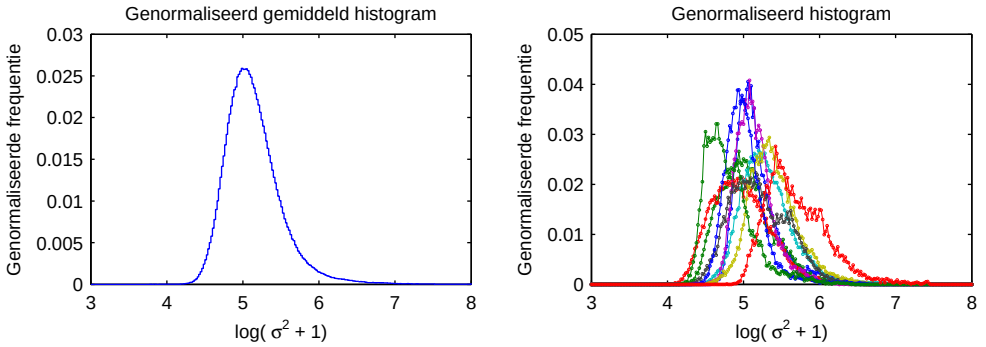
Bereik histogram: $[3, 8]$

Bandbreedte h: 0.02

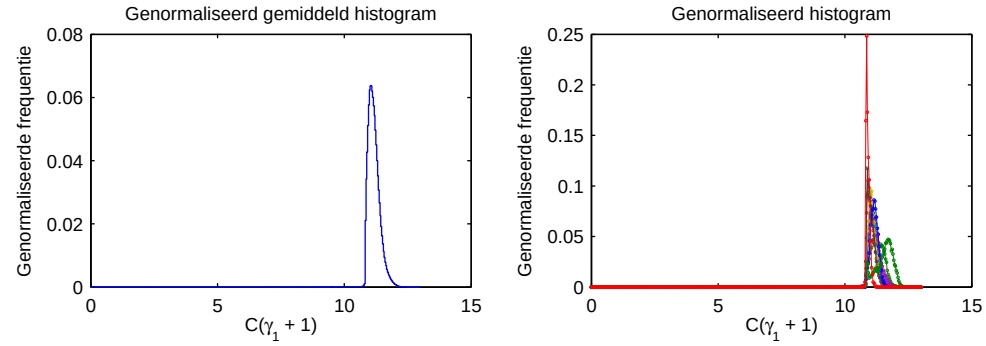
[4]De waarde -50000 is de laagste waarde tegengekomen in het testbereik en daarom gekozen als ondergrens



Figuur 4.5: Bereikanalyse van de spectrale centroide, (links) genormaliseerd gemiddeld histogram en (rechts) het genormaliseerde histogram van enkele liedjes.De x-as geeft de geschaalde centroïde weer, de y-as de genormaliseerde frequentie. De variantie van de specifieke liedhistogrammen ten opzichte van het gemiddelde histogram en het totale bereik is echter eerder beperkt.



Figuur 4.6: Bereikanalyse van de spectrale spreiding, (links) genormaliseerd gemiddeld histogram en (rechts) het genormaliseerde histogram van enkele liedjes. De x-as geeft de geschaalde spreiding weer, de y-as de genormaliseerde frequentie. De variantie van de specifieke liedhistogrammen ten opzichte van het gemiddelde histogram en het totale bereik is eerder beperkt.



Figuur 4.7: Bereikanalyse van de spectrale skewness, (links) genormaliseerd gemiddeld histogram en (rechts) het genormaliseerde histogram van enkele liedjes. De x-as geeft de geschaalde skewness weer, de y-as de genormaliseerde frequentie. Ten gevolge van het noodzakelijke grote bereik is de variantie eerder beperkt.

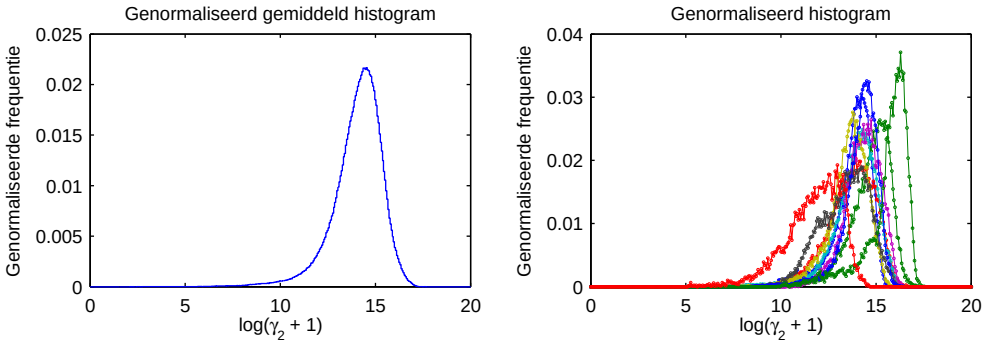
4.5.1.6 Kurtosis γ_2

Empirisch bereik: $[0, Inf]$

Schaling : $\log(\gamma_2 + 1)$

Bereik histogram: $[0, 20]$

Bandbreedte h: 0.0571



Figuur 4.8: Bereikanalyse van de spectrale kurtosis, (links) genormaliseerd gemiddeld histogram en (rechts) het genormaliseerde histogram van enkele liedjes. De x-as geeft de geschaalde kurtosis weer, de y-as de genormaliseerde frequentie. De specifieke liedhistogram wijken matig af van het gemiddelde spectrogram

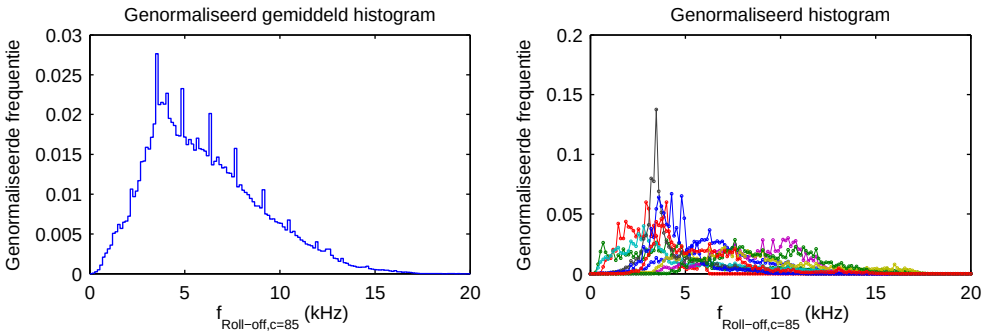
4.5.1.7 Spectrale roll-off

Empirisch bereik: $[0, 22050]$

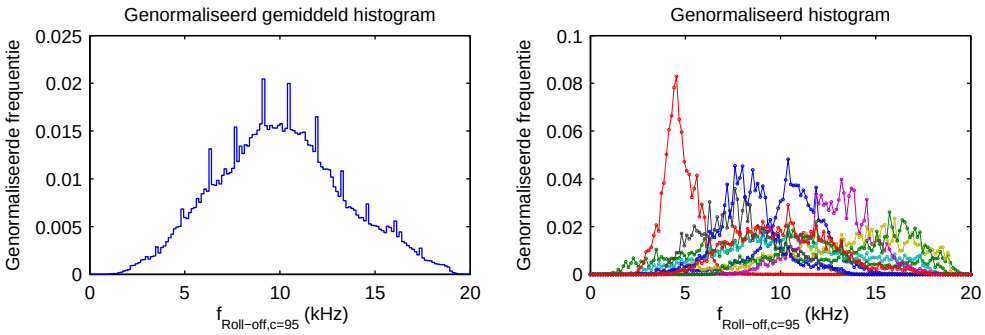
Schaling : /

Bereik histogram: $[0, 20]$

Bandbreedte h: 0.0571



Figuur 4.9: Bereikanalyse van de spectrale roll-off voor c = 0.85, (links) genormaliseerd gemiddeld histogram en (rechts) het genormaliseerde histogram van enkele liedjes.De x-as geeft de spectrale roll-off weer, de y-as de genormaliseerde frequentie. De specifieke liedhistogram wijken sterk af van het gemiddelde spectrogram.



Figuur 4.10: Bereikanalyse van de spectrale roll-off voor c = 0.95, (links) genormaliseerd gemiddeld histogram en (rechts) het genormaliseerde histogram van enkele liedjes. De x-as geeft de spectrale roll-off weer, de y-as de genormaliseerde frequentie. De variantie van de individuele liedhistogrammen is groter dan bij de spectrale roll-off met c =0.85

4.5.1.8 Spectrale helderheid

Empirisch bereik: $[0, 1]$

Schaling : /

Bereik histogram: $[0, 1]$

Bandbreedte h: 0.0077

4.5.1.9 Totale volume

Empirisch bereik: /

Schaling : /

Bereik histogram: $[0, 40000]$

Bandbreedte h: 266.6

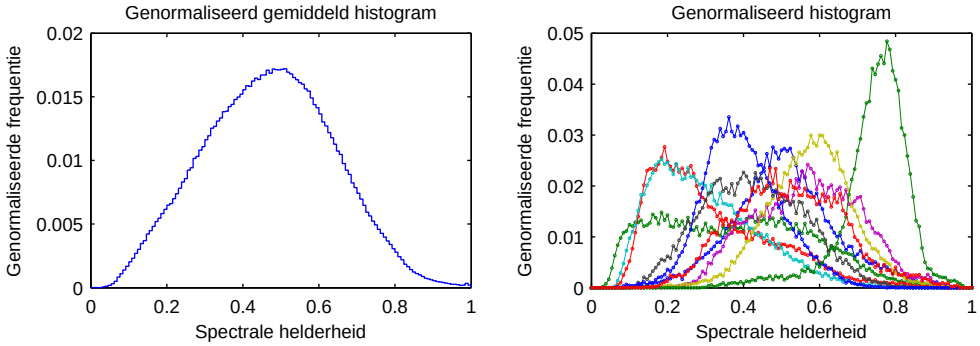
4.5.1.10 Perceptuele scherpte

Empirisch bereik: $[0, 60]$

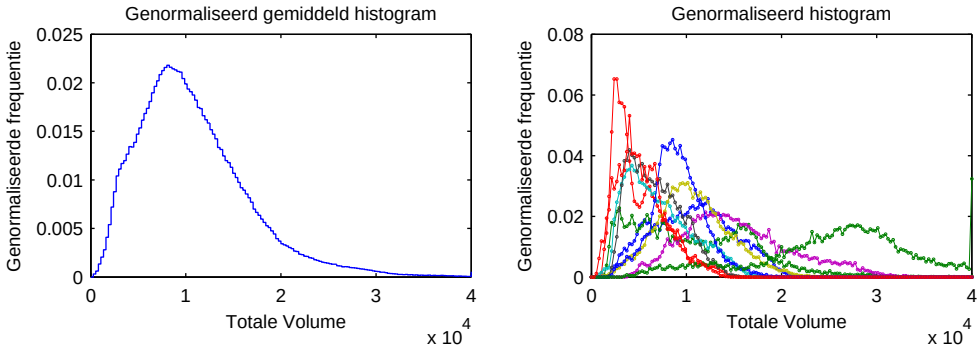
Schaling : /

Bereik histogram: $[0, 60]$

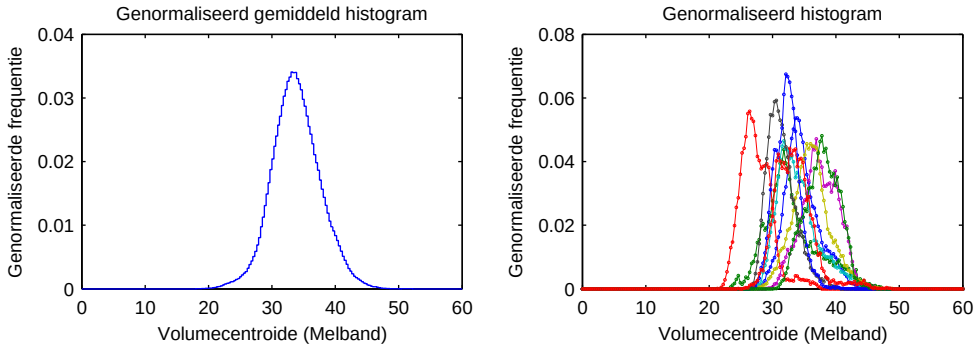
Bandbreedte h: 0.3



Figuur 4.11: Bereikanalyse van de spectrale helderheid, (links) genormaliseerd gemiddeld histogram en (rechts) het genormaliseerde histogram van enkele liedjes. De x-as geeft de spectrale helderheid weer, de y-as de genormaliseerde frequentie. Individuele liedhistogrammen vertonen een sterke variantie ten opzichte van elkaar en het gemiddelde histogram.



Figuur 4.12: Bereikanalyse van het totale volume, (links) genormaliseerd gemiddeld histogram en (rechts) het genormaliseerde histogram van enkele liedjes. De x-as geeft het totale volume weer, de y-as de genormaliseerde frequentie. Individuele liedhistogrammen vertonen een vrij sterke variantie ten opzichte van elkaar en het gemiddelde histogram.



Figuur 4.13: Bereikanalyse van de perceptuele scherpte, (links) genormaliseerd gemiddeld histogram en (rechts) het genormaliseerde histogram van enkele liedjes.De x-as geeft de perceptuele scherpte weer, de y-as de genormaliseerde frequentie. De liedhistogrammen varieren slechts vrij matig ten opzichte van elkaar en het totale bereik.

4.5.1.11 Perceptuele spreiding

Empirisch bereik: $[0, 1]$

Schaling : /

Bereik histogram: $[0.5, 1]$

Bandbreedte h: 0.0014

4.5.1.12 Spectrale vlakheid

Empirisch bereik: /

Schaling : /

Bereik histogram: $[0, 25]$

Bandbreedte h: 0.1412

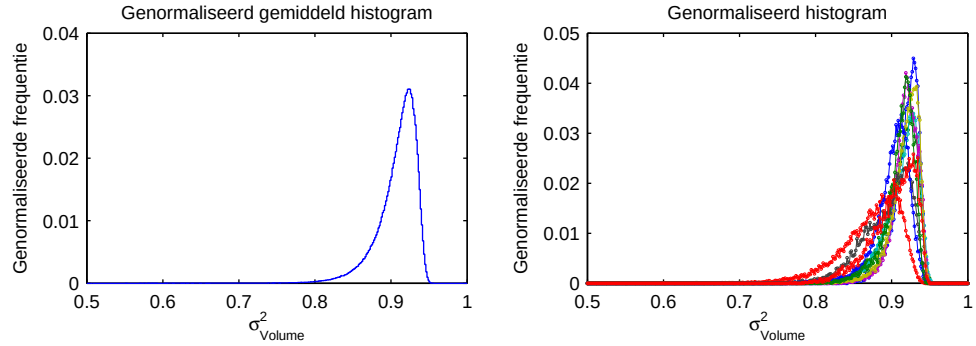
4.5.1.13 Spectrale entropie

Empirisch bereik: /

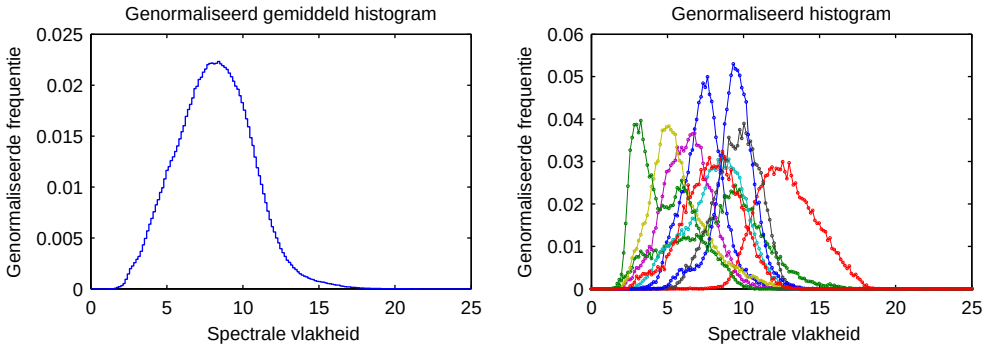
Schaling : /

Bereik histogram: $[0.2, 1]$

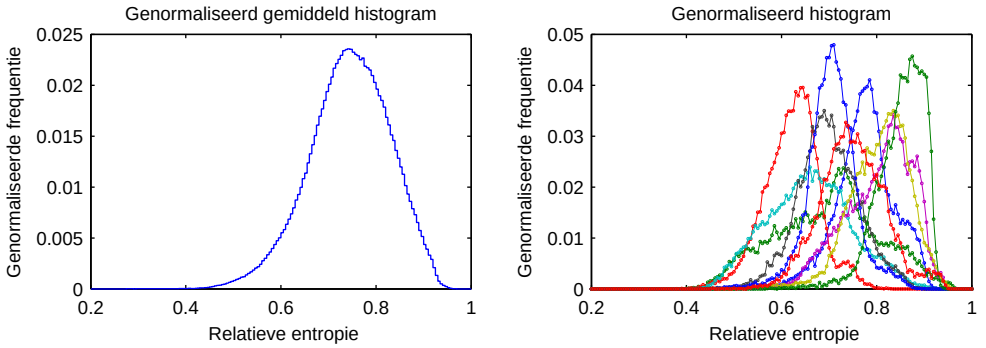
Bandbreedte h: 0.005



Figuur 4.14: Bereikanalyse van de perceptuele spreiding, (links) genormaliseerd gemiddeld histogram en (rechts) het genormaliseerde histogram van enkele liedjes. De x-as geeft de perceptuele spreiding weer, de y-as de genormaliseerde frequentie. Slechts weinig onderlinge variantie kan opgemerkt worden.



Figuur 4.15: Bereikanalyse van de spectrale vlakheid, (links) genormaliseerd gemiddeld histogram en (rechts) het genormaliseerde histogram van enkele liedjes. De x-as geeft de spectrale vlakheid weer, de y-as de genormaliseerde frequentie. Deze feature biedt een goede onderlinge variantie tussen de liedjes.



Figuur 4.16: Bereikanalyse van de spectrale entropie, (links) genormaliseerd gemiddeld histogram en (rechts) het genormaliseerde histogram van enkele liedjes. De x-as geeft de spectrale entropie weer, de y-as de genormaliseerde frequentie. Deze feature biedt een goede onderlinge variantie tussen de liedjes.

4.5.2 Cepstrale features

Om de datarate van de cepstrale features verder te verlagen, worden 3 stappen ondernomen. Als eerste stap wordt een lineaire principale componenten analyse uitgevoerd zodat de dimensie verlaagd kan worden. Deze toont dat er een zeer sterke eigenwaarde is die vrijwel uitsluitend door de 1e Mel-band verklaard wordt. Deze eerste Mel-band correleert met de spectrale energie, dewelke inderdaad in sterke mate onafhankelijk is van de andere banden. In experimenten in verband met timbre wordt deze term dikwijls verwaarloosd, maar hier is er voor gekozen om deze te behouden omwille van twee redenen. Een eerste reden is dat in de literatuur de meningen erover verdeeld zijn (o.a.[6]) dat deze band nuttig kan zijn bij classificatieexperimenten, met zowel voor als tegenstanders. Een tweede reden is dat de perceptuele impact van geluiden tevens mee bepaald wordt door de energie. Zo zal bijvoorbeeld een harde kickdrum in een lied een volledig andere impact hebben, en perceptueel met andere zaken geassocieerd worden, dan dezelfde maar zachtere drum op de achtergrond.

Om het aantal te weerhouden principale componenten te beslissen werd de impact van de eerste eigenwaarde dan ook genegeerd en werd gekozen om ongeveer 95% van de resterende variantie te behouden[5], wat resulteert in totaal aantal van 15 behouden principale componenten.

Deze analyse werd tevens voor dMFCC's en ddMFCC's uitgevoerd, waarvan de resultaten consistent zijn met die van MFCC's op een onderdrukte eerste eigenwaarde na. Verschillende venstergroottes tonen eveneens hetzelfde beeld.

Als tweede stap in de verwerking van (d)(d)MFCC's wordt een K-means clustering uitgevoerd. Alternatieven voor deze stap zijn o.a. vectorquantisatie, anchor space en modellering met een Gaussian Mixture Model. [6] toont echter dat k-means clustering vergelijkbare resultaten vertoont, terwijl de benodigde rekenkracht minder is dan die van GMM's en vectorquantisatie. Tevens is er in tegenstelling tot anchor space geen expliciete voorkennis nodig over de ruimte, de enige parameter is het aantal clusters, wat in een volgende stap geoptimaliseerd wordt.

De derde en laatste stap is de modellering van de opeenvolging van clusterindices, waartoe de (d)(d)MFCC's gereduceerd zijn, als Markov ketens.

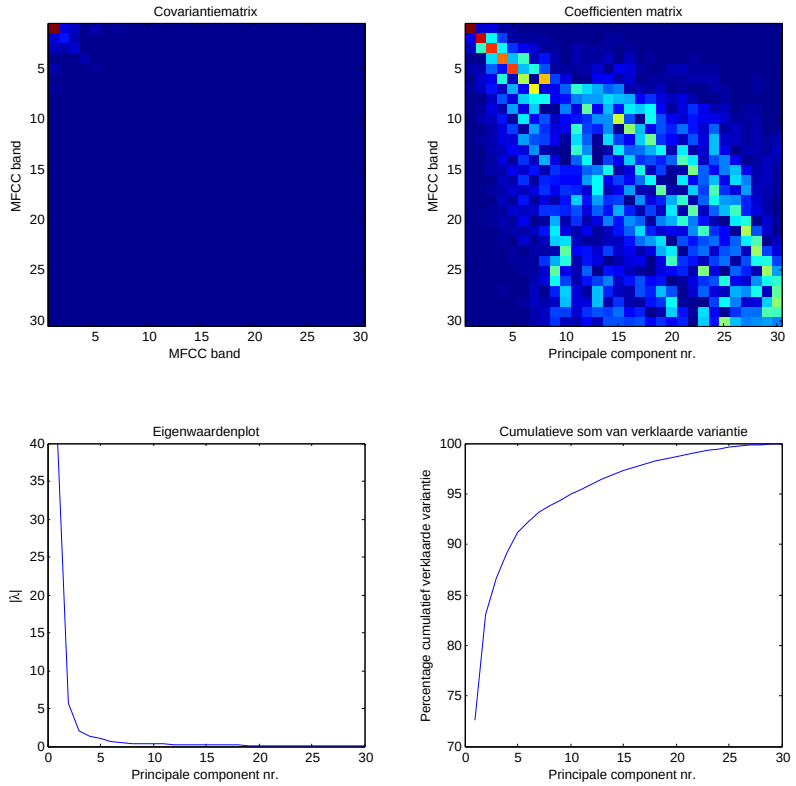
In deze twee stappen moeten nog twee parameters geoptimaliseerd worden. Dit wordt gedaan aan de hand van een vergelijkend classificatieexperiment op een subset (400 en 200 random gekozen trainings resp.testpunten) van de data. In navolging van [23] wordt een klein aantal clusterpunten gekozen en wegens computationele beperkingen voor lagere ordes van Markovketens.

De resultaten van dit experiment zijn onderstaand afgebeeld.

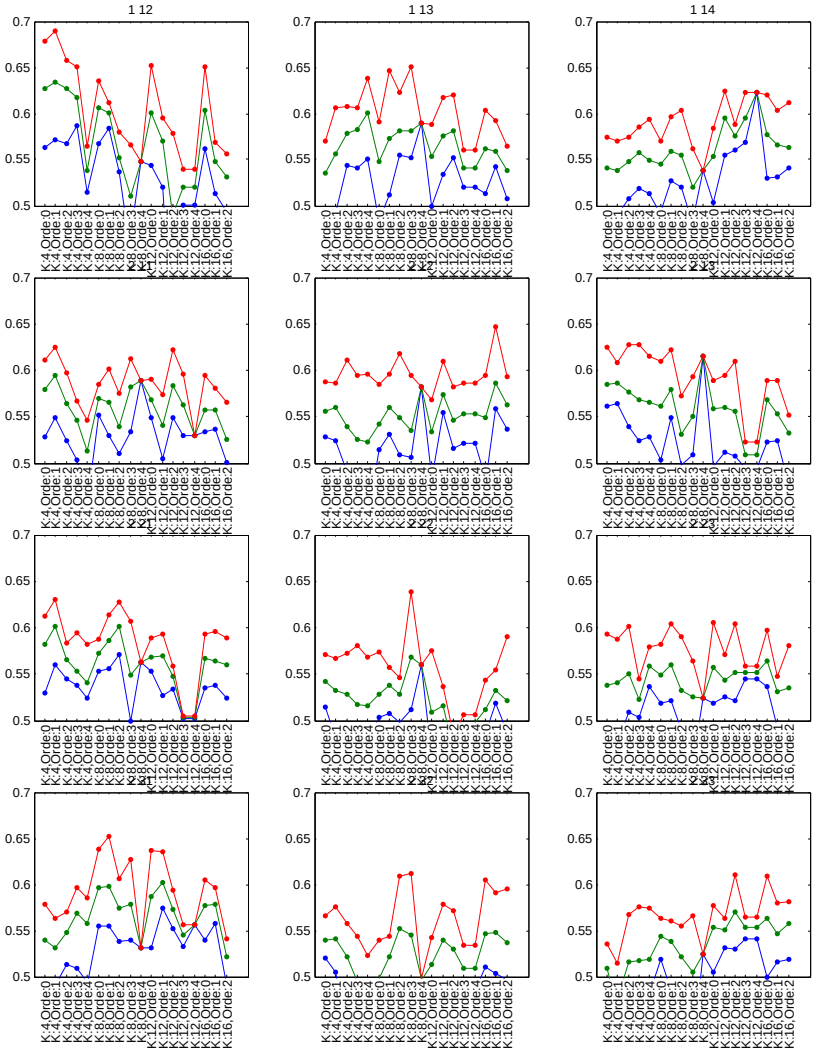
Uit het vergelijkend experiment zijn echter geen eenduidige resultaten te trekken, waarschijnlijk te wijten aan de relatief kleine subset en kleine aantal iteraties. Een algemene tendens is echter de meestal negatieve invloed van hogere ordes markovketens in de MFCC's , terwijl deze in (d)dMFCC's dikwijls een positieve invloed hebben. Voor het aantal clusters is ongeveer dezelfde conclusie te trekken. Opmerkelijk is ook dat de referentie implementatie iets beter presteert dan de verbeterde implementatie. Dit kan te wijten zijn aan een beter optimum in de clustering van de referentie implementatie of een gebiasede testset naar de implementatie van de MFCC's toe.

[5] Een aggresievere selectie zou ook mogelijk zijn maar er werd gekozen voor deze redelijk conservatieve schatting om eventuele niet lineaire verbanden niet te sterk te reduceren.

Wegens de niet eenduidige resulaten werd telkens voor elke (d)(d)MFCCde de beste clustergrootte voor het best presterende $0^e$of 1^e orde model geselecteerd, resulterend in een vector van 732 elementen.



Figuur 4.17: Principale componenten analyse van een MFCC feature; (linksboven) Covariantiematrix , (rechtsboven) Coefficientenmatrix van de gevonden principale componenten, (linksonder) Eigenwaarden van de gevonden componenten en (rechtsonder) de cumulatieve som van de verklaarde variantie



Figuur 4.18: Vergelijkende figuur van de classificatieperformantie op een kleine dataset van de verschillende markovmodellen van MFCC en d(d)MFCC's.
De x-as van geeft telkens het gebruikte markovmodel weer, namelijk de orde van het model en het aantal mogelijke staten K. De y-as geeft de ROCa classificatiescores weer. (rood) is de maximale score, (groen) de gemiddelde en (blauw) de laagste score.
De eerste kolom geeft modellen weer gebouwd met MFCC data, de tweede en derde respectievelijk van dMFCC en ddMFCC data. De rijen geven de verschillende implementaties weer. De eerste rij is de referentie implementatie (MFCC gebaseerd op SLaney), de tweede rij is de ontworpen MFCC implementatie voor een kleine venster, de derde rij is die voor een middelmatig venster en de laatste rij voor een groot venster.

Hoofdstuk 5

Internet features

5.1 Community metadata

Community metadata of culturele metadata is de verzamelnaam voor de extramuzikale gegevens die refereren naar een bepaalde artiest of een bepaalde song. Deze vormen een alternatieve bron van data om relaties tussen artiesten en songs te analyseren. Deze culturele metadata kan verschillende vormen aannemen. Enerzijds kan gebruik gemaakt worden van "harde" data die rechtstreeks relateren naar de artiest of het liedje, zoals datum van uitgave en genre informatie, anderzijds kan ook gebruik gemaakt worden van "zachtere" gebruiker gerelateerde informatie zoals afspeellijsten, reviews, tags en dergelijke. In deze thesis wordt gebruik gemaakt van 5 soorten data namelijk; datum van uitgave, genre (2), afspeellijsten, tags en google similarity afstand. De eigenschappen van deze worden in volgende secties besproken. Andere internetfeatures kunnen gevonden worden in [31, 32]

Een minpunt van de internetgebaseerde features is dat deze slechts gesbaseerd zijn op een momentopname, wat voor enkele problemen zorgt. Een eerste probleem is dat de beschikbare informatie groter is voor recentere en populairdere muziek en artiesten. Een tweede probleem is dat de internetfeatures hierdoor "helderziend" kunnen worden in de zin dat voor oudere liedjes de informatie over het al dan niet hit geweest zijn reeds aanwezig is. (Men zou immers simpelweg het archief van de hitlijsten kunnen doorzoeken). Een laatste probleem is het gebrek aan initiële informatie voor nieuwe artiesten en liedjes, doch dit probleem is niet relevant voor het beoogde onderzoek. Deze problemen beperken de rechtstreekse bruikbaarheid van de internetfeatures voor hitclassificatie, maar gebruikt in samenspraak met andere features kunnen deze eventueel wel bijdragen tot de modellering van het concept "hit".

5.1.1 Datum van uitgave

Het tijdstip waarop het lied gemaakt en geïntroduceerd is waarschijnlijk een van de meest relevante culturele kenmerken mogelijk, en wordt bijgevolg ook geïncorporeerd als feature.

5.1.2 Genre

Genres geven een ruwe categorisatie van gelijkaardig klinkende muziek. De exacte notie van gelijkaardige muziek is gecontesteerd, wat resulteert in een constant evoluerende taxonomie van verschillende genres en subgenres waarvan de noties elkaar veelvuldig overlappen. Een eenduidig classificatiesysteem is dus onbestaande maar ondanks deze onduidelijkheid is genre wel een voor mensen vrij natuurlijk concept om muziek te beschrijven. Wegens deze natuurlijkheid en bruikbaarheid van het genreconcept is de classificatie van muziek in genres dan ook veelvuldig onderzocht in MIR.[21, 32, 25]

5.1.3 Tags en afspeellijsten

Tags en afspeellijsten zijn twee vormen van gebruiker gegenereerde data die middels collaboratief filteren gecondenseerd kunnen worden tot betekenisvolle extramuzikale informatie. Tags enerzijds zijn labels die gebruikers kunnen geven aan muziek, artiesten, verzamelingen, ... en afspeellijsten anderzijds zijn de lijsten van de luistergewoontes van gebruikers. Omdat deze ruwe data gebruikerspecifiek is, is deze slecht generaliseerbaar, maar door deze data voor vele gebruikers te verzamelen en te vergelijken kunnen meer algemeen geldende relaties en labels afgeleid worden.

5.1.4 Google similarity distance

De google similarity distance is een methode gedefinieerd in [33] om de gerelateerdheid van twee of meer begrippen te meten over (een deel van) het wereldwijde web. De stelling uit \Rudi07 is dat de hoeveelheid mensen die bijdragen aan het wereldwijde web zo groot is, en zo gevarieerd is dat de zoekmachinekans van een frase, gedefineerd als het aantal resultaten dat een zoekmachine voor deze frase vind ten opzichte van de grootte van het web, de actuele kans van die frase in de maatschappij benaderd. Deze kans, de genormaliseerde google afstand, definiëren zij als volgt:

$$NGD\left(x,y\right)=\frac{max\left\{\log G(x),\log G(y)\right\}-\log G\left(x,y\right),}{\log N-min\left\{\log G(x),\log G(y)\right\}}$$

met $G(z)$de googlefunctie, het aantal resultaten dat google vindt voor de zoekterm z, $G(z,u)$ de googlefunctie op zoekterm z AND u (dus "z" + "$\ddot{u}$" in google) en N de grootte van het doorzochte corpus (in casu dus de grootte van het wereldwijde web volgens google, $\pm$20 miljard pagina's .

Toegepast op muziek geeft de NGD dus de mogelijkheid om relaties tussen artiesten en hun liedjes onderling te kwantificeren.

5.2 Implementatie

Voor het verkrijgen van internetfeatures werden vier bronnen geraadpleegd; hitlijsten, LastFM, FreeDB en Google ter bekomen van in totaal 5 internetfeatures namelijk datum van uitgave,

tags, genre, similarity, NGD.

5.2.1 Datum van uitgave

Dit feature wordt uit de eerder vermelde /Ref hitlijsten gehaald. Als datum van uitgave wordt die datum gekozen waarop het lied het meest relevant is, m.a.w de week waarin het de hoogste notering in de hitlijsten behaalde. De feature wordt uitgedrukt in het aantal weken ten opzichte van een centrale referentie, namelijk 1 januarie 1950.

5.2.2 LastFM Toptags

De LastFM Toptags informatie wordt bekomen door voor elke artiest in de database via de LastFM API de meest populaire tags te downloaden. De bekomen informatie is per artiest een set $\{t_{k,}, w_{t_k}\}_{artiest}$van bekomen taglabels t_k en hun relevantie w_{t_k} (met $w_{t_k} \in]0, 100]$). De taginformatie zelf heeft, in LastFM's web 2.0 geest, een zeer breed bereik, zoals bijvoorbeeld: genre-info (rock, metal, ...), emotie-info (romantic, love, favourites, ...), geografische informatie, producer info, ...

Twee problemen stellen zich echter met deze data. Een eerste punt is dat deze niet voor elke artiest voorhanden is. Zo is voor de 1010 unieke artiesten in de database, slechts voor 408 hiervan taginformatie beschikbaar. Dit is waarschijnlijk te wijten aan de brede opzet van de database in combinatie met het relatief jonge bestaan van LastFM.

Het tweede probleem dat zich stelt met de bekomen data is dat deze sterk vervuild is, wegens het volledig user-generated zijn van de tags zonder enige vorm van controle. Dit uit zich in tags die enkel verschillen in spelling, overgespecifieerde tags en nonsenstags. Tabel 5.2.2 illustreert dit.

Het effect van deze vervuiling is, naast ruis, dat de beschikbare informatie uitgesmeerd wordt over de verschillende tags zodat de minder belangrijke tags nog sterk bijdragen aan het totale volume aan tags, zoals geïllustreerd in figuur 5.2.2. Dit maakt dat de staart van de verdeling niet zonder meer verwaarloosd kan worden (de 100 belangrijkste tags zijn bijvoorbeeld slechts representatief voor 35% van het totale volume).

Aangezien het aantal tags te groot is om alle fouten te corrigeren, wordt een meer pragmatische methode gekozen om de foutenlast te remediëren. Zo worden allereerst alle tags die slechts eenmaal voorkomen verwijderd, zodat het totale corpus van 10120 tags gereduceerd wordt tot 3426. Aan elke unieke tag T_i wordt vervolgens een gewicht B_{T_i} toegekend met

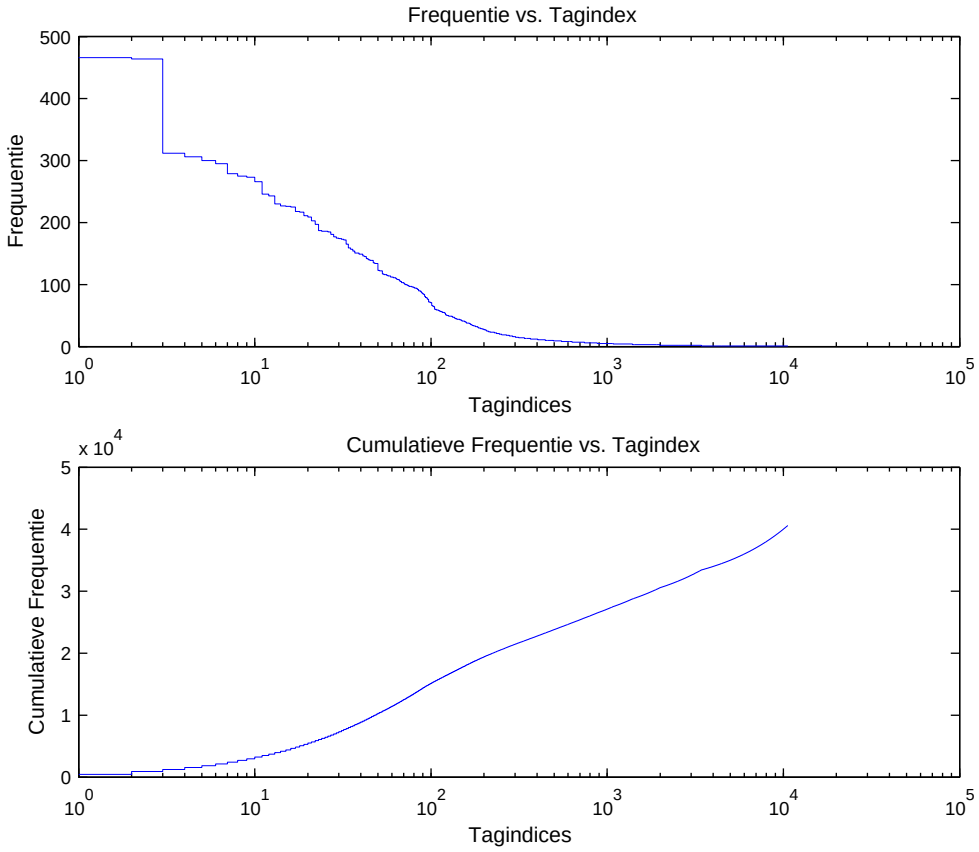
$$B_{T_i} = IDF^{-1} = \left(\log \frac{|A|}{|\{a_{j:} : T_i \in a_j\}|} \right)^{-1}$$

met $|A|$ het aantal artiesten in de corpus (408) en $|\{a_{j:} : T_i \in a_j\}|$ het aantal sets waarin de term T_i voorkomt. De feature vector voor elke artiest wordt dan gegeven door:

$$tagfeature(artiest) = \{f_i\}_{i=1}^{i=|T_i|}, \; met \; f_i = \sum w_{t_k} B_{T_i} : t_k \approx T_i \; \forall k, i$$

Tabel 5.1: Overzicht van de tags gebruikt op LastFM

Top 20 LastFMtags	Laagste 20 LastFMtags
pop	always listening never getting tired
Hip-Hop	daniel miller
rock	Hard Rock Artists
indie	seen -em
jazz	summer music
House	ann arbor
dance	Mood Lift
Progressive rock	tagged artist
rnb	with female singers
breakbeat	rlz
classic rock	legal drug
trance	troya2
electronic	Flaw
rap	polish
female vocalists	WorD
Disco	ALL metal tracks
alternative	auto-music
funk	seen at Pinkpop
punk rock	barely adequate



Figuur 5.1: Analyse van het corpus van LastFM tags. (boven) De frequentie van voorkomen van elk taglabel versus de tagindices, gerangschikt volgens frequentie en (onder) de cumulatieve frequentie van de tagindices gerangschikt volgens frequentie.

waarbij gebruik gemaakt wordt van inexacte tagvergelijkingen. Deze inexacte vergelijking heeft een dubbel effect, namelijk het remediëren van eenvoudige inconsistenties in spelling en het herdistribueren van te specifieke (genre) informatie over algemene categorieën. Zo zullen enerzijds categorieën als ('rock n roll', 'rock&roll', 'rock and roll') samengevoegd worden en anderzijds tags als 'punk rock' zowel bijdragen tot 'punk', 'rock' als tot 'punkrock'.

5.2.3 ID3 genre & FreeDB genre

De bron van zowel het ID3 als FreeDB genrefeature is de FreeDB database die gebruikt werd voor het importeren van muzikale metadata. De incorporatie van twee verschillende genrevelden in deze database is historisch gegroeid en deze hebben ook niet noodzakelijk dezelfde waarde. De eigenschappen van beide zijn ook verschillend en daarom worden deze als aparte features behandeld.

Tabel 5.2: FreeDB genres en hun frequentie in de classificatiedatabase

FreeDB Genre	Frequentie
' '	179
Blues	33
Classical	26
Country	22
Data	6
Folk	7
Jazz	7
Misc	933
Newage	44
Rock	589
Soundtrack	15

5.2.3.1 FreeDB genre

Het FreeDB genre is een genrelabel uit een set van elf mogelijke kandidaten, dat voor elke cd door gebruikers wordt toegekend. Deze beperkte genretaxonomie heeft als voordeel dat de informatie compact is, en wegens de welomlijnde genres ook meestal correct. Het FreeDB genre heeft echter wel 2 belangrijke minpunten. Zo is het zoals vermeld per cd gedefinieerd, wat voor sterk genreoverschrijdende artiesten en verzamelcd's een probleem vormt. Tevens wordt het label door FreeDB ook gebruikt voor interne databaseclassificatie. Dit laatste werd problematisch toen bepaalde genres (voornamelijk het Rock genre) te veel cd's bevatten voor de gebruikte database[1], waardoor nieuwe cd's noodgedwongen verkeerdelijk gelabeld moesten worden. Een overzicht van de FreeDB genres, samen met hun frequentie van voorkomen in de database wordt gegeven in tabel 5.2.3.1[2] [3]. De enkele 'data' classificaties in de database zijn waarschijnlijk door voorgaande te verklaren, en de overvloed aan 'Misc' is, naast eventueel de voorgaande reden, te wijten aan het (bewust) hoge aantal verzamelcd's in de database.

5.2.3.2 ID3 genre

Het ID3 genre label is in tegenstelling tot het FreeDB label een vrij in te vullen label[4], wat zoals eerder vermeld enerzijds tot meer informatie maar dikwijls ook tot meer ruis leidt. In de labels aanwezig in de database waren dan ook enkele inconsistenties, maar gezien het kleine aantal gebruikte labels (47) werden deze handmatig gecorrigeerd tot de bekomen set van 38 labels in tabel 5.2.3.2.

De gecorrigeerde labels zijn echter nog niet allen even zinvol, en daarom worden de categorieën {'1991' 'Ballad' 'Crossover' 'Primus' 'Top 40'} samengevoegd met {'Other'}, {'Hard Rock'

[1] Naast de implementatieproblemen is er ook nog een intrinsiek probleem met de uniekheid van de gebruikte cd-identifier, de TOC, waardoor collisions tussen verschillende cd's kunnen optreden en er ook noodgedwongen verkeerdelijk gelabeld moet worden.

[2] Het ' ' staat voor de afwezigheid van FreeDB genredata

[3] De 11e, hier afwezige, categorie is reggae

[4] Hoewel in ID3v1 dit nog beperkt was tot een lijst van 255 labels, werd deze restrictie in latere versies opgeheven.

Tabel 5.3: Bewerkte ID3 genres en hun frequentie in de classificatiedatabase

ID3 genre	freq.	ID3genre	freq.
' '	275	Hard Rock	1
1991	11	Indie	7
Acid Jazz	3	New Age	2
Alternative	83	New Wave	5
Alternative Rock	19	Oldies	522
Ballad	1	Other	71
Blues	10	Pop	245
Britpop	5	Pop,Funk	11
Classic Rock	14	Primus	6
Crossover	1	Progressive Rock	12
Dance	110	R&B	16
Disco	5	Rap	4
Easy Listening	5	Rock	263
Electronic	7	Rock & Roll	26
Folk,Rock	1	Rock,Dance	5
Funk	8	Soul	9
Garage Punk	2	Synthpop	19
General Pop	7	Top 40	35
Grunge	5	sixties	30

'Folk,Rock'} met {'Rock'} en {General Pop} met {Pop}. Gezien de overlap en spaarsheid van sommige genres wordt er op die genres nog een gewogen inexacte matching toegepast om de informatie gedeeltelijk naar algemenere categorieën te distribueren.

De ID3 genre feature wordt gevormd door:

$$
\begin{aligned}
ID3genre(song) &= \{f_i\}_{i=1}^{i=|T_i|},\ met\ f_i = 0.5C(t,T_i) + 0.5\sum C'(t,T_i), \\
&\quad met \begin{cases} C(t,T_i) = 1 & als\ t = T_i \\ C'(t,T_i) = 1 & als\ t \approx T_i \end{cases}
\end{aligned}
$$

5.2.4 Artist Similarity

De bron similarity informatie is afkomstig van de categorie 'similar artists' van LastFM, die deze zelf berekent op basis van het aantal malen samen voorkomen van artiesten in de afspeellijsten van gebruikers. De data kan via de LastFM API bekomen worden, maar querying met de 1010 unieke artiesten uit de database leverde slechts voor 110 artiesten informatie op. Door het gebrek aan een voldoende grote dataset werd deze feature daarom niet weerhouden.

5.2.5 Google Distances

De normalised google distance wordt gebruikt om de mate van overeenkomst te meten tussen 2 begrippen. Gegeven de database met artiest, titel, genre, ... informatie kunnen veel gegevens gemeten worden. Een mogelijke benadering zou erin kunnen bestaan om, vergelijkbaar met anchor space modellen, de database te vergelijken met vooraf bepaalde begrippen, om op deze basis voorkennis te integreren. Deze aanpak werd hier niet gekozen, wegens dezelfde reden als bij de featureverwerking, het gebrek aan echte voorkennis en de beperkingen die deze zou meebrengen.

Daarom werd besloten van de database an sich uit te gaan en de onderlinge genormaliseerde google afstanden tussen de artiesten en de liedjes in de database te berekenen.

Zoals de naam impliceert wordt de data bekomen door google te querien[5], maar aangezien het aantal benodigde queries echter kwadratisch toeneemt met het aantal liedjes, werd echter besloten om slechts data te verzamelen voor een subset van de database, meer bepaald die liedjes waarvoor ook een tagfeature voor bestaat. De volgende afstanden werden berekend

$$
\begin{aligned}
NGD_{artiest}(i,j) &= NGD\left((artiest_i, artiest_j) +' music'\right) \\
NGD_{titel}(i,,j) &= NGD\left((titel_i + artiest_i, titel_j + artiest_j) +' music'\right) \\
NGD_{LastFM,artiest}(i,j) &= NGD\left((artiest_i, artiest_j) +' site : www.last.fm'\right) \\
NGD_{LastFM,titel}(i,,j) &= NGD\left((titel_i + artiest_i, titel_j + artiest_j) +' site : www.last.fm'\right)
\end{aligned}
$$

Het bijvoegen van de 'music' specificator in google heeft als doel om de zoekresulaten te beperken tot diegene die gerelateerd zijn met muziek, aangezien vele titels en artiestennamen ook andere niet-muzikale gebruiken hebben.

De LastFM afstanden worden berekend over LastFM heen. Dit heeft als voordeel dat google tevens de afspeellijsten en hitlijsten van de gebruikers doorzoekt, zonder dat deze gebruikers expliciet gespecifieerd moeten worden of hun gegevens apart gedownload worden.

[5]Hoewel andere zoekrobotten natuurlijk ook mogelijk zijn \Ref

Hoofdstuk 6

Classificatie-experimenten

Deze sectie geeft een overzicht van de classificatieexperimenten en hun resultaten. Allereerst wordt de opbouw van de experimenten besproken en de manier waarop de resultaten geëvalueerd worden. Hierna worden de uitgevoerde experimenten en hun resultaten besproken.

6.1 Opbouw van de experimenten

6.1.1 Dataset

De dataset bestaat uit de 1861 unieke liedjes die gerefereerd konden worden naar de gebruikte hitlijst. Deze data werd vervolgens verdeeld in twee klassen, hits en niet hits, op basis van de hoogste positie die de liedjes haalden in de hitlijst. De grens voor hits is een gecontesteerd begrip, maar mede door de verdeling van de klassen zoals getoond in figuur 3.3, werd de grens vastgelegt op de top 10.

Deze dataset werd random verdeeld in een trainingsset van 1240 liedjes en een testset van 621 liedjes.

6.1.2 Performantie evaluatie

Om de performantie van de classifier te evalueren wordt gebruik gemaakt van 2 criteria, enerzijds de classificatiescore en anderzijds de oppervlakte onder de ROC-curve (ROCa). De misclassificatiescore is de ratio van het aantal misclassificaties in de testset en de grootte van de testset.

$$misclassificatiescore = 1 - \frac{\#misclassificaties}{\#grootte\,testset}$$

De ROCa score wordt bekomen door de oppervlakte onder de ROC curve te meten. De ROC of Receiver Operator Characteristic is een plot van de sensiviteit versus de 1-specificiteit voor een binaire classifier. De sensiviteit en 1 specificiteit zijn gedefinieerd als:

$$
\begin{aligned}
sensiviteit &= \frac{\#echte\,positieven}{\#echte\,positieven + \#valse\,negatieven} \\
1 - specificiteit &= \frac{\#echte\,negatieven}{\#echte\,negatieven + \#valse\,positieven}
\end{aligned}
$$

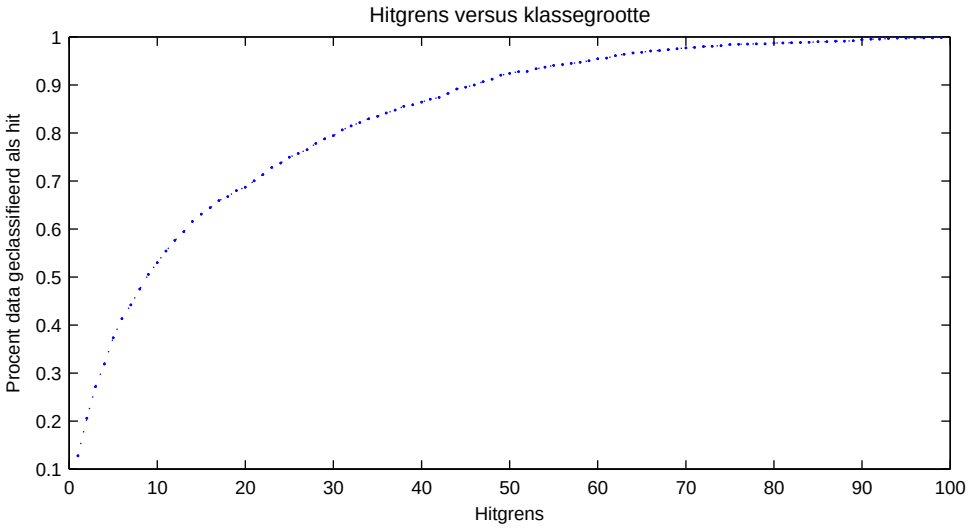
Door de threshold van de classifier te variëren bekomt men vervolgens een curve. De oppervlakte onder deze curve is een indicatie van de performantie van de classifier. Een ROCa van 0.5 betekent dat de classifier niet beter is dan de random classifier, een ROCa van 1 staat voor de perfecte classifier[1].

Om betrouwbaarheidsinformatie te bekomen voor de classifiers werd de data random herverdeeld in een test en trainigsset waarna de performantie op deze sets gechecked werd. Door dit enkele keren te herhalen kan de betrouwbaarheid van het resultaat ingeschat worden.

6.1.3 Classificatiesoftware

Voor experimenten met LS-SVM's werd gebruik gemaakt van de LS-SVMlab1 toolbox, en voor experimenten met reguliere SVM' werd gebrui gemaakt van LIBSVM met de matlabinterface.

[1]Indien de ROCa kleiner is dan 0.5 zou dit betekenen dat de classifier slechter is dan de random classifier, wat voor binaire classificatie natuurlijk niet zinnig is. Door de klassen om te wisselen bekomt men dan ook de echte performantie.



Figuur 6.1: Verdeling van de data

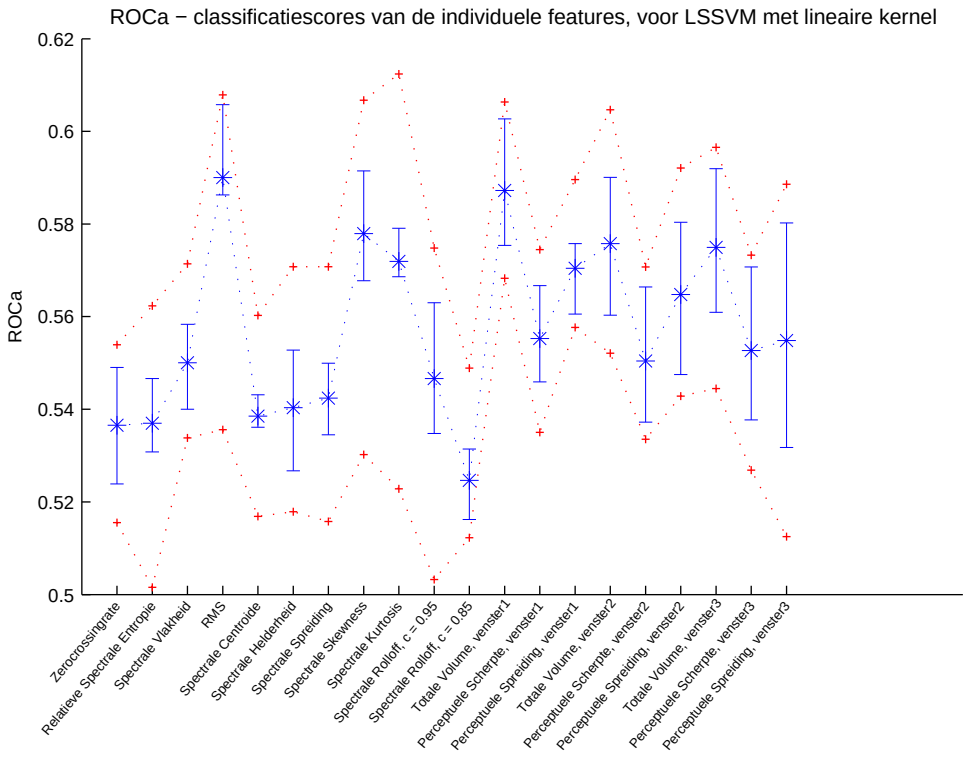
6.2 Classificatieexperimenten

6.2.1 Evaluatie van de audiofeatures

6.2.1.1 Evaluatie van de temporele en spectrale features

Een evaluatie werd uitgevoerd voor de bepaling van de performantie van de temporele en spectrale audiofeatures. De resultaten worden weergegeven in figuur 6.2.

Uit deze evaluatie blijkt dat de features die de verdeling van de geluidssterkte modelleren, RMS en Totale Volume, enerzijds en de hogere orde statistische spectrale features, Spectrale Skewness en Kurtosis, anderzijds, de beste performantie bieden. De geboden performantie ligt echter lager dan deze van de cepstrale features.



Figuur 6.2: Vergelijking van de ROCa classificatiescores van de verschillende features: (blauw) het interkwartiel bereik, met de gemiddelde waarde gemarkeerd door een asteriks, (rood) weergave van het volledige bereik.

6.2.1.2 Evaluatie van de temporele en spectrale features voor verschillende kernels

Sinds de temporele en spectrale features gemodelleerd werden als distributies, werd gekeken naar andere kernels die een betere afstandsmaat hiervoor vormen. Als afstandsmaat tussen

empirische distributies $P(i)$ en $Q(i)$, met kansdichtheid $p(i)$ resp. $q(i)$, worden de volgende 3 beschouwd.

- L1-norm

$$\| P, Q \|_1 = \sum |p(i) - q(i)|_i \tag{6.1}$$

- Kullback–Leibler divergentie

$$D_{KL}(P, Q) = \sum_i p(i) \log \frac{p(i)}{q(i)} \tag{6.2}$$

- De Kolmogorov-Smirnov teststatistiek

$$D_{KS}(P, Q) = \sup_i |P(i) - Q(i)| \tag{6.3}$$

De L1 norm is een volwaardige norm en voldoet als kernelfunctie aan de Mercer condities zodat deze rechtstreeks als kernel gebruikt kan worden. De KL-divergentie en KS-teststatistiek zijn echter geen volwaardige norm. Daarom wordt de bekomen afstandsmatix A getransformeerd tot een geldige kernelmatrix. De gebruikte transformatie is $\Omega = AA^T$. Gegeven een set datapunten $\mathcal{A}$ met distributiefeature $P_k(i)$komt deze transformatie overeen met het toepassen van een lineaire kernel op de data met getransformeerde featurevector F_k :

$$F_k = \{D(P_k, P_i)\}_{i=1}^{i=|\mathcal{A}|} \tag{6.4}$$

Er kon niet voldoende data verzameld worden om dit aspect grondig te evalueren maar voorlopige resultaten geven aan dat het gebruik van de Kolmogorov-Smirnov teststatistiek en de L1 norm de classificatieperformantie positief beinvloeden. De kernel gebaseerd op de Kullback–Leibler divergentie gaf daarentegen weinig meerwaarde.

6.2.1.3 Evaluatie van SVM versus LSSVM voor audiofeatures

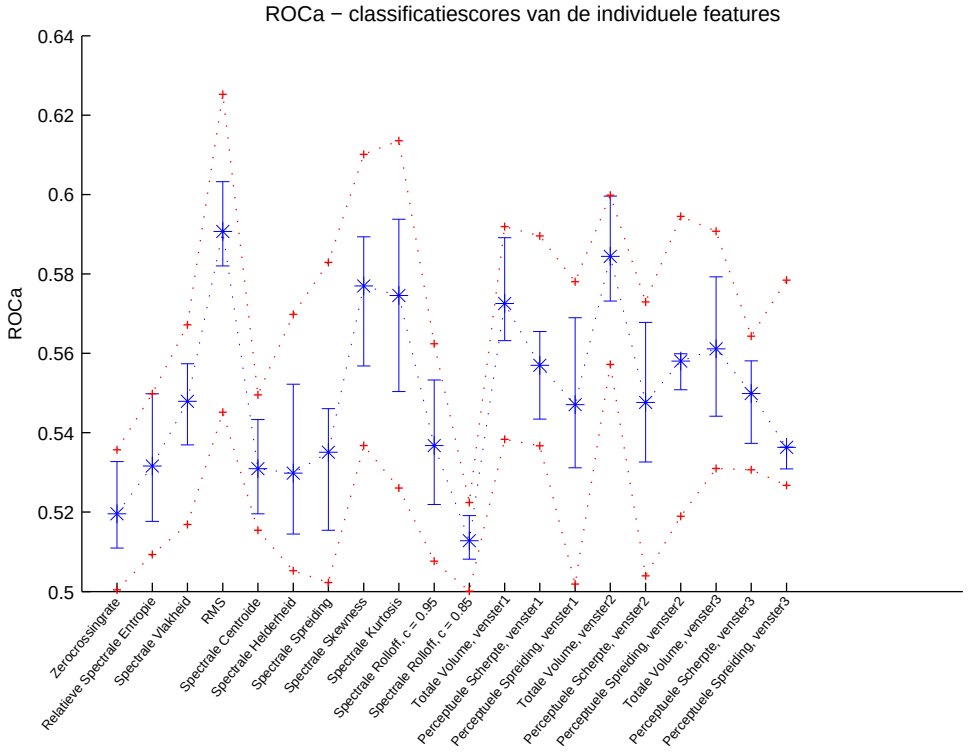
Om eventuele verschillen SVM en de LSSVM implementatie werden de audiofeatures ook geevalueerd met de LIBSVM. De resultaten hiervan worden getoond in figuur6.3.

Als we deze performantie vergelijken met die van LSSVM dan zien we dat deze slechts weinig verschilt. LSSVM presteerd soms iets beter, maar dit is waarschijnlijk te verklaren door de betere modelselectie algoritmen in de LSSVM implementatie.

6.2.2 Evaluatie van de communitymetadata-features

Sinds de community metadata-features slechts relevant zijn bij gecombineerd gebruik met audiofeatures worden deze samen met deze geëvalueerd. Om de berekeningen te beperken worden echter niet alle audiofeatures gebruikt maar wordt gekozen voor de meest performante op zich staande audiofeature, namelijk de de referentie markov-feature.

We zien in de figuur duidelijk dat gemiddelde waarden van de evaluaties met internetfeature hoger liggen dan deze van de referentie.



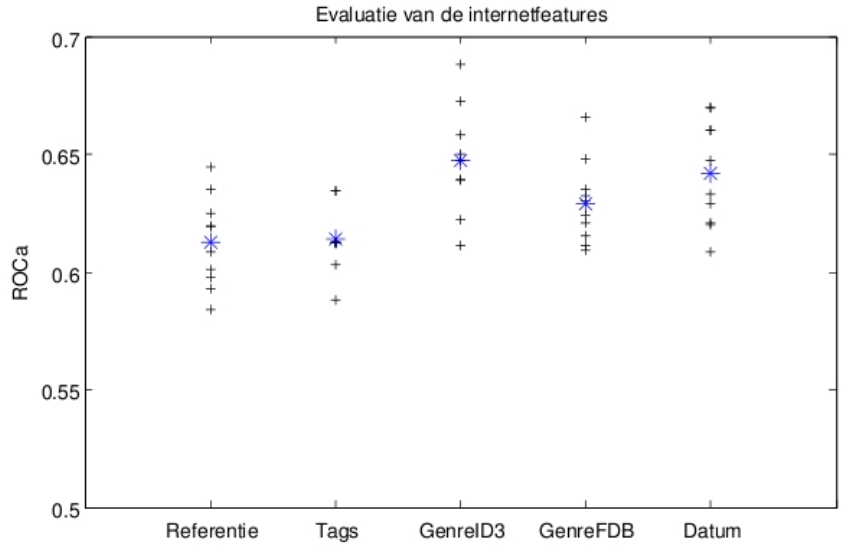
Figuur 6.3: Vergelijking van de ROCa classificatiescores van de verschillende features met LIBSVM: (blauw) het interkwartiel bereik, met de gemiddelde waarde gemarkeerd door een asteriks, (rood) weergave van het volledige bereik.

6.2.3 Evaluatie van de gecombineerde features

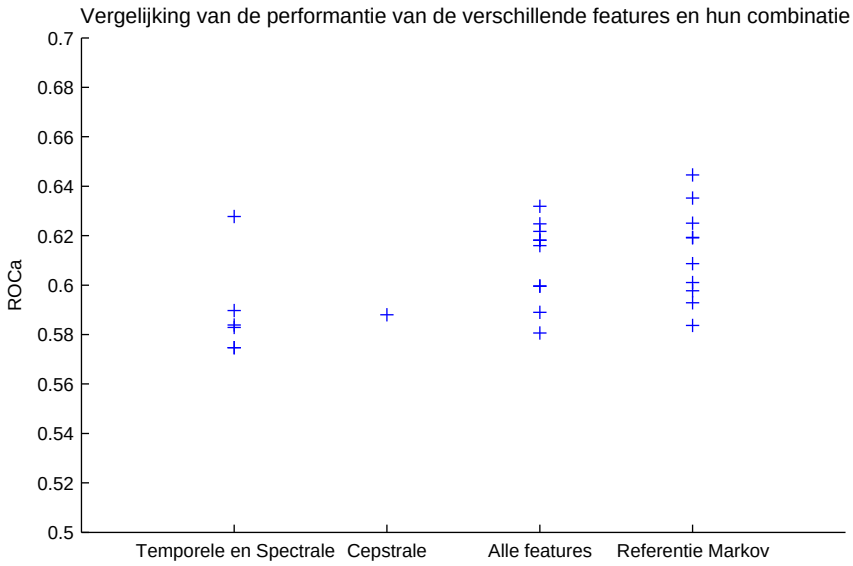
In dit experiment worden alle features samen beschouwd. De resultaten worden weergegeven in figuur 6.5. Opvallend is dat de combinatie van de features minder goed scoort dan de individuele markovfeatures of deze gecombineerd met de internetfeatures. Dit is waarschijnlijk op te lossen met extra tuning van de feature vector zodat de belangrijke features niet onderdrukt worden door de minder belangrijke.

6.2.4 Evaluatie van de classificatie voor verschillende hitgrenzen

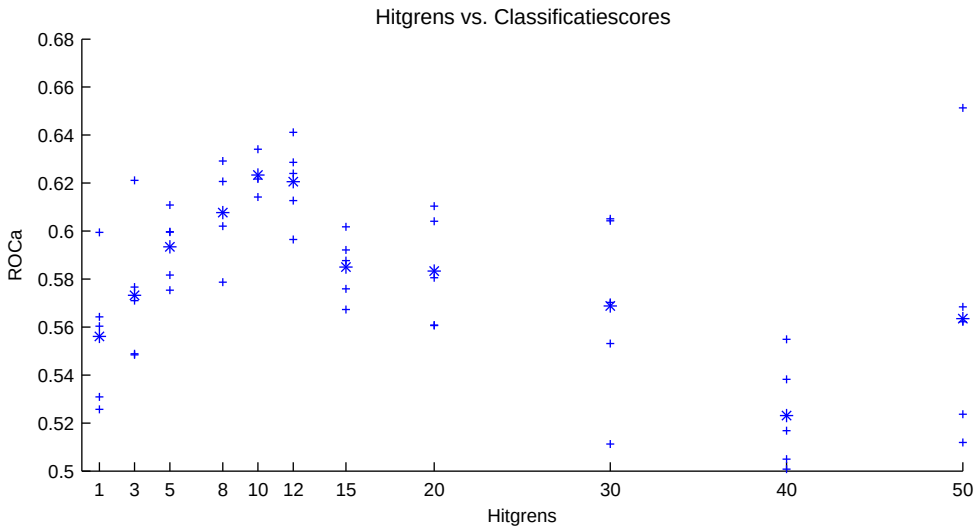
Als laatste experiment werd ervoor gekozen om de performantie van de classifier uit te zetten ten opzichte van de hitgrens. De resultaten van dit experiment worden gegeven in figuur 6.6. We zien dat de performantie afhangt van de hitgrens, maar gezien het verloop van de meetpunten is dit hoogstwaarschijnlijk te wijten aan het onevenwicht in de klassen. De performantie is hier immers maximaal op het punt dat de data in twee klassen van ongeveer gelijke grootte verdeeld zijn en daalt bij toenemend onevenwicht.



Figuur 6.4: Evaluatie van de performantie van de internetfeatures. Deze werden geevalueerd samen met een referentieaudiofeature, nl. de referentie markov-feature. De asteriks geeft de gemiddelde waarde aan en we zien duidelijk dat de internetfeatures een meerwaarde betekenen.



Figuur 6.5: Vergelijking van de performantie van de gecombineerde audiofeatures ten opzichte van de deelfeatures



Figuur 6.6: Evaluatie van de classificatie performantie in functie van de hitgrens

Hoofdstuk 7

Conclusie

Een experiment werd opgezet om de modelleerbaarheid van hitmuziek te testen. Hiervoor werd eerst een database ontworpen, die nadien gelinkt werd met de engelse UK Singles chart. Vervolgens werd een set van features gedefiniëerd, zowel gebaseerd op de muziekdata zelf als op community metadata gedatamined van het internet. Deze features werden dan gebruikt om enkele classificatieexperimenten uit te voeren.

Gebaseerd op de resultaten kan gezegd worden dat de stelling dat hitlijsten gemodelleerd kunnen worden juist is. Hoewel de classificatie met een gemiddelde ROCa waarde van 0.64 nog verre van correct is geeft het toch aan dat hits een gemeenschappelijke factor delen. Deze conclusie wordt nog verder gevalideerd als gekeken wordt naar de gebruikte database die een brede sampling is van zowel verschillende jaren als stijlen en daardoor toch als een vrij 'harde' database om te modelleren bestempeld kan worden.

Naast dit hoofdexperiment werden nog enkele andere experimenten gedaan met verschillende features en kernels om de classificatieperformantie te verbeteren. Hoewel de resultaten van deze experimenten nog niet helemaal conclusief zijn, kan toch reeds gesteld worden dat de gedane aanpassingen, de internetfeatures, de modellering van de audiofeatures en het gebruik van specifieke kernels, de classificatieresultaten verbeteren.

Naast deze classificatieresultaten is misschien de belangrijkste verwezelijking van deze thesis wel het opzetten van een structuur om grootschalige experimenten met muziek uit te voeren, wat gezien de computationele vereisten, zowel op het vlak van processing power als op IO, verre van triviaal is.

Bibliografie

[1] J. Suykens, "Lesnotas least squares support vector machines 2007-2008."

[2] C. Cortes en V. Vapnik, "Support vector networks," in *Machine Learning*, 1995, pp. 273–297.

[3] F. I. McKay C., McEnnis Daniel, "A large publicly accessible prototype audio database for music research." *McGill University*, 2006.

[4] [Online beschikbaar]: Rechtenvanbibliotheken,mediatheken, archievenenmuseaindeinformatiesamenleving-9november2001;http://www.vvbad.be/node/151

[5] [Online beschikbaar]: DeKamerkeurtdenieuweauteurswetgoed;http://www.vvbad.be/node/141

[6] A. Berenzweig, B. Logan, D. P. W. Ellis, en B. Whitman, "A large-scale evaluation of acoustic and subjective music similarity measures," in *Computer Music Journal*, 2003, pp. 99–105.

[7] [Online beschikbaar]: http://www.exactaudiocopy.de/en/index.php/overview/basic-technology/extraction-technology/

[8] [Online beschikbaar]: http://www.hydrogenaudio.org/forums/index.php

[9]

[10] S. Sigurdsson, K. B. Petersen, en T. Lehn-Schiøler, "Mel frequency cepstral coefficients: An evaluation of robustness of mp3 encoded music," in *Proceedings of the Seventh International Conference on Music Information Retrieval (ISMIR)*, 2006. [Online beschikbaar]: http://www2.imm.dtu.dk/pubdb/p.php?4690

[11] [Online beschikbaar]: http://wiki.hydrogenaudio.org/index.php?title=Lossless_comparison

[12]

[13] B. Whitman, "Learning the meaning of music," Ph.D. dissertatie, Massachusetts Institute of Technology, MA, USA, June 2005. [Online beschikbaar]: https://dspace.mit.edu/bitstream/1721.1/32500/1/61896668.pdf

[14] G. Peeters., "A large set of audio features for sound description (similarity and classification) in the cuidado project," 2003. [Online beschikbaar]: http://www.ircam.fr/anasyn/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf.

[15] B. J. C. P. G. E. H. P. L. A. Gouyon f., Amatriain X., *Sound to Sense: Sense to Sound: A State-of-the-Art. Version 0.1.* S2S2 Consortium, Florence., 2005, hfdst. 4: Content processing of musical audio signals.

[16] P. K. E. P. G. Widmer, S. Dixon en T. Pohle., *Sound to Sense: Sense to Sound: A State-of-the-Art. Version 0.1.* S2S2 Consortium, Florence., 2005, hfdst. 5 :From Sound to "Sense" via Feature Extraction and Machine Learning: Deriving High-level Descriptors for Characterising Music.

[17] X. Zhang en W. R. Zbigniew, "Analysis of sound features for music timbre recognition," in *MUE '07: Proceedings of the 2007 International Conference on Multimedia and Ubiquitous Engineering.* Washington, DC, USA: IEEE Computer Society, 2007, pp. 3–8.

[18] E. Micheli-Tzanakou, A. Ademoglu, en C. Enderwick, "Other types of feature extraction methods," pp. 109–132, 2000.

[19] A. A. Livshin en X. Rodet, "Musical instrument identification in continuous recordings," in *in 7th International Conference on Digital Audio Effects (DAFX-4*, 2004, pp. 222–227.

[20] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *In International Symposium on Music Information Retrieval*, 2000.

[21] E. Pampalk, A. Flexer, en G. Widmer, "Improvements of audio-based music similarity and genre classification," 2005.

[22] T. Pohle, E. Pampalk, en G. Widmer, "Evaluation of frequently used audio features for classification of music into perceptual categories," in *Proceedings of the Fourth International Workshop on Content-Based Multimedia Indexing*, 2005.

[23] B. L. Ruth Dhanaraj, "Automatic prediction of hit songs," August 2005. [Online beschikbaar]: http://www.hpl.hp.com/techreports/2005/HPL-2005-149.pdf

[24] W. D. h. Xavier Rodet, "Discrete cepstrum coefficients as perceptual features," 2003.

[25] C. Tzanetakis, "Musical genre classification of audio signals," *IEEE Tr. Speech and Audio Processing*, vol. 10(5), pp. 293–302, 2002.

[26] *MIRToolbox 1.0 User Guide.*

[27] G. K. C. e. o. v. M. i. o. t. s. v. t. i. P. o. t. S.-. v. . . p. .-. T. Ganchev, N. Fakotakis, "Comparative evaluation of various mfcc implementations on the speaker verification task." *Proceedings of the SPECOM*, vol. 1, pp. 191–194, 2005.

[28] H. Hermansky, "Perceptual linear prediction (plp) analysis for speech," *J. Acoust. Soc. Am.*, vol. 87, pp. 1738–1752, 1990.

[29] B. A. Turlach, "Bandwidth selection in kernel density estimation: A review," in *CORE and Institut de Statistique*, 1993, pp. 23–493.

[30] C. Elkan, "Using the triangle inequality to accelerate k-means," 2003.

[31] P. Knees, E. Pampalk, en G. Widmer, "Artist classification with web-based data." in *ISMIR*, 2004. [Online beschikbaar]: http://dblp.uni-trier.de/db/conf/ismir/ismir2004.html

[32] M. Schedl, T. Pohle, P. Knees, en G. Widmer, "Assigning and visualizing music genres by web-based co-occurrence analysis," in *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR'06)*, Victoria, British Columbia, Canada, October 2006.

[33] R. L. Cilibrasi en P. M. B. Vitanyi, "The google similarity distance," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, nr. 3, pp. 370–383, March 2007. [Online beschikbaar]: http://portal.acm.org/citation.cfm?id=1263333